NPAM predictive model improvements and piloting NPAM on partner lands

Clinton T. Moore[1]*, Jill J. Gannon*[2]*, Terry L. Shaffer*[3]

[1] U.S. Geological Survey, Georgia Cooperative Fish and Wildlife Research Unit

[2] U.S. Fish and Wildlife Service

[3] U.S. Geological Survey, Northern Prairie Wildlife Research Center

Cooperator Science Series # 130-2018

About the Cooperator Science Series:

The Cooperator Science Series was initiated in 2013. Its purpose is to facilitate the archiving and retrieval of research project reports resulting primarily from investigations supported by the U.S. Fish and Wildlife Service (FWS), particularly the Wildlife and Sport Fish Restoration Program. The online format was selected to provide immediate access to science reports for FWS, state and tribal management agencies, the conservation community, and the public at large.

All reports in this series have been subjected to a peer review process consistent with the agencies and entities conducting the research. For U.S. Geological Survey authors, the peer review process (http://www.usgs.gov/usgs-manual/500/502-3.html) also includes review by a bureau approving official prior to dissemination. Authors and/or agencies/institutions providing these reports are solely responsible for their content. The FWS does not provide editorial or technical review of these reports. Comments and other correspondence on reports in this series should be directed to the report authors or agencies/institutions. In most cases, reports published in this series are preliminary to publication, in the current or revised format, in peer reviewed scientific literature. Results and interpretation of data contained within reports may be revised following further peer review or availability of additional data and/or analyses prior to publication in the scientific literature.

The Cooperator Science Series is supported and maintained by the FWS, National Conservation Training Center at Shepherdstown, WV. The series is sequentially numbered with the publication year appended for reference and started with Report No. 101-2013. Various other numbering systems have been used by the FWS for similar, but now discontinued report series. Starting with No. 101 for the current series is intended to avoid any confusion with earlier report numbers.

Contractual References:

This document (USGS IPDS #: IP-090144) was developed in conjunction with the US Fish and Wildlife Service through funding provided through the Refuge Cooperative Research Program.

Recommended citation:

Moore, C. T., J. J. Gannon, and T. L. Shaffer. 2018. NPAM predictive model improvements and piloting NPAM on partner lands**.** U.S. Department of Interior, Fish and Wildlife Service, Cooperator Science Series FWS/CSS-130-2018, Washington, D. C. https://doi.org/ 10.3996/css74130649

For additional copies or information, contact:

Clinton Moore
U.S. Geological Survey
Georgia Cooperative Fish and Wildlife Research Unit
E-mail: cmoore@usgs.gov

NPAM PREDICTIVE MODEL IMPROVEMENTS AND PILOTING NPAM ON PARTNER LANDS

FINAL REPORT

Clinton T. Moore, Jill J. Gannon, Terry L. Shaffer
U.S. Geological Survey

Submitted to USGS Ecosystems Mission Area in fulfillment of funding provided through the Refuge Cooperative Research Program

31 January 2018

CONTENTS

ACKNOWLEDGMENTS

We thank Cami Dixon and Jennifer Zorn of the U.S. Fish and Wildlife Service who provided data, background information, and responses to our many queries over the course of our work. We appreciate the assistance of Jeff Printz (USDA Natural Resources Conservation Service) and David Pyke (U.S. Geological Survey) who provided expert guidance on the use of Ecological Site Descriptions. We extend thanks to the professionals from natural resource agencies and non-governmental organizations who responded to our inquiries about piloting NPAM on partner lands. We thank the U.S. Fish and Wildlife Service members of the NPAM Advisory Team (Cami Dixon, Pauline Drobney, Vanessa Fields, Todd Frerichs, Todd Grant, Sara Vacek, Jennifer Zorn) for their guidance and wisdom throughout the work. None of our work was possible without the efforts of scores of dedicated U.S. Fish and Wildlife Service personnel who carry out management on native prairie units of the National Wildlife Refuge System and carefully monitor plant community responses. We thank Matthew Andersen and Lianne Ball (both U.S. Geological Survey) for arranging funding through the USGS Ecosystems Mission Area and Refuge Cooperative Research Programs. Robin Russell and Max Post van der Burg reviewed the report and provided helpful suggestions for its improvement. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

INTRODUCTION

The Native Prairie Adaptive Management (NPAM) program was jointly developed by the U.S. Geological Survey (USGS) and U.S. Fish and Wildlife Service (USFWS) in 2009 to guide habitat management decisions under a framework that reduces uncertainty about effects of actions. The NPAM program addresses the management of vegetation on units of native prairie owned by the USFWS National Wildlife Refuge System (NWRS) in the northern Great Plains. Across the region, aggressive invasions of introduced cool-season grasses, principally smooth brome (*Bromus inermis*) and Kentucky bluegrass (*Poa pratensis*), have reduced prairie biodiversity and degraded habitat quality for many wildlife taxa. Native prairies were shaped by historic defoliation disturbances (fire, ungulate grazing). Invasions by introduced grasses on NWRS units occurred over several decades, coinciding with NWRS management policies that largely excluded defoliation to promote dense cover habitat for nesting waterfowl (Murphy and Grant 2005). Managers of NWRS units believe that reintroduction of these disturbances offer the greatest hope of confronting these invasions and reversing declines in biodiversity. However, there is no clear insight or consensus as to which actions to take, or how current invasion conditions affect that choice. The NPAM program provides that guidance, conditional on current level of understanding about biological response to actions, using a modeling and monitoring structure to update understanding through time (Gannon et al. 2013).

The NPAM program is built on sets of models that make predictions about vegetation response to management actions carried out on an annual decision cycle. Each model within the set expresses a specific hypothesis about how the response to an action is sensitive (or not) to conditions such as the grass species that is the dominant invader, the defoliation history of the unit, current level of invasion, substitutability of one action for another, etc. After actions are taken, site monitoring yields data that inform about the relative predictive performance of the models. System learning is achieved as models gain or lose predictive credibility, and the gain in knowledge is applied to future decision guidance. The NPAM program operates on both mixed-grass and tallgrass prairies, but distinct sets of decision alternatives and predictive models apply to each grassland type to accommodate differences in native prairie community structure and in conventional management practices and to facilitate learning around hypotheses unique to these types. The NPAM program has been providing decision guidance each year since 2010. As of 2017, 128 units were enrolled in NPAM, representing 20 NWRS refuges or complexes across four States and two USFWS regions.

Development of NPAM was daunted by technical challenges that required important concessions to be made at the outset. First, little or no data could be found to inform the parameterization of the models. Consequently, the predictive models were developed through an expert elicitation exercise involving USFWS members of the development team (Gannon et al. 2013). Their responses were used in place of empirical data to quantify the models. The approach was accepted as a satisfactory measure to initiate the program, conditional on a commitment to revisit the models when data emerged under the program. Second, there was widespread agreement among team members that abiotic factors such as site characteristics, weather conditions, and climate patterns likely influence vegetation response to actions. Again, no data existed to explore or to quantify these hypothesized relationships. Thus, these questions were set aside during development of NPAM, with a commitment to explore them – and possibly incorporate them into the decision framework – at a later time. Third, the development team saw value in extending NPAM’s reach beyond NWRS so that a region-wide, broad-based partnership effort could be mounted to confront the invasion threat to native prairies and to more rapidly learn about effective practices. However, constraining the program to NWRS units, at least in the first years, was recognized as the most prudent strategy for developing and testing the approach.

In 2012, we received support under the USGS-USFWS Refuge Cooperative Research Program to focus on the following areas:

1) investigate the replacement of expert-derived model parameters with quantities estimated directly from NPAM data collected over the period 2010-2016;
2) use data from NPAM and other sources to construct models of response to management actions and abiotic drivers, and assess the implications of incorporating abiotic factors in the NPAM framework; and
3) explore the potential of broadening the NPAM network to participants outside of NWRS.

This report summarizes these efforts and is divided into chapters corresponding to these research focus areas.

CHAPTER 1: Empirical Parameterization of NPAM Conceptual Models

Summary:
Predictive models operate at the core of NPAM, and they quantitatively express biological relationships that define each competing conceptual model. However, the lack of data at the inception of NPAM required the use of expert judgment to parameterize models for both mixed grass and tallgrass systems. As of 2016, seven cycles of decision making and monitoring have been completed in the mixed grass system, and five (with treatments structured around the concept of the cool-season phenological window) have been completed in the tallgrass system. We constructed linear models with terms and constraints that approximated each of the competing conceptual hypotheses under which NPAM operates, and we fit these models to field data. Our goal was to derive empirical versions of the same models initially quantified through expert judgment. We propose a way to produce a set of composite models that average predictions by the two approaches, relying more on predictions by the expert-parameterized version for places in the state space where data remain sparse. In the future, our analyses can be replicated with additional years of data and the composite models can be re-computed, each time diminishing the influence of expert-parameterized quantities in the outcome.

METHODS

Data preparation

The NPAM program is conducted across approximately 120 management units (the number varies slightly each year) of the U.S. Fish and Wildlife Service National Wildlife Refuge System in Minnesota, North Dakota, South Dakota, and Montana. Management units are parcels of unbroken, native sod with invasions by two introduced cool-season grasses, smooth brome and Kentucky bluegrass. The program objective of NPAM is to arrest and reverse invasions on all units. Most units (~80) occur in the western, drier mixed grass prairie, and the rest occur in the eastern, wetter tallgrass prairie. Units are monitored annually for vegetation composition. Between monitoring episodes, a treatment is applied and recorded. The treatment alternatives vary by grass type. In mixed grass units, treatment alternatives in place since the 2009-2010 management cycle are Rest, Graze, Burn, and Burn/Graze combination. In tallgrass units, treatment alternatives used since the 2011-2012 cycle are Rest, Graze within a defined window based on smooth brome flowering phenology (Graze/Window), Burn within the phenological window (Burn/Window), and Defoliate (haying at any time of the year, or burning or grazing outside of the phenological window). Descriptions of treatments and monitoring protocols are found in Gannon et al. (2013).

Under the NPAM monitoring program, 25-meter monitoring transects are randomly distributed across the unit, at an approximate density of 0.5/ha and separated by at least 65m. The belt transect sampling approach (Grant et al. 2004) is used, where the dominant type of vegetation is recorded within every 0.5-m segment along the transect. For our analyses, we aggregated vegetation types into four classes: native grasses and forbs, smooth brome (SB), Kentucky bluegrass (KB), and remainder (RM; other non-native vegetation or native trees and shrubs). For each transect ($j = 1, 2, \ldots, J_i$) in unit i in year t, we tabulated the frequency of occurrence of each vegetation class over all segments, yielding counts $n_{ijt,\mathrm{NP}}$, $n_{ijt,\mathrm{SB}}$, $n_{ijt,\mathrm{KB}}$, and $n_{ijt,\mathrm{RM}}$, respectively.

A defoliation index, D_{it}, was assigned to each unit *i* in year *t* to reflect the amount of recent management disturbance. The index was the number of years (out of the preceding seven years) in which any form of defoliation actions occurred on the unit, multiplied by a weighting factor negatively related to number of years since the most recent defoliation action. Thus, D_{it} integrated frequency of defoliation with recentness of its occurrence.

Linear Model Estimation

We constructed a set of year and unit-level summary data of vegetation composition for analysis. For each unit and year combination, we used the transect-level component counts ($n_{ijt,\mathrm{NP}}$, $n_{ijt,\mathrm{SB}}$, $n_{ijt,\mathrm{KB}}$, $n_{ijt,\mathrm{RM}}$) to estimate the unit-level component proportions $\underline{p}_{it} = (p_{it,\mathrm{NP}}, p_{it,\mathrm{SB}}, p_{it,\mathrm{KB}}, p_{it,\mathrm{RM}})$ under a Dirichlet-multinomial model. We considered this vector to represent the vegetation composition of management unit *i* in year *t*. Furthermore, because means under the Dirichlet distribution are strictly >0, the estimation procedure produced non-zero values of the proportion vector, permitting the formation of ratios among the components for analysis in linear models.

To make the $\underline{p}_{it}$ usable as outcomes in normal-response linear models, we created a set of "pivot logits": $r_{it,\mathrm{SB}} = \log(p_{it,\mathrm{SB}}/p_{it,\mathrm{NP}})$, $r_{it,\mathrm{KB}} = \log(p_{it,\mathrm{KB}}/p_{it,\mathrm{NP}})$, $r_{it,\mathrm{RM}} = \log(p_{it,\mathrm{RM}}/p_{it,\mathrm{NP}})$. That is, we formulated linear models in terms of change in each of the undesired components (SB, KB, RM) relative to change of native grasses and forbs.

In the foregoing, we assume that the time index increments by 1 immediately following an action and before prairie measurement; thus time index *t*+1 refers to the plant response occurring after a measurement of conditions and a subsequent action in year *t*. We obtained predictor variables for treatment (A_{it}), dominant invader type (V_{it}), defoliation index (D_{it}), and composition level of native grasses and forbs (N_{it}). The competing hypotheses represented in NPAM are formed based on the influence or absence of these variables. In mixed grass units, A_{it} = {Rest, Graze, Burn, Burn/Graze}. In tallgrass units, A_{it} = {Rest, Graze within Window, Burn within Window, Defoliate}. We created the predictor variable N_{it} as $\log(p_{it,\mathrm{NP}}/[1\text{-}p_{it,\mathrm{NP}}])$. We used a set of dominance rules based on the composition vector $\underline{p}_{it}$ to assign a type of dominance by one of the undesired components (SB, KB, or RM), or of codominance (CO) between SB and KB:

$$V_{it} = \begin{cases} \mathrm{SB}, & \text{if } p_{it,\mathrm{RM}}/(1-p_{it,\mathrm{NP}}) < 2/3 \quad \text{and} \quad p_{it,\mathrm{SB}}/(p_{it,\mathrm{SB}}+p_{it,\mathrm{KB}}) \ge 2/3 \\ \mathrm{CO}, & \text{if } p_{it,\mathrm{RM}}/(1-p_{it,\mathrm{NP}}) < 2/3 \quad \text{and} \quad p_{it,\mathrm{SB}}/(p_{it,\mathrm{SB}}+p_{it,\mathrm{KB}}) < 2/3 \quad \text{and} \quad p_{it,\mathrm{KB}}/(p_{it,\mathrm{SB}}+p_{it,\mathrm{KB}}) < 2/3 \\ \mathrm{KB}, & \text{if } p_{it,\mathrm{RM}}/(1-p_{it,\mathrm{NP}}) < 2/3 \quad \text{and} \quad p_{it,\mathrm{KB}}/(p_{it,\mathrm{SB}}+p_{it,\mathrm{KB}}) \ge 2/3 \\ \mathrm{RM}, & \text{if } p_{it,\mathrm{RM}}/(1-p_{it,\mathrm{NP}}) \ge 2/3 \end{cases}$$

For the pivot logit for component *q* (*q*={SB, KB, RM}) in post-treatment year *t*+1, we fit a global linear model with the predictor variables A_{it}, V_{it}, D_{it}, N_{it}, the pivot logit for component *q* for the pre-treatment year *t*, r_{itq}, and certain interactions among these effects:

$$r_{i,t+1,q} = \beta_{q0} + \beta_{q1} r_{itq} + \tau_q^{(A)} + \delta_q^{(V)} + (\tau\delta)_q^{(AV)} + \gamma_q D_{it} + (\tau\gamma)_q^{(A)} D_{it} + (\delta\gamma)_q^{(V)} D_{it} + (\tau\delta\gamma)_q^{(AV)} D_{it} + \alpha_q N_{it} + (\tau\alpha)_q^{(A)} N_{it} + \varepsilon_{itq} .$$

Here,

β_{q0} is the model intercept,
β_{q1} is the slope term for the lagged value of component *q*,
$\tau_q^{(A)}$ is the additive effect of treatment *A* on component *q*,

$\delta_q^{(V)}$ is the additive effect of dominant invader type V on component q,
γ_q is the slope term for the defoliation index predictor,
α_q is the slope term for the composition level of native grasses and forbs, and
ε_{itq} is the additive unexplained error.

The model included two-way interaction effects between:

treatment and dominant invader type ($\tau\delta$),
treatment and defoliation index ($\tau\gamma$),
treatment and amount of native grasses and forbs ($\tau\alpha$), and
dominant invader type and defoliation index ($\delta\gamma$).

The model included one three-way interaction between treatment, dominant invader type, and defoliation index ($\tau\delta\gamma$).

The global model was formulated to approximate the structure of the most complex model used in NPAM for either grass type, Model 4. Model 4 posits the following regarding the vegetation state response of the unit to management actions: (1) that the response depends on the type of invader that is dominant (i.e., treatments are differentially effective depending on the dominant invader type); (2) that the response depends on the history of defoliation of the unit (i.e., treatments are differentially effective according to whether the unit has had recent, frequent management disturbance); and (3) that the response depends on overall level of invasion (specifically, treatment efficacy is reduced with increase in invasion level). For implementation in NPAM, Model 4 was parameterized based on results of an expert elicitation exercise, in combination with post hoc adjustment of values to achieve the hypothesized interactive patterns (Gannon et al. 2013: Appendices 3-5).

Because all other models in NPAM (mixed grass Models 1-3; tallgrass Models 1-3 and 5-6) are successive simplifications of Model 4, we produced approximate forms of those models by imposing selected zero constraints on the global model. Under Model 3, the dependency of response to the overall level of invasion is not recognized. Thus, in the global model, we imposed the constraints $\alpha_q = 0$ and $(\tau\alpha)_q^{(A)} = 0$ to produce a structure approximating Model 3. Under Model 2, the dependency of response is not dependent on overall level of invasion or on defoliation history. Therefore, in addition to those constraints imposed for Model 3, we imposed the constraints $\gamma_q = 0$, $(\tau\gamma)_q^{(A)} = 0$, $(\delta\gamma)_q^{(V)} = 0$, and $(\tau\delta\gamma)_q^{(AV)} = 0$ to produce a structure that approximated Model 2. Finally, under Model 1, response is not sensitive to the form of defoliation treatment, only that some form of defoliation occurs. Therefore, in addition to those constraints imposed for Models 2 and 3, we imposed the constraints $\delta_q^{(V)} = 0$ and $(\tau\delta)_q^{(AV)} = 0$ to produce a model approximating Model 1.

Within each of these models, we placed additional equality constraints on specific parameters under certain conditions to approximately replicate finer-grain assumptions within the NPAM models (Tables 1.1, 1.2). For either grass type, Model 1 carries an assumption that vegetation response is sensitive only to the application of a defoliation treatment versus rest; however, the model assumes that all forms of defoliation are equivalent.

Models 2-4 for mixed grass units carry response-specific constraints that are enforced depending on the response being modeled (SB, KB, or RM). Under Model 2, the SB response in SB-dominated units is assumed to be equivalent under Graze or Burn/Graze treatments, the KB response in KB-dominated units is assumed to be equivalent under Burn or Burn/Graze treatments, and the RM response in RM-dominated units is assumed to be equivalent under any form of defoliation treatment. Furthermore, in co-dominated units, the Graze and Burn treatments are assumed to have the same effect whether the response is SB or KB (Table 1.1). Under Models 3 and 4 for mixed grass units, we constrained the relationship between treatment and defoliation history to be equivalent for any form of active treatment (Burn, Graze, or Burn/Graze), but we permitted that relationship to be different for the inactive (Rest) treatment (Table 1.1). As under Model 2, we set constraints according to the response being modeled (Table 1.1).

Response-specific constraints were similarly imposed for tallgrass Models 2-4. Under Model 2, the RM response in RM-dominated units is assumed to be equivalent under any form of defoliation treatment (Table 1.2). Constraints under Models 3 and 4 were exactly analogous to those for mixed grass. That is, we constrained the relationship between treatment and defoliation history to be equivalent for any form of active treatment (Burn/Window, Graze/Window, or Defoliate), but we permitted that relationship to be different for the inactive (Rest) treatment (Table 1.2).

We produced Models 5 and 6 for tallgrass units by varying response-specific constraints imposed on tallgrass Model 3. Model 5 posits the hypothesis that under SB dominance, grazing within the phenological window is as effective on the SB response as is burning within the window. All other constraints remained as set in Model 3 (Table 1.2). Model 6 hypothesizes that for any invader, the Defoliation treatment is no more effective than Rest (Table 1.2).

Finally, in models that included the three-way interaction between treatment, dominant invader type, and defoliation index ($\tau\delta\gamma$), we specified additional equality constraints to produce estimable functions for effects supported by insufficient data (Table 1.3). For example, in mixed grass, only a single data value was available for estimating the linear relationship between response and defoliation level within the Burn-RM treatment-invader type combination.

We fit each approximating model (1-4 for mixed grass, 1-6 for tallgrass) to each set of response ($r_{it,\mathrm{SB}}$, $r_{it,\mathrm{KB}}$, $r_{it,\mathrm{RM}}$) data using PROC REG in SAS (SAS Institute 2013), employing a means-effect expression of the model and suppressing the intercept term in the design matrix. We computed AIC (Akaike 1974; Burnham and Anderson 2002) for each model to evaluate its parsimony for the data.

Expression of linear models in state and transition format

Predictions of vegetation response under NPAM are provided by state and transition models. Following monitoring, each management unit is assigned to one of 16 vegetation states ($S_t = \{1, 2, \ldots, 16\}$), corresponding to four levels of coverage by native grasses and forbs (0-30%, 30-45%, 45-60%, and 60-100%) in combination with four types of invader dominance (SB, CO, KB, RM). The models are probabilities of transition from a given state i at time t to each state j, $j=\{1, \ldots, 16\}$ at time $t+1$, conditional on the type of action taken and three levels of defoliation index, L_t = {"Low" ($D_t < 2$), "Medium" ($2 \leq D_t < 4$), "High" ($D_t \geq 4$}. For alternative model m, m = M1-M4 (mixed grass) or m = T1-T6 (tallgrass), the probability of transition from state i to state j is:

$$\Pr_m(S_{t+1}=\{j\} \mid S_t=\{i\}, A_t, L_t),\ i, j = 1, \ldots, 16.$$

We used a Monte Carlo procedure to express the linear models in a state and transition model format for each grass type (mixed grass, tallgrass). The procedure was structured to reflect variation due to uncertainty about true system state (i.e., inherent uncertainty in a state and transition model brought about by the representation of continuous state metrics as discrete states) and uncertainty due to sampling variability in model parameter estimation. For each combination of vegetation state, defoliation level, and treatment (16 × 3 × 4 = 192 combinations), we drew 1,000 random values of vegetation composition $\underline{p}_t = (p_{t,\mathrm{NP}}, p_{t,\mathrm{SB}}, p_{t,\mathrm{KB}}, p_{t,\mathrm{RM}})$ and defoliation index D_t. Each random draw thus resulted in a set of predictor variables, $X_t = \{r_{tq}, A_t, V_t, D_t, N_t\}$ available as input to the linear models. We input the variables into each alternative linear model to predict a model-specific mean and variance (σ^2_{SB}, σ^2_{KB}, σ^2_{RM}) of each response at time t+1, $r_{t+1,\mathrm{SB}}$, $r_{t+1,\mathrm{KB}}$, and $r_{t+1,\mathrm{RM}}$, for the set of predictors. We then drew 500 sets of $r^*_{t+1,\mathrm{SB}}$, $r^*_{t+1,\mathrm{KB}}$, and $r^*_{t+1,\mathrm{RM}}$ values from independent normal distributions specified by the corresponding means and variances. We applied a back-transformation to each set of values to produce a sampled value of $\underline{p}_{t+1}^* = (p^*_{t+1,\mathrm{NP}}, p^*_{t+1,\mathrm{SB}}, p^*_{t+1,\mathrm{KB}}, p^*_{t+1,\mathrm{RM}})$. The procedure thus produced 500,000 values of $\underline{p}_{t+1}^*$ per combination of vegetation state, defoliation level, and treatment. We mapped each $\underline{p}_{t+1}^*$ into a state category (1-16: four levels of coverage by native grasses and forbs in combination with four types of invader dominance), and we tabulated the frequency of assignment into each state. The proportion of assignments among the states thus provided an empirical estimate of $\Pr_m(S_{t+1}=\{j\} \mid S_t=\{i\}, A_t, L_t)$ for model m.

We also obtained an average standard error of the prediction for each combination of model, defoliation level, action, and vegetation state, by first averaging the standard error metrics σ_{SB}, σ_{KB}, and σ_{RM} over the 1,000 draws, then obtaining the geometric average of those means. The resulting value, $\sigma_\bullet$, is a general inverse measure of the reliability of the empirical estimate of transition probability (smaller values of $\sigma_\bullet$ = greater precision). Low reliability can be due to scarcity of data at specific combinations of vegetation state, defoliation level, and action, but it can also be due to inherent variability in the type of response; for example, the RM response is highly dependent on what plant groups constitute that dominance category at any point in time. The value $\sigma_\bullet$ will decrease to the degree that more data become available for these conditions or as inherent variability can be reduced (e.g., finer scale measurement and modeling of the RM category). Greater precision can also be obtained through assumptions made in modeling. For example, by modeling defoliation level as a continuous covariate in the linear models rather than a categorical covariate, data are “borrowed” across levels of defoliation, yielding higher precision overall for this effect, particularly at moderate levels.

Composite empirical-expert models

The process just described produces an empirical estimate of transition probability given current state, action, and defoliation level for model m, $\Pr_m(S_{t+1}=\{j\} \mid S_t=\{i\}, A_t, L_t)$. At the initiation of NPAM, analogous transition probabilities $\Pr^E_m(S_{t+1}=\{j\} \mid S_t=\{i\}, A_t, L_t)$ were constructed through elicitation of expert judgment. Although an aspiration for NPAM is to one day rely on predictive models parameterized solely through field-derived, empirical data, the program is now in a relative stage of infancy in which experience has not been sufficient in all system states to generate reliable data, if any data at all. Therefore, a prudent course of action may be to average the predictions from empirical models and expert elicitation.

Letting $P = \Pr_m(S_{t+1}=\{j\} \mid S_t=\{i\}, A_t, L_t)$ and letting $P^E = \Pr^E_m(S_{t+1}=\{j\} \mid S_t=\{i\}, A_t, L_t)$, then an expression that combines the two forms of predictions into an average response $\underline{P}$ is

$$\underline{P} = (P^E + wP) / (w + 1),$$

where w is a relative weight for the empirically-derived probability ($0 \le w < \infty$). When $w = 1$, the expert-elicited and the empirical values are given equal weight. The empirically-derived value can be given greater or lesser weight by increasing or decreasing w, respectively. The expression can be rewritten to express $\underline{P}$ as a weighted average of P^E and P:

$$\begin{aligned}\underline{P} &= \left(1-\frac{w}{1-w}\right)P^{E}+\left(\frac{w}{1-w}\right)P\,, \\ &= (1-\theta)\,P^{E}+\theta\,P\end{aligned}$$

where θ is the fraction of weight (0-1) provided to the empirically-derived value.

As previously indicated, the estimates of $\Pr_m(S_{t+1}=\{j\} \mid S_t=\{i\}, A_t, L_t)$ do not have equal precision due to the imbalance of data across combinations of states and treatments. This argues for weighting empirical values proportional to their precision. The choice of approach is subjective, but one possible scaling reflects the standard error of the prediction relative to the full range of prediction standard error across all models and both grass types. As the range is exaggerated due to strong right-skew in the distribution of standard errors, logarithmic transformation is appropriate to apply in the scaling formula:

$$s'=\frac{\log(\sigma_{\bullet})-\log(\sigma_{\bullet}^{\text{MIN}})}{\log(\sigma_{\bullet}^{\text{MAX}})-\log(\sigma_{\bullet}^{\text{MIN}})}\,.$$

Here, $\sigma_{\bullet}$ is the standard error of the prediction associated with P, $\sigma_{\bullet}^{\text{MIN}}$ is the minimum value of $\sigma_{\bullet}$ observed across all competing models, all state-treatment combinations, and over both grass types, and $\sigma_{\bullet}^{\text{MAX}}$ is the corresponding maximum value of $\sigma_{\bullet}$. The minimum value $s' = 0$ is produced when the standard error of the prediction matches the smallest value of standard error across all models and grass types. Conversely, the maximum value $s' = 1$ corresponds to the standard error matching the largest value observed across all models and grass types.

Smaller values of s' are reflective of higher relative precision for the prediction, and they justify placing greater weight on the empirical estimate in the computation of $\underline{P}$. On the other hand, greater values of s' suggest that little confidence can be placed in the empirical quantity, and that reducing the influence of the expert-elicited value in the computation of $\underline{P}$ introduces risk of producing an unreliable composite estimate. Assigning degree of weight to the expert-elicited and empirical components of the composite measure is a subjective determination, but under the following premises, weights can be objectively assigned:

1) For the empirical estimate with highest precision (i.e., where $\sigma_{\bullet} = \sigma_{\bullet}^{\text{MIN}}$), assign full weight ($\theta = 1$) to the empirical quantity; and
2) For the empirical estimate with least precision (i.e., where $\sigma_{\bullet} = \sigma_{\bullet}^{\text{MAX}}$), assign a minimum weight $\theta = \theta^{*}$ to the empirical quantity, and assign weight 1- θ^{*} to the expert-elicited quantity.

For all other empirical estimates, an intermediate weight (between $\theta = 0$ and $\theta = \theta^{*}$) proportional to s' is assigned as follows:

$$\theta=\frac{1}{1+s'(1-\theta^{*})/\theta^{*}}\,.$$

All programs and scripts used to conduct these analyses are collected in the program code archive (Appendix G).

RESULTS

Linear model estimation

In the mixed grass system, we obtained *n* = 541 consecutive year-pairs with valid treatment applications, and we obtained *n* = 231 of the same for the tallgrass system. For mixed grass models, Model 4 received greatest AIC weight for the SB and KB responses, whereas Model 2 received greatest weight for the RM response (Table 1.4). For tallgrass models, Model 1 received greatest AIC weight for all responses (Table 1.4).

Expression of linear models in state and transition format

Results of the simulation work are easiest to visualize on a single slice of one of the complete state and transition models. For example, consider the portion of the state and transition model representing mixed grass Model 2, for the condition of combination burn/graze treatment at intermediate defoliation (2 ≤ index ≤ 4) (Fig. 1.1). Each row of this matrix is the set of probabilities $\Pr_{m=\mathrm{M2}}(S_{t+1}=\{j\} \mid S_t=\{i\}, A_t =$ Burn/Graze, L_t = index 2-4) for vegetation state *i* (*i* = 1,…, 16). Notice that for the highest (60-100%) and lowest (0-30%) levels of native prairie cover, the empirical estimates for this model place much of the probability on the 4x4 diagonal of the matrix, which means that the treatment under these conditions largely maintains the native cover as is. However, in intermediate conditions of native prairie cover (30-45%, 45-60%), there is evidence that the burn/graze treatment led to improvement in condition (i.e., increased native cover, indicated by dark-shaded blue squares to the left of the diagonal). But the improvement appeared to be invader-state dependent: improvement was observed for sites dominated or co-dominated by KB (rows 6-7, 10-11), but sites dominated by SB exhibited either status quo or some chance of degradation (rows 5, 9).

Placing the matrix next to the corresponding matrix for expert-elicited values illustrates differences between the way the treatment was envisioned to affect vegetation composition under the given model and the empirical pattern of response revealed by the data (Fig. 1.2). The matrices in this figure are again for mixed grass Model 2, for the condition of the combination burn/graze treatment at intermediate defoliation. The leftmost matrix (red) represents the array of transition probabilities as predicted by the experts, and they serve as the probabilities currently in use within NPAM. The middle matrix (blue) represents the empirically-estimated probabilities; it is the same matrix shown in the previous figure (Fig. 1.1). Patterns in probability based on expert elicitation were relatively constrained within the state transition space compared with empirical patterns. Experts predicted high degrees of transition among invasive dominance classes within a native prairie cover level (i.e., the 4x4 diagonal), but increases in native cover level (movements to the left of the 4x4 diagonal) were thought to be rare, and decreases in native cover level (movements to the right of the 4x4 diagonal) were predicted to be practically non-existent. The empirical estimates revealed that improvement in native cover was quite likely for certain vegetation conditions, and degradation in native cover could also occur by this treatment.

Rows within the empirical matrix are not estimated with equal precision. The column of gray cells to the left of the empirical matrix represents (through degree of shading) the relative amount of precision for the estimated set of probabilities. In this matrix (Fig. 1.2), there is a 3-fold difference between the minimum (0.18) and maximum (0.55) standard error. Differences may be due to the amount of data available for each vegetation state for this treatment and defoliation level combination, the inherent variability in the response related to the type of dominant invader, or some combination of both. For the burn/graze combination treatment at intermediate defoliation level, precision is smaller for units dominated by RM, whether because of sparse data or high inherent variability; thus, cells in the "Precision" column corresponding to these states have light shading (Fig. 1.2).

Composite empirical-expert models

Information from the expert-elicited and empirically-derived versions of the model may be averaged to yield a composite version of the model (green matrix). Weights (θ, 1-θ) are applied to the empirical and expert-elicited probability values, respectively. In this example (Fig. 1.2), for purposes of illustration, the minimum weight given to an empirical probability, θ^*, was arbitrarily set to 0.2. However, because no empirically-derived prediction standard error for these conditions was as great as the largest standard error across all models and grass types (8.8), no empirical probability value in this table received the minimum weight of 0.2. Similarly, because no prediction standard error was as small as the smallest standard error across all models and grass types (0.06), no empirical probability value was given full weight, either. Instead, the range of weight provided to the empirical probability was θ = (0.36, 0.54) corresponding to conditions {NP 60-100, RM dominated} and {NP 30-45, KB dominated}, respectively. Effects of the differential weighting due to precision can be observed in the table. For example, the transition for state {NP 45-60, KB dominated} (row 7) was estimated with relatively high precision (SE = 0.18). The set of composite probabilities (green matrix) for this state more closely resemble the empirical probabilities (blue) than the expert-elicited probabilities (red). Conversely, the transition for state {NP 45-60, RM dominated} (row 8) was estimated with poor precision (SE = 0.54), with no clear pattern in vegetation response. Thus, the composite probabilities bear close resemblance to the expert-elicited values.

The linked workbook[1] provides the trio of expert-elicited, empirically-derived, and composite versions of the full set of state and transition models, using the averaging formula above and given a user-specified value of θ^*. Note that the selection of weight for θ^* shown in the workbook is subjective, and there may be arguments for using a value other than 0.5. The incorporation of standard error in the averaging formula is also subjective, but the approach used here is straightforward and has a reasonable basis.

The workbook is organized with one NPAM grass type model per sheet. On each sheet, 12 trios of matrices are displayed, with the 4 treatments arranged in blocks of rows, and the 3 defoliation levels arranged in blocks of columns. With regard to the precision columns, shades are darkest for the simplest models (e.g., Model 1 for both grass types requires estimation of the fewest parameters) and where data are abundant (e.g., rest treatments, conditions of low defoliation history, states where RM is not dominant). At the bottom of each sheet is a user-controlled slider that adjusts the value of the minimum weight θ^* provided to the empirical probability corresponding to the least precise empirical prediction. Moving the slider leftward reduces θ^* and therefore diminishes the influence of the empirical estimate. Moving the slider rightward has the opposite effect. By moving the slider to either extreme, the user may create a composite model that exactly reproduces the expert-elicited version or the empirically-derived version. The slider is linked to all the sheets: adjusting the slider on any sheet also moves the slider on the other sheets.

---

[1] Moore, C. T., J. J. Gannon, and T. L. Shaffer. 2018. NPAM predictive model improvements and piloting NPAM on partner lands: trio of expert-elicited, empirically-derived, and composite versions of the full set of state and transition models. U.S. Department of Interior, Fish and Wildlife Service, Cooperator Science Series FWS/CSS-130-2018, Washington, D. C. https://doi.org/10.3996/css59218717

SYNTHESIS

The NPAM program is in its infancy. Although data are accumulating rapidly (sample sizes in the hundreds), the multiple interacting factors that form the competing NPAM models result in many parameters that may never be well estimated for persistent lack of data. However, we made use of certain modeling tactics to allow inference to be made in data-sparse environments, and we developed a strategy for averaging both strong and weak empirical evidence with information derived from experts. We believe that a composite set of models can be derived for both mixed grass and tallgrass systems, ultimately replacing the current models derived solely on expert opinion.

Our linear models contained four predictors: the prior-year vegetation component, treatment, type of dominant invader, and defoliation level. If all predictors are treated as levels of discrete factors, parameters for higher-order interactions become difficult to estimate because specific factor combinations may be supported by few or no data. Treatment and type of dominant invader are inherently discrete factors, but we modeled defoliation level as a continuous covariate, which was key to reducing issues of data sparseness in higher order interactions. By assuming linearity in the response and imposing this type of structure in the models, we managed to borrow information across the range of the defoliation index, which could not have occurred if defoliation had been modeled as a categorical effect.

Despite the imposition of structure to overcome data sparseness, there remained some model parameters that could not be estimated in a few of the higher-order models. In these cases, we imposed additional equality constraints to assure that the model yielded estimable functions. This involved equating parameter values to functions of sums of other parameters that were estimable. Some of these equality constraints may be unnecessary in the future as more data become available.

For mixed grass, evidence is strong that Model 4 is the most appropriate for these data. In other words, among the competing model structures built to approximate the NPAM conceptual models for mixed grass, best fit to the data (minimum AIC) was obtained under Model 4. We note that for mixed grass under NPAM, model weight has steadily accumulated on Model 1 and has remained unchanged on Model 4. It is important to recognize that in this study, parameters are being estimated from data, as opposed to setting model parameters at fixed values derived from expert judgment, as was done at the outset of NPAM. Thus, if directional bias exists in some of the expert-elicited parameters, Model 1 may serve as an adequate default model to explain pattern that higher-order models cannot. Despite the AIC evidence that Model 4 provides the best fit to the data, this is no indication that Model 4 is the “true” model. In fact, patterns in the data may be even more complex than can be captured by any of these models, and Model 4, the most complex of these, may be judged most appropriate because it is the best approximation to truth. For tallgrass, the most parsimonious model was Model 1. Again, there is no assertion about “correctness” of this model, only that biological pattern in the tallgrass system has yet to emerge, and the simplicity of Model 1 is currently the best at describing the system.

As mentioned earlier, the choice of assigning weight to expert-elicited information and empirical evidence is completely arbitrary. The models described here provide weight to the empirically-derived estimate of transition probability according to its level of precision, but above some chosen minimum threshold of weight. Where precision for the empirical estimate is greatest, the estimate receives exclusive weight in computing the composite transition probability. Where precision is least, the empirically-derived probability receives the minimum allowed weighting.

Once composite versions of the models are obtained, the optimization work last conducted in 2011-2012 may be repeated using the updated models. This work requires specialized experience and knowledge of adaptive stochastic dynamic programming. One difficulty is that the software used to compute policies in 2011-2012 (Lubow 1995; 1997) is no longer supported and does not run on existing operating systems; thus, policy derivation will require an alternative software solution, e.g., MDPSolve (Fackler 2015). At this opportunity, the estimates of partial controllability may also be updated with current information and incorporated into the optimization.

Once new policies are computed, a decision remains about model belief weights that have accumulated to that point. A reasonable solution is to leave the weights as they are, but an argument could be made to re-start the weights at equal values (i.e., reflecting complete uncertainty). Either way, it is important to keep in mind that only the parameters in the models will have changed, not the model structures themselves. From either starting point, with sufficient time, model belief weights will calibrate to the new predictions.

This opportunity of “outer-loop learning”, if taken now, will have occurred about 7-8 years into the implementation of NPAM. It would be possible to repeat these analyses at relatively long intervals for periodically updating the transition probabilities of the predictive models. A scheme could be devised (e.g., weighting of empirical evidence in proportion to the passage of time) that progressively reduces the influence of the expert-elicited model values every time the data are re-analyzed.

CHAPTER 2: ANALYSIS OF ABIOTIC FACTORS ON NATIVE PRAIRIE RESPONSE

Summary:
We considered the role that time and location-specific abiotic factors play in predicting response of amount of native prairie cover to management actions. We fit models containing predictors of habitat (static and transect-specific), climate (static and unit-specific), and recent precipitation (dynamic and unit-specific). The goal of this effort was to evaluate the importance of including abiotic variables in developing future predictive models for NPAM. We also report on long-term, project-wide trends in response of native prairie vegetative cover to management on NPAM units, in both mixed grass and tallgrass systems.

METHODS

Data Preparation

The NPAM program is conducted across approximately 120 management units (the number varies slightly each year) of the U.S. Fish and Wildlife Service National Wildlife Refuge System in Minnesota, North Dakota, South Dakota, and Montana. Management units are parcels of unbroken, native sod experiencing invasions by two introduced cool-season grasses, smooth brome and Kentucky bluegrass. The program objective of NPAM is to arrest and reverse invasions on all units. Most units (~80) occur in the western, drier mixed grass prairie, and the rest occur in the eastern, wetter tallgrass prairie. Units are monitored annually for vegetation composition. Between monitoring episodes, a treatment is applied and recorded. The treatment alternatives vary by grass type. In mixed grass units, treatment alternatives in place since the 2009-2010 management cycle are Rest, Graze, Burn, and Burn/Graze combination. In tallgrass units, treatment alternatives used since the 2010-2011 cycle are Rest, Graze within a defined window based on smooth brome flowering phenology (Graze/Window), Burn within the phenological window (Burn/Window), and Defoliate (haying at any time of the year, or burning or grazing outside of the phenological window). Descriptions of treatments and monitoring protocols are found in Gannon et al. (2013).

Under the NPAM monitoring program, 25-meter monitoring transects are randomly distributed across the unit, at an approximate density of 0.5/ha and separated by at least 65m. The belt transect sampling approach (Grant et al. 2004) is used, where the dominant type of vegetation is recorded within every 0.5-m segment along the transect. For our analyses, we aggregated vegetation types into four classes: native grasses and forbs, smooth brome (SB), Kentucky bluegrass (KB), and remainder (RM; other non-native vegetation or native trees and shrubs). For each transect ($j = 1, 2, \ldots, J_i$) in unit i in year t, we tabulated the frequency of occurrence of each vegetation class over all segments, yielding counts $n_{ijt,\mathrm{NP}}$, $n_{ijt,\mathrm{SB}}$, $n_{ijt,\mathrm{KB}}$, and $n_{ijt,\mathrm{RM}}$, respectively. From these counts, we computed the proportion of the transect dominated by native grasses and forbs, $x_{ijt} = n_{ijt,\mathrm{NP}} / (n_{ijt,\mathrm{NP}} + n_{ijt,\mathrm{SB}} + n_{ijt,\mathrm{KB}} + n_{ijt,\mathrm{RM}})$.

We obtained monthly precipitation totals for each transect for years 2007-2016, based on geographically interpolated estimates provided by the National Oceanic and Atmospheric Administration, National Weather Service (Appendix A). We averaged the totals over all transects in the unit. We obtained a prior-year precipitation value for unit i and year t, P_{it}, by summing the unit-level averages over the 12 months of previous calendar year t-1. Similarly, we obtained transect-level long-term (1961-1990) average values of mean annual precipitation and mean temperature during the warmest month based on geographically interpolated estimates provided by the USDA Forest Service, Rocky Mountain Research Station, Moscow Forestry Sciences Laboratory (Appendix B). We averaged these data over all transects in the unit to obtain unit-level values of mean annual precipitation (CP_i) and mean temperature during the warmest month (CT_i).

We used Ecological Site Description (ESD) data from the Natural Resources Conservation Service to characterize vulnerability of the site to invasion by smooth brome and Kentucky bluegrass. Data were available to us at the microsite level (soil survey map unit) for all states in the NPAM focus region except Minnesota. With expert guidance (Jeff Printz, USDA Natural Resources Conservation Service; David Pyke, U.S. Geological Survey; personal communication), we assigned all ESD classes into either of two broad categories: "Invade", meaning classes that that experts believe are less resistant to invasion by smooth brome or Kentucky bluegrass, and "Maintain", meaning classes that experts believe are more resistant to invasion by smooth brome and Kentucky bluegrass (Appendix C). We intersected soil map layers (with their attendant ESD attributes) with a map layer of NPAM transects, and we computed the proportion of each transect in each of the two classes (Appendix C). Variable Z_{ij} is the proportion of transect j of unit i classified as the "Invade" type, which we term "invasion vulnerability".

Modeling and Estimation

Because data on ESD were lacking for Minnesota at the time of our study, we conducted two analyses of native prairie composition response: (1) effects of long-term climate (precipitation and temperature), annual precipitation, treatment, and invasion vulnerability within mixed grass units, and (2) effects of long-term climate, annual precipitation, and treatment for all units in NPAM. In both analyses, the count of 0.5-m segments dominated by native cover, $n_{ijt,\mathrm{NP}}$ for unit i, transect j, year t was the response variable of interest.

Mixed grass analysis – Using data only from mixed grass units, we modeled $n_{ijt,\mathrm{NP}}$ as a binomial response with probability p_{ijt} in the following linear model:

$$\log\left(\frac{p_{ijt}}{1-p_{ijt}}\right)=\mu+\upsilon(CP_i,CT_i)+\beta P_{it}+\tau^{(A_{it})}+\alpha Z_{ij}+(\tau\alpha)^{(A_{it})}Z_{ij}+(\tau\beta)^{(A_{it})}P_{it}+\delta x_{ij,t-1}+\eta_t+\varepsilon_{ijt}$$

where

μ is the overall mean,

υ is a unit-specific linear function of climate variables CP_i and CT_i (centered and scaled to unit variance),

β is a regression slope for prior-year total precipitation (centered and scaled to unit variance),

τ is the effect of treatment A administered at unit i at time t,

α is a regression slope for invasion vulnerability (centered to its mean),

δ is a regression slope for prior-year native prairie composition proportion (centered to its mean),

η is a random year effect,

ε is a random noise term, and

$(\tau\alpha)$, $(\tau\beta)$ are 2-way interaction terms involving treatment, invasion vulnerability, and precipitation.

The term $\upsilon(CP_i, CT_i)$ is a linear function of the climate terms CP_i, CT_i, and their interaction. It increments or decrements the overall mean by an amount related to the climatic position of the unit, thus allowing the average native prairie composition response to vary according to the unit's wetness and warmness.

The terms β and α are regression slopes for prior-year precipitation and invasion vulnerability variables, respectively. The term τ represents the effect of treatment $A \in$ {Rest, Graze, Burn, Burn/Graze Combination, Unassignable} (the "Unassignable" treatment class was recognized in the model to allow use of all the data, but its effect is not reported in Results). The model includes 2-way interactions between treatment and each of the regression terms. Interactions permit the effects of precipitation and invasion vulnerability to vary in strength and direction according to treatment type. The term δ is an autoregressive coefficient that adjusts the response for the effect of amount of native prairie composition in the previous year. The term η is a random year effect that expresses annual variation in native prairie composition that are not explained through other time-dependent predictors. Values of η are drawn from a zero-centered normal distribution with standard deviation σ_η. The term ε represents overdispersion for the binomial distribution to account for other sources of unexplained variation; values of ε are drawn from a zero-centered normal distribution with standard deviation σ_ε.

<u>All-units analysis</u> – Using data from all mixed grass and tallgrass units, we constructed a similar model by dropping terms related to invasion vulnerability and adding a term related to grassland type:

$$\log\left(\frac{p_{ijt}}{1-p_{ijt}}\right) = \mu + \gamma^{(G_i)} + \upsilon(CP_i, CT_i) + \beta P_{it} + \tau^{(A_{it})} \\ + (\tau\gamma)^{(A_{it}*G_i)} + (\tau\beta)^{(A_{it})} P_{it} + (\gamma\beta)^{(G_i)} P_{it} + (\tau\gamma\beta)^{(A_{it}*G_i)} P_{it} + \delta x_{ij,t-1} + \eta_t^{(G_i)} + \varepsilon_{ijt}$$

The term γ represents an amount to add to or subtract from the mean due to grassland type (mixed grass or tallgrass). The grassland type effect also interacts with treatment ($\tau\gamma$), prior-year precipitation ($\gamma\beta$), and the treatment-precipitation interaction ($\tau\gamma\beta$). Interactions between grassland type and treatment were required in the model because the coding for treatment (A_{it} = {1, 2, 3, 4, 5}) did not distinguish between mixed grass and tallgrass treatments. Thus, the interactions allowed us to estimate the effects of treatment separately for mixed grass and tallgrass prairie. The model also contained grassland-specific year random effects (η), allowing annual variation to differ between grassland types. Values of η are drawn from zero-centered normal distributions with standard deviation σ_η specific to grassland type. All other terms are as described in the mixed grass analysis.

<u>Estimation</u> – For each model, we used Markov Chain Monte Carlo (MCMC) simulation in JAGS 4.2.0 (Plummer 2003) through the jagsUI package (Kellner 2016) in R (Version 3.3.2, R Development Core Team 2016) to approximate posterior distributions of all parameters, given vague priors. We also provided vague prior distributions for missing values (responses and predictors) which allowed us to use partial data records. To assess goodness of fit of each model, we computed a discrepancy measure (Bayesian p-value) based on comparing squared Pearson residuals of the predicted responses to those computed from the data. We sampled posterior distributions of additional derived quantities for further interpretation of the model. For the mixed grass model, we computed response means at 3 levels of ESD Invade proportion (0, 0.5, 1.0) for each treatment type. For the all-units model, we computed response means at 3 levels of prior-year rainfall (-2 SD, mean, +2 SD) by treatment type within grassland type. For each grassland type and for all units enrolled in NPAM, we estimated the trend in response over the period 2010-2016, adjusting for all other effects (i.e., all effects held to their means). We simulated 3 MCMC chains beyond an initial burn-in of 500 samples until convergence was achieved for all parameters of the model (all R-hat statistics <1.1 [Gelman and Rubin 1992] and visual confirmation of mixture in the traces). We updated the chains for 10,000 iterations to generate posterior distributions of model parameters and the additional derived quantities. All programs and scripts used to conduct these analyses are collected in the program code archive (Appendix G).

RESULTS

Mixed Grass Analysis

Results for this model were based on 30,000 posterior distribution samples after discarding 9,900 samples as burn-in. Overdispersion in native prairie response in mixed grass units was considerable, and accounting for it through a random effect produced a model with adequate fit to the data (Bayesian p-value = 0.789). We found a strong interaction between treatment and invasion vulnerability (Table 2.1; Fig. 2.1) in the native prairie response. Overall, treatments involving fire (Burn and Burn/Graze Combination) produced greater proportions of native prairie than did Rest or Graze. Increased invasion vulnerability was associated with reduced native prairie composition. However, the association varied by treatment. All treatments except Graze indicated a decline of effectiveness (lower native prairie composition response) as invasion vulnerability increased from 0 to 1. The effect of the Graze treatment was seemingly unaltered by variation in invasion vulnerability.

All-units Analysis

Results for this model were based on 30,000 samples from the posterior distribution after discarding 30,300 samples as burn-in. Accounting for overdispersion led to a model that fit adequately to data from all mixed grass and tallgrass units (Bayesian p-value = 0.797). The strong 3-way interaction in this model suggested that treatment interacted with prior-year precipitation and that the patterns were different in the two grassland types (Table 2.2). In mixed grass (Fig. 2.2), responses to treatments involving fire (Burn and Burn/Graze Combination) were greater than Graze, and the response under Graze was greater than under Rest except following drier than average conditions. All treatments except Rest indicated an increase in treatment effectiveness (greater native prairie composition response) as precipitation increased from drier than average to wetter than average. For the Rest treatment, the association was slightly negative.

In the tallgrass type, native prairie composition response was sensitive to treatment only at higher than average levels of prior-year precipitation, where phenology-based treatments (Graze/Window, Burn/Window) were more effective than either Rest or Defoliate (Fig. 2.3). Otherwise, no clear pattern of treatment effectiveness was evident.

Climate is an important driver in native prairie composition at the unit level (Fig. 2.4), including a strong interaction between temperature and precipitation. At drier sites, estimated native cover declines rapidly with increasing temperature, whereas almost no relationship occurs between native cover and temperature at moister sites. The strong negative relationship between moisture and native cover at cooler sites is not evident at warmer sites.

After adjustment for all other effects in the model, annual means of native prairie composition across the NPAM project are variable, but they trend in a positive direction over the period 2010-2016 (Fig. 2.5). The average annual increase in the odds ratio for percent native prairie cover was 6.1% (95% credible interval 3.1—8.9%). The same trends are observed within grassland types (Fig. 2.6). The trends are visually identical between grassland types, but the model does not constrain them to be.

SYNTHESIS

Decision guidance under NPAM is currently informed by three pieces of information: (1) grass type, (2) current-year vegetation composition, and (3) treatment history. The latter two pieces of information are dynamic conditions, and annual changes in the states of these variables are responsible for temporal variation in management guidance at a given site on the landscape. Thus, in its current form, NPAM provides decision guidance under one limited representation of a static habitat condition (mixed grass versus tallgrass type) and two dynamic conditions, one of which (vegetation composition) is the target of management itself.

However, grassland communities are complex, and it is conceivable that their responses to management actions may depend on environmental characteristics not recognized in the NPAM framework and which operate over a range of spatial, temporal, and biological scales. Thus, the potential exists for greater management performance if decision guidance could be more narrowly targeted to specific environmental attributes. The objective of this investigation was to perform an initial assessment of this potential by modeling and describing the relationship between vegetation composition, treatment, and two abiotic conditions: invasion vulnerability and prior-year precipitation. These variables were proposed as predictors on the advice of USFWS members of the NPAM Advisory Team, who believed that these two measures accounted for much among-unit, within-unit, and among year variability in response to treatment. Furthermore, they could be gathered one time at the time of unit enrollment (invasion vulnerability) or collected easily on an annual basis (prior-year precipitation). We included a third abiotic condition, long-term climate, to model how a unit's average native cover response varied with its climatic position.

Invasion vulnerability is a static condition assessed at the transect level. At the time of our analysis, ESD data from which the invasion vulnerability score was derived were not available for the state of Minnesota; thus, our assessment was limited only to mixed grass units. Higher invasion vulnerability score was associated with lower amount of native grasses and forbs, and this pattern was observed for every treatment. However, the pattern was much attenuated on transects that received the Graze treatment: a Graze treatment resulted in about the same response in native cover proportion whether the transect had a high or low invasion vulnerability score. Although the treatments involving fire were most effective at increasing amount of native cover, our results suggest that increasing native cover by any of the NPAM treatments is a challenge on sites with high invasion vulnerability, compared with sites without. Experts believe that sites more vulnerable to invasion are moister sites distinguished by topographic position (e.g., bottoms and toe slopes), edaphic characteristics, and soil chemical properties.

Prior-year precipitation is a dynamic (changing) condition assessed at the management unit scale. For the Rest treatment, the only treatment common to the two grassland types, wetter than average conditions were associated with reduced native cover; however, the strength of the pattern was much greater at tallgrass sites. On mixed grass units, active forms of treatment (Graze, Burn, Burn/Graze) implemented under wetter than average conditions resulted in increased native cover response compared to drier than average conditions. On tallgrass units, the relationship with precipitation varied by type of active treatment. Wetter than average conditions resulted in increased native cover under the Graze/Window treatment compared to drier conditions; whereas the opposite was observed under either Burn/Window or Defoliate.

Over all NPAM units, native cover was sensitive to climatic position of the unit. Native cover was on average greatest on units that were cooler *and* drier and least on units that were wetter *or* warmer.

Because response to treatment was found to be dependent on site-specific abiotic conditions, taking such conditions into account in the selection of management actions may lead to improved management performance. The approach taken to incorporate abiotic conditions into the NPAM optimization framework depends on the static versus dynamic nature of the condition.

For the static condition of invasion vulnerability (or any similar static characteristic found to affect treatment response), a reasonable approach is to classify every unit into one of a few condition classes and stratify modeling and optimization around levels of these classes. NPAM already provides decision support based on a biotic condition class: separate models and decision policies are used for mixed grass and tallgrass units. Implementing an abiotic class structure – for example, invasion vulnerability at two levels – would involve classifying each NPAM unit into one of the possible levels, developing a predictive model of vegetation composition response specific to each level, and generating a decision policy that applies to each level. For the specific case of invasion vulnerability, there are at least two challenges. First, data to classify the entire NPAM project area were not available to us at the time of this analysis; with availability of the necessary data, it will be possible to classify every transect according to its invasion vulnerability. Second, whereas invasion vulnerability is assessed at the transect level, NPAM treatments are proposed and applied at the unit scale based on conditions perceived at that scale. Therefore, each unit would receive an invasion vulnerability assignment based on a composite or aggregate determination of invasion vulnerability scores from all transects in the unit, and a single action would be carried out across the unit. Consequently, while the action may be appropriate for some sites within the unit, it may not be appropriate for others. Conceivably, the action may produce a less desirable outcome for the unit than one obtained by following decision guidance that ignores the invasion vulnerability condition. On units where this lack of treatment precision may be of serious concern, the only realistic (but perhaps impractical) recourse is to subdivide the unit into new, smaller units, each of which are homogeneous with respect to the abiotic condition. Clearly, adopting such a strategy to increase treatment precision is traded off for efficiency in applying treatments. Simulation analysis of new decision models could be used to examine this tradeoff.

Incorporating the dynamic condition of prior-year precipitation (or any other condition that varies on an annual basis) into the decision framework requires development of additional system models and data collection mechanisms. The models that predict vegetation composition outcome as a consequence of treatment require augmentation with a component that recognizes the influence of annual precipitation. Additionally, the optimization step requires a companion dynamic model to predict annual precipitation into future time steps. To serve as a predictor in the biological model, total prior-year precipitation must be available at the time of vegetation monitoring, either directly measured at the unit or inferred from NOAA weather station reports. Already, NPAM makes use of two dynamic conditions to generate management action proposals: vegetation condition and defoliation history. For both conditions, supporting models and monitoring mechanisms are in place. Similar structures are needed to accommodate annual precipitation.

NPAM makes use of optimal policies that were generated by an adaptive stochastic dynamic programming algorithm (Lubow 1995; 1997). Computational efficiency of the algorithm is highly sensitive to number of dynamic states and discrete levels of states. Each dynamic state added, such as prior-year precipitation, slows the derivation of optimal policies, sometimes requiring weeks, or longer, to converge on a solution. Tradeoffs may be made in other dimensions of the problem; for example, reducing number of levels of another state variable permits the addition of a new state variable while preserving overall problem size. Alternatively, a larger problem size could be accommodated by foregoing the adaptive computation of policies. That is, the treatment of model uncertainty as a static rather than dynamic condition for purposes of optimization greatly reduces problem size in that dimension, permitting the modeling of greater complexity in other state dimensions. However, this approach results in the loss of capacity for decision making to actively pursue learning, such that any learning that does occur happens passively as an unplanned byproduct of decision making (Williams 1996; Moore and Conroy 2006). Furthermore, this approach requires that the decision policy be computed annually at every decision cycle, necessitating running the optimization algorithm each time. Heuristic alternatives to stochastic dynamic programming that find approximate rather than exact solutions are recently emerging (e.g., Nicol and Chadès 2011).

Incorporating static conditions in optimization does not cause the same concerns. For *k* levels of a static condition that is added, *k* optimal policies are generated, one for each level of the condition.

This study has confirmed that the abiotic conditions of site invasion vulnerability and prior-year precipitation interact with treatments. Thus, there is the potential to improve management performance of NPAM by taking these abiotic conditions into account to increase precision of treatment applications. However, the expected gain in conservation outcomes comes at the cost of added analytical, computational, and field burden. Simulation work could quantify this tradeoff, ultimately helping to make the determination of whether expected benefits are worth the costs.

We have also demonstrated that NPAM is achieving its conservation objective of increasing amount of native grass and forb cover over the entire practice area and within each grassland biome in its 7-year history. The annual rate of increase in the odds ratio for percent native prairie cover was 6.1%.

CHAPTER 3: PILOTING NPAM ON PARTNER LANDS

Summary:
We evaluated the potential of expanding the NPAM decision framework beyond the USFWS National Wildlife Refuge System (NWRS). This evaluation included multiple steps: conduct reach out, provide information, appraise interest, assess compatibility, and evaluate the capability of NPAM to include additional partners. Reach out was directed toward non-USFWS NWRS agencies and organizations that manage native prairie lands within the U.S. Prairie Pothole Region (PPR). We informed them about the NPAM initiative, including all aspects of the decision framework and criteria for involvement. Gauging interest in collaborating with the USFWS NWRS by participating in NPAM, and assessing the compatibility of these agencies/organizations with the existing NPAM criteria and protocols, was the intended objective. Evaluation of expanding NPAM to other partners was completed given an examination of the current infrastructure of NPAM to accommodate such expansion. In response to our reach-out efforts, we attracted representatives of seven different agencies/organizations to participate in an informational webinar and to complete a post-webinar questionnaire aimed at gauging their interest and assessing the compatibility between their agency/organization management approach and that of NPAM.

METHODS

Identification and contact of non-USFWS potential partners

We identified 10 organizations, located across the Prairie Pothole Region of Minnesota, North Dakota, South Dakota, and Montana, that we believed might share similar management challenges and objectives with the USFWS NWRS – the ones that motivated the development of NPAM – and might benefit from participation in the initiative if the USFWS were to expand it beyond refuges (federal, state, and non-government agencies: Bureau of Land Management (BLM), Bureau of Reclamation (BOR), National Park Service (NPS), USDA United States Forest Service (USFS), Minnesota Department of Natural Resources (MNDNR), Montana Fish Wildlife and Parks (MTFWP), North Dakota Game and Fish (NDGF), South Dakota Game Fish and Parks (SDGFP), Ducks Unlimited (DU), and The Nature Conservancy (TNC); Table 3.1). We obtained the contact information for 21 representatives from these organizations (including managers, biologists, and ecologists; Table 3.1) and sent them an email correspondence (Appendix D) describing our motivations and inquiring as to their interest in learning more about NPAM by participating in a webinar and follow-up questionnaire.

Informational Webinar

We developed an hour-long presentation to be delivered via an online webinar to the interested individuals who responded to our initial contact (Appendix E). The purpose of the presentation was to share the details of the NPAM initiative, explore the potential for expanding NPAM to partners beyond USFWS NWRS, describe the follow-up questionnaire, and answer any questions. Pertinent details of NPAM that we conveyed to webinar participants included a description of the natural resource problem, the genesis of NPAM, a short summary of adaptive management, an overview of the technical elements and infrastructure of NPAM, an explanation of what cooperators gain from participation in NPAM, and a description of how a new non-USFWS partner might potentially participate in NPAM.

Questionnaire

We designed a spreadsheet-based questionnaire to be completed by those individuals who participated in the webinar (Appendix F). The purpose of the questionnaire was to assess the compatibility between NPAM, as it currently exists, and the needs and abilities of these non-USFWS NWRS potential partners. The questionnaire included questions from seven subject matters: land units, management objective, management actions, vegetation monitoring, data sharing, involvement, and potential benefits and barriers to implementation of NPAM to the partner's agency. With this information, we assess the potential to expand NPAM beyond the USFWS NWRS.

RESULTS

Identification and contact of non-USFWS potential partners – Responses

Six of the original 21 individuals responded to our email correspondence, affirming that they were interested in learning about NPAM and exploring the possibility of potential partnership by participating in a webinar and follow-up questionnaire (Table 3.1). An additional 14 individuals, whom had received our email by way of forwards from the original recipients, also responded to us affirming their interest (Table 3.1). The 20 interested persons represented two federal agencies (NPS, USFS), three state agencies (MNDNR, NDGF, SDGFP), one non-government organization (TNC), and one cooperative venture (Plains and Prairie Pothole Landscape Conservation Cooperative). The individuals we contacted who did not express interest represented two federal agencies (BLM, BOR), one state agency (MTFWP), and one non-government organization (DU).

Informational Webinar

The informational webinar was delivered on 17 January 2014. Eighteen individuals, representing eight organizations – three federal agencies (BOR, NPS, USFS), three state agencies (MNDNR, NDGF, SDGFP), one non-government organization (TNC), and one cooperative venture (PPP LCC) – participated in the webinar (Table 3.1). These individuals were a combination of original contacts, forwarded contacts, and new arrivals. Seven people who expressed interest in the webinar did not participate. Six people who did not originally indicate interest in the webinar did participate; two of these people were new arrivals (i.e., not on either the original or forwarded contact list; Table 3.1).

Questionnaire

The follow-up questionnaire was sent on 17 January 2014 with responses requested by 7 February 2014. Thirteen of the 18 webinar participants provided responses to the questionnaire (Table 3.1). Of the eight organizations represented by participants of the webinar, we received the following number of completed questionnaires per organization: one BOR, two NPS, one USFS, one MNDNR, four NDGF, one SDGF, and three TNC. With exception of the PPP LCC, we received at least one completed questionnaire from each organization that participated in the webinar. We did not expect the representative from the PPP LCC to complete a questionnaire because LCCs do not directly own or manage land but instead promote collaborative partnerships among agencies; the PPP LCC representative participated in the webinar simply as an interested party. A summary of the responses is provided, per subject matter, below.

Land Units – All respondents stated that their organizations have land units over which they have control and that some of these units contain native sod. Six of the responses were for mixed-grass prairies only, two for tallgrass prairies only, and five for both mixed and tallgrass prairies. Two respondents made no indication regarding the presence of smooth brome (SB). Of the 11 who did provide a response, all were affirmative; they reported a range of 1-100% coverage with an average coverage ranging from 5-80%. Two of these 11 respondents stated that SB was the main non-native plant of concern on their units; the remaining nine cited Kentucky bluegrass (KB), annual bromes (*Bromus* spp.), woody vegetation, noxious weeds, and crested wheatgrass (*Agropyron cristatum*) as the main non-native plant of concern on their units. All 13 respondents noted the presence of KB on their units, reporting coverage ranging from 5-100%, with an average of 25-35%. Out of 12 of the respondents who answered the question, six stated that KB was the main non-native plant of concern on their native sod units. The other six identified annual bromes, woody vegetation, noxious weeds, redtop (*Agrostis* spp.), cattail (*Typha* spp.), reed canary grass (*Phalaris arundinacea*), and leafy spurge (*Euphorbia esula*) as their main non-native plant of concern. The respondent from SDGFP identified SB and KB as the two main non-native plants of concern.

Management Objective – Twelve of the respondents provided information regarding the management objective of their organization for their native sod land units, as well as their opinion as to whether their management objective is compatible with the stated management objective of NPAM – to increase the cover of native grasses and forbs while minimizing cost. None of the objectives were completely aligned with that of NPAM, but almost all had some level of overlap. Objective foci sorted into three themes: (1) native flora; (2) habitat to support native fauna; and (3) habitat for game birds. The most commonly cited objectives, and those most closely aligned with NPAM (though not an explicit component), included native plant diversity and vegetation structural diversity. These objectives were sometimes cited as the focus in-and-of-themselves, and sometimes noted in conjunction with providing habitat for invertebrates/pollinators, grassland birds, and other wildlife species. The second theme of objectives, less aligned with NPAM, included animal species diversity and ecosystem function. The third theme of objectives, and the least compatible with NPAM, focused on managing vegetation height and density for cover of nesting game birds. The management objectives of TNC appeared to be the most closely aligned with those of NPAM (native flora theme), followed by the NPS, MNDNR, and USFS (native flora and native fauna themes), and then the NDGF and SDGFP (habitat for game birds).

Management Actions – All but two (NPS, BOR) of the 13 respondents acknowledged that annual management decisions are made on the land units under the control of their agency. Of the two indicating no process for making annual management decisions for the lands, one stated it would be possible for such decisions to be made (BOR). Decision makers vary by organization, but are a type of regional, district, or local land manager. For agencies with large land bases, it was common to have multiple decision makers, dependent upon the location of the land unit, who have autonomy with regard to interpreting the management objectives of their agencies and making management decisions. For example, NDGF is divided into six districts, each district headed by a different manager. This setup is similar to that of refuge or complex managers involved in NPAM.

Working within a 1 September to 31 August management year, and waiting until 1 September to make a decision regarding the management to carry out that year, was inconsistent among respondents: three respondents stated this timeline was feasible (NPS, NDGF, SDGF), two stated it was not workable (NPS and USFS), and the remaining eight responded that it “may” be possible. Willingness to consider guidance regarding management before making a decision was also variable among respondents: four respondents agreed they would entertain the receipt of guidance (USFS, NDGF, SDGFP, and TNC), one said they would not (NPS), and the remaining eight reported they “may” be willing. In terms of the ability to make management decisions on individual units that are independent between years, six respondents acknowledged this ability, two reported to lack this ability (BOR, USFS), and four responded that they “may” have this ability. Most respondents (nine) agreed their organization has the ability to make independent decisions regarding different land units, whereas three respondents stated it “may” be possible.

All 11 respondents who indicated management control over mixed-grass prairie units provided information on the feasibility of carrying out specific management actions (i.e., Rest, Graze, Burn, and Burn/Graze combination) on those units. Rest and Graze were the most commonly available management actions, receiving nine and seven affirmative responses, respectively. While none of the respondents expressed an inability to carry out either of these two actions, access to cattle herds and an absence of fencing were indicated as two potential complications to implementing grazing. Burn and Burn/Graze combination actions were less available for use; four respondents stated an ability to carry out prescribed burns, three reported the ability to implement burn/graze combination management, one stated the inability to carry out either management action (BOR), and the remaining respondents acknowledged it “may” be possible to carry out these actions. Four respondents cited lack of funding and human resources as impediments to carrying out prescribed burns. Two respondents reported the ability to carry out all four actions (MNDNR, TNC), eight expressed a mix of affirmative and potential ability for the four actions, and one noted a mix of potential and negative ability to carrying out the four actions.

All seven respondents who indicated management control over tallgrass prairie units provided information on the feasibility of implementing specific management actions (i.e., Rest, Graze within window, Burn within window, and Defoliate) on those units. Rest was reported as the most available action, receiving five affirmative and one “maybe” response; one respondent did not provide a response regarding the feasibility of Rest. Grazing within the window was also deemed highly achievable, with four affirmative, two “maybe”, and one negative response. As with mixed-grass, burning was reported to be less available; it received four affirmative, two “maybe”, and one negative response. Carrying out a Defoliate (i.e., graze or burn outside of the phenological window, or hay at any time) was reported to be equally achievable as carrying out a graze within the window. One respondent reported that no form of active management could be carried out on the specific lands owned by the employing agency (NPS). One of the respondents reported the ability to carry out all four actions, whereas the other five respondents reported a mixture of affirmative and potential ability for the four actions.

Tallgrass management actions require the ability to assess smooth brome phenology to determine the timing of the cool-season window. It is this timing that differentiates between a graze or burn conducted within the window and a defoliate action. Six of the seven respondents for tallgrass units reported on their ability to assess phenology on the unit both during the growing season and at the time of management application. Four respondents stated they had the ability to make this phenological observation and assessment, whereas two respondents noted that such an assessment “may” be possible.

Other questions we asked respondents had to do with the ability to control the application timing and method of their actions, the ability to apply their actions to entire land units, and their willingness to abstain from broadcast herbicides or other non-approved treatments. Regarding control over timing and application method, nine respondents reported affirmatively, one expressed inability (NPS), and three noted they may have this control. Spatial application of the action over entire units is less tractable, with five reporting to have this ability, three replying to be incapable (BOR, NPS, USFS), and four stating they may have this ability. Willingness to refrain from non-acceptable treatments was reported to be high, with nine affirmative, two negative (USFS, BOR), and one "maybe" (MNDNR) response.

Lastly, we asked about the ability to record the details of their management applications and to enter these management details into a web-based data system by August 25 of each year. Nine respondents stated the ability to record the details, whereas only six reported the ability to enter these details into a database. The BOR stated they would not be able to record nor enter their management details. The USFS indicated they could possibly record the details but could not enter them into a database by the designated date. MNDNR and two NDGF respondents noted they could record the details but can only possibly meet the data entry deadline.

Vegetation Monitoring – Existing monitoring of vegetation among the agencies is low; five respondents reported some level of vegetation monitoring (NPS, USFS, MNDNR, TNC), six communicated occasional monitoring efforts (BOR, NPS, NDGF, SDGFP, TNC), and two acknowledged they do not conduct any monitoring (NDGF). More variable is the type of monitoring that occurs among those that do carry it out. The most intensive monitoring reported is carried out by MNDNR and TNC on units enrolled in the Grassland Monitoring Team (GMT) project. The GMT monitoring protocol includes three tiers of monitoring that vary by the level of detail and effort involved. Participants of the GMT select which tier of monitoring they will conduct. The tier of least detail fully meets the plant community information needs of the NPAM vegetation monitoring protocol; the upper two tiers provide more detail (e.g., plant species diversity) than is necessary for NPAM. The GMT monitoring efforts, however, are completed on a less frequent cycle (every 3-5 years) and at a lower density (one transect per 10 acres) than the NPAM monitoring protocol. Of the other agencies that reported vegetation monitoring activity, TNC of South Dakota conducts transects at a density of one per 100 acres to gather data on native versus non-native plants, as well as presence of SB and KB; this monitoring is carried out on a 3-5 year rotation. USFS reported that they conduct monitoring to assess vegetation structure, utilization, and some composition. The NPS noted they carry out monitoring for vegetation height and species presence/absence on some random points within their park boundaries. Of those agencies that reported to sometimes monitor the vegetation on their land (BOR, NDGF, SDGFP), these tended to be casual visual observations to assess health of their grasslands and, in some cases, encroachment of SB and KB.

The ability to carry out NPAM level vegetation monitoring is reported to be very low. When asked if their agencies could carry out the NPAM vegetation monitoring protocol on an annual basis on some of their land units, nine respondents replied "maybe" and four said "no". The same answers were given regarding the ability to complete monitoring and enter the data by the August 25 deadline. Lack of financial and human resources were cited as the reason for not being capable of implementing a NPAM-level monitoring protocol on even a few of their units.

Data Sharing – Seven of the twelve respondents stated that their agency would be willing to share the management and vegetation monitoring data in the NPAM centralized database, one respondent reported "no" (BOR), and four stated "maybe". The BOR respondent cited lack of staff time as the reason for not sharing the data. A NDGF respondent, whose response was "maybe", noted that if they shared their data, they would expect USFWS to share their data in return.

Involvement – When we posed the following question, "If, at some point in the future, NPAM decided to expand to include compatible partners outside of the USFWS NWRS, and if your agency and the Advisory Team of NPAM both agreed that there was a good fit, would your agency be interested in enrolling one or more units in NPAM?", we received responses ranging from "Strongly Agree" to "Strongly Disagree". Two TNC respondents, along with the SDGFP respondent, strongly agreed that they would be interested in participating under the aforementioned description. Three NDGF respondents, along with one NPS respondent, stated that they "somewhat agree" that their agency would be willing to participate. Two respondents, NDGF and TNC, were "not sure", the MNDNR respondent somewhat disagreed, and the BOR strongly disagreed. Most respondents noted the person who would be able to make the decision regarding participation. Only a single respondent (TNC) indicated to have dedicated staff that could carry out the NPAM protocols. One respondent reported that they might have the available staff (NDGF), six reported that they were not sure whether they had the staff (NPS, NDGF, SDGFP, TNC), two somewhat disagreed that they have the staff (MNDNR, NDGF), and one strongly disagreed that they had the staff to dedicate to NPAM protocols (BOR). Assuming staff were available, five respondents strongly agreed that they would be willing to have these staff participate in NPAM training sessions (NDGF, SDGFP, TNC), two somewhat agreed (NDGF), two were not sure (NPS, NDGF), one somewhat disagreed (MNDNR), and one strongly disagreed (BOR). Two TNC respondents and one NDGF respondent cited financial and human resources to conduct annual vegetation monitoring as what they saw as the biggest hurdle to their successful participation; though they all clearly stated that they did not see this as an insurmountable obstacle. A NPS respondent noted there could be potential to try NPAM in Knife River Indian Villages, and if successful, then expand involvement. Another NPS respondent stated they may have more interest in involvement if NPAM expanded to include restored grasslands, as opposed to only native sod. The USFS respondent cited the widespread use of broadcast herbicides and other types of non-acceptable NPAM actions that they use as a reason that USFS lands would not be compatible with the NPAM initiative. A NDGF respondent noted that most of their native sod units are located west of the Missouri River, which is outside of the Prairie Pothole Region and thus would likely not meet criteria for enrollment in NPAM.

Other – Of the 11 responses regarding potential benefits to their agency if they participated in NPAM, all but two of the respondents (BOR, NDGF) noted seeing benefits of such a partnership. The perceived benefits included science-based management, more effective management against SB and KB, defensible management, and collaborative learning focused on improved management to benefit the quality of native grasslands.

When asked about potential barriers to their agency's participation in NPAM, all 12 respondents cited insufficient financial and human resources to carry out annual vegetation monitoring. Annual management decisions (TNC), ability to negotiate with grazing tenants (TNC), lack of resources to carry out prescribed burns (NDGF), exclusion of broadcast herbicides from admissible treatments (USFS), lack of control over management actions (NPS), and involvement with other adaptive management projects (i.e., GMT; MNDNR) were also reported as potential barriers to their agency's participation in NPAM.

When queried as to the change that would be the most difficult for their agency to implement to participate in NPAM, eight of the ten responses pinpointed increased financial and human resources. The need for these resources was mostly targeted towards the ability to carry out vegetation monitoring and entry of these data by the annual deadlines. Two respondents, however, noted the need for resources for other needs: implementing prescribed burns (NDGF) and dividing their land into defined management units (NPS). The need to change their grazing contract system was indicated by TNC. Determining how the NPAM and GMT efforts can be combined to complement each other was also noted (MNDNR). One respondent (SDGFP) observed the need for a paradigm shift, to think beyond the focal game bird and recognize the value of other native prairie resources. Attitudes regarding whether these obstacles can be overcome were varied, ranging from "yes" to "not sure". The route to overcome the funding and human resources hurdle is the most straightforward, though success is uncertain. Inquiry regarding whether respondents could foresee a path to future participation also produced varied responses, from “not sure” to “strongly agree”.

Six respondents commented on aspects of their agency’s existing approach to management of their native prairies that they thought coincided well with NPAM. One NDGF respondent indicated that the act of annual decision making is coincident between the two approaches. Another NDGF respondent simply observed that their agency is very flexible, which would open the door to adopting NPAM protocols. The SDGFP respondent indicated that the agency embraces any effort and opportunities to improve native prairie through management actions. All three TNC respondents observed aspects of the two approaches that are well aligned with each other. The management objectives of NPAM are closely aligned with that of TNC and TNC strongly supports science-based management. Additionally, though methodology is different from NPAM, TNC already implements grazing and fire treatments, and carries out vegetation monitoring programs.

All respondents acknowledged that their agencies collaborate with partners external to their agency. Collaborations spanned the gamut of federal, state, county, and non-government agencies. A TNC respondent expounded on the type of activities on which they have collaborated, and these included research, land management, government relations, and outreach/education. Bureaucracy was identified as the number one difficulty in establishing and maintaining successful collaborative partnerships with external agencies.

Lastly, we asked if the participants found the informational webinar useful to their agency, as well as encouraged them to share any additional comments. Two respondents strongly agreed, four somewhat agreed, and four were not sure if their participation in the webinar was valuable. Respondents indicating the least ability, benefit, or interest in the involvement of their agencies with the NPAM program (BOR, USFS) did not respond to this question. Interestingly, several of the respondents noted their overall approval of NPAM as a program and said they look forward to following its progress and seeing what they can glean from it (i.e., as external non-participants in the program).

DISCUSSION

Questionnaire

Land Units – All responding agencies have land units with some native sod in mixed and/or tall grass prairie and with some cover of SB and KB. The physiography of the BOR land units (narrow strips adjacent to reservoirs), however, likely renders them incompatible with NPAM. Furthermore, the NPS noted that their land is not divided into distinct management units but instead is considered one extensive piece of land. While all have SB and KB, BOR and NPS reported that neither are main invasive species of concern. In terms of physical configuration of land units and their focal invasive species, USFS, MNDNR, NDGF, SDGFP, and TNC appear compatible with NPAM.

Management Objective – The management objectives of TNC and MNDNR, while not the same as those of NPAM, appear to be the most compatible with those of NPAM. Both TNC and MNDNR focus on more than the singular aim of increasing cover of native grasses and forbs while decreasing cover of the invasive cool-season grasses SB and KB; they target invasive vegetation other than SB and KB, and they explicitly aim to increase structural diversity, native plant diversity, support for animal diversity, especially pollinators and grassland birds, and ecosystem function. Both TNC and MNDNR participate on the GMT, an adaptive management approach for tallgrass prairies of Minnesota that is similar to NPAM. The management objective of the GMT, while inclusive of the NPAM objective, is farther reaching and includes floristic diversity and structural diversity. The NPS and USFS both expressed that their objectives were compatible with that of NPAM, however, neither provided sufficient detail of their objectives to allow an external assessment of the level of agreement. NDGF and SDGFP management objectives are currently not well aligned with that of NPAM, as they are focused on grassland habitat as a means to support wildlife, game species, and hunting opportunities. Some respondents, however, noted that a shift that better balances healthy native plant communities with their current wildlife-centric objectives might be possible. BOR did not provide information regarding their management objective. The lack of response here, taken together with responses to other questions, leads us to conclude that the objective of BOR is not compatible with NPAM.

<u>Management Actions</u> – MNDNR and TNC appear to be the most compatible in the type of management actions available to implement and the ability to make and carry out annual decisions and actions according to the NPAM protocol. The respondent from SDGFP reported abilities that are potentially aligned with that of NPAM; however, it was noted that implementation of prescribed burns is difficult, and lack of financial and human resources would make enrollment of more than a few units unlikely. NDGF also appears to have a management structure similar to NPAM (e.g., land unit decision makers and annual decisions); however, the agency has even more restrictions than SDGFP regarding abilities to carry out prescribed burns, and in some cases, even grazes. NPS, as a whole, does not appear to have a setup that is conducive to participation in NPAM. One NPS respondent reported possible alignment with the NPAM management protocol; however, this respondent noted the inability to implement actions on entire units because the NPS land is not divided into management units. The other NPS respondent, who was responding for two specific parks (Pipestone and Tallgrass Native Prairie), reported complete incompatibility with the NPAM management action protocol. The management structure of the USFS is also incongruous with that of NPAM; reasons include incompatibility with the management year, inability to make annual decisions on an independent basis over time, inability to treat entire units, and exclusion of broadcast herbicide use as an admissible action. Lastly, the BOR respondent made it clear on several points that the NPAM management protocol was not suitable for the BOR.

<u>Vegetation Monitoring</u> – Responses indicated that MNDNR and TNC, who participate on the GMT, have the most potential to align with NPAM, though they have definite incongruities. As noted earlier, the GMT monitoring protocols allow participants to choose among three different levels of monitoring detail, dependent upon their interests, all of which meet the NPAM needs in terms of determining cover of NP, SB, KB, and RM. However, GMT monitoring is carried out at a lower spatial density than is NPAM (1 transect per 10 acres as opposed to 1 transect per 5 acres) and is carried out less frequently (every 3-5 years as opposed to annually). Both MNDNR and TNC respondents noted insufficient resources as a potential conflict in carrying out the higher density and frequency of monitoring required by the NPAM protocol. One NPS respondent stated the existence of monitoring on their land and noted the potential to carry out monitoring according to the NPAM protocol. Both NDGF and SDGFP reported an overall lack of monitoring and lack of resources to carry out monitoring according to the NPAM protocol. USFS and BOR also reported the inability to complete NPAM-consistent monitoring.

<u>Data Sharing</u> – The intent of this question was to ascertain the willingness to share data, not the ability to collect and enter the data. The intended question was, “Given you have the ability to collect and enter the data according to NPAM protocols, are you willing to share the data in the NPAM database?” The question, unfortunately, was not worded this clearly; given the statement of the BOR respondent

citing lack of staff time as the reason for the negative response, it is likely that other respondents, those whom responded “maybe”, were interpreting this question in various ways. Additionally, the proprietary nature of the comment from a NDGF respondent regarding the expectation of two-way sharing indicates an incomplete understanding of how the data in the NPAM databased are used. It was explained in the webinar that all NPAM cooperators benefit from participating in NPAM by receiving annual management guidance for their units based on the current state of each unit (as processed with the management and monitoring data entered per unit) and the current state of understanding regarding how the system responds (as processed given the data provided across all units). In addition to the management guidance, cooperators receive detailed summaries of their entered data, which they can use however they see fit. Various NPAM cooperators receive summaries of their own data; they do not receive summaries or raw data from other cooperators. Aside from these unfavorable responses, several respondents were non-committal regarding their willingness to share their monitoring and management data on units enrolled in NPAM; these included USFS and NDGF. MNDNR and TNC were agreeable to sharing their data, again congruent with NPAM protocols.

<u>Involvement</u> – TNC, by their own assertion, is the most likely to want to participate and enroll units in NPAM, as well as have the resources to do so; they did, however, cite funding and staff as the limiting factor to their potential involvement. Obstacles aside, the SDGFP respondent expressed high interested in participation. MNDNR, while expressing only tentative interest in participation, cited the agency’s existing commitment to the GMT as the primary reason. The underlying implication was that if it could demonstrate how NPAM and GMT could complement each other (i.e., not require investment in a whole new protocol on separate units and not undermine the current protocols on the existing GMT units) then MNDNR would have interest in participation in NPAM. NDGF interest in NPAM participation proved to be equivocal, at best; lack of resources, in addition to the fact that most native sod units are located outside of the PPR and thus would not meet NPAM criteria, were noted. Interest for NPS involvement was also dubious. One NPS respondent proposed the potential to try NPAM in Knife River Indian Villages, and if successful, then expand involvement with other parks. Knife River, however, consists of short grass prairie and thus would not meet the NPAM criteria. Another NPS respondent suggested interest would potentially increase if NPAM was expanded to include restored grasslands, a proposal that is outside the bounds of NPAM. Use of inadmissible NPAM actions (e.g., broadcast herbicide and biocontrol agents) excludes USFS from participating in NPAM. Lastly, the BOR respondent expressed no interest in participating in NPAM.

<u>Other</u> – Overall, TNC respondents had the most optimistic view of their potential participation in NPAM, including the benefits seen to the collaboration, as well as the magnitude of existing obstacles and their ability to overcome these obstacles. The MNDNR respondent also saw the benefits of NPAM participation, but indicated unwillingness to consider such participation until integration of NPAM and the GMT could be established. Though the SDGFP respondent was demonstrably quite favorable of the NPAM initiative and showed definite interest in SDGFP collaboration, current obstacles abound; lack of financial and human resources to carry out management actions, vegetation monitoring, and data entry, as well as a basic conflict in the management objective were among these hindrances. Despite these issues, the SDGFP respondent exhibited enthusiasm regarding interest in overcoming these barriers.

CONCLUSIONS

We diligently attempted to reach out to non-USFWS NWRS agencies and organizations that manage native prairie lands within the PPR. In response to our reach-out efforts, we attracted representatives of seven different agencies/organizations to participate in an informational webinar and to complete a post-webinar questionnaire aimed at gauging their interest and assessing the compatibility between their agency/organization management approach and that of NPAM.

Overall, TNC, and to a lesser extent, the MNDNR, have the highest compatibility with NPAM. Not coincidentally, both organizations are participants in the GMT adaptive management program, which is active in the tallgrass prairies of Minnesota, a fact that ostensibly acts as a deterrent to the MNDNR's interest in participating in NPAM. The compatibility of TNC's objectives, type and implementation of management, and type and implementation of vegetation monitoring is not without hurdles. Funding and staff are the main impediments, but are not identified as insurmountable problems. Should NPAM expand to include outside partners, TNC interest is strong and their likeness to NPAM makes them a viable partner. If NPAM managers determined that NPAM could easily integrate with the GMT program, then it is likely that the MNDNR would also be an interested and capable partner. The main incompatibility between NPAM and the GMT program is the frequency of management and monitoring; the NPAM initiative is built around an annual timeline, whereas the GMT program is designed around a 3-5 year timeline. This timeline discordance, unfortunately, does not have a straightforward workaround. Interest from the MNDNR, however, may resurface if they have access to more funding and human resources, or if the GMT program comes to an end. We are not aware of the future plans for the GMT program; however, it would be worthwhile to investigate the potential integration of these two programs. Such an examination is especially relevant given the paucity of tallgrass units enrolled in NPAM and the potential gain to NPAM if tallgrass units are added through partnerships.

Collaboration of NPAM with the other agencies with which we made contact – NDGF, SDGFP, NPS, USFS, and BOR – seemed far less likely for a variety of reasons noted above. If NPAM managers were to attempt to work through stated obstacles, it would be most beneficial to the NPAM initiative to focus on the agencies that manage tallgrass units – particular districts of NDGF and SDGFP, NPS, and USFS.

Possible cross-agency arrangements of resource sharing may permit small-scale expansions of NPAM to other partners. For example, parcels of native prairie that are under different organizational ownerships are sometimes found adjacent to each other. In these cases, it may be feasible to design schemes for sharing monitoring effort and access to management tools. Such schemes may allow some of the lowest participation obstacles to be overcome with minimal additional time and resource burden by any one organization.

Expansion of NPAM beyond its current USFWS NWRS cooperators is contingent on more than simply interest and compatibility with outside partners; NPAM may currently lack the capacity to accommodate these additional partners. We have not made an official investigation as to the capacity of NPAM, but some initial forays of the topic with the NPAM Advisory Team lead us to conclude that NPAM does not currently have the infrastructure necessary to support such an expansion. Concerns regarding additional workload and responsibility for the NPAM coordinator and NPAM database coordinator were legitimately raised. Furthermore, the centralized cooperator database and the coordinator database would require modification to accommodate outside partners. Such modification would require additional funding and time. A cursory cost-benefit analysis reveals that the added encumbrance of time and resources by USFWS NWRS NPAM cooperators in supporting a partnership would not be outweighed by the potential gain in learning arising by a partnership. Under current circumstances, a pursuit of NPAM to non-NWRS partners does not appear constructive.

LITERATURE CITED

Akaike, H. 1974. A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19:716-723.

Burnham, K. P., and D. R. Anderson. 1998. Model selection and multimodel inference: a practical information-theoretic approach. Springer-Verlag. New York, New York, USA.

Fackler, P. L. 2015. MDPSolve: Software for dynamic optimization. Version 2015b. https://github.com/PaulFackler/MDPSolve/

Gannon, J. J., T. L. Shaffer, and C. T. Moore. 2013. Native Prairie Adaptive Management: a multi region adaptive approach to invasive plant management on Fish and Wildlife Service owned native prairies. U.S. Geological Survey Open File Report 2013-1279.

Gelman, A., and D. B. Rubin. 1992. Inference from iterative simulation using multiple sequences. Statistical Science 7:457-511.

Grant, T. A., E. M. Madden, R. K. Murphy, K. A. Smith, and M. P. Nenneman. 2004. Monitoring native prairie vegetation: the belt transect method. Ecological Restoration 22:106-112.

Kellner, K. 2016. Package jagsUI: A wrapper around 'rjags' to streamline 'JAGS' analyses. Version 1.4.4. https://github.com/kenkellner/jagsUI

Lubow, B. C. 1995. SDP—Generalized software for solving stochastic dynamic optimization problems. Wildlife Society Bulletin 23:738–742.

Lubow, B. C. 1997. Adaptive stochastic dynamic programming (ASDP) — Supplement to SDP user’s guide, version 2.0., Fort Collins, Colo., Colorado Cooperative Fish and Wildlife Research Unit, Colorado State University, 31 p.

Moore, C. T., and M. J. Conroy. 2006. Optimal regeneration planning for old-growth forest: Addressing scientific uncertainty in endangered species recovery through adaptive management. Forest Science 52: 155-172.

Nicol, S., and I. Chadès. 2011. Beyond stochastic dynamic programming: a heuristic sampling method for optimizing conservation decisions in very large state spaces. Methods in Ecology and Evolution 2:221-228.

Plummer, M. 2003. JAGS: a program for analysis of Bayesian graphical models using Gibbs sampling. Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003), 20-22 March 2003, Vienna, Austria.

R Core Team. 2016. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

SAS Institute Inc. 2013. SAS/STAT® 13.1 User’s Guide. Cary, NC: SAS Institute, Inc.

Williams, B. K. 1996. Adaptive optimization and the harvest of biological populations. Mathematical Biosciences 136:1-20.

Table 1.1. Response-specific constraints applied to linear models to approximate biological relationships expressed in conceptual models for mixed grass.

Model	Response		
	SB	KB	RM
1	$\tau_q^{(\text{Graze})} = \tau_q^{(\text{Burn})} = \tau_q^{(\text{Burn/Graze})}$	$\tau_q^{(\text{Graze})} = \tau_q^{(\text{Burn})} = \tau_q^{(\text{Burn/Graze})}$	$\tau_q^{(\text{Graze})} = \tau_q^{(\text{Burn})} = \tau_q^{(\text{Burn/Graze})}$
2	$(\tau\delta)_{\text{SB}}^{(\text{Graze,SB})} = (\tau\delta)_{\text{SB}}^{(\text{Burn/Graze,SB})}$ $(\tau\delta)_{\text{SB}}^{(\text{Graze,CO})} = (\tau\delta)_{\text{SB}}^{(\text{Burn,CO})}$	$(\tau\delta)_{\text{KB}}^{(\text{Burn,KB})} = (\tau\delta)_{\text{KB}}^{(\text{Burn/Graze,KB})}$ $(\tau\delta)_{\text{KB}}^{(\text{Graze,CO})} = (\tau\delta)_{\text{KB}}^{(\text{Burn,CO})}$	$(\tau\delta)_{\text{RM}}^{(\text{Graze,RM})} = (\tau\delta)_{\text{RM}}^{(\text{Burn,RM})} = (\tau\delta)_{\text{RM}}^{(\text{Burn/Graze,RM})}$
3	$(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/Graze,SB})}$ $(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/Graze,CO})}$	$(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/Graze,KB})}$ $(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/Graze,CO})}$	$(\tau\delta\gamma)_{\text{RM}}^{(\text{Graze,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Burn,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Burn/Graze,RM})}$
4	$(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/Graze,SB})}$ $(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/Graze,CO})}$	$(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/Graze,KB})}$ $(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/Graze,CO})}$	$(\tau\delta\gamma)_{\text{RM}}^{(\text{Graze,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Burn,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Burn/Graze,RM})}$

Table 1.2. Response-specific constraints applied to linear models to approximate biological relationships expressed in conceptual models for tallgrass. Graze/W= Graze within phenological window. Burn/W= Burn within phenological window. Defol=Defoliate.

Model	Response		
	SB	KB	RM
1	$\tau_q^{(\text{Graze/W})} = \tau_q^{(\text{Burn/W})} = \tau_q^{(\text{Defol})}$	$\tau_q^{(\text{Graze/W})} = \tau_q^{(\text{Burn/W})} = \tau_q^{(\text{Defol})}$	$\tau_q^{(\text{Graze/W})} = \tau_q^{(\text{Burn/W})} = \tau_q^{(\text{Defol})}$
2			$(\tau\delta)_{\text{RM}}^{(\text{Graze/W,RM})} = (\tau\delta)_{\text{RM}}^{(\text{Burn/W,RM})} = (\tau\delta)_{\text{RM}}^{(\text{Defol,RM})}$
3	$(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze/W,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/W,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Defol,SB})}$ $(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze/W,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/W,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Defol,CO})}$	$(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze/W,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/W,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Defol,KB})}$ $(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze/W,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/W,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Defol,CO})}$	$(\tau\delta\gamma)_{\text{RM}}^{(\text{Graze/W,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Burn/W,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Defol,RM})}$
4	$(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze/W,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/W,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Defol,SB})}$ $(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze/W,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/W,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Defol,CO})}$	$(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze/W,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/W,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Defol,KB})}$ $(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze/W,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/W,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Defol,CO})}$	$(\tau\delta\gamma)_{\text{RM}}^{(\text{Graze/W,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Burn/W,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Defol,RM})}$
5	$(\tau\delta)_{\text{SB}}^{(\text{Graze/W,SB})} = (\tau\delta)_{\text{SB}}^{(\text{Burn/W,SB})}$ $(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze/W,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/W,SB})}$ $(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze/W,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/W,CO})}$	$(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze/W,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/W,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Defol,KB})}$ $(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze/W,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/W,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Defol,CO})}$	$(\tau\delta\gamma)_{\text{RM}}^{(\text{Graze/W,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Burn/W,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Defol,RM})}$
6	$(\tau\delta)_{\text{SB}}^{(\text{Defol,SB})} = (\tau\delta)_{\text{SB}}^{(\text{Rest,SB})}$ $(\tau\delta)_{\text{SB}}^{(\text{Defol,CO})} = (\tau\delta)_{\text{SB}}^{(\text{Rest,CO})}$ $(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze/W,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/W,SB})}$ $(\tau\delta\gamma)_{\text{SB}}^{(\text{Defol,SB})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Rest,SB})}$ $(\tau\delta\gamma)_{\text{SB}}^{(\text{Graze/W,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Burn/W,CO})}$ $(\tau\delta\gamma)_{\text{SB}}^{(\text{Defol,CO})} = (\tau\delta\gamma)_{\text{SB}}^{(\text{Rest,CO})}$	$(\tau\delta)_{\text{KB}}^{(\text{Defol,KB})} = (\tau\delta)_{\text{KB}}^{(\text{Rest,KB})}$ $(\tau\delta)_{\text{KB}}^{(\text{Defol,CO})} = (\tau\delta)_{\text{KB}}^{(\text{Rest,CO})}$ $(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze/W,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/W,KB})}$ $(\tau\delta\gamma)_{\text{KB}}^{(\text{Defol,KB})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Rest,KB})}$ $(\tau\delta\gamma)_{\text{KB}}^{(\text{Graze/W,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Burn/W,CO})}$ $(\tau\delta\gamma)_{\text{KB}}^{(\text{Defol,CO})} = (\tau\delta\gamma)_{\text{KB}}^{(\text{Rest,CO})}$	$(\tau\delta)_{\text{RM}}^{(\text{Defol,RM})} = (\tau\delta)_{\text{RM}}^{(\text{Rest,RM})}$ $(\tau\delta\gamma)_{\text{RM}}^{(\text{Graze/W,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Burn/W,RM})}$ $(\tau\delta\gamma)_{\text{RM}}^{(\text{Defol,RM})} = (\tau\delta\gamma)_{\text{RM}}^{(\text{Rest,RM})}$

Table 1.3. Equality constraints imposed in linear models to create estimable functions for effects supported by insufficient data. Constraints were applied to the model for each response q = {SB, KB, RM}.

Mixed grass, effect $(\tau\delta\gamma)_q^{(\mathrm{Burn,RM})}$: 1 data sample

$$(\tau\delta\gamma)_q^{(\mathrm{Burn,RM})} = \tfrac{1}{3}\sum_{i\neq\mathrm{Burn}}(\tau\delta\gamma)_q^{(i,\mathrm{RM})} + \tfrac{1}{9}\sum_{i\neq\mathrm{Burn}}\sum_{j\neq\mathrm{RM}}(\tau\delta\gamma)_q^{(i,j)} - \tfrac{1}{3}\sum_{i\neq\mathrm{RM}}(\tau\delta\gamma)_q^{(\mathrm{Burn},i)}$$

Tallgrass, effect $(\tau\delta\gamma)_q^{(\mathrm{Burn/W,CO})}$: 1 data sample

$$(\tau\delta\gamma)_q^{(\mathrm{Burn/W,CO})} = \tfrac{1}{3}\sum_{i\neq\mathrm{Burn/W}}(\tau\delta\gamma)_q^{(i,\mathrm{CO})} + \tfrac{1}{6}\sum_{i\neq\mathrm{Burn/W}}\sum_{j\neq\mathrm{CO,RM}}(\tau\delta\gamma)_q^{(i,j)} - \tfrac{1}{2}\sum_{i\neq\mathrm{CO,RM}}(\tau\delta\gamma)_q^{(\mathrm{Burn/W},i)}$$

Tallgrass, effect $(\tau\delta\gamma)_q^{(\mathrm{Burn/W,RM})}$: 1 data sample

$$(\tau\delta\gamma)_q^{(\mathrm{Burn/W,RM})} = \tfrac{1}{3}\sum_{i\neq\mathrm{Burn/W}}(\tau\delta\gamma)_q^{(i,\mathrm{RM})} + \tfrac{1}{6}\sum_{i\neq\mathrm{Burn/W}}\sum_{j\neq\mathrm{CO,RM}}(\tau\delta\gamma)_q^{(i,j)} - \tfrac{1}{2}\sum_{i\neq\mathrm{CO,RM}}(\tau\delta\gamma)_q^{(\mathrm{Burn/W},i)}$$

Table 1.4. Model selection and mean squared error results for linear models parameterized to approximate NPAM conceptual models.

			Response	SM			Response	KB			Response	RM	
	Model	df	MSE	AIC	w	df	MSE	AIC	w	df	MSE	AIC	w
Mixed Grass	1	3	0.818	-105.6	0.000	3	1.307	147.8	0.000	3	1.192	98.2	0.150
(n=541)	2	15	0.782	-118.6	0.015	15	1.249	135.2	0.000	15	1.160	94.9	0.779
	3	28	0.767	-116.4	0.005	28	1.226	137.5	0.000	31	1.151	106.2	0.003
	4	32	0.747	-126.9	0.979	32	1.173	117.5	1.000	35	1.130	99.7	0.068
Tallgrass	1	3	0.935	-12.6	1.000	3	1.338	70.2	0.998	3	1.493	95.5	0.984
(n=231)	2	17	0.954	5.5	0.000	17	1.335	83.1	0.002	15	1.471	103.7	0.016
	3	28	0.944	12.9	0.000	28	1.337	93.2	0.000	30	1.515	123.8	0.000
	4	32	0.934	13.8	0.000	32	1.309	91.8	0.000	34	1.513	126.9	0.000
	5	29	0.949	15.0	0.000	28	1.337	93.2	0.000	30	1.515	123.8	0.000
	6	26	0.943	10.9	0.000	26	1.326	89.5	0.000	29	1.540	126.7	0.000

Table 2.1. Estimated effects (mean, standard deviation, 95% credible interval) from a linear-logit model of counts of native-dominated transect segments for mixed grass prairie units in NPAM (n = 92 units; 1837 transects), 2010-2016. All predictor covariates (long-term precipitation CP_i, long-term temperature CT_i, prior-year total precipitation $P_{it,}$ invasion vulnerability Z_{ij}, prior-year native prairie composition proportion x_{ijt}) were centered to their mean; some (CP_i, CT_i, P_{it}) were also scaled to unit variance.

Effect	**Mean**	**SD**	**Credible Interval**
Mean (μ)	-1.030	0.055	(-1.137, -0.927)
Unit-level effects			
Climate: precipitation υ (CP)	-0.177	0.037	(-0.246, -0.106)
Climate: temperature υ (CT)	-0.100	0.032	(-0.162, -0.037)
Climate: interaction υ(CP x CT)	0.098	0.016	(0.066, 0.130)
Prior-year total precipitation (β)	0.220	0.051	(0.120, 0.321)
Treatment (τ)			
Rest	-0.686	0.062	(-0.801, -0.566)
Graze	-0.412	0.059	(-0.525, -0.295)
Burn	0.959	0.071	(0.822, 1.102)
Burn/Graze	0.941	0.096	(0.749, 1.125)
Unassignable	-0.802	0.185	(-1.171, -0.428)
Treatment x Precipitation ($\tau\beta$)			
Rest	-0.160	0.053	(-0.265, -0.055)
Graze	-0.097	0.053	(-0.199, 0.011)
Burn	-0.127	0.069	(-0.270, 0.002)
Burn/Graze	-0.105	0.118	(-0.342, 0.114)
Unassignable	0.490	0.128	(0.241, 0.736)
Transect-level effects			
Invasion vulnerability (α)	-0.950	0.213	(-1.379, -0.564)
Treatment x Invasion vulnerability ($\tau\alpha$)			
Rest	-0.120	0.233	(-0.582, 0.350)
Graze	0.881	0.232	(0.455, 1.340)
Burn	0.477	0.282	(-0.064, 1.049)
Burn/Graze	-0.213	0.338	(-0.858, 0.455)
Unassignable	-1.026	0.776	(-2.664, 0.376)
Prior-year native prairie (δ)	6.224	0.071	(6.082, 6.363)
Random effects			
Annual level (σ_η)	0.531	0.234	(0.273, 1.111)
Transect level (σ_ε)	2.331	0.022	(2.287, 2.375)

Table 2.2. Estimated effects (mean, standard deviation, 95% credible interval) from a linear-logit model of counts of native-dominated transect segments for all prairie units in NPAM (mixed grass: *n* = 92 units, 1837 transects; tallgrass: n = 46 units, 658 transects), 2010-2016. All predictor covariates (long-term precipitation CP_i, long-term temperature CT_i, prior-year total precipitation $P_{it,}$ prior-year native prairie composition proportion x_{ijt}) were centered to their mean; some (CP_i, CT_i, P_{it}) were also scaled to unit variance.

Effect	**Mean**	**SD**	**Credible Interval**
Mean (μ)	-0.837	0.051	(-0.933, -0.732)
Grassland-level effects (γ)			
Mixed grass	-0.159	0.051	(-0.258, -0.057)
Tallgrass	0.159	0.051	(0.057, 0.258)
Unit-level effects			
Climate: precipitation υ (*CP*)	-0.039	0.044	(-0.123, 0.047)
Climate: temperature υ (*CT*)	-0.165	0.027	(-0.220, -0.113)
Climate: interaction υ(*CP* x *CT*)	0.057	0.021	(0.017, 0.097)
Prior-year total precipitation (β)	0.036	0.065	(-0.107, 0.143)
Treatment (τ)^1			
A	-0.513	0.057	(-0.630, -0.409)
B	-0.303	0.085	(-0.478, -0.146)
C	0.607	0.092	(0.420, 0.785)
D	0.297	0.071	(0.150, 0.431)
E	-0.088	0.172	(-0.421, 0.249)
Grass Type x Precipitation ($\gamma\beta$)			
Mixed grass	0.169	0.064	(0.052, 0.303)
Tallgrass	-0.169	0.064	(-0.303, -0.052)
Treatment x Precipitation ($\tau\beta$)^1			
A	-0.213	0.068	(-0.346, -0.085)
B	0.130	0.082	(-0.027, 0.294)
C	-0.028	0.095	(-0.219, 0.156)
D	-0.015	0.092	(-0.199, 0.166)
E	0.127	0.189	(-0.267, 0.511)
Treatment x Grass Type ($\tau\gamma$)			
Mixed: Rest	-0.218	0.058	(-0.333, -0.108)
Mixed: Graze	-0.096	0.088	(-0.262, 0.073)
Mixed: Burn	0.281	0.088	(0.105, 0.464)
Mixed: Burn/Graze	0.645	0.071	(0.509, 0.785)
Mixed: Unassignable	-0.612	0.164	(-0.926, -0.269)
Tallgrass: Rest	0.218	0.058	(0.108, 0.333)
Tallgrass: Graze within Window	0.096	0.088	(-0.073, 0.262)
Tallgrass: Burn within Window	-0.281	0.088	(-0.464, -0.105)
Tallgrass: Defoliate	-0.645	0.071	(-0.785, -0.509)
Tallgrass: Unassignable	0.612	0.164	(0.269, 0.926)

Effect	**Mean**	**SD**	**Credible Interval**
Treatment x Grass Type x Precipitation ($\tau\gamma\beta$)			
Mixed: Rest	-0.026	0.067	(-0.159, 0.098)
Mixed: Graze	-0.197	0.079	(-0.351, -0.047)
Mixed: Burn	-0.089	0.096	(-0.268, 0.115)
Mixed: Burn/Graze	-0.059	0.089	(-0.237, 0.112)
Mixed: Unassignable	0.370	0.188	(-0.014, 0.747)
Tallgrass: Rest	0.026	0.067	(-0.098, 0.159)
Tallgrass: Graze within Window	0.197	0.079	(0.047, 0.351)
Tallgrass: Burn within Window	0.089	0.096	(-0.115, 0.268)
Tallgrass: Defoliate	0.059	0.089	(-0.112, 0.237)
Tallgrass: Unassignable	-0.370	0.188	(-0.747, 0.014)
Transect-level effects			
Prior-year native prairie (δ)	6.438	0.062	(6.316, 6.558)
Random effects			
Annual level: Mixed grass (σ_η)	0.515	0.219	(0.265, 1.068)
Annual level: Tallgrass (σ_η)	0.457	0.233	(0.177, 1.030)
Transect level (σ_ε)	2.281	0.019	(2.244, 2.320)

[1] Treatments differ by grass type; therefore, treatment effects in absence of interaction with grass type are not directly interpretable. Thus, treatment levels are represented by nominal indicators.

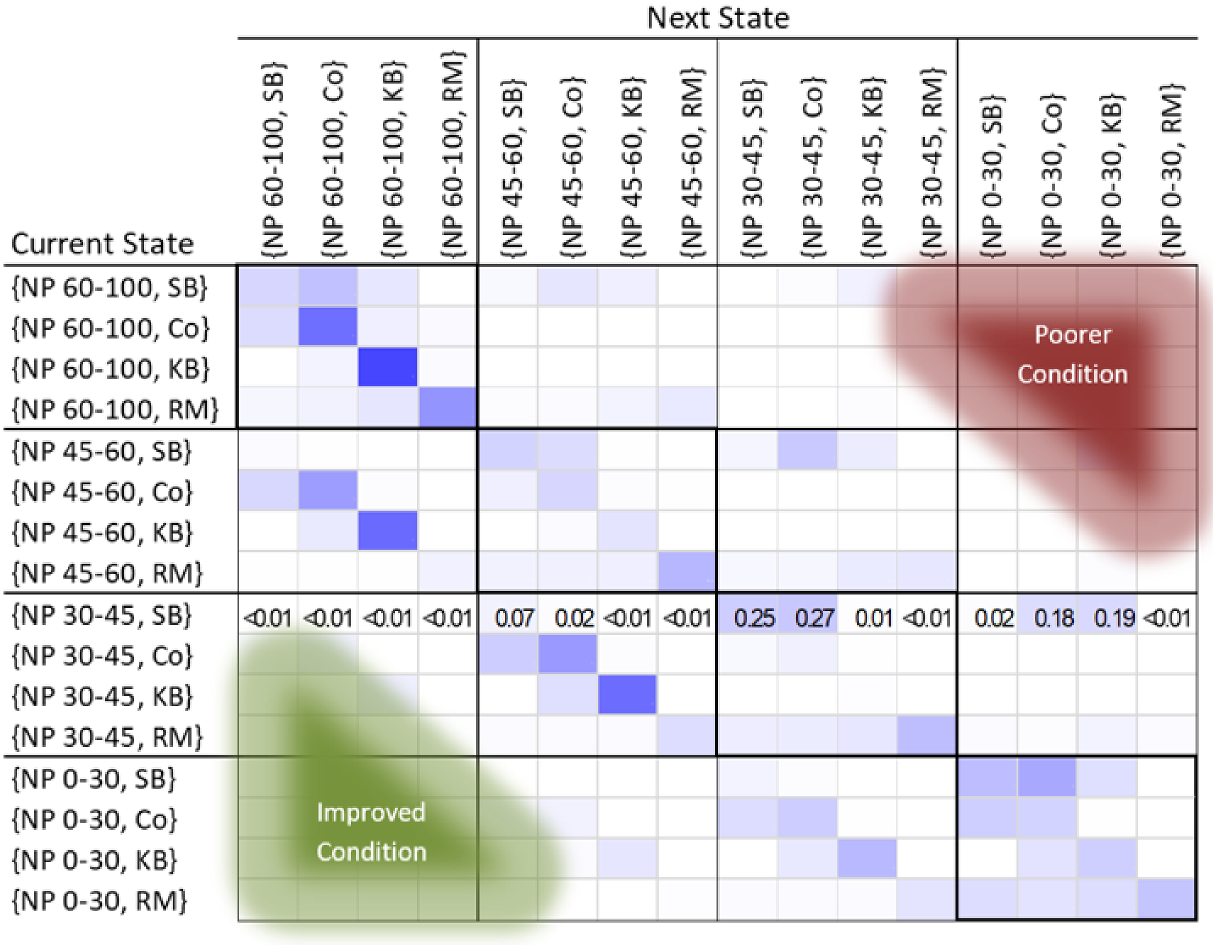
Table 3.1: Federal, state, and non-government organization individuals identified and contacted regarding interest in participating in an informational webinar and follow-up questionnaire. The "Contact" column denotes whether the person was on the original contact list ('O'), received the invite via forward from an original recipient ('F'), or was an expected participant on the day of the webinar ('N'). The "Interest" column indicates, with an 'X', those whom affirmed their interest in participating in an informational webinar and follow-up questionnaire. Presence during the webinar is indicated in the "Webinar Presence" column with an 'X'. An 'X' in the "Questionnaire" column indicates that the person completed and submitted the questionnaire after the webinar. One individual had another employee who did not participate in the webinar complete the questionnaire (BOR). One individual did not complete the questionnaire but provided written feedback in an email (denoted as "Narrative").

Type	Organization[1]	Position	Contact	Interest	Webinar Presence	Questionnaire
Federal	BLM	Biologist	O			
Federal	BLM	Biologist	O			
Federal	BOR	Environmental Specialist	O			
Federal	BOR	Environmental Specialist	O		X	
Federal	BOR	Natural Resource Specialist	N			X
Federal	BOR	Manager	O			
Federal	BOR	Director	O			
Federal	NPS	Exotic Plant Management Team	F	X		
Federal	NPS	Plant Ecologist	F	X		
Federal	NPS	I&M Program Coordinator	O	X	X	
Federal	NPS	Invasive Plant Program Leader	F	X	X	Narrative
Federal	USFS	Range Management Specialist	O	X	X	X
Federal	USFS	Unknown	F	X		
State	MNDNR	Ecologist	O		X	X
State	MNDNR	Prairie Ecologist	O	X		
State	MTFWP	Biologist	O			
State	MTFWP	Biologist	O			
State	MTFWP	Biologist	O			
State	MTFWP	Biologist	O			
State	NDGF	Division Assistant Chief	N		X	X
State	NDGF	Management Supervisor	F	X	X	X

Type	Organization[1]	Position	Contact	Interest	Webinar Presence	Questionnaire
State	NDGF	Management Supervisor	F	X	X	X
State	NDGF	Biologist	O		X	
State	NDGF	Private Lands Program Section Leader	F	X	X	
State	NDGF	Management Section Leader	F	X		
State	NDGF	Management Supervisor	F	X	X	X
State	NDGF	Private Lands Biologist	N		X	
State	SDGFP	Biologist	O	X		
State	SDGFP	Habitat Program Administrator	F	X	X	X
State	SDGFP	Unknown	F	X		
Non-Government	DU	Biologist	O			
Non-Government	DU	Manager	O			
Non-Government	DU	Manager	O			
Non-Government	TNC	Prairie Ecologist (MN, ND, SD)	O	X	X	X
Non-Government	TNC	Preserve Manager (SD)	F	X	X	X
Non-Government	TNC	Preserve Manager (ND)	F	X	X	X
Other	PPP LCC	Science Coordinator	O	X	X	

[1] Bureau of Land Management (BLM); Bureau of Reclamation (BOR); National Park Service (NPS); U.S. Forest Service (USFS); Minnesota Department of Natural Resources (MNDNR); Montana Fish, Wildlife, and Parks (MTFWP); North Dakota Game and Fish (NDGF); South Dakota Game, Fish, and Parks (SDGFP); Ducks Unlimited (DU); The Nature Conservancy (TNC); Plains and Prairie Pothole Landscape Conservation Cooperative (PPP LCC)

Figure 1.1. A portion of the estimated state and transition model for mixed grass Model 2, for the condition of combination burn/graze treatment and intermediate defoliation index. Rows of the matrix represent current vegetation state, and columns represent state at next time period. On each row are probabilities of transitioning from the current state to each of the 16 possible states in the next time period. The region to the right of the diagonal represents movement to poorer condition (less native prairie cover), the region to the left represents improvement (more native prairie cover), the 4x4-cell diagonal represents maintenance of the same native prairie cover (without respect to type of dominant invader), and the 1x1 cell diagonal represents no change in vegetation state (same native prairie cover and same dominant invader). Cell shading is proportional to probability, with lighter shading representing lower probability and darker shading representing higher probability. Probability values are revealed for one of the rows.



Current State	{NP 60-100, SB}	{NP 60-100, Co}	{NP 60-100, KB}	{NP 60-100, RM}	{NP 45-60, SB}	{NP 45-60, Co}	{NP 45-60, KB}	{NP 45-60, RM}	{NP 30-45, SB}	{NP 30-45, Co}	{NP 30-45, KB}	{NP 30-45, RM}	{NP 0-30, SB}	{NP 0-30, Co}	{NP 0-30, KB}	{NP 0-30, RM}
{NP 60-100, SB}																
{NP 60-100, Co}																
{NP 60-100, KB}																
{NP 60-100, RM}																
{NP 45-60, SB}																
{NP 45-60, Co}																
{NP 45-60, KB}																
{NP 45-60, RM}																
{NP 30-45, SB}	<0.01	<0.01	<0.01	<0.01	0.07	0.02	<0.01	<0.01	0.25	0.27	0.01	<0.01	0.02	0.18	0.19	<0.01
{NP 30-45, Co}																
{NP 30-45, KB}																
{NP 30-45, RM}																
{NP 0-30, SB}																
{NP 0-30, Co}																
{NP 0-30, KB}																
{NP 0-30, RM}																

Figure 1.2. Displays of the expert-elicited, empirically-derived, and composite versions of the state and transition model for mixed grass Model 2, for the condition of combination burn/graze treatment and intermediate defoliation index. Average composite standard error for the rows of the empirical matrix, representing precision of values in the row, is displayed as a column of gray-shaded cells, with lightest cells representing least precision (largest standard error) and darkest cells representing highest precision (smallest standard error). For illustration, the shading of the precision column is relative to the minimum standard error within the column; it is not absolute. The composite version of the model gives weight (θ, displayed as a column of numbers to the left of the composite matrix) to the empirically-derived matrix, row by row, in accordance to degree of precision available for the empirical result; complementary weight ($1 - \theta$) is assigned to rows of the expert-elicited matrix.

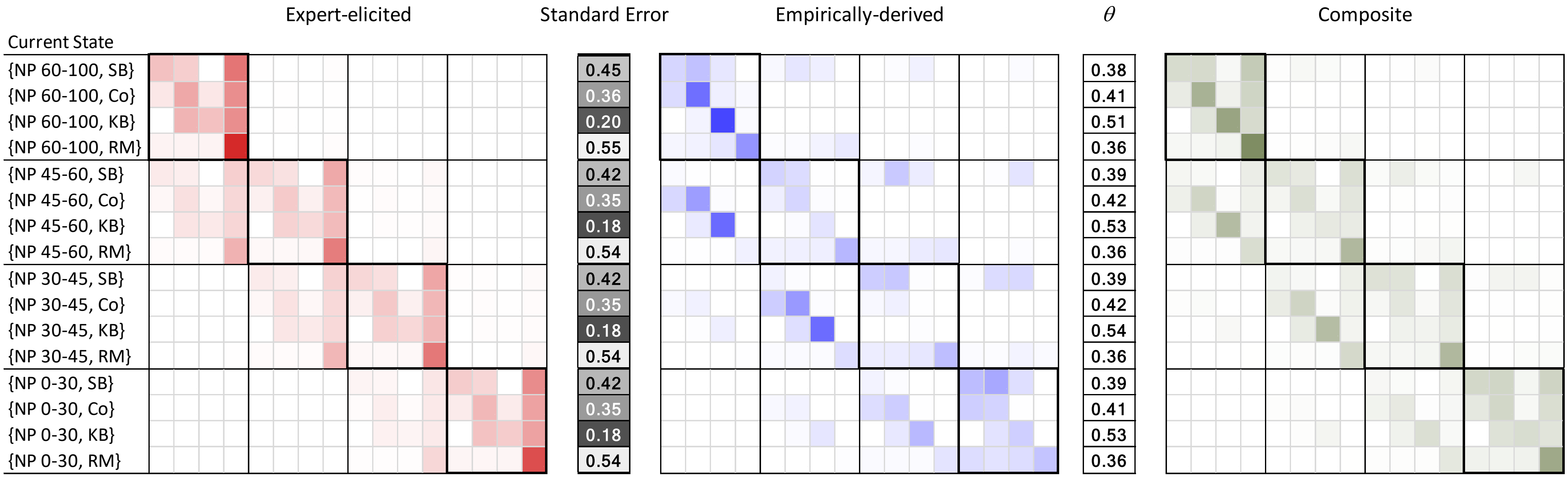


Figure 2.1. Estimated response (mean proportion of native vegetative cover) to combinations of treatment (Rest, Graze, Burn, Burn/Graze Combination) and ESD Invasion Vulnerability score (0, 0.5, 1.0), based on data from mixed grass units only.

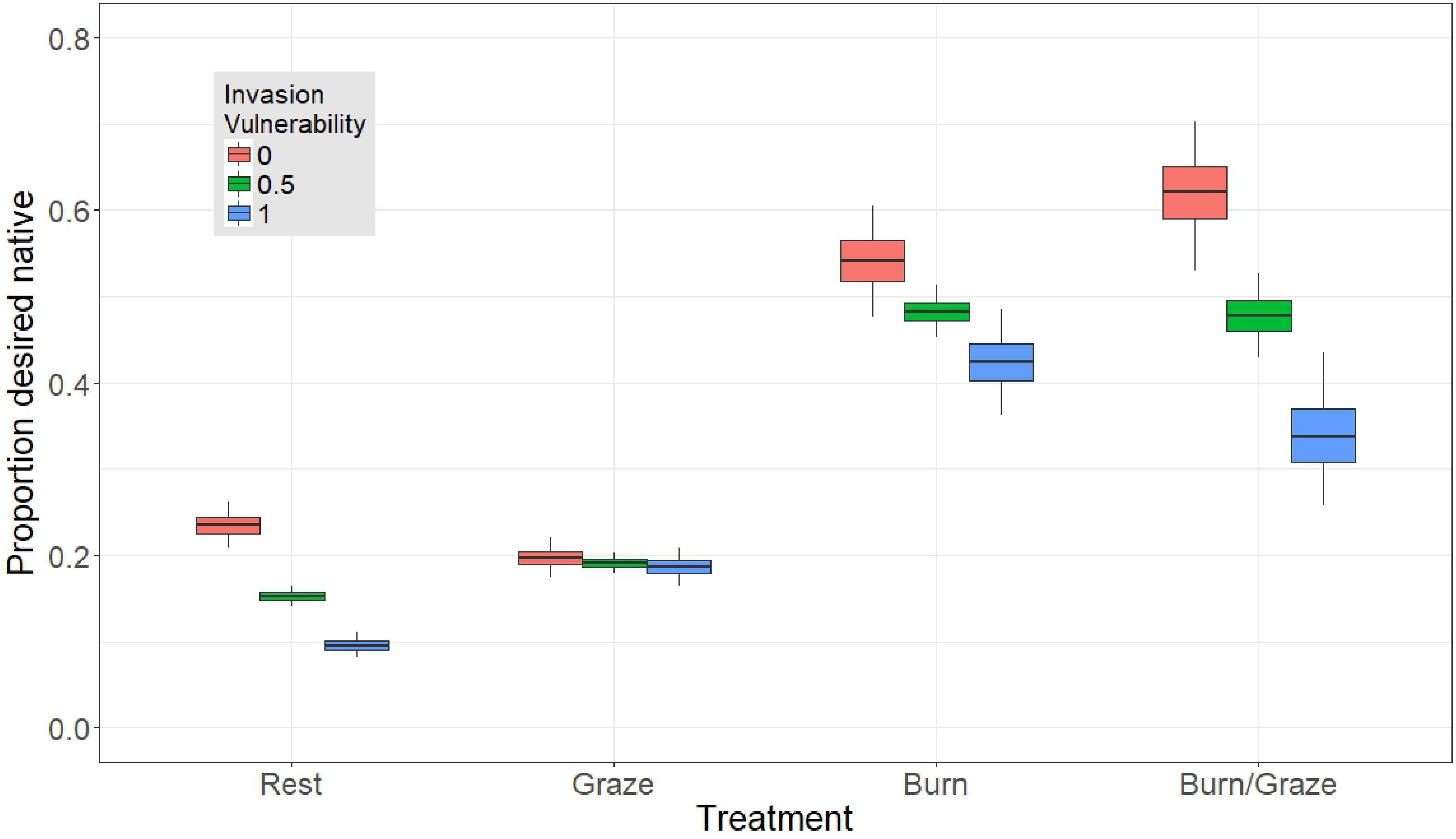


Figure 2.2. Estimated response (mean proportion of native vegetative cover) to combinations of treatment (Rest, Graze, Burn, Burn/Graze Combination) and prior-year rainfall (mean – 2SD [Dry], mean [Average], mean + 2SD [Wet]) for mixed grass units.

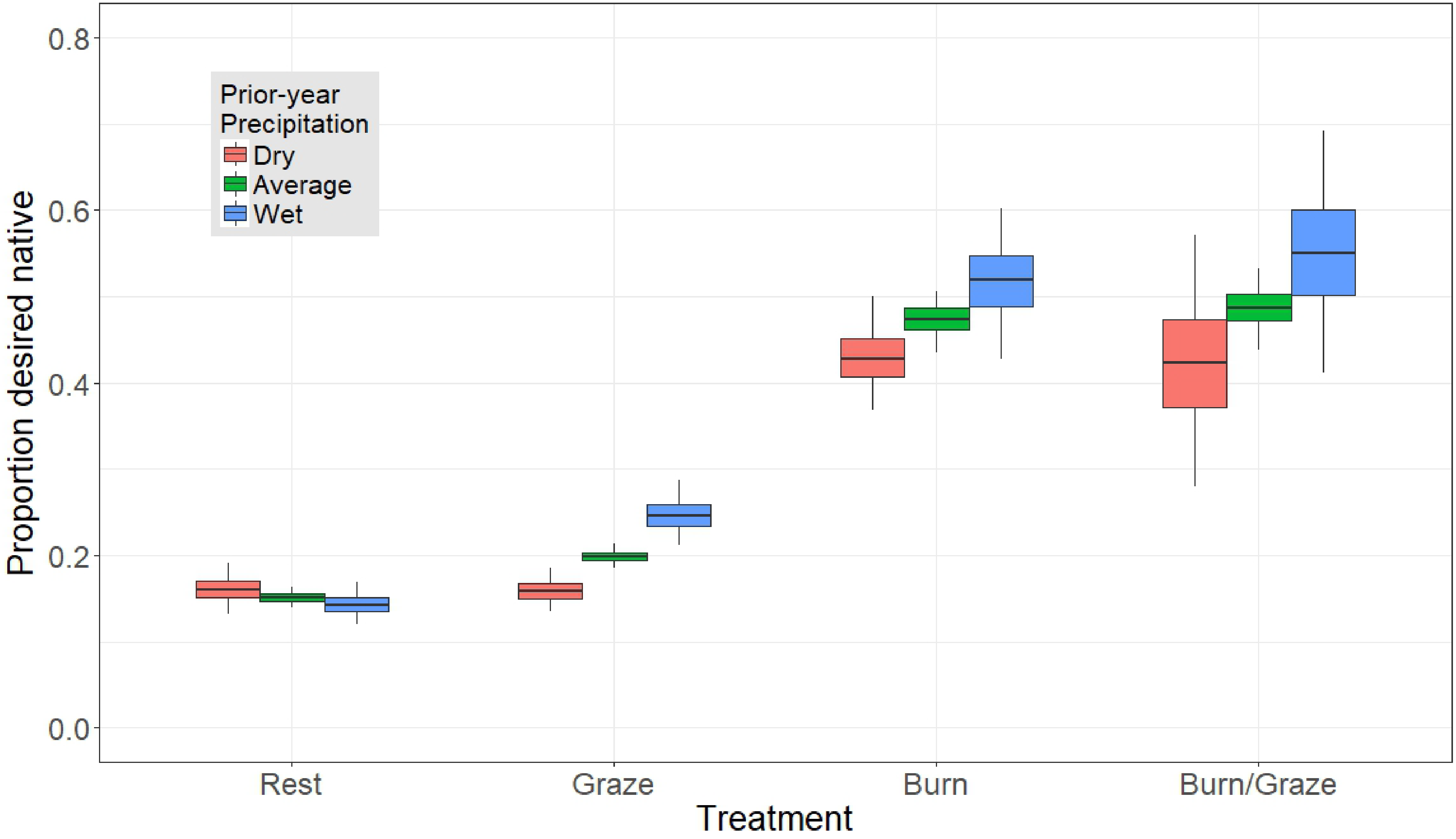


Figure 2.3. Estimated response (mean proportion of native vegetative cover) to combinations of treatment (Rest, Graze within Window, Burn within Window, Defoliate) and prior-year rainfall (mean – 2SD [Dry], mean [Average], mean + 2SD [Wet]) for tallgrass units.

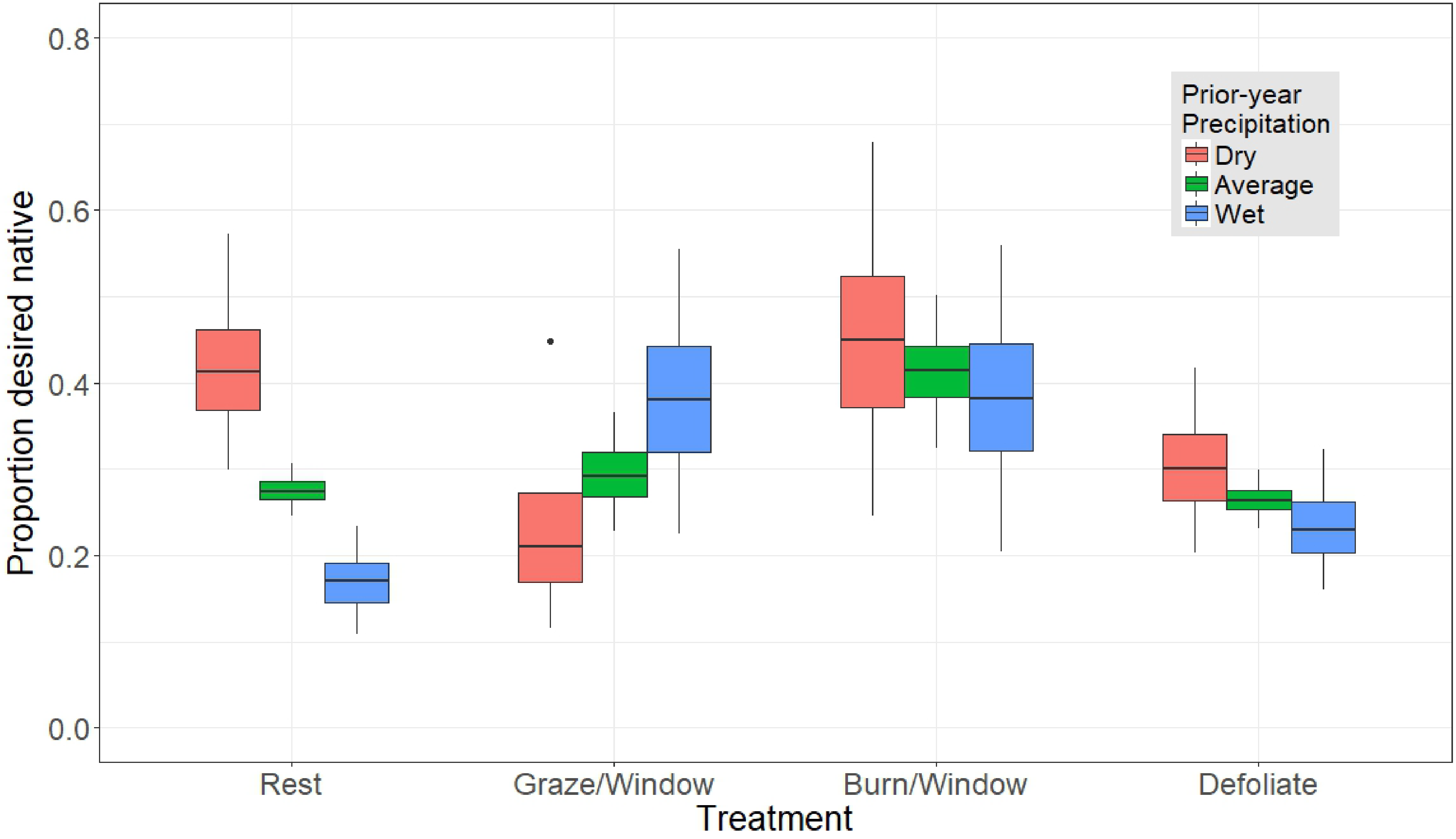


Figure 2.4. Unit-level mean response (mean proportion of native vegetation cover) at combinations of long-term precipitation and temperature climate variables. At drier sites, response diminishes rapidly with increasing temperature, whereas at wetter sites, little variation with temperature is observed.

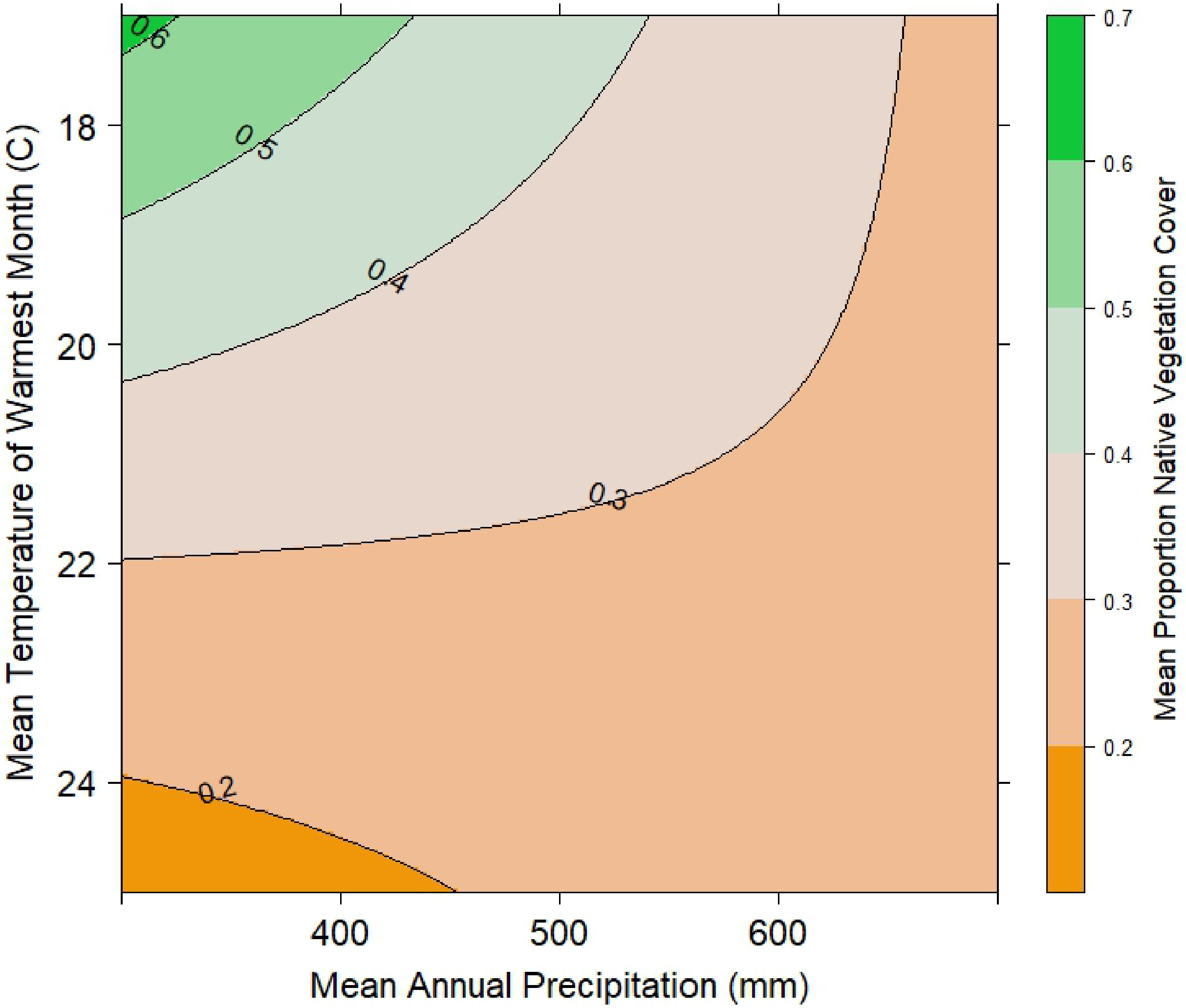


Figure 2.5. Annual means and trend in native vegetation cover proportion across the NPAM project, after adjustment for other model parameters (climate, prior-year precipitation, treatment, and prior-year native prairie percentage).

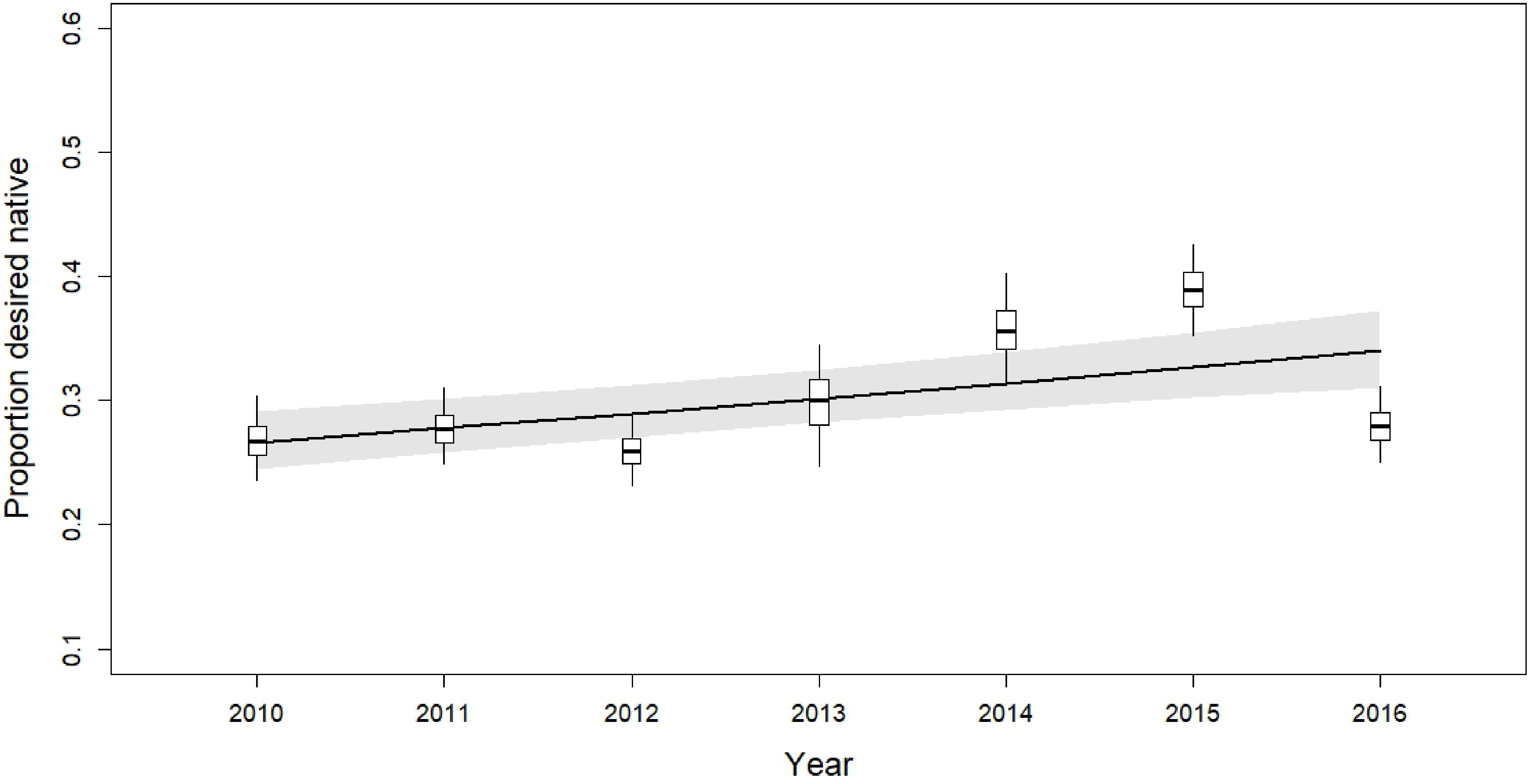
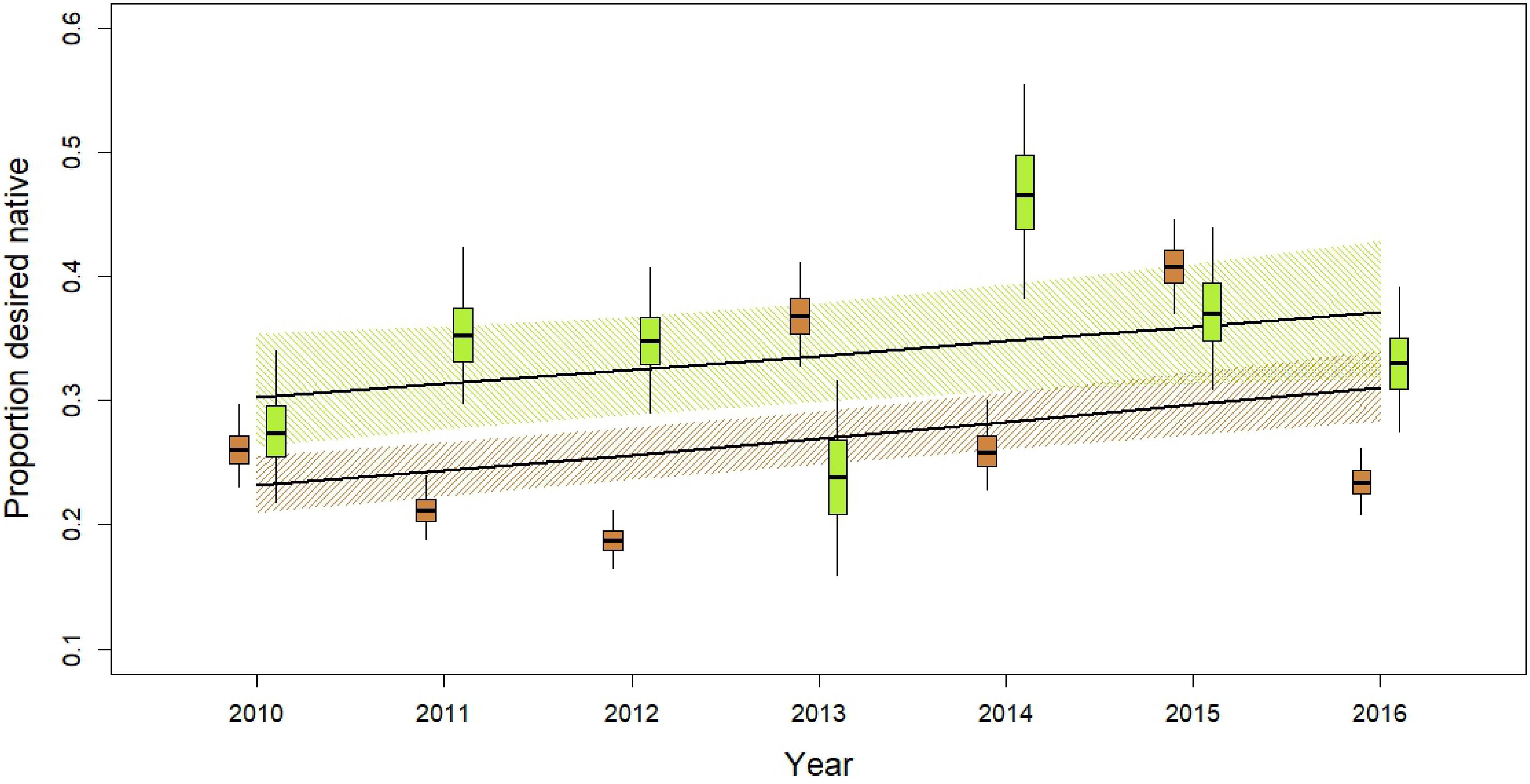


Figure 2.6. Annual means and trend, for mixed grass (brown) and tallgrass (green) types, in native vegetation cover proportion across the NPAM project, after adjustment for other model parameters (climate, prior-year precipitation, treatment, and prior-year native prairie percentage).



Appendix A: Methods used to collect monthly, accumulated precipitation data at the level of the NPAM transect.

Source: NOAA National Weather Service Advanced Hydrologic Prediction Service
Website: http://water.weather.gov/precip/download.php
Spatial Reference: Point shapefile for transect start and end points used to extract values

- Geographic Coordinate System: GCS_North_American_1983
- Datum: D_North_American_1983

Data:

- Spatial extent: United States
- Spatial resolution: 4x4 km
- Temporal scale: Monthly
- Temporal resolution: Daily
- Metric: Inches
- Format: Shapefile of grid points
- General Information: The precipitation data are quality-controlled, multi-sensor (radar and rain gauge) precipitation estimates obtained from National Weather Service (NWS) River Forecast Centers (RFCs). The original data are in XMRG format and projected in the Hydrologic Rainfall Analysis Project (HRAP) grid coordinate system, a polar stereographic projection true at 60°N / 105°W. Software reads each participating RFC's XMRG file and grabs the 24-hour precipitation estimate for each HRAP grid cell.

Link to Metadata: http://water.weather.gov/precip/metadata/DailyObservedPrecip.xml

Variables

- 108 variables, one for each month from September 2007 through August 2016. The measure is the accumulated precipitation, in inches, over the month.

Original Files

- GIS\Shapefiles\NPAM_Transects_2009to2016.shp
- GIS\Shapefiles\PPR_Buffer_1km.shp
- GIS\Shapefiles\PPR_Buffer16km.shp
- GIS\GIS_Data\state_bounds\state_bounds.shp

ArcGIS Project

- GIS\Projects\Precipitation.mxd

Files Created

- GIS\Shapefiles\NPAM_Transects_2009to2016_Precip.shp
- 108 clipped precipitation point shapefiles, one per month and year from September 2007 through August 2016
    - GIS\GIS_Data\Precipitation\NWS_USA_Precip_Sep2007\Precip_Sep07.shp

    ⋮
    - GIS\GIS_Data\Precipitation\NWS_USA_Precip_Aug2016\Precip_Aug2016.shp
- 108 clipped precipitation raster coverages, one per month and year from September 2007 through August 2016
    - GIS\Raster\Precipitation\sep07

    ⋮
    - GIS\Raster\Precipitation\aug16

Methods

The precipitation data were obtained as 108 separate point shapefiles, one for each month from September 2007 through August 2016. Each of these 108 files were downloaded, separately, from the following website, http://water.weather.gov/precip/download.php. For each download we completed the following steps/selections on the website: (1) Choose Format "Shapefile"; (2) Select the "Year" and "Month" from the dropdown menus; (3) Select Download "Monthly"; (4) Select Product "Observed"; and (5) Press the "Download" button. Files were downloaded as "tar.gz" files; for example, the file for the month of September 2007 was "nws_precip_month2date_observed_shape_20071001.tar.gz". The downloaded files, once extracted included four file types – .shp, .shx, .dbf, and .prj – all preceded by the file name "nws_precip_20071001" (again, this is the example for September 2007). The file for each month was approximately 0.5 to 7.0 MB in size. We used the program "7-Zip" to extract the usable files from the tar.gz file.

We added each of the 108 point shapefiles to a project named, "Precipitation.mxd". Because the shapefiles covered the area of the entire United States, we clipped each shapefile to the boundary of the Prairie Pothole Region (PPR), buffered by 16 km. The clip was completed using Geoprocessing>Clip with the following selections: Input feature = original point shapefile, Clip feature = PPR_Buffer16km.shp, Output feature = "Precip_[3-letter month and 2-digit year, no spaces].shp", and XY Tolerance = 1000 meters. This step was completed 108 times, once for each month.

Next we created a raster coverage from the point shapefile of each month via interpolation. We used ArcToolbox>SpatialAnalyst>Interpolation>Spline with the following selections: Input point features = clipped point shapefile, Z value field = Globvalue, Output raster = default, Output cell size = default, Spline type = regularized, Weight = 0.5, and Number of points = 12. We completed this step 108 times, once for each month.

Next we clipped the raster coverage to the PPR, buffered by 1 km. The clip was completed using ArcToolbox>Data Management Tools> Raster>Raster Processing>Clip with the following selections: Input Raster = name of raster file created from previous step, Output extent = PPR_Buffer_1km.shp, Check Box for "Use Input Features for Clipping Geometry", Output raster dataset = [3-letter month and 2-digit year, no spaces], NoData value = [none]. We completed this step 108 times, once for the raster coverage of each month created in the previous step.

We extracted the precipitation values from the clipped raster coverage for each month and year to the point shapefiles for all of the transects. We did this using ArcToolbox>SpatialAnalyst>Extraction> Extract Multi Values to Points. This tool allows the user to input multiple raster coverages per single shapefile; thus we were able to do this extraction step in a single action, once for the single transect point shapefile, while inputting all 108 raster coverages. The original transect point shapefile is noted under "Original Files" above, whereas the resulting transect point shapefile is noted under "Files Created". The shapefile created is given the same name with "_Precip" tacked onto the end. This shapefile contains the monthly accumulated precipitation, in inches, for the 108 months from September 2007 through August 2016.

Lastly, we imported the .dbf file of each of the resulting transect shapefiles to Excel to create the resulting data file, "Transects_Precipitation(sep07-aug16).xls.

Resulting Data File:

- Transects_Precipitation(sep07-aug16).xls
    - Complete data for 2790 NPAM transects.

- Fields include
  - ObjectID
  - Station_Name
  - Unit_Name
  - TransectID
  - Trasect_Name
  - X1, Northing UTM for starting point of transect
  - Y1, Easting UTM for starting point of transect
  - Lat, Latitude for starting point of transect
  - Long, Longitude for starting point of transect
  - _2009, management year 2009; 0=transect not active, 1=transect active and monitored, -1=transect active but not monitored
  - _2010, management year 2010; same as above
  - _2011, management year 2011; same as above
  - _2012, management year 2012; same as above
  - _2013, management year 2013; same as above
  - _2014, management year 2014; same as above
  - _2015, management year 2015; same as above
  - _2016, management year 2016; same as above
  - Azimuth
  - Slope
  - Aspect
  - 108 columns, one for each month from September 2007 through August 2016, noted by a three-letter month and the last two digits of the year (e.g., sep07 and aug16).

Appendix B: Methods used to collect long-term precipitation and temperature (LTPT) data at the level of the NPAM transect. LTPT data include mean annual precipitation (MAP), growing season precipitation (GSP), mean maximum temperature in the warmest month of the year (MAXT), and mean temperature in the warmest month of the year (MTWM).

Source: Rocky Mountain Research Station: Moscow Forestry Sciences Laboratory (USFS)

- Research on Forest Climate Change: Potential Effects of Global Warming on Forests and Plant Climate Relationships in Western North America and Mexico
- Website: http://charcoal.cnre.vt.edu/climate/current/

Spatial Reference: Point shapefile for transect start and end points used to extract values

- Geographic Coordinate System: GCS_North_American_1983
- Datum: D_North_American_1983

Precipitation data:

- Metric: Millimeter (mm)
- Temporal extent: 1961-1990
- Spatial extent: North America
- Spatial resolution: 1 Kilometer
- Format: ASCII Grid text file

Temperature Data:

- Metric: Degrees Celsius (C) to the tenth; divide the number by '10' to get degrees Celsius to the decimal point.
- Temporal scale: 1961-1990
- Spatial resolution: 1 Kilometer resolution
- Format: ASCII Grid text file

Links to Metadata from the Research Station:

- http://charcoal.cnre.vt.edu/climate/current/allNA/derivedGrids/map.metadata.xml
- http://charcoal.cnre.vt.edu/climate/current/allNA/derivedGrids/gsp.metadata.xml
- http://charcoal.cnre.vt.edu/climate/current/allNA/derivedGrids/mmax_tenths.metadata.xml
- http://charcoal.cnre.vt.edu/climate/current/allNA/derivedGrids/mtwm_tenths.metadata.xml

Variables

- Precipitation
  - Mean Annual Precipitation (MAP)
  - Growing Season Precipitation (GSP)
- Temperature
  - Mean Maximum temperature in the warmest month of the year (MAXT)
  - Mean temperature in the warmest month of the year (MTWM)

Original Files

- GIS\GIS_Data\PRecip&Temp_1960to1991\gsp\gsp.txt
- GIS\GIS_Data\PRecip&Temp_1960to1991\map\map.txt
- GIS\GIS_Data\PRecip&Temp_1960to1991\mmax_tenths\mmax_tenths.txt
- GIS\GIS_Data\PRecip&Temp_1960to1991\mtwm_tenths\mtwm_tenths.txt
- GIS\Shapefiles\NPAM_Transects_2009to2016.shp
- GIS\Shapefiles\PPR_Buffer.shp
- GIS\GIS_Data\state_bounds\state_bounds.shp

ArcGIS Project

- GIS\Projects\LongTermPrecip&Temp.mxd

Files Created

- GIS\Raster\LongtermPrecip&Temp\gsp_ppr
- GIS\Raster\LongtermPrecip&Temp\map_ppr
- GIS\Raster\LongtermPrecip&Temp\mtwm_ppr
- GIS\Raster\LongtermPrecip&Temp\mmax_ppr
- GIS\Shapefiles\NPAM_Transects_2009to2016_LTPT.shp

Methods

The long-term climate data were obtained as a four separate text files in ASCII grid format, which describe the data across all of North America. These four files are listed under "Original Files" above. Within ArcMap, we used ArcToolbox>Conversion Tools>To Raster>ASCII to Raster to convert the four ASCII files into four raster coverages: one for each data type. We then clipped each of the four raster coverages using a shapefile of the Prairie Pothole Region (PPR), buffered by 1 kilometer. This resulted in four raster coverages – gsp_ppr, map_ppr, mtwm_ppr, and mmax_ppr – one for each data type for the spatial coverage of the PPR. The four raster files are listed above under "Files Created". The clip was completed using ArcToolbox>Data Management Tools> Raster>Raster Processing>Clip. We then extracted the data from each of the four raster coverages for each of the transect start points (i.e., X1, Y1). We used ArcToolbox>Spatial Analyst>Extract>Extract Multi Values to Points for this step. We completed this step for the transect shapefile listed above under "Original Files". Using the 'Extract Multi Values to Points' tools allowed us to extract the data from all four raster coverages to a single transect shapefile in a single step. The resulting shapefile is listed above under "Files Created". There was no reason for selecting the transect start point over the transect end point to do the data extraction; however, given these are long-term data, collected at a spatial resolution of 1km, and given the transect start and end points are only 25m apart, we determined the choice was inconsequential.

Resulting Data File:

- Transects_LTPrecip&Temp(Update2016).xlsx
    - Complete data for 2790 NPAM transects.
    - Fields include
        - ObjectID
        - Station_Name
        - Unit_Name
        - TransectID
        - Trasect_Name
        - X1, Northing UTM for starting point of transect
        - Y1, Easting UTM for starting point of transect
        - Lat, Latitude for starting point of transect
        - Long, Longitude for starting point of transect
        - _2009, management year 2009; 0=transect not active, 1=transect active and monitored, -1=transect active but not monitored
        - _2010, management year 2010; same as above
        - _2011, management year 2011; same as above
        - _2012, management year 2012; same as above
        - _2013, management year 2013; same as above
        - _2014, management year 2014; same as above
        - _2015, management year 2015; same as above
        - _2016, management year 2016; same as above

- Azimuth
- Slope
- Aspect
- GSP
- MAP
- MTWM
- MAX

Appendix C: Methods used to collect and derive Ecological Site Description data at the level of the NPAM transect.

Source: USDA NRCS Web Soil Survey
Website: http://websoilsurvey.sc.egov.usda.gov/App/HomePage.htm
Spatial Reference: Point shapefile for transect start and end points used to extract values

- Projected Coordinate System: NAD_1983_UTM
- UTM Zones: 12N, 13N, 14N, and 15N
- Projection: Transverse Mercator
- Unit: Meter
- Geographic Coordinate System: GCS_North_American_1983
- Datum: D_North_American_1983

Shapefile Data:

  - Spatial extent: County
  - Spatial scale: 1:12,000 – 1:20,000
  - Format: Polygon Shapefile
  - Geographic Coordinate System: Geographic WGS84
  - Metadata are present as text files within each downloaded soil layer

Tabular Data:

- Downloaded along with the shapefile data, the tabular data contain Access databases with detailed data for each soil layer.

Variables of Shapefile Data:

- AreaSymbol – This is the field name of the soil layer
- MuSym – This is the field name of the map unit code within the soil layer

Variables of Tabular Data:

- SoilLayer – Code of the soil layer
- MUSYM – Code of the map unit within the soil layer
- MUKEY – Key for the map unit
- MUNAME – Name of the map unit
- CompPct – Percent of the component that comprises the map unit
- CompName – Name of the component
- EcoSiteID – ID of the ecological site
- EcoSiteName – Name of the ecological site

Original Files
Shapefile Data (includes 58 soil shapefiles)

- GIS\Shapefiles\TransectPoints_BowdoinWMD.shp
- GIS\Shapefiles\TransectPoints_UTM12N.shp
- GIS\Shapefiles\TransectPoints_UTM13N.shp
- GIS\Shapefiles\TransectPoints_UTM14N.shp
- GIS\Shapefiles\TransectPoints_UTM15N.shp
- GIS\Shapefiles\Transects_CenterPoints_Missing_2014.shp
- GIS\GIS_Data\state_bounds\state_bounds.shp
- GIS\GIS_Data\SSURGO\MT\soil_mt091\soil_mt091\spatial\soilmu_a_mt091.shp

  ⋮
  GIS\GIS_Data\SSURGO\MT\soil_mt661\soil_mt661\spatial\soilmu_a_mt661.shp
- GIS\GIS_Data\SSURGO\ND\soil_nd005\soil_nd005\spatial\soilmu_a_nd005.shp
  ⋮
  GIS\GIS_Data\SSURGO\ND\soil_nd105\soil_nd105\spatial\soilmu_a_nd105.shp
- GIS\GIS_Data\SSURGO\SD\soil_sd003\soil_sd003\spatial\soilmu_a_sd003.shp
  ⋮
  GIS\GIS_Data\SSURGO\SD\soil_sd603\soil_sd603\spatial\soilmu_a_sd603.shp
- GIS\GIS_Data\SSURGO\MN\soil_mn011\soil_mn011\spatial\soilmu_a_mn011.shp
  ⋮
  GIS\GIS_Data\SSURGO\MN\soil_mn087\soil_mn087\spatial\soilmu_a_mn087.shp

Tabular Data (includes 58 Access files)

- GIS\GIS_Data\SSURGO\MT\soil_mt091\soil_mt091\soildb_MT_2002\soildb_MT_2002.mdb
  ⋮
  GIS\GIS_Data\SSURGO\MT\soil_mt661\soil_mt661\soildb_MT_2002\soildb_MT_2002.mdb
- GIS\GIS_Data\SSURGO\ND\soil_nd005\soil_nd005\soildb_US_2002\soildb_US_2002.mdb
  ⋮
  GIS\GIS_Data\SSURGO\ND\soil_nd105\soil_nd105\soildb_US_2002\soildb_US_2002.mdb
- GIS\GIS_Data\SSURGO\SD\soil_sd003\soil_sd003\soildb_SD_2002\soildb_SD_2002.mdb
  ⋮
  GIS\GIS_Data\SSURGO\SD\soil_sd603\soil_sd603\soildb_SD_2002\soildb_SD_2002.mdb
- GIS\GIS_Data\SSURGO\MN\soil_mn011\soil_mn011\soildb_MN_2003\soildb_MN_2003.mdb
  ⋮
  GIS\GIS_Data\SSURGO\MN\soil_mn087\soil_mn087\soildb_MN_2003\soildb_MN_2003.mdb

ArcGIS Project

- GIS\Projects\Soils.mxd

Files Created

Shapefile Data

- GIS\GIS_Data\SSURGO\Soils_UTM12N_Merge.shp
- GIS\GIS_Data\SSURGO\Soils_UTM13N_Merge.shp
- GIS\GIS_Data\SSURGO\Soils_UTM14N_Merge.shp
- GIS\GIS_Data\SSURGO\Soils_UTM15N_Merge.shp
- GIS\Shapefiles\TransectPoints_BowdoinWMD_Soil.shp
- GIS\Shapefiles\TransectPoints_UTM12N_Soil.shp
- GIS\Shapefiles\TransectPoints_UTM13N_Soil.shp
- GIS\Shapefiles\TransectPoints_UTM14N_Soil.shp
- GIS\Shapefiles\TransectPoints_UTM15N_Soil.shp
- GIS\Shapefiles\Transects_CenterPoints_Missing_2014_Soil.shp

Tabular Data

- Abiotics\ESDs&Soils\Transect_ESDs\Transects_ALL(minusMN)_Soil.xlsx
- Abiotics\ESDs&Soils\Transect_ESDs\Transects_MapUnitComponents_ALL.xlsx
- Abiotics\ESDs&Soils\ESD_Classifications.xlxs
- Abiotics\ESDs&Soils\ESDs.accdb
- Abiotics\ESDs&Soils\Transect_ESDs\ Transects_ESDClassified_I&M(sum)_final.xlsx

Final Data File:

- Transcects_ESDClassified_I&M(sum)_final.xls
    - Complete data for 2437 NPAM transects.
    - Fields include
        - Unit_Name
        - TransectID
        - Trasect_Name
        - SumInvasive
        - SumMaintain
        - SumNA
        - Sum

Methods

Shapefile Data – We started by identifying the counties where we have NPAM management units. We created a new ArcGIS project and named it Soils.mxd. We added the transect point shapefiles listed under "Original Files" above. We downloaded an ArcGIS polygon layer of USA counties (downloaded from http://www.arcgis.com/home/item.html?id=a00d6b6149b34ed3b833e10fb72ef47b) and added this layer to the project. We used an overlay between our transect point shapefiles and the USA counties shapefile to identify the counties where we need soil data. We then downloaded the soil information for each county where we have NPAM transects. To obtain these soil data, we went to the USDS NRCS Web Soil Survey website located at the following link: http://websoilsurvey.sc.egov.usda.gov/App/HomePage.htm. From this site, we clicked on "Start WSS", which takes the user to http://websoilsurvey.sc.egov.usda.gov/App/WebSoilSurvey.aspx From here, we selected the "Download Soils Data" tab. On this tab, we selected SSURGO data. Then we selected the state and the county of interest. We selected "Area Symbol" as the 'Sort by' option. Then we used the link to download the specified zip file that includes both tabular and spatial data. After unzipping the downloaded file we added the soil shapefile to the ArcGIS project. We completed this task 58 times, once for each county soil download (Table C1). We then merged soil shapefiles according to the UTM zone within which they were located (i.e., 12N, 13N, 14N, or 15N; Table C1). The last step we completed in ArcGIS, was to intersect each of the six transect point shapefiles with the appropriate merged soil file. This was done by Geoprocessing>Intersect; with the following selections as one example: Input features = TransectPoints_UTM12N.shp, Soils_UTM12N_Merge.shp; Output feature class = TransectPoints_UTM12N_Soil.shp; JoinAttributes = ALL; XY tolerance = null. This intersection adds the attributes of the soil shapefile to the attributes of the transect point shapefile, according to their spatial overlap.

Tabular Data – We opened up the .dbf file of each transect point soil shapefile (e.g., TransectPoints_UTM12N_Soil.dbf) in Excel and saved it as an .xlsx file. These files contained the following fields: ObjectID, Unit_Name, TransectID, Transect_Name, UTM_Zone, X1, Y1, X2, Y2, Azimuth, Slope, Aspect, SoilLayer, MUSYM, and MUKEY. The field "SoilLayer" contains the "Area Symbol" of the county soil layer, for example, MT091. MUSYM is the map unit symbol. MUKEY is the map unit key. A "map unit" indicates the individual soil polygons within each county soil layer. We then appended these six files together, excluding the transects in MN, to create a single Excel file, "Transects_ALL(minusMN)_Soil.xlsx". We excluded MN transects, because MN soil layers have not yet been categorized into Ecological Site Descriptions (ESDs) by the NRCS. This Excel file contains soil information for 2,262 NPAM transects. From this Excel file we created another file, "Transects_MapUnitComponents_ALL.xlxs". Within this file, we created a spreadsheet, "MapUnits", which contained a list of the unique (i.e., no duplicates) soil layer, MUSYM, and MUKEY units that exist among the 2,262 NPAM transects; there were 344 unique soil combinations represented (Table C2). The number of map units within each soil layer varied considerably; some soil layers contained as few as a single map unit (e.g., ND035), whereas others contained as many as 28 map units (ND093).

Using the list shown in Table C2, we went to the tabular soil data that had been downloaded along with the shapefile data from the Web Soil Survey site. Each soil layer (e.g., MT091) has an Access file containing all of its tabular data. One at a time, we opened each of the 52 Access files and completed the following steps. We created a new query and named it "MapUnit_Components". To this query we added three tables: "mapunit", "component", and "coecoclass". We linked the "mapunit" table to the "component" table via the "mukey" field. We linked the "component" table to the "coecoclass" table via the "cokey" field. We then added the following fields to the query table: "musym", "mukey", and "muname" from the "mapunit" table; "comppct_r" and "compname" from the "component" table; and "ecoclassid" and "ecoclassname" from the "coecoclass" table. Within the "Criteria" cell for the "musym" field, we typed in the "MUSYM" codes for the soil layer listed in Table C2. These codes were entered in quotes, exactly as listed, and combined with an 'Or' statement. For example, the criteria for the query we developed within the MT091 Access file read, "BcD" Or "DoB" Or "LhB" Or "Mz" Or "WaE" Or "WmB" Or "WzB" Or "ZaE" Or "ZwE". The specified criteria ensured that we only extracted the tabular data that we needed from the full county soil layer. We then ran the query and exported the resulting table as an Excel file. After creating, running, and exporting these data for all 52 soil layers, we appended them into a single spreadsheet, "MapUnits_Components_ALL" within the aforementioned file, "Transects_MapUnitComponents_ALL.xlxs". This spreadsheet contains 2,329 records and includes the following fields: SoilLayer, MUSYM, MUKEY, MUNAME, CompPct, CompName, EcoSiteID, and EcoSiteName. "CompPct" is the percent of the map unit that is composed of each component. Though the component percentages typically sum to 100, in some cases there are non-sites within the map unit that do not have associated components; in these cases, the sum will be less than 100. "CompName" is the name of each component of which the map unit is comprised; for example, five components – Blanchard, Dooley, Lihen, Zahill, and Parshall – make up the BcD map unit of the MT091 soil layer. "EcoSiteID" is an identification code for ecological sites that is based on the soil layer, map unit, and component. For example, the EcoSiteID "R053AE076MT" is associated with the MT091 soil layer, BcD map unit, and Blanchard component. The first letter of the EcoSiteID is "site type", with is "R" for "Rangeland". The next combination of four numbers and letters, here shown as "053A", denotes the Major Land Resource Area (MLRA). The last two letters of the EcoSite ID indicate the state. The meaning of the four letters and numbers before the state abbreviation is unknown to us. "EcoSiteName" is the name for the specific EcoSiteID. For example, the name for EcoSiteID R053AE076MT is Sands (Sa) 10-14" p.z.

Shapefiles of ecological sites exist and are available for download; however, the ecological sites are assigned to each map unit based on the ecological site with the highest cumulative percent within the map unit. For example, map unit MT091_BcD contains 92% Sands, 6% Sandy, and 2% Clayey-Steep (Table C3). The ecological site assigned to this map unit, based on the automated dominance method, would be Sands. This is probably fine in a case like MT091_BcD, which is highly dominated by a single ecological site; however, this is often not the case. For example, ecological sites within ND093_G147C are more broadly distributed, containing 41% Thin Loamy, 23% Loamy, 22% Loamy Overflow, 3% Clayey, 3% Limy Subirrigated, 3% Sandy, 3% Very Shallow, and 2% Wet Meadow (Table C3). Under the automated dominance method, this map unit would be classified with a Thin Loamy ecological site based on the 41%; however, the remaining 59% of the map unit is composed of seven other ecological sites. Given this particular map unit measures approximately 400 x 350 meters (which is a comparatively small map unit size), the transect located within this map unit measures 1 x 25 meters, and we do not know spatially where the eight different ecological sites exist within the map unit relative to the spatial location of the transect, we believed that using the dominance method to classify the overall ecological site of each map unit and thus assign that ecological site to the transect would too often mischaracterize the transect, as well as lead to an undesirable loss of available detail. Instead, we used the method we describe in the next paragraph to classify our NPAM transects.

Rather than assign map units to a single ecological site type, with the help of range ecologists David Pyke (USGS) and Jeff Printz (NRCS), we classified each of the 49 ecological sites that exist within the locations of the NPAM transects into two categories – maintain or invade. Ecological sites that are not favorable for the invasive cool-season grasses smooth brome and Kentucky bluegrass and are suitable for native grasses and forbs were classified as “maintain”. These sites, to which native species are better adapted, tend to be harsher, with sandier soils and steeper slopes that retain less moisture. Ecological sites that favor invasive grasses, allowing them to out-compete native grasses and forbs, were classified as “invade”. The complete list of ecological sites overlapped by the NPAM transects, and the classification of these sites as “maintain” or “invade” is provided in Table C4. A small number of map units contained components that were classified as “non-sites”; we could not classify these ecological non-sites as either invade or maintain and thus classified them as NA. We emphasize that the binary assignment of the ecological sites was based solely on expert interpretation of ecological site descriptions and not on site assessments of vegetation composition.

After classification of ecological sites into more likely to maintain natives or be invaded by smooth brome and Kentucky bluegrass, the next step was to assign these classifications to the ecological sites composing each map unit where a transect was located and then sum the percent composition of each type (i.e., maintain or invade) for the map unit. Our methods for completing this step are described herein. We created a new Access database, ESD.accdb. To this database, we imported three Excel files that we had created and described above: (1) Transects_ALL(minusMN)_Soil.xlsx; (2) Transects_MapUnitComponents_ALL.xlsx; and (3) ESD_Classifications.xlxs. We then created a new query, “qry_Component_ESD_Classificaiton”. To this query we added the three imported tables above. We linked the Transects Soil table to the Map Unit Components table via the MUKEY field. We linked the Map Unit Components table to the ESD Classification table via the EcoSiteName field. We then added the following fields to the query: Unit_Name, TransectID, Transect_Name, SoilLayer, MUSYM, and MUKEY from the Transect Soil table; MUNAME, CompName, CompPct, and EcoSiteName from the Map Unit Components table; and ESD and Classification from the ESD Classification table. We ran this query and created the table “tbl_Component_ESD_Classifcation”; we added this table to the Access database. We then created a second query, “qry_SumInvasive”. To this query we added the table “tbl_Component_ESD_Classification”. We than added the following fields to the query: Unit_Name, TransectID, Transect_Name, SoilLayer, MUSYM, MUNAME, CompPct, and Classication. In the “Total” row, we selected “Sum” for the CompPct field and “Group By” for all other fields. In the “Criteria” row, we entered “I” for the Classification field. We ran this query, created a table, “SumInvasive”, and added this table to the Access database. We then created a third query, “qry_SumMaintain”. This query was exactly like the previous query, with the exception that we changed the criteria for the Classification field to “M”. As before, we ran this query, created a table, “SumMaintain”, and added the table to the Access database. We then created a fourth query, “qry_SumOther”. This query was the same as the previous two, with the exception that we changed the criterial for the Classification field to “NA”. As before, we ran this query, created a table, “SumOther”, and added the table to the Access database. We ran a fifth and final query, “qry_SumI&M&NA”. For this query, we added four tables: Transects Soil, SumInvasive, SumMaintain, and SumOther. We linked the Transects Soil table to each of the other three tables via the field “Transect_Name”. To the query, we added the following fields: Unit_Name, TransectID, and Transect_Name from the Transects Soil table; SumOfCompPct from the SumInvasive table, SumOfCompPct from the SumMaintain table, and SumOfCompPct from the SumOther table. We ran this query and exported the resulting table to get our final data "Transects_ESDClassified_I&M(sum)_final.xls".

The Transects_ESDClassified_I&M(sum)_final.xls data contains seven fields – Unit_Name, TransectID, Trasect_Name, SumInvasive, SumMaintain, SumNA, Sum – and complete data for 2437 NPAM transects. SumInvasive is the percent of the map unit, within which the transect is located, that is composed of ecological site components that we classified as "invade". SumMaintain is the percent of the map unit, within which the transect is located, that is composed of ecological sites that we classified as "maintain"; the ecological sites came from the "EcoSiteName" field and the percentages came from the "CompPct" field (e.g., Table C3). SumNA is the percent of the map unit that is composed of ecological site components that we did not classify as either invade or maintain. The "Sum" field is the sum of the SumInvade, SumMaintain, and SumNA fields. The components for the majority (2,114) of the transects sum to 100%. Of the 323 transects whose components sum to less than 100%, 315 of them sum to between 90-99%, four sum to 75%, and five sum to less than 20%.

Recall the example shown in Table C3 and discussed above. A transect that fell within soil layer MT091 and map unit BcD, under the dominance method, would have been classified as a Sands ESD because Sands is the dominant ecological site contained within the map unit (92%; Table C3). If we had used the dominance method to classify the map unit to a dominant ESD and then used the information in Table C4 to assign that ESD dichotomously as "invade" or "maintain", we would have ended up classifying the map unit as "maintain". Under the method we devised, however, we describe a transect that fell within soil layer MT091, map unit BcD, as containing 100% maintain ecological site components (or, conversely, 0% invade ecological site components; Table C5). MT091_BcD is a homogenous map unit; thus, there is not much difference between the end result of the dominance method and our method. More interesting is the second example, looking at the end result of the two method for a transect that fell within soil layer ND093, map unit G147C. Under the dominance method, this map unit would have been assigned a Thin Loamy ESD because Thin Loamy is the dominant ecological site contained within in the map unit (41%; Table C3). If we had then used the information in Table C4 to assign that ESD dichotomously as "invade" or "maintain", we would have ended up classifying the entire map unit as "invade". Under the method we devised, however, we describe a transect that fell within soil layer ND093, map unit G147C, as containing 51% invade ecological site components and 49% maintain ecological site components (Table C5). In this example, which is representative of many of the NPAM transects (Table C6), the map unit is composed of approximately 50:50 invade and maintain ecological site components. Given the mixture of components within a map unit, the small size of a transect relative to a map unit, and the unknown spatial location of the ecological site components within the map unit relative to the location of the transect, it is clear that the method we employed is superior to the dominance method for our needs.

Lastly, a note regarding our methods when a transect overlapped more than one map unit. The transect shapefiles we used contained points for both ends of the transect. Thus, we collected the map unit that intersected with the two points of each transect. In several cases, transects intersected more than one map unit. In these cases, we estimated the length of the transect that fell within each of its intersecting map units (we did this using the measuring tool in ArcGIS), classified the components of each map unit as "invade", "maintain", or "na", and then calculated a weighted sum of the components given the length of the transect that fell within each map unit. This was a rather time consuming approach. While it likely resulted in different measurements for the transect, a more efficient and likely acceptable shortcut would have been to simply use a single midpoint for each transect. We recommend using a midpoint method in future applications.

Table C1: Soil layers downloaded for NPAM transects; there are 58. The state and county were identified and entered in the dropdown fields on the Web Soil Survey in order to download the necessary soil file. The UTM zone indicates which soil files we merged, by UTM zone, to intersect the soil and transect shapefiles in fewer steps (i.e., reduced to six as opposed to 58 actions).

State	County	Soil Layer	UTM Zone
MT	Sheridan	MT091	13N
MT	Toole	MT101	12N
MT	Blaine	MT608	12N
MT	Phillips	MT641	12N
MT	Roosevelt	MT661	13N
ND	Benson	ND005	14N
ND	Bouttineau	ND009	14N
ND	Burke	ND013	13N
ND	Burleigh	ND015	14N
ND	Dickey	ND021	14N
ND	Eddy	ND027	14N
ND	Foster	ND031	14N
ND	Grand Forks	ND035	14N
ND	Griggs	ND039	14N
ND	Kidder	ND043	14N
ND	LaMoure	ND045	14N
ND	Logan	ND047	14N
ND	McHenry	ND049	14N
ND	McIntosh	ND051	14N
ND	McLean	ND055	14N
ND	Mountrail	ND061	13N
ND	Renville	ND075	14N
ND	Richland	ND077	14N
ND	Sargent	ND081	14N
ND	Sheridan	ND083	14N
ND	Stutsman	ND093	14N
ND	Towner	ND095	14N
ND	Ward	ND101	14N
ND	Wells	ND103	14N
ND	Williams	ND105	13N
SD	Aurora	SD003	14N
SD	Bon Homme	SD009	14N
SD	Brown	SD013	14N
SD	Campbell	SD021	14N
SD	Charles Mix	SD023	14N
SD	Clark	SD025	14N
SD	Codington	SD029	14N
SD	Deuel	SD039	14N
SD	Edmunds	SD045	14N
SD	Hamlin	SD057	14N
SD	Hand	SD059	14N
SD	Hyde	SD069	14N
SD	Jerauld	SD073	14N
SD	Lake	SD079	14N
SD	McPherson	SD089	14N
SD	Marshall	SD091	14N
SD	Miner	SD097	14N
SD	Minnehaha	SD099	14N
SD	Roberts	SD109	14N
SD	Spink	SD115	14N
SD	Buffalo	SD603	14N
MN	Big Stone	MN011	14N
MN	Clay	MN027	14N
MN	Cottonwood	MN033	15N
MN	Lac qui Parle	MN073	14N
MN	Manhomen	MN087	15N

Table C2: Soil map units represented in the transect point soil shapefiles across all NPAM transects, with exception of those located in MN. There are 344 unique soil layer map units.

Soil Layer	MUSYM	MUKEY
MT091	BcD	348835
MT091	DoB	348843
MT091	LhB	348857
MT091	Mz	348862
MT091	WaE	348876
MT091	WmB	348878
MT091	WzB	348880
MT091	ZaE	348883
MT091	ZwE	348884
MT101	37B	345149
MT101	402A	345161
MT101	503B	345204
MT101	698D	345250
MT101	698E	345251
MT608	39	346450
MT608	68	346517
MT608	95	346573
MT608	124	346275
MT608	133	346297
MT641	392B	347702
MT641	564C	347754
MT641	567C	347757
MT661	42	348247
ND005	F100A	2587778
ND005	F118A	2587647
ND005	F12A	2587644
ND005	F143A	2587722
ND005	F143B	2587610
ND005	F143C	2587645
ND005	F143D	2587731
ND005	F143F	2587619
ND005	F144B	2587662
ND005	F427C	2587636
ND005	F481A	2587688
ND005	F680F	2587607
ND009	F143A	2586020
ND009	F143B	2585888
ND009	F147C	2585874
ND009	F147D	2586028
ND009	F148D	2585884
ND009	F159A	2585966
ND009	F996	2585942
ND013	882	341207
ND013	2176	341117
ND013	2177	341118
ND013	2184	341125

Soil Layer	MUSYM	MUKEY
ND013	C132B	2564059
ND013	C135D	2596889
ND013	C148C	2564070
ND013	C155F	2564081
ND013	C165F	2564058
ND013	C3A	2564056
ND013	C5A	2564067
ND013	C75A	2564082
ND015	C132C	2564099
ND015	C135D	2596897
ND015	C210B	2564134
ND015	C354B	2564138
ND015	C412A	2596898
ND015	C415C	2564142
ND015	C486B	2564152
ND015	C522B	2564131
ND015	C580A	2596821
ND015	C661A	2564189
ND015	C816A	2564124
ND015	C816B	2564107
ND015	C870E	2564182
ND015	C874B	2595638
ND021	C135C	2595690
ND021	C156F	2595434
ND021	C210B	2564214
ND027	G229B	2576231
ND027	G229C	2576232
ND027	G229D	2576233
ND027	G230B	2576240
ND027	G25A	2575404
ND031	G229F	2576407
ND031	G322A	2576431
ND031	G330A	2576434
ND031	G384B	2576444
ND031	G390A	2576446
ND031	G3A	2576377
ND031	G4A	2576378
ND031	G749A	2576476
ND031	G749B	2576477
ND031	G750C	2576478
ND031	G750D	2576479
ND035	I176A	2642574
ND039	G101A	2576576
ND039	G119A	2576580
ND039	G12A	2576565
ND039	G362A	2576624

Soil Layer	MUSYM	MUKEY
ND039	G4A	2576563
ND043	C156F	2596655
ND045	C135D	2596957
ND045	C210B	2566442
ND047	C132B	2566478
ND047	C135C	2566533
ND047	C135D	2596969
ND047	C156F	2566494
ND047	C270A	2566504
ND047	C800B	2566488
ND049	F101A	2585633
ND049	F25A	2585714
ND049	F273A	2585813
ND049	F273B	2585670
ND049	F329A	2585578
ND049	F366A	2585642
ND049	F381B	2585777
ND049	F384B	2585761
ND049	F815B	2585584
ND049	F819A	2585806
ND051	C132B	2566566
ND051	C135C	2566619
ND051	C135D	2596981
ND051	C156F	2566585
ND051	C162F	2566595
ND051	C870E	2595530
ND055	C132B	2566637
ND055	C132C	2566636
ND055	C135D	2596997
ND055	C165F	2566662
ND055	C850A	2566657
ND055	C897E	2596869
ND061	C135D	2597011
ND061	C154C	2566732
ND061	C165F	2566749
ND061	C272A	2566742
ND061	C3A	2595568
ND075	F142B	2585374
ND075	F143B	2585334
ND075	F143C	2585369
ND075	F147C	2585319
ND075	F147D	2585438
ND075	F147F	2585418
ND075	F157B	2585330
ND075	F3A	2585435
ND075	F562A	2585422

Soil Layer	MUSYM	MUKEY
ND075	F732B	2585458
ND077	I203A	2641023
ND077	I205A	2641168
ND077	I368A	2641057
ND081	G193A	2579567
ND081	G193C	2579525
ND081	G197C	2579492
ND081	G198B	2579555
ND081	G199A	2579478
ND081	G199B	2579507
ND081	G206A	2579486
ND081	I465A	2641246
ND081	I474A	2641248
ND081	I554A	2641259
ND083	C132C	2566782
ND083	C135D	2597022
ND083	C148C	2566781
ND083	C156F	2566795
ND083	C210B	2566796
ND083	C3A	2566813
ND083	C5A	2566807
ND093	C132C	2566829
ND093	C135D	2661634
ND093	C156F	2566846
ND093	C165F	2566844
ND093	C210B	2566847
ND093	C275A	2566855
ND093	C276A	2595587
ND093	C5A	2566866
ND093	C870E	2595591
ND093	C874C	2595583
ND093	C893C	2595590
ND093	C996	2661635
ND093	G101A	2581340
ND093	G112A	2581341
ND093	G135A	2581353
ND093	G143B	2581322
ND093	G143C	2581321
ND093	G147C	2581314
ND093	G147F	2581312
ND093	G250A	2581354
ND093	G269A	2581357
ND093	G272E	2581362
ND093	G426A	2581380
ND093	G586F	2581399
ND093	G596A	2581400
ND093	G680F	2581415
ND093	G731A	2581416
ND093	G759B	2581424

Soil Layer	MUSYM	MUKEY
ND095	F119A	2587066
ND095	F143B	2586198
ND095	F312B	2586204
ND095	F4A	2586196
ND095	F6A	2587051
ND101	C132C	2566885
ND101	C156F	2595596
ND101	C5A	2566929
ND101	C870E	2566937
ND101	C895C	2566945
ND103	C132C	2566950
ND103	C165F	2595707
ND103	C210B	2566957
ND105	1710	339024
ND105	C132C	2566992
ND105	C135D	2661664
ND105	C3A	2567041
ND105	C767C	2661667
ND105	C800B	2567006
ND105	C870E	2595614
SD003	HdA	354851
SD003	Hv	354859
SD003	Hw	354860
SD003	Wp	354871
SD009	CsB	418255
SD009	EnC	418269
SD009	EoD	418270
SD009	EpC	418271
SD009	HnB	418276
SD009	HpC	418278
SD009	Wo	418293
SD013	Fy	356293
SD013	HtB	356314
SD013	Un	356369
SD013	Wz	356389
SD021	36B	354707
SD021	53D	354721
SD021	53E	354722
SD021	57B	354726
SD021	57C	354727
SD021	59D	354729
SD021	61E	354730
SD023	Db	355013
SD023	EeB	355020
SD023	EeC	355021
SD025	BcC	417686
SD025	BrD	417697
SD025	MsA	417759
SD025	MtB	417761

Soil Layer	MUSYM	MUKEY
SD029	J111C	1684320
SD029	J115D	1684306
SD029	J117E	1684323
SD029	J140A	1585799
SD039	BkA	416834
SD039	BkB	416835
SD039	BmC	416836
SD039	Cu	416845
SD039	Dv	416848
SD039	Pc	416885
SD039	ReB	416895
SD039	So	416905
SD043	DbA	354887
SD043	EaB	354893
SD043	EaC	354894
SD043	HeB	354901
SD043	HgA	354902
SD043	Hv	354906
SD043	Oa	354910
SD043	Te	354915
SD045	He	354266
SD045	NmA	354276
SD045	NpB	354277
SD045	VwC	354288
SD045	VzE	354289
SD045	WnB	354292
SD045	WnC	354293
SD045	WtB	354295
SD057	BuE	417259
SD057	BxE	417263
SD057	ByD	417265
SD057	Lr	417289
SD057	PsB	417302
SD057	PwB	417305
SD057	SaD	417312
SD059	ReB	352669
SD059	RrC	352674
SD059	Tp	352683
SD059	WmB	352689
SD059	WnA	352691
SD059	WnB	352692
SD059	WpA	352693
SD059	WpB	352694
SD059	WuC	352699
SD059	ZrE	352709
SD069	CrA	353788
SD069	Cs	353789
SD069	Df	353790
SD069	Du	353792

Soil Layer	MUSYM	MUKEY
SD069	ErB	353795
SD069	HdA	353805
SD069	Ho	353809
SD069	RcA	353841
SD069	RcB	353842
SD069	RpC	353844
SD073	EaB	356427
SD073	EtD	356430
SD073	HwB	356446
SD073	HwC	356447
SD073	Ln	356449
SD073	Pr	356456
SD089	15B	354927
SD089	17B	354932
SD089	17C	354933
SD091	AsE	416532
SD091	BmF	416541
SD091	BtE	416543
SD091	FrB	416557

Soil Layer	MUSYM	MUKEY
SD091	FrC	416558
SD091	FrD	416559
SD091	FsC2	416560
SD091	FsE	416561
SD091	FtD	416562
SD091	SmE	416609
SD097	CnA	355390
SD097	CnB	355391
SD097	Du	355395
SD097	Wp	355412
SD099	EeB	446033
SD099	EuC	446041
SD099	La	446059
SD099	Lb	446060
SD109	BbE	417804
SD109	FoC	417822
SD109	FoD	417823
SD109	FvE	417828
SD109	HbA	417833

Soil Layer	MUSYM	MUKEY
SD109	Mr	417850
SD109	Pa	417852
SD109	Ra	417861
SD109	ReB	417863
SD109	SwB	417877
SD109	VhA	417881
SD109	VnB	417883
SD115	BxD	356574
SD115	FmA	356615
SD115	FmB	356616
SD115	FsA	356620
SD115	Pr	356702
SD115	Py	356703
SD603	BeB	355482
SD603	BgB	355484
SD603	EaA	355497
SD603	GhA	355503
SD603	GkB	355504
SD603	HgB	355505

Table C3: Excerpts from the spreadsheet "MapUnit_Components_ALL.xlsx". SoilLayer is code of the county soil layer. MUSYM is the map unit name. MUKEY is the map unit key. CompPct is the percent of each component that comprises the map unit. CompName is the name of each component that comprises the map unit. EcoSiteID is the code for each ecological site within the map unit. EcoSiteName is the name of the ecological site within the map unit. This table is a small section of the full Excel file for demonstrative purposes outlined in the body of this appendix.

SoilLayer	MUSYM	MUKEY	CompPct	CompName	EcoSiteID	EcoSiteName
MT091	BcD	348835	90	Blanchard	R053AE076MT	Sands (Sa) 10-14" p.z.
MT091	BcD	348835	5	Dooley	R053AE062MT	Sandy (Sy) 10-14" p.z.
MT091	BcD	348835	2	Lihen	R053AE076MT	Sands (Sa) 10-14" p.z.
MT091	BcD	348835	2	Zahill	R053AE065MT	Clayey-Steep (CyStp) 10-14" p.z.
MT091	BcD	348835	1	Parshall	R053AE062MT	Sandy (Sy) 10-14" p.z.
ND093	G147C	2581314	28	Buse	R055BY068ND	Thin Loamy
ND093	G147C	2581314	23	Barnes	R055BY064ND	Loamy
ND093	G147C	2581314	22	Darnen	R055BY059ND	Loamy Overflow
ND093	G147C	2581314	7	Langhei	R055BY068ND	Thin Loamy
ND093	G147C	2581314	6	Langhei	R055BY068ND	Thin Loamy
ND093	G147C	2581314	3	Cresbard	R055BY056ND	Clayey
ND093	G147C	2581314	3	Hamerly	R055BY058ND	Limy Subirrigated
ND093	G147C	2581314	3	Swenoda	R055BY062ND	Sandy
ND093	G147C	2581314	3	Sioux	R055BY069ND	Very Shallow
ND093	G147C	2581314	2	Lowe	R055BY071ND	Wet Meadow

Table C4: Ecological sites intersected by the NPAM transects and their classification into the two categories, "maintain" or "invade". M = maintain. I = Invade. NA = Not applicable. The general ecological site description (ESD) for the specific ecological site is also noted.

EcoSiteName	Classification	ESD
Clayey	I	Clayey
Clayey (Cy) RRU 58A-C 11-14" p.z.	I	Clayey
Clayey (Cy) 10-14" p.z.	I	Clayey
Clayey Overflow	I	Clayey Overflow
Clayey-Steep (CyStp) 10-14" p.z.	M	Clayey-Steep
Claypan	I	Claypan
Claypan (Cp) 10-14" p.z.	I	Claypan
Closed Depression	M	Closed Depression
Dense Clay (DC) RRU 58A-C 11-14" p.z.	M	Dense Clay
Dense Clay (DC) 10-14" p.z.	M	Dense Clay
Draft Overflow (Ov) RRU 46-C 13-19" p.z.	I	Draft Overflow
Gravel (Gr) 10-14" p.z.	M	Gravel
Limy Subirrigated	I	Limy Subirrigated
Linear Meadow	M	Linear Meadow
Loamy	I	Loamy
Loamy Overflow	I	Loamy Overflow
Non-Site	NA	NA
Overflow (Ov) 10-14" p.z.	I	Loamy Overflow
Overflow (Ov) 13-19" p.z.	I	Loamy Overflow
Saline Lowland	M	Saline Lowland
Saline Subirrigated	M	Saline Subirrigated
Saline Upland (SU) RRU 58A-C 11-14" p.z.	M	Saline Upland
Saline Upland (SU) 10-14" p.z.	M	Saline Upland
Sands	M	Sands
Sands (Sa) 10-14" p.z.	M	Sands
Sandy	M	Sandy
Sandy (Sy) 10-14" p.z.	M	Sandy
Sandy Claypan	M	Sandy Claypan
Shallow Gravel	M	Shallow Gravel
Shallow Loamy	M	Shallow Loamy
Shallow Marsh	M	Shallow Marsh
Shallow to Gravel	M	Shallow Gravel
Shallow to Gravel (SwGr) 10-14" p.z.	M	Shallow Gravel
Silty (Si) RRU 58A-C 11-14" p.z.	I	Loamy
Silty (Si) 10-14" p.z.	I	Loamy
Silty (Si) 13-19" p.z.	I	Loamy
Silty-Steep (SiStp) 10-14" p.z.	M	Silty-Steep
Subirrigated	I	Subirrigated
Subirrigated (Sb) 10-14" p.z.	I	Subirrigated
Subirrigated Sands	M	Subirrigated Sands
Thin Claypan	M	Thin Claypan
Thin Hilly (TH) 10-14" p.z.	I	Thin Hilly
Thin Loamy	M	Thin Loamy
Thin Silty (TSi) 15-19" p.z.	M	Thin Loamy
Thin Upland	M	Thin Loamy
Very Shallow	M	Very Shallow
Wet Land	M	Shallow Marsh
Wet Meadow	M	Wet Meadow
Wet Meadow (WM) 10-14" p.z.	M	Wet Meadow

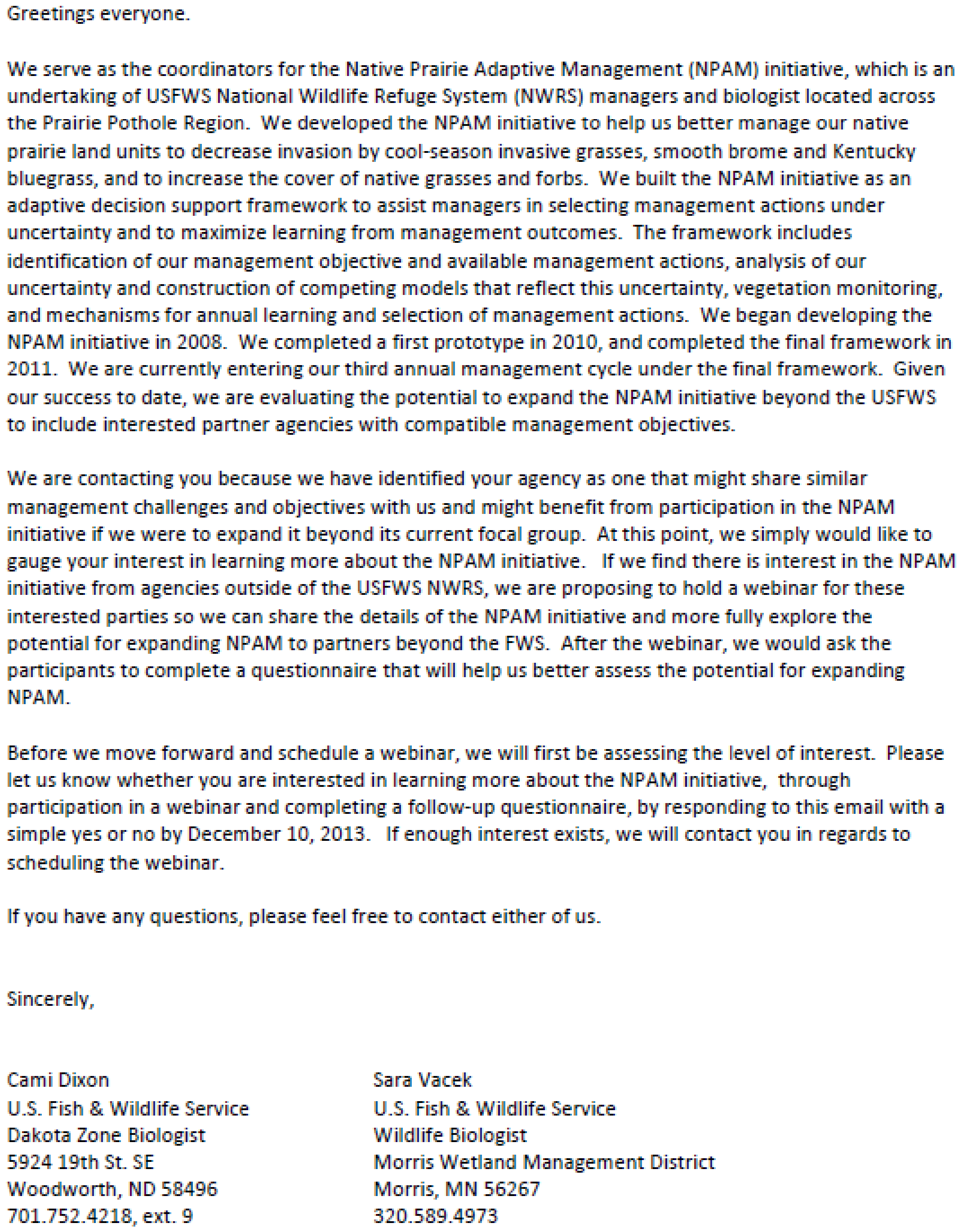
Table C5: Expansion of the example shown in Table C3, adding the classification information from Table C4, and summing the percent components for each type – Invade, Maintain, and NA – as well as over all components.

SoilLayer	MUSYM	CompPct	CompName	EcoSiteName	ESD	Classification	Sum Invade	Sum Maintain	Sum NA	Sum
MT091	BcD	90	Blanchard	Sands (Sa) 10-14" p.z.	Sands	M	0	100	0	100
MT091	BcD	5	Dooley	Sandy (Sy) 10-14" p.z.	Sandy	M				
MT091	BcD	2	Lihen	Sands (Sa) 10-14" p.z.	Sands	M				
MT091	BcD	2	Zahill	Clayey-Steep (CyStp) 10-14" p.z.	Clayey-Steep	M				
MT091	BcD	1	Parshall	Sandy (Sy) 10-14" p.z.	Sandy	M				
ND093	G147C	28	Buse	Thin Loamy	Thin Loamy	M	51	49	0	100
ND093	G147C	23	Barnes	Loamy	Loamy	I				
ND093	G147C	22	Darnen	Loamy Overflow	Loamy Overflow	I				
ND093	G147C	7	Langhei	Thin Loamy	Thin Loamy	M				
ND093	G147C	6	Langhei	Thin Loamy	Thin Loamy	M				
ND093	G147C	3	Cresbard	Clayey	Clayey	I				
ND093	G147C	3	Hamerly	Limy Subirrigated	Limy Subirrigated	I				
ND093	G147C	3	Swenoda	Sandy	Sandy	M				
ND093	G147C	3	Sioux	Very Shallow	Very Shallow	M				
ND093	G147C	2	Lowe	Wet Meadow	Wet Meadow	M				

Table C6: Range and average percent of map units that are comprised of invade and maintain ecological site components. Based on the composition of all NPAM transects (excluding those located in MN).

	Invade	**Maintain**
Average	53.4	46.6
Min	0	0
Max	100	100

Appendix D: Letter sent via email to the individuals in the list of identified contacts in Table 3.1.



Greetings everyone.

We serve as the coordinators for the Native Prairie Adaptive Management (NPAM) initiative, which is an undertaking of USFWS National Wildlife Refuge System (NWRS) managers and biologist located across the Prairie Pothole Region. We developed the NPAM initiative to help us better manage our native prairie land units to decrease invasion by cool-season invasive grasses, smooth brome and Kentucky bluegrass, and to increase the cover of native grasses and forbs. We built the NPAM initiative as an adaptive decision support framework to assist managers in selecting management actions under uncertainty and to maximize learning from management outcomes. The framework includes identification of our management objective and available management actions, analysis of our uncertainty and construction of competing models that reflect this uncertainty, vegetation monitoring, and mechanisms for annual learning and selection of management actions. We began developing the NPAM initiative in 2008. We completed a first prototype in 2010, and completed the final framework in 2011. We are currently entering our third annual management cycle under the final framework. Given our success to date, we are evaluating the potential to expand the NPAM initiative beyond the USFWS to include interested partner agencies with compatible management objectives.

We are contacting you because we have identified your agency as one that might share similar management challenges and objectives with us and might benefit from participation in the NPAM initiative if we were to expand it beyond its current focal group. At this point, we simply would like to gauge your interest in learning more about the NPAM initiative. If we find there is interest in the NPAM initiative from agencies outside of the USFWS NWRS, we are proposing to hold a webinar for these interested parties so we can share the details of the NPAM initiative and more fully explore the potential for expanding NPAM to partners beyond the FWS. After the webinar, we would ask the participants to complete a questionnaire that will help us better assess the potential for expanding NPAM.

Before we move forward and schedule a webinar, we will first be assessing the level of interest. Please let us know whether you are interested in learning more about the NPAM initiative, through participation in a webinar and completing a follow-up questionnaire, by responding to this email with a simple yes or no by December 10, 2013. If enough interest exists, we will contact you in regards to scheduling the webinar.

If you have any questions, please feel free to contact either of us.

Sincerely,

Cami Dixon
U.S. Fish & Wildlife Service
Dakota Zone Biologist
5924 19th St. SE
Woodworth, ND 58496
701.752.4218, ext. 9

Sara Vacek
U.S. Fish & Wildlife Service
Wildlife Biologist
Morris Wetland Management District
Morris, MN 56267
320.589.4973

Appendix E: Webinar presented to NPAM potential partners, 17 January 2014. See supplemental PDF of the slides at https://doi.org/10.3996/css82205527

Appendix F: Questionnaire sent to NPAM potential partners on 17 January 2014. See the questionnaire at https://doi.org/10.3996/css59949292

Appendix G: Program code archive

Synopsis of program code used for analyses. Although these programs have been used by the U.S. Geological Survey (USGS), no warranty, expressed or implied, is made by the USGS or the U.S. Government as to the accuracy and functioning of these programs and related program material nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith.

Chapter 1

- `DirichletEstimation_mixed.sas` and `D-m estimation_tall.sas` – SAS programs read raw data for mixed grass and tallgrass (respectively) prairie units from NPAM central database and estimate unit-level vegetation component proportions under a Dirichlet-multinomial model.
- `Mixed.sas` and `Tall.sas` – SAS programs read unit-level vegetation component proportions for mixed grass and tallgrass units (respectively) and compute component pivot logits, lagged predictors, and categorical predictors for linear models.
- `Reg_mixed(Response3)_ResponseSpecificConstraints_Float.sas` and `Reg_tall(Response3)_ResponseSpecificConstraints_Float - 6modeloutput.sas` – SAS programs fit Models 1-4 to mixed grass data and Models 1-6 to tallgrass data (respectively) and write estimated model parameters (model coefficients and error terms) to output files.
- `bootstrap3.sas` – SAS program reads estimated linear model parameters from a set of input files and runs Monte Carlo simulation to compute corresponding sets of state-and-transition models that are written to Excel spreadsheets.

Chapter 2

- `merge data files3_single file 2016.sas` – SAS file that reads Excel files of precipitation data, climate data, and transect-level ESD assignments, and merges data with SAS database files of treatment and monitoring data. Prepares data for analysis in R and writes output to a CSV file.
- `abiotic2016esd.r` and `abiotic2016esd.jags` – R and JAGS scripts to approximate posterior distributions of parameters for the mixed grass analysis.
- `abiotic2016prcp2.r` and `abiotic2016prcp.jags` – R and JAGS scripts to approximate posterior distributions of parameters for the all-units analysis.

```
options ls=140 ps=60 pageno=1;

** Directory for storage of MCMC results (SAS database format) **;
%let workpath = C:\Northern Prairie Project\Models\StatisticalModels_ProcReg\SAS\mixed\Thru2016;

** Parameters for bootstrap and MCMC samplers **
  ** Values are currently set for high performance and do not need to be changed **
  ** For these settings, samplers can take several minutes per unit -- be patient **;
  ** Random number seed **;
  %let rseed = 1234567;

  ** MCMC sampler parameters **;
    %let fuzz = 1e-200;       ** Small threshold value for Dirichlet generator **;
    %let mcmcreps = 1000000;  ** Number of MCMC samples **;
    %let tunereps = 5000;     ** Number of burn-in samples **;

libname results "&workpath";
/*
** MAKES A FAKE DATASET CALLED "TEST" WITH 2 UNITS, 3 YEARS EACH **;
**;
data test (keep=unit year trtmnt index transect sb kb np rm p_sb p_kb p_np p_rm);
  ** Make random, overdispersed veg counts **;
  array ntran[6] _temporary_ (15, 20, 25, 110, 100, 120);
  array trt[6] $ _temporary_ ("Rest", "Rest", "Graze", "B/G", "Rest", "Burn");
  array p[6,4] _temporary_ (.5, .3, .1, .1,
                            .2, .3, .4, .1,
                            .4, .01, .2, .39,
                            .1, .1, .79, .01,
                            .1, .5, .1, .3,
                            .1, .1, .1, .7 );
  seed = &rseed;
  i = 0;
  s = 1;  ** Degree of overdispersion;
  do unit="A","B";
    do year=2010 to 2012;
      i = i+1;
      trtmnt = trt[i];
      index = 7*ranuni(0);  ** Generate a random defoliation value **;
        do transect=1 to ntran[i];
          ** SB component;
          call rannor(seed, x);
          x0 = p[i,1];
          p_sb = 1/(1 + exp(-(x*s + log(x0/(1-x0)))));
          call ranbin(seed, 50, p_sb, sb);
          ** KB component;
          call rannor(seed, x);
          x0 = p[i,2]/(1-p[i,1]);
          p2c = 1/(1 + exp(-(x*s + log(x0/(1-x0)))));
          p_kb = p2c*(1-p_sb);
          call ranbin(seed, 50-sb, p2c, kb);
          ** NP component;
          call rannor(seed, x);
          x0 = p[i,3]/(1-p[i,1]-p[i,2]);
          p3c = 1/(1 + exp(-(x*s + log(x0/(1-x0)))));
          p_np = p3c*(1-p_sb-p_kb);
          call ranbin(seed, 50-sb-kb, p3c, np);
          ** RM component;
```

```
          rm = 50-sb-kb-np;
          p_rm = 1 - p_sb - p_kb - p_np;
          output;
        end;
    end;
  end;
  run;

proc print data=test;
  title1 "Printout of a fake dataset";
  var unit year trtmnt index transect sb kb np rm p_sb p_kb p_np p_rm;
  run;

title1;
*/

** IMPORT REAL DATA AND FORMAT TO RUN PROGRAM **;

PROC IMPORT OUT= WORK.veg09
            DATATABLE= "monitor2009_mixed"
            DBMS=ACCESSCS REPLACE;
     DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
     SCANMEMO=YES;
     USEDATE=NO;
     SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg10
            DATATABLE= "monitor2010_mixed"
            DBMS=ACCESSCS REPLACE;
     DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
     SCANMEMO=YES;
     USEDATE=NO;
     SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg11
            DATATABLE= "monitor2011_mixed"
            DBMS=ACCESSCS REPLACE;
     DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
     SCANMEMO=YES;
     USEDATE=NO;
     SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg12
            DATATABLE= "monitor2012_mixed"
            DBMS=ACCESSCS REPLACE;
     DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
     SCANMEMO=YES;
     USEDATE=NO;
     SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg13
            DATATABLE= "monitor2013_mixed"
            DBMS=ACCESSCS REPLACE;
```

```
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg14
          DATATABLE= "monitor2014_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg15
          DATATABLE= "monitor2015_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg16
          DATATABLE= "monitor2016_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;

PROC IMPORT OUT= WORK.mgmt09
          DATATABLE= "mgmt2009_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt10
          DATATABLE= "mgmt2010_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt11
          DATATABLE= "mgmt2011_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
```

```
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt12
          DATATABLE= "mgmt2012_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt13
          DATATABLE= "mgmt2013_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt14
          DATATABLE= "mgmt2014_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt15
          DATATABLE= "mgmt2015_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt16
          DATATABLE= "mgmt2016_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;

PROC IMPORT OUT= WORK.index09
          DATATABLE= "index2009_mixed"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
```

```
   SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index10
         DATATABLE= "index2010_mixed"
         DBMS=ACCESSCS REPLACE;
   DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
   SCANMEMO=YES;
   USEDATE=NO;
   SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index11
         DATATABLE= "index2011_mixed"
         DBMS=ACCESSCS REPLACE;
   DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
   SCANMEMO=YES;
   USEDATE=NO;
   SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index12
         DATATABLE= "index2012_mixed"
         DBMS=ACCESSCS REPLACE;
   DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
   SCANMEMO=YES;
   USEDATE=NO;
   SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index13
         DATATABLE= "index2013_mixed"
         DBMS=ACCESSCS REPLACE;
   DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
   SCANMEMO=YES;
   USEDATE=NO;
   SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index14
         DATATABLE= "index2014_mixed"
         DBMS=ACCESSCS REPLACE;
   DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
   SCANMEMO=YES;
   USEDATE=NO;
   SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index15
         DATATABLE= "index2015_mixed"
         DBMS=ACCESSCS REPLACE;
   DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
   SCANMEMO=YES;
   USEDATE=NO;
   SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index16
```

```
            DATATABLE= "index2016_mixed"
            DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;

data veg09 (keep=unit year transect SB KB NP RM);
        length unit $40.;
        format unit $40.;
        informat unit $40.;
        length year $4.;
        format year $4.;
        informat year $4.;
        length transect $8.;
        format transect $8.;
        informat transect $8.;
        set veg09;
run;
data veg10 (keep=unit year transect SB KB NP RM);
        length unit $40.;
        format unit $40.;
        informat unit $40.;
        length year $4.;
        format year $4.;
        informat year $4.;
        length transect $8.;
        format transect $8.;
        informat transect $8.;
        set veg10;
        if unit="EAST ML LAKE 10" then unit="East ML Lake 10";
run;
data veg11 (keep=unit year transect SBn KBn NPn RMn);
        length unit $40.;
        format unit $40.;
        informat unit $40.;
        length year $4.;
        format year $4.;
        informat year $4.;
        length transect $8.;
        format transect $8.;
        informat transect $8.;
        set veg11;
        SBn=SB*1;
        KBn=KB*1;
        NPn=NP*1;
        RMn=RM*1;
        if unit="EAST ML LAKE 10" then unit="East ML Lake 10";
run;
data veg11;
        set veg11;
        rename SBn=SB;
        label SBn=SB;
        rename KBn=KB;
        label KBn=KB;
```

```
		rename NPn=NP;
		label NPn=NP;
		rename RMn=RM;
		label RMn=RM;
run;
data veg12 (keep=unit year transect SBn KBn NPn RMn);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length transect $8.;
		format transect $8.;
		informat transect $8.;
		set veg12;
		SBn=SB*1;
		KBn=KB*1;
		NPn=NP*1;
		RMn=RM*1;
run;
data veg12;
		set veg12;
		rename SBn=SB;
		label SBn=SB;
		rename KBn=KB;
		label KBn=KB;
		rename NPn=NP;
		label NPn=NP;
		rename RMn=RM;
		label RMn=RM;
run;
data veg13 (keep=unit year transect SBn KBn NPn RMn);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length transect $8.;
		format transect $8.;
		informat transect $8.;
		set veg13;
		SBn=SB*1;
		KBn=KB*1;
		NPn=NP*1;
		RMn=RM*1;
run;
data veg13;
		set veg13;
		rename SBn=SB;
		label SBn=SB;
		rename KBn=KB;
		label KBn=KB;
		rename NPn=NP;
		label NPn=NP;
		rename RMn=RM;
```

```
		label RMn=RM;
run;
data veg14 (keep=unit year transect SBn KBn NPn RMn);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length transect $8.;
		format transect $8.;
		informat transect $8.;
		set veg14;
		SBn=SB*1;
		KBn=KB*1;
		NPn=NP*1;
		RMn=RM*1;
run;
data veg14;
		set veg14;
		rename SBn=SB;
		label SBn=SB;
		rename KBn=KB;
		label KBn=KB;
		rename NPn=NP;
		label NPn=NP;
		rename RMn=RM;
		label RMn=RM;
run;
data veg15 (keep=unit year transect SBn KBn NPn RMn);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length transect $8.;
		format transect $8.;
		informat transect $8.;
		set veg15;
		SBn=SB*1;
		KBn=KB*1;
		NPn=NP*1;
		RMn=RM*1;
run;
data veg15;
		set veg15;
		rename SBn=SB;
		label SBn=SB;
		rename KBn=KB;
		label KBn=KB;
		rename NPn=NP;
		label NPn=NP;
		rename RMn=RM;
		label RMn=RM;
run;
data veg16 (keep=unit year transect SBn KBn NPn RMn);
```

```
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length transect $8.;
		format transect $8.;
		informat transect $8.;
		set veg16;
		SBn=SB*1;
		KBn=KB*1;
		NPn=NP*1;
		RMn=RM*1;
run;
data veg16;
		set veg16;
		rename SBn=SB;
		label SBn=SB;
		rename KBn=KB;
		label KBn=KB;
		rename NPn=NP;
		label NPn=NP;
		rename RMn=RM;
		label RMn=RM;
run;
data mgmt09 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt09;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt10 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt10;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
		if unit="EAST ML LAKE 10" then unit="East ML Lake 10";
run;
data mgmt11 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt11;
```

```
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
		if unit="EAST ML LAKE 10" then unit="East ML Lake 10";
run;
data mgmt12 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt12;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt13 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt13;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt14 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt14;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt15 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt15;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt16 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt16;
		rename final_treatment_classification=trtmnt;
```

```
		label final_treatment_classification=trtmnt;
run;

data index09 (keep=unit index year);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length index 8;
		format index best12.;
		informat index 12.;
		set index09;
		year="2009";
run;
data index10 (keep=unit index year);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length index 8;
		format index best12.;
		informat index 12.;
		set index10;
		year="2010";
		if unit="EAST ML LAKE 10" then unit="East ML Lake 10";
run;
data index11 (keep=unit index year);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length index 8;
		format index best12.;
		informat index 12.;
		set index11;
		year="2011";
		if unit="EAST ML LAKE 10" then unit="East ML Lake 10";
run;
data index12 (keep=unit index year);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length index 8;
		format index best12.;
		informat index 12.;
		set index12;
		year="2012";
		if unit="EAST ML LAKE 10" then unit="East ML Lake 10";
```

```
run;
data index13 (keep=unit index year);
	length unit $40.;
	format unit $40.;
	informat unit $40.;
	length year $4.;
	format year $4.;
	informat year $4.;
	length index 8;
	format index best12.;
	informat index 12.;
	set index13;
	year="2013";
	if unit="EAST ML LAKE 10" then unit="East ML Lake 10";
run;
data index14 (keep=unit index year);
	length unit $40.;
	format unit $40.;
	informat unit $40.;
	length year $4.;
	format year $4.;
	informat year $4.;
	length index 8;
	format index best12.;
	informat index 12.;
	set index14;
	year="2014";
	if unit="Tande" and index > 0 then delete;
run;
data index15 (keep=unit index year);
	length unit $40.;
	format unit $40.;
	informat unit $40.;
	length year $4.;
	format year $4.;
	informat year $4.;
	length index 8;
	format index best12.;
	informat index 12.;
	set index15;
	year="2015";
run;
data index16 (keep=unit index year);
	length unit $40.;
	format unit $40.;
	informat unit $40.;
	length year $4.;
	format year $4.;
	informat year $4.;
	length index 8;
	format index best12.;
	informat index 12.;
	set index16;
	year="2016";
run;

proc sort data=veg09;
```

```
		by unit;
run;
proc sort data=veg10;
		by unit;
run;
proc sort data=veg11;
		by unit;
run;
proc sort data=veg12;
		by unit;
run;
proc sort data=veg13;
		by unit;
run;
proc sort data=veg14;
		by unit;
run;
proc sort data=veg15;
		by unit;
run;
proc sort data=veg16;
		by unit;
run;

proc sort data=mgmt09;
		by unit;
run;
proc sort data=mgmt10;
		by unit;
run;
proc sort data=mgmt11;
		by unit;
run;
proc sort data=mgmt12;
		by unit;
run;
proc sort data=mgmt13;
		by unit;
run;
proc sort data=mgmt14;
		by unit;
run;
proc sort data=mgmt15;
		by unit;
run;
proc sort data=mgmt16;
		by unit;
run;

proc sort data=index09;
		by unit;
run;
proc sort data=index10;
		by unit;
run;
proc sort data=index11;
		by unit;
```

```
run;
proc sort data=index12;
		by unit;
run;
proc sort data=index13;
		by unit;
run;
proc sort data=index14;
		by unit;
run;
proc sort data=index15;
		by unit;
run;
proc sort data=index16;
		by unit;
run;

data _09;
		merge veg09 mgmt09 index09;
		by unit;
run;
data _10;
		merge veg10 mgmt10 index10;
		by unit;
run;

data _11;
		merge veg11 mgmt11 index11;
		by unit;
run;
data _12;
		merge veg12 mgmt12 index12;
		by unit;
run;
data _13;
		merge veg13 mgmt13 index13;
		by unit;
run;
data _14;
		merge veg14 mgmt14 index14;
		by unit;
run;
data _15;
		merge veg15 mgmt15 index15;
		by unit;
run;
data _16;
		merge veg16 mgmt16 index16;
		by unit;
run;

data complete;
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
```

```
	informat year $4.;
	length trtmnt $255.;
	format trtmnt $255.;
	informat trtmnt $255.;
	length index 8;
	format index best12.;
	informat index 12.;
	length transect $8.;
	format transect $8.;
	informat transect $8.;
	length NP 8;
	format NP best12.;
	informat NP 12.;
	length SB 8;
	format SB best12.;
	informat SB 12.;
	length KB 8;
	format KB best12.;
	informat KB 12.;
	length RM 8;
	format RM best12.;
	informat RM 12.;
run;

proc append base=complete data=_09 force;
run;
proc append base=complete data=_10 force;
run;
proc append base=complete data=_11 force;
run;
proc append base=complete data=_12 force;
run;
proc append base=complete data=_13 force;
run;
proc append base=complete data=_14 force;
run;
proc append base=complete data=_15 force;
run;
proc append base=complete data=_16 force;
run;

/*September 16, 2016:  I need to check if these modifications are still valid and necessary.  Yes, these
modifications are necessary given name changes to units (splitting or merging) in order to have the paired
data for consecutive years.*/
data complete;
	set complete;
	if unit="" then delete;
	if unit="Peterson South" and year="2012" then index=4.762870635;
	if unit="Peterson South" and year="2012" then trtmnt="Burn/Graze";
	if unit="G28 North" and year="2014" then trtmnt="Rest";
	if unit="G28 South" and year="2014" then trtmnt="Rest";
	if unit="Woodworth Station Unit 7 East" and year="2014" then trtmnt="Rest";
	if unit="Woodworth Station Unit 7 West" and year="2014" then trtmnt="Rest";
run;

proc sort data=complete;
	by unit year;
```

```
run;

data missing;
            set complete;
            where NP*SB*KB*RM=.;
run;

data complete;
            set complete;
            if NP*SB*KB*RM=. then delete;
run;

libname model "C:\Northern Prairie Project\Models\StatisticalModels_ProcReg\SAS\mixed\Thru2016";

data model.complete_float;
            set work.complete;
run;

** VARIABLE NAMES ASSUMED:  UNIT, YEAR, TRTMNT, INDEX, TRANSECT, NP, SB, KB, RM **;

** Sort data by unit then by year;
proc sort data=model.complete_float out=a;
            by unit year;
run;

** Count up number of unit x year combinations and store in macro variable;
data unityrs (keep=unit year trtmnt index unityear) a;
  set a;
  by unit year;
  retain unityear 0;
  if first.year then
    do;
      unityear+1;
      call symput('n_unityrs',put(unityear,3.));
      output unityrs;
    end;
  output a;
  run;

** MCMC sampler **;
%macro mcmc;

  ** Clear out any previous results database to receive these results **;
  proc datasets library=work nolist nowarn;
    delete mcmc;
    run;

  %* Macro loop over the year-unit combinations **;
  %do i=1 %to &n_unityrs;

    data b;
      set a (where=(unityear=&i));
      call symput('unit',trim(unit));
      call symput('year',put(year,4.));
    run;

    proc iml;
```

```
*** Procedure definitions ***;

start stateassign(x);
  ** This procedure returns an assignment of vegetation state (1-16)
  **   from composition proportions of SB, KB, NP, and RM.
  ** Argument
  **  x: n x 4 vector of vegetation composition proportions
  ** Return
  **  y: n x 1 vector -- assignment of state **;
  n = nrow(x);
  r = j(n,4,0);
  np = 1 + (x[,3]>=0.3) + (x[,3]>=0.45) + (x[,3]>=0.6);
  if x[,3]<1.0 then r[,3] = x[,4] / (1 - x[,3]);
  else r[,3] = 0;
  if x[,1]+x[,2]>0 then
    do;
      r[,1] = x[,1] / (x[,1] + x[,2]);
      r[,2] = 1 - r[,1];
    end;
  else
    do;
      r[,1] = 0;
      r[,2] = 0;
    end;
  dom = 4 * (r[,3] >= 2/3) +
        1 * (2/3 >= r[,3]) # (r[,1] >= 2/3) +
        3 * (2/3 >= r[,3]) # (r[,2] >= 2/3) +
        2 * (2/3 >= r[,3]) # (2/3 >= r[,1]) # (2/3 >= r[,2]);
  state = 4*(np-1) + dom;
  return(state);
finish;

start GenerateDirich(alpha);
  ** Generates a matrix of Dirichlet random variates of size K **;
  ** Argument
  **   alpha: J x K matrix of Dirichlet parameters
  ** Return
  **   g: J x K matrix of J Dirichlet draws, each of dimension K (each row sums to 1) ;
  g = j(nrow(alpha),ncol(alpha),1);
  **  call randgen(g,'gamma',alpha);  ** This form of call does not work **;
  do r=1 to nrow(alpha);
    do c=1 to ncol(alpha);
      call randgen(z,'gamma',alpha[r,c]);
      g[r,c] = max(z,&fuzz);
    end;
  end;
  g = g / (j(1,ncol(alpha),1) @ g[,+]);
  return(g);
finish;

start logFullCondAlphaK(p,alpha,k);
  ** Computes conditional (on kth alpha) log likelihood of Dirichlet
  ** Arguments
  **   p: J x 1 vector of kth-component proportions
  **   alpha: K x 1 vector of Dirichlet parameters
  **   k: scalar referencing focal Dirichlet parameter
```

```
  ** Return
  **   y: scalar  ;
  y = ( (alpha[k]-1) * (log(p))[+] ) - log(alpha[k]) +
     ( nrow(p) * (lgamma(alpha[+]) - lgamma(alpha[k])) );
  return(y);
finish;

start mcmc(x,mcmcreps,tunetime,rseed,phase,y,samples);
  ** This procedure performs MCMC sampling to compute posterior distribution
  **   of parameters of unit-level Dirichlet and posterior distribution of
  **   vegetation state assignment.
  **   (Based on Gauss code provided by Bill Link, PWRC)
  ** Arguments
  **  x : n x 4 matrix of vegetation type counts
  **  mcmcreps: number of MCMC samples
  **  tunereps: number of tuning samples
  **  rseed: random number seed
  **  phase: indicator of which pair this procedure is processing
  ** Returns
  **  y : 1 x 24 vector of output
  **          pos 1-4: vegetation component simulation means
  **          pos 5-8: vegetation component simulation SDs
  **          pos 8-24: means of state assignments
  **  samples : mcmcreps x 1 vector of state samples  **;
  Nmultis = nrow(x);
  Ncells = ncol(x);
  call randseed(&rseed);
  call randgen(rseed,'uniform');
  rseed = int(rseed*10e8);
  allalpha = j(mcmcreps+tunetime,Ncells,0);
  allalpha[1,] = j(1,Ncells,1)/2;
  maxstepsize = allalpha[1,]/2;
  h = datetime();
  h0 = h;
  i = 1;
  do while(i<mcmcreps + tunetime);
    ** Progress monitor (updates log every 15 seconds) **;
    if datetime()-h0>15 then
      do;
        h0 = datetime();
        progress = i/(mcmcreps+tunetime)*100;
        remain = ((h0-h)/progress*100 - (h0-h)) / 60;
        put "MCMC step " phase 1.0 " of 1, " progress 3.0
            "% complete, about " remain 4.1
            " minutes remain for this phase (&unit, &year)";
      end;
    i = i+1;
    alphacurr = allalpha[i-1,];
    ** Gibbs sampling of p vectors **;
    pcurr = GenerateDirich( (alphacurr @ j(Nmultis,1,1)) + x );
    ** Gibbs sampling of alphas **;
    k = 0;
    do while(k<Ncells);
      k = k+1;
      alphacand = alphacurr;
      alphacand[k] = alphacand[k] + (2*uniform(rseed)-1) * maxstepsize[k];
      if alphacand[k]<=0 then
```

```
        do;
          if i<tunetime then maxstepsize[k] = maxstepsize[k] * 0.98;
        end;
      else
        do;
          logr = logFullCondAlphaK(pcurr[,k],alphacand,k) -
                 logFullCondAlphaK(pcurr[,k],alphacurr,k);
          if log(uniform(rseed))<logr then
            do;
              alphacurr = alphacand;
              if i<tunetime then maxstepsize[k] = maxstepsize[k] * 1.025;
            end;
          else if i<tunetime then maxstepsize[k] = maxstepsize[k] * 0.98;
        end;
    end;
    allalpha[i,] = alphacurr;
  end;
  ** Remove burn-in samples **;
  allalpha = allalpha[tunetime+1:mcmcreps+tunetime,];
  ** Posterior samples of component proportions **;
  p = allalpha / (j(1,Ncells,1) @ allalpha[,+]);
  ** Posterior samples of state assignments **;
  samples = stateassign(p);
  pavgsum = p[+,];
  pavgsum2 = p[##,];
  pavgmean = pavgsum/mcmcreps;
  pavgsd = sqrt( ( pavgsum2 - (pavgsum # pavgsum)/mcmcreps ) / mcmcreps );
  m = ((samples @ j(1,16,1)) = (j(mcmcreps,1,1) @ (1:16)))[+,] / mcmcreps;
  ** Return MCMC posterior samples of state assignment **;
  y = shape(pavgmean,1,4) || shape(pavgsd,1,4) || m;
  put "MCMC step " phase 1.0 " of 1 finished (&unit, &year)" /;
finish;


** Read data and create input matrices **;
use b;
read all var{sb kb np rm} into counts;

** MCMC execution **;
file log;
m1 = j(1,24,0);
s1 = j(&mcmcreps,1,0);
** Call MCMC procedure for each year **;
call mcmc(counts[,1:4],&mcmcreps,&tunereps,&rseed,1,m1,s1);
free s1;
x = (&i) || m1;

** Output results to dataset **;
names = 'unityear' || ('mean1':'mean4') || ('sd1':'sd4') || ('sa1':'sa16');
names = 'unityear' || 'sb' || 'kb' || 'np' || 'rm' ||
           'sb_sd' || 'kb_sd' || 'np_sd' || 'rm_sd' ||
           '_30SB' || '_30CO' || '_30KB' || '_30RM' ||
           '_45SB' || '_45CO' || '_45KB' || '_45RM' ||
           '_60SB' || '_60CO' || '_60KB' || '_60RM' ||
           '_100SB' || '_100CO' || '_100KB' || '_100RM';
create mcmcadd var names;
append from x;
```

```
      quit;

    ** Append the results for current unit to results for previous units **;
    proc datasets library=work nolist nowarn;
      append base=mcmc data=mcmcadd;
      run;

  %end;

  ** Write temporary datasets to permanent SAS databases **;
  data results.mcmc;  set mcmc;  run;

%mend;


%mcmc

data all_float;
  merge unityrs results.mcmc;
  by unityear;
  run;

proc print data=all_float;  run;

quit;


data model.all_float;
                set work.all_float;
run;
```

```
options ls=140 ps=60 pageno=1;

** Directory for storage of MCMC results (SAS database format) **;
%let workpath = C:\Northern Prairie Project\Models\StatisticalModels_ProcReg\SAS\tall\Thru2016;

** Parameters for bootstrap and MCMC samplers **
  ** Values are currently set for high performance and do not need to be changed **
  ** For these settings, samplers can take several minutes per unit -- be patient **;
  ** Random number seed **;
  %let rseed = 1234567;

  ** MCMC sampler parameters **;
    %let fuzz = 1e-200;        ** Small threshold value for Dirichlet generator **;
    %let mcmcreps = 1000000;  ** Number of MCMC samples **;
    %let tunereps = 5000;     ** Number of burn-in samples **;

libname results "&workpath";
/*
** MAKES A FAKE DATASET CALLED "TEST" WITH 2 UNITS, 3 YEARS EACH **;
**;
data test (keep=unit year trtmnt index transect sb kb np rm p_sb p_kb p_np p_rm);
  ** Make random, overdispersed veg counts **;
  array ntran[6] _temporary_ (15, 20, 25, 110, 100, 120);
  array trt[6] $ _temporary_ ("Rest", "Rest", "Graze", "B/G", "Rest", "Burn");
  array p[6,4] _temporary_ (.5, .3, .1, .1,
                            .2, .3, .4, .1,
                            .4, .01, .2, .39,
                            .1, .1, .79, .01,
                            .1, .5, .1, .3,
                            .1, .1, .1, .7 );
  seed = &rseed;
  i = 0;
  s = 1;  ** Degree of overdispersion;
  do unit="A","B";
    do year=2010 to 2012;
      i = i+1;
      trtmnt = trt[i];
      index = 7*ranuni(0);  ** Generate a random defoliation value **;
        do transect=1 to ntran[i];
          ** SB component;
          call rannor(seed, x);
          x0 = p[i,1];
          p_sb = 1/(1 + exp(-(x*s + log(x0/(1-x0)))));
          call ranbin(seed, 50, p_sb, sb);
          ** KB component;
          call rannor(seed, x);
          x0 = p[i,2]/(1-p[i,1]);
          p2c = 1/(1 + exp(-(x*s + log(x0/(1-x0)))));
          p_kb = p2c*(1-p_sb);
          call ranbin(seed, 50-sb, p2c, kb);
          ** NP component;
          call rannor(seed, x);
          x0 = p[i,3]/(1-p[i,1]-p[i,2]);
          p3c = 1/(1 + exp(-(x*s + log(x0/(1-x0)))));
          p_np = p3c*(1-p_sb-p_kb);
          call ranbin(seed, 50-sb-kb, p3c, np);
          ** RM component;
```

```
          rm = 50-sb-kb-np;
          p_rm = 1 - p_sb - p_kb - p_np;
          output;
        end;
    end;
  end;
  run;

proc print data=test;
  title1 "Printout of a fake dataset";
  var unit year trtmnt index transect sb kb np rm p_sb p_kb p_np p_rm;
  run;

title1;
*/

** IMPORT REAL DATA AND FORMAT TO RUN PROGRAM **;

PROC IMPORT OUT= WORK.veg09
            DATATABLE= "monitor2009_tall"
            DBMS=ACCESSCS REPLACE;
     DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
     SCANMEMO=YES;
     USEDATE=NO;
     SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg10
            DATATABLE= "monitor2010_tall"
            DBMS=ACCESSCS REPLACE;
     DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
     SCANMEMO=YES;
     USEDATE=NO;
     SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg11
            DATATABLE= "monitor2011_tall"
            DBMS=ACCESSCS REPLACE;
     DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
     SCANMEMO=YES;
     USEDATE=NO;
     SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg12
            DATATABLE= "monitor2012_tall"
            DBMS=ACCESSCS REPLACE;
     DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
     SCANMEMO=YES;
     USEDATE=NO;
     SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg13
            DATATABLE= "monitor2013_tall"
            DBMS=ACCESSCS REPLACE;
```

```
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg14
          DATATABLE= "monitor2014_tall"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg15
          DATATABLE= "monitor2015_tall"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.veg16
          DATATABLE= "monitor2016_tall"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;

PROC IMPORT OUT= WORK.mgmt09
          DATATABLE= "Mgmt2009_tall"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt10
          DATATABLE= "mgmt2010_tall"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt11
          DATATABLE= "mgmt2011_tall"
          DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
```

```
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt12
         DATATABLE= "mgmt2012_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt13
         DATATABLE= "mgmt2013_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt14
         DATATABLE= "mgmt2014_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt15
         DATATABLE= "mgmt2015_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.mgmt16
         DATATABLE= "mgmt2016_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;

PROC IMPORT OUT= WORK.index09
         DATATABLE= "index2009_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
```

```
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index10
         DATATABLE= "index2010_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index11
         DATATABLE= "index2011_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index12
         DATATABLE= "index2012_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index13
         DATATABLE= "index2013_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index14
         DATATABLE= "index2014_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index15
         DATATABLE= "index2015_tall"
         DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;
PROC IMPORT OUT= WORK.index16
```

```
        DATATABLE= "index2016_tall"
        DBMS=ACCESSCS REPLACE;
    DATABASE="C:\Northern Prairie Project\Database\Coordinator
Database\NPAMDatabase_08292016\NPAM_2016_DB\NPAM_2016_DB.mdb";
    SCANMEMO=YES;
    USEDATE=NO;
    SCANTIME=YES;
RUN;

data veg09 (keep=unit year transect SB KB NP RM);
        length unit $40.;
        format unit $40.;
        informat unit $40.;
        length year $4.;
        format year $4.;
        informat year $4.;
        length transect $8.;
        format transect $8.;
        informat transect $8.;
        set veg09;
run;
data veg10 (keep=unit year transect SB KB NP RM);
        length unit $40.;
        format unit $40.;
        informat unit $40.;
        length year $4.;
        format year $4.;
        informat year $4.;
        length transect $8.;
        format transect $8.;
        informat transect $8.;
        set veg10;
run;
data veg11 (keep=unit year transect SBn KBn NPn RMn);
        length unit $40.;
        format unit $40.;
        informat unit $40.;
        length year $4.;
        format year $4.;
        informat year $4.;
        length transect $8.;
        format transect $8.;
        informat transect $8.;
        set veg11;
        SBn=SB*1;
        KBn=KB*1;
        NPn=NP*1;
        RMn=RM*1;
run;
data veg11;
        set veg11;
        rename SBn=SB;
        label SBn=SB;
        rename KBn=KB;
        label KBn=KB;
        rename NPn=NP;
        label NPn=NP;
```

```
		rename RMn=RM;
		label RMn=RM;
run;
data veg12 (keep=unit year transect SBn KBn NPn RMn);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length transect $8.;
		format transect $8.;
		informat transect $8.;
		set veg12;
		SBn=SB*1;
		KBn=KB*1;
		NPn=NP*1;
		RMn=RM*1;
run;
data veg12;
		set veg12;
		rename SBn=SB;
		label SBn=SB;
		rename KBn=KB;
		label KBn=KB;
		rename NPn=NP;
		label NPn=NP;
		rename RMn=RM;
		label RMn=RM;
run;
data veg13 (keep=unit year transect SBn KBn NPn RMn);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length transect $8.;
		format transect $8.;
		informat transect $8.;
		set veg13;
		SBn=SB*1;
		KBn=KB*1;
		NPn=NP*1;
		RMn=RM*1;
run;
data veg13;
		set veg13;
		rename SBn=SB;
		label SBn=SB;
		rename KBn=KB;
		label KBn=KB;
		rename NPn=NP;
		label NPn=NP;
		rename RMn=RM;
		label RMn=RM;
run;
```

```
data veg14 (keep=unit year transect SBn KBn NPn RMn);
        length unit $40.;
        format unit $40.;
        informat unit $40.;
        length year $4.;
        format year $4.;
        informat year $4.;
        length transect $8.;
        format transect $8.;
        informat transect $8.;
        set veg14;
        SBn=SB*1;
        KBn=KB*1;
        NPn=NP*1;
        RMn=RM*1;
run;
data veg14;
        set veg14;
        rename SBn=SB;
        label SBn=SB;
        rename KBn=KB;
        label KBn=KB;
        rename NPn=NP;
        label NPn=NP;
        rename RMn=RM;
        label RMn=RM;
run;
data veg15 (keep=unit year transect SBn KBn NPn RMn);
        length unit $40.;
        format unit $40.;
        informat unit $40.;
        length year $4.;
        format year $4.;
        informat year $4.;
        length transect $8.;
        format transect $8.;
        informat transect $8.;
        set veg15;
        SBn=SB*1;
        KBn=KB*1;
        NPn=NP*1;
        RMn=RM*1;
run;
data veg15;
        set veg15;
        rename SBn=SB;
        label SBn=SB;
        rename KBn=KB;
        label KBn=KB;
        rename NPn=NP;
        label NPn=NP;
        rename RMn=RM;
        label RMn=RM;
run;
data veg16 (keep=unit year transect SBn KBn NPn RMn);
        length unit $40.;
        format unit $40.;
```

```
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length transect $8.;
		format transect $8.;
		informat transect $8.;
		set veg16;
		SBn=SB*1;
		KBn=KB*1;
		NPn=NP*1;
		RMn=RM*1;
run;
data veg16;
		set veg16;
		rename SBn=SB;
		label SBn=SB;
		rename KBn=KB;
		label KBn=KB;
		rename NPn=NP;
		label NPn=NP;
		rename RMn=RM;
		label RMn=RM;
run;

data mgmt09 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt09;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt10 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt10;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt11 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt11;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
```

```
run;
data mgmt12 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt12;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt13 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt13;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt14 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt14;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt15 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt15;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
data mgmt16 (keep=unit year trtmnt);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		set mgmt16;
		rename final_treatment_classification=trtmnt;
		label final_treatment_classification=trtmnt;
run;
```

```
data index09 (keep=unit index year);
	length unit $40.;
	format unit $40.;
	informat unit $40.;
	length year $4.;
	format year $4.;
	informat year $4.;
	length index 8;
	format index best12.;
	informat index 12.;
	set index09;
	year="2009";
run;
data index10 (keep=unit index year);
	length unit $40.;
	format unit $40.;
	informat unit $40.;
	length year $4.;
	format year $4.;
	informat year $4.;
	length index 8;
	format index best12.;
	informat index 12.;
	set index10;
	year="2010";
run;
data index11 (keep=unit index year);
	length unit $40.;
	format unit $40.;
	informat unit $40.;
	length year $4.;
	format year $4.;
	informat year $4.;
	length index 8;
	format index best12.;
	informat index 12.;
	set index11;
	year="2011";
run;
data index12 (keep=unit index year);
	length unit $40.;
	format unit $40.;
	informat unit $40.;
	length year $4.;
	format year $4.;
	informat year $4.;
	length index 8;
	format index best12.;
	informat index 12.;
	set index12;
	year="2012";
run;
data index13 (keep=unit index year);
	length unit $40.;
	format unit $40.;
	informat unit $40.;
	length year $4.;
```

```
		format year $4.;
		informat year $4.;
		length index 8;
		format index best12.;
		informat index 12.;
		set index13;
		year="2013";
run;
data index14 (keep=unit index year);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length index 8;
		format index best12.;
		informat index 12.;
		set index14;
		year="2014";
run;
data index15 (keep=unit index year);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length index 8;
		format index best12.;
		informat index 12.;
		set index15;
		year="2015";
run;
data index16 (keep=unit index year);
		length unit $40.;
		format unit $40.;
		informat unit $40.;
		length year $4.;
		format year $4.;
		informat year $4.;
		length index 8;
		format index best12.;
		informat index 12.;
		set index16;
		year="2016";
run;

proc sort data=veg09;
		by unit;
run;
proc sort data=veg10;
		by unit;
run;
proc sort data=veg11;
		by unit;
run;
```

```
proc sort data=veg12;
		by unit;
run;
proc sort data=veg13;
		by unit;
run;
proc sort data=veg14;
		by unit;
run;
proc sort data=veg15;
		by unit;
run;
proc sort data=veg16;
		by unit;
run;

proc sort data=mgmt09;
		by unit;
run;
proc sort data=mgmt10;
		by unit;
run;
proc sort data=mgmt11;
		by unit;
run;
proc sort data=mgmt12;
		by unit;
run;
proc sort data=mgmt13;
		by unit;
run;
proc sort data=mgmt14;
		by unit;
run;
proc sort data=mgmt15;
		by unit;
run;
proc sort data=mgmt16;
		by unit;
run;

proc sort data=index09;
		by unit;
run;
proc sort data=index10;
		by unit;
run;
proc sort data=index11;
		by unit;
run;
proc sort data=index12;
		by unit;
run;
proc sort data=index13;
		by unit;
run;
proc sort data=index14;
```

```
	by unit;
run;
proc sort data=index15;
	by unit;
run;
proc sort data=index16;
	by unit;
run;

data _09;
	merge veg09 mgmt09 index09;
	by unit;
run;
data _10;
	merge veg10 mgmt10 index10;
	by unit;
run;

data _11;
	merge veg11 mgmt11 index11;
	by unit;
run;
data _12;
	merge veg12 mgmt12 index12;
	by unit;
run;
data _13;
	merge veg13 mgmt13 index13;
	by unit;
run;
data _14;
	merge veg14 mgmt14 index14;
	by unit;
run;
data _15;
	merge veg15 mgmt15 index15;
	by unit;
run;
data _16;
	merge veg16 mgmt16 index16;
	by unit;
run;

data complete;
	length unit $40.;
	format unit $40.;
	informat unit $40.;
	length year $4.;
	format year $4.;
	informat year $4.;
	length trtmnt $255.;
	format trtmnt $255.;
	informat trtmnt $255.;
	length index 8;
	format index best12.;
	informat index 12.;
	length transect $8.;
```

```
		format transect $8.;
		informat transect $8.;
		length NP 8;
		format NP best12.;
		informat NP 12.;
		length SB 8;
		format SB best12.;
		informat SB 12.;
		length KB 8;
		format KB best12.;
		informat KB 12.;
		length RM 8;
		format RM best12.;
		informat RM 12.;
run;

proc append base=complete data=_09 force;
run;
proc append base=complete data=_10 force;
run;
proc append base=complete data=_11 force;
run;
proc append base=complete data=_12 force;
run;
proc append base=complete data=_13 force;
run;
proc append base=complete data=_14 force;
run;
proc append base=complete data=_15 force;
run;
proc append base=complete data=_16 force;
run;

data complete;
		set complete;
		if unit="" then delete;
run;

proc sort data=complete;
		by unit year;
run;

data missing;
		set complete;
		where NP*SB*KB*RM=.;
run;

data complete;
		set complete;
		if NP*SB*KB*RM=. then delete;
run;

libname model "C:\Northern Prairie Project\Models\StatisticalModels_ProcReg\SAS\tall\Thru2016";

data model.complete_float;
		set work.complete;
run;
```

```
*/

** VARIABLE NAMES ASSUMED:  UNIT, YEAR, TRTMNT, INDEX, TRANSECT, NP, SB, KB, RM **;

** Sort data by unit then by year;
proc sort data=model.complete_float out=a;
        by unit year;
run;

** Count up number of unit x year combinations and store in macro variable;
data unityrs (keep=unit year trtmnt index unityear) a;
  set a;
  by unit year;
  retain unityear 0;
  if first.year then
    do;
      unityear+1;
      call symput('n_unityrs',put(unityear,3.));
      output unityrs;
    end;
  output a;
  run;

** MCMC sampler **;
%macro mcmc;

  ** Clear out any previous results database to receive these results **;
  proc datasets library=work nolist nowarn;
    delete mcmc;
    run;

  %* Macro loop over the year-unit combinations **;
  %do i=1 %to &n_unityrs;

    data b;
      set a (where=(unityear=&i));
      call symput('unit',trim(unit));
      call symput('year',put(year,4.));
    run;

    proc iml;

      *** Procedure definitions ***;

      start stateassign(x);
        ** This procedure returns an assignment of vegetation state (1-16)
        **   from composition proportions of SB, KB, NP, and RM.
        ** Argument
        **  x: n x 4 vector of vegetation composition proportions
        ** Return
        **  y: n x 1 vector -- assignment of state **;
        n = nrow(x);
        r = j(n,4,0);
        np = 1 + (x[,3]>=0.3) + (x[,3]>=0.45) + (x[,3]>=0.6);
        if x[,3]<1.0 then r[,3] = x[,4] / (1 - x[,3]);
        else r[,3] = 0;
        if x[,1]+x[,2]>0 then
```

```
    do;
      r[,1] = x[,1] / (x[,1] + x[,2]);
      r[,2] = 1 - r[,1];
    end;
  else
    do;
      r[,1] = 0;
      r[,2] = 0;
    end;
  dom = 4 * (r[,3] >= 2/3) +
        1 * (2/3 >= r[,3]) # (r[,1] >= 2/3) +
        3 * (2/3 >= r[,3]) # (r[,2] >= 2/3) +
        2 * (2/3 >= r[,3]) # (2/3 >= r[,1]) # (2/3 >= r[,2]);
  state = 4*(np-1) + dom;
  return(state);
finish;

start GenerateDirich(alpha);
  ** Generates a matrix of Dirichlet random variates of size K **;
  ** Argument
  **   alpha: J x K matrix of Dirichlet parameters
  ** Return
  **   g: J x K matrix of J Dirichlet draws, each of dimension K (each row sums to 1) ;
  g = j(nrow(alpha),ncol(alpha),1);
  **  call randgen(g,'gamma',alpha);  ** This form of call does not work **;
  do r=1 to nrow(alpha);
    do c=1 to ncol(alpha);
      call randgen(z,'gamma',alpha[r,c]);
      g[r,c] = max(z,&fuzz);
    end;
  end;
  g = g / (j(1,ncol(alpha),1) @ g[,+]);
  return(g);
finish;

start logFullCondAlphaK(p,alpha,k);
  ** Computes conditional (on kth alpha) log likelihood of Dirichlet
  ** Arguments
  **   p: J x 1 vector of kth-component proportions
  **   alpha: K x 1 vector of Dirichlet parameters
  **   k: scalar referencing focal Dirichlet parameter
  ** Return
  **   y: scalar  ;
  y = ( (alpha[k]-1) * (log(p))[+] ) - log(alpha[k]) +
      ( nrow(p) * (lgamma(alpha[+]) - lgamma(alpha[k])) );
  return(y);
finish;

start mcmc(x,mcmcreps,tunetime,rseed,phase,y,samples);
  ** This procedure performs MCMC sampling to compute posterior distribution
  **   of parameters of unit-level Dirichlet and posterior distribution of
  **   vegetation state assignment.
  **   (Based on Gauss code provided by Bill Link, PWRC)
  ** Arguments
  **  x : n x 4 matrix of vegetation type counts
  **  mcmcreps: number of MCMC samples
  **  tunereps: number of tuning samples
```

```
** rseed: random number seed
** phase: indicator of which pair this procedure is processing
** Returns
** y : 1 x 24 vector of output
**         pos 1-4: vegetation component simulation means
**         pos 5-8: vegetation component simulation SDs
**         pos 8-24: means of state assignments
** samples : mcmcreps x 1 vector of state samples **;
Nmultis = nrow(x);
Ncells = ncol(x);
call randseed(&rseed);
call randgen(rseed,'uniform');
rseed = int(rseed*10e8);
allalpha = j(mcmcreps+tunetime,Ncells,0);
allalpha[1,] = j(1,Ncells,1)/2;
maxstepsize = allalpha[1,]/2;
h = datetime();
h0 = h;
i = 1;
do while(i<mcmcreps + tunetime);
  ** Progress monitor (updates log every 15 seconds) **;
  if datetime()-h0>15 then
    do;
      h0 = datetime();
      progress = i/(mcmcreps+tunetime)*100;
      remain = ((h0-h)/progress*100 - (h0-h)) / 60;
      put "MCMC step " phase 1.0 " of 1, " progress 3.0
          "% complete, about " remain 4.1
          " minutes remain for this phase (&unit, &year)";
    end;
  i = i+1;
  alphacurr = allalpha[i-1,];
  ** Gibbs sampling of p vectors **;
  pcurr = GenerateDirich( (alphacurr @ j(Nmultis,1,1)) + x );
  ** Gibbs sampling of alphas **;
  k = 0;
  do while(k<Ncells);
    k = k+1;
    alphacand = alphacurr;
    alphacand[k] = alphacand[k] + (2*uniform(rseed)-1) * maxstepsize[k];
    if alphacand[k]<=0 then
      do;
        if i<tunetime then maxstepsize[k] = maxstepsize[k] * 0.98;
      end;
    else
      do;
        logr = logFullCondAlphaK(pcurr[,k],alphacand,k) -
               logFullCondAlphaK(pcurr[,k],alphacurr,k);
        if log(uniform(rseed))<logr then
          do;
            alphacurr = alphacand;
            if i<tunetime then maxstepsize[k] = maxstepsize[k] * 1.025;
          end;
        else if i<tunetime then maxstepsize[k] = maxstepsize[k] * 0.98;
      end;
  end;
  allalpha[i,] = alphacurr;
```

```
        end;
        ** Remove burn-in samples **;
        allalpha = allalpha[tunetime+1:mcmcreps+tunetime,];
        ** Posterior samples of component proportions **;
        p = allalpha / (j(1,Ncells,1) @ allalpha[,+]);
        ** Posterior samples of state assignments **;
        samples = stateassign(p);
        pavgsum = p[+,];
        pavgsum2 = p[##,];
        pavgmean = pavgsum/mcmcreps;
        pavgsd = sqrt( ( pavgsum2 - (pavgsum # pavgsum)/mcmcreps ) / mcmcreps );
        m = ((samples @ j(1,16,1)) = (j(mcmcreps,1,1) @ (1:16)))[+,] / mcmcreps;
        ** Return MCMC posterior samples of state assignment **;
        y = shape(pavgmean,1,4) || shape(pavgsd,1,4) || m;
        put "MCMC step " phase 1.0 " of 1 finished (&unit, &year)" /;
      finish;


      ** Read data and create input matrices **;
      use b;
      read all var{sb kb np rm} into counts;

      ** MCMC execution **;
      file log;
      m1 = j(1,24,0);
      s1 = j(&mcmcreps,1,0);
      ** Call MCMC procedure for each year **;
      call mcmc(counts[,1:4],&mcmcreps,&tunereps,&rseed,1,m1,s1);
      free s1;
      x = (&i) || m1;

      ** Output results to dataset **;
      names = 'unityear' || ('mean1':'mean4') || ('sd1':'sd4') || ('sa1':'sa16');
      names = 'unityear' || 'sb' || 'kb' || 'np' || 'rm' ||
                      'sb_sd' || 'kb_sd' || 'np_sd' || 'rm_sd' ||
                      '_30SB' || '_30CO' || '_30KB' || '_30RM' ||
                      '_45SB' || '_45CO' || '_45KB' || '_45RM' ||
                      '_60SB' || '_60CO' || '_60KB' || '_60RM' ||
                      '_100SB' || '_100CO' || '_100KB' || '_100RM';
      create mcmcadd var names;
      append from x;

      quit;

    ** Append the results for current unit to results for previous units **;
    proc datasets library=work nolist nowarn;
      append base=mcmc data=mcmcadd;
      run;

  %end;

  ** Write temporary datasets to permanent SAS databases **;
  data results.mcmc;  set mcmc;  run;

%mend;
```

```
%mcmc

data all_float;
  merge unityrs results.mcmc;
  by unityear;
  run;

proc print data=all_float;  run;

quit;


data model.all_float;
		set work.all_float;
run;
```

```
options ps=120;

libname model "C:\Northern Prairie Project\Models\StatisticalModels_ProcReg\SAS\mixed\Thru2016";

data a;
        set model.all_float;
        NPt1=log(NP/(1-NP)); /* Logit of NP */
        SBt1=log(SB/NP); /* Response variables, SBt1, KBt1, and RMt1 are logits based on NP as the 'pivot'. */
        KBt1=log(KB/NP);
        RMt1=log(RM/NP);
run;

**Create NP, SB, KB, and RM at time t by looking back at the NP, SB, KB, and RM (t+1) from previous year for each unit**;
data a;
        set a;
        by unit year;
        if first.unityear then NPt=.; else NPt=lag(NPt1);
        if first.unityear then SBt=.; else SBt=lag(SBt1);
        if first.unityear then KBt=.; else KBt=lag(KBt1);
        if first.unityear then RMt=.; else RMt=lag(RMt1);
run;

data a;
        set a;
        by unit;
        if first.unit then NPt=.;
        if first.unit then SBt=.;
        if first.unit then KBt=.;
        if first.unit then RMt=.;
run;

**Determine predictor variable for dominance based on point estimates and our rule of dominance**;

data a;
        set a;
        sbkb = sb+kb;           ** Sum of SB & KB only **;
        other = sbkb+rm;                        ** Sum of SB, KB, and RM **;
do;
        if rm/other>=(2/3) then domt1 = "RM";           ** dominance by RM **;
        else
        do;
                if sb/sbkb>=(2/3) then domt1 = "SB";            ** dominance by SB **;
                else if kb/sbkb>=(2/3) then domt1 = "KB";    ** dominance by KB **;
                else domt1 = "Co";                                                      ** SB|KB co-dominance**;
        end;
        end;
run;

**Drop unneccesary fields**;
 data a;
        set a;
        drop sbkb other;
run;

**Rename index to index(t+1)**;
data a;
        set a;
```

```
		rename index=indext1;
run;

** Create DOM and INDEX at time t by looking back at the dom and index from previous year for each unit**;
data a;
		set a;
		by unit year;
		if first.unityear then domt="  "; else domt=lag(domt1);
		if first.unityear then indext=.; else indext=lag(indext1);
run;

data a;
		set a;
		by unit;
		if first.unit then domt="  ";
		if first.unit then indext=.;
run;


**Predictor variables are:  SBt, KBt, RMt, NPt, trtmnt, domt, indext**;

**Response variabe is SBt1, KBt1, RMt1**;

**Drop unit.year where trtmnt is not classified as one of the four acceptable treatments**;

data a2;
		set a;
		if trtmnt="Hay" then delete;
		if trtmnt="Unclassifiable" then delete;
run;

data model.mixed_float;
		set a2;
run;

data m;
		set model.mixed_float;
		if trtmnt="Burn/Graze" then trtfmt="aBurn/Graze";
		if trtmnt="Burn" then trtfmt="bBurn";
		if trtmnt="Graze" then trtfmt="cGraze";
		if trtmnt="Rest" then trtfmt="dRest";
		if domt="SB" then domtfmt="aSB";
		if domt="Co" then domtfmt="bCo";
		if domt="KB" then domtfmt="cKB";
		if domt="RM" then domtfmt="dRM";
run;

***checking frequency of trtmnt and domt combinations**;
proc freq data=m;
		tables trtfmt*domtfmt;
		where NPt ne .;
run;

***Global Model***;

proc glm data=model.mixed_float;
		class trtmnt domt;
```

```
		model SBt1=SBt trtmnt domt indext trtmnt*domt trtmnt*indext domt*indext trtmnt*domt*indext npt npt*trtmnt/noint ss3
solution;
		lsmeans trtmnt*domt;
run;
quit;

proc glm data=model.mixed_float;
		class trtmnt domt;
		model KBt1=KBt trtmnt domt indext trtmnt*domt trtmnt*indext domt*indext trtmnt*domt*indext npt npt*trtmnt/noint ss3
solution;
		lsmeans trtmnt*domt;
run;
quit;

proc glm data=model.mixed_float;
		class trtmnt domt;
		model RMt1=RMt trtmnt domt indext trtmnt*domt trtmnt*indext domt*indext trtmnt*domt*indext npt npt*trtmnt/noint ss3
solution;
		lsmeans trtmnt*domt;
run;
quit;

***checking frequency of trtmnt and domt combinations because the df in the model is only 6, as opposed to the possible 9**;
proc freq data=model.mixed_float;
		tables trtmnt*domt;
		where SBt ne .;
run;

***Model 1 - Generalized***;
proc glm data=model.mixed_float;
		class trtmnt;
		model SBt1=SBt trtmnt/ noint ss3 solution;
		contrast 'Active trt equal'
		   trtmnt 1 -1 0 0,
	trtmnt 1 0 -1 0;
		estimate 'Active Avg' trtmnt 1 1 1 0/divisor=3;
run;
quit;

proc glm data=model.mixed_float;
		class trtmnt;
		model KBt1=KBt trtmnt/ noint ss3 solution;
		contrast 'Active trt equal'
		   trtmnt 1 -1 0 0,
	trtmnt 1 0 -1 0;
		estimate 'Active Avg' trtmnt 1 1 1 0/divisor=3;
run;
quit;

proc glm data=model.mixed_float;
		class trtmnt;
		model RMt1=RMt trtmnt/noint ss3 solution;
		contrast 'Active trt equal'
		   trtmnt 1 -1 0 0,
	trtmnt 1 0 -1 0;
		estimate 'Active Avg' trtmnt 1 1 1 0/divisor=3;
run;
```

```
quit;

***Model 1 - Active v. Rest.  Renamed all active trtmnts as "Active" to comply with the constraint of M1 that BG=B=G.
Do not know how to set constraints for inequalities.  Therefore, the constraint regarding Active treatments relative to
Rest (that is, according to M1 Active < Rest) is not set.  Also the constraint regarding Rest that Rest > 0 is not set.***;

***Create new variable M1trt and name all active trtmnts "Active"**;

data m1;
        set model.mixed_float;
        if trtmnt="Burn/Graze" then M1trt="Active";
        if trtmnt="Burn" then M1trt="Active";
        if trtmnt="Graze" then M1trt="Active";
        if trtmnt="Rest" then M1trt="Rest";
run;

proc glm data=m1;
        class M1trt;
        model SBt1=SBt M1trt/ noint ss3 solution;
        contrast 'Active equal Rest'
          M1trt 1 -1;
run;
quit;

proc glm data=m1;
        class M1trt;
        model KBt1=KBt M1trt/ noint ss3 solution;
        contrast 'Active equal Rest'
          M1trt 1 -1;
run;
quit;

proc glm data=m1;
        class M1trt;
        model RMt1=RMt M1trt/ noint ss3 solution;
        contrast 'Active equal Rest'
          M1trt 1 -1;
run;
quit;




***Model 2 - Generalized (i.e., no imposed constraints)***;

***Change naming of trtmnt and domt so reference level is Rest and RM***;
data m2;
        set model.mixed_float;
        if trtmnt="Burn/Graze" then trtfmt="aBurn/Graze";
        if trtmnt="Burn" then trtfmt="bBurn";
        if trtmnt="Graze" then trtfmt="cGraze";
        if trtmnt="Rest" then trtfmt="dRest";
        if domt="SB" then domtfmt="aSB";
        if domt="Co" then domtfmt="bCo";
        if domt="KB" then domtfmt="cKB";
        if domt="RM" then domtfmt="dRM";
run;
```

```
proc glm data=m2;
		class trtfmt domtfmt;
		model SBt1=SBt trtfmt domtfmt trtfmt*domtfmt/ noint ss3 solution;
run;
quit;

proc glm data=m2;
		class trtfmt domtfmt;
		model KBt1=KBt trtfmt domtfmt trtfmt*domtfmt/ noint ss3 solution;
run;
quit;

proc glm data=m2;
		class trtfmt domtfmt;
		model RMt1=RMt trtfmt domtfmt trtfmt*domtfmt/ noint ss3 solution;
run;
quit;

***checking frequency of trtmnt and domt combinations**;
proc freq data=m2;
		tables trtfmt*domtfmt;
		where SBt ne .;
run;

***Model 2. Do not know how to set constraints for inequalities.  Therefore, only setting constraints for the
equalities and leaving everything else. Don't know how to set equaility contraints either.  Decided to go with
Primary, Secondary, and Rest (see below).***;

***Create new variable M2trt and name **;

data m2;
		set model.mixed_float;
		if domt="SB" and trtmnt="Burn/Graze" then M2trt="aPrimary";
		if domt="SB" and trtmnt="Graze" then M2trt="aPrimary";
		if domt="SB" and trtmnt="Burn" then M2trt="bSecondary";
		if domt="KB" and trtmnt="Burn/Graze" then M2trt="aPrimary";
		if domt="KB" and trtmnt="Burn" then M2trt="aPrimary";
		if domt="KB" and trtmnt="Graze" then M2trt="bSecondary";
		if domt="Co" and trtmnt="Burn/Graze" then M2trt="aPrimary";
		if domt="Co" and trtmnt="Burn" then M2trt="bSecondary";
		if domt="Co" and trtmnt="Graze" then M2trt="bSecondary";
		if domt="RM" and trtmnt="Burn/Graze" then M2trt="aPrimary";
		if domt="RM" and trtmnt="Burn" then M2trt="aPrimary";
		if domt="RM" and trtmnt="Graze" then M2trt="aPrimary";
		if trtmnt="Rest" then M2trt="cRest";
		if domt="SB" then domtfmt="aSB";
		if domt="Co" then domtfmt="bCo";
		if domt="KB" then domtfmt="cKB";
		if domt="RM" then domtfmt="dRM";
run;


proc glm data=m2;
		class M2trt domtfmt;
		model SBt1=SBt M2trt domtfmt M2trt*domtfmt/ noint ss3 solution;
run;
```

```
quit;

proc glm data=m2;
	class M2trt domtfmt;
	model KBt1=KBt M2trt domtfmt M2trt*domtfmt/ noint ss3 solution;
run;
quit;

proc glm data=m2;
	class M2trt domtfmt;
	model RMt1=RMt M2trt domtfmt M2trt*domtfmt/ noint ss3 solution;
run;
quit;

***checking frequency of trtmnt and domt combinations**;
proc freq data=m2;
	tables M2trt*domtfmt;
	where SBt ne .;
run;


***Model 3 - Generalized (i.e., no imposed constraints)***;

***Change naming of trtmnt and domt so reference level is Rest and RM***;
data m3;
	set model.mixed_float;
	if trtmnt="Burn/Graze" then trtfmt="aBurn/Graze";
	if trtmnt="Burn" then trtfmt="bBurn";
	if trtmnt="Graze" then trtfmt="cGraze";
	if trtmnt="Rest" then trtfmt="dRest";
	if domt="SB" then domtfmt="aSB";
	if domt="Co" then domtfmt="bCo";
	if domt="KB" then domtfmt="cKB";
	if domt="RM" then domtfmt="dRM";
run;

proc glm data=m3;
	class trtfmt domtfmt;
	model SBt1=SBt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext/ noint ss3
solution;
run;
quit;

proc glm data=m3;
	class trtfmt domtfmt;
	model KBt1=KBt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext/ noint ss3
solution;
run;
quit;

proc glm data=m3;
	class trtfmt domtfmt;
	model RMt1=RMt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext/ noint ss3
solution;
run;
quit;
```

```
***checking frequency of trtmnt and domt combinations**;
proc freq data=m3;
        tables trtfmt*domtfmt trtfmt*domtfmt*indext;
        where SBt ne .;
run;


***Model 4 - Generalized (i.e., no imposed constraints)***;

***Change naming of trtmnt and domt so reference level is Rest and RM***;
data m4;
        set model.mixed_float;
        if trtmnt="Burn/Graze" then trtfmt="aBurn/Graze";
        if trtmnt="Burn" then trtfmt="bBurn";
        if trtmnt="Graze" then trtfmt="cGraze";
        if trtmnt="Rest" then trtfmt="dRest";
        if domt="SB" then domtfmt="aSB";
        if domt="Co" then domtfmt="bCo";
        if domt="KB" then domtfmt="cKB";
        if domt="RM" then domtfmt="dRM";
run;

proc glm data=m4;
        class trtfmt domtfmt;
        model SBt1=SBt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext npt npt*trtfmt/
noint ss3 solution;
run;
quit;

proc glm data=m4;
        class trtfmt domtfmt;
        model KBt1=KBt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext npt npt*trtfmt/
noint ss3 solution;
run;
quit;

proc glm data=m4;
        class trtfmt domtfmt;
        model RMt1=RMt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext npt npt*trtfmt/
noint ss3 solution;
run;
quit;
```

```
options ps=60;

libname model "C:\Northern Prairie Project\Models\StatisticalModels_ProcReg\SAS\tall\Thru2016";

data a;
		set model.all_float;
		NPt1=log(NP/(1-NP)); /* Logit of NP */
		SBt1=log(SB/NP); /* Response variables, SBt1, KBt1, and RMt1 are logits based on NP as the 'pivot'. */
		KBt1=log(KB/NP);
		RMt1=log(RM/NP);
run;

**Create NP, SB, KB, and RM at time t by looking back at the NP, SB, KB, and RM (t+1) from previous year for each unit**;
data a;
		set a;
		by unit year;
		if first.unityear then NPt=.; else NPt=lag(NPt1);
		if first.unityear then SBt=.; else SBt=lag(SBt1);
		if first.unityear then KBt=.; else KBt=lag(KBt1);
		if first.unityear then RMt=.; else RMt=lag(RMt1);
run;

data a;
		set a;
		by unit;
		if first.unit then NPt=.;
		if first.unit then SBt=.;
		if first.unit then KBt=.;
		if first.unit then RMt=.;
run;

**Determine predictor variable for dominance based on point estimates and our rule of dominance**;

data a;
		set a;
		sbkb = sb+kb;			** Sum of SB & KB only **;
		other = sbkb+rm;					** Sum of SB, KB, and RM **;
do;
		if rm/other>=(2/3) then domt1 = "RM";			** dominance by RM **;
		else
		do;
				if sb/sbkb>=(2/3) then domt1 = "SB";			** dominance by SB **;
				else if kb/sbkb>=(2/3) then domt1 = "KB";	** dominance by KB **;
				else domt1 = "Co";								** SB|KB co-dominance**;
		end;
		end;
run;

**Drop unneccesary fields**;
 data a;
		set a;
		drop sbkb other;
run;

**Rename index to index(t+1)**;
data a;
		set a;
```

```
            rename index=indext1;
run;

** Create DOM and INDEX at time t by looking back at the dom and index from previous year for each unit**;
data a;
            set a;
            by unit year;
            if first.unityear then domt="  "; else domt=lag(domt1);
            if first.unityear then indext=.; else indext=lag(indext1);
run;

data a;
            set a;
            by unit;
            if first.unit then domt="  ";
            if first.unit then indext=.;
run;

**Predictor variables are:  vector NPt, SBt, KBt, RMt, trtmnt, dom, index**;

**Response variables are SBt1, KBt1, RMt1 **;


***Look at the "unclassifiable" trtmnts to see if they are usable***;
data un;
            set a;
            where trtmnt="Unclassifiable";
run;

data a2;
            set a;
            if trtmnt="Unclassifiable" then delete;
run;

data model.tall_float;
            set a2;
run;
/*
data t;
            set model.tall_float;
            if NPt=. then delete;
run;

***Global Model***;
***Change naming of trtmnt and dom so reference level is Rest and RM***;
data mg;
            set t;
            if trtmnt="Burn w/in window" then trtfmt="aBurn w/in window";
            if trtmnt="Graze w/in window" then trtfmt="bGraze w/in window";
            if trtmnt="Defoliate" then trtfmt="cDefoliate";
            if trtmnt="Rest" then trtfmt="dRest";
            if domt="SB" then domtfmt="aSB";
            if domt="Co" then domtfmt="bCo";
            if domt="KB" then domtfmt="cKB";
            if domt="RM" then domtfmt="dRM";
run;
```

```
proc glm data=mg;
		class trtfmt domtfmt;
		model SBt1=SBt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext NPt
NPt*trtfmt/noint ss3 solution;
run;
quit;

proc glm data=mg;
		class trtfmt domtfmt;
		model KBt1=KBt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext NPt
NPt*trtfmt/noint ss3 solution;
run;
quit;

proc glm data=mg;
		class trtfmt domtfmt;
		model RMt1=RMt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext NPt
NPt*trtfmt/noint ss3 solution;
run;
quit;

***checking frequency of trtmnt and dom combinations because the df in the model is only 6, as opposed to the possible 9**;
proc freq data=mg;
		tables trtfmt*domtfmt;
		where SBt ne .;
run;

***Model 1 - Generalized***;

***Change naming of trtmnt and dom so reference level is Rest and RM***;
data m1;
		set model.tall;
		if trtmnt="Burn w/in window" then trtfmt="aBurn w/in window";
		if trtmnt="Graze w/in window" then trtfmt="bGraze w/in window";
		if trtmnt="Defoliate" then trtfmt="cDefoliate";
		if trtmnt="Rest" then trtfmt="dRest";
		if domt="SB" then domtfmt="aSB";
		if domt="Co" then domtfmt="bCo";
		if domt="KB" then domtfmt="cKB";
		if domt="RM" then domtfmt="dRM";
run;

proc glm data=m1;
		class trtfmt;
		model SBt1=SBt trtfmt/ noint ss3 solution;
		contrast 'Active trt equal'
		  trtfmt 1 -1 0 0,
	trtfmt 1 0 -1 0;
		estimate 'Active Avg' trtfmt 1 1 1 0/divisor=3;
run;
quit;

proc glm data=m1;
		class trtfmt;
		model KBt1=KBt trtfmt/ noint ss3 solution;
		contrast 'Active trt equal'
		  trtfmt 1 -1 0 0,
```

```
      trtfmt 1 0 -1 0;
              estimate 'Active Avg' trtfmt 1 1 1 0/divisor=3;
run;
quit;

proc glm data=m1;
              class trtfmt;
              model RMt1=RMt trtfmt/noint ss3 solution;
              contrast 'Active trt equal'
                 trtfmt 1 -1 0 0,
      trtfmt 1 0 -1 0;
              estimate 'Active Avg' trtfmt 1 1 1 0/divisor=3;
run;
quit;

***Model 1 - Active v. Rest.  Renamed all active trtmnts as "Active" to comply with the constraint of M1 that BG=B=G.
Do not know how to set constraints for inequalities.  Therefore, the constraint regarding Active treatments relative to
Rest (that is, according to M1 Active < Rest) is not set.  Also the constraint regarding Rest that Rest > 0 is not set.***;

***Create new variable M1trt and name all active trtmnts "Active"**;

data m1;
              set model.tall;
              if trtmnt="Burn w/in window" then M1trt="Active";
              if trtmnt="Graze w/in window" then M1trt="Active";
              if trtmnt="Defoliate" then M1trt="Active";
              if trtmnt="Rest" then M1trt="Rest";
run;

proc glm data=m1;
              class M1trt;
              model SBt1=SBt M1trt/ noint ss3 solution;
              contrast 'Active equal Rest'
                 M1trt 1 -1;
run;
quit;

proc glm data=m1;
              class M1trt;
              model KBt1=KBt M1trt/ noint ss3 solution;
              contrast 'Active equal Rest'
                 M1trt 1 -1;
run;
quit;

proc glm data=m1;
              class M1trt;
              model RMt1=RMt M1trt/ noint ss3 solution;
              contrast 'Active equal Rest'
                 M1trt 1 -1;
run;
quit;


***Model 2 - Generalized (i.e., no imposed constraints)***;

***Change naming of trtmnt and domt so reference level is Rest and RM***;
```

```
data m2;
		set model.tall_float;
		if trtmnt="Burn w/in window" then trtfmt="aBurn w/in window";
		if trtmnt="Graze w/in window" then trtfmt="bGraze w/in window";
		if trtmnt="Defoliate" then trtfmt="cDefoliate";
		if trtmnt="Rest" then trtfmt="dRest";
		if domt="SB" then domtfmt="aSB";
		if domt="Co" then domtfmt="bCo";
		if domt="KB" then domtfmt="cKB";
		if domt="RM" then domtfmt="dRM";
run;

proc glm data=m2;
		class trtfmt domtfmt;
		model SBt1=SBt trtfmt domtfmt trtfmt*domtfmt/ noint ss3 solution;
run;
quit;

proc glm data=m2;
		class trtfmt domtfmt;
		model KBt1=KBt trtfmt domtfmt trtfmt*domtfmt/ noint ss3 solution;
run;
quit;

proc glm data=m2;
		class trtfmt domtfmt;
		model RMt1=RMt trtfmt domtfmt trtfmt*domtfmt/ noint ss3 solution;
run;
quit;

***checking frequency of trtmnt and domt combinations**;
proc freq data=m2;
		tables trtfmt*domtfmt;
		where SBt ne .;
run;

***Model 3 - Generalized (i.e., no imposed constraints)***;

***Change naming of trtmnt and domt so reference level is Rest and RM***;
data m3;
		set model.tall_float;
		if trtmnt="Burn w/in window" then trtfmt="aBurn w/in window";
		if trtmnt="Graze w/in window" then trtfmt="bGraze w/in window";
		if trtmnt="Defoliate" then trtfmt="cDefoliate";
		if trtmnt="Rest" then trtfmt="dRest";
		if domt="SB" then domtfmt="aSB";
		if domt="Co" then domtfmt="bCo";
		if domt="KB" then domtfmt="cKB";
		if domt="RM" then domtfmt="dRM";
run;

proc glm data=m3;
		class trtfmt domtfmt;
		model SBt1=SBt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext/ noint ss3
solution;
run;
quit;
```

```
proc glm data=m3;
	class trtfmt domtfmt;
	model KBt1=KBt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext/ noint ss3
solution;
run;
quit;

proc glm data=m3;
	class trtfmt domtfmt;
	model RMt1=RMt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext/ noint ss3
solution;
run;
quit;

***checking frequency of trtmnt and domt combinations**;
proc freq data=m3;
	tables trtfmt*domtfmt trtfmt*domtfmt*indext;
	where SBt ne .;
run;

***Model 4 - Generalized (i.e., no imposed constraints)***;

***Change naming of trtmnt and domt so reference level is Rest and RM***;
data m4;
	set model.tall_float;
	if trtmnt="Burn w/in window" then trtfmt="aBurn w/in window";
	if trtmnt="Graze w/in window" then trtfmt="bGraze w/in window";
	if trtmnt="Defoliate" then trtfmt="cDefoliate";
	if trtmnt="Rest" then trtfmt="dRest";
	if domt="SB" then domtfmt="aSB";
	if domt="Co" then domtfmt="bCo";
	if domt="KB" then domtfmt="cKB";
	if domt="RM" then domtfmt="dRM";
run;

proc glm data=m4;
	class trtfmt domtfmt;
	model SBt1=SBt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext NPt NPt*trtfmt/
noint ss3 solution;
run;
quit;

proc glm data=m4;
	class trtfmt domtfmt;
	model KBt1=KBt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext NPt NPt*trtfmt/
noint ss3 solution;
run;
quit;

proc glm data=m4;
	class trtfmt domtfmt;
	model RMt1=RMt trtfmt domtfmt indext trtfmt*domtfmt trtfmt*indext domtfmt*indext trtfmt*domtfmt*indext NPt NPt*trtfmt/
noint ss3 solution;
run;
quit;
```

```
options ps=120;

libname model "C:\Northern Prairie Project\Models\StatisticalModels_ProcReg\SAS\mixed\Thru2016";

Data mixed;
  set model.mixed_float;
  if NPt=. then delete;
run;


***checking frequency of trtmnt and domt combinations**;
proc freq data=mixed;
          tables trtmnt*domt;
run;


***checking frequency of trtmnt and domt and index level combinations**;
data a;
          set mixed;
          i="M";
          if indext<2 then i="L";
          if indext>4 then i="H";
run;

proc freq data=a;
          tables i*trtmnt*domt;
run;


/* The above data are ill-conditioned in the following ways:
                    1. There is only 1 unit wiht Trtment=B and Dom=RM which means we can't estimate a slope for that combo

  Because of the above condition, the "usual" hypotheses in an ANCOVA like this can not be tested.  As a result, GLM is testing
  non-sensical hypotheses in place of the usual ones.  I verified this by requesting that GLM spit out the Type III estimable functions
  (e3 option on the model statement).  I didn't take the time to figure out exactly what those hypotheses were but I looked closely
  enough to verify that they are rubbish.  From a practical sense, this means we should not put much stock in the Type III tests
  that are coming out of GLM.  If we need to test hypotheses, we should construct our own via the Test statement in PROC REG.
That way
  we will know what the hypotheses are.
*/


/* At this point I am ready to code the full-rank design matrix for use with PROC REG */

data m;
 set mixed;
  if NPt=. then delete;
  /* In its most general form, model 4 has 16 interecepts or constant terms, one for each trtmnt*dom group. */
  /* Therefore, I assign 16 indicator variables to identify the treatment*dom class. */
  /* td11 corresponds to row 1, column 1 in the 4x4 schematics in Jill's spreadsheet */
  /* td12 is row 1, column2; td13 is row 1, col 3, etc. */

  Array td {16} td11-td14 td21-td24 td31-td34 td41-td44;
  do i = 1 to 16; td[i]=0; end;  /* initialize all tds to zero */
  if trtmnt='Burn/Graze' then do;
    if domt='SB' then td11=1;
    if domt='Co' then td12=1;
```

```
    if domt='KB' then td13=1;
    if domt='RM' then td14=1;
  end;
  if trtmnt='Burn' then do;
    if domt='SB' then td21=1;
    if domt='Co' then td22=1;
    if domt='KB' then td23=1;
    if domt='RM' then td24=1;
  end;
  if trtmnt='Graze' then do;
    if domt='SB' then td31=1;
    if domt='Co' then td32=1;
    if domt='KB' then td33=1;
    if domt='RM' then td34=1;
  end;
  if trtmnt='Rest' then do;
    if domt='SB' then td41=1;
    if domt='Co' then td42=1;
    if domt='KB' then td43=1;
    if domt='RM' then td44=1;
  end;

  /* In its most general form, model 4 has 16 distinct Index slopes.  Assign those variables here. */
  Array tdi {16} tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44;
  do i = 1 to 16;
    tdi[i]=td[i]*indext;
  end;

  /* In its most general form, model 4 has 4 distinct slopes associated with np.  Assign those variables here. */
  if trtmnt='Burn/Graze' then np1 = npt; else np1=0;
  if trtmnt='Burn' then np2 = npt; else np2=0;
  if trtmnt='Graze' then np3 = npt; else np3=0;
  if trtmnt='Rest' then np4 = npt; else np4=0;
run;
/* The design matrix is now complete */

/* Fit Model 1: Impose constraints to fit the general model concept */
/* Model 1 Concept:  All treatments are equally effective, regardless of dominant invader, defoliation history, or invasion level.
No dependence on veg state or defoliation state. */

proc reg data=m outest=Mf_M1SBrs covout;
 M1_SBrs: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 1 */
/* The design matrix is already full rank for Model 1 and does not require the constraints of Models 2-4 to make full rank */
 Restrict td11=td12=td13=td14=td21=td22=td23=td24=td31=td32=td33=td34; /* Effects are the same across all four dominance
types and three active treatments - BG, Burn, Graze */
 Restrict td41=td42=td43=td44; /* Effect of Rest is the same across all four dominance types */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

run;
quit;
```

```
proc reg data=m outest=Mf_M1KBrs covout;
 M1_KBrs: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 1 */
/* The design matrix is already full rank for Model 1 and does not require the constraints of Models 2-4 to make full rank */
 Restrict td11=td12=td13=td14=td21=td22=td23=td24=td31=td32=td33=td34; /* Effects are the same across all four dominance types and three active treatments - BG, Burn, Graze */
 Restrict td41=td42=td43=td44; /* Effect of Rest is the same across all four dominance types */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

run;
quit;

proc reg data=m outest=Mf_M1RMrs covout;
 M1_RMrs: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 1 */
/* The design matrix is already full rank for Model 1 and does not require the constraints of Models 2-4 to make full rank */
 Restrict td11=td12=td13=td14=td21=td22=td23=td24=td31=td32=td33=td34; /* Effects are the same across all four dominance types and three active treatments - BG, Burn, Graze */
 Restrict td41=td42=td43=td44; /* Effect of Rest is the same across all four dominance types */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

run;
quit;

/* Fit Model 2: Impose constraints to fit the general model concept, as well as the model specifics (i.e. set response-specific constraints on td). */
/* Model 2 concept:  Treatment effectiveness is state-dependent and depends on the type of invader that is dominant. */

proc reg data=m outest=Mf_M2SBrs covout;
 M2_SBrs: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 2 */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;
/* Set the following equality constraints specific to Model 2 */
 Restrict td11=td31; /* If SB dom, BG=G */
 Restrict td22=td32; /* If CO dom, B=G */
 *Restrict td13=td23; /* If KB dom, BG=B */
 *Restrict td14=td24=td34; /* If RM dom, BG=B=G */
```

```
run;
quit;

proc reg data=m outest=Mf_M2KBrs covout;
 M2_KBrs: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* Set constraints for Model 2 */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;
/* Set the following equality constraints specific to Model 2 */
 *Restrict td11=td31; /* If SB dom, BG=G */
 Restrict td22=td32; /* If CO dom, B=G */
 Restrict td13=td23; /* If KB dom, BG=B */
 *Restrict td14=td24=td34; /* If RM dom, BG=B=G */

run;
quit;


proc reg data=m outest=Mf_M2RMrs covout;
 M2_RMrs: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 2 */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;
/* Set the following equality constraints specific to Model 2 */
 *Restrict td11=td31; /* If SB dom, BG=G */
 *Restrict td22=td32; /* If CO dom, B=G */
 *Restrict td13=td23; /* If KB dom, BG=B */
 Restrict td14=td24=td34; /* If RM dom, BG=B=G */

run;
quit;

/* Fit Model 3: Impose constraints to fit the general model concept, as well as response-specif constaints. */
/* Model 3 concept:  Treatment effectiveness is state-dependent and depends on the type of invader that is dominant, and on the
defoliation history. */

proc reg data=m outest=Mf_M3SBrs covout;
 M3_SBrs: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 3 */
/* Set the following constraints to make the data set full rank */
  Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23;
/* Remove to use restiction below */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints specific to Model 3 */
```

```
 Restrict tdi11=tdi21=tdi31; /*BGxSBxindex=BurnxSBxindex=GrazexSBxindex, but RestxSBxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB, CO, KB, and RM are allowed to be different.*/
 Restrict tdi12=tdi22=tdi32; /*BGxCOxindex=BurnxCOxindex=GrazexCOxindex, but RestxCOxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for CO, while slope for Rest is left unconstrained.*/
 *Restrict tdi13=tdi23=tdi33; /*BGxKBxindex=BurnxKBxindex=GrazexKBxindex, but RestxKBxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for KB, while slope for Rest is left unconstrained.*/
 *Restrict tdi14=tdi24=tdi34; /*BGxRMxindex=BurnxRMxindex=GrazexRMxindex, but RestxRMxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for RM, while slope for Rest is left unconstrained.*/

run;
quit;

 proc reg data=m outest=Mf_M3KBrs covout;
 M3_KBrs: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

 /*Set constraints for Model 3 */
 /* Set the following constraints to make the data set full rank */
  Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23;
/* Remove to use restiction below */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints specific to Model 3 */
 *Restrict tdi11=tdi21=tdi31; /*BGxSBxindex=BurnxSBxindex=GrazexSBxindex, but RestxSBxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB, CO, KB, and RM are allowed to be different.*/
 Restrict tdi12=tdi22=tdi32; /*BGxCOxindex=BurnxCOxindex=GrazexCOxindex, but RestxCOxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for CO, while slope for Rest is left unconstrained.*/
 Restrict tdi13=tdi23=tdi33; /*BGxKBxindex=BurnxKBxindex=GrazexKBxindex, but RestxKBxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for KB, while slope for Rest is left unconstrained.*/
 *Restrict tdi14=tdi24=tdi34; /*BGxRMxindex=BurnxRMxindex=GrazexRMxindex, but RestxRMxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for RM, while slope for Rest is left unconstrained.*/

 run;
 quit;

 proc reg data=m outest=Mf_M3RMrs covout;
 M3_RMrs: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 3 */
/* Set the following constraints to make the data set full rank */
  *Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23;
/* Remove to use restiction below */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints specific to Model 3 */
 *Restrict tdi11=tdi21=tdi31; /*BGxSBxindex=BurnxSBxindex=GrazexSBxindex, but RestxSBxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB, CO, KB, and RM are allowed to be different.*/
 *Restrict tdi12=tdi22=tdi32; /*BGxCOxindex=BurnxCOxindex=GrazexCOxindex, but RestxCOxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for CO, while slope for Rest is left unconstrained.*/
 *Restrict tdi13=tdi23=tdi33; /*BGxKBxindex=BurnxKBxindex=GrazexKBxindex, but RestxKBxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for KB, while slope for Rest is left unconstrained.*/
 Restrict tdi14=tdi24=tdi34; /*BGxRMxindex=BurnxRMxindex=GrazexRMxindex, but RestxRMxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for RM, while slope for Rest is left unconstrained.*/

run;
```

```
quit;

/* Fit Model 4: Impose constraints to fit the general model concept, as well as response-specific contraints. */
/* Model 3 concept:  Treatment effectiveness is state-dependent and depends on the type of invader that is dominant, and on the
defoliation history. */

proc reg data=m outest=Mf_M4SBrs covout;
 M4_SBrs: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* The following constraints make the design matrix full rank */
/* These are based on an assumption of no interaction */
  Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23;
/* Remove to use restiction below */
/* Set same equality constraints for tdi as used for Model 3 */
  Restrict tdi11=tdi21=tdi31; /*BGxSBxindex=BurnxSBxindex=GrazexSBxindex, but RestxSBxindex left unconstrained. Slope for
active treatments - BG, Burn, and Graze - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB,
CO, KB, and RM are allowed to be different.*/
  Restrict tdi12=tdi22=tdi32; /*BGxCOxindex=BurnxCOxindex=GrazexCOxindex, but RestxCOxindex left unconstrained. Slope for
active treatments - BG, Burn, and Graze - are set the same for CO, while slope for Rest is left unconstrained.*/
  *Restrict tdi13=tdi23=tdi33; /*BGxKBxindex=BurnxKBxindex=GrazexKBxindex, but RestxKBxindex left unconstrained. Slope for
active treatments - BG, Burn, and Graze - are set the same for KB, while slope for Rest is left unconstrained.*/
  *Restrict tdi14=tdi24=tdi34; /*BGxRMxindex=BurnxRMxindex=GrazexRMxindex, but RestxRMxindex left unconstrained. Slope for
active treatments - BG, Burn, and Graze - are set the same for RM, while slope for Rest is left unconstrained.*/
/* Set equality constraints for np*trtmnt specific to Model 4.  There are none. */
  *Restrict np1=np2=np3; /* playing with idea of setting the change in effect of BG, Burn, and Graze with chang in NP to equal */
  *Restrict np4=0; /* playing with idea of the change in effect of Rest with change in NP being zero */

 run;
 quit;

proc reg data=m outest=Mf_M4KBrs covout;
 M4_KBrs: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* The following constraints make the design matrix full rank */
/* These are based on an assumption of no interaction */
  Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23;
/* Remove to use restiction below */
/* Set same equality constraints for tdi as used for Model 3 */
  *Restrict tdi11=tdi21=tdi31; /*BGxSBxindex=BurnxSBxindex=GrazexSBxindex, but RestxSBxindex left unconstrained. Slope for
active treatments - BG, Burn, and Graze - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB,
CO, KB, and RM are allowed to be different.*/
  Restrict tdi12=tdi22=tdi32; /*BGxCOxindex=BurnxCOxindex=GrazexCOxindex, but RestxCOxindex left unconstrained. Slope for
active treatments - BG, Burn, and Graze - are set the same for CO, while slope for Rest is left unconstrained.*/
  Restrict tdi13=tdi23=tdi33; /*BGxKBxindex=BurnxKBxindex=GrazexKBxindex, but RestxKBxindex left unconstrained. Slope for
active treatments - BG, Burn, and Graze - are set the same for KB, while slope for Rest is left unconstrained.*/
  *Restrict tdi14=tdi24=tdi34; /*BGxRMxindex=BurnxRMxindex=GrazexRMxindex, but RestxRMxindex left unconstrained. Slope for
active treatments - BG, Burn, and Graze - are set the same for RM, while slope for Rest is left unconstrained.*/
/* Set equality constraints for np*trtmnt specific to Model 4.  There are none. */
  *Restrict np1=np2=np3; /* playing with idea of setting the change in effect of BG, Burn, and Graze with chang in NP to equal */
  *Restrict np4=0; /* playing with idea of the change in effect of Rest with change in NP being zero */

 run;
 quit;

proc reg data=m outest=Mf_M4RMrs covout;
 M4_RMrs: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;
```

```
/* The following constraints make the design matrix full rank */
/* These are based on an assumption of no interaction */
  *Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23;
/* Remove to use restiction below */
/* Set same equality constraints for tdi as used for Model 3 */
  *Restrict tdi11=tdi21=tdi31; /*BGxSBxindex=BurnxSBxindex=GrazexSBxindex, but RestxSBxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB, CO, KB, and RM are allowed to be different.*/
  *Restrict tdi12=tdi22=tdi32; /*BGxCOxindex=BurnxCOxindex=GrazexCOxindex, but RestxCOxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for CO, while slope for Rest is left unconstrained.*/
  *Restrict tdi13=tdi23=tdi33; /*BGxKBxindex=BurnxKBxindex=GrazexKBxindex, but RestxKBxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for KB, while slope for Rest is left unconstrained.*/
  Restrict tdi14=tdi24=tdi34; /*BGxRMxindex=BurnxRMxindex=GrazexRMxindex, but RestxRMxindex left unconstrained. Slope for active treatments - BG, Burn, and Graze - are set the same for RM, while slope for Rest is left unconstrained.*/
/* Set equality constraints for np*trtmnt specific to Model 4. There are none. */
  *Restrict np1=np2=np3; /* playing with idea of setting the change in effect of BG, Burn, and Graze with chang in NP to equal */
  *Restrict np4=0; /* playing with idea of the change in effect of Rest with change in NP being zero */

 run;
 quit;

 ***Append covariance output from all models***;

 proc append base=Mf_m1sbrs data=Mf_m2sbrs;
 run;

 proc append base=Mf_m1sbrs data=Mf_m3sbrs;
 run;

 proc append base=Mf_m1sbrs data=Mf_m4sbrs;
 run;

 proc append base=Mf_m1kbrs data=Mf_m2kbrs;
 run;

 proc append base=Mf_m1kbrs data=Mf_m3kbrs;
 run;

 proc append base=Mf_m1kbrs data=Mf_m4kbrs;
 run;

 proc append base=Mf_m1rmrs data=Mf_m2rmrs;
 run;

 proc append base=Mf_m1rmrs data=Mf_m3rmrs;
 run;

 proc append base=Mf_m1rmrs data=Mf_m4rmrs;
 run;

 libname model2 "C:\Northern Prairie Project\Models\StatisticalModels_ProcReg\ProcReg_Mixed\Thru2016";

***Make the appended output permanent***;
 data model2.Mf_SBrs;
	set Mf_m1sbrs;
run;
```

```
 data model2.Mf_KBrs;
        set Mf_m1kbrs;
run;

 data model2.Mf_RMrs;
        set Mf_m1rmrs;
run;

 *****************************************************************************************************;
/* Fit Model 1 generalized (i.e., no equality constraints) */

proc reg data=m;
 M1Gen_SB: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

 /*Set constraints for Model 1 */
 /* The design matrix is already full rank for Model 1 and does not require the constraints of Models 2-4 to make full rank */
 Restrict td11=td12=td13=td14; /* effect of BG is the same across all four dominance types */
 Restrict td21=td22=td23=td24; /* effect of Burn is the same across all four dominance types */
 Restrict td31=td32=td33=td34; /* effect of Graze is the same across all four dominance types */
 Restrict td41=td42=td43=td44; /* effect of Rest is the same across all four dominance types */
 /* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

 /* Test if active treatments are equal */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
          td11+td12+td13+td14 = td31+td32+td33+td34 / print;
run;
quit;



proc reg data=m;
 M1Gen_KB: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 1 */
/* The design matrix is already full rank for Model 1 and does not require the constraints of Models 2-4 to make full rank */
 Restrict td11=td12=td13=td14; /* effect of BG is the same across all four dominance types */
 Restrict td21=td22=td23=td24; /* effect of Burn is the same across all four dominance types */
 Restrict td31=td32=td33=td34; /* effect of Graze is the same across all four dominance types */
 Restrict td41=td42=td43=td44; /* effect of Rest is the same across all four dominance types */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

/* Test if active treatments are equal */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
          td11+td12+td13+td14 = td31+td32+td33+td34 / print;
run;
quit;
```

```
proc reg data=m;
 M1Gen_RM: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-
np4/noint;

/*Set constraints for Model 1 */
/* The design matrix is already full rank for Model 1 and does not require the constraints of Models 2-4 to make full rank */
 Restrict td11=td12=td13=td14; /* effect of BG is the same across all four dominance types */
 Restrict td21=td22=td23=td24; /* effect of Burn is the same across all four dominance types */
 Restrict td31=td32=td33=td34; /* effect of Graze is the same across all four dominance types */
 Restrict td41=td42=td43=td44; /* effect of Rest is the same across all four dominance types */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

/* Test if active treatments are equal */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
            td11+td12+td13+td14 = td31+td32+td33+td34 / print;
run;
quit;


/* Fit Model 2 generalized (i.e., no equality constraints) */

proc reg data=m;
 M2Gen_SB: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 2 */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

/* coefficient on SBt=0. Row 103 */
SBt:   test SBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 104 */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
            td11+td12+td13+td14 = td31+td32+td33+td34,
                                      td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 105 */
dom:      test td11+td21+td31+td41 = td12+td22+td32+td42,
            td11+td21+td31+td41 = td13+td23+td33+td43,
                                      td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* trt*dom interaction.  Row 106 */
trt_dom:      test td21 - td11 = td22 - td12,
                td21 - td11 = td23 - td13,
                td21 - td11 = td24 - td14,
                                            td31 - td11 = td32 - td12,
                td31 - td11 = td33 - td13,
                                            td31 - td11 = td34 - td14,
```

```
                                        td41 - td11 = td42 - td12,
           td41 - td11 = td43 - td13,
                                        td41 - td11 = td44 - td14 / print;
run;
quit;

proc reg data=m;
 M2Gen_KB: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 2 */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

/* coefficient on SBt=0. Row 133 */
KBt:   test KBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 134 */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
           td11+td12+td13+td14 = td31+td32+td33+td34,
                                   td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 135 */
dom:      test td11+td21+td31+td41 = td12+td22+td32+td42,
           td11+td21+td31+td41 = td13+td23+td33+td43,
                                   td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* trt*dom interaction.  Row 136 */
trt_dom:      test td21 - td11 = td22 - td12,
           td21 - td11 = td23 - td13,
           td21 - td11 = td24 - td14,
                                        td31 - td11 = td32 - td12,
           td31 - td11 = td33 - td13,
                                        td31 - td11 = td34 - td14,
                                        td41 - td11 = td42 - td12,
           td41 - td11 = td43 - td13,
                                        td41 - td11 = td44 - td14 / print;
run;
quit;


proc reg data=m;
 M2Gen_RM: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-
np4/noint;

/*Set constraints for Model 2 */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

/* coefficient on SBt=0. Row 163 */
```

```
RMt:   test RMt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 164 */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
            td11+td12+td13+td14 = td31+td32+td33+td34,
                                          td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 165 */
dom:      test td11+td21+td31+td41 = td12+td22+td32+td42,
            td11+td21+td31+td41 = td13+td23+td33+td43,
                                          td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* trt*dom interaction.  Row 166 */
trt_dom:      test td21 - td11 = td22 - td12,
               td21 - td11 = td23 - td13,
               td21 - td11 = td24 - td14,
                                                      td31 - td11 = td32 - td12,
               td31 - td11 = td33 - td13,
                                                      td31 - td11 = td34 - td14,
                                                      td41 - td11 = td42 - td12,
               td41 - td11 = td43 - td13,
                                                      td41 - td11 = td44 - td14 / print;
run;
quit;



/* Fit Model 3 generalized (i.e., no equality constraints imposed) */

proc reg data=m;
 M3Gen_SB: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

 /*Set constraints for Model 3 */
 /* Set the following constraints to make the data set full rank */
 Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23; /*
Remove to use restiction below */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;

/* coefficient on SBt=0. Row 270 */
SBt:   test SBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 271 */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
            td11+td12+td13+td14 = td31+td32+td33+td34,
                                          td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 272 */
dom:      test td11+td21+td31+td41 = td12+td22+td32+td42,
            td11+td21+td31+td41 = td13+td23+td33+td43,
                                          td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 273 */
indx: test tdi11 + tdi12 + tdi13 + tdi14 +
         tdi21 + tdi22 + tdi23 + tdi24 +
                           tdi31 + tdi32 + tdi33 + tdi34 +
                           tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;
```

```
/* trt*dom interaction.  Row 274 */
trt_dom:      test td21 - td11 = td22 - td12,
              td21 - td11 = td23 - td13,
              td21 - td11 = td24 - td14,
                                        td31 - td11 = td32 - td12,
              td31 - td11 = td33 - td13,
                                        td31 - td11 = td34 - td14,
                                        td41 - td11 = td42 - td12,
              td41 - td11 = td43 - td13,
                                        td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 275 */
indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
           tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                 tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 276 */
indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
           tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                 tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* Index*trt*dom -- many ways to express this 3-way interaction hypothesis. Row 277 */
indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
              tdi21 - tdi11 = tdi23 - tdi13,
              tdi21 - tdi11 = tdi24 - tdi14,
                                        tdi31 - tdi11 = tdi32 - tdi12,
              tdi31 - tdi11 = tdi33 - tdi13,
                                        tdi31 - tdi11 = tdi34 - tdi14,
                                        tdi41 - tdi11 = tdi42 - tdi12,
              tdi41 - tdi11 = tdi43 - tdi13,
                                        tdi41 - tdi11 = tdi44 - tdi14 / print;
run;
quit;


proc reg data=m;
 M3Gen_KB: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

 /*Set constraints for Model 3 */
 /* Set the following constraints to make the data set full rank */
 Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23; /*
Remove to use restiction below */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;

 /* coefficient on SBt=0. Row 321 */
KBt:   test KBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 322 */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
           td11+td12+td13+td14 = td31+td32+td33+td34,
                                 td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 323 */
dom:      test td11+td21+td31+td41 = td12+td22+td32+td42,
           td11+td21+td31+td41 = td13+td23+td33+td43,
```

```
                                  td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 324 */
indx: test tdi11 + tdi12 + tdi13 + tdi14 +
          tdi21 + tdi22 + tdi23 + tdi24 +
                         tdi31 + tdi32 + tdi33 + tdi34 +
                         tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;

/* trt*dom interaction.  Row 325 */
trt_dom:      test td21 - td11 = td22 - td12,
              td21 - td11 = td23 - td13,
              td21 - td11 = td24 - td14,
                                      td31 - td11 = td32 - td12,
              td31 - td11 = td33 - td13,
                                      td31 - td11 = td34 - td14,
                                      td41 - td11 = td42 - td12,
              td41 - td11 = td43 - td13,
                                      td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 326 */
indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
            tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                  tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 327 */
indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
            tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                  tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* Index*trt*dom -- many ways to express this 3-way interaction hypothesis. Row 328 */
indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
              tdi21 - tdi11 = tdi23 - tdi13,
              tdi21 - tdi11 = tdi24 - tdi14,
                                      tdi31 - tdi11 = tdi32 - tdi12,
              tdi31 - tdi11 = tdi33 - tdi13,
                                      tdi31 - tdi11 = tdi34 - tdi14,
                                      tdi41 - tdi11 = tdi42 - tdi12,
              tdi41 - tdi11 = tdi43 - tdi13,
                                      tdi41 - tdi11 = tdi44 - tdi14 / print;
run;
quit;

proc reg data=m;
 M3Gen_RM: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-
np4/noint;

 /*Set constraints for Model 3 */
 /* Set the following constraints to make the data set full rank */
  Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23;
/* Remove to use restiction below */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;

 /* coefficient on SBt=0. Row 372 */
RMt:   test RMt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 373 */
```

```
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
            td11+td12+td13+td14 = td31+td32+td33+td34,
                                          td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 374 */
dom:      test td11+td21+td31+td41 = td12+td22+td32+td42,
            td11+td21+td31+td41 = td13+td23+td33+td43,
                                          td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 375 */
indx: test tdi11 + tdi12 + tdi13 + tdi14 +
         tdi21 + tdi22 + tdi23 + tdi24 +
                            tdi31 + tdi32 + tdi33 + tdi34 +
                            tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;

/* trt*dom interaction.  Row 376 */
trt_dom:      test td21 - td11 = td22 - td12,
               td21 - td11 = td23 - td13,
               td21 - td11 = td24 - td14,
                                                   td31 - td11 = td32 - td12,
               td31 - td11 = td33 - td13,
                                                   td31 - td11 = td34 - td14,
                                                   td41 - td11 = td42 - td12,
               td41 - td11 = td43 - td13,
                                                   td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 377 */
indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
            tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                          tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 378 */
indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
            tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                          tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* Index*trt*dom -- many ways to express this 3-way interaction hypothesis. Row 379 */
indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
               tdi21 - tdi11 = tdi23 - tdi13,
               tdi21 - tdi11 = tdi24 - tdi14,
                                                   tdi31 - tdi11 = tdi32 - tdi12,
               tdi31 - tdi11 = tdi33 - tdi13,
                                                   tdi31 - tdi11 = tdi34 - tdi14,
                                                   tdi41 - tdi11 = tdi42 - tdi12,
               tdi41 - tdi11 = tdi43 - tdi13,
                                                   tdi41 - tdi11 = tdi44 - tdi14 / print;
run;
quit;


/* Fit Model 4 generalized. */

proc reg data=m;
 M4_SBGen: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* The following constraints make the design matrix full rank */
/* These are based on an assumption of no interaction */
```

```
 Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23; /*
Remove to use restiction below */


/* Construct type III tests (akin to rows 423-432 in Jill's spreadsheet). */

/* coefficient on SBt=0. row 423 */
 SBt:   test SBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 424 */
 trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
             td11+td12+td13+td14 = td31+td32+td33+td34,
                                        td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 425 */
 dom:     test td11+td21+td31+td41 = td12+td22+td32+td42,
            td11+td21+td31+td41 = td13+td23+td33+td43,
                                       td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 426 */
 indx: test tdi11 + tdi12 + tdi13 + tdi14 +
         tdi21 + tdi22 + tdi23 + tdi24 +
                          tdi31 + tdi32 + tdi33 + tdi34 +
                          tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;

/* trt*dom interaction. Row 427 */
trt_dom:     test td21 - td11 = td22 - td12,
               td21 - td11 = td23 - td13,
               td21 - td11 = td24 - td14,
                                            td31 - td11 = td32 - td12,
               td31 - td11 = td33 - td13,
                                            td31 - td11 = td34 - td14,
                                            td41 - td11 = td42 - td12,
               td41 - td11 = td43 - td13,
                                            td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 428 */
 indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
             tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                        tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 429 */
 indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
             tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                        tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* index*trt*dom:  there are many ways to express this 3-way interaction hypothesis. Row 430 */
 indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
                tdi21 - tdi11 = tdi23 - tdi13,
                tdi21 - tdi11 = tdi24 - tdi14,
                                             tdi31 - tdi11 = tdi32 - tdi12,
                tdi31 - tdi11 = tdi33 - tdi13,
                                             tdi31 - tdi11 = tdi34 - tdi14,
                                             tdi41 - tdi11 = tdi42 - tdi12,
                tdi41 - tdi11 = tdi43 - tdi13,
                                             tdi41 - tdi11 = tdi44 - tdi14 / print;
```

```
/* NP -- avg of the slopes = 0.  Row 431 */
 NP: test 0.25*np1 + 0.25*np2 + 0.25*np3 + 0.25*np4 = 0 / print;
              /* Note that the above is equivalent to np1 + np2 + np3 + np4 = 0, which is how I specify remaining tests */

/* test for equalitly of np slopes across the four treatments. Row 432 of the Mixed spreadsheet */
 NP_trt: test np1=np2=np3=np4 / print;

run;
quit;


proc reg data=m;
 M4_KBGen: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

 /* The following constraints make the design matrix full rank */
 /* These are based on an assumption of no interaction */
 Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23; /*
Remove to use restiction below */


/* Construct type III tests (akin to rows 479-488 in Jill's spreadsheet). */

/* coefficient on SBt=0. row 479 */
 KBt:   test KBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 480 */
 trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
             td11+td12+td13+td14 = td31+td32+td33+td34,
                                     td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 481 */
 dom:     test td11+td21+td31+td41 = td12+td22+td32+td42,
            td11+td21+td31+td41 = td13+td23+td33+td43,
                                    td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 482 */
 indx: test tdi11 + tdi12 + tdi13 + tdi14 +
          tdi21 + tdi22 + tdi23 + tdi24 +
                        tdi31 + tdi32 + tdi33 + tdi34 +
                        tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;

/* trt*dom interaction. Row 483 */
trt_dom:      test td21 - td11 = td22 - td12,
                td21 - td11 = td23 - td13,
                td21 - td11 = td24 - td14,
                                        td31 - td11 = td32 - td12,
                td31 - td11 = td33 - td13,
                                        td31 - td11 = td34 - td14,
                                        td41 - td11 = td42 - td12,
                td41 - td11 = td43 - td13,
                                        td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 484 */
 indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
             tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                     tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;
```

```
/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 485 */
 indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
        tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                         tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* Index*trt*dom: there are many ways to express this 3-way interaction hypothesis. Row 486 */
 indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
          tdi21 - tdi11 = tdi23 - tdi13,
          tdi21 - tdi11 = tdi24 - tdi14,
                              tdi31 - tdi11 = tdi32 - tdi12,
          tdi31 - tdi11 = tdi33 - tdi13,
                              tdi31 - tdi11 = tdi34 - tdi14,
                              tdi41 - tdi11 = tdi42 - tdi12,
          tdi41 - tdi11 = tdi43 - tdi13,
                              tdi41 - tdi11 = tdi44 - tdi14 / print;

/* NP -- avg of the slopes = 0.  Row 487 */
 NP: test 0.25*np1 + 0.25*np2 + 0.25*np3 + 0.25*np4 = 0 / print;
        /* Note that the above is equivalent to np1 + np2 + np3 + np4 = 0, which is how I specify remaining tests */

/* test for equalitly of np slopes across the four treatments. Row 488 of the Mixed spreadsheet */
 NP_trt: test np1=np2=np3=np4 / print;

run;
quit;

proc reg data=m;
 M4_RMGen: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-
np4/noint;

 /* The following constraints make the design matrix full rank */
 /* These are based on an assumption of no interaction */
 Restrict 9*tdi24 = 3*tdi14 + 3*tdi34 + 3*tdi44 + tdi11+tdi12+tdi13+tdi31+tdi32+tdi33+tdi41+tdi42+tdi43 - 3*tdi21 - 3*tdi22 - 3*tdi23; /*
Remove to use restiction below */


/* Construct type III tests (akin to rows 536-545 in Jill's spreadsheet). */

/* coefficient on SBt=0. row 536 */
 RMt:   test RMt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 537 */
 trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
        td11+td12+td13+td14 = td31+td32+td33+td34,
                         td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 538 */
 dom:     test td11+td21+td31+td41 = td12+td22+td32+td42,
        td11+td21+td31+td41 = td13+td23+td33+td43,
                         td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 539 */
 indx: test tdi11 + tdi12 + tdi13 + tdi14 +
      tdi21 + tdi22 + tdi23 + tdi24 +
              tdi31 + tdi32 + tdi33 + tdi34 +
              tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;
```

```
/* trt*dom interaction. Row 540 */
trt_dom:      test td21 - td11 = td22 - td12,
              td21 - td11 = td23 - td13,
              td21 - td11 = td24 - td14,
                                        td31 - td11 = td32 - td12,
              td31 - td11 = td33 - td13,
                                        td31 - td11 = td34 - td14,
                                        td41 - td11 = td42 - td12,
              td41 - td11 = td43 - td13,
                                        td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 541 */
 indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
            tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 542 */
 indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
            tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* Index*trt*dom : there are many ways to express this 3-way interaction hypothesis. Row 543 */
 indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
               tdi21 - tdi11 = tdi23 - tdi13,
               tdi21 - tdi11 = tdi24 - tdi14,
                                        tdi31 - tdi11 = tdi32 - tdi12,
               tdi31 - tdi11 = tdi33 - tdi13,
                                        tdi31 - tdi11 = tdi34 - tdi14,
                                        tdi41 - tdi11 = tdi42 - tdi12,
               tdi41 - tdi11 = tdi43 - tdi13,
                                        tdi41 - tdi11 = tdi44 - tdi14 / print;

/* NP -- avg of the slopes = 0.  Row 544 */
 NP: test 0.25*np1 + 0.25*np2 + 0.25*np3 + 0.25*np4 = 0 / print;
            /* Note that the above is equivalent to np1 + np2 + np3 + np4 = 0 */

/* test for equalitly of np slopes across the four treatments. Row 545 of the Mixed spreadsheet */
 NP_trt: test np1=np2=np3=np4 / print;

run;
quit;
```

```
options ps=120;

*libname model "C:\Northern Prairie Project\Models\StatisticalModels_ProcReg\SAS\tall\Thru2016";
libname model "C:\Data\RCRP\NP\model empirical analysis\2016 data\models\tall";

Data tall;
  set model.tall_float;
  if NPt=. then delete;
run;


***checking frequency of trtmnt and domt combinations**;
proc freq data=tall;
        tables trtmnt*domt;
run;

***checking frequency of trtmnt and index combinations**;
data a;
        set tall;
        i="M";
        if indext<2 then i="L";
        if indext>4 then i="H";
run;

proc freq data=a;
        tables i*trtmnt;
run;

***checking frequency of domt and index combinations**;
proc freq data=a;
        tables i*domt;
run;

***checking frequency of trtmnt and domt and index level combinations**;
proc freq data=a;
        tables i*trtmnt*domt;
run;

/* The above data are ill-conditioned in the following way:
                1.There is only 1 unit with Trtment=Burn WW and Dom=Co which means we can't estimate a slope for that
trtmnt*dom combo
                2.There is only 1 unit with Trtment=Burn WW and Dom=RM which means we can't estimate a slope for that
trtmnt*dom combo

   Because of the above condition, the "usual" hypotheses in an ANCOVA like this cannot be tested.  As a result, GLM is testing
   non-sensical hypotheses in place of the usual ones.
*/


/* Code the full-rank design matrix for use with PROC REG */

data t;
 set tall;
  if NPt=. then delete;
  /* In its most general form, model 4 has 16 interecepts or constant terms, one for each trtmnt*dom group. */
  /* Therefore, I assign 16 indicator variables to identify the treatment*dom class. */
  /* td11 corresponds to row 1, column 1 in the 4x4 schematics in Jill's spreadsheet */
```

```
  /* td12 is row 1, column2; td13 is row 1, col 3, etc. */

  Array td {16} td11-td14 td21-td24 td31-td34 td41-td44;
  do i = 1 to 16; td[i]=0; end;  /* initialize all tds to zero */
  if trtmnt='Burn w/in window' then do;
    if domt='SB' then td11=1;
    if domt='Co' then td12=1;
    if domt='KB' then td13=1;
    if domt='RM' then td14=1;
  end;
  if trtmnt='Graze w/in window' then do;
    if domt='SB' then td21=1;
    if domt='Co' then td22=1;
    if domt='KB' then td23=1;
    if domt='RM' then td24=1;
  end;
  if trtmnt='Defoliate' then do;
    if domt='SB' then td31=1;
    if domt='Co' then td32=1;
    if domt='KB' then td33=1;
    if domt='RM' then td34=1;
  end;
  if trtmnt='Rest' then do;
    if domt='SB' then td41=1;
    if domt='Co' then td42=1;
    if domt='KB' then td43=1;
    if domt='RM' then td44=1;
  end;

  /* In its most general form, model 4 has 16 distinct Index slopes.  Assign those variables here. */
  Array tdi {16} tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44;
  do i = 1 to 16;
    tdi[i]=td[i]*indext;
  end;

  /* In its most general form, model 4 has 4 distinct slopes associated with NPI.  Assign those variables here. */
  if trtmnt='Burn w/in window' then np1 = npt; else np1=0;
  if trtmnt='Graze w/in window' then np2 = npt; else np2=0;
  if trtmnt='Defoliate' then np3 = npt; else np3=0;
  if trtmnt='Rest' then np4 = npt; else np4=0;
run;
/* The design matrix is now complete */

/* Fit Model 1: Impose constraints to fit the general model concept, as well as any response-specific constraints to enforce equalities
specific to the
   model (of which M1 has none).*/
/* Model 1 Concept:  All treatments are equally effective, regardless of dominant invader, defoliation history, or invasion level.  No
dependence
on veg state or defoliation state. */

proc reg data=t outest=Tf_M1SBrs covout;
 M1_SBrs: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 1 */
/* The design matrix is already full rank for Model 1 and does not require the constraints of Models 3-6 to make full rank */
 Restrict td11=td12=td13=td14=td21=td22=td23=td24=td31=td32=td33=td34; /* treatment effects are the same across all four
dominance types and three active treatments - Bww, Gww, Defoliate */
```

```
 Restrict td41=td42=td43=td44; /* effect of Rest is the same across all four dominance types */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

run;
quit;

proc reg data=t outest=Tf_M1KBrs covout;
 M1_KBrs: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 1 */
/* The design matrix is already full rank for Model 1 and does not require the constraints of Models 3-6 to make full rank */
 Restrict td11=td12=td13=td14=td21=td22=td23=td24=td31=td32=td33=td34; /* treatment effects are the same across all four
dominance types and three active treatments - Bww, Gww, Defoliate */
 Restrict td41=td42=td43=td44; /* effect of Rest is the same across all four dominance types */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

run;
quit;

proc reg data=t outest=Tf_M1RMrs covout;
 M1_RMrs: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 1 */
/* The design matrix is already full rank for Model 1 and does not require the constraints of Models 3-6 to make full rank */
 Restrict td11=td12=td13=td14=td21=td22=td23=td24=td31=td32=td33=td34; /* treatment effects are the same across all four
dominance types and three active treatments - Bww, Gww, Defoliate */
 Restrict td41=td42=td43=td44; /* effect of Rest is the same across all four dominance types */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

run;
quit;


/* Fit Model 2: Impose constraints to fit the general model concept, as well as any response-specific constraints to td to enforce
equalities specific
   to the model. */
/* Model 2 concept:  Treatment effectiveness is state-dependent and depends on the type of invader that is dominant. */

proc reg data=t outest=Tf_M2SBrs covout;
 M2_SBrs: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 2 */
```

```
/* The design matrix is already full rank for Model 2 and does not require the constraints of Models 3-6 to make full rank */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;
/* Set the following equality constraints */
 *Restrict td14=td24=td34; /* If RM dom, Bww=Gww=D */

run;
quit;

proc reg data=t outest=Tf_M2KBrs covout;
 M2_KBrs: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* Set constraints for Model 2 */
/* The design matrix is already full rank for Model 2 and does not require the constraints of Models 3-6 to make full rank */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;
/* Set the following equality constraints */
 *Restrict td14=td24=td34; /* If RM dom, Bww=Gww=D */

run;
quit;


proc reg data=t outest=Tf_M2RMrs covout;
 M2_RMrs: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 2 */
/* The design matrix is already full rank for Model 2 and does not require the constraints of Models 3-6 to make full rank */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;
/* Set the following equality constraints */
 Restrict td14=td24=td34; /* If RM dom, Bww=Gww=D */

run;
quit;


/* Fit Model 3: Impose constraints to fit the general model concept, as well as any response-specific constraints on tdi to enforce equalities specific
   to the model. */
/* Model 3 concept:  Treatment effectiveness is state-dependent and depends on the type of invader that is dominant, and on the defoliation history. */

proc reg data=t outest=Tf_M3SBrs covout;
 M3_SBrs: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;
```

```
/* Set constraints for Model 3 */
/* Set the following constraints to make the data set full rank */
 *Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints */
 Restrict tdi11=tdi21=tdi31; /*BwwxSBxindex=GwwxSBxindex=DxSBxindex, but RestxSBxindex left unconstrained. Slope for active
treatments - Bww, Gww, and Defoliate - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB,
CO, KB, and RM are allowed to be different.*/
 Restrict tdi12=tdi22=tdi32; /*BwwxCOxindex=GwwxCOxindex=DxCOxindex, but RestxCOxindex left unconstrained. Slope for
active treatments - Bww, Gww, and Defoliate - are set the same for CO, while slope for Rest is left unconstrained.*/
 *Restrict tdi13=tdi23=tdi33; /*BwwxKBxindex=GwwxKBxindex=DxKBxindex, but RestxKBxindex left unconstrained. Slope for active
treatments - Bww, Gww, and Defoliate - are set the same for KB, while slope for Rest is left unconstrained.*/
 *Restrict tdi14=tdi24=tdi34; /*BwwxRMxindex=GwwxRMxindex=DxRMxindex, but RestxRMxindex left unconstrained. Slope for
active treatments - Bww, Gww, and Defoliate - are set the same for RM, while slope for Rest is left unconstrained.*/

run;
quit;

 proc reg data=t outest=Tf_M3KBrs covout;
 M3_KBrs: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 3 */
/* Set the following constraints to make the data set full rank */
 *Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints */
 *Restrict tdi11=tdi21=tdi31; /*BwwxSBxindex=GwwxSBxindex=DxSBxindex, but RestxSBxindex left unconstrained. Slope for active
treatments - Bww, Gww, and Defoliate - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB,
CO, KB, and RM are allowed to be different.*/
 Restrict tdi12=tdi22=tdi32; /*BwwxCOxindex=GwwxCOxindex=DxCOxindex, but RestxCOxindex left unconstrained. Slope for
active treatments - Bww, Gww, and Defoliate - are set the same for CO, while slope for Rest is left unconstrained.*/
 Restrict tdi13=tdi23=tdi33; /*BwwxKBxindex=GwwxKBxindex=DxKBxindex, but RestxKBxindex left unconstrained. Slope for active
treatments - Bww, Gww, and Defoliate - are set the same for KB, while slope for Rest is left unconstrained.*/
 *Restrict tdi14=tdi24=tdi34; /*BwwxRMxindex=GwwxRMxindex=DxRMxindex, but RestxRMxindex left unconstrained. Slope for
active treatments - Bww, Gww, and Defoliate - are set the same for RM, while slope for Rest is left unconstrained.*/

 run;
 quit;

 proc reg data=t outest=Tf_M3RMrs covout;
 M3_RMrs: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 3 */
/* Set the following constraints to make the data set full rank */
 Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 *Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
```

```
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints */
 *Restrict tdi11=tdi21=tdi31; /*BwwxSBxindex=GwwxSBxindex=DxSBxindex, but RestxSBxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB, CO, KB, and RM are allowed to be different.*/
 *Restrict tdi12=tdi22=tdi32; /*BwwxCOxindex=GwwxCOxindex=DxCOxindex, but RestxCOxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for CO, while slope for Rest is left unconstrained.*/
 *Restrict tdi13=tdi23=tdi33; /*BwwxKBxindex=GwwxKBxindex=DxKBxindex, but RestxKBxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for KB, while slope for Rest is left unconstrained.*/
 Restrict tdi14=tdi24=tdi34; /*BwwxRMxindex=GwwxRMxindex=DxRMxindex, but RestxRMxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for RM, while slope for Rest is left unconstrained.*/

run;
quit;

/* Fit Model 4: Impose constraints to fit the general model concept,as well as any response-specific constraints to enforce equalities specific
   to the model. */
/* Model 4 concept:  Treatment effectiveness is state-dependent and depends on the type of invader that is dominant, the defoliation history, and
   the level of invasion. */

proc reg data=t outest=Tf_M4SBrs covout;
 M4_SBrs: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* The following constraint makes the design matrix full rank */
/* These are based on an assumption of no interaction */
 *Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality constraint below. */
 Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality constraint below. */
/* Set same equality constraints for tdi as used for Model 3 */
  Restrict tdi11=tdi21=tdi31; /*BwwxSBxindex=GwwxSBxindex=DefoliatexSBxindex, but RestxSBxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB, CO, KB, and RM are allowed to be different.*/
  Restrict tdi12=tdi22=tdi32; /*BwwxCOxindex=GwwxCOxindex=DefoliatexCOxindex, but RestxCOxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for CO, while slope for Rest is left unconstrained.*/
  *Restrict tdi13=tdi23=tdi33; /*BwwxKBxindex=GwwxKBxindex=DefoliatexKBxindex, but RestxKBxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for KB, while slope for Rest is left unconstrained.*/
  *Restrict tdi14=tdi24=tdi34; /*BwwxRMxindex=GwwxRMxindex=DefoliatexRMxindex, but RestxRMxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for RM, while slope for Rest is left unconstrained.*/
/* Set equality constraints for np*trtmnt specific to Model 4.  There are none. */
  *Restrict np1=np2=np3; /* playing with idea of setting the change in effect of Bww, Gww, and Defoliate with change in NP to equal */
  *Restrict np4=0; /* playing with idea of the change in effect of Rest with change in NP being zero */

 run;
 quit;


proc reg data=t outest=Tf_M4KBrs covout;
 M4_KBrs: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* The following constraint makes the design matrix full rank */
/* These are based on an assumption of no interaction */
```

```
 *Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
/* Set same equality constraints for tdi as used for Model 3 */
  *Restrict tdi11=tdi21=tdi31; /*BwwxSBxindex=GwwxSBxindex=DefoliatexSBxindex, but RestxSBxindex left unconstrained. Slope
for active treatments - Bww, Gww, and Defoliate - are set the same for SB, while slope for Rest is left unconstrained. Note that
slopes of SB, CO, KB, and RM are allowed to be different.*/
  Restrict tdi12=tdi22=tdi32; /*BwwxCOxindex=GwwxCOxindex=DefoliatexCOxindex, but RestxCOxindex left unconstrained. Slope
for active treatments - Bww, Gww, and Defoliate - are set the same for CO, while slope for Rest is left unconstrained.*/
  Restrict tdi13=tdi23=tdi33; /*BwwxKBxindex=GwwxKBxindex=DefoliatexKBxindex, but RestxKBxindex left unconstrained. Slope for
active treatments - Bww, Gww, and Defoliate - are set the same for KB, while slope for Rest is left unconstrained.*/
  *Restrict tdi14=tdi24=tdi34; /*BwwxRMxindex=GwwxRMxindex=DefoliatexRMxindex, but RestxRMxindex left unconstrained. Slope
for active treatments - Bww, Gww, and Defoliate - are set the same for RM, while slope for Rest is left unconstrained.*/
/* Set equality constraints for np*trtmnt specific to Model 4.  There are none. */
  *Restrict np1=np2=np3; /* playing with idea of setting the change in effect of Bww, Gww, and Defoliate with change in NP to equal
*/
  *Restrict np4=0; /* playing with idea of the change in effect of Rest with change in NP being zero */

 run;
 quit;

proc reg data=t outest=Tf_M4RMrs covout;
 M4_RMrs: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* The following constraint makes the design matrix full rank */
/* These are based on an assumption of no interaction */
 Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 *Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
/* Set same equality constraints for tdi as used for Model 3 */
  *Restrict tdi11=tdi21=tdi31; /*BwwxSBxindex=GwwxSBxindex=DefoliatexSBxindex, but RestxSBxindex left unconstrained. Slope
for active treatments - Bww, Gww, and Defoliate - are set the same for SB, while slope for Rest is left unconstrained. Note that
slopes of SB, CO, KB, and RM are allowed to be different.*/
  *Restrict tdi12=tdi22=tdi32; /*BwwxCOxindex=GwwxCOxindex=DefoliatexCOxindex, but RestxCOxindex left unconstrained. Slope
for active treatments - Bww, Gww, and Defoliate - are set the same for CO, while slope for Rest is left unconstrained.*/
  *Restrict tdi13=tdi23=tdi33; /*BwwxKBxindex=GwwxKBxindex=DefoliatexKBxindex, but RestxKBxindex left unconstrained. Slope
for active treatments - Bww, Gww, and Defoliate - are set the same for KB, while slope for Rest is left unconstrained.*/
  Restrict tdi14=tdi24=tdi34; /*BwwxRMxindex=GwwxRMxindex=DefoliatexRMxindex, but RestxRMxindex left unconstrained. Slope
for active treatments - Bww, Gww, and Defoliate - are set the same for RM, while slope for Rest is left unconstrained.*/
/* Set equality constraints for np*trtmnt specific to Model 4.  There are none. */
  *Restrict np1=np2=np3; /* playing with idea of setting the change in effect of Bww, Gww, and Defoliate with change in NP to equal
*/
  *Restrict np4=0; /* playing with idea of the change in effect of Rest with change in NP being zero */

 run;
 quit;


/* Fit Model 5:  Impose constraints to fit the general model concept, as well as any response-specific constraints to enforce
equalities specific
   to the model. */
/* Model 5 concept:  Treatment effectiveness is state-dependent and depends on the type of invader that is dominant, and on the
defoliation history. */

proc reg data=t outest=Tf_M5SBrs covout;
```

```
 M5_SBrs: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* Set constraints for Model 5 */
/* Set the following constraints to make the data set full rank */
 *Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality constraint below. */
 Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality constraint below. */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints specific to Model 5 for trtxdom that Bww=Gww when SB dom */
 Restrict td11=td21; /* BwwxSB=GwwxSB */
/* Set equality constraints for tdi, similar as those used for Model 3, but left Defoliate unconstrained */
 Restrict tdi11=tdi21; /*BwwxSBxindex=GwwxSBxindex, but DefoliatexSBxindex and RestxSBxindex are left unconstrained. Note that slopes of SB, CO, KB, and RM are allowed to be different.*/
 Restrict tdi12=tdi22; /*BwwxCOxindex=GwwxCOxindex, but DefoliatexCOxindex and RestxCOxindex are left unconstrained. */
 *Restrict tdi13=tdi23=tdi33; /*BwwxKBxindex=GwwxKBxindex=DxKBxindex, but RestxKBxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for KB, while slope for Rest is left unconstrained.*/
 *Restrict tdi14=tdi24=tdi34; /*BwwxRMxindex=GwwxRMxindex=DxRMxindex, but RestxRMxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for RM, while slope for Rest is left unconstrained.*/

run;
quit;

 proc reg data=t outest=Tf_M5KBrs covout;
 M5_KBrs: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 5 */
/* Set the following constraints to make the data set full rank */
 *Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality constraint below. */
 Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality constraint below. */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints specific to Model 5 */
 *Restrict td11=td21; /* BwwxSB=GwwxSB */
/* Set equality constraints for tdi, same as those used for Model 3 */
 *Restrict tdi11=tdi21=tdi31; /*BwwxSBxindex=GwwxSBxindex=DxSBxindex, but RestxSBxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB, CO, KB, and RM are allowed to be different.*/
 Restrict tdi12=tdi22=tdi32; /*BwwxCOxindex=GwwxCOxindex=DxCOxindex, but RestxCOxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for CO, while slope for Rest is left unconstrained.*/
 Restrict tdi13=tdi23=tdi33; /*BwwxKBxindex=GwwxKBxindex=DxKBxindex, but RestxKBxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for KB, while slope for Rest is left unconstrained.*/
 *Restrict tdi14=tdi24=tdi34; /*BwwxRMxindex=GwwxRMxindex=DxRMxindex, but RestxRMxindex left unconstrained. Slope for active treatments - Bww, Gww, and Defoliate - are set the same for RM, while slope for Rest is left unconstrained.*/

 run;
 quit;

 proc reg data=t outest=Tf_M5RMrs covout;
 M5_RMrs: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 5 */
/* Set the following constraints to make the data set full rank */
```

```
 Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 *Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints specific to Model 5 */
 *Restrict td11=td21; /* BwwxSB=GwwxSB */
/* Set equality constraints for tdi, same as those used for Model 3 */
 *Restrict tdi11=tdi21=tdi31; /*BwwxSBxindex=GwwxSBxindex=DxSBxindex, but RestxSBxindex left unconstrained. Slope for active
treatments - Bww, Gww, and Defoliate - are set the same for SB, while slope for Rest is left unconstrained. Note that slopes of SB,
CO, KB, and RM are allowed to be different.*/
 *Restrict tdi12=tdi22=tdi32; /*BwwxCOxindex=GwwxCOxindex=DxCOxindex, but RestxCOxindex left unconstrained. Slope for
active treatments - Bww, Gww, and Defoliate - are set the same for CO, while slope for Rest is left unconstrained.*/
 *Restrict tdi13=tdi23=tdi33; /*BwwxKBxindex=GwwxKBxindex=DxKBxindex, but RestxKBxindex left unconstrained. Slope for active
treatments - Bww, Gww, and Defoliate - are set the same for KB, while slope for Rest is left unconstrained.*/
 Restrict tdi14=tdi24=tdi34; /*BwwxRMxindex=GwwxRMxindex=DxRMxindex, but RestxRMxindex left unconstrained. Slope for
active treatments - Bww, Gww, and Defoliate - are set the same for RM, while slope for Rest is left unconstrained.*/

run;
quit;



/* Fit Model 6:  Impose constraints to fit the general model concept, as well as any response-specific constraints to enforce
equalities specific
   to the model. */
/* Model 6 concept:  Treatment effectiveness is state-dependent and depends on the type of invader that is dominant, and on the
defoliation history. */

proc reg data=t outest=Tf_M6SBrs covout;
 M6_SBrs: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* Set constraints for Model 6 */
/* Set the following constraints to make the data set full rank */
 *Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints specific to Model 6 regarding effectiveness of Defoliate and Rest.  Since Model 6 says D=R for all
dominance scenarios,
 should we place all of the restrictions below and not just those for SB dom? */
 Restrict td31=td41; /* DxSB=RxSB */
 Restrict td32=td42; /* DxCO=RxCO */
 *Restrict td33=td43; /* DxKB=RxKB */
 *Restrict td34=td44; /* DxRM=RxRM */
/* Set equality constraints on tdi, similar to those for Model 3 */
/* Model 3 tdi constraints set Bww=Gww=D, with Rest unconstrained.  But since Model6 states that D=R, as set in the td constaints
above,
   what does this mean for the tdi constraints?  Do we set Bww=Gww and leave D and R unconstrained?  Do we set Bww=Gww and
D=R?
   Do we leave it as Bww=Gww=D (i.e., same as M3)? In discussion with Terry and Clint on July 15, 2014, we decided that the
appropriate tdi constraint
   is to se Bww=Gww and D=R.  */
 Restrict tdi11=tdi21; /*BwwxSBxindex=GwwxSBxindex. Index slope for Bww and Gww are set the same for SB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
```

```
 Restrict tdi31=tdi41; /*DxSBxindex=RxSBxindex. Index slope for Defoliate and Rest are set the same for SB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 Restrict tdi12=tdi22; /*BwwxCOxindex=GwwxCOxindex. Index slope for Bww and Gww are set the same for CO. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 Restrict tdi32=tdi42; /*DxCOxindex=RxCOxindex. Index slope for Defoliate and Rest are set the same for CO. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 *Restrict tdi13=tdi23; /*BwwxKBxindex=GwwxKBxindex. Index slope for Bww and Gww are set the same for KB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 *Restrict tdi33=tdi43; /*DxKBxindex=RxKBxindex. Index slope for Defoliate and Rest are set the same for KB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 *Restrict tdi14=tdi24; /*BwwxRMxindex=GwwxRMxindex. Index slope for Bww and Gww are set the same for RM. Slopes of SB,
CO, KB, and RM are allowed to be different.*/
 *Restrict tdi34=tdi44; /*DxRMxindex=RxRMxindex. Index slope for Defoliate and Rest are set the same for RM. Slopes of SB, CO,
KB, and RM are allowed to be different.*/

run;
quit;

 proc reg data=t outest=Tf_M6KBrs covout;
 M6_KBrs: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 6 */
/* Set the following constraints to make the data set full rank */
 *Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints specific to Model 6 regarding effectiveness of Defoliate and Rest.  Since Model 6 says D=R for all
dominance scenarios,
 should we place all of the restrictions below and not just those for KB dom? */
 *Restrict td31=td41; /* DxSB=RxSB */
 Restrict td32=td42; /* DxCO=RxCO */
 Restrict td33=td43; /* DxKB=RxKB */
 *Restrict td34=td44; /* DxRM=RxRM */
/* Set equality constraints on tdi, similar to those for Model 3 */
/* Model 3 tdi constraints set Bww=Gww=D, with Rest unconstrained.  But since Model6 states that D=R, as set in the td constaints
above,
   what does this mean for the tdi constraints?  Do we set Bww=Gww and leave D and R unconstrained?  Do we set Bww=Gww and
D=R?
   Do we leave it as Bww=Gww=D (i.e., same as M3)? In discussion with Terry and Clint on July 15, 2014, we decided that the
appropriate tdi constraint
   is to se Bww=Gww and D=R.  */
 *Restrict tdi11=tdi21; /*BwwxSBxindex=GwwxSBxindex. Index slope for Bww and Gww are set the same for SB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 *Restrict tdi31=tdi41; /*DxSBxindex=RxSBxindex. Index slope for Defoliate and Rest are set the same for SB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 Restrict tdi12=tdi22; /*BwwxCOxindex=GwwxCOxindex. Index slope for Bww and Gww are set the same for CO. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 Restrict tdi32=tdi42; /*DxCOxindex=RxCOxindex. Index slope for Defoliate and Rest are set the same for CO. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 Restrict tdi13=tdi23; /*BwwxKBxindex=GwwxKBxindex. Index slope for Bww and Gww are set the same for KB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 Restrict tdi33=tdi43; /*DxKBxindex=RxKBxindex. Index slope for Defoliate and Rest are set the same for KB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
```

```
 *Restrict tdi14=tdi24; /*BwwxRMxindex=GwwxRMxindex. Index slope for Bww and Gww are set the same for RM. Slopes of SB,
CO, KB, and RM are allowed to be different.*/
 *Restrict tdi34=tdi44; /*DxRMxindex=RxRMxindex. Index slope for Defoliate and Rest are set the same for RM. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 run;
 quit;

 proc reg data=t outest=Tf_M6RMrs covout;
 M6_RMrs: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 6 */
/* Set the following constraints to make the data set full rank */
 Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 *Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;
/* Set equality constraints specific to Model 6 regarding effectiveness of Defoliate and Rest.  Since Model 6 says D=R for all
dominance scenarios,
 should we place all of the restrictions below and not just those for RM dom? */
 *Restrict td31=td41; /* DxSB=RxSB */
 *Restrict td32=td42; /* DxCO=RxCO */
 *Restrict td33=td43; /* DxKB=RxKB */
 Restrict td34=td44; /* DxRM=RxRM */
/* Set equality constraints on tdi, similar to those for Model 3 */
/* Model 3 tdi constraints set Bww=Gww=D, with Rest unconstrained.  But since Model6 states that D=R, as set in the td constaints
above,
   what does this mean for the tdi constraints?  Do we set Bww=Gww and leave D and R unconstrained?  Do we set Bww=Gww and
D=R?
   Do we leave it as Bww=Gww=D (i.e., same as M3)? In discussion with Terry and Clint on July 15, 2014, we decided that the
appropriate tdi constraint
   is to se Bww=Gww and D=R.  */
 *Restrict tdi11=tdi21; /*BwwxSBxindex=GwwxSBxindex. Index slope for Bww and Gww are set the same for SB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 *Restrict tdi31=tdi41; /*DxSBxindex=RxSBxindex. Index slope for Defoliate and Rest are set the same for SB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 *Restrict tdi12=tdi22; /*BwwxCOxindex=GwwxCOxindex. Index slope for Bww and Gww are set the same for CO. Slopes of SB,
CO, KB, and RM are allowed to be different.*/
 *Restrict tdi32=tdi42; /*DxCOxindex=RxCOxindex. Index slope for Defoliate and Rest are set the same for CO. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 *Restrict tdi13=tdi23; /*BwwxKBxindex=GwwxKBxindex. Index slope for Bww and Gww are set the same for KB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 *Restrict tdi33=tdi43; /*DxKBxindex=RxKBxindex. Index slope for Defoliate and Rest are set the same for KB. Slopes of SB, CO,
KB, and RM are allowed to be different.*/
 Restrict tdi14=tdi24; /*BwwxRMxindex=GwwxRMxindex. Index slope for Bww and Gww are set the same for RM. Slopes of SB,
CO, KB, and RM are allowed to be different.*/
 Restrict tdi34=tdi44; /*DxRMxindex=RxRMxindex. Index slope for Defoliate and Rest are set the same for RM. Slopes of SB, CO,
KB, and RM are allowed to be different.*/

run;
quit;

/***
 ***Append covariance output from models 1 and 2***;

 proc append base=Tf_m1sbrs data=Tf_m2sbrs;
```

```
 run;

 proc append base=Tf_m1kbrs data=Tf_m2kbrs;
 run;

 proc append base=Tf_m1rmrs data=Tf_m2rmrs;
 run;
***/

***Make the appended output permanent***;

*libname model "C:\Northern Prairie Project\Models\StatisticalModels_ProcReg\ProcReg_Tall\Thru2016";

 data model.Tf_SBrs;
		set Tf_m1sbrs Tf_m2sbrs Tf_m3sbrs Tf_m4sbrs Tf_m5sbrs Tf_m6sbrs;
run;

 data model.Tf_KBrs;
		set Tf_m1kbrs Tf_m2kbrs Tf_m3kbrs Tf_m4kbrs Tf_m5kbrs Tf_m6kbrs;
run;

 data model.Tf_RMrs;
		set Tf_m1rmrs Tf_m2rmrs Tf_m3rmrs Tf_m4rmrs Tf_m5rmrs Tf_m6rmrs;
run;

**** GENERALIZED VERSIONS OF MODELS ****;

/* Fit Model 1 generalized (i.e., no equality constraints) */

proc reg data=t;
 M1Gen_SB: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

 /*Set constraints for Model 1 */
 /* The design matrix is already full rank for Model 1 and does not require the constraints of Models 3-6 to make full rank */
 Restrict td11=td12=td13=td14; /* effect of Burn w/in window is the same across all four dominance types */
 Restrict td21=td22=td23=td24; /* effect of Graze w/in window is the same across all four dominance types */
 Restrict td31=td32=td33=td34; /* effect of Defoliate is the same across all four dominance types */
 Restrict td41=td42=td43=td44; /* effect of Rest is the same across all four dominance types */
 /* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

 /* Test if active treatments are equal */
trtmnt:	test td11+td12+td13+td14 = td21+td22+td23+td24,
		td11+td12+td13+td14 = td31+td32+td33+td34 / print;
run;
quit;



proc reg data=t;
 M1Gen_KB: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 1 */
```

```
/* The design matrix is already full rank for Model 1 and does not require the constraints of Models 3-6 to make full rank */
 Restrict td11=td12=td13=td14; /* effect of Burn w/in window is the same across all four dominance types */
 Restrict td21=td22=td23=td24; /* effect of Graze w/in window is the same across all four dominance types */
 Restrict td31=td32=td33=td34; /* effect of Defoliate is the same across all four dominance types */
 Restrict td41=td42=td43=td44; /* effect of Rest is the same across all four dominance types */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

/* Test if active treatments are equal */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
            td11+td12+td13+td14 = td31+td32+td33+td34 / print;
run;
quit;

proc reg data=t;
 M1Gen_RM: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-
np4/noint;

/*Set constraints for Model 1 */
/* The design matrix is already full rank for Model 1 and does not require the constraints of Models 3-6 to make full rank */
 Restrict td11=td12=td13=td14; /* effect of Burn w/in window is the same across all four dominance types */
 Restrict td21=td22=td23=td24; /* effect of Graze w/in window is the same across all four dominance types */
 Restrict td31=td32=td33=td34; /* effect of Defoliate is the same across all four dominance types */
 Restrict td41=td42=td43=td44; /* effect of Rest is the same across all four dominance types */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

/* Test if active treatments are equal */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
            td11+td12+td13+td14 = td31+td32+td33+td34 / print;
run;
quit;


/* Fit Model 2 generalized (i.e., no equality constraints) */

proc reg data=t;
 M2Gen_SB: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 2 */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

/* coefficient on SBt=0. Row 165*/
SBt:   test SBt=0 / print;
```

```
/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 166*/
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
         td11+td12+td13+td14 = td31+td32+td33+td34,
                                td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 167 */
dom:     test td11+td21+td31+td41 = td12+td22+td32+td42,
         td11+td21+td31+td41 = td13+td23+td33+td43,
                                td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* trt*dom interaction.  Row 168 */
trt_dom:     test td21 - td11 = td22 - td12,
             td21 - td11 = td23 - td13,
             td21 - td11 = td24 - td14,
                                          td31 - td11 = td32 - td12,
             td31 - td11 = td33 - td13,
                                          td31 - td11 = td34 - td14,
                                          td41 - td11 = td42 - td12,
             td41 - td11 = td43 - td13,
                                          td41 - td11 = td44 - td14 / print;
run;
quit;

proc reg data=t;
 M2Gen_KB: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 2 */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

/* coefficient on SBt=0. Row 195 */
KBt:   test KBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 196 */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
         td11+td12+td13+td14 = td31+td32+td33+td34,
                                td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 197 */
dom:     test td11+td21+td31+td41 = td12+td22+td32+td42,
         td11+td21+td31+td41 = td13+td23+td33+td43,
                                td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* trt*dom interaction.  Row 198 */
trt_dom:     test td21 - td11 = td22 - td12,
             td21 - td11 = td23 - td13,
             td21 - td11 = td24 - td14,
                                          td31 - td11 = td32 - td12,
             td31 - td11 = td33 - td13,
                                          td31 - td11 = td34 - td14,
                                          td41 - td11 = td42 - td12,
             td41 - td11 = td43 - td13,
```

```
                                              td41 - td11 = td44 - td14 / print;
run;
quit;


proc reg data=t;
 M2Gen_RM: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-
np4/noint;

/*Set constraints for Model 2 */
/* Set tdi and np to zero */
 Restrict tdi11=tdi12=tdi13=tdi14=0;
 Restrict tdi21=tdi22=tdi23=tdi24=0;
 Restrict tdi31=tdi32=tdi33=tdi34=0;
 Restrict tdi41=tdi42=tdi43=tdi44=0;
 Restrict np1=np2=np3=np4=0;

/* coefficient on SBt=0. Row 225 */
RMt:   test RMt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 226 */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
          td11+td12+td13+td14 = td31+td32+td33+td34,
                                  td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 227 */
dom:      test td11+td21+td31+td41 = td12+td22+td32+td42,
          td11+td21+td31+td41 = td13+td23+td33+td43,
                                  td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* trt*dom interaction.  Row 228 */
trt_dom:      test td21 - td11 = td22 - td12,
              td21 - td11 = td23 - td13,
              td21 - td11 = td24 - td14,
                                              td31 - td11 = td32 - td12,
              td31 - td11 = td33 - td13,
                                              td31 - td11 = td34 - td14,
                                              td41 - td11 = td42 - td12,
              td41 - td11 = td43 - td13,
                                              td41 - td11 = td44 - td14 / print;
run;
quit;




/* Fit Model 3 generalized (i.e., no equality constraints imposed) */

proc reg data=t;
 M3Gen_SB: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 3 */
/* Set the following constraints to make the data set full rank */
 *Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
```

```
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;

/* coefficient on SBt=0. Row 255 */
SBt:   test SBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 256 */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
            td11+td12+td13+td14 = td31+td32+td33+td34,
                                          td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 257 */
dom:      test td11+td21+td31+td41 = td12+td22+td32+td42,
            td11+td21+td31+td41 = td13+td23+td33+td43,
                                          td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 258 */
indx: test tdi11 + tdi12 + tdi13 + tdi14 +
         tdi21 + tdi22 + tdi23 + tdi24 +
                            tdi31 + tdi32 + tdi33 + tdi34 +
                            tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;

/* trt*dom interaction.  Row 259 */
trt_dom:      test td21 - td11 = td22 - td12,
               td21 - td11 = td23 - td13,
               td21 - td11 = td24 - td14,
                                                   td31 - td11 = td32 - td12,
               td31 - td11 = td33 - td13,
                                                   td31 - td11 = td34 - td14,
                                                   td41 - td11 = td42 - td12,
               td41 - td11 = td43 - td13,
                                                   td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 260 */
indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
            tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                          tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 261 */
indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
            tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                          tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* Index*trt*dom -- many ways to express this 3-way interaction hypothesis. Row 262 */
indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
               tdi21 - tdi11 = tdi23 - tdi13,
               tdi21 - tdi11 = tdi24 - tdi14,
                                                   tdi31 - tdi11 = tdi32 - tdi12,
               tdi31 - tdi11 = tdi33 - tdi13,
                                                   tdi31 - tdi11 = tdi34 - tdi14,
                                                   tdi41 - tdi11 = tdi42 - tdi12,
               tdi41 - tdi11 = tdi43 - tdi13,
                                                   tdi41 - tdi11 = tdi44 - tdi14 / print;
run;
quit;
```

```
proc reg data=t;
 M3Gen_KB: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/*Set constraints for Model 3 */
/* Set the following constraints to make the data set full rank */
 *Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;

 /* coefficient on SBt=0. Row 306 */
KBt:   test KBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 307 */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
            td11+td12+td13+td14 = td31+td32+td33+td34,
                                         td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 308 */
dom:      test td11+td21+td31+td41 = td12+td22+td32+td42,
            td11+td21+td31+td41 = td13+td23+td33+td43,
                                         td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 309 */
indx: test tdi11 + tdi12 + tdi13 + tdi14 +
         tdi21 + tdi22 + tdi23 + tdi24 +
                           tdi31 + tdi32 + tdi33 + tdi34 +
                           tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;

/* trt*dom interaction.  Row 310 */
trt_dom:      test td21 - td11 = td22 - td12,
               td21 - td11 = td23 - td13,
               td21 - td11 = td24 - td14,
                                                     td31 - td11 = td32 - td12,
               td31 - td11 = td33 - td13,
                                                     td31 - td11 = td34 - td14,
                                                     td41 - td11 = td42 - td12,
               td41 - td11 = td43 - td13,
                                                     td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 311 */
indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
            tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                         tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 312 */
indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
            tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                         tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* Index*trt*dom -- many ways to express this 3-way interaction hypothesis. Row 313 */
indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
               tdi21 - tdi11 = tdi23 - tdi13,
               tdi21 - tdi11 = tdi24 - tdi14,
                                                     tdi31 - tdi11 = tdi32 - tdi12,
```

```
                tdi31 - tdi11 = tdi33 - tdi13,
                                                  tdi31 - tdi11 = tdi34 - tdi14,
                                                  tdi41 - tdi11 = tdi42 - tdi12,
                tdi41 - tdi11 = tdi43 - tdi13,
                                                  tdi41 - tdi11 = tdi44 - tdi14 / print;
run;
quit;

proc reg data=t;
 M3Gen_RM: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-
np4/noint;

/*Set constraints for Model 3 */
/* Set the following constraints to make the data set full rank */
 Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 *Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
/* Constrain np to be zero */
 Restrict np1=np2=np3=np4=0;

 /* coefficient on SBt=0. Row 357 */
RMt:   test RMt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 358 */
trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
             td11+td12+td13+td14 = td31+td32+td33+td34,
                                         td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 359 */
dom:      test td11+td21+td31+td41 = td12+td22+td32+td42,
             td11+td21+td31+td41 = td13+td23+td33+td43,
                                         td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 360 */
indx: test tdi11 + tdi12 + tdi13 + tdi14 +
          tdi21 + tdi22 + tdi23 + tdi24 +
                        tdi31 + tdi32 + tdi33 + tdi34 +
                        tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;

/* trt*dom interaction.  Row 361 */
trt_dom:      test td21 - td11 = td22 - td12,
                td21 - td11 = td23 - td13,
                td21 - td11 = td24 - td14,
                                                  td31 - td11 = td32 - td12,
                td31 - td11 = td33 - td13,
                                                  td31 - td11 = td34 - td14,
                                                  td41 - td11 = td42 - td12,
                td41 - td11 = td43 - td13,
                                                  td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 362 */
indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
             tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                         tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 363 */
```

```
indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
          tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                    tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* Index*trt*dom -- many ways to express this 3-way interaction hypothesis. Row 364 */
indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
             tdi21 - tdi11 = tdi23 - tdi13,
             tdi21 - tdi11 = tdi24 - tdi14,
                                         tdi31 - tdi11 = tdi32 - tdi12,
             tdi31 - tdi11 = tdi33 - tdi13,
                                         tdi31 - tdi11 = tdi34 - tdi14,
                                         tdi41 - tdi11 = tdi42 - tdi12,
             tdi41 - tdi11 = tdi43 - tdi13,
                                         tdi41 - tdi11 = tdi44 - tdi14 / print;
run;
quit;



/* Fit Model 4 generalized. */

proc reg data=t;
 M4_SBGen: model SBt1 = SBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

/* The following constraints make the design matrix full rank */
/* These are based on an assumption of no interaction */
 Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 *Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */



/* Construct type III tests (akin to rows 408-417 in Jill's spreadsheet). */

/* coefficient on SBt=0. row 408 */
 SBt:   test SBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 409 */
 trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
           td11+td12+td13+td14 = td31+td32+td33+td34,
                                   td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 410 */
 dom:     test td11+td21+td31+td41 = td12+td22+td32+td42,
          td11+td21+td31+td41 = td13+td23+td33+td43,
                                  td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 411 */
 indx: test tdi11 + tdi12 + tdi13 + tdi14 +
        tdi21 + tdi22 + tdi23 + tdi24 +
                      tdi31 + tdi32 + tdi33 + tdi34 +
                      tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;

/* trt*dom interaction. Row 412 */
trt_dom:      test td21 - td11 = td22 - td12,
             td21 - td11 = td23 - td13,
```

```
                td21 - td11 = td24 - td14,
                                                td31 - td11 = td32 - td12,
                td31 - td11 = td33 - td13,
                                                td31 - td11 = td34 - td14,
                                                td41 - td11 = td42 - td12,
                td41 - td11 = td43 - td13,
                                                td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 413 */
 indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
             tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                  tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 414 */
 indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
             tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                  tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* index*trt*dom:  there are many ways to express this 3-way interaction hypothesis. Row 415 */
 indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
                tdi21 - tdi11 = tdi23 - tdi13,
                tdi21 - tdi11 = tdi24 - tdi14,
                                                tdi31 - tdi11 = tdi32 - tdi12,
                tdi31 - tdi11 = tdi33 - tdi13,
                                                tdi31 - tdi11 = tdi34 - tdi14,
                                                tdi41 - tdi11 = tdi42 - tdi12,
                tdi41 - tdi11 = tdi43 - tdi13,
                                                tdi41 - tdi11 = tdi44 - tdi14 / print;

/* NP -- avg of the slopes = 0.  Row 416 */
 NP: test 0.25*np1 + 0.25*np2 + 0.25*np3 + 0.25*np4 = 0 / print;
             /* Note that the above is equivalent to np1 + np2 + np3 + np4 = 0, which is how I specify remaining tests */

/* test for equalitly of np slopes across the four treatments. Row 417 */
 NP_trt: test np1=np2=np3=np4 / print;

run;
quit;


proc reg data=t;
 M4_KBGen: model KBt1 = KBt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-np4/noint;

 /* The following constraints make the design matrix full rank */
 /* These are based on an assumption of no interaction */
 Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 *Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */


/* Construct type III tests (akin to rows 464-473 in Jill's spreadsheet). */

/* coefficient on SBt=0. row 464 */
 KBt:   test KBt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 465 */
```

```
 trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
            td11+td12+td13+td14 = td31+td32+td33+td34,
                                         td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 466 */
 dom:     test td11+td21+td31+td41 = td12+td22+td32+td42,
           td11+td21+td31+td41 = td13+td23+td33+td43,
                                        td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 467 */
 indx: test tdi11 + tdi12 + tdi13 + tdi14 +
          tdi21 + tdi22 + tdi23 + tdi24 +
                         tdi31 + tdi32 + tdi33 + tdi34 +
                         tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;

/* trt*dom interaction. Row 468 */
trt_dom:      test td21 - td11 = td22 - td12,
              td21 - td11 = td23 - td13,
              td21 - td11 = td24 - td14,
                                                 td31 - td11 = td32 - td12,
              td31 - td11 = td33 - td13,
                                                 td31 - td11 = td34 - td14,
                                                 td41 - td11 = td42 - td12,
              td41 - td11 = td43 - td13,
                                                 td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 469 */
 indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
            tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                         tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 470 */
 indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
            tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                         tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* Index*trt*dom: there are many ways to express this 3-way interaction hypothesis. Row 471 */
 indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
               tdi21 - tdi11 = tdi23 - tdi13,
               tdi21 - tdi11 = tdi24 - tdi14,
                                                  tdi31 - tdi11 = tdi32 - tdi12,
               tdi31 - tdi11 = tdi33 - tdi13,
                                                  tdi31 - tdi11 = tdi34 - tdi14,
                                                  tdi41 - tdi11 = tdi42 - tdi12,
               tdi41 - tdi11 = tdi43 - tdi13,
                                                  tdi41 - tdi11 = tdi44 - tdi14 / print;

/* NP -- avg of the slopes = 0.  Row 472 */
 NP: test 0.25*np1 + 0.25*np2 + 0.25*np3 + 0.25*np4 = 0 / print;
            /* Note that the above is equivalent to np1 + np2 + np3 + np4 = 0, which is how I specify remaining tests */

/* test for equalitly of np slopes across the four treatments. Row 473 */
 NP_trt: test np1=np2=np3=np4 / print;

run;
quit;
```

```
proc reg data=t;
 M4_RMGen: model RMt1 = RMt td11-td14 td21-td24 td31-td34 td41-td44 tdi11-tdi14 tdi21-tdi24 tdi31-tdi34 tdi41-tdi44 np1-
np4/noint;

/* The following constraints make the design matrix full rank */
/* These are based on an assumption of no interaction */
 Restrict 6*tdi12 = 2*tdi22 + 2*tdi32 + 2*tdi42 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */
 *Restrict 6*tdi14 = 2*tdi24 + 2*tdi34 + 2*tdi44 + tdi21+tdi31+tdi41+tdi23+tdi33+tdi43 - 3*tdi11 - 3*tdi13; /* Remove to use equality
constraint below. */


/* Construct type III tests (akin to rows 520-529 in Jill's spreadsheet). */

/* coefficient on SBt=0. row 520 */
 RMt:   test RMt=0 / print;

/* trtmnt main effect: i.e. intercepts averaged across dom are equal. Row 521 */
 trtmnt:   test td11+td12+td13+td14 = td21+td22+td23+td24,
              td11+td12+td13+td14 = td31+td32+td33+td34,
                                        td11+td12+td13+td14 = td41+td42+td43+td44 / print;

/* dom main effect: i.e. intercepts averaged across trtmnt are equal. Row 522 */
 dom:     test td11+td21+td31+td41 = td12+td22+td32+td42,
             td11+td21+td31+td41 = td13+td23+td33+td43,
                                       td11+td21+td31+td41 = td14+td24+td34+td44 / print;

/* index main effect: avg of slopes = 0, or equivalently that sum of slopes = 0. Row 523 */
 indx: test tdi11 + tdi12 + tdi13 + tdi14 +
           tdi21 + tdi22 + tdi23 + tdi24 +
                          tdi31 + tdi32 + tdi33 + tdi34 +
                          tdi41 + tdi42 + tdi43 + tdi44 = 0 / print;

/* trt*dom interaction. Row 524 */
trt_dom:      test td21 - td11 = td22 - td12,
                 td21 - td11 = td23 - td13,
                 td21 - td11 = td24 - td14,
                                              td31 - td11 = td32 - td12,
                 td31 - td11 = td33 - td13,
                                              td31 - td11 = td34 - td14,
                                              td41 - td11 = td42 - td12,
                 td41 - td11 = td43 - td13,
                                              td41 - td11 = td44 - td14 / print;

/* index*trtmnt interaction: lines are paralell after averaging across dom. Row 525 */
 indx_trt: test tdi11+tdi12+tdi13+tdi14 = tdi21+tdi22+tdi23+tdi24,
              tdi11+tdi12+tdi13+tdi14 = tdi31+tdi32+tdi33+tdi34,
                                        tdi11+tdi12+tdi13+tdi14 = tdi41+tdi42+tdi43+tdi44 / print;

/* index*dom interaction: lines are paralell after averaging across trtmnt. Row 526 */
 indx_dom: test tdi11+tdi21+tdi31+tdi41 = tdi12+tdi22+tdi32+tdi42,
              tdi11+tdi21+tdi31+tdi41 = tdi13+tdi23+tdi33+tdi43,
                                        tdi11+tdi21+tdi31+tdi41 = tdi14+tdi24+tdi34+tdi44 / print;

/* Index*trt*dom : there are many ways to express this 3-way interaction hypothesis. Row 527 */
 indx_trt_dom: test tdi21 - tdi11 = tdi22 - tdi12,
                 tdi21 - tdi11 = tdi23 - tdi13,
```

```
            tdi21 - tdi11 = tdi24 - tdi14,
                                        tdi31 - tdi11 = tdi32 - tdi12,
            tdi31 - tdi11 = tdi33 - tdi13,
                                        tdi31 - tdi11 = tdi34 - tdi14,
                                        tdi41 - tdi11 = tdi42 - tdi12,
            tdi41 - tdi11 = tdi43 - tdi13,
                                        tdi41 - tdi11 = tdi44 - tdi14 / print;

/* NP -- avg of the slopes = 0.  Row 528 */
 NP: test 0.25*np1 + 0.25*np2 + 0.25*np3 + 0.25*np4 = 0 / print;
          /* Note that the above is equivalent to np1 + np2 + np3 + np4 = 0 */

/* test for equalitly of np slopes across the four treatments. Row 529 */
 NP_trt: test np1=np2=np3=np4 / print;

run;
quit;




/* Model 5 Generalized is the same model as Model 3 Generalized */




/* Model 6 Generalized is the same model as Model 3 Generalized */
```

```
%let gtype = T;      ** options: M or T;     *** CHANGE INPUT FILE BLOCK BELOW TO CREATE NP_IN ***;
%let mtype = RS;     ** options: NC or RS;
%let ftype = F;      ** options: N (none) or F (float);
** Number of draws of component vector x defoliation index per combination of vegetation state x defoliation level (48
combinations);
**  Use draws = 1 for non-stochastic sample at center of distribution;
%let draws = 1000; *1000;
** Number of bootstrap samples per response component;
**  Use bsamples=1 for non-stochastic at mean of distribution;
%let bsamples = 500; *500;
%let rseed = 27695;

%let dir=c:\data\RCRP\NP\model empirical analysis\2016 data;

options ls=200 ps=60;

/**
  Model inputs:  Dominant invader (4 levels), NP level (4 levels), treatment (4 levels), defoliation level (3 levels)
  Procedure:
      1) Loop 1: Cycle over 4 treatments x 16 veg states x 3 defoliation levels (192 levels)
      2) Loop 2: Generate random 4-component (SB,KB,NP,RM) vectors and defoliation indices (0-7) consistent with each veg state
x defoliation level combination (x1000)
      3) For each sampled array of inputs (predictor values for the model), compute prediction mean and its variance
      4) Loop 3: Draw a sample of predictions from this normal distribution (x1000)
      5) Convert each prediction back to a 4-component vector and into its corresponding vegetation state
      6) Tabulate frequencies of veg state across samples of starting state and predicted value
**/

libname in "&dir";

/***
data m_nc;   ** Mixed, no constraints, no float;
  set in.m_sbnc (drop=SBt1 rename=(SBt=lagX))
      in.m_kbnc (drop=KBt1 rename=(KBt=lagX))
      in.m_rmnc (drop=RMt1 rename=(RMt=lagX));
  if _NAME_ in ("SBt", "KBt", "RMt") then _NAME_ = "lagX";
  gtype = 1;
  run;

data m_rs;   ** Mixed, response specific constraints, no float;
  set in.m_sbrs (drop=SBt1 rename=(SBt=lagX))
      in.m_kbrs (drop=KBt1 rename=(KBt=lagX))
      in.m_rmrs (drop=RMt1 rename=(RMt=lagX));
  if _NAME_ in ("SBt", "KBt", "RMt") then _NAME_ = "lagX";
  gtype = 1;
  run;

data mf_nc;   ** Mixed, no constraints, floating management year;
  set in.mf_sbnc (drop=SBt1 rename=(SBt=lagX))
      in.mf_kbnc (drop=KBt1 rename=(KBt=lagX))
      in.mf_rmnc (drop=RMt1 rename=(RMt=lagX));
  if _NAME_ in ("SBt", "KBt", "RMt") then _NAME_ = "lagX";
  gtype = 1;
  run;
***

data np_in;  *data mf_rs;   ** Mixed, response specific constraints, floating management year;
```

```
  set in.mf_sbrs (drop=SBt1 rename=(SBt=lagX))
      in.mf_kbrs (drop=KBt1 rename=(KBt=lagX))
      in.mf_rmrs (drop=RMt1 rename=(RMt=lagX));
  if _NAME_ in ("SBt", "KBt", "RMt") then _NAME_ = "lagX";
  gtype = 1;
  run;

/***
data t_nc;   ** Tall, no constraints, no float;
  set in.t_sbnc (drop=SBt1 rename=(SBt=lagX))
      in.t_kbnc (drop=KBt1 rename=(KBt=lagX))
      in.t_rmnc (drop=RMt1 rename=(RMt=lagX));
  if _NAME_ in ("SBt", "KBt", "RMt") then _NAME_ = "lagX";
  gtype = 2;
  run;

data t_rs;   ** Tall, response specific constraints, no float;
  set in.t_sbrs (drop=SBt1 rename=(SBt=lagX))
      in.t_kbrs (drop=KBt1 rename=(KBt=lagX))
      in.t_rmrs (drop=RMt1 rename=(RMt=lagX));
  if _NAME_ in ("SBt", "KBt", "RMt") then _NAME_ = "lagX";
  gtype = 2;
  run;

data tf_nc;   ** Tall, no constraints, floating management year;
  set in.tf_sbnc (drop=SBt1 rename=(SBt=lagX))
      in.tf_kbnc (drop=KBt1 rename=(KBt=lagX))
      in.tf_rmnc (drop=RMt1 rename=(RMt=lagX));
  if _NAME_ in ("SBt", "KBt", "RMt") then _NAME_ = "lagX";
  gtype = 2;
  run;
***/

data np_in;  *data tf_rs;   ** Tall, response specific constraints, floating management year;
  set in.tf_sbrs (drop=SBt1 rename=(SBt=lagX))
      in.tf_kbrs (drop=KBt1 rename=(KBt=lagX))
      in.tf_rmrs (drop=RMt1 rename=(RMt=lagX));
  if _NAME_ in ("SBt", "KBt", "RMt") then _NAME_ = "lagX";
  gtype = 2;
  run;

/***/

/**
proc contents data=m_nc;  run;
proc contents data=m_rs;  run;
proc contents data=mf_nc;  run;
proc contents data=mf_rs;  run;
proc contents data=t_nc;  run;
proc contents data=t_rs;  run;
proc contents data=tf_nc;  run;
proc contents data=tf_rs;  run;

proc print data=m_nc;  run;
proc print data=m_rs;  run;
proc print data=mf_nc;  run;
proc print data=mf_rs;  run;
```

```
proc print data=t_nc;  run;
proc print data=t_rs;  run;
proc print data=tf_nc;  run;
proc print data=tf_rs;  run;
**/

data np;
  /***
  set m_nc m_rs mf_nc (in=a) mf_rs (in=b)
      t_nc t_rs tf_nc (in=c) tf_rs (in=d);
  if (a | b | c | d) then ftype = 2;
  else ftype = 1;
  ***/
  set np_in;  ftype = 2;

  model = input(substr(_MODEL_,2,1),1.);
  if substr(_MODEL_,6,2)="nc" then mtype = 1;
  else mtype = 2;
  if _DEPVAR_="SBt1" then rtype = 1;
  else if _DEPVAR_="KBt1" then rtype = 2;
  else rtype = 3;
  if _TYPE_="PARMS" then ptype = 1;
  else ptype = 2;
  order = _n_;
  run;

proc sort data=np;
  by gtype ftype mtype model rtype ptype order;
  run;

/**
proc print data=np;  run;
**/

proc iml;

  ** Read in model estimates;
  cnames = {gtype ftype mtype model rtype ptype rmse lagx td11 td12 td13 td14 td21 td22 td23 td24 td31 td32 td33 td34 td41 td42
td43 td44
          tdi11 tdi12 tdi13 tdi14 tdi21 tdi22 tdi23 tdi24 tdi31 tdi32 tdi33 tdi34 tdi41 tdi42 tdi43 tdi44 np1 np2 np3 np4};
  use np;
  read all var {gtype ftype mtype model rtype ptype _RMSE_ lagx
          td11 td12 td13 td14 td21 td22 td23 td24 td31 td32 td33 td34 td41 td42 td43 td44
          tdi11 tdi12 tdi13 tdi14 tdi21 tdi22 tdi23 tdi24 tdi31 tdi32 tdi33 tdi34 tdi41 tdi42 tdi43 tdi44
          np1 np2 np3 np4} into est [colname=cnames];

  ** Indicator for grass type;
  if "&gtype" = "M" then do;
    grass_select = 1;
    nmodels = 4;
  end;
  else do;
    grass_select = 2;
    nmodels = 6;
  end;

  ** Indicator for model constraint type;
```

```
 if "&mtype" = "NC" then mtype_select = 1;
 else mtype_select = 2;

 ** Indicator for floating management year;
 if "&ftype" = "N" then ftype_select = 1;
 else ftype_select = 2;


 ** Generate input data as random draws of veg composition and defoliation index;
   ** Coefficients for random selection function for veg component vector;
   vegc = {1.8 1.2   0 2 2 1,
        1.8 1.2   0 2 1 1,
        1.8 1.2   0 2 0 1,
        1.8 1.2   2 1 0 3,
        1.35 0.45 0 2 2 1,
        1.35 0.45 0 2 1 1,
        1.35 0.45 0 2 0 1,
        1.35 0.45 2 1 0 3,
        0.9  0.45 0 2 2 1,
        0.9  0.45 0 2 1 1,
        0.9  0.45 0 2 0 1,
        0.9  0.45 2 1 0 3,
        0    0.9  0 2 2 1,
        0    0.9  0 2 1 1,
        0    0.9  0 2 0 1,
        0    0.9  2 1 0 3};
   vegc = vegc/3;

   ** Quantiles for defoliation index;
   dq1 = {0 1 7 8 23 28 28 28 28 28 28 28 28 28 28 28,
       6.668019 5.715445 5.284782 4.762871 4.403985 . . . . . . . . . . .};
   dq1[1,] = dq1[1,]/28;
   dq2 = {0 20 21 31 35 50 60 66 67 67 67 67 67 67 67 67,
       3.810297 3.655293 3.523188 2.924234 2.857722 2.642391 2.193176 2 . . . . . . . .};
   dq2[1,] = dq2[1,]/67;
   dq3 = {0 6 11 14 18 21 22 23 24 25 27 28 29 30 31 32,
       1.905148 1.761594 1.5 1.462117 1 0.952574 0.880797 0.806824 0.731059 0.537883 0.5 0.268941 0.238406 0.119203
0.047426 0};
   dq3[1,] = dq3[1,]/33;
   dq = dq1 // dq2 // dq3;

   ** Draw uniform variates;
   draws = (uniform(j(&draws*48,4,&rseed)) * (&draws^=1)) + (j(&draws*48,4,0.5) * (&draws=1));
   ** Columns of vxd are: veg state, defoliation level, NP, SB, KB, RM, defoliation index;
   vxd = j(&draws*48,7);

   ** Cycle over 48 levels of veg state x defoliation;
   ** Veg states: {NP 60+ (1-4), NP 45-60 (5-8), NP 30-45 (9-12), NP <30 (13-16)} x {SB, Co, KB, RM};
   ** Defoliation levels: {4+ (1), 2-4 (2), <2 (3)};
   do v=1 to 16;
    do d=1 to 3;

     ** Draw a random veg component vector and defoliation index for given veg state and defol level;
     draw_index = (v-1)*3 + d;
     pullrange = ((draw_index-1)*&draws+1:draw_index*&draws);
     drawsel = draws[pullrange,];
```

```
    ** Use coefficients above to constrain random selection of NP, RM, and SB;
    np0 = vegc[v,1] + vegc[v,2]*drawsel[,1];
    rm0 = (1-np0) # (vegc[v,3] + vegc[v,4]*drawsel[,2]);
    sb0 = (1-np0-rm0) # (vegc[v,5] + vegc[v,6]*drawsel[,3]);
    kb0 = 1-np0-rm0-sb0;
    vegdraw = np0 || sb0 || kb0 || rm0;

    ** Draw a random defoliation index from quantiles above;
    x = 16 - ( dq[(d-1)*2+1,] @ j(&draws,1,1) > drawsel[,4] @ j(1,16,1) )[,+];
    defdraw = dq[(d-1)*2+2,x]`;
    vxd[pullrange,] = (j(&draws,1,1) @ (v || d)) || vegdraw || defdraw;

   end;
  end;

** Create design matrix based on combination of treatments, veg composition, defoliation index;
  ** Create an input data matrix by combining vegetation data with treatment levels;
  ** 1=Burn/Graze, 2=Burn, 3=Graze, 4=Rest for Mixed;
  ** 1=BurnWW, 2=GrazeWW, 3=Defoliate, 4=Rest for Tall;
  indata = (vxd[,1:2] @ j(4,1,1)) || shape((1:4)`,48*4*&draws,1) || (vxd[,3:7] @ j(4,1,1));

  ** Indicator for dominant grass type;
  domtype = mod(indata[,1],4) + 4*(mod(indata[,1],4)=0);

  ** Create the pivot logits for SB, KB, RM (lagged grass component in design matrix);
  z = log(indata[,5:7]/indata[,4]);
  ** Create the logits for NP;
  logitnp = log(indata[,4]/(1-indata[,4]));

  ** Create the generic design matrix (all effects except lagged grass component);
  x1 = design((indata[,3]-1)*4 + domtype);
  x2 = x1 # indata[,8];
  x3 = design(indata[,3]) # logitnp;
  x = x1 || x2 || x3;

  ** An index for the veg state x defoliation x treatment combination;
  scenario = (indata[,1]-1)*3*4 + (indata[,2]-1)*4 + indata[,3];
        groupmean = design(scenario);

  ** Accumulate the frequency of state outcomes here;
  outcount = j(nmodels*(16*3*4),21,0);

** For the given grass type, float type, and model constraint type, loop over alternative models;
do model=1 to nmodels;

 ** Extract parameter matrices according to grass type, float type, model type, and model;
 ** Make a parameter vector and a V-C matrix for each response;
 select = (est[,1]=grass_select) # (est[,2]=ftype_select) # (est[,3]=mtype_select) # (est[,4]=model);
 sb_p = est[loc(select # (est[,5]=1) # (est[,6]=1)>0),8:44];
 sb_v = est[loc(select # (est[,5]=1) # (est[,6]=2)>0),8:44];
 sb_e = est[loc(select # (est[,5]=1) # (est[,6]=1)>0),7];
 kb_p = est[loc(select # (est[,5]=2) # (est[,6]=1)>0),8:44];
 kb_v = est[loc(select # (est[,5]=2) # (est[,6]=2)>0),8:44];
 kb_e = est[loc(select # (est[,5]=2) # (est[,6]=1)>0),7];
 rm_p = est[loc(select # (est[,5]=3) # (est[,6]=1)>0),8:44];
 rm_v = est[loc(select # (est[,5]=3) # (est[,6]=2)>0),8:44];
 rm_e = est[loc(select # (est[,5]=3) # (est[,6]=1)>0),7];
```

```
  ** Predict outcome means for each component;
  m = ((z[,1] || x) * sb_p`) || ((z[,2] || x) * kb_p`) || ((z[,3] || x) * rm_p`);

  ** Prediction variances;
  v = j(192*&draws,3,0);
  do i=1 to 192*&draws;
   v[i,1] = (z[i,1] || x[i,]) * sb_v * ((z[i,1] || x[i,])`)*sb_e*sb_e;
   v[i,2] = (z[i,2] || x[i,]) * kb_v * ((z[i,2] || x[i,])`)*kb_e*kb_e;
   v[i,3] = (z[i,3] || x[i,]) * rm_v * ((z[i,3] || x[i,])`)*rm_e*rm_e;
  end;

        ** Obtain average prediction SE over samples;
        meanse = groupmean`*sqrt(v) / (groupmean`*j(192*&draws,3,1));

        ** Obtain geometric mean of prediction SE over components;
        geomeanse = meanse[,#] ## (1/3);
        print model meanse geomeanse;

/**
  v11 = (z[,1] || x) * sb_v;
  v1 = vecdiag(v11 * (z[,1] || x)`)*sb_e*sb_e;
  v21 = (z[,2] || x) * kb_v;
  v2 = vecdiag(v21 * (z[,2] || x)`)*kb_e*kb_e;
  v31 = (z[,3] || x) * rm_v;
  v3 = vecdiag(v31 * (z[,3] || x)`)*rm_e*rm_e;
  v = v1 || v2 || v3;
*   v = vecdiag((z[,1] || x) * sb_v * (z[,1] || x)`)*sb_e*sb_e ||
    vecdiag((z[,2] || x) * kb_v * (z[,2] || x)`)*kb_e*kb_e ||
    vecdiag((z[,3] || x) * rm_v * (z[,3] || x)`)*rm_e*rm_e;
**/

  boot0 = indata[,1:3] || m || v;

  ** Bootstrap simulation for generating distribution of outcome states;
    do scen=1 to 192;   ** Select 1 veg x defoliation x treatment scenario at a time;

     ** Construct submatrix of means and variances for this combination;
     boot = boot0[loc(scenario=scen),];

     ** Draw a random matrix of normals;
     b = normal(j(&draws*&bsamples,3,&rseed));

     ** Expand the rows of submatrix to accommodate random matrix;
     boot = boot @ j(&bsamples,1,1);

     ** Generate a random prediction for each component;
     p = (sqrt(boot[,7:9]) # b)*(&bsamples^=1) + boot[,4:6];

     ** Back-transform;
     bp = exp(p) / (1 + exp(p)[,+]);
     ** Components of PRED:  NP (1), SB (2), KB (3), RM (4);
     pred = (1-bp[,+]) || bp;

     ** State assignment of prediction;
     r = j(&draws*&bsamples,3,0);
     np_level = 5 - (1 + (pred[,1]>=0.3) + (pred[,1]>=0.45) + (pred[,1]>=0.6));
```

```
        ** elementwise test of divisibility -- q holds 1-pred[,1] if pred[,1]<1, or -1 otherwise;
        q = (pred[,1]<1.0)#(1-pred[,1]) - (pred[,1]>=1.0);
        r[,3] = (pred[,4]/q)#(q>0);
        ** elementwise test of divisibility -- q holds pred[,2]+pred[,3] if sum>0, or -1 otherwise;
        xsum = pred[,2] + pred[,3];
        q = (xsum>0)#xsum - (xsum<=0);
        r[,1] = (pred[,2]/q)#(q>0);
        r[,2] = (1-r[,1])#(q>0);
        dom = 4 * (r[,3] >= 2/3) +
             1 * (2/3 >= r[,3]) # (r[,1] >= 2/3) +
             3 * (2/3 >= r[,3]) # (r[,2] >= 2/3) +
             2 * (2/3 >= r[,3]) # (2/3 >= r[,1]) # (2/3 >= r[,2]);
        pred_state = 4*(np_level-1) + dom;

        ** Frequency of outcome states;
        counts = ((pred_state @ j(1,16,1)) = ( (1:16) @ j(&draws*&bsamples,1,1) ))[+,];
        outcount[(model-1)*192 + scen,] = ( model || boot[1,(2||3||1)] || geomeanse[scen] ||

counts/(&draws*&bsamples) );

      end;

/****
      ** Draw a random matrix of normals;
      b = normal(j(48*4*&draws*&bsamples,3,&rseed));

      ** Expand prediction matrix to size of random matrix;
      boot = (indata[,1:3] || m || v) @ j(&bsamples,1,1);

      ** Generate a random prediction for each component;
      p = (sqrt(boot[,7:9]) # b)*(&bsamples^=1) + boot[,4:6];

      ** Back-transform;
      bp = exp(p) / (1 + exp(p)[,+]);
      ** Components of PRED:  NP (1), SB (2), KB (3), RM (4);
      pred = (1-bp[,+]) || bp;

      ** State assignment of prediction;
      r = j(48*4*&draws*&bsamples,3,0);
      np_level = 5 - (1 + (pred[,1]>=0.3) + (pred[,1]>=0.45) + (pred[,1]>=0.6));
      ** elementwise test of divisibility -- q holds 1-pred[,1] if pred[,1]<1, or -1 otherwise;
      q = (pred[,1]<1.0)#(1-pred[,1]) - (pred[,1]>=1.0);
      r[,3] = (pred[,4]/q)#(q>0);
      ** elementwise test of divisibility -- q holds pred[,2]+pred[,3] if sum>0, or -1 otherwise;
      xsum = pred[,2] + pred[,3];
      q = (xsum>0)#xsum - (xsum<=0);
      r[,1] = (pred[,2]/q)#(q>0);
      r[,2] = (1-r[,1])#(q>0);
      dom = 4 * (r[,3] >= 2/3) +
           1 * (2/3 >= r[,3]) # (r[,1] >= 2/3) +
           3 * (2/3 >= r[,3]) # (r[,2] >= 2/3) +
           2 * (2/3 >= r[,3]) # (2/3 >= r[,1]) # (2/3 >= r[,2]);
      pred_state = 4*(np_level-1) + dom;

      ** Frequency of outcome states;
      outstate = boot[,1:3] || pred_state;
      scenario = (boot[,1]-1)*3*4 + (boot[,2]-1)*4 + boot[,3];
```

```
        counts = design(scenario)` * design(pred_state);

        indices = j(192,3,0);
        indices[,1] = floor(((1:192)`-0.1)/12)+1;
        indices[,2] = mod(floor(((1:192)`-0.1)/4)+1,3) + 3*(mod(floor(((1:192)`-0.1)/4)+1,3)=0);
        indices[,3] = mod((1:192)`,4) + 4*(mod((1:192)`,4)=0);

****/



  end;

  cname = {model dl t vs gmse s60sb s60co s60kb s60rm s45sb s45co s45kb s45rm s30sb s30co s30kb s30rm s0sb s0co s0kb
s0rm};
  create in.countsb3_&gtype._&ftype._&mtype from outcount [colname=cname];
  append from outcount;

  quit;

  /**********/

proc format;
  value defolfmt 1='<2' 2='2-4' 3='>4';
  value trtfmt_m 1="Rest" 2="Graze" 3="Burn" 4="B/G";
  value trtfmt_t 1="Rest" 2="GrazeW" 3="BurnW" 4="Defol";
  value statefmt  1="{NP 60-100, SB}"
                  2="{NP 60-100, Co}"
                  3="{NP 60-100, KB}"
                  4="{NP 60-100, RM}"
                  5="{NP 45-60, SB}"
                  6="{NP 45-60, Co}"
                  7="{NP 45-60, KB}"
                  8="{NP 45-60, RM}"
                  9="{NP 30-45, SB}"
                 10="{NP 30-45, Co}"
                 11="{NP 30-45, KB}"
                 12="{NP 30-45, RM}"
                 13="{NP 0-30, SB}"
                 14="{NP 0-30, Co}"
                 15="{NP 0-30, KB}"
                 16="{NP 0-30, RM}";
  run;

data a;
  retain model defolx trtmnt vegstate gmse;
  set in.countsb3_&gtype._&ftype._&mtype;
  array s[2,4] _temporary_ (4 3 2 1, 4 2 1 3);
  if "&gtype"="M" then ts = s[1,t];
  if "&gtype"="T" then ts = s[2,t];
  trtmnt = put(ts,trtfmt_&gtype..);
  vegstate = put(vs,statefmt.);
  defolx = put(dl,defolfmt.);
  drop t;
  run;

proc sort data=a out=in.countsb3_&gtype._&ftype._&mtype;
```

```
  by model dl ts vs;
  run;

proc print data=in.countsb3_&gtype._&ftype._&mtype;
  title1 "Estimated transition probabilities for grass type &gtype, model type &mtype";
  title2 "Number of input draws = &draws, Number of bootstrap predictions = &bsamples";
  by model dl ts;
  id model dl ts;
  var vs gmse s60sb s60co s60kb s60rm s45sb s45co s45kb s45rm s30sb s30co s30kb s30rm s0sb s0co s0kb s0rm;
  label model="Model" dl="Defoliation Level" ts="Treatment" vs="Veg State" gmse="GM Pred SE";
  format dl defolfmt. ts trtfmt_&gtype.. vs statefmt.;
  run;

title1;
title2;

data a; set in.countsb3_&gtype._&ftype._&mtype (where=(model=1));  run;
proc export data=a (drop=model dl ts vs)
    outfile="&dir\outb3_&gtype._&ftype._&mtype..xlsx"
    dbms=xlsx label replace;
  sheet="&gtype._m1";
  newfile=yes;
  run;

data a; set in.countsb3_&gtype._&ftype._&mtype (where=(model=2));  run;
proc export data=a (drop=model dl ts vs)
    outfile="&dir\outb3_&gtype._&ftype._&mtype..xlsx"
    dbms=xlsx label replace;
  sheet="&gtype._m2";
  newfile=no;
  run;

data a; set in.countsb3_&gtype._&ftype._&mtype (where=(model=3));  run;
proc export data=a (drop=model dl ts vs)
    outfile="&dir\outb3_&gtype._&ftype._&mtype..xlsx"
    dbms=xlsx label replace;
  sheet="&gtype._m3";
  newfile=no;
  run;

data a; set in.countsb3_&gtype._&ftype._&mtype (where=(model=4));  run;
proc export data=a (drop=model dl ts vs)
    outfile="&dir\outb3_&gtype._&ftype._&mtype..xlsx"
    dbms=xlsx label replace;
  sheet="&gtype._m4";
  newfile=no;
  run;

data a; set in.countsb3_&gtype._&ftype._&mtype (where=(model=5));  run;
proc export data=a (drop=model dl ts vs)
    outfile="&dir\outb3_&gtype._&ftype._&mtype..xlsx"
    dbms=xlsx label replace;
  sheet="&gtype._m5";
  newfile=no;
  run;

data a; set in.countsb3_&gtype._&ftype._&mtype (where=(model=6));  run;
```

```
proc export data=a (drop=model dl ts vs)
     outfile="&dir\outb3_&gtype._&ftype._&mtype..xlsx"
     dbms=xlsx label replace;
  sheet="&gtype._m6";
  newfile=no;
  run;

quit;


  /**********/
```

```
options ls=200 ps=60;

%let dir=C:\data\RCRP\NP\abiotic;
%let trtfile = all_float;    ** ALL or ALL_FLOAT **;

filename outdata "&dir";
libname in "&dir\data2016";

title1;

** Read main file of precipitation data **;
proc import out=work.precip (rename=(unit_name=u_name transect_name=t_name))
            datafile= "&dir\precipitation\Transects_Precipitation(sep07-aug16)_namesfixed.xlsx"
            dbms=xlsx replace;
  sheet="NPAM_Transects_2009to2016_Preci";
  getnames=yes;
  *mixed=no;
  *scantext=yes;
  *usedate=yes;
  *scantime=yes;
  run;
/**
** Read file of precip data for recently added transects **;
proc import out=work.precip_add (rename=(unit_name=u_name transect_n=t_name))
            datafile= "&dir\precipitation\Transects_CenterPoints_Missing_2014.xlsx"
            dbms=excel replace;
  range="Transects_CenterPoints_Missing_$";
  getnames=yes;
  mixed=no;
  scantext=yes;
  usedate=yes;
  scantime=yes;
  run;
**/
** Read file of long term climate data **;
proc import out=work.climate (rename=(unit_name=u_name transect_name=t_name))
            datafile= "&dir\precipitation\Transects_LTPrecip_Temp(Update2016)_namesfixed.xlsx"
            dbms=xlsx replace;
  sheet="NPAM_Transects_2009to2016_LTPT";
  getnames=yes;
  *mixed=no;
  *scantext=yes;
  *usedate=yes;
  *scantime=yes;
  run;

** Read from the ESD data workbook;
proc import out=work.esd (rename=(transectID=transectIDchar unit_name=u_name transect_name=t_name
                                  sumInvasive=esdI sumMaintain=esdM sumNA=esdNA))
            datafile= "&dir\ESD\Transects_ESDClassified_I%nrstr(&M)(sum)_final_24Nov2014_namesfixed.xlsx"
            dbms=xlsx replace;
  sheet="Sum_I%nrstr(&M)";
  getnames=yes;
  *mixed=no;
  *scantext=yes;
  *usedate=yes;
  *scantime=yes;
```

```
  run;
/**
** Rename variables in the precip_add data to allow concatenation with precip **;
data precip_add;
  set precip_add;
  array pr1[83] sep07s_ oct07s nov07s dec07s
                jan08s feb08s mar08s apr08s_ may08s jun08s jul08s aug08s sep08s oct08s nov08s dec08s
                jan09s feb09s mar09s apr09s may09s jun09s jul09s aug09s sep09s oct09s nov09s dec09s
                jan10s feb10s mar10s apr10s may10s jun10s jul10s aug10s sep10s oct10s nov10s dec10s
                jan11s feb11s mar11s apr11s may11s jun11s jul11s aug11s sep11s oct11s nov11s dec11s
                jan12s feb12s mar12s apr12s may12s jun12s jul12s aug12s sep12s oct12s nov12s dec12s
                jan13s feb13s mar13s apr13s may13s jun13s jul13s aug13s sep13s oct13s nov13s dec13s
                jan14s feb14s mar14s apr14s may14s jun14s jul14s__;
  array pr2[83] sep07 oct07 nov07 dec07
                jan08 feb08 mar08 apr08 may08 jun08 jul08 aug08 sep08 oct08 nov08 dec08
                jan09 feb09 mar09 apr09 may09 jun09 jul09 aug09 sep09 oct09 nov09 dec09
                jan10 feb10 mar10 apr10 may10 jun10 jul10 aug10 sep10 oct10 nov10 dec10
                jan11 feb11 mar11 apr11 may11 jun11 jul11 aug11 sep11 oct11 nov11 dec11
                jan12 feb12 mar12 apr12 may12 jun12 jul12 aug12 sep12 oct12 nov12 dec12
                jan13 feb13 mar13 apr13 may13 jun13 jul13 aug13 sep13 oct13 nov13 dec13
                jan14 feb14 mar14 apr14 may14 jun14 jul14;
  do i=1 to 83;
    pr2[i] = pr1[i];
  end;
  if scan(u_name,1)="WAUBAY" then delete;  ** Records already in main file;
  if find(u_name,"Peterson") then delete;  ** Record already in main file;
  run;

data precip;
  set precip precip_add;
  run;
**/
proc sort data=precip;  by u_name t_name;  run;

proc sort data=climate;  by u_name t_name;  run;

** Merge precipitation and climate data **;
data precip;
  merge precip climate (keep=u_name t_name gsp map mtwm maxt);
  by u_name t_name;
  run;

proc sort data=esd;  by u_name t_name;  run;

** Merge precipitation/climate and ESD data at the transect level **;
data both /*(where=(in_MN=0))*/;
  merge precip (in=one) esd (in=two);
  by u_name t_name;
  if u_name="" then delete;
  if one & ^two then precip_only = 1;
  else precip_only = 0;
  if two & ^one then esd_only = 1;
  else esd_only = 0;
  if scan(u_name,1) in("BIG","DETROIT","MORRIS","WINDOM") then in_MN = 1;  ** Identify Minnesota units **;
  else in_MN = 0;
  if scan(u_name,1)="SOURIS" & find(u_name,"HB 7") then delete;  ** Get rid of duplicate unit **;
  t_temp = t_name;
```

```
  if find(t_temp,"Peterson SW- 17") then do;
    u_name = substr(u_name,1,43) || "South";
    t_name = substr(t_temp,1,43) || "South- " || substr(t_temp,48,2);
  end;
  else t_name = t_temp;
  if find(u_name,"Peterson SE") | find(u_name,"Peterson SW") then delete;
  if find(u_name,"Lasher Unit A") & (find(t_name,"11") | find(t_name,"22") | find(t_name,"30")) then delete;
**  if find(u_name,"Lasher Unit A") & (find(t_name,"11") | find(t_name,"22") | find(t_name,"30")) then do;
    ** These transects reassigned in 2013 **;
**    substr(u_name,51,1) = "B";
**    substr(t_name,51,1) = "B";
**  end;
  ** For units split in 2015, back-propagate unit names into previous years **;
  delim = ':';
  if strip(scan(u_name,3,delim))="G28" & input(substr(t_name,40,2),?? 2.) in (0,3,6,9,10,12,13,19,22,25,29,30,32,35,36,39,44,46)
then do;
    u_name = substr(u_name,1,37) || " North";
		t_name = substr(t_name,1,37) || " North- " || substr(t_name,40,2);
		end;
  if strip(scan(u_name,3,delim))="G28" & input(substr(t_name,40,2),?? 2.) in
(20,21,23,24,26,27,28,31,33,34,37,38,40,41,42,43,45,47,48) then do;
    u_name = substr(u_name,1,37) || " South";
		t_name = substr(t_name,1,37) || " South- " || substr(t_name,40,2);
		end;
  if strip(scan(u_name,3,delim))="Woodworth Station Unit 7" & input(substr(t_name,67,3),?? 3.) in
(5,10,12,15,21,30,31,32,47,177,191,198) then do;
    u_name = substr(u_name,1,64) || " East";
		t_name = substr(t_name,1,64) || " East- " || substr(t_name,67,3);
		end;
  if strip(scan(u_name,3,delim))="Woodworth Station Unit 7" & input(substr(t_name,67,3),?? 3.) in
(1,2,4,7,8,9,11,18,19,22,23,24,25,26,27,28,29,33,42,170,174,175,178,181,193,194,210,216) then do;
    u_name = substr(u_name,1,64) || " West";
		t_name = substr(t_name,1,64) || " West- " || substr(t_name,67,3);
		end;
  ** This unit was re-named in 2015 **;
  if strip(scan(u_name,3,delim))="Hepner WPA" then do;
    u_name = substr(u_name,1,37);
		t_name = substr(t_name,1,37) || substr(t_name,42,4);
		end;
  ** These units were combined in 2014 -- split out again for analysis **;
  /**
  if strip(scan(u_name,3,delim))="Buss" & input(substr(t_name,50,2),?? 2.) in (10,11,15,16,25,26,27,28,29,30,31,6,7,9) then do;
    u_name = substr(u_name,1,45) || " Paddock 1";
		t_name = substr(t_name,1,45) || " Paddock 1- BU" || substr(t_name,50,2);
		end;
  if strip(scan(u_name,3,delim))="Buss" & input(substr(t_name,50,2),?? 2.) in (1,12,17,18,19,20,21,22,23,24,5) then do;
    u_name = substr(u_name,1,45) || " Paddock 2";
		t_name = substr(t_name,1,45) || " Paddock 2- BU" || substr(t_name,50,2);
		end;
  **/
  if strip(scan(u_name,3,delim))="Buss" then delete;
  ** Remove redundant records **;
  if strip(scan(u_name,3,delim)) in("G28 North","G28 South","Woodworth Station Unit 7 East","Woodworth Station Unit 7
West","Hepner") & precip_only=1 then delete;
  run;

proc sort data=both;  by u_name t_name;  run;
```

```
/**
proc print data=both;  var u_name t_name in_MN precip_only esd_only sep07 esdI esdM esdNA;  run;
**/

** Obtain unit-level averages for precipitation and climate data **;
proc summary data=both mean nway;
  class u_name;
  var sep07 oct07 nov07 dec07
      jan08 feb08 mar08 apr08 may08 jun08 jul08 aug08 sep08 oct08 nov08 dec08
      jan09 feb09 mar09 apr09 may09 jun09 jul09 aug09 sep09 oct09 nov09 dec09
      jan10 feb10 mar10 apr10 may10 jun10 jul10 aug10 sep10 oct10 nov10 dec10
      jan11 feb11 mar11 apr11 may11 jun11 jul11 aug11 sep11 oct11 nov11 dec11
      jan12 feb12 mar12 apr12 may12 jun12 jul12 aug12 sep12 oct12 nov12 dec12
      jan13 feb13 mar13 apr13 may13 jun13 jul13 aug13 sep13 oct13 nov13 dec13
      jan14 feb14 mar14 apr14 may14 jun14 jul14 aug14 sep14 oct14 nov14 dec14
      jan15 feb15 mar15 apr15 may15 jun15 jul15 aug15 sep15 oct15 nov15 dec15
      jan16 feb16 mar16 apr16 may16 jun16 jul16 aug16
      gsp map mtwm maxt;
  output out=precip_means mean=;
  run;
/**
proc summary data=climate mean nway;
  class u_name;
  var gsp map mtwm maxt;
  output out=climate_means mean=;
  run;
**/

** Merge transect-level ESD data with unit-level climate/precip data **
**   Create accumulated precipitation variables **;
data a;
  merge both (keep=u_name t_name azimuth slope aspect esdI esdM esdNA in=one)
        precip_means;
  by u_name;
  length unit $ 40 transect $ 8;
  delim = ':';
  unit = left(scan(u_name,3,delim));
  transect = left(scan(t_name,-1));
*  retain unit 0;
*  if first.u_name then unit+1;
*  unit_tran = _n_;
  if sum(esdI, esdM)>=90 then do;
    if esdI=. then pInv = 0;
    else if esdM=. then pInv = 1;
    else pInv = esdI/sum(esdI, esdM);
  end;
  else pInv = .;
  jan07 = .; feb07 = .; mar07 = .; apr07 = .; may07 = .; jun07 = .; jul07 = .; aug07 = .;
  sep16 = .; oct16 = .; nov16 = .; dec16 = .;
  array pr[10,12] jan07 feb07 mar07 apr07 may07 jun07 jul07 aug07 sep07 oct07 nov07 dec07
                  jan08 feb08 mar08 apr08 may08 jun08 jul08 aug08 sep08 oct08 nov08 dec08
                  jan09 feb09 mar09 apr09 may09 jun09 jul09 aug09 sep09 oct09 nov09 dec09
                  jan10 feb10 mar10 apr10 may10 jun10 jul10 aug10 sep10 oct10 nov10 dec10
                  jan11 feb11 mar11 apr11 may11 jun11 jul11 aug11 sep11 oct11 nov11 dec11
                  jan12 feb12 mar12 apr12 may12 jun12 jul12 aug12 sep12 oct12 nov12 dec12
                  jan13 feb13 mar13 apr13 may13 jun13 jul13 aug13 sep13 oct13 nov13 dec13
                  jan14 feb14 mar14 apr14 may14 jun14 jul14 aug14 sep14 oct14 nov14 dec14
```

```
                  jan15 feb15 mar15 apr15 may15 jun15 jul15 aug15 sep15 oct15 nov15 dec15
                  jan16 feb16 mar16 apr16 may16 jun16 jul16 aug16 sep16 oct16 nov16 dec16;
  array prsum[2,8] psa08 psa09 psa10 psa11 psa12 psa13 psa14 psa15
                   psg09 psg10 psg11 psg12 psg13 psg14 psg15 psg16;
  ** Nearly all precip values for the following unit were 0.00, so set all to missing **;
  if find(u_name,"Touch The Sky") then do;
    do i=1 to 10;
                  do j=1 to 12;
                    pr[i,j] = .;
                                 end;
                  end;
                end;
  ** Create season precipitation totals for prior year (psa) and growing season (psg) **;
  do i=1 to 8;
    prsum[1,i] = 0;
    prsum[2,i] = 0;
    do j=1 to 12;
      prsum[1,i] = prsum[1,i] + pr[i+1,j];
      if j>=7 then prsum[2,i] = prsum[2,i] + pr[i+1,j];
      else prsum[2,i] = prsum[2,i] + pr[i+2,j];
    end;
  end;
/***
  do year=2009 to 2014;
    unit_year = (unit-1)*6 + year-2008;
    ** Make up data for treatment;
    if year=2009 then trt = .;
    else trt = int(ranuni(12345)*4)+1;
    rain_prior = prsum[1,year-2008];
    rain_grow = prsum[2,year-2008];
    ** Make up data for NP count;
    seed = 12345;
    stops = 50;
    pnative = ranuni(seed);
    call ranbin(seed,stops,pnative,np);
    output;
  end;
***/
  run;

/**
proc print data=a;  var u_name t_name gsp map mtwm maxt;  run;
**/

** Merge unit-level files of vegetation data **;
data unit_level;
  set in.all_float_2016_mixed (in=one) in.all_float_2016_tall;
  if one then gtype = "M";
  else gtype = "T";
  run;

proc sort data=unit_level;  by unit year;  run;

** Merge transect-level files of vegetation data **;
data transect_level;
  set in.complete_float_2016_mixed in.complete_float_2016_tall;
  run;
```

```
proc sort data=transect_level;  by unit year;  run;

** Merge unit-level and transect-level vegetation files **;
data monitor (keep=gtype unit yr trtmnt transect np sb kb rm nonmatched rename=(yr=year));
  merge unit_level (in=one drop=sb kb np rm) transect_level (in=two drop=trtmnt index);
  by unit year;
  yr = input(year,4.);
  unit = left(unit);
  transect = left(transect);
  if scan(unit,1)="Thacker" then unit = "Thacker WPA - " || scan(unit,3);  ** Reformat a non-matching string **;
  if unit="Peterson SE" | unit="Peterson SW" then unit = "Peterson South";  ** Units combined in 2013 **;
  if unit="Lasher Unit A" & transect in ("11", "22", "30") then unit = "Lasher Unit B"; ** Transects reassigned in 2013 **;
  ** Take care of units split in 2015 (remove redundant records inserted for 2014) **;
  if scan(unit,1)="G28" & yr<=2014 then do;
              if input(transect,?? 2.) in (0,3,6,9,10,12,13,19,22,25,29,30,32,35,36,39,44,46) then do;
     if unit="G28 North" & yr=2014 then delete;
                unit = "G28 North";
                end;
    if input(transect,?? 2.) in (20,21,23,24,26,27,28,31,33,34,37,38,40,41,42,43,45,47,48) then do;
                if unit="G28 South" & yr=2014 then delete;
     unit = "G28 South";
                end;
              end;
  if scan(unit,1)="Woodworth" & yr<=2014 then do;
    if input(transect,?? 3.) in (5,10,12,15,21,30,31,32,47,177,191,198) then do;
                if scan(unit,5)="East" & yr=2014 then delete;
                unit = "Woodworth Station Unit 7 East";
                end;
    if input(transect,?? 3.) in (1,2,4,7,8,9,11,18,19,22,23,24,25,26,27,28,29,33,42,170,174,175,178,181,193,194,210,216) then do;
                if scan(unit,5)="West" & yr=2014 then delete;
                unit = "Woodworth Station Unit 7 West";
                end;
              end;
  ** Take care of unit re-named in 2015 and remove redundant records inserted for 2014 **;
  if scan(unit,1)="Hepner" & yr<=2014 then do;
    if unit="Hepner" & yr=2014 then delete;
              unit = "Hepner";
              end;
  ** Treat units combined in 2014 as split units **;
  if unit="Buss" then do;
    if yr=2013 then delete;  ** Remove redundant records inserted for 2013 **;
    if input(substr(transect,3,2),?? 2.) in (10,11,15,16,25,26,27,28,29,30,31,6,7,9) then unit = "Buss Paddock 1";
              if input(substr(transect,3,2),?? 2.) in (1,12,17,18,19,20,21,22,23,24,5) then unit = "Buss Paddock 2";
              end;
  if one+two<2 then nonmatched = 1;
  else nonmatched = 0;
  run;

proc print data=monitor (where=(nonmatched=1));  run;

proc sort data=a;  by unit transect;  run;
/**
proc print data=a;  by unit transect;  var unit transect u_name t_name;  run;
**/

proc sort data=monitor;  by unit transect year;  run;
```

```
** Merge vegetation monitoring data with climate/precip/ESD data, at transect level **;
data b;
  merge monitor (in=one) a (in=two);
  by unit transect;
  if one & ^two then nomatch =      "Monitor only";
  else if two & ^one then nomatch = "Abiotic only";
  else nomatch = "            ";
  run;

/**
proc print data=b (where=(nomatch="Monitor only"));
  title1 "Mixed grass transects lacking PRECIP or ESD data";
  var unit transect year;
  run;

proc print data=b (where=(nomatch="Abiotic only"));
  title1 "Transects found in PRECIP or ESD files, but not in monitoring data";
  var u_name t_name;
  run;

data units;
  set b;
  by unit;
  if first.unit;
  run;

proc print data=units;
  title1 "Unit-level data";
  ** var unit u_name psa08 psa09 psa10 psa11 psa12 psa13 psg09 psg10 psg11 psg12 psg13 psg14 gsp_ppr map_ppr mtwm_ppr
mmax_ppr;
  var unit u_name psa08 psa09 psa10 psa11 psa12 psa13 psg09 psg10 psg11 psg12 psg13 psg14 psg15 psg16 gsp map mtwm
maxt;
  run;
**/

** Select abiotic data corresponding only to records found in monitoring file **;
data a_sub;
  merge monitor (in=one keep=unit transect) a (in=two);
  by unit transect;
  if one & first.transect;
  run;

** Create vegetation monitoring file with abiotic variables **;
data b;
  merge monitor a_sub;
  by unit transect;
  run;

proc sort data=b out=unityear;  by u_name year;  run;

** From matched data, create a file of unit-level x year data (treatments and precip/climate) **;
data unityear (keep=u_name unit year trtmnt /*psa08-psa15 psg09-psg16 gsp map mtwm maxt*/);
  set unityear;
  by u_name year;
  if first.year;
  run;
```

```
proc sort data=b out=tranlevel;  by u_name t_name;  run;

** From matched data, create a file of transect-level data (ESD info) **;
data tranlevel (keep=u_name t_name plnv);
  set tranlevel;
  by u_name t_name;
  if first.t_name;
  run;

proc sort data=b out=all;  by u_name t_name year;  run;

** Fill out monitoring data with missing year placeholders **;
data pad;
  set all (keep=gtype u_name t_name unit transect psa08-psa15 psg09-psg16 gsp map mtwm maxt);
  by u_name t_name;
  if first.t_name then do year=2009 to 2016;
    output;
  end;
  run;

data all (keep=gtype u_name t_name transect year np sb kb rm psa08-psa15 psg09-psg16 gsp map mtwm maxt);
  merge all (drop=unit gtype psa08-psa15 psg09-psg16 gsp map mtwm maxt) pad;
  by u_name t_name year;
  run;

proc sort data=all;  by u_name year;  run;

** Merge padded monitoring data with unit-level data;
data all;
  merge all unityear;
  by u_name year;
  run;

/**
** Merge with long term climate data;
data all;
  merge all (in=one) climate_means (in=two);
  by u_name;
  if two and ^one then delete;
  run;
**/

proc sort data=all;  by u_name t_name;  run;

** Merge padded monitoring data with transect-level data
**  Assign treatment indicators by grass type, create a lagged variable for NP **;
data all (keep=gtype u_name t_name unit transect plnv year trt rain_prior rain_grow stops np npp lagnpp gsp map mtwm maxt);
  merge all tranlevel;
  by u_name t_name;
  array prsum[2,8] psa08 psa09 psa10 psa11 psa12 psa13 psa14 psa15
                   psg09 psg10 psg11 psg12 psg13 psg14 psg15 psg16;
  rain_prior = prsum[1,year-2008];
  rain_grow = prsum[2,year-2008];
  if gtype="M" then do;
    if trtmnt="Rest" then trt = 11;
    else if trtmnt="Graze" then trt = 12;
```

```
    else if trtmnt="Burn" then trt = 13;
    else if trtmnt="Burn/Graze" then trt = 14;
    else if trtmnt="Unclassifiable" | trtmnt="Hay" then trt = 15;
    else trt = .;
            end;
  else do;
    if trtmnt="Rest" then trt = 21;
            else if trtmnt="Graze w/in window" then trt = 22;
            else if trtmnt="Burn w/in window" then trt = 23;
            else if trtmnt="Defoliate" or (year=2009 & trtmnt in("Burn","Graze","Burn/Graze")) then trt = 24;
            else if trtmnt="Unclassifiable" then trt = 25;
            else trt = .;
    end;
  stops = np+sb+kb+rm;
  npp = np/stops;
  lagnpp = lag(npp);
  run;

** Count all years with non-missing data, per transect **;
data count (keep=u_name t_name unit transect count)
     countunit (keep=u_name unit trancount plus1count difference gsp map mtwm maxt);
  set all;
  by u_name t_name;
  retain count trancount plus1count;
  if first.u_name then do;
    trancount = 0;
    plus1count = 0;
  end;
  if first.t_name then count = 0;
  if stops>. then count+1;
  if last.t_name then do;
    output count;
    trancount+1;
    if count>1 then plus1count+1;
    difference = plus1count-trancount;
  end;
  if last.u_name then output countunit;
  run;

/**
proc print data=count;  var unit t_name count;  run;
**/
proc print data=countunit;  var u_name trancount plus1count difference gsp map mtwm maxt;  run;


** Exclude transects containing only 1 observation over 2009-2016 **
** For all transects, drop the year=2009 observation (use only its lagged NP data) **;
data all (drop=unit transect count);
  merge all count (keep=u_name t_name count);
  by u_name t_name;
  if count<=1 then delete;
  if first.t_name then delete;
  run;

proc export data=work.all outfile="&dir\abiotic.merged2016.csv"
     dbms=csv replace;
  putnames=yes;
```

```
  run;


/**
** Create indices **;
data all;
  set all (rename=(unit=unit_id transect=transect_id));
  by u_name t_name;
  retain unit unit_tran 0;
  if first.u_name then unit+1;
  if first.t_name then unit_tran+1;
  unit_tran_year = _n_;
  run;

proc sort data=all;  by u_name year;  run;

data all;
  set all;
  by u_name year;
  retain unit_year 0;
  if first.year then unit_year+1;
  run;

proc sort data=all;  by u_name t_name;  run;


proc print data=all;
  var unit unit_tran unit_year unit_id gsp mtwm transect_id u_name pInv year trt rain_prior rain_grow stops np npp lagnpp;
  format u_name $20.;
  run;


data all1;
  set all;
  by u_name t_name;
  array y[5] m1-m5;
  array r[5] r1-r5;
  array t[5] t1-t5;
  retain m1-m5 r1-r5 t1-t5;
  if first.t_name then do i=1 to 5;
    y[i] = .;
    r[i] = .;
    t[i] = .;
  end;
  do i=1 to 5;
    if i=year-2009 then do;
      y[i] = stops;
      r[i] = rain_prior;
      t[i] = trt;
    end;
  end;
  if last.t_name then output;
  run;

proc print data=all1;
  by u_name;
  var unit unit_id unit_tran transect_id pInv m1-m5 r1-r5 t1-t5;
```

```
  format r1-r5 5.2;
  run;

proc format;
  value missfmt .="  NA  ";
  run;



data _null_;
  set all end=lastone;
  by u_name t_name;
  retain n_units n_trans n_obs;
  if _n_=1 then do;
    file outdata("unit_level.txt");
    put @1 "unit_idx[] gsp[] map[] mtwm[] maxt[]";
    file outdata("tran_level.txt");
    put @1 "tran_idx[] pinv[]";
    file outdata("np.txt");
    put @1 "obs.unit[] obs.tran[] obs.year[] obs.n[] year[] stops[] np[] lagnpp[]";
    file outdata("np_inits.txt");
    put @1 "np[] lagnpp[]";
    file outdata("pinv_inits.txt");
    put @1 "pinv[]";
    n_units = 0;
    n_trans = 0;
    n_obs = 0;
  end;
  if first.u_name then do;
    n_units+1;
    file outdata("unit_level.txt");
    put @1 unit 3. @5 (gsp map mtwm maxt) (7.3 +1);
  end;
  if first.t_name then do;
    n_trans+1;
    file outdata("tran_level.txt");
    put @1 unit_tran 4. @6 pInv missfmt.;
    file outdata("pinv_inits.txt");
    if pInv=. then put @1 "0.5";
    else put @1 "NA";
  end;
  n_obs+1;
  file outdata("np.txt");
  if stops=. then stops = 50;
  put @1 unit 3. @5 unit_tran 4. @10 unit_year 3. @14 unit_tran_year 5. @20 year 4. @25 stops 2. @28 np missfmt. @35 lagnpp
missfmt.;
  file outdata("np_inits.txt");
  if np=. & lagnpp>. then put @1 "25" @4 "NA";
  if np>. & lagnpp=. then put @1 "NA" @4 "0.5";
  if np=. & lagnpp=. then put @1 "25" @4 "0.5";
  if np>. & lagnpp>. then put @1 "NA" @4 "NA";
  if lastone then do;
    file outdata("unit_level.txt");
    put @1 "END";
    file outdata("tran_level.txt");
    put @1 "END";
    file outdata("np.txt");
```

```
    put @1 "END";
    file outdata("np_inits.txt");
    put @1 "END";
    file outdata("pinv_inits.txt");
    put @1 "END";
    file outdata("ssize.txt");
    put @1 "list(n_units=" n_units ", n_trans=" n_trans ", n_obs=" n_obs ",";
  end;
  run;

data _null_;
  set all end=lastone;
  retain first countmiss;
  file outdata("ssize.txt") mod ls=120;
  if _n_=1 then do;
    put "miss_np=c(" @;
    first = 1;
    countmiss = 0;
  end;
  if np=. then do;
    countmiss+1;
    if first then put unit_tran_year @;
    else put "," unit_tran_year @;
    first = 0;
  end;
  if lastone then put ")," / "n_miss_np=" countmiss ",";
  run;

data _null_;
  set all end=lastone;
  retain first countmiss;
  file outdata("ssize.txt") mod ls=120;
  if _n_=1 then do;
    put "miss_lag=c(" @;
    first = 1;
    countmiss = 0;
  end;
  if lagnpp=. then do;
    countmiss+1;
    if first then put unit_tran_year @;
    else put "," unit_tran_year @;
    first = 0;
  end;
  if lastone then put ")," / "n_miss_lag=" countmiss ",";
  run;

data _null_;
  set all end=lastone;
  by u_name t_name;
  retain first countmiss;
  file outdata("ssize.txt") mod ls=120;
  if _n_=1 then do;
    put "miss_pinv=c(" @;
    first = 1;
    countmiss = 0;
  end;
  if first.t_name & pInv=. then do;
```

```
    countmiss+1;
    if first then put unit_tran @;
    else put "," unit_tran @;
    first = 0;
  end;
  if lastone then put ")," / "n_miss_pinv=" countmiss ",";
  run;

proc sort data=all;  by u_name year t_name;  run;

data _null_;
  set all end=lastone;
  by u_name year;
  retain n_year;
  file outdata("year_level.txt");
  if _n_=1 then do;
    file outdata("year_level.txt");
    put @1 "year_idx[] trt[] prcp_prev[] prcp_grow[]";
    n_year = 0;
    file outdata("trt_inits.txt");
    put @1 "trt[] prcp_prev[] prcp_grow[]";
  end;
  if first.year then do;
    file outdata("year_level.txt");
    put @1 unit_year 3. @5 trt missfmt. @12 rain_prior missfmt. @28 rain_grow missfmt.;
    n_year+1;
    file outdata("trt_inits.txt");
    if trt=. then put @1 "1  23 23";
    else put @1 "NA NA NA";
  end;
  if lastone then do;
    file outdata("year_level.txt");
    put @1 "END";
    file outdata("trt_inits.txt");
    put @1 "END";
    file outdata("ssize.txt") mod;
    put "n_year=" n_year ",";
  end;
  ** Mean, SD of precip approx 23 and 5 **;
  run;

data _null_;
  set all end=lastone;
  by u_name year;
  retain first countmiss;
  file outdata("ssize.txt") mod ls=120;
  if _n_=1 then do;
    put "miss_trt=c(" @;
    first = 1;
    countmiss = 0;
  end;
  if first.year & trt=. then do;
    countmiss+1;
    if first then put unit_year @;
    else put "," unit_year @;
    first = 0;
  end;
```

```
  if lastone then put ")," / "n_miss_trt=" countmiss ") )";
  run;

***/
quit;
```

```
## Analysis of ESD + climate/precip data for mixed grass 2010-2016

#setwd('C:/projects/RCRP/NP/abiotic')
#setwd('c:/users/clint/dropbox/NPAM/abiotic')
#setwd('c:/documents and settings/clint/my documents/my dropbox/NPAM/abiotic')
setwd('C:/Users/ctmoore/Dropbox/NPAM/abiotic')
library(rjags)

abiotic_all <- read.csv("abiotic.merged2016.csv")
abiotic <- abiotic_all[which(abiotic_all$gtype=='M'),]
n.obs <- length(abiotic$NP)
year_factor <- factor(abiotic$year)

# Index for unique combinations of year and unit
#unit.year <- as.integer(interaction(year_factor,abiotic$u_name))
unit.year <- as.integer(factor(interaction(year_factor,abiotic$u_name)))
n.unit.year <- length(unique(unit.year))

# Index for unit
unit <- as.integer(factor(abiotic$u_name))
n.unit <- length(unique(unit))

# Index for transect
unit.tran <- as.integer(factor(abiotic$t_name))
n.unit.tran <- length(unique(unit.tran))

# Create vector of ESD data (pinv) -- one record per transect
pinv <- abiotic$pInv[!duplicated(cbind(abiotic$pInv, unit.tran))]

# Identify those records with missing NP value
miss.np <- which(is.na(abiotic$NP))
n.miss.np <- length(miss.np)

# Identify those records with missing value of lagged NP proportion
miss.lag <- which(is.na(abiotic$lagnpp))
n.miss.lag <- length(miss.lag)

# Identify those transects with missing ESD (pinv) data
#miss.pinv <- unique(unit.tran[which(is.na(abiotic$pInv))])
#miss.pinv <- unique(unit.tran[which(is.na(pinv))])
#miss.pinv <- unit.tran[which(is.na(pinv))]
miss.pinv <- which(is.na(pinv))
n.miss.pinv <- length(miss.pinv)

# Create matrix of climate variables -- one record per unit
# GSP (1) = growing season precip, MAP (2) = mean annual precip,
# MTWM (3) = mean temp in warmest month, MAXT (4) = mean max temp in warmest month
climate <- cbind(abiotic$GSP,abiotic$MAP,abiotic$MTWM,abiotic$MAXT)
climate <- climate[!duplicated(cbind(climate, unit)),]

# Create matrix of treatment and rainfall data -- one record per unit x year combination
# Note: Mixed grass treatments 11-15, Tallgrass treatments 21-25
annual <- cbind(abiotic$trt,abiotic$rain_prior,abiotic$rain_grow)
annual <- annual[!duplicated(cbind(annual,unit.year)),]

# Identify those unit x year combinations with missing treatment data
#miss.trt <- unique(unit.year[which(is.na(abiotic$trt))])
```

```
miss.trt <- which(is.na(annual[,1]))
n.miss.trt <- length(miss.trt)

# Provide initial values for unit-years with missing treatment or rainfall values
annual.inits <- matrix(rep(NA,n.unit.year*3),n.unit.year,3)
annual.inits[is.na(annual[,1]),1] <- 1
annual.inits[is.na(annual[,2]),2] <- 0
annual.inits[is.na(annual[,3]),3] <- 0

prcp.raw <- annual[,2]
np <- abiotic$NP
lagnpp <- abiotic$lagnpp
## Convert mixed grass treatment to integers 1-5
trt <- annual[,1]-10
stops <- abiotic$stops
stops[is.na(stops)] <- 50


# Bundle data
jags.data <- list(trt = trt,
            prcp.mean = mean(prcp.raw,na.rm=TRUE),
            prcp.sd = sd(prcp.raw,na.rm=TRUE),
            prcp.raw = prcp.raw,
            pinv = pinv,
            pinv.mean = mean(pinv,na.rm=TRUE),
            obs.unit = unit,
            obs.tran = unit.tran,
            obs.year = unit.year,
            lagnpp = lagnpp,
            lagnpp.mean = mean(lagnpp,na.rm=TRUE),
            year = abiotic$year,
            np = np,
            stops = stops,
            n.obs = n.obs,
            n.unit = n.unit,
            n.unit.tran = n.unit.tran,
            n.unit.year = n.unit.year,
            n.miss.lag = n.miss.lag,
            n.miss.pinv = n.miss.pinv,
            n.miss.trt = n.miss.trt,
            gsp.norm = (climate[,1]-mean(climate[,1]))/sd(climate[,1]),
            map.norm = (climate[,2]-mean(climate[,2]))/sd(climate[,2]),
            mtwm.norm = (climate[,3]-mean(climate[,3]))/sd(climate[,3]),
            maxt.norm = (climate[,4]-mean(climate[,4]))/sd(climate[,4]),
            miss.lag = miss.lag,
            miss.pinv = miss.pinv,
            miss.trt = miss.trt)



# Initial values
inits <- function(){list(a = rnorm(1,0,0.1),
               b = rnorm(1,0,0.1),
               c_parm = rnorm(4,0,0.1),
               ac_parm = rnorm(4,0,0.1),
               bc_parm = rnorm(4,0,0.1),
               d = rnorm(1,0,0.1),
```

```
                    sd.obs = runif(1,0,10),
                    u.precip = rnorm(1,0,0.1),
                    u.temp = rnorm(1,0,0.1),
                    u.int = rnorm(1,0,0.1),
                    mu = rnorm(1,0,0.1),
                    miss.lag.val = rep(0.5,n.miss.lag),
                    miss.pinv.val = rep(0.5,n.miss.pinv),
                    miss.trt.val = rep(1,n.miss.trt),
                    miss.prcp.val = rep(23,n.miss.trt))}

# Parameters monitored
parameters <- c("a", "b", "c", "ac", "bc", "d", "mu", "u.precip", "u.temp", "u.int", "e",
            "sd.obs",
            "sd.e",
            "T.int", "T.prcp", "T.inv", "NP.prcp", "NP.inv",
            "Y", "Y.p", "Y.trend", "yeartrend",
            "Bayes.pval", "fit", "fit.new"
            )

# MCMC settings
ni <- 100000
nt <- 1
nb <- 500
nc <- 3
na <- 100
iter.inc <- 100
R.hat.lim <- 1.1


## jm <- jags.model(data = jags.data, inits = inits, file = "abiotic2016esd.jags",
##              n.chains = nc, n.adapt = nb, quiet = F)

## jc <- coda.samples(model=jm, variable.names=parameters, n.iter=ni, thin=nt)

## Output from rjags
## jcout <- summary(out)
## stats <- as.matrix(jcout$statistics)
## quants <- as.matrix(jcout$quantiles)
## Year.eff <- cbind(stats[(substr(rownames(stats),1,3)=="Y.p"),],quants[(substr(rownames(quants),1,3)=="Y.p"),])
## Year.trend <- cbind(stats[(substr(rownames(stats),1,3)=="Y.t"),],quants[(substr(rownames(quants),1,3)=="Y.t"),])
## NP.inv.est <- cbind(stats[(substr(rownames(stats),1,4)=="NP.i"),],quants[(substr(rownames(quants),1,4)=="NP.i"),])
## NP.prcp.est <- cbind(stats[(substr(rownames(stats),1,4)=="NP.p"),],quants[(substr(rownames(quants),1,4)=="NP.p"),])
## clim.est <- rbind( c(stats[(rownames(stats)=="mu"),],quants[(rownames(quants)=="mu"),]),
##               c(stats[(rownames(stats)=="u.precip"),],quants[(rownames(quants)=="u.precip"),]),
##               c(stats[(rownames(stats)=="u.temp"),],quants[(rownames(quants)=="u.temp"),]),
##               c(stats[(rownames(stats)=="u.int"),],quants[(rownames(quants)=="u.int"),]) )
## col.med <- 7
## col.q2.5 <- 5
## col.q97.5 <- 9
## quants.range <- (5:9)

library(jagsUI)
options(max.print=2000)
start <- Sys.time()
paste("Started",start)
#out <- jags(data = jags.data, inits=inits, parameters.to.save=parameters,
#          model.file="abiotic2016esd.jags",
```

```
#       n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, n.adapt = na,
#       parallel=TRUE)
#out <- autojags(data = jags.data, inits=inits, parameters.to.save=parameters,
#       model.file="abiotic2016esd.jags", n.chains = nc, n.thin = nt, n.burnin = nb,
#       iter.increment = iter.inc, save.all.iter=TRUE, Rhat.limit=R.hat.lim, max.iter=ni,
#       codaOnly=c("T.int","T.prcp","T.inv","NP.prcp","NP.inv",
#               "Y","Y.p","Y.trend","yeartrend",
#               "Bayes.pval", "fit", "fit.new"),
#       parallel=TRUE)
out1 <- update(out,n.iter=10000)
end <- Sys.time()
paste("Started",start,"; Ended",end)
end - start

## Output from jagsUI
out <- out1

Year.eff <- out$summary[(substr(rownames(out$summary),1,3)=="Y.p"),]
Year.trend <- out$summary[(substr(rownames(out$summary),1,4)=="Y.tr"),]
NP.inv.est <- out$summary[(substr(rownames(out$summary),1,4)=="NP.i"),]
NP.prcp.est <- out$summary[(substr(rownames(out$summary),1,4)=="NP.p"),]
clim.est <- rbind( out$summary[(rownames(out$summary)=="mu"),],
                out$summary[(rownames(out$summary)=="u.precip"),],
                out$summary[(rownames(out$summary)=="u.temp"),],
                out$summary[(rownames(out$summary)=="u.int"),])
col.med <- 5
col.q2.5 <- 3
col.q97.5 <- 7
quants.range <- (3:7)

## Trend plot of NP response
Year.level <- factor(2010:2016)
par(new=FALSE)
plot((2010:2016),Year.trend[,col.med], xlim=c(2009.5, 2016.5), ylim=c(0,0.6), type='n', axes=F, ylab="", xlab="")
polygon(c(2010:2016,2016:2010),c(Year.trend[,col.q97.5],rev(Year.trend[,col.q2.5])),col="grey90",border=NA)
lines((2010:2016),Year.trend[,col.med], lwd = 2)
par(new=TRUE)
boxplot(Year.eff[,col.q2.5:col.q97.5]~Year.level,ylim=c(0,0.6),range=0,pars=list(boxwex=0.1,staplewex=0,whisklty=1),
    ylab="Proportion desired native", xlab="Year",cex.axis=1.1,cex.lab=1.5)


# Exclude non-assignable treatments
NP.inv.est <- NP.inv.est[-c(5,10,15),]
NP.prcp.est <- NP.prcp.est[-c(5,10,15),]

inv.level <- factor(c(0,0.5,1))
prcp.level <- factor(levels=c(-2,0,2),labels=c('Dry','Average','Wet'))
#trt.level <- factor(levels=(1:5),labels=c('Rest','Graze','Fire','Combination','Other'))
trt.level <- factor(levels=(1:4),labels=c('Rest','Graze','Burn','Burn/Graze'))

## Treatment x Invasion Vulnerability interaction
#d <- as.data.frame(cbind(rep(c(1:3),each=5),rep(c(1:5),3),NP.inv.est[,quants.range]))
d <- as.data.frame(cbind(rep(c(1:3),each=4),rep(c(1:4),3),NP.inv.est[,quants.range]))
names(d)[1]<-paste("Inv")
names(d)[2]<-paste("Trt")
d <- reshape(d,varying=list(names(d)[quants.range]),direction="long",timevar="Pctile",times=c("2.5%","25%","50%","75%","97.5"),
      v.names="Proportion.NP",idvar=c("Inv","Trt"))
```

```
d$Inv <- factor(d$Inv,labels=c('0','0.5','1'))
#d$Trt <- factor(d$Trt,labels=c('Rest','Graze','Fire','Combination','Other'))
d$Trt <- factor(d$Trt,labels=c('Rest','Graze','Burn','Burn/Graze'))
library(ggplot2)
#ggplot(data = d, aes(x = Inv, y = Proportion.NP)) + geom_boxplot(aes(fill=Trt), width=0.6) + theme_bw() +
#       labs(x="Invasion Vulnerability",y="Proportion of desired native",fill="Treatment")
d2 <- d[order(d$Trt,d$Inv),]
ggplot(data = d2, aes(x = Trt, y = Proportion.NP)) + geom_boxplot(aes(fill=Inv), width=0.6) + theme_bw() +
    coord_cartesian(ylim = c(0,0.8)) + scale_y_continuous(breaks=pretty(d2$Proportion.NP, n=4)) +
    labs(x="Treatment",y="Proportion desired native",fill="Invasion \nVulnerability") +
    theme(axis.text=element_text(size=18),axis.title=element_text(size=20),
        axis.title.x=element_text(vjust=-0.5),axis.title.y=element_text(vjust=2),
        legend.position=c(0.145,0.8),legend.text=element_text(size=16),legend.title=element_text(size=16),
        legend.background=element_rect(fill="grey90"))

## Treatment x Precipitation interaction
#d <- as.data.frame(cbind(rep(c(1:3),each=5),rep(c(1:5),3),NP.prcp.est[,quants.range]))
d <- as.data.frame(cbind(rep(c(1:3),each=4),rep(c(1:4),3),NP.prcp.est[,quants.range]))
names(d)[1]<-paste("Prcp")
names(d)[2]<-paste("Trt")
d <- reshape(d,varying=list(names(d)[quants.range]),direction="long",timevar="Pctile",times=c("2.5%","25%","50%","75%","97.5"),
      v.names="Proportion.NP",idvar=c("Prcp","Trt"))
d$Prcp <- factor(d$Prcp,labels=c('Dry','Average','Wet'))
#d$Trt <- factor(d$Trt,labels=c('Rest','Graze','Fire','Combination','Other'))
d$Trt <- factor(d$Trt,labels=c('Rest','Graze','Burn','Burn/Graze'))
library(ggplot2)
#ggplot(data = d, aes(x = Prcp, y = Proportion.NP)) + geom_boxplot(aes(fill=Trt), width=0.6) + theme_bw() +
#       labs(x="Prior-year Precipitation",y="Proportion of desired native",fill="Treatment")
d2 <- d[order(d$Trt,d$Prcp),]
ggplot(data = d2, aes(x = Trt, y = Proportion.NP)) + geom_boxplot(aes(fill=Prcp), width=0.6) + theme_bw() +
    coord_cartesian(ylim = c(0,0.8)) + scale_y_continuous(breaks=pretty(d2$Proportion.NP, n=4)) +
    labs(x="Treatment",y="Proportion desired native",fill="Prior-year \nPrecipitation") +
    theme(axis.text=element_text(size=18),axis.title=element_text(size=20),
        axis.title.x=element_text(vjust=-0.5),axis.title.y=element_text(vjust=2),
        legend.position=c(0.145,0.8),legend.text=element_text(size=16,hjust="c"),legend.title=element_text(size=16),
        legend.background=element_rect(fill="grey90"))



## Plot response surface for climate-driven unit effect
library(lattice)
library(RColorBrewer)
library(colorspace)
resp.color <- brewer.pal(11,"BrBG")
resp.color <- rev(diverge_hcl(12,h=c(130,43),c=100,l=c(70,90)))
map.norm <- (climate[,2]-mean(climate[,2]))/sd(climate[,2])
mtwm.norm <- (climate[,3]-mean(climate[,3]))/sd(climate[,3])

clim.resp <- function(x, y){1 / (1 + exp(-(clim.est[1,1] + clim.est[2,1]*x + clim.est[3,1]*y + clim.est[4,1]*x*y)))}
prcp <- seq(300,700,length=200)
temp <- seq(17,25,length=200)
np.resp <- outer((prcp-mean(climate[,2]))/sd(climate[,2]),
          (temp-mean(climate[,3]/10))/sd(climate[,3]/10),
           clim.resp)
#persp(prcp,temp,np.resp)
#contour(prcp,temp,np.resp)
#grid <- expand.grid(prcp=prcp,temp=temp)
```

```
#grid$z <- as.vector(np.resp)
#levelplot(np.resp ~ prcp*temp,grid,xlim=c(300,650),ylim=rev(c(17,25)))
contourplot(np.resp, row.values=seq(300,700,length=200), column.values=seq(17,25,length=200),
        aspect=1, colorkey=TRUE, region=TRUE, xlim=c(300,700), ylim=rev(c(17,25)),
        xlab=list(label="Mean Annual Precipitation (mm)",cex=1.5),
        ylab=list(label="Mean Temperature of Warmest Month (C)",cex=1.5),
        scales=list(cex=1.5),labels=list(cex=1.3),label.style="align",col.regions=resp.color)
#par(new=TRUE,pin=c(6,6))
#plot(climate[,2],climate[,3]/10, xlim=c(300,650), ylim=rev(c(17,25)), type='p', pch=1, axes=T,
#     col="black",ylab="", xlab="")
#surface3d(prcp,temp,np.resp)
```

```
model
 {
   ## Model for ESD and rainfall, mixed grass units only

   ## Likelihood
   for (i in 1:n.obs)
     {

       logitp[i] <- mu + re.unit[obs.unit[i]] +
                     a * pinv.norm[obs.tran[i]] +

             ac[trt[obs.year[i]]] * pinv.norm[obs.tran[i]] +
                     b * prcp.norm[obs.year[i]] +
                     c[trt[obs.year[i]]] +
                     bc[trt[obs.year[i]]] * prcp.norm[obs.year[i]] +
                     d * lagnp.norm[i] +
                     e[year[i]-2009] +
                     z.obs[i]
       ## Standardize lagnpp to mean
       lagnp.norm[i] <- lagnpp[i] - lagnpp.mean
       ## lagnp.norm[i] <- lagnpp[i] - 0.5
       z.obs[i] ~ dnorm(0,tau.obs)
       p[i] <- 1 / (1+exp(-logitp[i]))
       np[i] ~ dbin(p[i],stops[i])
       mean[i] <- p[i]*stops[i]
       denom[i] <- sqrt(mean[i]*(1-p[i])*p[i])

       Pearson.resi[i] <- (np[i] - mean[i]) / denom[i]
       D[i] <- pow(Pearson.resi[i],2)
       np.new[i] ~ dbin(p[i],stops[i])
       Pearson.resi.new[i] <- (np.new[i] - mean[i]) / denom[i]
       D.new[i] <- pow(Pearson.resi.new[i],2)

     }

   ## Bayesian P-value
   fit <- sum(D[])
   fit.new <- sum(D.new[])
   Bayes.pval <- step(fit.new - fit)

   ## Priors for fixed year effects
   for (i in 1:6)
     {

#       e[i] ~ dnorm(0, 0.37)
       e[i] ~ dnorm(0, tau.e)

     }
   e[7] <- -(e[1] + e[2] + e[3] + e[4] + e[5] + e[6])
   tau.e <- pow(sd.e,-2)
   sd.e ~ dunif(0,10)

   ## Unit-level climate effects model
   re.unit <- u.precip*map.norm + u.temp*mtwm.norm + u.int*map.norm*mtwm.norm

   ## Center pinv predictor
   pinv.norm <- pinv - pinv.mean
```

```
## Normalize precipitation for use as predictor
prcp.norm <- (prcp.raw-prcp.mean) / prcp.sd

## Estimation of missing predictor values
  for (i in 1:n.miss.lag)
    {

      #                                       lagnpp[miss.lag[i]] ~ dunif(0.01, 0.99)
      miss.lag.val[i] ~ dbeta(1, 1)
      lagnpp[miss.lag[i]] <- miss.lag.val[i]

    }

  for (i in 1:n.miss.pinv)
    {

      #                                       pinv[miss.pinv[i]] ~ dunif(0, 1)
      miss.pinv.val[i] ~ dbeta(1, 1)
      pinv[miss.pinv[i]] <- miss.pinv.val[i]

    }

  for (i in 1:n.miss.trt)
    {

      miss.trt.val[i] ~ dcat(trtprob[])
      trt[miss.trt[i]] <- miss.trt.val[i]
      ## No missing annual rainfall
      ## miss.prcp.val[i] ~ dnorm(prcp.mean, prcp.prec)
      ## prcp.raw[miss.trt[i]] <- miss.prcp.val[i]

    }

## Priors for treatment effects and interactions with ESD, precip
for (i in 1:4)
  {

    trtprob[i] <- 0.2
    c_parm[i] ~ dnorm(0, 0.37)
    c[i] <- c_parm[i]
    ac_parm[i] ~ dnorm(0, 0.37)
    ac[i] <- ac_parm[i]
    bc_parm[i] ~ dnorm(0, 0.37)
    bc[i] <- bc_parm[i]

  }

## Zero-sum constraints
trtprob[5] <- 0.2
c[5] <- -(c[1]+c[2]+c[3]+c[4])
ac[5] <- -(ac[1]+ac[2]+ac[3]+ac[4])
bc[5] <- -(bc[1]+bc[2]+bc[3]+bc[4])

## Priors for unit-level climate effects
u.precip ~ dnorm(0, 0.37)
u.temp ~ dnorm(0, 0.37)
```

```
u.int ~ dnorm(0, 0.37)

mu ~ dnorm(0, 0.37)

## Priors for main regression terms
a ~ dnorm(0, 0.37)
b ~ dnorm(0, 0.37)
d ~ dnorm(0, 0.37)

## Priors for observation-level random effects
tau.obs <- 1/pow(sd.obs,2)
sd.obs ~ dunif(0,10)

## Derived quantities for effects
for (i in 1:5){
  T.int[i] <- mu + c[i]
  T.prcp[i] <- b + bc[i]
  T.inv[i] <- a + ac[i]
  for (j in 1:3){
    NP.prcp[i,j] <- 1/(1 + exp(-(T.int[i] + T.prcp[i]*(2*j-4))))
    NP.inv[i,j] <- 1/(1 + exp(-(T.int[i] + T.inv[i]*(0.5*j-1))))
  }
}

## Derived quantities for trend (2010-2016)
for (i in 1:7){
  Y[i] <- mu + e[i]
  Y.p[i] <- 1/(1+exp(-Y[i]))
  Y.trend[i] <- 1/(1+exp(-(mu+yeartrend*(i-4))))
}
yeartrend <- (-3*Y[1] - 2*Y[2] - 1*Y[3] + 0*Y[4] + 1*Y[5] + 2*Y[6] + 3*Y[7])/28

}
```

```
## Analysis of climate/precip data for mixed + tallgrass 2010-2016
## This version codes treatment as 1-5 in both grass types, and includes grass type * treatment interaction

#setwd('C:/projects/RCRP/NP/abiotic')
#setwd('c:/users/clint/dropbox/NPAM/abiotic')
#setwd('c:/documents and settings/clint/my documents/my dropbox/NPAM/abiotic')
setwd('C:/Users/ctmoore/Dropbox/NPAM/abiotic')

abiotic <- read.csv("abiotic.merged2016.csv")
n.obs <- length(abiotic$NP)
year_factor <- factor(abiotic$year)

# Index for unique combinations of year and unit
unit.year <- as.integer(interaction(year_factor,abiotic$u_name))
n.unit.year <- length(unique(unit.year))

# Index for grass type
g <- as.integer(abiotic$gtype)
n.g <- 2
g.unit.year <- g[!duplicated(cbind(g, unit.year))]

# Index for unit
unit <- as.integer(abiotic$u_name)
n.unit <- length(unique(unit))

# Index for transect
unit.tran <- as.integer(abiotic$t_name)
n.unit.tran <- length(unique(unit.tran))

# Create vector of ESD data (pinv) -- one record per transect
#pinv <- abiotic$pInv[!duplicated(cbind(abiotic$pInv, unit.tran))]

# Identify those records with missing NP value
miss.np <- which(is.na(abiotic$NP))
n.miss.np <- length(miss.np)

# Identify those records with missing value of lagged NP proportion
miss.lag <- which(is.na(abiotic$lagnpp))
n.miss.lag <- length(miss.lag)

# Identify those transects with missing ESD (pinv) data
#miss.pinv <- unique(unit.tran[which(is.na(abiotic$pInv))])
#miss.pinv <- unique(unit.tran[which(is.na(pinv))])
#miss.pinv <- unit.tran[which(is.na(pinv))]
#miss.pinv <- which(is.na(pinv))
#n.miss.pinv <- length(miss.pinv)

# Create matrix of climate variables -- one record per unit
# GSP (1) = growing season precip, MAP (2) = mean annual precip,
# MTWM (3) = mean temp in warmest month, MAXT (4) = mean max temp in warmest month
climate <- cbind(abiotic$GSP,abiotic$MAP,abiotic$MTWM,abiotic$MAXT)
climate <- climate[!duplicated(cbind(climate, unit)),]

# Create matrix of treatment and rainfall data -- one record per unit x year combination
# Note: Mixed grass treatments 11-15, Tallgrass treatments 21-25
annual <- cbind(abiotic$trt,abiotic$rain_prior,abiotic$rain_grow)
annual <- annual[!duplicated(cbind(annual,unit.year)),]
```

```
# Identify those unit x year combinations with missing treatment data
#miss.trt <- unique(unit.year[which(is.na(abiotic$trt))])
find.miss.trt <- is.na(annual[,1])
miss.trt <- which(find.miss.trt)
n.miss.trt <- length(miss.trt)
# miss.trt.g <- (5 * (g.unit.year==2))[find.miss.trt]

# Provide initial values for unit-years with missing treatment or rainfall values
annual.inits <- matrix(rep(NA,n.unit.year*3),n.unit.year,3)
inds <- which(unit.year %in% miss.trt)
# annual.inits[find.miss.trt,1] <- miss.trt.g + 1
annual.inits[find.miss.trt,1] <- 1
annual.inits[is.na(annual[,2]),2] <- 0
annual.inits[is.na(annual[,3]),3] <- 0

prcp.raw <- annual[,2]
# Identify those unit x year combinations with missing rainfall data
miss.prcp <- which(is.na(prcp.raw))
n.miss.prcp <- length(miss.prcp)

np <- abiotic$NP
lagnpp <- abiotic$lagnpp
# convert treatments to 1-5 coding in each grass type
trt <- annual[,1]-10-10*(annual[,1]>20)
stops <- abiotic$stops
stops[is.na(stops)] <- 50


# Bundle data
jags.data <- list(trt = trt,
          prcp.mean = mean(prcp.raw,na.rm=TRUE),
          prcp.sd = sd(prcp.raw,na.rm=TRUE),
          prcp.raw = prcp.raw,
          obs.unit = unit,
          obs.year = unit.year,
          g = g.unit.year,
          lagnpp = lagnpp,
          lagnpp.mean = mean(lagnpp,na.rm=TRUE),
          year = abiotic$year,
          np = np,
          stops = stops,
          n.obs = n.obs,
          n.unit = n.unit,
          n.unit.year = n.unit.year,
          n.miss.lag = n.miss.lag,
          n.miss.trt = n.miss.trt,
          n.miss.prcp = n.miss.prcp,
          gsp.norm = (climate[,1]-mean(climate[,1]))/sd(climate[,1]),
          map.norm = (climate[,2]-mean(climate[,2]))/sd(climate[,2]),
          mtwm.norm = (climate[,3]-mean(climate[,3]))/sd(climate[,3]),
          maxt.norm = (climate[,4]-mean(climate[,4]))/sd(climate[,4]),
          miss.lag = miss.lag,
          miss.trt = miss.trt,
          miss.prcp = miss.prcp)
```

```
# Initial values
inits <- function(){list(b = rnorm(1,0,0.1),
                         gamma_parm = rnorm(1,0,0.1),
                         c_parm = rnorm(4,0,0.1),
                         bc_parm = rnorm(4,0,0.1),
                         bg_parm = rnorm(1,0,0.1),
                         cg_parm = rnorm(4,0,0.1),
                         bcg_parm = rnorm(4,0,0.1),
                         d = rnorm(1,0,0.1),
#                          e_parm = rnorm(6,0,0.1),
#                          eg_parm = rnorm(6,0,0.1),
#                          trend = rnorm(1,0,0.1),
#                          g.trend_parm = rnorm(1,0,0.1),
#                          sd.e = runif(1,0,10),
                         sd.eg1 = runif(1,0,10),
                         sd.eg2 = runif(1,0,10),
                         sd.obs = runif(1,0,10),
                         u.precip = rnorm(1,0,0.1),
                         u.temp = rnorm(1,0,0.1),
                         u.int = rnorm(1,0,0.1),
                         mu = rnorm(1,0,0.1),
                         miss.lag.val = rep(0.5,n.miss.lag),
                         miss.trt.val = rep(1,n.miss.trt),
                         miss.prcp.val = rep(23,n.miss.prcp))}

# Parameters monitored
parameters <- c("b", "c",
                "bc", "cg","bcg",
#                 "e",
                "eg",
                "gamma", "bg", "d", "mu", "sd.obs",
#                 "sd.e",
                "sd.eg1",
                "sd.eg2",
                "u.precip", "u.temp", "u.int",
#                 "T.int", "T.prcp",
                "NP.prcp",
                "Y", "Y.p", "Y.trend",
                "Y.m", "Y.m.p", "Y.m.trend",
                "Y.t", "Y.t.p", "Y.t.trend",
#                 "trend","g.trend",
                "yeartrend","yeartrend.m","yeartrend.t",
                "Bayes.pval", "fit", "fit.new"
                )

#parameters <- c("b", "d", "mu", "u.precip", "u.temp", "u.int",
#                 "gamma", "bg", "sd.obs", "Bayes.pval", "fit", "fit.new")

# MCMC settings
ni <- 100000
nt <- 1
nb <- 500
nc <- 3
iter.inc <- 100
R.hat.lim <- 1.1
```

```
##library(rjags)
## jm <- jags.model(data = jags.data, inits = inits, file = "abiotic2016prcp.jags",
##              n.chains = nc, n.adapt = nb, quiet = F)

## out <- coda.samples(model=jm, variable.names=parameters, n.iter=ni, thin=nt)

## Output from rjags
## jcout <- summary(out)
## stats <- as.matrix(jcout$statistics)
## quants <- as.matrix(jcout$quantiles)
## Year.eff <- cbind(stats[(substr(rownames(stats),1,3)=="Y.p"),],quants[(substr(rownames(quants),1,3)=="Y.p"),])
## Year.trend <- cbind(stats[(substr(rownames(stats),1,3)=="Y.t"),],quants[(substr(rownames(quants),1,3)=="Y.t"),])
## NP.prcp.est <- cbind(stats[(substr(rownames(stats),1,4)=="NP.p"),],quants[(substr(rownames(quants),1,4)=="NP.p"),])
## clim.est <- rbind( c(stats[(rownames(stats)=="mu"),],quants[(rownames(quants)=="mu"),]),
##                c(stats[(rownames(stats)=="u.precip"),],quants[(rownames(quants)=="u.precip"),]),
##                c(stats[(rownames(stats)=="u.temp"),],quants[(rownames(quants)=="u.temp"),]),
##                c(stats[(rownames(stats)=="u.int"),],quants[(rownames(quants)=="u.int"),]) )
## col.med <- 7
## col.q2.5 <- 5
## col.q97.5 <- 9
## quants.range <- (5:9)

library(jagsUI)
options(max.print=2000)
start <- Sys.time()
paste("Started",start)
#out <- jags(data = jags.data, inits=inits, parameters.to.save=parameters,
#        model.file="abiotic2016prcp.jags", n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb,
#        parallel=TRUE)
#out <- autojags(data = jags.data, inits=inits, parameters.to.save=parameters,
#        model.file="abiotic2016prcp.jags", n.chains = nc, n.thin = nt, n.burnin = nb,
#        iter.increment = iter.inc, save.all.iter=TRUE, Rhat.limit=R.hat.lim, max.iter=ni,
#        codaOnly=c("NP.prcp","Y","Y.p","Y.trend","Y.m","Y.m.p","Y.m.trend",
#                "Y.t","Y.t.p","Y.t.trend","yeartrend","yeartrend.m","yeartrend.t",
#                "Bayes.pval", "fit", "fit.new"),
#        parallel=TRUE)
out1 <- update(out,n.iter=10000)
end <- Sys.time()
paste("Started",start,"; Ended",end)
end - start

## Output from jagsUI
out <- out1

Year.eff <- out$summary[(substr(rownames(out$summary),1,3)=="Y.p"),]
Year.eff.m <- out$summary[(substr(rownames(out$summary),1,5)=="Y.m.p"),]
Year.eff.t <- out$summary[(substr(rownames(out$summary),1,5)=="Y.t.p"),]
Year.trend <- out$summary[(substr(rownames(out$summary),1,4)=="Y.tr"),]
Year.trend.m <- out$summary[(substr(rownames(out$summary),1,5)=="Y.m.t"),]
Year.trend.t <- out$summary[(substr(rownames(out$summary),1,5)=="Y.t.t"),]
NP.prcp.est <- out$summary[(substr(rownames(out$summary),1,4)=="NP.p"),]
clim.est <- rbind( out$summary[(rownames(out$summary)=="mu"),],
            out$summary[(rownames(out$summary)=="u.precip"),],
            out$summary[(rownames(out$summary)=="u.temp"),],
            out$summary[(rownames(out$summary)=="u.int"),])
col.med <- 5
col.q2.5 <- 3
```

```
col.q97.5 <- 7
quants.range <- (3:7)

## Trend plot of NP response over both grass types
Year.level <- factor(2010:2016)
par(new=FALSE)
plot((2010:2016),Year.trend[,col.med], xlim=c(2009.5, 2016.5), ylim=c(0.1,0.6), type='n', axes=F, ylab="", xlab="")
polygon(c(2010:2016,2016:2010),c(Year.trend[,col.q97.5],rev(Year.trend[,col.q2.5])),col="grey90",border=NA)
lines((2010:2016),Year.trend[,col.med], lwd = 2)
par(new=TRUE)
boxplot(Year.eff[,col.q2.5:col.q97.5]~Year.level,ylim=c(0.1,0.6),range=0,pars=list(boxwex=0.1,staplewex=0,whisklty=1),
    ylab="Proportion desired native", xlab="Year",cex.axis=1.1,cex.lab=1.5)

## Trend plot of NP response by grass type
par(new=FALSE)
plot((2010:2016),Year.trend.m[,col.med], xlim=c(2009.5, 2016.5), ylim=c(0.1,0.6), type='n', axes=F, ylab="", xlab="")
polygon(c(2010:2016,2016:2010),c(Year.trend.m[,col.q97.5],rev(Year.trend.m[,col.q2.5])),col="tan3",density=30,angle=45,border=N
A)
lines((2010:2016),Year.trend.m[,col.med], lwd = 2)
#par(new=TRUE)  --  not necessary since polygon() and lines() are low-level plotting functions.
polygon(c(2010:2016,2016:2010),c(Year.trend.t[,col.q97.5],rev(Year.trend.t[,col.q2.5])),col="olivedrab2",density=30,angle=-
45,border=NA)
lines((2010:2016),Year.trend.t[,col.med], lwd = 2)
par(new=TRUE)
##  Replaced "at=1:7+/-0.1" with "at=2010:2016+/-0.1" in the two calls to boxplot().
boxplot(Year.eff.m[,col.q2.5:col.q97.5]~Year.level,xlim=c(2009.5,
2016.5),ylim=c(0.1,0.6),range=0,pars=list(boxwex=0.1,staplewex=0,whisklty=1),
    ylab="",xlab="",axes=FALSE,col="tan3",at=2010:2016-0.1)
boxplot(Year.eff.t[,col.q2.5:col.q97.5]~Year.level,range=0,pars=list(boxwex=0.1,staplewex=0,whisklty=1),col="olivedrab2",axes=FAL
SE,add=TRUE,at=2010:2016+0.1)
par(new=TRUE)
plot((2010:2016),Year.trend.t[,col.med],xlim=c(2009.5,2016.5),ylim=c(0.1,0.6),type='n',ylab="Proportion desired
native",xlab="Year",cex.axis=1.1,cex.lab=1.5)

##  The way I caught the discrepancy between the x-limits used in the plot vs. the boxplot calls was by checking the output of
par("usr").  This returns the bounding box of the plotting region, in the same units as the data being plotted -- very useful!

## Treatment x Precipitation interaction
# Exclude non-assignable treatments
NP.prcp.est <- NP.prcp.est[-c(9,10,19,20,29,30),]

prcp.level <- factor(levels=c(-2,0,2),labels=c('Dry','Average','Wet'))
#trt.level <- factor(levels=(1:5),labels=c('Rest','Graze','Fire','Combination','Other'))
trt.m.level <- factor(levels=(1:4),labels=c('Rest','Graze','Burn','Burn/Graze'))
trt.t.level <- factor(levels=(1:4),labels=c('Rest','Graze/Window','Burn/Window','Defoliate'))

#d <- as.data.frame(cbind(rep(c(1:3),each=5),rep(c(1:5),3),NP.prcp.est[,quants.range]))
d <- cbind(rep(c(1:3),each=4),rep(c(1:4),3))
d <- cbind(d[rep(1:nrow(d),each=2),],rep(c(1:2),nrow(d)))
d <- as.data.frame(cbind(d,NP.prcp.est[,quants.range]))
names(d)[1] <- paste("Prcp")
names(d)[2] <- paste("Trt")
names(d)[3] <- paste("Type")
d <- reshape(d,varying=list(names(d)[4:8]),direction="long",timevar="Pctile",times=c("2.5%","25%","50%","75%","97.5"),
        v.names="Proportion.NP",idvar=c("Prcp","Trt","Type"))
d$Prcp <- factor(d$Prcp,labels=c('Dry','Average','Wet'))
```

```
d.m <- d[d$Type==1,]
d.t <- d[d$Type==2,]
#d$Trt <- factor(d$Trt,labels=c('Rest','Graze','Fire','Combination','Other'))
d.m$Trt <- factor(d.m$Trt,labels=c('Rest','Graze','Burn','Burn/Graze'))
d.t$Trt <- factor(d.t$Trt,labels=c('Rest','Graze/Window','Burn/Window','Defoliate'))

library(ggplot2)
#ggplot(data = d, aes(x = Prcp, y = Proportion.NP)) + geom_boxplot(aes(fill=Trt), width=0.6) + theme_bw() +
#	labs(x="Prior-year Precipitation",y="Proportion of desired native",fill="Treatment")

d2 <- d.m[order(d.m$Trt,d.m$Prcp),]
ggplot(data = d2, aes(x = Trt, y = Proportion.NP)) + geom_boxplot(aes(fill=Prcp), width=0.6) + theme_bw() +
  coord_cartesian(ylim = c(0,0.8)) + scale_y_continuous(breaks=pretty(d2$Proportion.NP, n=4)) +
  labs(x="Treatment",y="Proportion desired native",fill="Prior-year \nPrecipitation") +
  theme(axis.text=element_text(size=18),axis.title=element_text(size=20),
	axis.title.x=element_text(vjust=-0.5),axis.title.y=element_text(vjust=2),
	legend.position=c(0.145,0.8),legend.text=element_text(size=16,hjust="c"),legend.title=element_text(size=16),
	legend.background=element_rect(fill="grey90"))

d2 <- d.t[order(d.t$Trt,d.t$Prcp),]
ggplot(data = d2, aes(x = Trt, y = Proportion.NP)) + geom_boxplot(aes(fill=Prcp), width=0.6) + theme_bw() +
  coord_cartesian(ylim = c(0,0.8)) + scale_y_continuous(breaks=pretty(d2$Proportion.NP, n=4)) +
  labs(x="Treatment",y="Proportion desired native",fill="Prior-year \nPrecipitation") +
  theme(axis.text=element_text(size=18),axis.title=element_text(size=20),
	axis.title.x=element_text(vjust=-0.5),axis.title.y=element_text(vjust=2),
	legend.position=c(0.855,0.8),legend.text=element_text(size=16,hjust="c"),legend.title=element_text(size=16),
	legend.background=element_rect(fill="grey90"))


## Plot response surface for climate-driven unit effect
library(lattice)
library(RColorBrewer)
library(colorspace)
resp.color <- brewer.pal(11,"BrBG")
resp.color <- rev(diverge_hcl(12,h=c(130,43),c=100,l=c(70,90)))
map.norm <- (climate[,2]-mean(climate[,2]))/sd(climate[,2])
mtwm.norm <- (climate[,3]-mean(climate[,3]))/sd(climate[,3])

clim.resp <- function(x, y){1 / (1 + exp(-(clim.est[1,1] + clim.est[2,1]*x + clim.est[3,1]*y + clim.est[4,1]*x*y)))}
prcp <- seq(300,700,length=200)
temp <- seq(17,25,length=200)
np.resp <- outer((prcp-mean(climate[,2]))/sd(climate[,2]),
                 (temp-mean(climate[,3]/10))/sd(climate[,3]/10),
                  clim.resp)
#persp(prcp,temp,np.resp)
#contour(prcp,temp,np.resp)
#grid <- expand.grid(prcp=prcp,temp=temp)
#grid$z <- as.vector(np.resp)
#levelplot(np.resp ~ prcp*temp,grid,xlim=c(300,650),ylim=rev(c(17,25)))
contourplot(np.resp, row.values=seq(300,700,length=200), column.values=seq(17,25,length=200),
            aspect=1, colorkey=TRUE, region=TRUE, xlim=c(300,700), ylim=rev(c(17,25)),
            xlab=list(label="Mean Annual Precipitation (mm)",cex=1.4),
            ylab=list(label="Mean Temperature of Warmest Month (C)",cex=1.4),
            scales=list(cex=1.4),labels=list(cex=1.1),label.style="align",col.regions=resp.color)
## Unsure how to place a legend on color key. Stupid approach below requires fiddling with text position
par(new=TRUE,mar=c(0.1,0.1,0.1,0.1))
plot(1:10,1:10,type="n",axes=FALSE,xaxt='n',yaxt='n',ann=FALSE,xlab="",ylab="")
```

```
text(8.6,5.8,"Mean Proportion Native Vegetation Cover",cex=1.1,srt=90)
#par(new=TRUE,pin=c(6,6))
#plot(climate[,2],climate[,3]/10, xlim=c(300,650), ylim=rev(c(17,25)), type='p', pch=1, axes=T,
#     col="black",ylab="", xlab="")
#surface3d(prcp,temp,np.resp)
```

```
model
 {
        ## Model for rainfall only, using both grass types
        ## Version computing trend as a derived variable

  ## Likelihood
  for (i in 1:n.obs)
   {

    logitp[i] <- mu + gamma[g[obs.year[i]]] +
                re.unit[obs.unit[i]] +
                b * prcp.norm[obs.year[i]] +
                c[trt[obs.year[i]]] +
                bc[trt[obs.year[i]]] * prcp.norm[obs.year[i]] +
                bg[g[obs.year[i]]] * prcp.norm[obs.year[i]] +
                cg[g[obs.year[i]],trt[obs.year[i]]] +
                bcg[g[obs.year[i]],trt[obs.year[i]]] * prcp.norm[obs.year[i]] +
                d * lagnp.norm[i] +
#                 e[year[i]-2009] +
                eg[g[obs.year[i]],year[i]-2009] +
                z.obs[i]
    ## Standardize lagnpp to mean
    lagnp.norm[i] <- lagnpp[i] - lagnpp.mean
    ## lagnp.norm[i] <- lagnpp[i] - 0.5
    z.obs[i] ~ dnorm(0,tau.obs)
    p[i] <- 1 / (1+exp(-logitp[i]))
    np[i] ~ dbin(p[i],stops[i])
    mean[i] <- p[i]*stops[i]
    denom[i] <- sqrt(mean[i]*(1-p[i])*p[i])

    Pearson.resi[i] <- (np[i] - mean[i]) / denom[i]
    D[i] <- pow(Pearson.resi[i],2)
    np.new[i] ~ dbin(p[i],stops[i])
    Pearson.resi.new[i] <- (np.new[i] - mean[i]) / denom[i]
    D.new[i] <- pow(Pearson.resi.new[i],2)

   }

  ## Bayesian P-value
  fit <- sum(D[])
  fit.new <- sum(D.new[])
  Bayes.pval <- step(fit.new - fit)

  ## Priors for fixed year effects and interaction with grass type
#   for (i in 1:6)
#    {
#
#      e_parm[i] ~ dnorm(0, 0.37)
#      e[i] <- e_parm[i]
#      eg_parm[i] ~ dnorm(0, 0.37)
#      eg[1,i] <- eg_parm[i]
#      eg[2,i] <- -eg[1,i]
#
#    }
#   e[7] <- -(e[1] + e[2] + e[3] + e[4] + e[5] + e[6])
#   eg[1,7] <- -(eg[1,1] + eg[1,2] + eg[1,3] + eg[1,4] + eg[1,5] + eg[1,6])
#   eg[2,7] <- -eg[1,7]
```

```
    ## Random year effect
    for (i in 1:6)
      {
#       e_val[i] ~ dnorm(0, tau.e)
#       e[i] <- e_val[i]
        eg_val[1,i] ~ dnorm(0, tau.eg1)
        eg_val[2,i] ~ dnorm(0, tau.eg2)
        eg[1,i] <- eg_val[1,i]
        eg[2,i] <- eg_val[2,i]
#       eg[2,i] <- -eg[1,i]
      }
#     e[7] <- -(e[1] + e[2] + e[3] + e[4] + e[5] + e[6])
    eg[1,7] <- -(eg[1,1] + eg[1,2] + eg[1,3] + eg[1,4] + eg[1,5] + eg[1,6])
    eg[2,7] <- -(eg[2,1] + eg[2,2] + eg[2,3] + eg[2,4] + eg[2,5] + eg[2,6])
#     eg[2,7] <- -eg[1,7]

    ## Priors for random year effects
#     tau.e <- pow(sd.e,-2)
#     sd.e ~ dunif(0,10)
    tau.eg1 <- pow(sd.eg1,-2)
    sd.eg1 ~ dunif(0,10)
    tau.eg2 <- pow(sd.eg2,-2)
    sd.eg2 ~ dunif(0,10)

    ## Unit-level climate effects model
#     for (i in 1:n.unit)
#       {
#
        #                                   re.unit[i] ~ dnorm(0, unit.tau)
        #                                   b[i] ~ dnorm(b.mean, b.tau)
#       re.unit[i] <- u.precip*map.norm[i] + u.temp*mtwm.norm[i] +
#                     u.int*map.norm[i]*mtwm.norm[i]
#
#       }
    re.unit <- u.precip*map.norm + u.temp*mtwm.norm + u.int*map.norm*mtwm.norm

    ## Normalize precipitation for use as predictor
#     for (i in 1:n.unit.year)
#       {
#
#       prcp.norm[i] <- (prcp.raw[i]-prcp.mean) / prcp.sd
#
#       }
    prcp.norm <- (prcp.raw-prcp.mean) / prcp.sd

    ## Estimation of missing predictor values
     for (i in 1:n.miss.lag)
       {

        #                                   lagnpp[miss.lag[i]] ~ dunif(0.01, 0.99)
        miss.lag.val[i] ~ dbeta(1, 1)
        lagnpp[miss.lag[i]] <- miss.lag.val[i]

       }

     for (i in 1:n.miss.trt)
```

```
    {

      miss.trt.val[i] ~ dcat(trtprob[])
      trt[miss.trt[i]] <- miss.trt.val[i]

    }

  for (i in 1:5) { trtprob[i] <- 0.2 }

  for (i in 1:n.miss.prcp)
    {

      miss.prcp.val[i] ~ dnorm(prcp.mean, prcp.prec)
      prcp.raw[miss.prcp[i]] <- miss.prcp.val[i]

    }

## Priors for treatment effects and interactions with precip, grassland type
for (i in 1:4)
  {

    c_parm[i] ~ dnorm(0, 0.37)
    c[i] <- c_parm[i]
    bc_parm[i] ~ dnorm(0, 0.37)
    bc[i] <- bc_parm[i]
    cg_parm[i] ~ dnorm(0, 0.37)
    cg[1,i] <- cg_parm[i]
    cg[2,i] <- -cg[1,i]
    bcg_parm[i] ~ dnorm(0, 0.37)
    bcg[1,i] <- bcg_parm[i]
    bcg[2,i] <- -bcg[1,i]

  }

## Zero-sum constraints
c[5] <- -(c[1]+c[2]+c[3]+c[4])
bc[5] <- -(bc[1]+bc[2]+bc[3]+bc[4])
cg[1,5] <- -(cg[1,1]+cg[1,2]+cg[1,3]+cg[1,4])
cg[2,5] <- -cg[1,5]
bcg[1,5] <- -(bcg[1,1]+bcg[1,2]+bcg[1,3]+bcg[1,4])
bcg[2,5] <- -bcg[1,5]

## Priors for grassland type and rainfall interaction effects
gamma_parm ~ dnorm(0, 0.37)
gamma[1] <- gamma_parm
gamma[2] <- -gamma[1]
bg_parm[1] ~ dnorm(0, 0.37)
bg[1] <- bg_parm[1]
bg[2] <- -bg[1]

## Priors for unit-level climate effects
u.precip ~ dnorm(0, 0.37)
u.temp ~ dnorm(0, 0.37)
u.int ~ dnorm(0, 0.37)

mu ~ dnorm(0, 0.37)
```

```
  ## Priors for main regression terms
  b ~ dnorm(0, 0.37)
  d ~ dnorm(0, 0.37)

  ## Priors for observation-level random effects
  tau.obs <- 1/pow(sd.obs,2)
  sd.obs ~ dunif(0,10)
  prcp.prec <- 1/(prcp.sd*prcp.sd)

  ## Derived quantities for effects
  for (j in 1:2) {
   for (i in 1:5){
    T.int[j,i] <- mu + gamma[j] + c[i] + cg[j,i]
    T.prcp[j,i] <- b + bg[j] + bc[i] + bcg[j,i]
    for (k in 1:3){
     NP.prcp[j,i,k] <- 1/(1 + exp(-(T.int[j,i] + T.prcp[j,i]*(2*k-4))))
    }
   }
  }

  ## Derived quantities for trend (2010-2016)
  for (i in 1:7){
#     Y[i] <- mu + e[i]
   Y[i] <- mu + (eg[1,i]+eg[2,i])/2
   Y.p[i] <- 1/(1 + exp(-Y[i]))
   Y.trend[i] <- 1/(1 + exp(-(mu + yeartrend*(i-4))))
#     Y.m[i] <- mu + gamma[1] + e[i] + eg[1,i]
   Y.m[i] <- mu + gamma[1] + eg[1,i]
   Y.m.p[i] <- 1/(1 + exp(-Y.m[i]))
   Y.m.trend[i] <- 1/(1 + exp(-(mu + gamma[1] + yeartrend.m*(i-4))))
#     Y.t[i] <- mu + gamma[2] + e[i] + eg[2,i]
   Y.t[i] <- mu + gamma[2] + eg[2,i]
   Y.t.p[i] <- 1/(1 + exp(-Y.t[i]))
   Y.t.trend[i] <- 1/(1 + exp(-(mu + gamma[2] + yeartrend.t*(i-4))))
  }
  yeartrend <- (-3*Y[1] - 2*Y[2] - 1*Y[3] + 0*Y[4] + 1*Y[5] + 2*Y[6] + 3*Y[7])/28
  yeartrend.m <- (-3*Y.m[1] - 2*Y.m[2] - 1*Y.m[3] + 0*Y.m[4] + 1*Y.m[5] + 2*Y.m[6] + 3*Y.m[7])/28
  yeartrend.t <- (-3*Y.t[1] - 2*Y.t[2] - 1*Y.t[3] + 0*Y.t[4] + 1*Y.t[5] + 2*Y.t[6] + 3*Y.t[7])/28

  }
```