5-2000

Comparing Nonlinear and Nonparametric Modeling Techniques for Mapping and Stratification in Forest Inventories of the Interior Western USA

Gretchen Gengenbach Moisen
Utah State University

COMPARING NONLINEAR AND NONPARAMETRIC MODELING TECHNIQUES FOR MAPPING AND STRATIFICATION IN FOREST INVENTORIES OF THE INTERIOR WESTERN USA

by

Gretchen Gengenbach Moisen

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Mathematical Sciences

UTAH STATE UNIVERSITY
Logan, Utah

2000

ABSTRACT

Comparing Nonlinear and Nonparametric Modeling Techniques

for Mapping and Stratification in Forest Inventories

of the Interior Western USA

by

Gretchen Gengenbach Moisen, Doctor of Philosophy

Utah State University, 2000

Major Professor: Dr. D. Richard Cutler
Department: Mathematics and Statistics

Recent emphasis has been placed on merging regional forest inventory data with satellite-based information both to improve the efficiency of estimates of population totals, and to produce regional maps of forest variables. There are numerous ways in which forest class and structure variables may be modeled as functions of remotely sensed variables, yet surprisingly little work has been directed at surveying modern statistical techniques to determine which tools are best suited to the tasks given multiple objectives and logistical constraints. Here, a series of analyses to compare nonlinear and nonparametric modeling techniques for mapping a variety of forest variables, and for stratification of field plots, was conducted using data in the Interior Western United States. The analyses compared four statistical modeling techniques for predicting two discrete and four continuous forest inventory variables. The modeling techniques include generalized additive models (GAMs), classification and regression trees (CARTs), multivariate adaptive regression splines (MARS), and artificial neural networks (ANNs). Alternative stratification schemes were also compared for estimating population totals. The analyses were conducted within six ecologically different regions using a variety of satellite-based predictor variables. The work resulted in the development of an objective modeling box that automatically models spatial response variables as functions of any assortment of predictor variables through the four nonlinear or nonparametric modeling techniques. In comparing the different modeling techniques, all proved themselves workable in an automated environment, though ANNs were more problematic. When their potential mapping ability was explored through a simple simulation, tremendous advantages were seen in use of MARS and ANN for prediction over GAMs, CART, and a simple linear model. However, much smaller differences were seen when using real data. In some instances, a simple linear approach worked virtually as well as the more complex models, while small gains were seen using more complex models in other instances. In real data runs, MARS performed (marginally) best most often for binary variables, while GAMs performed (marginally) best most often for continuous variables. After considering a subjective "ease of use" measure, computing time and other predictive performance measures, it was determined that MARS had many advantages over other modeling techniques. In addition, stratification tests illustrated cost-effective means to improve precision of estimates of forest population totals. Finally, the general effect of map accuracy on the relative precision of estimates of population totals obtained under simple random sampling compared to that obtained under stratified random sampling was established and graphically illustrated as a tool for management decisions.

(191 pages)

ACKNOWLEDGMENTS

I would like to thank my advisor, Richard Cutler, as well as the members of my committee, Adele Cutler, Dan Coster, Joe Koebbe, Dave Roberts, and Rob Lilieholm, for their help through the graduate process and input into the research. I am indebted to the Forest Inventory and Analysis Program of the US Forest Service for funding, data, and for providing wonderfully interesting problems. I thank the good folks at FIA including: Dwane Vanhooser for his bottom-line perspective and support, Renee Obrien for her flexibility and willingness to listen, Ron Tymcio for years of laughs, and Tracey Frescino, without whom I never would have finished. I am grateful for the help I received from many others including Shirley Waters, Sharon Woudenberg, Dennis Collins, Keith Lannom, Bill Cooke, Ray Czaplewski, and Tamre Cardoso, to name a few. I am grateful to Dave Chojnacky for putting me on the research path, to Tom Edwards for many, many things (except for introducing me to T.B. Murphy), and to friends and colleagues all over the map for their relentless nagging to finish. Thank you too, to my extended family for their tireless support, to my friends and neighbors who kept my son safe and happy during crazy times, to my mom-in-law for her love of research and for really sharp pencils, to my sister who always let me whine, to my mom for grape jelly, and to my brother who was smarter than anybody. I especially thank my husband, Steve for his support and willingness to put up with life with a graduate student for over a decade, and my son, Ben, who always reminded me of what was important.

Gretchen Gengenbach Moisen

CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Problem Description

Forest inventory programs, like those conducted by the USDA Forest Service, Forest Inventory and Analysis (FIA) program in the Rocky Mountain Research Station, are under increased pressure to produce more information, more often, at reduced costs. The traditional objective of FIA has been to estimate statewide forest population totals (e.g., forest area, volume, growth and mortality) approximately every 10 years. Historically, this has been accomplished through a two-phase sampling procedure with phase one consisting of aerial photo-based information collected on a 1-km sample grid, and phase two consisting of a subset of that grid (usually 5 km) visited in the field (Chojnacky 1998). Photo-interpreted cover-type and ownership are typically used for stratification of phase two field points, resulting in improved precision of estimates of forest population totals. This strategy of combining aerial photo and field data through double sampling for stratification has been used by FIA in the Rocky Mountain States for many years.

While the current two-phase sampling procedure used by FIA provides unbiased and precise estimates of forest resources at regional scales, some problems exist. For example, aerial photography available in any given state will vary in quality, scale, and date. Also, inconsistencies exist between photo-interpreters in terms of correct location of sample points on photos and correct vegetation classification. In addition, the process is extremely expensive and slow. Consequently, there is a need to develop methods that use satellite data in lieu of photo-interpretation (PI) that maintain the required precision in FIA estimates of population totals.

In addition to this need to improve the two-phase estimation process, there is also a need to expand the forest inventory product line to include maps of forest resources. The most valuable management tool to many land managers is a map depicting the spatial arrangement of forest attributes at resolutions finer than those obtainable from current FIA sampling grids. These can be difficult to generate. While vegetation cover-type maps produced by programs like the USDI Gap Analysis program (see Homer, Ramsey, Edwards, and Falconer 1997; Scott et al. 1993) have been useful in meeting the need for fine-scaled information, these maps are extremely expensive to produce and lack any spatial depiction of structural attributes (e.g., basal area, canopy closure, stand density) for their forest types. This reduces their usefulness for identifying suitable wildlife habitat (Edwards, Deshler, Foster, and Moisen 1996), or for estimating forest characteristics necessary for sound forest management such as attributes of vegetation under the trees' canopies (Stenbeck and Congalton 1990), or stand density and volume (Franklin 1986).

Consequently, recent emphasis has been placed on merging forest inventory data with satellite information both to improve the efficiency of estimates of population totals through less expensive stratification, and produce regional maps of forest variables. There are numerous sources of ancillary data, and tremendous effort has been directed at acquiring finer resolution data from a wide variety of newly developed air- and space-borne platforms. There are also numerous ways in which forest class and structure variables may be modeled as functions of remotely sensed and other ancillary variables, yet surprisingly little work has been directed at surveying modern statistical techniques to determine which tools are best suited to estimation and mapping tasks given multiple objectives and logistical constraints.

Preliminary Work

Recent work by Moisen and Edwards (1999) explored ways to merge forest inventory and satellite-based data in Northern Utah. In this study, generalized linear models (GLMs) were used to construct approximately unbiased and efficient estimates of population totals while providing a mechanism for prediction to map forest structure in space. Forest type and timber volume of five tree species groups were modeled as functions of a variety of satellite-based predictor variables. Predictor variables included elevation, aspect, slope, geographic coordinates, and vegetation cover-types based on satellite data from both the Advanced Very High Resolution Radiometer (AVHRR) and Thematic Mapper (TM) platforms. The relative precision of estimates of area by forest type and mean cubic-foot volumes under six different models, including the traditional double sampling for stratification (DSS) strategy, was examined.

The study generated some interesting results. First, only very small gains in precision were realized using expensive photo-interpreted or TM-based data for stratification, while models based on topography and spatial coordinates alone were competitive. This had substantial cost-savings implications for phase one in the two-phase sampling process. Second, after comparing the predictive capability of the models through cross-validated map accuracy measures, the models including the TM-based vegetation were shown to perform best overall, while topography and spatial coordinates alone provided substantial information at very low cost. In addition, the study illustrated that by using a fairly flexible model form, i.e., a GLM with higher order interactions and polynomial terms, more information could be squeezed out of inexpensive auxiliary digital data than previously thought possible in these inventories.

The study raised a number of questions. GLMs did well, but would more flexible models show appreciable improvements in the results? What kind of results would we see in other ecological regions? What kind of results would we see using data from different satellite platforms? Could the process be automated to be suitable for a production environment? (A production environment implies that someone without modeling experience can push the button that builds the models and produces desired output for any ecoregion, response variable, or predictor set.) These and other questions motivated the following dissertation.

Dissertation Overview and Objectives

In this dissertation, nonlinear and nonparametric models were compared for mapping and stratification in forest inventories of the interior western United States. The research involved five statistical modeling techniques for predicting two discrete and four continuous forest inventory variables. The modeling techniques included: generalized additive models (GAM), classification and regression trees (CART), multivariate adaptive regression splines (MARS), and artificial neural networks (ANN). In addition, a simple linear model (LM) was used as a benchmark against which to judge the other models. The two discrete inventory variables included a forest/non-forest classification, as well as a binary classification within forested areas. The four continuous response variables were tree biomass per acre (BIOTOT), average tree age (STAGE), quadratic mean tree diameter (QMDALL), and percent crown cover (CRCOV). The analyses were conducted within six ecologically different regions (two each in Arizona, Montana, and Utah). Predictor variables included topography, spatial position, unclassified spectral data from the AVHRR sensor, and a national vegetation cover map derived from TM imagery. Models were first built, refined, and automated using data in one region. Automated modeling strategies were then applied to data in all regions to evaluate model performance. Predictive performance (map accuracy) of all discrete and continuous variables were compared across modeling techniques, ecoregions, and predictor variable sets using independent test data. In addition, the precision of estimates of area by forest type, total population tree volume, and total population tree growth were compared when predicted forest type maps were used as the basis for stratification (i.e., predicted vegetation types comprise the strata). All models were evaluated for suitability in a production environment.

Objectives of this research were:

Objective 1: To develop an automated mapping and stratification system for forest inventories in the interior west.

Objective 2: To determine which statistical modeling techniques are suitable for a forest inventory "production environment."

Objective 3: To determine if introducing more flexible statistical models to forest inventory mapping and stratification procedures makes an appreciable difference in accuracy of forest maps and precision in estimates of population totals, respectively.

The phrase "production environment" above implies that model fitting must be completely automated requiring nothing more than the push of a button. The phrase "appreciable difference" is defined by usability standards set by forest managers and other users of forest inventory information.

Questions answered about modeling two discrete (binary) variables include:

Question 1: How accurately does each modeling technique predict the two responses by ecoregion?

Question 2: What is the precision of estimates of population totals by ecoregion when predicted maps are used as the basis for stratification?

Question 3: What is the relationship between classification accuracy of maps and efficiency of stratification (i.e., reduction in standard errors on population estimates when maps are used as the basis for stratification)?

Questions answered about modeling four continuous variables include:

Question 4: How accurately does each modeling technique predict the four responses by ecoregion?

Question 5: Is there a substantial improvement in map accuracy over simply assigning stratum means for each response?

The outcomes of this research include:

Outcome 1: Development of less costly strategies for stratifying forest inventory field data in the interior west.

Outcome2: Development of a methodology for mapping diverse forest inventory variables that is suitable for the FIA production environment.

Road Map

A technical description and literature review of the modeling techniques and stratification procedures is provided next in Chapter 2. Chapter 3 contains a description of the study regions, field response variables, and digital satellite data used as predictor variables along with data extraction processes. Chapter 4 documents the development of the objective modeling environment, and describes the evaluation criteria used in subsequent analyses. Chapter 5 contains mapping and stratification results for discrete and continuous response variables in all ecoregions. These results and their implications are discussed in Chapter 6, along with conclusions and ideas for further research.

CHAPTER 2

DESCRIPTION AND LITERATURE REVIEW OF MODELING TECHNIQUES

In the following chapter, a description of each of the nonparametric or nonlinear modeling techniques (GAMs, CARTs, MARS, and ANNs) is given, followed by a technique-specific literature review. To review all the applied literature employing each technique would fill many chapters. Consequently, the intent of the literature review following the description of each technique was to reference ecological and remote sensing applications that illustrated sound model fitting strategies, evaluated strengths and shortcomings of the techniques, or compared relative performance of one technique over another in a natural resource setting. DeVeaux, Psichogios, and Ungar (1993) and DeVeaux (1995) provide more general discussions comparing these modeling techniques, and Table 2-1 illustrates technique differences at-a-glance.

Generalized Additive Models

GAM Overview

Generalized additive models (Hastie and Tibshirani 1986) are an extension of generalized linear models (GLMs) (Nelder and Wedderburn 1972), which are, in turn, an extension of the classical linear model. We assume the response y has an exponential family density (e.g., gamma, chi-square, beta, binomial, Poisson, negative binomial, etc.) with mean linked to the predictors via

$$\mu = f(\mathbf{x}) = g^{-1}\left(a_0 + \sum_{i=1}^{p} f_i(\mathbf{x}_i) \right),$$

Table 2-1. Overview of Candidate Modeling Techniques.

	Formulation	Strengths	Limitations
GAM	$f(\mathbf{x}) = g^{-1}\left(a_0 + \sum_{i=1}^{p} f_i(\mathbf{x}_i)\right)$	Interpretable if original predictors have intrinsic meaning to investigator	Limited to lower order interactions
CART	$f(\mathbf{x}) = a_m, \text{for } \mathbf{x} \in R_m$	Exploits low local dimensionality of functions Can handle interactions Interpretable if tree is simple Rapid to construct and make predictions from	Approximation function discontinuous at sub-region boundaries Simple functions can be difficult to approximate Interpretation is complex with many inner branches representing higher-order interactions
MARS	$f(\mathbf{x}) = a_0 + \sum_{K_m=1} f_i(\mathbf{x}_i) + \sum_{K_m=2} f_{ij}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{K_m=3} f_{ijk}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) + \ldots$	Final function is smooth New splits aren't dependent on previous splits Graphical displays of main effects and low order interactions Selects predictor variables, order of interaction, and amount of smoothing automatically Competitive with ANN when few active predictors and low interaction order	No predictions intervals given All uncertainty estimates must be done via cross-validation Highly collinear predictors lead to highly erratic behavior and loss of interpretability
ANN	$f_l(\mathbf{x}) = \sigma\left(\sum_{k=1}^{K} w_{2kl}\,\sigma\left(\sum_{j=1}^{J} w_{1jk}\mathbf{x}_\mathrm{j} + \theta_k\right) + \theta_l\right).$	Often outperforms all others for prediction Can be used directly to predict multiple responses No implicit assumptions of linearity, normal or i.i.d. errors	Un-interpretable No prediction intervals given All uncertainty estimates must be done via cross-validation Caution has to be exercised to avoid "over-fitting," modeling noise as well as underlying phenomenon Other methods may be preferable for low dimensional or simple structure

where the link function may be any monotonic, differentiable function, and $f_1, f_2, \ldots, f_p$ are smooth functions estimated in a nonparametric fashion. A local scoring algorithm is used to estimate the f_j's. This algorithm uses scatterplot smoothers to generalize the usual Fisher scoring procedure for computing maximum likelihood estimates. Any scatterplot smoother can be used, such as a running mean, running median, Loess, kernel estimate, or spline, (see Hastie and Tibshirani 1990, for a discussion on smoothers). The resulting smooth functions can be used for data description, prediction, or to suggest covariate transformations such as polynomial terms resulting in simplified parametric, or semi-parametric models. GAMs have the advantage over simple linear models of being able to model nonlinear relationships in the predictor variables. For large data sets, this flexibility can yield better predictive capability and provides greater opportunity for exploratory analyses. For those familiar with regression methods, GAMs may be more easily interpreted than regression trees, and they provide a continuous predicted response. An open question, however, is how to handle interactions among the predictor variables. In a case where the number of predictor variables is few, bivariate functions may be estimated using bivariate smoothers. When the number of predictor variables is large, deciding which pairs of variables to model simultaneously in a GAM can be difficult and time consuming (as it is with linear models). In addition, GAMs require crossvalidation methods to determine appropriate levels of smoothness (see Hastie and Tibshirani 1990). This, too, can be computationally intensive.

GAMs in Natural Resource Mapping Applications

GAMs have caught the attention of ecologists for mapping applications, and some of the advantages of this class of models are expressed in Yee and Mitchell (1991). They have been used to model the presence of several tree species as functions of climatic variables in New Zealand (Yee and Mitchell 1991), to conduct wildlife viability analyses in Australia (Norton and Mitchell 1993), to model vegetation composition as functions of topography and disturbance variables in Glacier National Park (Brown 1994), and to predict the distribution of Eucalyptus species (Austin, Nicholls, Doherty, and Meyers 1994; Austin and Meyers 1996). In the forest inventory arena, Moisen, Edwards, and Cutler (1996) used GAMs to model species presence and tree volume as functions of topography and a TM-based cover type map. Most recently, Frescino, Edwards,and Moisen (in press) used GAMs to model forest type, basal area, shrub cover, and snag density as functions of TM- and AVHRR-derived products, temperature precipitation topography and geology in the Uinta Mountain Range in Utah.

Classification and Regression Trees

CART Overview

Classification and regression trees, also known as *recursive partitioning regression*, dates back to Morgan and Sonquist (1963) and has received more recent attention through Breiman, Friedman, Olshen, and Stone (1984). (My use of the acronym here is not to be confused with any proprietary software or trademarks.) CARTs subdivide the space spanned by the predictor variables into regions $\{R_m\}$ for which the values of the response variable are approximately equal, and then estimate the response variable by a constant, a_m, in each of these regions. That is,

$$f(\mathbf{x}) = a_m, \text{for } \mathbf{x} \in R_m .$$

The tree is called a *classification tree* if the response variable is qualitative, and a *regression tree* if the response variable is numeric. The initial node on a tree is called the root. From the root, the model is fit using binary recursive partitioning. This means the data are successively broken into left and right branches with the splitting rules defined by the predictor variable values. For example, a first split might occur where $x_1 < c_1$, where c_1 is a constant. Then, $\hat{f}(\mathbf{x}) = a_1, \text{for } x_1 < c_1$, and $\hat{f}(\mathbf{x}) = a_2, \text{for } x_1 \geq c_1$. A second split might occur where $x_1 < c_1$ and $x_2 < c_{2,}$, and so on. Splits are chosen that maximize the "value" of a split. This value may be computed in many different ways. For classification problems, splits are chosen that most reduce the impurity of the distribution at the node, while in regression problems, the value of a split is measured as the reduction in the residual sum of squares. Splitting continues down to the "terminal" nodes where response values are all the same within a node or data are too sparse for additional splitting. At the terminal node, the predicted response is given that is the average or majority of the response values in that node for continuous or discrete variables, respectively. Pruning the tree to avoid overfitting the data can be accomplished a number of different ways, and is discussed further in Chapter 4.

Strengths of CARTs include the facts that interactions are accommodated through the splitting process and trees with low order interactions, in particular, are easy to interpret and explain to non-technical audiences. This can be important when considering inclusion of auxiliary data from a variety of satellite platforms. However, disadvantages of CARTs include discontinuity at the nodes and the poor approximation of simple functions (like straight lines).

CART in Natural Resource Mapping Applications

Classification and regression trees have received a considerable amount of attention in the natural resource arena. They have been used in a wide variety of applications including studying the effect of a variety of factors on the establishment of oak seedlings (Michaelson, Davis, and Borchert 1987), predicting Christmas tree growth (Hockman, Burger, and Smith 1990, assessing the effect of human disturbance on breeding in bald eagles (Grubb and King 1991), extracting the relationship between plant disease and pine mortality (Baker, Verbyla, and Hodges 1993), and predicting soil drainage class using remotely sensed and digital elevation data (Cialella, Dubayah, Lawrence, and Levine 1997). Their specific use in land cover mapping applications has also grown rapidly. Friedl and Brodley (1997) compared decision trees to maximum likelihood and linear discriminant function classifiers in land cover mapping applications and found that the trees consistently outperformed the other methods in classification accuracy. Vogelmann, Sohl, and Howard (1998) developed decision making rules and models using several ancillary data layers to resolve confusion in spectral classes that represented two or more targeted land cover categories. The approach led to adoption of CART methodology in early national land cover mapping, and modification of CARTs in conjunction with other modeling techniques for current national vegetation mapping efforts directed by the US Geological Survey.

Multivariate Adaptive Regression Splines

MARS Overview

MARS, developed by Friedman (1991) is a flexible nonparametric regression method that generalizes the piecewise constant functions of CART to continuous functions by fitting (multivariate) splines in the regions $R_{m,}$ and matching up the values at the boundaries of the R_m. An intuitive form for writing the MARS model is

$$\hat{f}(\mathbf{x}) = a_0 + \sum_{K_m=1} f_i(\mathbf{x}_i) + \sum_{K_m=2} f_{ij}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{K_m=3} f_{ijk}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) + \ldots,$$

but the notation requires further explanation. Here, the first sum is over all basis functions that involve only one variable. Each function in this first sum can be expressed as

$$f_i(\mathbf{x}_i) = \sum_{\substack{K_m=1 \\ i \in V(m)}} a_m B_m(\mathbf{x}_i)$$

where *V(m)* is the variable set associated with the m^{th} basis function, $B_{m,}$ that survives backward selection strategies. The second sum is over all basis functions that involve two variables, where each bivariate function can be expressed as

$$f_i(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\substack{K_m=2 \\ i,j \in V(m)}} a_m B_m(\mathbf{x}_i, \mathbf{x}_j).$$

The third sum is over all basis functions that involve three variables, and so on.

MARS is not subject to some of the limitations of GAMs because it automatically selects the amount of smoothing required for each predictor as well as the interaction order of the predictors. This makes it perhaps more suitable for a production environment where time consuming and subjective modeling decisions are undesirable.

In cases of moderate sample sizes, 50<N<1000, and moderate to high dimension, 3<n<20, MARS has proven effective for applications where both prediction and understanding are the objective (DeVeaux 1995). It has the desirable properties of rapid computability and, unlike CART, offers smoothness as a property as well. Unlike linear models and GAMs, however, MARS does not provide prediction intervals and all estimates of uncertainty must computed by crossvalidation.

MARS in Natural Resource Mapping Applications

Use of MARS in the applied literature is sparse, and apparently nonexistent in ecological or remote sensing applications. The reason may be in part because user-friendly software is not readily available, nor has the modeling technique been "marketed" in high-profile applications.

Artificial Neural Networks

ANN Overview

Neural networks have received considerable attention as a means to build accurate models for prediction, control, and optimization when the functional form of the underlying equations is unknown. This modeling technique has permeated literature in many fields including statistics (e.g., Ripley 1994, 1996; Stern 1996; Cheng and Titterington 1994), remote sensing (e.g., Atkinson and Tatnall 1997; Skidmore, Turner, Brinkhof, and Knowls 1997; Wang and Dong 1997) and ecology and engineering (Paruelo and Tomasel 1997; Wythoff 1993).

Although there are a variety of ways to construct these models, “backpropagation networks” appear to be the most frequently used in practice. A backpropagation network with one hidden layer is a nonlinear statistical model of the form

$$f_l(\mathbf{x}) = \sigma\left(\sum_{k=1}^{Kl} w_{2kl}\sigma\left(\sum_{j=1}^{J} w_{1jk}\mathbf{x}_{\mathrm{j}} + \theta_k \right) + \theta_l \right).$$

The response (*output*) is a transformation of a weighted combination of the predictor (*input*) variables. The σ in the above equation is a bounded, monotonic, and differentiable function, with a logistic function the most common choice. That is,

$$\sigma(x) = \frac{\exp(x)}{1+\exp(x)}.$$

The numerous coefficients *w* (weights) and intercepts θ (bias terms) are estimated (also known as *training* or *learning* in neural network jargon) through an optimization method similar to steepest descent (*backpropagation*). Because so many parameters can be estimated, there is danger in overfitting the model. By sacrificing an unlimited number of degrees of freedom, a modeler can eventually get a perfect fit. In that case one would be modeling noise as well as the underlying phenomenon, and prediction for unvisited sites could be severely compromised. The preferred method to avoid overfitting involves using a large enough network to avoid underfitting, then limiting the number of iterations of the fitting procedure through crossvalidation. Neural networks are frequently used for prediction in high dimensional problems like those permeating the engineering fields. As with MARS and CART, no prediction intervals are given and crossvalidation is necessary to construct measures of uncertainty. Neural networks are difficult, if not impossible, to interpret. They are hard to visualize, and two very different functional forms can yield the same predicted values. In addition, local minima in the objective function present obstacles to finding a reasonable model.

ANN in Natural Resource Mapping Applications

A large body of remote sensing literature is dedicated to the use of neural networks in classification of digital satellite data. Atkinson and Tatnall (1997) described the use of neural networks in remote sensing literature over the past decade, and review common approaches. ANNs have received both positive and negative reviews, largely based on computational issues. Paruelo and Tomasel (1997) compared ANNs to regression models in their ability to predict functional characteristics of ecosystems, showing clear advantages to the ANNs. Skidmore et al. (1997) mapped eucalypt forests and concluded that ANNs do not offer significant advantages over conventional classification schemes while Zhang and Yuan (1997) preferred neural networks to traditional remote sensing approaches for modeling vegetation types using TM data in northern Arizona. Successes have also been documented by Bruzzone, Conese, Maselle, and Roll (1997) when using neural networks to identify complex rural areas. Gong, Pu, and Chen (1996) described the technical aspects of using multiple data inputs at a variety of scales in mapping ecological land systems through neural networks. Kanellopoulos and Wilkinson (1997) offered substantial advice on "best practice" techniques to optimize network training and overall classification performance. They described their experiences related to network architecture, optimization algorithms, and transformation of input data to name a few. Foody and Aurora (1997) evaluated some of the factors affecting the accuracy of classification using neural networks, illustrating how dimensionality of a data set and characteristics of the training versus test set affect classification more than changes in network architecture. Paola and Schowengerdt (1997) also illustrated hidden layer size made little difference in final classification accuracies. Other examples include the use of neural networks in cloud classification (Lewis, Cote, and Tatnall 1997), sub-pixel analysis (Atkinson, Cutler, and Lewis 1997; Foody, Lucas, Curran, and Honzak 1997), and modeling stand size and density (Wang and Dong 1997). There is also an increased use in remotely sensed change detection projects. (Dai and Khorram 1999; Levien et al. 1999).

The use of ANNs in ecology has also grown rapidly over the last decade. Colasanti (1991) and Edwards and Morse (1995) saw the potential for ANNs in modeling complex ecological systems. Recent applications of neural networks in the ecological literature are quite diverse. They have been used in predicting the presence of a Himalayan river bird (Manel, Dias, and Omerod 1999), estimating the daily pH of a river as a function of river discharge and solar radiation (Moatar, Fessant, and Poirel 1999), studying the relationship between lead concentration in grass and urban descriptors (Dimopoulos, Chronopoulos, Chronopolous-Sereli, and Lek 1999), selecting a minimal set of driving variables to model water vapor and carbon exchange of coniferous forest ecosystems (Van Wijk and Bouten 1999), estimating phytoplankton production (Scardi and Harding 1999), modeling ocean color (Brosse, Guegan, Toureng, and Lek 1999), modeling the abundance and diversity of arthropods (Lek-Ann, Deharveng, and Lek 1999), discriminating between natural and hatchery brown trout (Aurelle, Lek, Giraudel, and Berrebi 1999), predicting primary production in a coastal embayment (Barciela, Garcia, and Fernandez 1999), and the list goes on.

CHAPTER 3

DATA DESCRIPTION

Study Regions and Sample Design

Portions of six ecologically different regions defined by Bailey, Avers, King, and McNab (1994) were selected for analyses and are illustrated in Figure 3-1. Ecoregions range from the coniferous forests of northwestern Montana, to the Chihuahuan Desert of southeastern Arizona. MT1 and MT2 refer to two ecoregions in Montana, UT1 and UT2 are two within Utah, and AZ1 and AZ2 are in Arizona. Table 3-1 summarizes characteristics of each of the ecoregions along with available field plot data from FIA databases. Dates of forest inventory, sample grid intensity, and field plot layout differ by ecoregion as well as by land owner and vegetation type. Figures 3-2 and 3-3 illustrate the different types of grid patterns and plot layout, respectively. In MT1 and MT2, field data were collected in 1988-1989 on a 5-km grid across all ownerships except National Forest. Timberland areas (characterized by tree species normally preferred for commercial harvest) were sampled with 10-point variable radius plots, while forests not dominated by commercial species were sampled with fixed radius plot shapes. Data on National Forests in these two ecoregions were collected 4 years later, also on a 5-km grid, using 5-point variable radius plots on timberland locations. In UT1 and UT2, data were collected in the mid-1990's on a double 5-km grid on National Forest lands and a 5-km grid elsewhere. The phrases “double 5 km” and “double 10 km” imply twice as many plots as on a 5- or 10-km grid, respectively (see Figure 3-2). A new fixed radius plot was introduced in Arizona, and data were collected on a 5-km grid on National Forests and

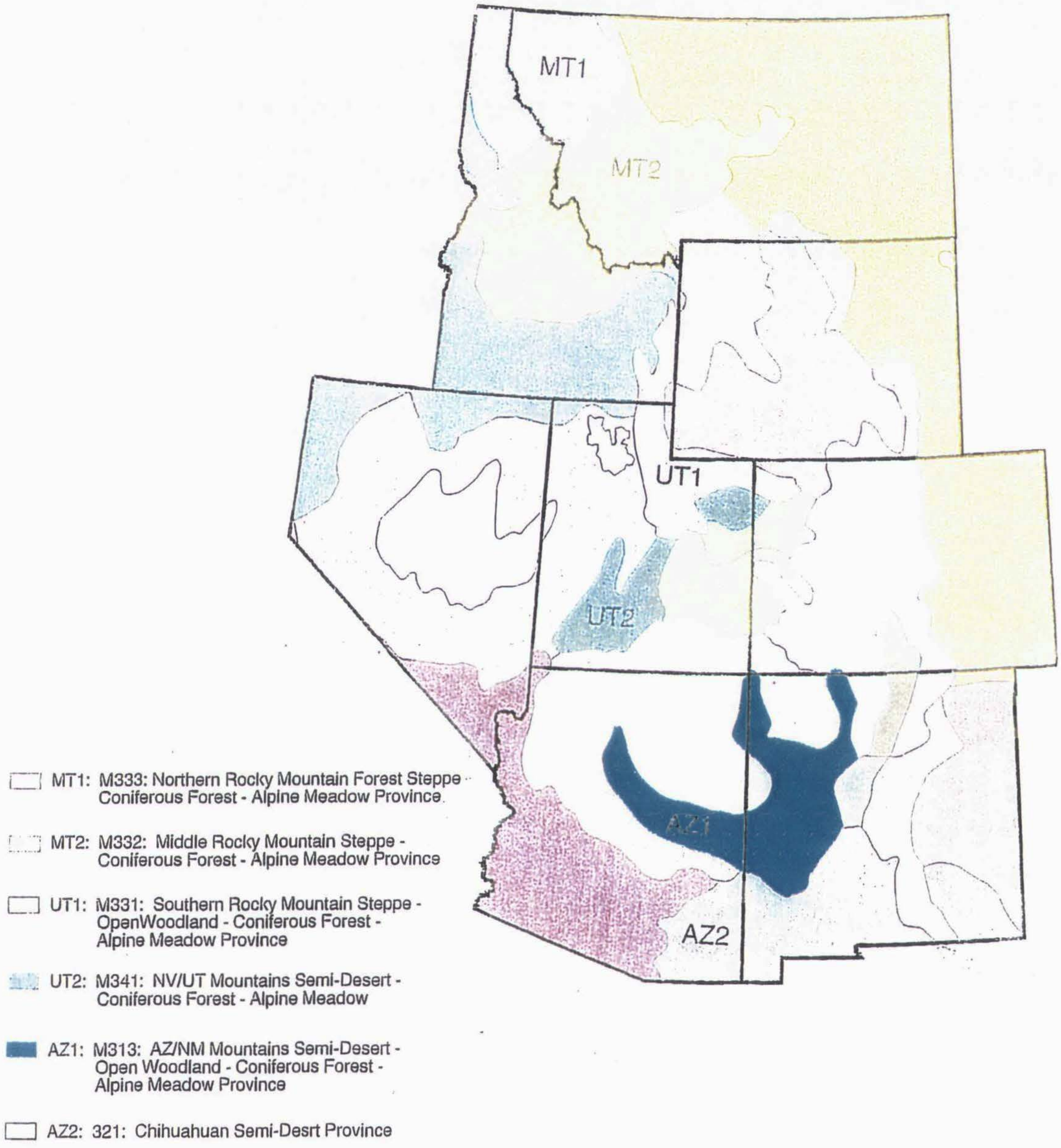

MT1
MT2
UT1
UT2
AZ1
AZ2
MT1: M333: Northern Rocky Mountain Forest Steppe - Coniferous Forest - Alpine Meadow Province
MT2: M332: Middle Rocky Mountain Steppe - Coniferous Forest - Alpine Meadow Province
UT1: M331: Southern Rocky Mountain Steppe - OpenWoodland - Coniferous Forest - Alpine Meadow Province
UT2: M341: NV/UT Mountains Semi-Desert - Coniferous Forest - Alpine Meadow
AZ1: M313: AZ/NM Mountains Semi-Desert - Open Woodland - Coniferous Forest - Alpine Meadow Province
AZ2: 321: Chihuahuan Semi-Desrt Province

Figure 3-1. Six Study Ecoregions Within the Intermountain West.

Table 3-1. Description of Six Study Ecoregions, Sampling Intensity, and Plot Layout. [1]

Label	Description	Size (ha)	Inventory Dates	Grid Intensity	Plot Layout	# Plots F	# Plots (Tot)
MT1	Northern Rocky Mountain Forest Steppe - Coniferous Forest - Alpine Meadow Province	4.43 M	NF: 1993-1996 Other: 1988-1989	All: 5 k	Other Tmbr:10pt NF Tmbr: 5/7 Wdld: old fixed	1393	1677
MT2	Middle Rocky Mountain Steppe - Coniferous Forest - Alpine Meadow Province	9.45 M	NF: 1996-1998 Other: 1988-1989	All: 5 k	nonNF Tmbr:10pt Other Tmbr: 5/7 Wdld: old fixed	1634	3727
UT1	Southern Rocky Mountain Steppe - Open Woodland - Coniferous Forest - Alpine Meadow Province	3.18 M	All: 1992-1996	NF: double 5 k Other: 5 k	Tmbr: 5 pt Wdld: old fixed	531	968
UT2	NV/UT Mountains Semi-Desert - Coniferous Forest - Alpine Meadow Province	3.16 M	All: 1993-1996	NF: double 5 k Other: 5 k	Tmbr: 5 pt Wdld: old fixed	829	1320
AZ1	AZ/NM Mountains Semi-Desrt - Open Woodland - Coniferous Forest - Alpine Meadow Province	2.85 M	NF,res, Tmbr: 1996-1997 Other: 1983	NF, res, Some IR: 5 k Timber/ Other: double 10 k	All: New fixed	664	1141
AZ2	Chihuahuan Semi- Desert Province	2.85 M	NF,res, Tmbr: 1996-1997 Other: 1983	NF, res, Some IR: 5 k Timber/ Other: double 10 k	All: New fixed	165	1129

[1] *NF*=National Forest; *Other*=lands outside NF; *res*=reserved lands; *Tmbr*=Timberland; *IR*=Indian reservations; *Wdld*=Woodland; *F*=Forested; *Tot*=Forested and Non-forested plots combined.

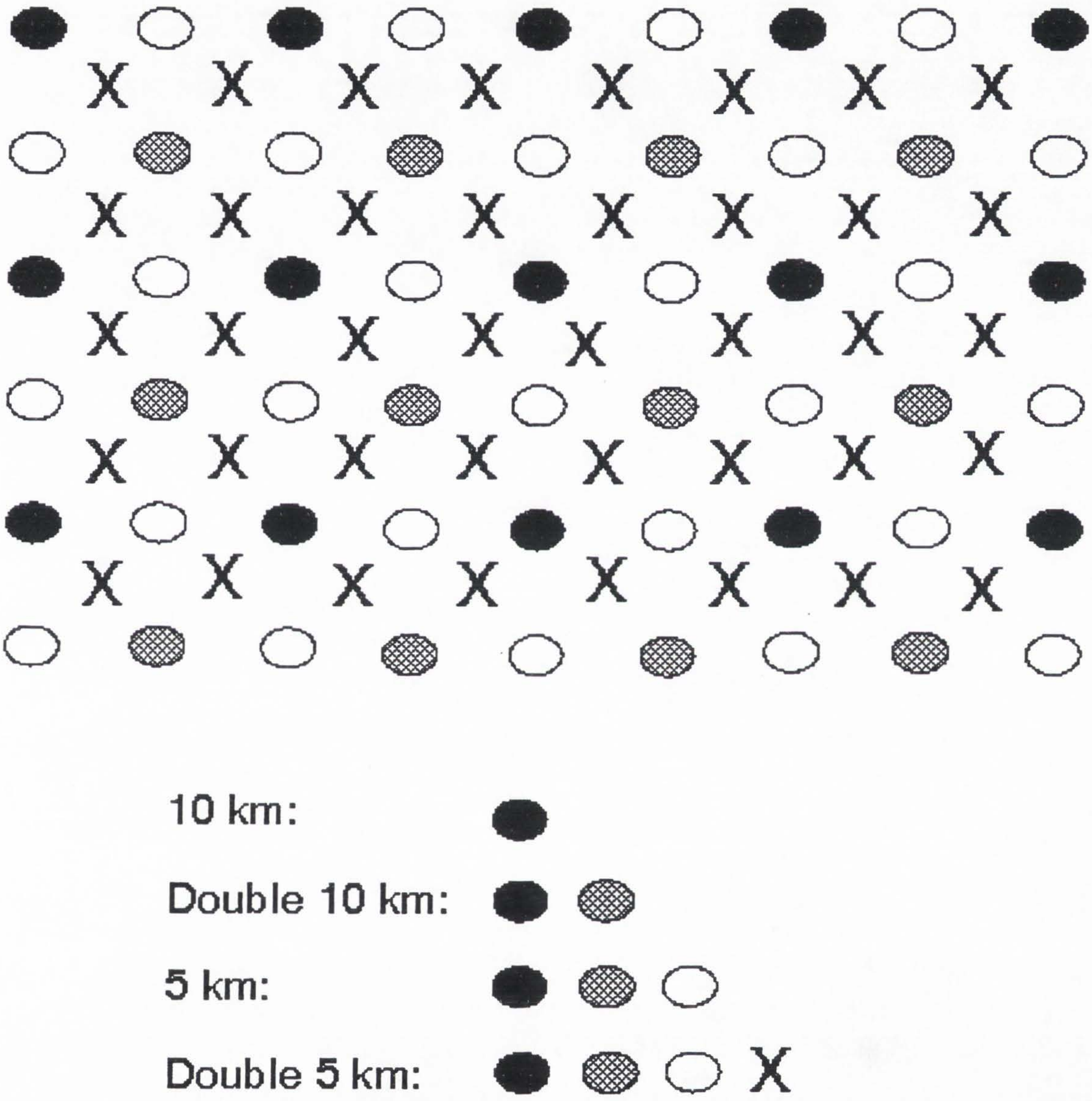


Figure 3-2. Schematic of Different Sampling Grid Intensities.

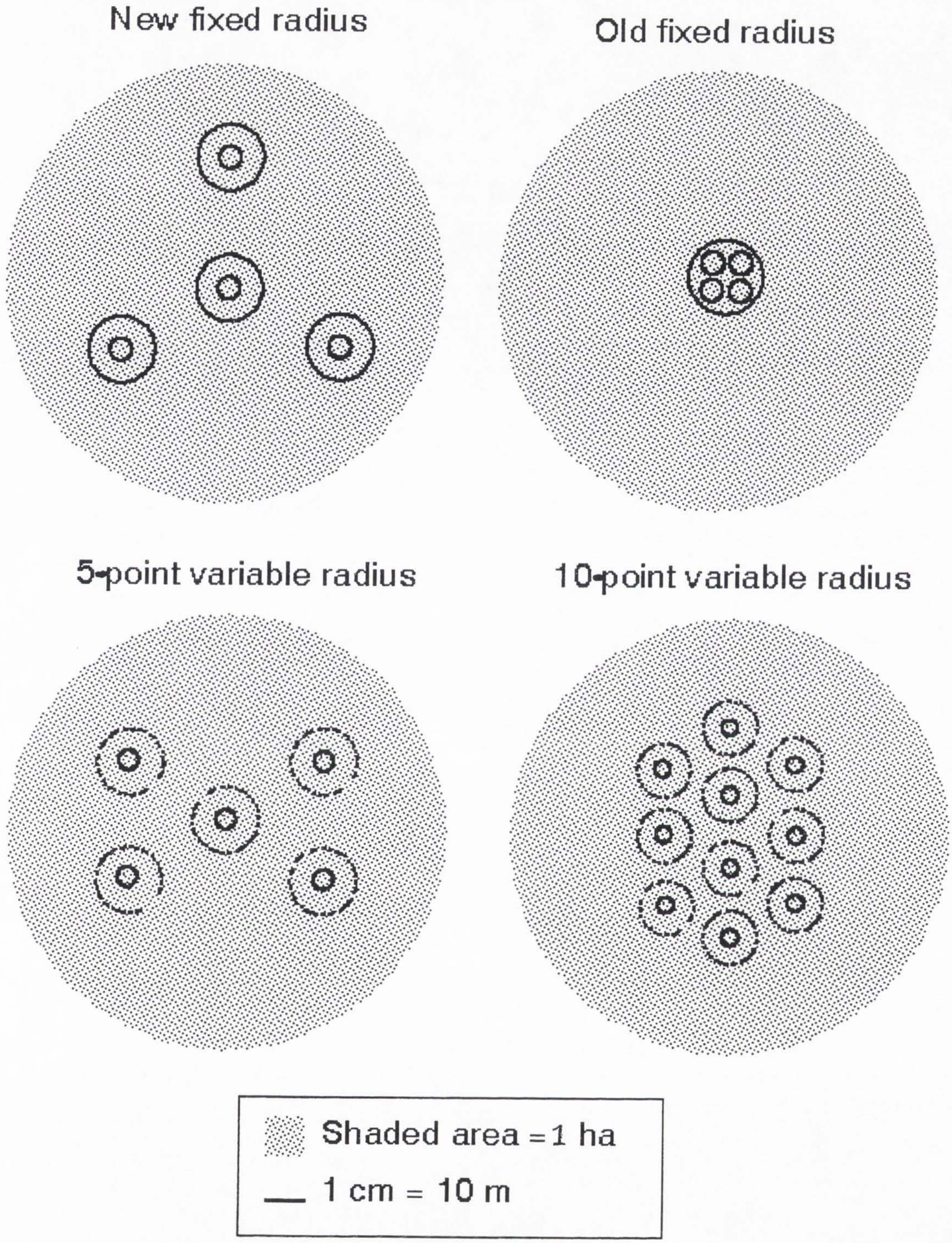


Figure 3-3. Four Different Plot Layouts.

within timberland strata, with a double 10 km grid visited elsewhere. Plot layout and sample design differences are more complicated than what Table 3-1 indicates, but standardized per-acre responses were retrieved under each layout.

Response Variables

At each FIA field location, extensive stand- and tree-level measurements were collected. Individual tree measurements were compiled and combined with stand-level variables to produce location-level summaries that comprise phase two of this two-phase design. Commonly used estimates of population totals include area by forest type, total tree volume, and total annual tree volume growth. Other variables of particular interest to forest planners and ecologists include forest type, biomass, crown cover, tree size, and stand age. Response variables and variables used to produce estimates of population means/totals are shown in Tables 3-2 and 3-3, respectively.

Predictor Variables

Predictor variables were extracted from four sources: (1) elevation, aspect, and slope from 1000-m digital elevation models produced by the Defense Mapping Agency (DMA); (2) spectral and positional data from a biweekly AVHRR composite; (3) vegetation cover type from the National Land Cover Data (NLCD) based on a 30-m resolution TM imagery; and (4) site-specific data including field-collected topographic information as well as UTM coordinates. Daily observations from the AVHRR platform are compiled biweekly to produce maximum normalized difference vegetation index

Table 3-2. Response Variables.

Type	Name	Description
Discrete response on all plots	FORTYP.2	0 = Non-forest 1 = Forest
Discrete response on forested plots only	FORTYP.3	1 = Woodland (Other forest in MT) 2 = Timberland (Spruce-fir in MT)
Continuous response on forested plots only	BIOTOT	Total tree biomass (lbs/acre)
Continuous response on forested plots only	CRCOV	Tree crown cover (%)
Continuous response on forested plots only	QMDALL	Quadratic mean tree diameter (in)
Continuous response on forested plots only	STAGE	Average age of dominant trees (yrs)

Table 3-3. Variables Used for Population Estimates.

Type	Name	Description
Discrete response on all plots	FORTYP.2	0 = Non-forest 1 = Forest
Discrete response on all plots	TWN	0 = Non-forest 1 = Woodland (Other forest in MT) 2 = Timberland (Spruce-fir in MT)
Continuous response on all plots	NVOLTOT	Live tree volume (cuft/ac) 0 on non-forest plots
Continuous response on all plots	NGRWCF	Net growth (cuft/yr) 0 on non-forest plots

(NDVI) composites of the U.S. These composites result in a near cloud-free image depicting maximum vegetation greenness for the compositing period. One such composite dated (June 1986) was used in these analyses and contains six bands of "least cloud" information including five spectral channels [one visible, one near infrared (NIR), and 3 infrared (IR)] as well as a Normalized Difference Vegetation Index (NDVI) that is computed NDVI=(NIR-IR)/(NIR+IR). These composites are distributed by USGS EROS Data Center.

The NLCD (http://edcwww.cr.usgs.gov/programs/lccp) is a land cover data set produced through a cooperative effort involving the U.S. Environmental Protection Agency, U.S. Geological Survey, U.S. Forest Service, and National Oceanic and Atmospheric Administration. This Thematic Mapper (TM)-based national data set (released in 2000) provides 21 mapped cover-types at 30-m resolution. In this study, cover-types were collapsed to a simple forest, shrubland, and non-forest type. A list of predictor variables and their descriptions is provided in Table 3-4.

Data Processing

Acquiring and processing data for modeling and analysis involved a considerable amount of work. Acquiring imagery and other auxiliary data sets involved cooperation with numerous government and private entities. Retrieval of response variables from the large and complex FIA database and extracting predictor variables from images required programming in SAS, Oracle, ArcInfo, ArcGrid, ArcView, and Imagine. Nine steps are described below, but the processes are changing rapidly to make them more efficient and

Table 3-4. Predictor Variables.

Type	Name	Description
Discrete predictor	NLCD	0 = Non-forest 40 = Forest 50 = Shrubland with trees
Continuous predictor	EASTING	UTM Easting – Zone 12
Continuous predictor	NORTHING	UTM Northing – Zone 12
Continuous predictor	ELEV.1K	Elevation (m) from 1km DMA
Continuous predictor	TRASP.1K	Radiation index derived by transforming aspect from 1km DMA
Continuous predictor	SLOPE.1K	Slope (%) from 1km DMA
Continuous predictor	AVH.1	Visible spectral band 1 from AVHRR composites
Continuous predictor	AVH.2	Near-IR spectral band 2 from AVHRR composites
Continuous predictor	AVH.3	IR spectral band 3 from AVHRR composites
Continuous predictor	AVH.4	IR spectral band 4 from AVHRR composites
Continuous predictor	AVH.5	IR spectral band 5 from AVHRR composites
Continuous predictor	NDVI	NDVI from AVHRR composites

well suited to the FIA production environment. The point in describing them here is to document the complexity in generating “simple” ASCII files of response and predictor variables for modeling and analysis. Most data retrieval and pre-model processing were performed on an IBM RS6000 F50 server with two processors and two gigabytes of memory. Some functions were also performed on a Pentium II PC with 64 megabytes of memory.

Choose Projection

A projection system is the mechanism for locating points, lines, or polygons on the earth using x and y coordinates. There are many projections to choose from, but the FIA sampling frame is based on the Universal Transverse Mercator, or UTM, grid. The earth is divided into a set of grid zones, within which UTM coordinates are unique, but between which they are not. As one moves away from the equator, the distance between zones is squeezed down in the east-west direction. Points on an equidistant grid in UTM units are actually much closer together when one gets closer to the poles of the earth. For modeling on a regional scale, this projection works well, but is unworkable on a National or global scale. Consequently, images or other geographic data sets come in a wide variety of projections and must be standardized to one projection in order to merge the spatial data together. Projection is a simple but sometimes computationally intensive process in ArcInfo. Commands for doing this and many other procedures in ArcInfo are given in Appendix A-1. Despite its discontinuity between zones and east-west “squeeze,” the UTM projection results in “prettier” (less distorted) regional maps, and is the these analyses. All the ecoregions fell within UTM zone 12 (UTM-12) except MT1 and MT2 that span 2 and 3 zones, respectively.

Locate Ecoregion and State Boundary Coverages

Ecoregion boundaries developed by Bailey, Avers, King, and McNab (1994) were available as vector coverages, and were modified in Utah to more closely follow ecological zones. State boundary coverages were readily available from the US Census Bureau, having been digitized from 1:100,000 scale maps. Both ecoregion and state boundaries were projected in UTM-12 and used throughout these data extraction processes.

Identify Sample Plots Within Ecoregions and Generate Point Coverages

UTM Easting and Northing coordinates on sample plots within an ecoregion were extracted from Oracle using an SQL query. A point coverage was then generated in ARC from these coordinates. Both the resulting ASCII files and point coverages were then used to extract response and predictor variables from these analyses.

Generate Intensive Grid for Mapping and Stratification

In addition to the ASCII file of UTM coordinates of field locations, an ASCII file of UTM coordinates on a 1-km grid was first generated from existing 1-km photo-interpretation data sets. Alternatively, these files can be generated through the ArcGrid *sample* command applied to any 1-km grid (like AVHRR data). In addition, finer resolution grids can be generated from existing 1-km grids through the ArcGrid *resample* command. The UTM coordinates on this much more intensive grid (like 90 m) can be written to an ASCII file again through the ArcGrid command *sample* (Appendix A-1). These very intensive grids are useful for predicting a response over fine resolution predictor variables. The maps are not necessarily more accurate, but "prettier."

Extract Response Variables from FIA Databases

Several different methods were used to get response variables for these analyses. In AZ1 and AZ2, the ASCII files containing UTM coordinates for sample plots within the ecoregion of interest were merged with SAS data tables of compiled Arizona data to extract select variables from the database. An example of this approach is given in Appendix A-2. In Utah and Montana, plot-level data were first extracted using an ArcInfo Arc Macro Language (AML) (Appendix A-3). The AML then generated a point coverage of these plots with selected location-level variables (like forest type) as attributes. The AML was not able to summarize tree-level information (like biomass and volume) and was replaced by a streamlined ArcView project with multiple Avenue scripts Appendix A-4). The project generates point coverages from spatial coordinates in Oracle and adds any selected variables to those points as attributes, simultaneously delivering an ASCII file (Appendix A-4).

Clip Statewide Grids from National AVHRR Images

Procedures for loading and viewing data from the 2-week composite AVHRR CDs in Imagine are given in Appendix A-5. Complications arose over incompatibility of Imagine and ArcInfo boundary coverages but following these instructions alleviated the problem. Once these data were loaded, an "area of interest" (AOI) layer (like a state or ecoregion) was created and used to subset the multi-band AVHRR image. Because this nationwide image came in a Lambert Azimuthal equal area projection, the state boundaries had to be reprojected to Lambert Azimuthal before subsetting. ("Rebuilding" polygon coverages is also necessary following projection.) Subsetted images for each band and NDVI values were then converted to GRID format and reprojected to UTM-12 for further processing in ArcInfo. This process of subsetting an image is documented in Appendix A-6 and resulted in six grids representing the five AVHRR spectral bands and NDVI for each state.

Clip Statewide Grids from National 1000 m DMA Files

This nationwide elevation grid was also projected in Lambert Azimuthal, so state boundary coverages in that same projection were used. Arizona, Montana, and Utah were clipped from the nationwide grid using Arc *latticeclip* and reprojected into UTM-12. Slope and aspect grids were created for each state using the DEM (elevation) through ArcGrid's *slope* and *aspect* commands. (Again, see Appendix A-1 for miscellaneous commands.)

Clip Statewide Grids from Regional NLCD Grids

The NLCD data was distributed as images in regional blocks. These images were converted to grids using *imagegrid* and came in an Albers NAD83 projection. Zero values in each regional grid had to be converted to NoData or Null values prior to merging regional grids using the command *merge*. After merging the grids to areas that would encompass each state, state boundaries in Albers NAD83 clipped the state using ArcGrid *gridclip*. The resulting grids were finally converted to UTM-12 projection.

Extract Predictor Variables at FIA Plot Locations

The ArcGrid *sample* command was used to extract predictor variables from the assortment of grids at field and intensified grid locations, specified in ascii files of UTM coordinates. An example of commands used to extract predictor variables in Montana is given in Appendix A-7.

CHAPTER 4

SYSTEM DEVELOPMENT

The following chapter describes the development of an objective mapping and stratification system (or "box") within the S-Plus environment. An objective (or production) environment implies that someone without modeling experience can push the button that builds the models and produces desired output for any ecoregion, response variable, or predictor set. Prior to the construction of this box, input data had to be collected and prepared, and ancillary programs obtained and installed. A schematic overview of the system including data collection, ancillary code, and the S-Plus modeling box is shown in Figure 4-1. The initial data collection process (described in Chapter 3) was conducted in a variety of computing environments including ArcInfo, ArcView, Imagine, Oracle, and SAS. Data were extracted from original formats, filtered and put in a standard flat file format. Ancillary programs were imported from a variety of sources documented below, and installed in the S-Plus environment. The box itself was developed in S-Plus and was run on a SUNW Ultra-1 Sparc workstation with 128 megabytes of RAM and 602 megabytes of swap. The modeling system contains five key programs that are described in detail in this chapter. The first program, **p1.data** (Appendix C-1), prepares the input data. Objective model building and evaluation using the five different techniques takes place in **p2.model** (Appendix C-2) while **p3.map** (Appendix C-3) produces predictions for import into ArcView for mapping. Stratification based on select predicted maps is applied and population estimates and variances produced in **p4.strat** (Appendix C-4). Finally, **p5.results** (Appendix C-5) compiles

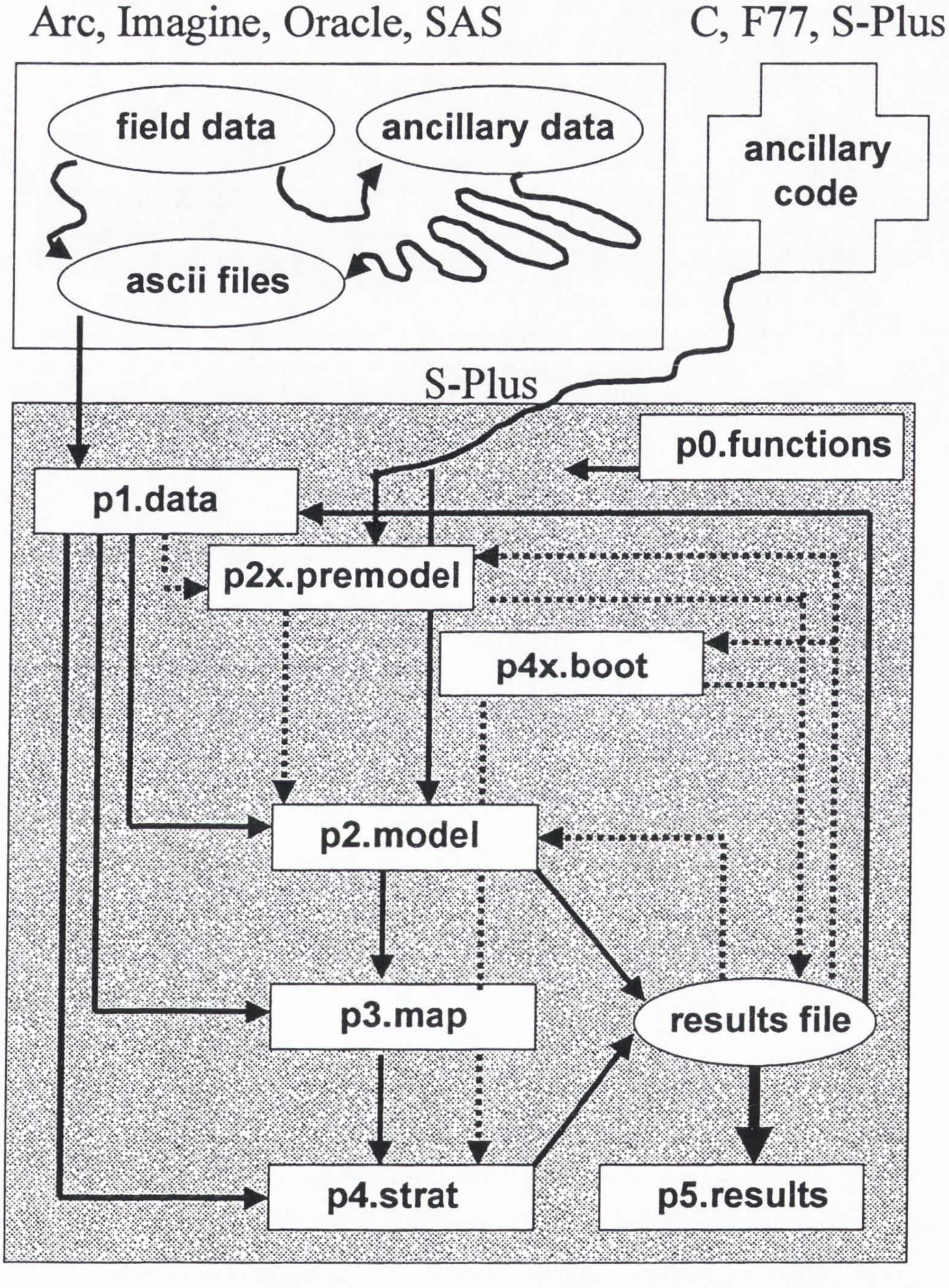


Figure 4-1. Schematic Overview of Modeling System.

performance measures and prepares graphical synopses of the results by ecoregion, response variable and predictor set. Program **p2x.model** supported the development of **p2.model** and **p4x.boot** (Appendix C-4x) provided additional information for variance calculations in **p4.strat**. In addition, **p0.functions** (Appendix C-0) is a file of customized functions, and the box itself runs from a program called **p.go** (Appendix C-00).

Data Input

This first program, **p1.data**, prepares the data for input into subsequent programs. Response and predictor variables from the designated ecoregion are first read from asciii files and put in standard format. Predictor variables from imagery are checked for missing values and merged with the appropriate field file, or saved as separate mapping files on a more intensive grid.

Transformations on predictor variables are kept to a minimum. Only the NLCD and circular aspect variables are modified. The NLCD classes are collapsed to one forest, one shrubland, and one non-forest class to avoid having only a handful of observations in the rare classes. The circular aspect variable is transformed to a radiation index (TRASP) used by Roberts and Cooper (1989). This takes the form

$$\text{TRASP} = \frac{1-\cos((\pi/180)(\text{aspect}-30))}{2}.$$

This transformation assigns a value of zero to land oriented in a north-northeast direction, (typically the coolest and wettest orientation), and a value of one on the hotter, drier south-southwesterly slopes.

The two discrete response variables, FORTYP.2 and FORTYP.3, are created by collapsing forest type (FORTYP) into forest/non-forest (FORTYP.2) and timberland/woodland within forested areas (FORTYP.3), respectively. Data files for modeling the discrete FORTYP.2 include all data from forest and non-forest locations while data for modeling FORTYP.3 includes only data from forested field locations. This is analogous to applying a forested "mask" over the data set to focus modeling on within-forest conditions. One of the last tasks of **p1.data** is to split the total and masked data files into 70% of the data for modeling and 30% for testing. This 30% was chosen because this is the approximate proportion of plots collected on an intensified (not the standard 5 km) sampling grid and withholding this additional amount gives an indication of predictive abilities given "standard" sample sizes. Response and predictor variables used in **p2.model** were given in Tables 3-2 and 3-4, respectively. Files used in **p3.map** for predicting response variables over an intensified grid contain all the predictor variables listed in Table 3-4 as well. Variables used to construct estimates of population means/totals in **p4.strat** are listed in Table 3-3.

An early concern involved the potential effect of spatial autocorrelation on the deterministic functions chosen for these analyses. As part of the preliminary modeling described in Moisen and Edwards (1999), directional variograms were constructed on a set of variables revealing large-scale spatial patterns driven largely by elevation. By fitting GLMs that included elevation, aspect, slope, and general geographic position as predictor variables, the data were "detrended" and nothing but noise was left in directional variograms of the residuals. An important point here is that responses are collected on a 5-km grid. In the interior west, field plots collected at this distance are likely to change drastically in elevation, slope, and aspect, and so the primary concern is modeling large-scale spatial variation through a deterministic model. The purpose in including UTM coordinates as predictor variables is to capture broad trend as opposed to trying to smooth the response in a small geographic area.

Modeling Strategies

Objective model building using the five different techniques takes place in **p2.model**. The following sections describe the development strategy for each of the five techniques. Models were first built, refined and automated using data in UT2 and **p2x.model**. These automated models were then applied to data from all ecoregions to test model performance, and results are reported in Chapter 5. Initial model fitting strategies were developed based on literature review, correspondence and discussions with a variety of experts, S-NEWS discussions, and trials in UT2.

NLCD Benchmark Models

By far, the simplest mapping and stratification strategy that could be adopted in these analyses is to simply "map" discrete variables by collapsing NLCD cover types, and "map" continuous variables by assigning the mean of the continuous variable within each NLCD class. This approach is implemented in **p2.model** though a function that collapses cover type classes, and through the use of the s-Plus function lm() for continuous variables. This is the simple benchmark against which other models are judged.

GAMs

The S-Plus functions gam(), step.gam(), and plot.gam() are used here. Both the binary forest/non-forest (FORTYP.2) and timberland/woodland (or spruce-fir/other) within forest (FORTYP.3) classifications are modeled using a binomial family.

Selection of an appropriate link function and variance-to-mean relationship for continuous variables, however, can be difficult. In Moisen and Edwards (1999), exploratory work revealed that the variances appeared proportional to the means (after adjustment for predictor variables), with proportionality constants substantially larger than one. The variance of volume by species (within bins defined by combinations of predictor variables) was plotted against mean volumes of those bins revealing linearly increasing patterns. Consequently, in earlier work, quasi-likelihood estimation was used in a “Poisson-like” model with a log link and variance proportional to the mean. Although this type of model is typically applied to count data, McCullagh and Nelder (1989, pp. 200-204) discuss an example application to continuous data.

This same approach was first adopted in **p2x.model**. However, one problem encountered in Moisen and Edwards (1999) was the large number of zeros (on non-forest lands) and this likely dominated the mean/variance relationship. In **p1.data**, a non-forest mask was applied as described above in the data input section, and only continuous variables on forested plots were modeled, assuming the mask could be reapplied at time of mapping to black out non-forest areas. The variance of continuous variables on forested plots (within bins defined by combinations of predictor variables) was plotted against mean values of those bins revealing no detectable patterns. Consequently, in **p2.model** a simple Gaussian family is specified for continuous responses, but an option can be implemented within the program to run a regression of variances on means and make a choice of family automatically by determining if the variance is proportional to 1, μ , μ^2, or μ^3 and then assigning a Gaussian, Poisson, gamma, or inverse Gaussian family, respectively.

For both continuous and discrete responses, predictor variables enter the model individually using a smoothing spline with a relatively large smoothing parameter to avoid fitting noise. Final models were selected by stepwise procedure invoked by *step.gam*. The function steps through various combinations of models along a path that is specified using an argument called *scope* in the *step.gam* function. This argument is a list whose elements correspond to terms in the original model. The step.gam interprets candidate forms for each model term based on their order of appearance in the scope function. For example, suppose the initial gam object looks like

```
Y ~ s(ELEV.1K) + s(TRASP.1K) + s(SLOPE.1K).
```

Then, a possible scope function is

```
$ELEV.1K:
 ~ 1 + ELEV.1K + s(ELEV.1K)

$TRASP.1K:
 ~ 1 + TRASP.1K + s(TRASP.1K)

$SLOPE.1K:
 ~ 1 + SLOPE.1K + s(SLOPE.1K).
```

Starting with the current model, a series of models is then constructed by moving each term up or down one step in the scope function. The first few candidate models would be

```
Start: Y ~ s(ELEV.1K) + s(TRASP.1K) + s(SLOPE.1K); AIC= 1197.427
Trial: Y ~  ELEV.1K + s(TRASP.1K) + s(SLOPE.1K); AIC= 1197.444
Trial: Y ~  s(ELEV.1K) + TRASP.1K + s(SLOPE.1K); AIC= 1192.541
Trial: Y ~  s(ELEV.1K) + s(TRASP.1K) + SLOPE.1K; AIC= 1207.651
Step :  Y ~ s(ELEV.1K) + TRASP.1K + s(SLOPE.1K) ; AIC= 1192.541 .
```

The model that results in the biggest decrease in AIC, where

$$AIC = D + 2df\phi$$

is selected as the new current model and the updating continues. Here, *D* is the residual deviance, *df* the effective degrees of freedom, and ϕ the dispersion parameter.

CARTs

The S-PLUS functions tree(), cv.tree(), prune.tree(), and ps.tree() are used for both classification models (classification trees) and for modeling continuous variables (regression trees). An initial tree is fit using all the predictor variables. Tree pruning, analogous to variable selection in regression, is the methodology used to prevent overfitting the training data with too many splits. Although many methods of pruning are available, pruning through cross-validation is most popular. By using cv.tree(), the optimal size is identified via 10-fold cross-validation. While this process was repeatable for classification in UT2, the "optimal size" was very different under different cross-validation runs for continuous variables. Consequently in **p2.model**, 20 cross-validatory splits are run and "majority rule," ie optimal size getting the most votes, used to determine pruning size for continuous variables.

MARS

Not part of the packaged S-Plus functions, Hastie and Tibshirani's mars() function is available through StatLib in the "mda" library. Because the library of functions was constructed for use in S-Plus-3 versions, getting the library up and running involved developing a new installation procedure to update old S-Plus version libraries and helpfiles, and replace S-Plus function calls to *dyn.load* within the scripts. Appendix B-1 gives the modified installation procedures.

As mentioned earlier, MARS automatically selects the amount of smoothing required for each predictor as well as the interaction order of the predictors. It is considered a projection method where variable selection is not a concern but the maximum level of interaction needs to be determined. Preliminary runs in UT2 for all response variables and levels of interaction ranging from 1 to 5 showed little improvement in fit and a tendency to produce unrealistic predictions for higher orders of interaction. Taking a conservative approach, only 2-level interactions are specified in **p2.model**. Because the MARS function as imported from StatLib did little more than fit the model and produce predictions, **p2.model** provides supplemental code that displays the contributing variables and identifies interactions so that the models are more interpretable.

ANNs

Nychka's FUNFITS S-PLUS function library was obtained by ftp for fitting ANN's from http://www.cgd.ucar.edu/stats/Funfits/index.shtml. As with the MARS library, getting FUNFITS up and running posed some challenges and required modifications to the installation procedure (Appendix B-2).

The neural networks sum of squares surface is probably best described as "pathological," having a very large number of local minima that easily trap unsophisticated search procedures and result in poor, non-reproducible models. The developers of FUNFITs took care in developing their search procedure for realistic starting values. This procedure proceeds as follows. For each of M hidden units in a single layer neural network, a rectangular region of feasible (with respect to the logistic function) parameter values is divided into a set of 250 (ngrind) boxes about the origin. Within each of these boxes, 100 (ntry) parameter sets are randomly generated from a uniform distribution. These parameters define initial models from which RMSEs are computed. The parameter set with the lowest RMSE of the 100 sets in each of the 250 boxes is identified, and the 250 parameter sets are used as starting values in an initial 250 "grinds" attempting to minimize the RMSE. The convergence tolerance for this first pass is set fairly high. Next, the parameters resulting from the best 20 (npolish) grinds based on RMSE are used as starting values in a second more refined set of minimizations where the convergence tolerance is set much lower. The "final" model is defined by the parameters resulting from the best of these 20 "polished" parameter sets.

Although the computing time can be quite slow for full search options, the often subjective choices about starting values, convergence criteria, and number of hidden units are done automatically, and the results are reproducible. However, running **nnreg** at the default ngrind, ntry, and npolish of 250, 100, and 20, respectively, resulted in painfully slow computing time. FUNFITS provides a "fast" option where ngrind, ntry, and npolish assume values of 100, 50, and 5. Initial runs in UT2 for all response variables for both the default and fast options revealed little if any gain in performance measures but tremendous differences in computing time. FUNFITS also selects the optimum number of hidden units based on cross-validation.

Evaluation Criteria

Several global measures were used to assess the predictive performance of the models. Let **x** be an r x r contingency table or error matrix set out in rows and columns that express the number of sample plots (of which there are n) predicted to belong to one of r classes relative to the true ground class (on the diagonal). The percent of correctly classified (PCC) plots is calculated

$$PCC = \left(\frac{1}{n} \sum_{i=1}^{r} x_{ii} \right) \times 100\%$$

The Kappa (KHAT) statistic (Cohen 1960) measures the proportion of correctly classified units after the probability of chance agreement has been removed, and has been used extensively in map accuracy work (Congalton 1991), and is calculated

$$\text{KHAT} = (\theta_1 - \theta_2)/(1 - \theta_2)\,,$$

where $$\theta_1 = \sum_{i=1}^{r} x_{ii} / n \text{ and}$$

$$\theta_2 = \sum_{i=1}^{r} x_{i\bullet} x_{\bullet i} / n^2 \,.$$

Predictive performance of models of the continuous variables were evaluated through independent estimates from test sets of global root mean squared error (RMSE),

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad ,$$

and proportion of plots within some user-specified range (PWI),

$$\text{PWI} = \frac{1}{n}\sum_{i=1}^{n} I\left\{\left|\hat{y}_i - y_i\right| < R\right\},$$

(e.g., proportion of plots predicted to within 50 cubic feet of the true volume). In addition, the correlation coefficient (ρ) between observed and predicted values

$$\rho = \frac{\sum y_i \hat{y}_i - \sum y_i \sum \hat{y}_i / n}{\sqrt{\sum y_i^2 - \left(\sum y_i\right)^2 / n}\ \sqrt{\sum \hat{y}_i^2 - \left(\sum \hat{y}_i\right)^2 / n}}$$

was calculated for each model.

In addition to the evaluation criteria above, the amount of time it took to run each model was recorded and considered in discussions about suitability of each of the models for a production environment.

Mapping

This program produces predictions for each response variable within ecoregion over an intensified grid of predictor variables. Predictions are exported to ascii files in format suitable for input into ArcView for display and analysis. The scale of the resulting maps is a function of the intensity at which predictor variables (as ArcInfo grids) are resampled. Here, a coarse 1-km grid was used for mapping to keep size and prediction times in check. A more visually appealing 90-m grid will be resampled for production of "pretty" maps following completion of these analyses.

When mapping over large geographic areas, one is guaranteed to run into values of predictor variables outside the range seen in the modeling dataset and extrapolation is unavoidable. In addition, high dimensional models with interaction confound the extrapolation problem and it is likely that nonlinear and nonparametric models produce unrealistic estimates. To prevent these few extreme values from completely overpowering evaluation criteria and map color schemes, model predictions were restricted from going below zero or above the maximum value seen in the model data set.

Stratification and Variance Reduction

For reasons discussed in Chapter 6, predicted FORTYP.2 from the NLCD and MARS models in each ecoregion were used as the basis for stratification of field plots in that ecoregion. Population estimates and variances were obtained in two different ways. The first uses a stratified random sampling (STR) formula. This is appropriate for the NLCD-based stratification but is a leap of faith for the MARS-based stratification because strata come from models driven by the very field data to be stratified, and the problem has the flavor of poststratification. Consequently, bootstrap variance estimates were constructed and compared to those obtained under STR. Further discussion of the bootstrap variance estimates follows the “Stratified Random Sampling” section.

Stratified Random Sampling

Following the notation of Cochran (1977), let the subscript h denote the strata, suffix i the unit within the stratum, and L the total number of stratum. An unbiased estimate of a population mean (Cochran, Equation 5.1) is

$$\overline{y_{st}} = \sum_{h=1}^{L} W_h \bar{y}_h ,$$

where

N_h is the total number of units in stratum *h,*

n is the number of sample units,

n_h is the number of sample units in stratum *h,*

y_{hi} is the value obtained for the *i*th secondary point in stratum *h,*

$W_h = \frac{N_h}{N}$ is the weight for stratum *h*,

$\bar{y}_h = \frac{1}{n_h} \sum_{i=1}^{n_h} y_{hi}$ is the sample mean for stratum *h*.

When the finite population correction factor is negligible, the estimated variance (Cochran, Equation 5.13) is

$$v(\overline{y_{st}}) = \sum_{h=1}^{L} \frac{W_h^2 s_h^2}{n_h},$$

where

$$s_h^2 = \frac{1}{n_h - 1} \sum_{i=1}^{n_h} (y_{hi} - \bar{y}_h)^2$$

is the sample variance for stratum *h* for continuous *y,* that reduces to

$$s_h^2 = \frac{n_h \bar{y}_h (1 - \bar{y}_h)}{(n_h - 1)}$$

when y_{hi}s assume values of 0 and 1.

Bootstrap Estimates

In the second approach, bootstrap estimates of the variances on select population estimates were generated using the MARS models for FORTYP.2 and FORTYP.3 in each ecoregion. A sample of size *n* was selected with replacement from the original modeling data. The MARS models were fit, predictions made over a 1-km grid using FORTYP.2, and further classification done using FORTYP.3 for points predicted to be forested by FORTYP.2. These two-step MARS predictions over the 1000-m grid formed the population of strata for determining stratum weights. Then, estimates of mean population area, volume, and growth for this bootstrap sample were calculated using stratum weights from above. Another sample of size *n* was selected with replacement and the process repeated 100 times. The variances of these 100 population estimates were then compared to those obtained using the STR formulation.

Results

The program **p5.results** compiles mapping and stratification performance measures and prepares graphical synopses of the results by ecoregion, response variable, and predictor set using trellis graphics functions within S-Plus.

CHAPTER 5

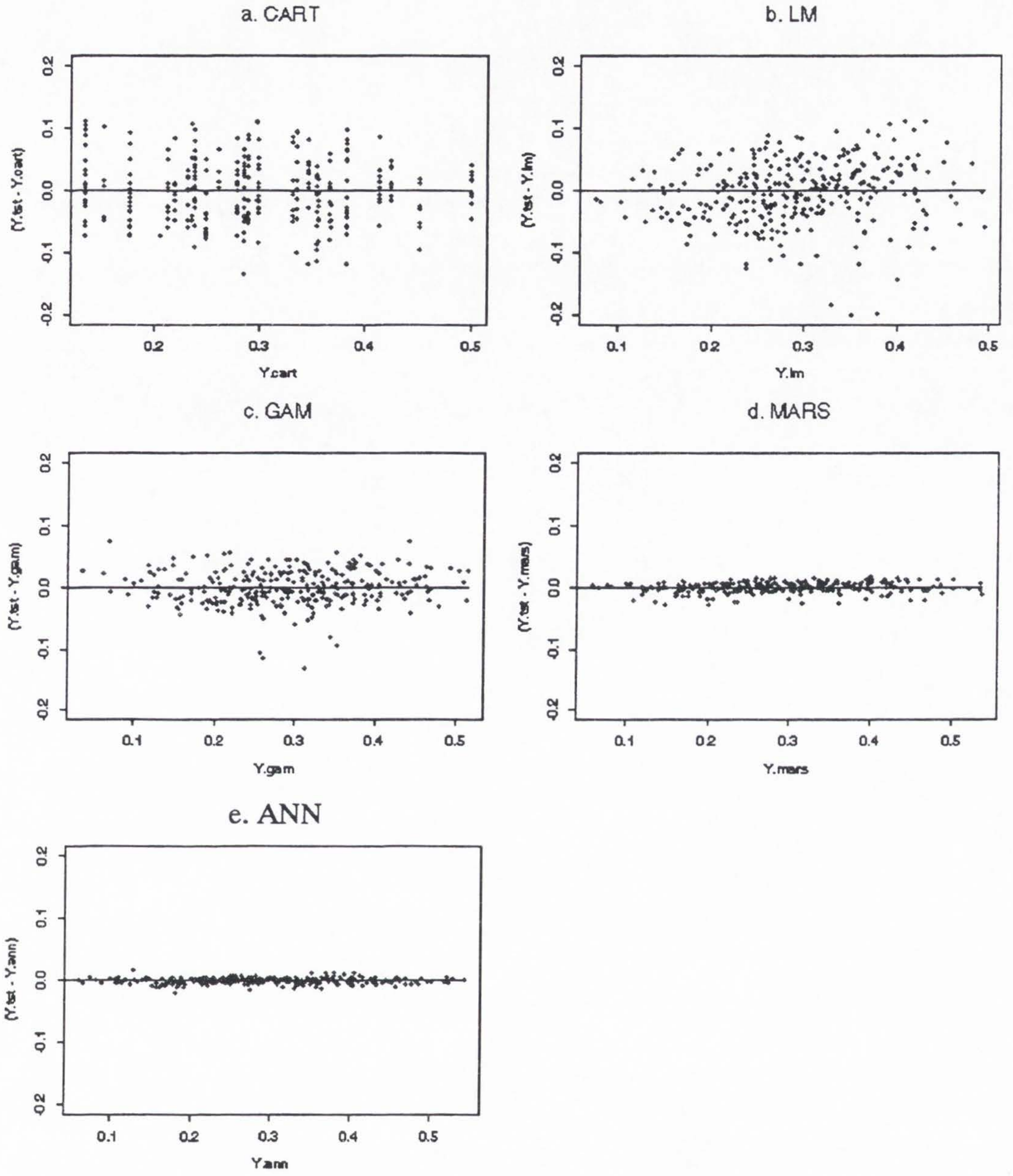
RESULTS

System Tests

Before running data from all the ecoregions through the modeling system, a simple test was conducted to insure that the modeling techniques were operational. Following DeVeaux et al. (1993), 1000 each of ten uniformly distributed predictor variables X1-X10 were generated. Next, a response Y was specified as a function of only X1-X5,

$$Y = 2\sin(\pi * X1 * X2) + .4(X3 - .5)^2 + .2(X4) + .1(X5),$$

with no error term. A simple linear model along with the GAM, CART, MARS and ANN from the modeling box were used to fit the relationship between Y and the X1-X10. Residual plots under each of the modeling techniques are shown in Figures 5-1 a-d. These plots, generated from test data, illustrate the effectiveness of MARS and ANNs in deciphering complex relationships. Table 5-1 also reveals some of the strengths and weaknesses of the different techniques. CART models identified the contributing predictor variable (X1-X5), but had an RMSE that was 10% higher than a linear model, and 10 times the RMSE of ANNs. LM too had a high RMSE because of its inability to detect the nonlinearity or interaction between terms. GAM residuals were considerably better, but the model's stepwise procedures incorrectly identified X8 and X10 as contributing predictor variables in addition to the correct ones. Both MARS and ANN did exceptionally well, and MARS correctly identified the contributing variables and order of interaction. Recall the performance measures for continuous variables included root mean


a. CART

b. LM

c. GAM

d. MARS

e. ANN

Figure 5-1. Residual Plots from Test Data.

Table 5-1. Modeling Results for Continuous Test Data.

Model	Contributing variables	RMSE	PWI (25%)	RHO	TIME
CART	X4, X1, X3, X5, X2	.030	76	.843	202
LM	All	.027	83	.873	1
GAM	s(X1), s(X2), s(X3), X4, X5, s(X9), s(X10)	.014	95	.966	201
MARS	X1*X2, X3, X4, X5	.004	100	.997	43
ANN	All with 7 hidden units selected	.001	100	1.000	336

squared error, correlation between truth and predicted, percent of predicted plots with 25% of the truth, and computational run time. Again, ANNs and MARS performed best overall but MARS had a much faster computing time.

Next, simulations were run to illustrate the effect of random noise on the performance of each modeling technique. Following from the example above, the response was generated as

$$Y = .4\sin(\pi * X1 * X2) + .8(X3 - .5)^2 + .2(X4) + .2(X5) - .05 + \varepsilon,$$

with error terms generated from a normal distribution with mean equal to zero, and standard deviations of .05, .5, and 1. Table 5-2 and Figure 5-2 illustrate the diminishing differences between performance measures with increasing noise in the system.

Next, simulations were designed to illustrate strengths and weaknesses of the modeling techniques. One thousand sets of three uniformly distributed predictor variables X1-X3 were generated. Response variables Y1-Y5 were specified as functions of X1-X3 given in the first column of Table 5-2 with no error term. As before, 300 of the 1000 simulated response and predictor variable combinations were withheld as a test set. A simple linear model along with a GAM, CART, MARS and ANN were used to fit the

Table 5-2. Effect of Adding Normally Distributed Error with Increasing Standard Deviations (in parentheses).

	RMSE			PWI			RHO		
MODEL	(.50)	(.50)	(1)	(.50)	(.50)	(1)	(.50)	(.50)	(1)
LM	0.199	0.358	0.579	0.183	0.103	0.087	0.836	0.650	0.421
GAM	0.148	0.328	0.577	0.240	0.130	0.077	0.913	0.717	0.439
CART	0.136	0.318	0.586	0.340	0.127	0.053	0.929	0.738	0.421
MARS	0.047	0.297	0.603	0.767	0.133	0.077	0.992	0.775	0.429
ANN	0.037	0.292	0.556	0.870	0.160	0.097	0.995	0.784	0.508

model along with a GAM, CART, MARS, and ANN were used to fit the relationship between Y!-Y5 and the X1-X3. Models were tested using independent test data and the resulting values for RHO, percent within .1, and RMSE are shown in Table 5-3. In addition, residual plots (again using test data) for all response variables and models are shown in Figure 5-3. Y1 illustrates how each of the models performs when the response is a simple linear function of predictor variables. All models predict quite well over the test set with the exception of CART that is known to have trouble approximating linear functions. Next, Y1 illustrates a response that assumes values of 0 or 1 based on a simple bivariate step function. Here, CART excels because of its ability to assign values above and below simple threshold values, while the other models did quite poorly. Y3 is an illustration of data generated from a gamma distribution, with predictor variables affecting the mean in a nonlinear fashion. Here, the LM fails, CART perform on slightly better, GAMs with a log link performed much better (as it should), as did MARS and ANN. Y4 allows for a 2-way interactions and both linear and nonlinear terms. The nonlinearity causes trouble with the LM, the linear term confuses CART, and the

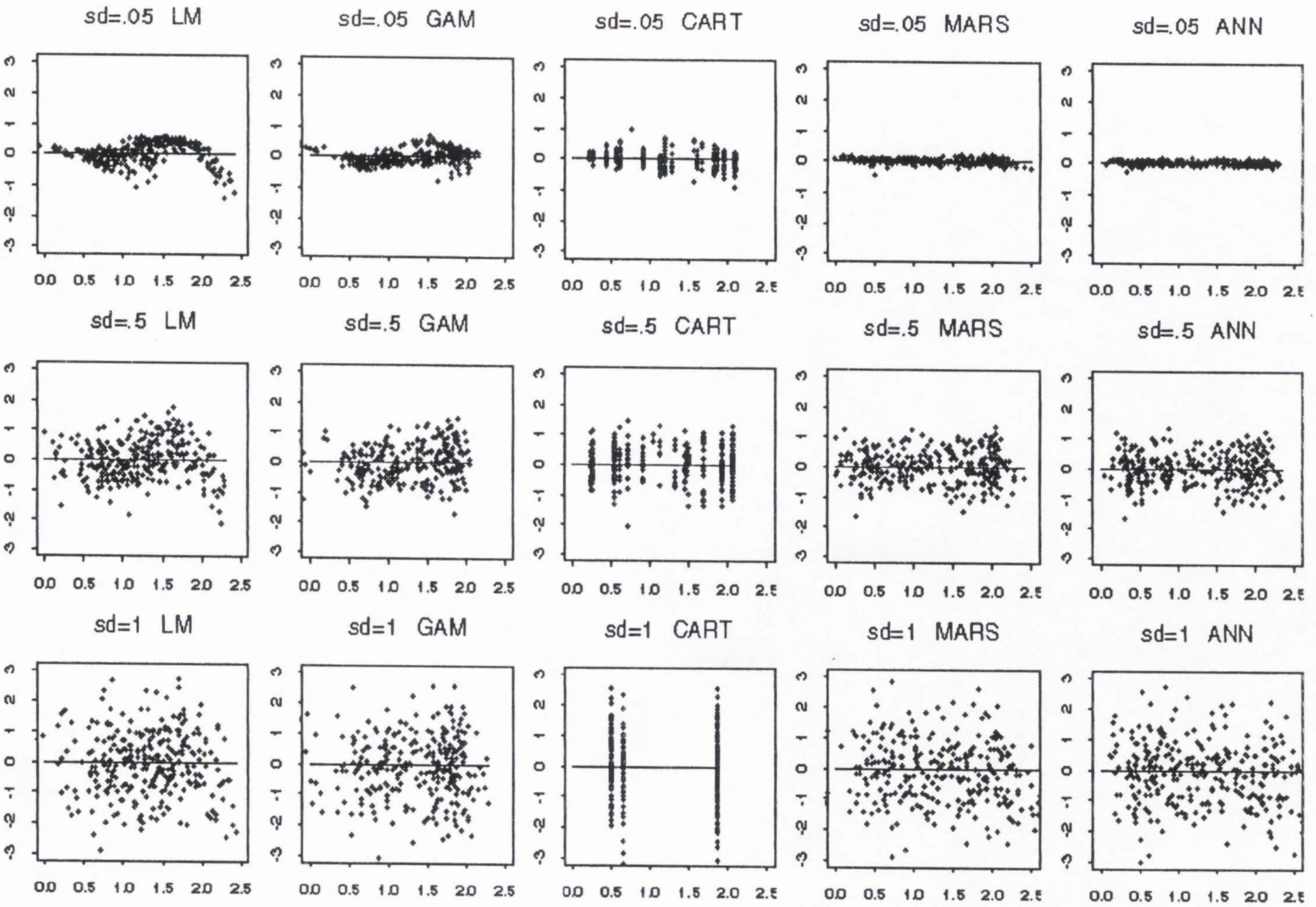


Figure 5-2. Effect of Adding Increasing Noise to System on Relative Model Performance.

Table 5-3. Modeling Results for Five Simulated Data Sets.

	Model		PWI	
True Formula	Technique	RHO	(.1)	RMSE
Y1=2X1 + 3X2	LM	1.00	100	0.00
	GAM	1.00	100	0.00
	CART	0.95	19	0.17
	MARS	1.00	100	0.00
	ANN	1.00	100	0.00
Y2=ceiling((X1-.5)(X2-.5))	LM	0.01	0	0.28
	GAM	0.05	0	0.27
	CART	1.00	100	0.00
	MARS	0.84	33	0.15
	ANN	0.89	48	0.12
$Y3 \sim \text{gamma}(10 + \sin(2\pi X1) + (20X2 - 10)^2)/100$	LM	0.11	21	0.16
	GAM	0.97	87	0.04
	CART	0.97	81	0.04
	MARS	0.97	84	0.04
	ANN	0.98	86	0.04
	LM	0.91	27	0.17
$Y3 \sim \text{gamma}(10 + \sin(2\pi X1) + (20X2 - 10)^2)/100$	GAM	0.96	31	0.11
	CART	0.94	29	0.14
	MARS	1.00	97	0.03
	ANN	1.00	100	0.01
$Y5 = 50\sigma((X3 - .5)\sigma(X1 - .5X2 - .13X3))$	LM	1.00	37	0.16
	GAM	0.99	36	0.17
	CART	0.98	18	0.21
	MARS	1.00	100	0.02
	ANN	1.00	100	0.00

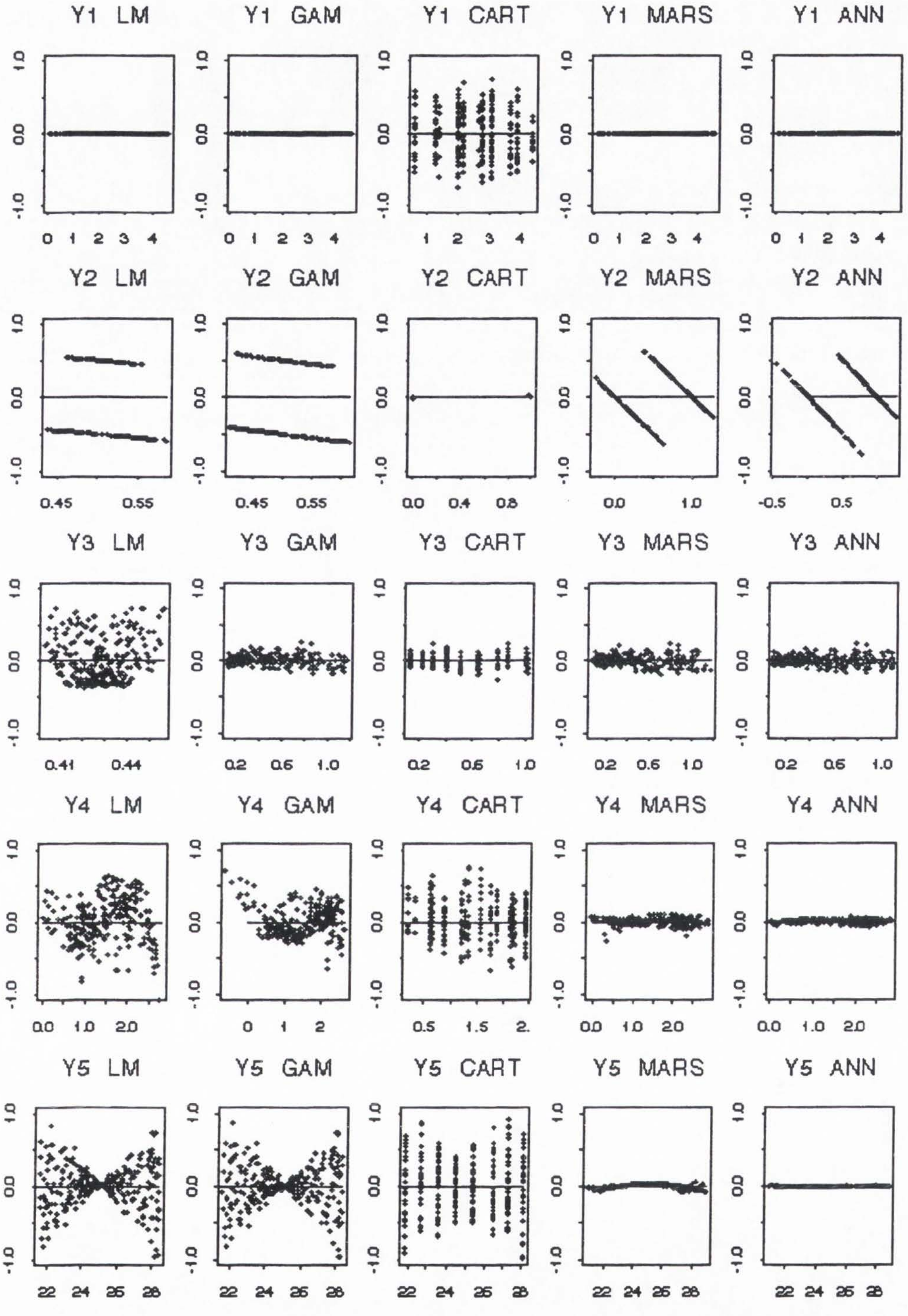


Figure 5-3. Illustration of Strength and Weaknesses of Different Modeling Techniques.

and ANN clear leaders for this illustration. Finally, Y5 illustrates how well ANNs perform with nested logistic functions, with MARS trailing as a distant second.

Mapping Results

Discrete Variables

Results from predictive mapping of the discrete variables are displayed in Figures 5.4-5.6, and presented in Appendix D-1. Figures 5-4, 5-5, and 5-6 show results for the three performance measures PCC (percent correctly classified), Kappa, and total computing time, respectively. Each individual dotplot on a page illustrates results by modeling technique (y axis) by variable (columns) within ecoregion (rows). The trellis graphics allow for quick visualization of a very large number of total model fits. Modeling techniques were ordered from best to worst (descending down Y axes in each plot) according to the median value of each performance measure across all variables and ecoregions.

The PCC (percent correctly classified) and Kappa results (Figures 5-4 and 5-5) suggest little difference between modeling techniques for identification of forest/non-forest but illustrate substantial gains over the NLCD approach in finer separation into three classes (FORTYP.3). These gains are made regardless of the nonlinear or nonparametric model chosen. The top two techniques (based on median values for individual performance measures) are MARS and GAMs for PCC (by a very slim margin), and GAMs and MARS for Kappa. By looking at the run time plot (Figure 5-6), there is a clear computational advantage in the NLCD and MARS models.

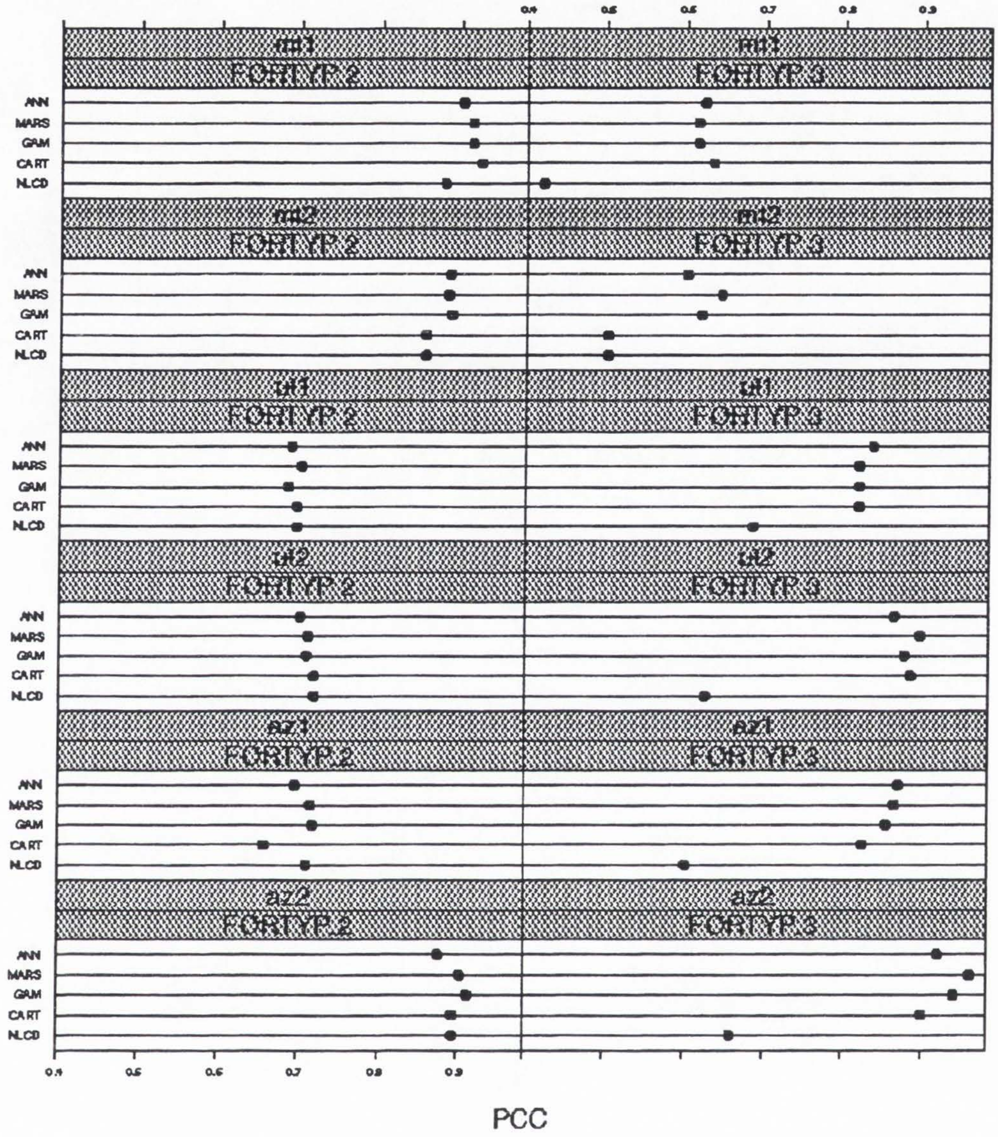


Figure 5-4. PCC for Models of Discrete Variables Using Test Data.

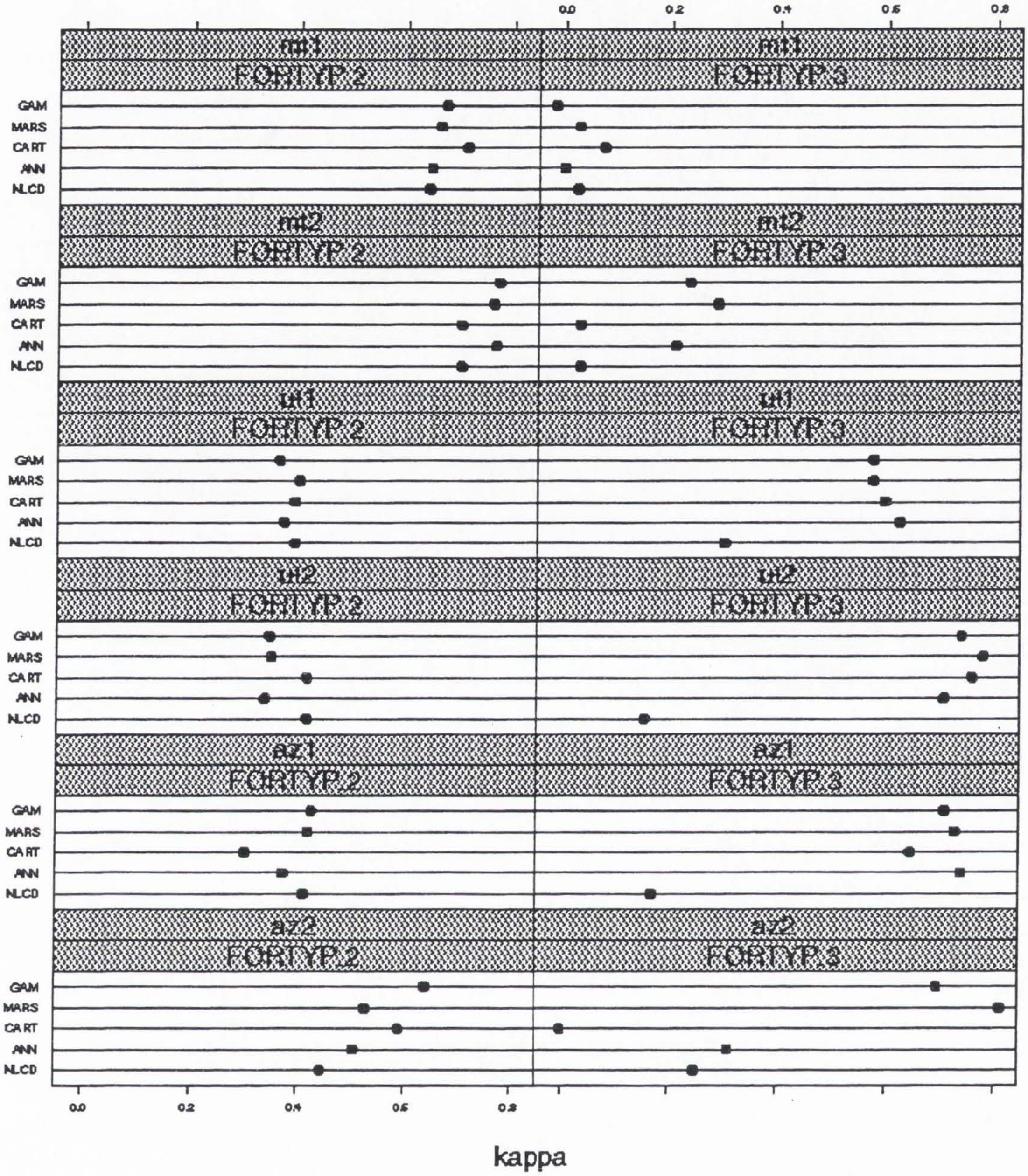


Figure 5-5. Kappa Statistic for Models of Discrete Variables Using Test Data.

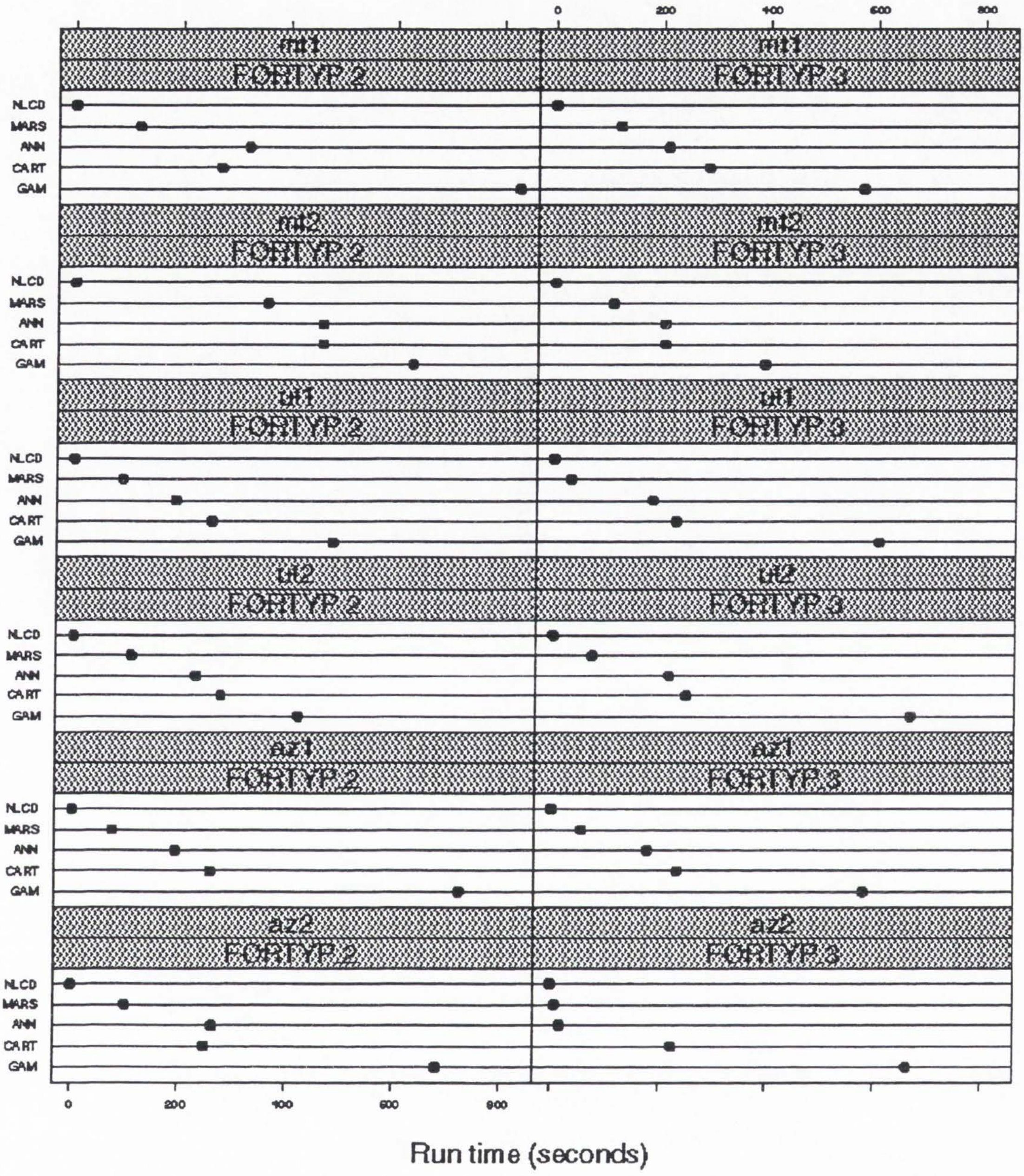


Figure 5-6. Computational Run Time for Models of Discrete Variables Using Test Data.

To further simplify the numerous graphs, a ranking scheme was devised whereby models were given a value from 1 (worst) to 5 (best) by variable and ecoregion for each of the performance measures. These values were then summed across performance measures and model ranks displayed in Figure 5-7. For discrete variables, the order using this strategy is MARS at the top, followed by GAM, CART, ANN, and NLCD. Of course, this gives equal weight to all performance measures and does not reflect the tiny differences that ultimately determine rank.

An example of a 1-km resolution map of predicted forest/non-forest in UT2 is given in Figure 5-8. An ascii file of UTM coordinates and predicted values were brought into a pre-made ArcView layout, easing the chore of generating map displays. "Prettier," finer resolution maps are currently in production for all variables in all ecoregions.

Continuous Variables

Results from predictive mapping of the continuous variables are tabled in Appendix D-2 and displayed in Figures 5-9 through 5-12. The layout in these figures is the same as in results for discrete variables. Results for the four performance measures (RMSE, RHO, PWI-25%, and computational run time) appear on the four separate figures. Figures 5-9 and 5-10 suggest that all five models often perform competitively for RMSE and PWI, but occasional erratic behavior by ANN, MARS, and CART can be anticipated. AZ2 was a good example where the small number of forested plots (165) and tremendous variability in total biomass made for unrealistic model prediction by ANN and MARS. However, better predictions were obtained in other ecoregions. GAM, ANN, and MARS appeared to perform best based on median values of RMSE, PWI, and RHO.

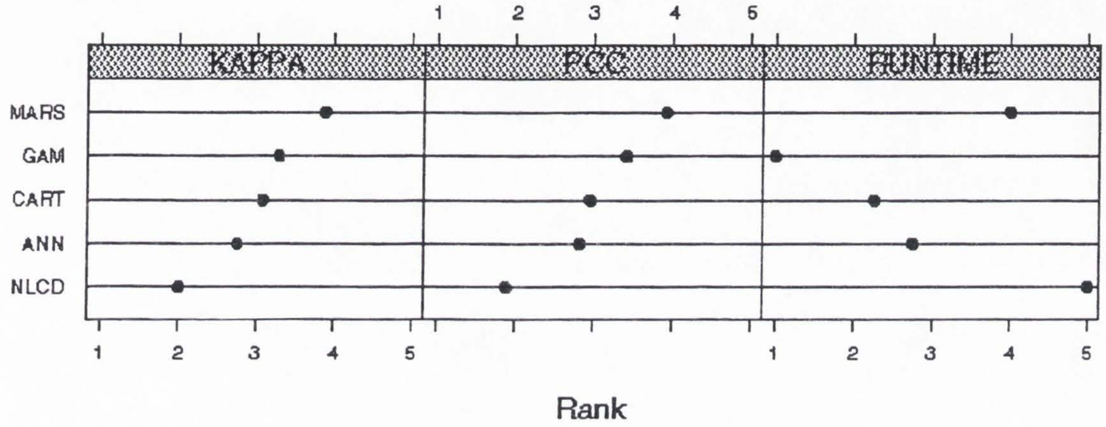


Figure 5-7. Ranking of Techniques for Discrete Variables.

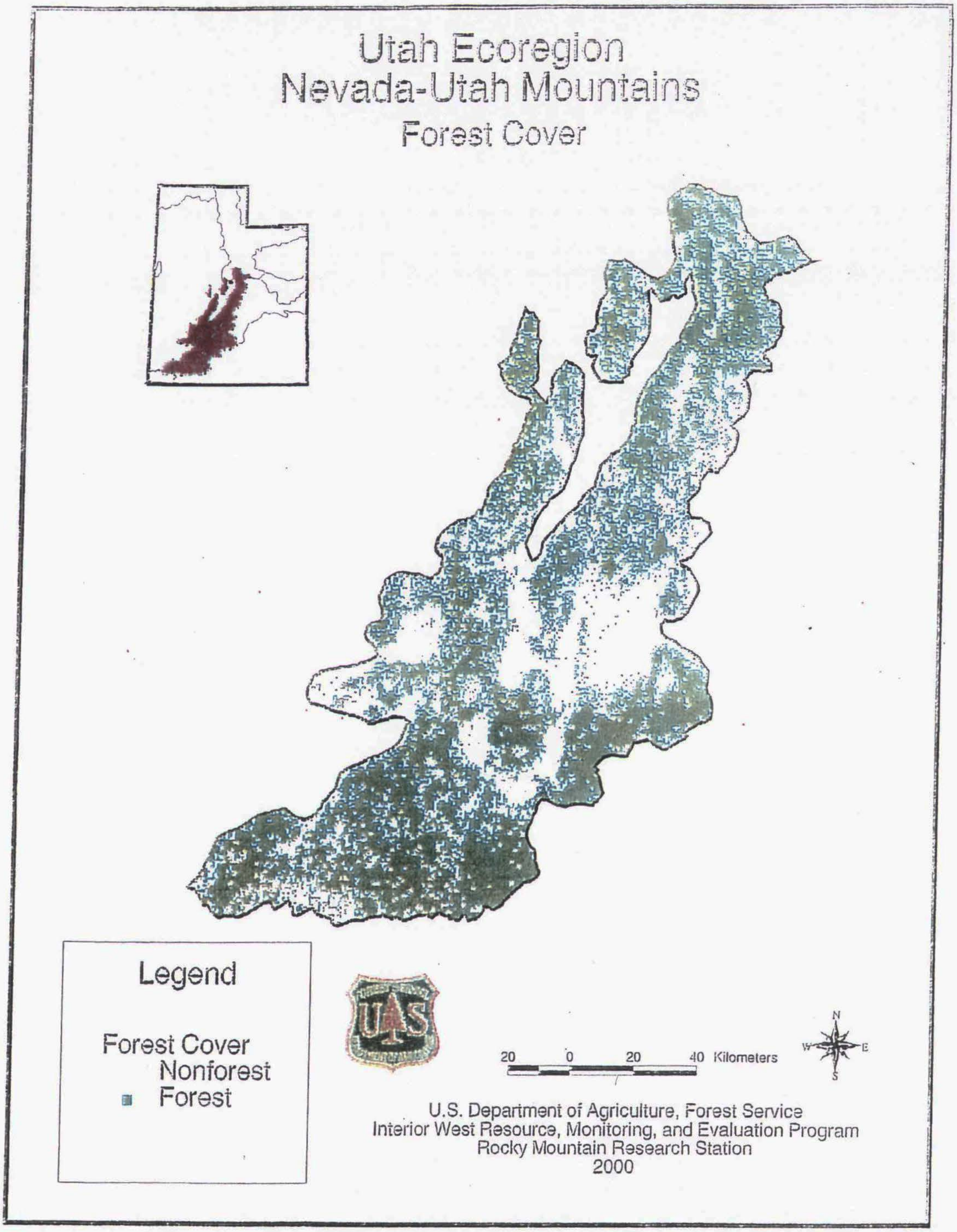

Utah Ecoregion
Nevada-Utah Mountains
Forest Cover
Legend
Forest Cover
Nonforest
▪ Forest
20 0 20 40 Kilometers
N
W
E
S
U.S. Department of Agriculture, Forest Service
Interior West Resource, Monitoring, and Evaluation Program
Rocky Mountain Research Station
2000

Figure 5-8. Map of Forested Areas Predicted from MARS Model in UT2.

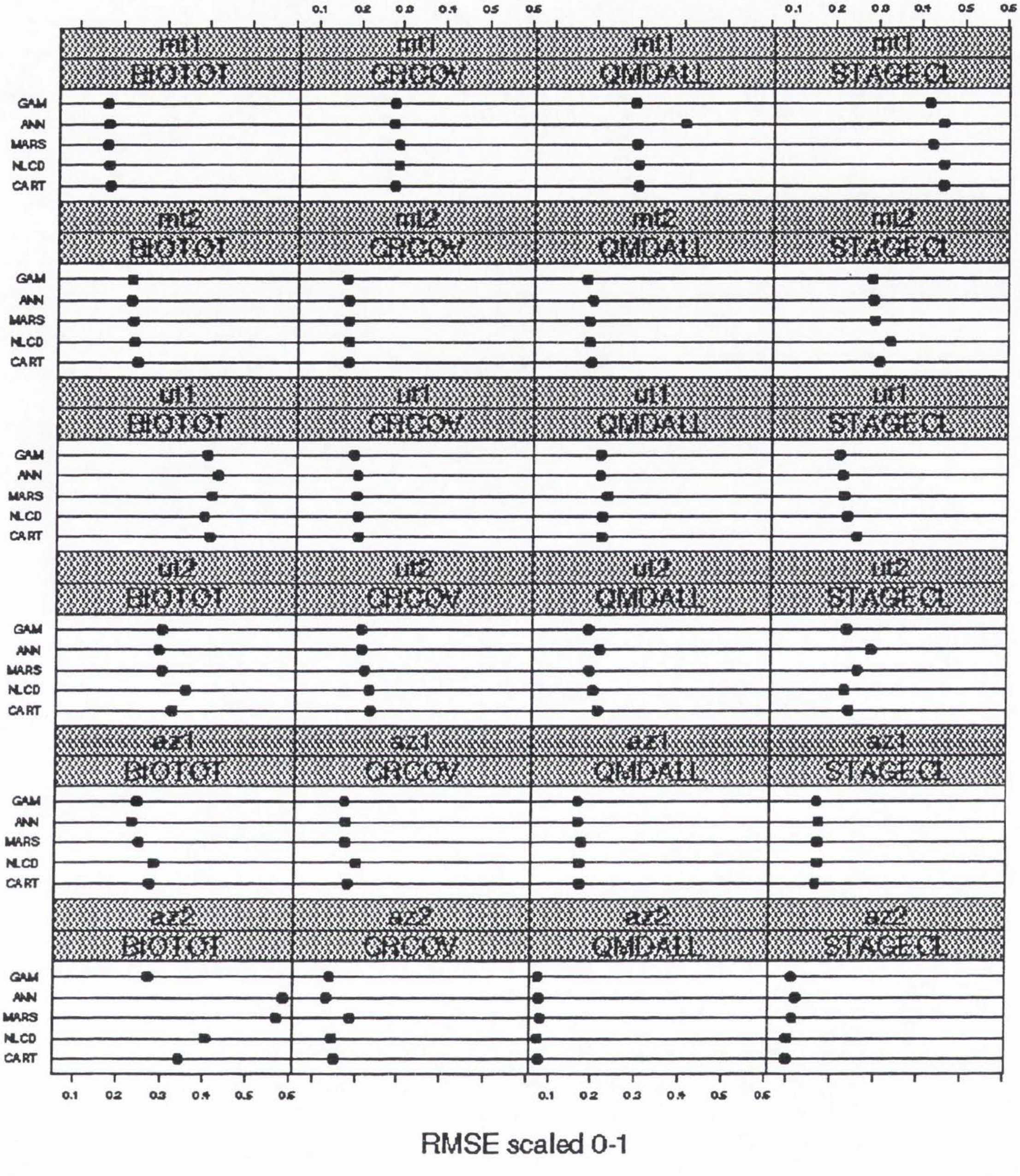


Figure 5-9. RMSE Expressed as a Proportion of the Mean for Models of Continuous Variables Using Test Data.

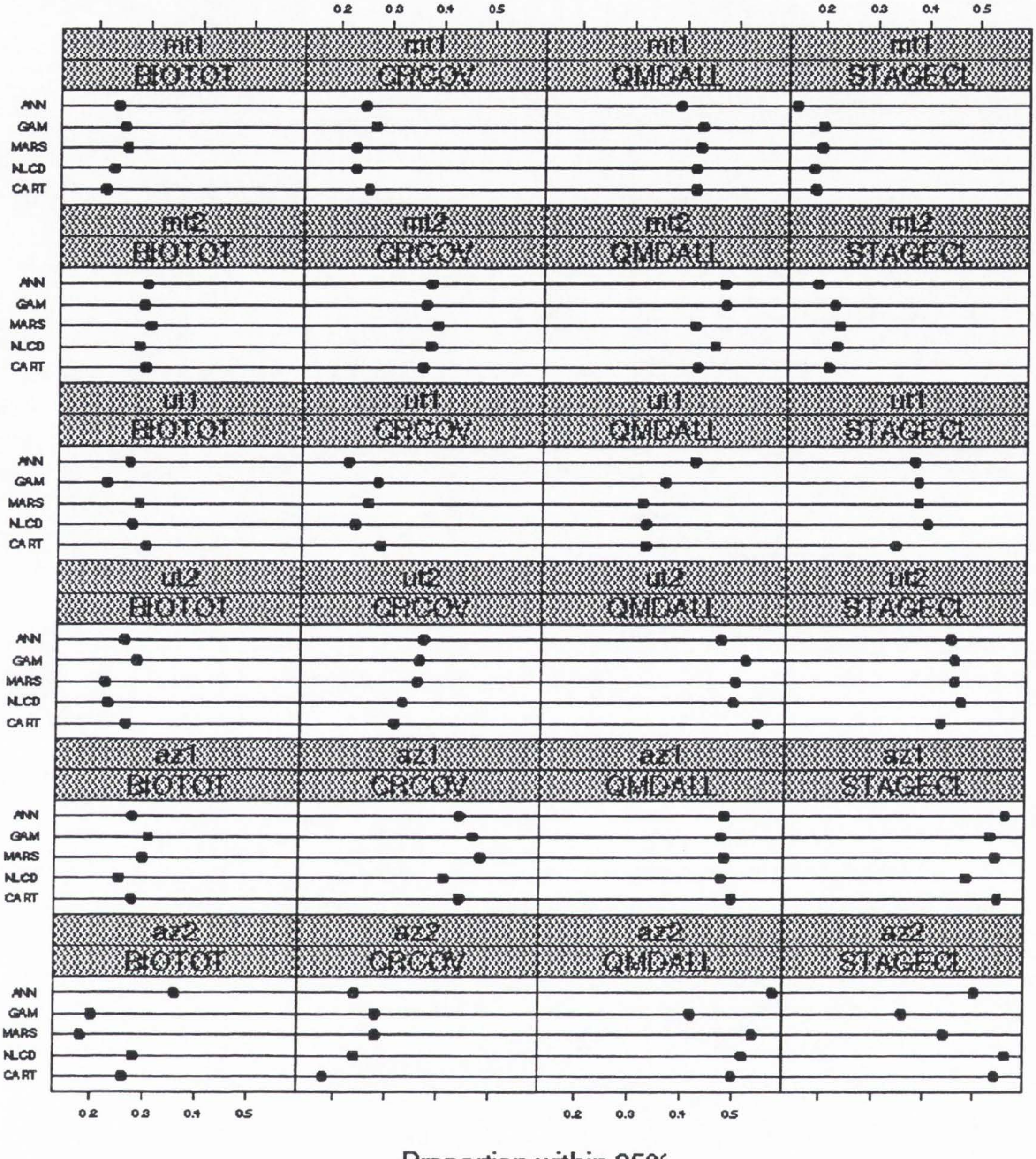


Figure 5-10. Proportion of Plots Within 25% of the Truth for Models of Continuous Variables Using Test Data.

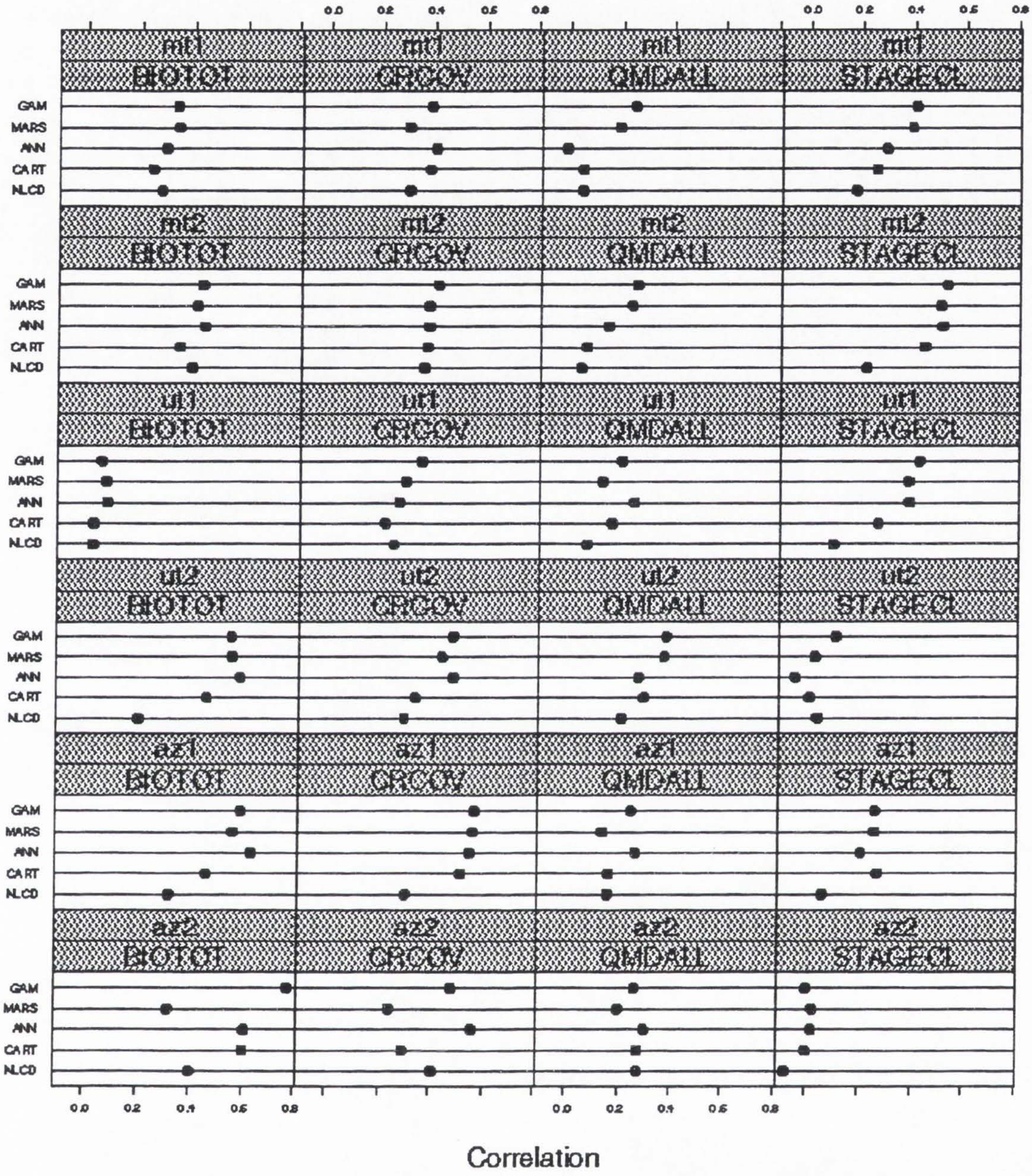


Figure 5-11. Correlation Between True and Predicted Values for Models of Continuous Variables Using Test Data.

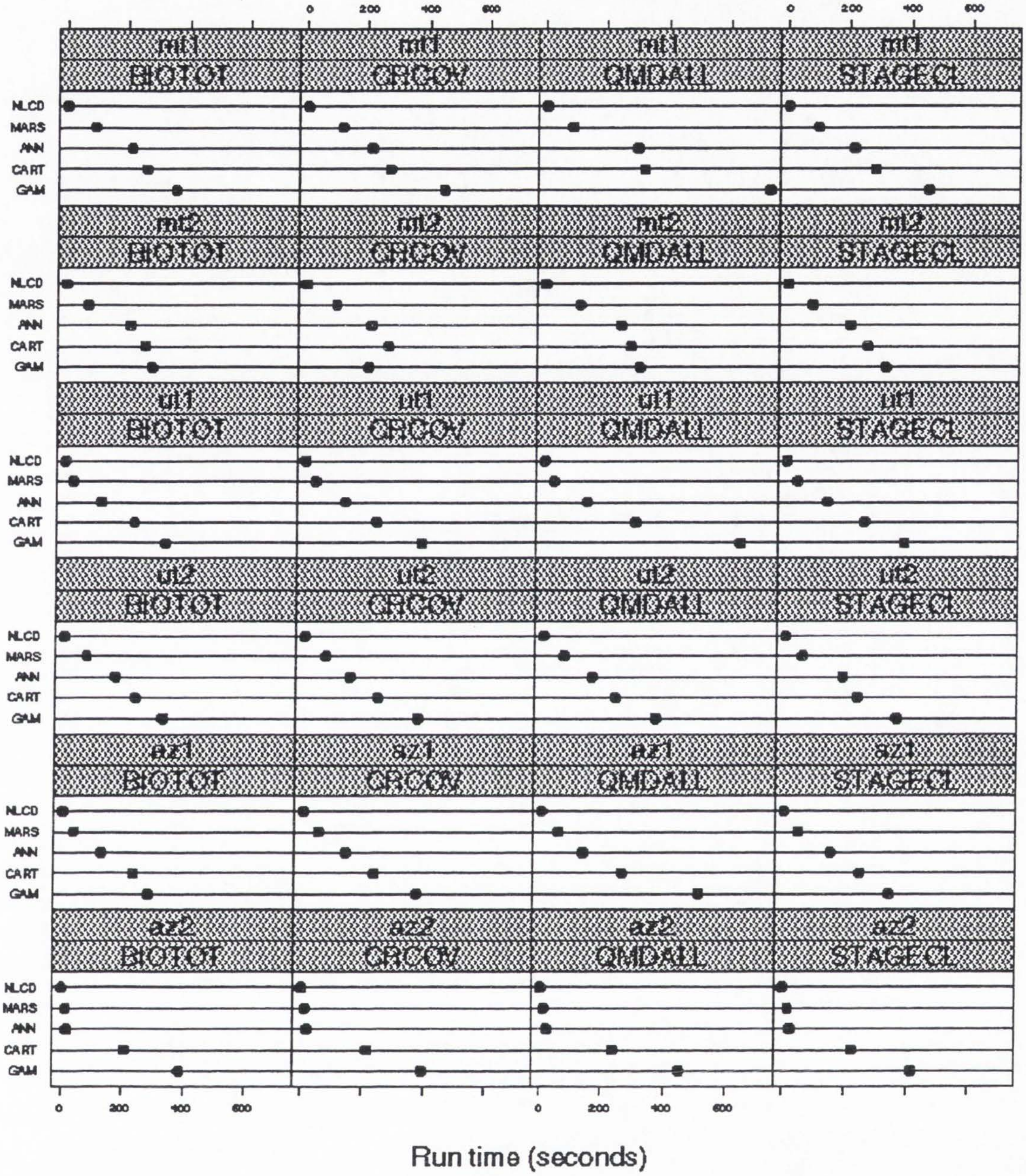


Figure 5-12. Computational Run Time for Models of Continuous Variables Using Test Data.

Values for RHO reflect the ability of GAMs and MARS to produce much more reasonable residual plots. An example of these plots for each modeling technique and response variable within UT2 is shown in Figure 5.13. These residual plots are representative of patterns seen in other ecoregions and, like the simulation examples in the first section of this chapter, illustrate the magnitude of the noise in the data as well as the small gains realized through alternative modeling techniques. As with the discrete variables, run time plots shown in Figure 5-12 illustrate the speed with which the simple NLCD and MARS models run relative to the others. Using the same ranking scheme as for models of discrete variables, Figure 5-14 puts GAMs first, followed by MARS, ANN, NLCD, and CART. Finally, Figure 5-15 is another example of a 1-km resolution map displaying predicted values for BIOTOT in UT2.

Stratification Results

Precision on Population Estimates

Figure 5-16 illustrates percent standard error in estimates of the four different population totals by stratification scheme within ecoregion. Figure 5-17 presents the same information using a different measure. Here, results are expressed as a ratio of the standard error under simple random sampling to the standard error under the scheme of interest. Recall the four population means include percent forest area, percent forest type within forested area, tree volume, and net annual growth. The stratification schemes include simple random sampling, stratification based on forest/non-forest calls from the NLCD data, and stratification using FORTYP.2 non-forest mask along with the FORTYP.3 classification of forested areas from MARS models in each ecoregion.

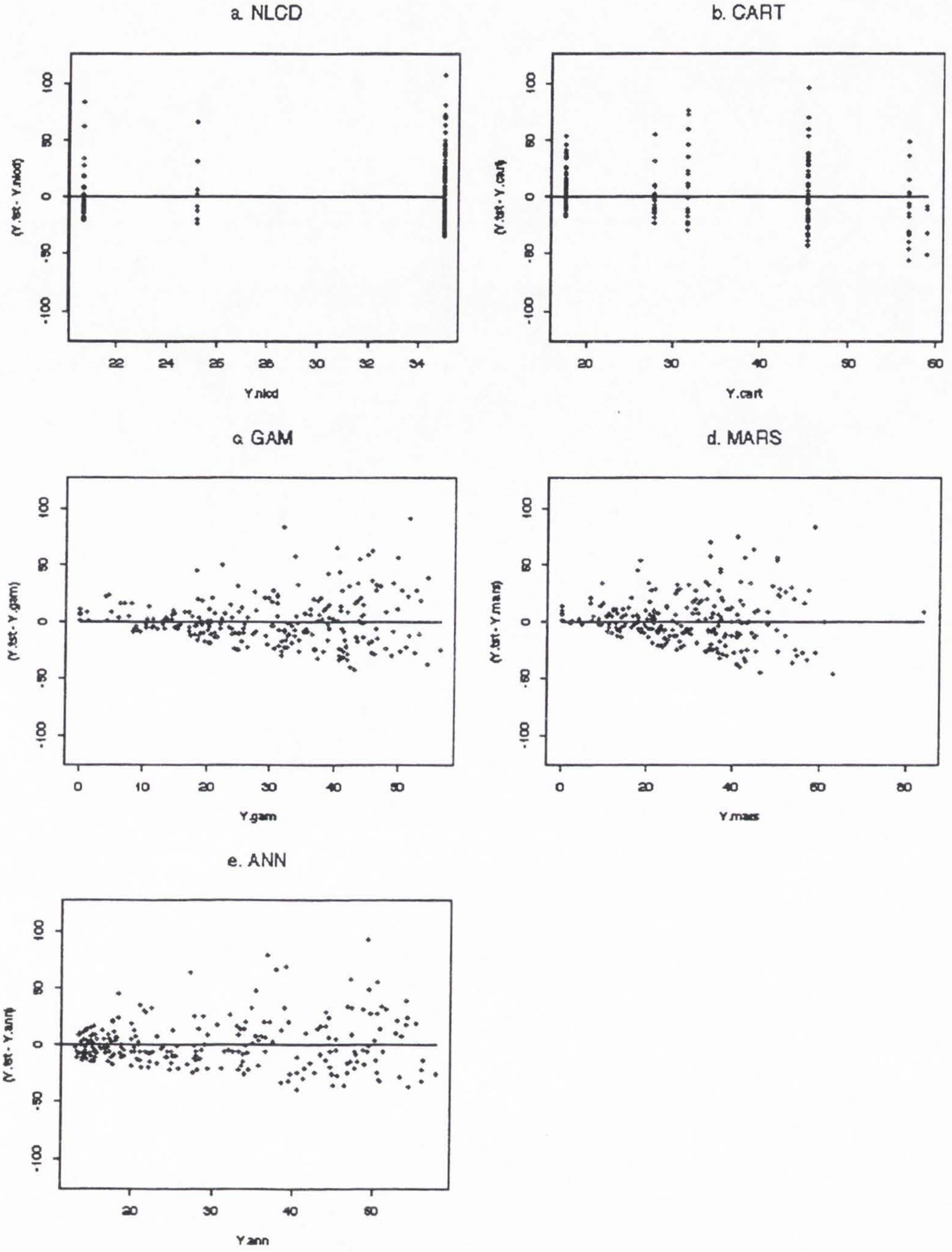


Figure 5-13. Residual Plots from Models of BIOTOT in UT2.

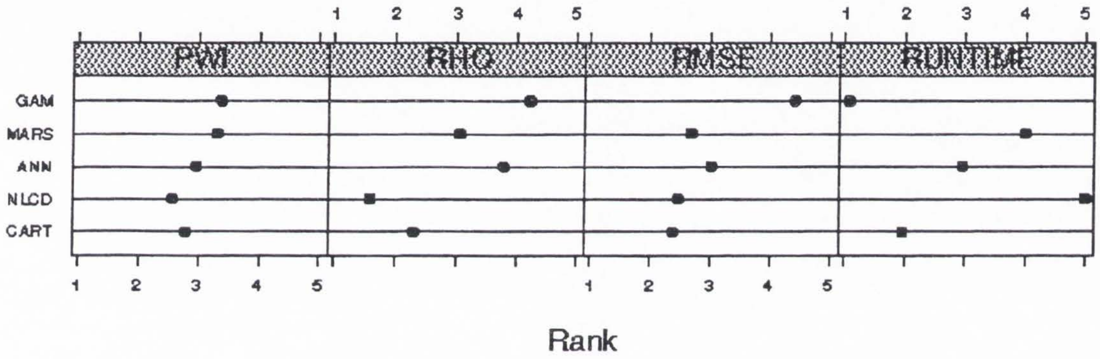


Figure 5-14. Ranking of Techniques for Continuous Variables.

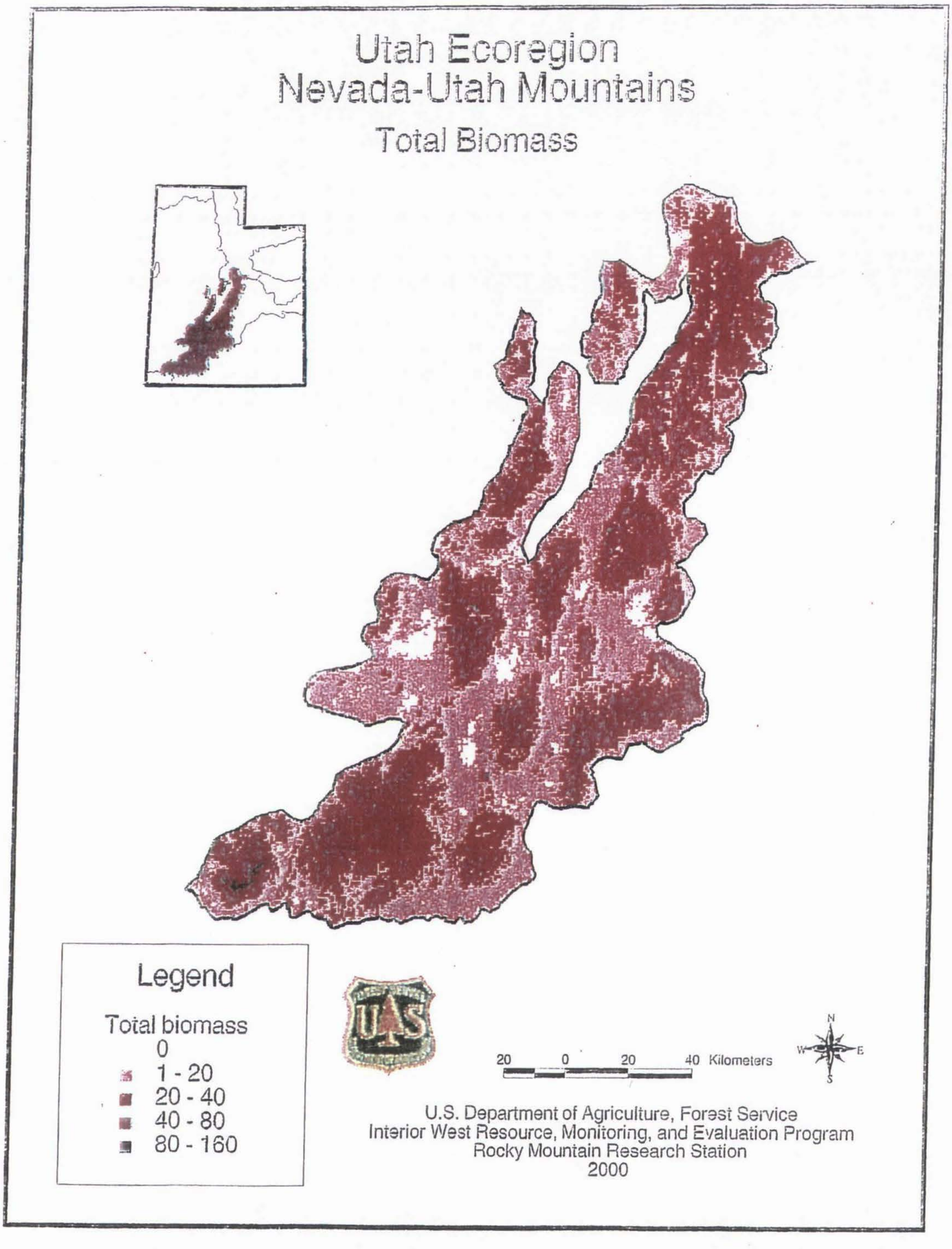

Utah Ecoregion
Nevada-Utah Mountains
Total Biomass
Legend
Total biomass
0
1 - 20
20 - 40
40 - 80
80 - 160
20 0 20 40 Kilometers
N
W
E
S
U.S. Department of Agriculture, Forest Service
Interior West Resource, Monitoring, and Evaluation Program
Rocky Mountain Research Station
2000

Figure 5-15. Map of Total Biomass Predicted from a MARS Model in UT2.

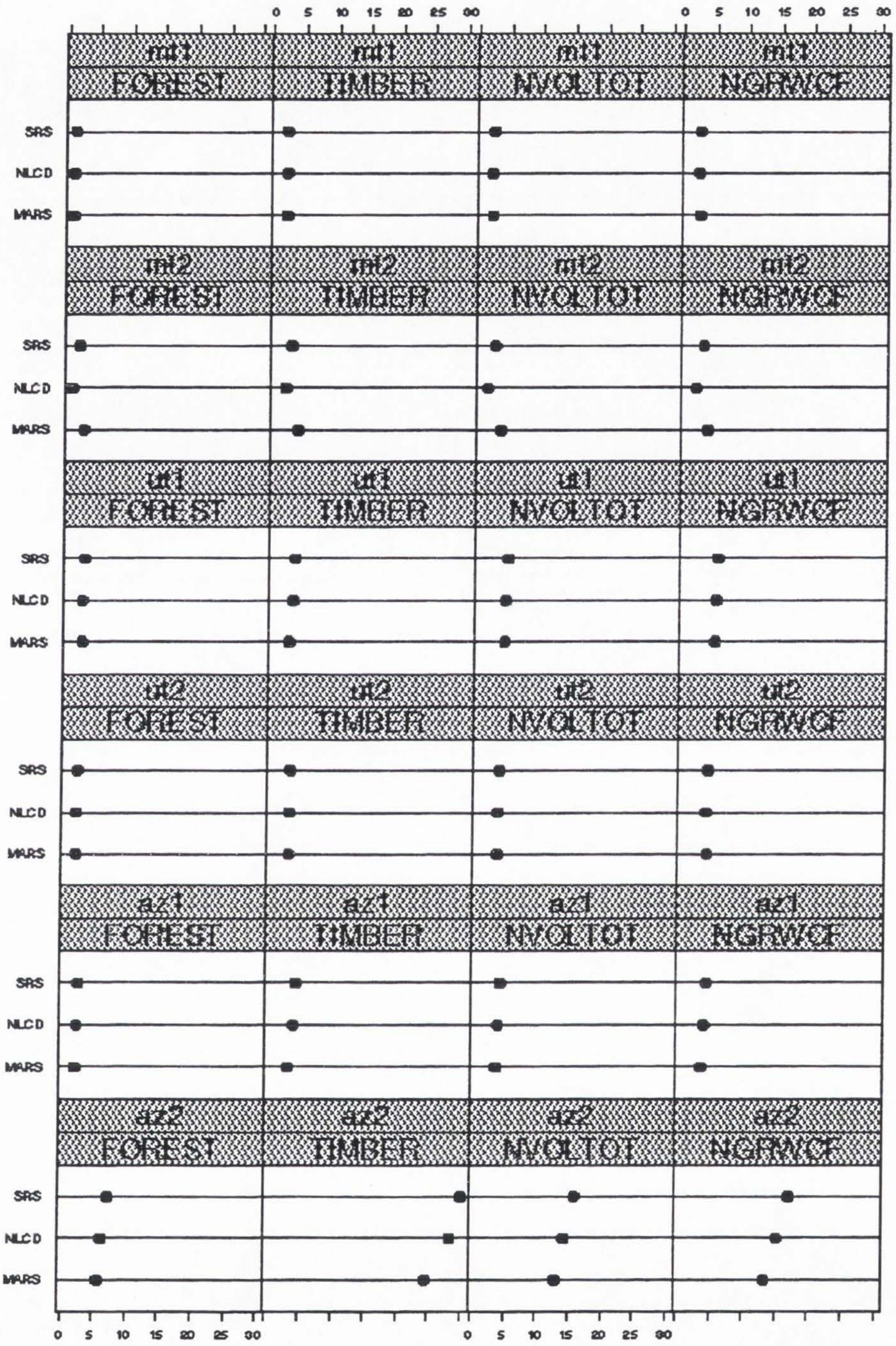


Figure 5-16. Percent Standard Error on Population Means.

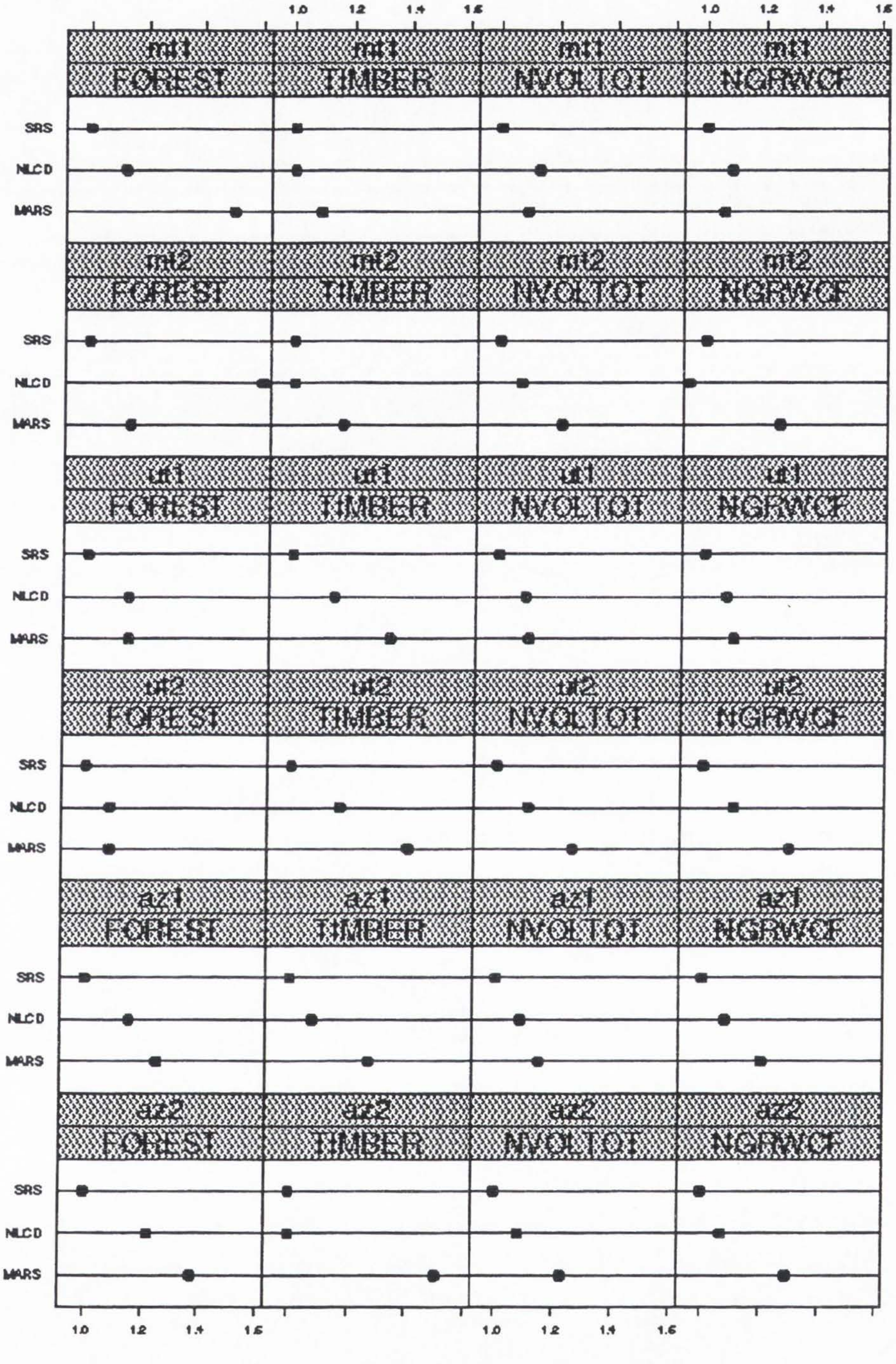


Figure 5-17. Ratio of Standard Errors on Estimates Using Simple Random Sampling to Standard Errors under Alternative Stratification Schemes.

In UT1, UT2, AZ1 and AZ2, this is stratification based on modeledtimberland/woodland/ non-forest. In MT1 and MT2, stratification is based on modeled spruce-fir/other forest/non-forest. Variances for this MARS as well as the NLCD approaches were obtained using stratified random sampling (STR) formulae as described in Chapter 4. Bootstrap estimates for the MARS strategy are discussed in the next paragraph. In Figure 5-16, stratification schemes are listed along the y-axis of each graph in the panels in order (best to worst) of percent standard error for each of the population variables. The figure shows the large gains in precision through NLCD stratification over SRS, and illustrates further gains using a more complex MARS stratification over the simple NLCD approach. As an example, Figure 5-17 illustrates how variance under simple random sampling may be 1.5 times larger than that under a MARS stratification in MT1 for estimating proportion of forest. This may seem like a substantial gain until one looks at the miniscule reduction in standard error when expressed as a percent of the mean (i.e., sample sizes are very large and standard errors small to begin with, Figure 5-16).

Bootstrap estimates of the variances under the MARS-based stratification scheme were run to see if the STR formulation was appropriate. Figure 5-18 illustrates results for four population estimates in AZ1. One hundred bootstrap samples were generated as described in Chapter 4, and bootstrap standard errors plotted for increasing number of iterations. The *circles* on the plots are the variances obtained for MARS-based stratification using STR formulation. The *triangles* are variances under the NLCD approach, and *plus* indicated variance using simple random sampling. The circles, triangles, and pluses are not related to number of bootstrap iterations, and are placed on

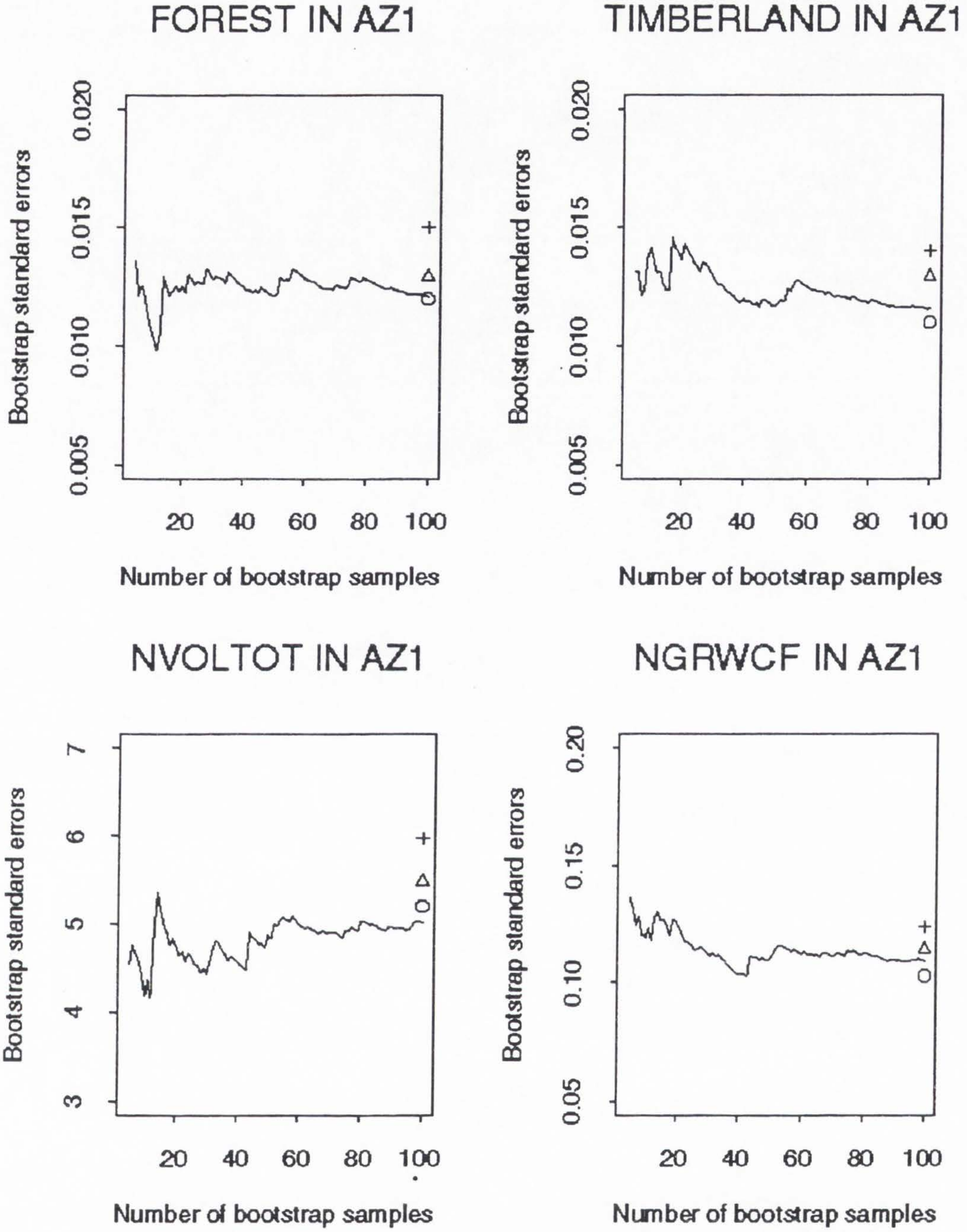


Figure 5-18. Results from Bootstrapping Variances of Estimates Using MARS-Based Stratification.

the plots to simply indicate values obtained using stratification or simple random sampling formulation of variances directly. Certainly, more investigation is needed before one can use STR formulation without reservation, but these initial runs are compelling. The ability to use sample plots in modeling strata that are in turn used to produce estimates of population totals would greatly enhance the current estimation process in forest inventories.

Relationship Between Map Accuracy and Precision Gains Through Stratification

One would expect that stratification based on the more accurately classified maps would result in smaller variances for estimates of population totals, particularly in estimates of area by forest type. However, Moisen and Edwards (1999) demonstrated that surprisingly small gains in efficiency in population estimates might be realized using "better" maps for stratification. The point was driven home again in Figure 5-16. Here, the general effect of map accuracy on the relative precision of estimates of population totals obtained under simple random sampling (SRS) to those obtained under stratified random sampling (STR) is explored analytically.

Estimating Population Proportions

In order to begin exploring the general relationship between map accuracy and the relative precision of estimates of population proportions obtained under the two designs, we need to make some simplifying assumptions. First, assume one of the strata (say h=1) is defined to closely mimic the class whose proportion we are interested in estimating. For example, if we are interested in estimating the proportion of timberland in a population, stratum 1 may be land *thought* to be timberland, stratum 2 may land *thought* to be woodland, and stratum 3 may land *thought* to be non-forest based on classified satellite imagery. Next, assume all the other strata are of approximately equal size, e.g., a 100,000 ha population may be divided into 50,000 ha of timberland, and 25,000 ha each of woodland and non-forest. Then for each of L strata we may write

$$0 < W_1 < 1, \text{ and } W_h = (1 - W_1)/(L-1), \text{ for } 2 \le h \le L. \tag{5.1}$$

Also assume classification accuracies are the same for all vegetation classes and that the misclassification is evenly distributed between classes. These accuracies will affect the proportion of the class of interest (P_h) within each of the strata as follows:

$$\begin{aligned} &P_1 = PCC, \text{ and} \\ &P_h = (1 - PCC)/(L-1), \text{ for } 2 \le h \le L. \end{aligned} \tag{5.2}$$

For example, if overall PCC for the map is 80%, then the PCC for timberland equals the PCC for woodland, which equals that for non-forest, namely 80%. If 80% of mapped timberland locations are indeed timberland, then 10% are misclassified as woodland and 10% as non-forest under the simplifying assumptions above. Table 5-4 illustrates what a confusion matrix might look like given 100 accuracy points collected in each stratum.

Recall that the true population proportion is the weighted sum of proportions across strata, so using Equation 5.1 we may write

$$P = \sum_{h=1}^{L} W_h P_h = W_1 P_1 + (L-1) W_h P_h = W_1 P_1 + (1 - W_1) P_h = W_1 P_1 + P_h - W_1 P_h, \tag{5.3}$$

where $2 \le h \le L$.

Now, the relative precision (RP) of SRS to STR is

Table 5-4. Confusion Matrix Given 100 Accuracy Points Collected in Each Stratum.

	True Timberland	True Woodland	True Non-Forest	User's Accuracy
Mapped Timberland	80	10	10	80%
Mapped Woodland	10	80	10	80%
Mapped Non-forest	10	10	80	80%
Producer's Accuracy	80%	80%	80%	80%

$$RP = \frac{V_{srs}}{V_{str}} = \frac{P(1-P)/n}{\sum_{h=1}^{L} \frac{W_h^2}{n_h} P_h(1-P_h)}.$$

Because $\frac{W_h^2}{n_n} = \frac{W_h^2}{W_h n} = \frac{W_h}{n}$, under proportional allocation the expression for the relative precision simplifies to

$$RP = \frac{P(1-P)}{\sum_{h=1}^{L} W_h P_h(1-P_h)}.$$

Under the assumptions and relationships stated in equations 5.1 – 5.3, the RP can be expressed as a function of W_1 and PCC alone by substituting for P, W_h, and P_h. The final equation is complicated and uninformative, but graphing the relative precision between SRS and STR for given W_1 and PCC values, as in Figure 5-19, sheds a considerable amount of light on the problem. From this graph we see relatively small gains in efficiency for PCC values below 70%. However, much more dramatic gains are realized for very high accuracy levels and mid-range primary stratum weights. The shape

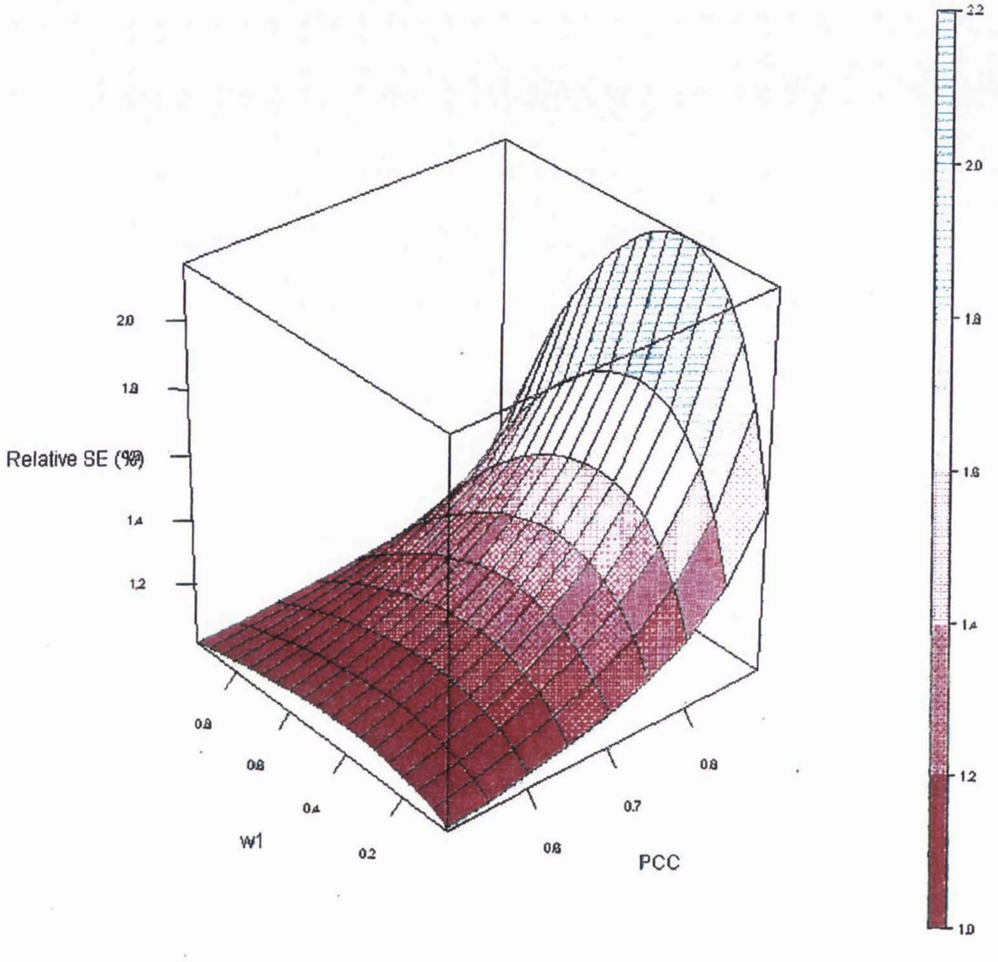


Figure 5-19. Ratio of Standard Errors Obtained Under Simple Random Sampling to Those Obtained under Stratified Random Sampling for Estimating Population Proportions.

of this graph remains unchanged regardless of the number of strata. Increasing the number of strata improves relative precision only if it effectively increases the "accuracy" of mapped classes for the proportion of interest. As an example, changing from a simple forest/non-forest stratification to a timberland/woodland/non-forest stratification might help distinguish timberland from other forested areas, thus improving an estimate of total area of timberland.

Estimating Population Means for Continuous Variables

The relative precision of STR to SRS when we are interested in estimating population means or totals from continuous variables may also be expressed as a function of the PCC and the percent reduction in variance in "pure" (100% accurate classes). For example, suppose we are interested in the total volume of wood in a region. Assume stratification into perfectly classified hardwood forest and softwood forest results in within strata variances that are some proportion (call it R) of the overall population variance for volume. As strata become less perfect in terms of their classification accuracy, that (1-R) gain in precision is reduced further. The following analysis quantifies that reduction.

If within "pure strata" variance is a proportion R of the overall population variance in a SRS, and the same as the SRS variance elsewhere, then we can write

$$S_h{}^2 = PCC(R)S^2 + (1 - PCC)S^2$$

for equal strata sizes. Then, under proportional allocation, the RP can be written

$$\begin{aligned} RP &= \frac{V_{srs}}{V_{str}} \\ &= \frac{S^2}{\sum W_h S_h^2} \\ &= \frac{S^2}{\frac{1}{L}(PCC(R)S^2 + (1-PCC)S^2)L} \\ &= \frac{1}{PCC(R) + (1-PCC)}. \end{aligned}$$

Here, the number of strata itself does not affect the RP unless it results in higher PCC values for the class of interest, or greater homogenization of continuous variables.

This relationship is illustrated in Figure 5-20. Note the flatness of the graph until variance within pure classes is around 30% or less of the population variance, or the accuracy extremely high.

Potential Reduction in Sample Size Through Stratification

For both estimation of population proportions and means of continuous variables, the percent reduction in sample size that would be possible under STR, if one only had to achieve the same precision as SRS, may be calculated as follows. Given

$$RP = \frac{P(1-P)}{\sum W_h P_h (1-P_h)} = \text{an expression involving } n_2.$$

Setting this expression equal to 1 and solving for n_2 in terms of the relative precision, RP, and n_1, one gets

$$1 = \frac{\frac{1}{n_1} P(1-P)}{\frac{1}{n_2} \sum W_h P_h (1-P_h)} = \frac{n_2}{n_1} RP.$$

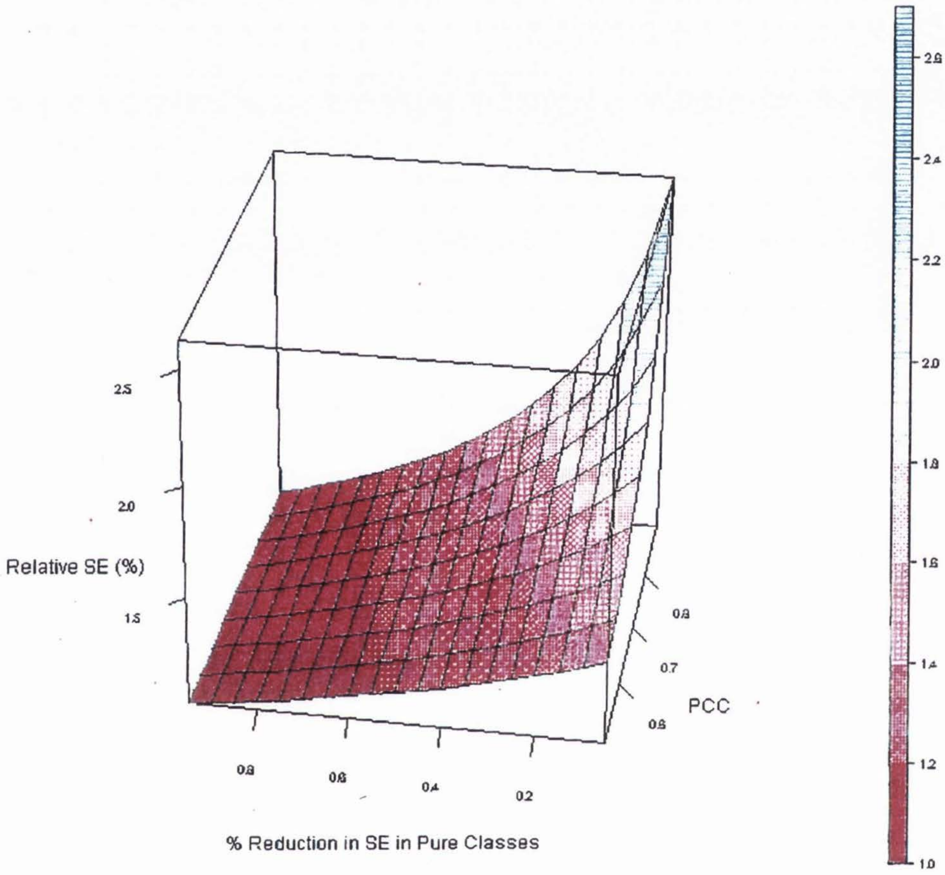


Figure 5-20. Ratio of Standard Errors Obtained under Simple Random Sampling to Those Obtained under Stratified Random Sampling for Estimating Continuous Population Variables.

So, $n_2 = \frac{n_1}{RP}$. This relationship is illustrated in Figures 5-21 and 5-22 for proportions and continuous variables, respectively.

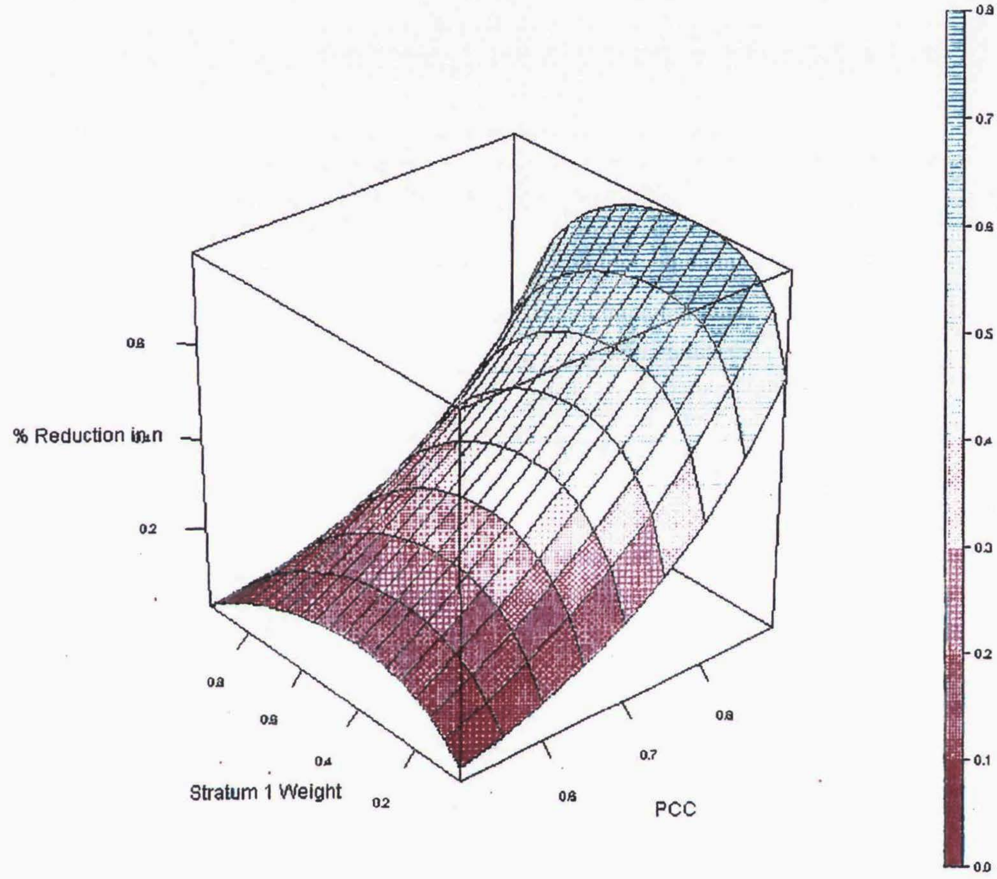


Figure 5-21. Percent Reduction in Sample Size Through Stratification for Estimating Population Proportions.

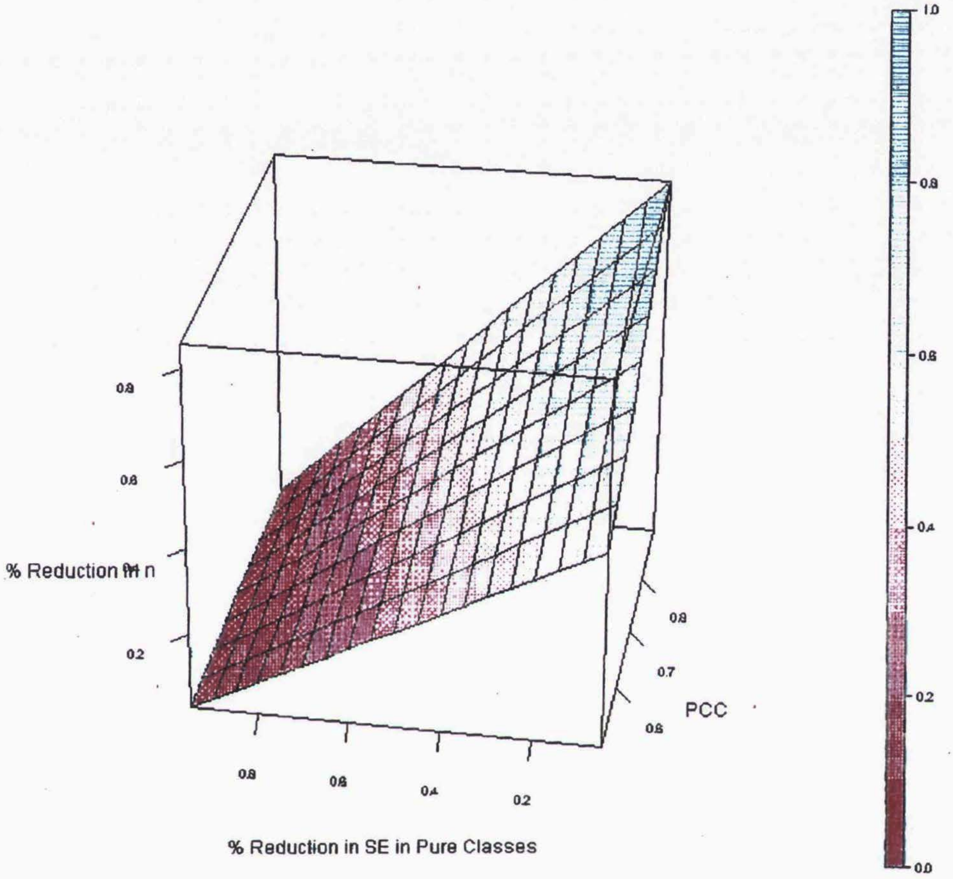


Figure 5-22. Percent Reduction in Sample Size Through Stratification for Estimating Continuous Population Variables.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

Discussion

Objectives

Recall that the first objective of this research was to develop an automated mapping and stratification system for forest inventories in the Interior West. The objective was accomplished and the system developed in an S-Plus computing environment. This modeling "box" serves as a key component in an interdisciplinary system for integrating ancillary data with forest inventories for delivery of new products and more cost-effective information (Figure 6-1). Any predictor variables in grid or ASCII format can be merged with FIA data using any of the five modeling techniques within the box. The outputs include predictions and potential for analysis using five modeling techniques, a report of map accuracy for discrete and continuous variables, estimates under an assortment of stratification strategies, as well as flat files for building predictive maps. The box also provides the machinery for a tremendous amount of future research using real and simulated data.

In addition, development of this modeling box prompted rapid development of other boxes in the interdisciplinary system shown in Figure 6-1. Data extraction processes from both the "Field" and the "Digital" boxes have come a long way, as described in Chapter 3. In addition, the rapid output of predictive maps made possible through the modeling box, as well as interest from forest managers, has prompted further

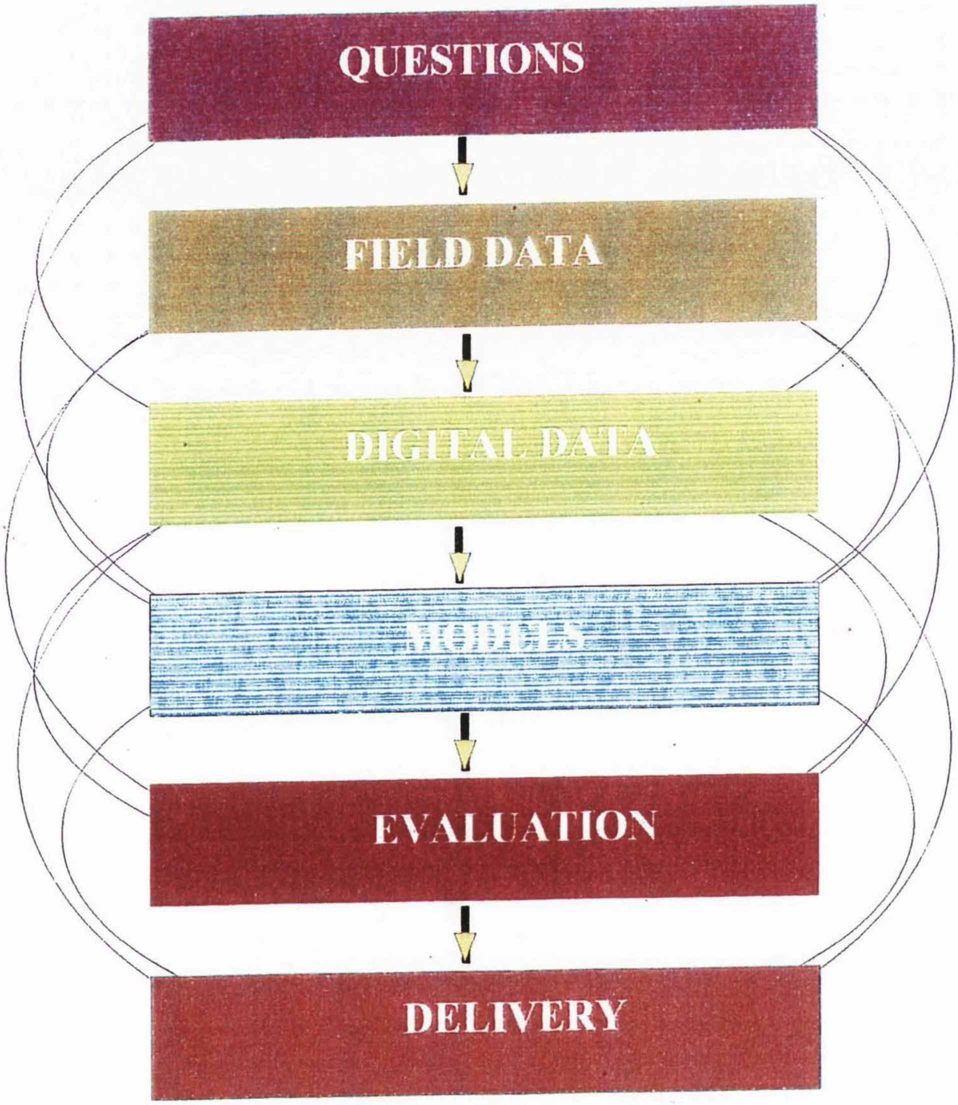


Figure 6-1. Interdisciplinary System.

development of the "Delivery" box for hard and softcopy maps, analysis tools, and web-based production.

The second objective was to determine which of the modeling techniques were best suited for a forest inventory production environment. All techniques tried here proved themselves workable in an automated environment, although ANNs were a bit more problematic. Computation run time is one area the modeling techniques differed substantially. Naturally, the simple NLCD model was extremely fast with no computational "glitches." GAMs and CARTs are normally quite fast but were considerably slower here because of the stepwise procedures for GAM and iterative runs searching for best tree size for CART. ANNs were the slowest in these applications, and have the potential to be cripplingly slow for "slow but safe" parameter optimization procedures in FUNFITS. Obviously, the simplest NLCD approach or another simple linear model is most readily incorporated into a production process. But of the more flexible techniques, MARS showed promise in a production environment because of its fast computing rate, little need for user "steering," and tendency to produce reasonable models when ANN failed. Certainly, any of the models could be made production suitable, and a sensible strategy may well be to keep all the tools in the toolkit, using several for each application.

The final objective was to determine if introducing more flexible statistical models into forest inventory mapping and stratification procedures makes an appreciable difference in accuracy of forest maps and precision in estimates of population totals, respectively. Many valuable lessons were learned through this work as discussed below.

Mapping

This simple simulation described in the beginning of Chapter 5 illustrated that use of a flexible and powerful modeling technique *can* make a *huge* difference in predictive performance when the signal-to-noise ratio is high. The test also shed some light on the character of each technique. It was surprising that CART performed worse than a simple linear model. It was also surprising that GAM's stepwise procedure was not able to exclude all the noncontributing variables. In addition, the ease with which both MARS and CART established the relationship of Y to the predictor variables was very informative.

The differences between modeling techniques using real data were far less impressive. In fact, for a number of variable/ecoregion combinations, only small differences were realized using any of the modeling techniques over a simple NLCD approach, particularly for distinguishing forest/nonforest, or in RMSE for continuous variables. Larger gains were realized, however, for further classification of forested areas (FORTYP.3) and in getting predictions that fell within a user-specified ballpark. In addition, slightly higher correlations were realized for MARS and GAMs. This was seen in residual plots where more realistic predictions were obtained for extreme lows (in both MARS and GAMs) and extreme highs (for MARS).

When starting this analysis with the real data, I had anticipated seeing marked differences between modeling techniques. The small gains seen with these data sets were at first disheartening, but understandable given the tremendous amount of noise in the data. Sources of noise are numerous and include: positional error in field plots, registration difficulties between plots and images, scale differences between data collected in the field and the imagery, differences in date, definitional differences, and the list goes on. Based on the results one might be inclined to stick with a simple NLCD model from mapping. Yet, the data are in a constant state of change. GPS coordinates with national standards are now being collected on all field plots, better resolution imagery with standardized registration procedures are becoming available, softcopy low altitude photography is under development, and better resolution topographic information will be available shortly. Given all that, the true benefit of a new predictor variable might be overlooked if only linear models were in place. So, building MARS or ANN into a predictive mapping system up front is likely to have big payoffs down the road, even if differences between that and a much simpler approach are only marginal right now.

Stratification

Finding that use of the simple NLCD data alone for stratification results in estimates of population totals that substantially improved SRS estimates, and meet National standards for precision, is very useful. This results in substantial cost savings over the prior two-phase sampling procedures using expensive and labor-intensive photo interpretation for stratification. The analyses also illustrate the additional gains that can be realized when using ancillary data and a modeling technique like MARS. These gains might provide some cost savings in annual inventory systems in the future where as much information as possible needs to be squeezed out of ancillary data.

Another valuable result is that an increase in accuracy in a map used for stratification does not translate linearly into gains in precision in estimates of population totals. The last section of Chapter 5 provides graphical tools for managers trying to decide how accurate a map is needed for stratification, and what the trade-off is between sampling size and precision in lieu of known accuracy for stratification maps. These are very helpful pieces of information for management decisions and tough choices on where sampling money should go.

Conclusions

Here, I developed an automated mapping and stratification system well suited for regional forest inventories in the interior west, but portable to other applications. Any predictor variables in grid or ASCII format can be merged with response variables using any of the five modeling techniques in the box. The outputs include predictions and potential for analysis using five modeling techniques, a report of map accuracy for discrete and continuous variables, estimates under an assortment of stratification strategies, as well as flat files for building predictive maps.

In comparing the different modeling techniques, all proved themselves workable in an automated environment, though the simple NLCD and MARS required the least amount of user input or "tinkering." When explored through a simple simulation, tremendous advantages were seen in use of MARS and ANN for prediction, but much smaller differences were seen when using real data because of noise or possible lack of nonlinear relationships between the response and predictor variables. The simple NLCD model had the computational advantage, but MARS performed (marginally) best most often for binary variables, while GAMs did (marginally) better most often for continuous variables. Ranking was based on measures of map accuracy, predictive performance, and computing run time. Although little appreciable difference was seen between the models, as better predictor variables become available, tremendous advantages may be realized using more flexible statistical techniques.

For stratification, using the simple NLCD data alone for stratification resulted in estimates of population totals that improved SRS estimates, and met National standards for precision. This results in substantial cost savings over the prior two-phase sampling procedures using expensive and labor-intensive photo interpretation for stratification. The analyses also illustrate the additional gains that can be realized when using ancillary data and a modeling technique like MARS. In addition, the general effect of map accuracy on the relative precision of estimates of population totals obtained under simple random sampling (SRS) to those obtained under stratified random sampling (STR) was established for simple sampling scenarios.

REFERENCES

Atkinson, P. M., Cutler, M. E. J., and Lewis, H. (1997), "Mapping Sub-Pixel Proportional Land Cover with AVHRR Imagery," *International Journal of Remote Sensing,* 18, 917–936.

Atkinson, P. M., and Tatnall, A. R. L. (1997), "Neural Networks in Remote Sensing," *International Journal of Remote Sensing*, 18, 699-709.

Aurelle, D., Lek, S., Giraudel, J., and Berrebi, P. (1999), "Microsatellites and Artificial Neural Networks: Tools for the Discrimination Between Natural and Hatchery Brown Trout in Atlantic Populations," *Ecological Modeling*, 120, 313-324.

Austin, M. P., and Meyers, J. A. (1996), "Current Approaches to Modeling the Environmental Niche of Eucalypts: Implications for Management of Forest Biodiversity," *Forest Ecology and Management*, 85, 95-106.

Austin, M. P., Nicholls, A. O., Doherty, M. D. and Meyers, J. A. (1994), "Determining Species Response Functions to an Environmental Gradient by Means of a Beta Function," *Journal of Vegetation Science,* 5, 215-228.

Bailey, R. G., Avers, P. E., King, T., and McNab, W. H. (eds.) (1994), "Ecoregions and Subregions of the United States" (map), U.S. Geological Survey, Washington DC.

Baker, F. A., Verbyla, D. L., and Hodges, C. S. Jr. (1993), "Classification and Regression Tree Analysis for Assessing Hazard of Pine Mortality Caused by Hetero Basidion Amosum," *Plant Disease,* 77, 136-139.

Barciela, R. M., Garcia, E., and Fernandez, E. (1999), "Modeling Primary Production in a Coastal Embayment Affected by Upwelling Using Dynamic Ecosystem Models and Artificial Neural Networks," *Ecological Modeling*, 120, 199-211.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees*, Monterey, CA: Wadsworth and Brooks/Cole.

Brosse, S., Guegan, J., Tourenq, J., and Lek, S. (1999), "The Use of Artificial Neural Networks to Assess Fish Abundance and Spatial Occupancy in a Littoral Zone of a Mesotrophic Lake," *Ecological Modeling,* 120, 299-311.

Brown, D. G. (1994), "Predicting Vegetation Types at Treeline Using Topography and Biophysical Disturbance Variables," *Journal of Vegetation Science*, 5, 641-656.

Bruzzone, L., Conese, C., Maselli, F., and Roll, F. (1997), "Multisource Classification of Complex Rural Areas by Statistical and Neural Network Approaches," *Photogrammetric Engineering and Remote Sensing,* 63, 523–533.

Cheng, B., and Titterington, D. M. (1994), "Neural Networks: A Review from a Statistical Perpspective," *Statistical Science*, 9, 2-54.

Chojnacky, D. C. (1998), "Double Sampling for Stratification: a Forest Inventory Application in the Interior West," Research Paper RMRS-RP-7, Department of Agriculture, Forest Service, Rocky Mountain Research Station, Ogden, UT.

Cialella, A.T., Dubayah, R., Lawrence, W., and Levine, E. (1997), "Predicting Soil Drainage Class Using Remotely Sensed and Digital Elevation Data," *Photogrammetric Engineering and Remote Sensing* 63, 171-178.

Cochran, W.G. (1977), *Sampling Techniques*, New York: Wiley.

Cohen, J. (1960), "A Coefficient of Agreement of Nominal Scales," *Educational and Psychological Measurement* 20, 37-46.

Colasanti, R. L. (1991), "Discussions of the Possible Use of Neural Network Algorithms in Ecological Modeling," *Binary*, 3, 13-15.

Dai, X. L., and Khorram, S. (1999), "Remotely Sensed Change Detection Based on Artificial Neural Networks," *Photogrammetric Engineering and Remote Sensing* 65, 1187–1194.

DeVeaux, R. D., (1995), "A Guided Tour of Modern Regression Methods," in *1995 Fall Technical Conference Proceedings of the Section on Physical and Engineering Sciences*, eds. P. D. Baker and T. Cardoso, January 1995, New York, 102-109.

DeVeaux, R. D., Psichogios, D. C., and Ungar, L. H. (1993), "A Comparison of Two Nonparametric Estimation Schemes: MARS and Neural Networks," *Computers in Chemical Engineering*, 8, 819-837.

Dimopoulos, I., Chronopoulos, J., Chronopoulos-Sereli, A., and Lek, S. (1999), "Neural Network Models to Study Relationships Between Lead Concentration in Grasses and Permanent Urban Descriptors in Athens City," *Ecological Modeling*, 120, 157-165.

Edwards, M., and Morse, D. R. (1995), "The Potential for Computer-Aided Identification in Biodiversity Research," *Trend in Ecological Evolution*, 10, 153-158.

Edwards, T. C., Jr., Deshler, E., Foster, D., and Moisen, G. G. (1996), "Adequacy of Wildlife Habitat Relation Models for Estimating Spatial Distributions of Terrestrial Vertebrates," *Conservation Biology,* 10, 263-270.

Foody, G. M., and Arora, M. K. (1997), "An Evaluation of Some Factors Affecting the Accuracy of Classification by an Artificial Neural Network," *International Journal of Remote Sensing,* 18, 799–810.

Foody, G. M., Lucas, R. M., Curran, P. J. and Honzak, M. (1997), "Nonlinear Mixture Modeling Without End-Members Using an Artificial Neural Network," *International Journal of Remote Sensing*, 18, 937–954.

Frescino, T. S, Edwards, T. C., Jr., and Moisen, G. G. (In press), "Modeling Spatially Explicit Forest Structural Attributes Using Generalized Additive Models," *Journal of Vegetation Science.*

Friedl, M.A. and C.E. Brodley (1997), "Decision Tree Classification of Land Cover from Remotely Sensed Data," *Remote Sensing of Environment*, 61, 399-409.

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines," *Annals of Statistics*, 19, 1-141.

Gong, P., Pu, R., and Chen, J. (1996), "Mapping Ecological Land Systems and Classification Uncertainties from Digital Elevation and Forest Cover Data Using Neural Networks," *Photogrammetric Engineering and Remote Sensing* 62, 1249–1260.

Grubb, T.G., and King, R.M. (1991), "Assessing Human Disturbance of Breeding Bald Eagles with Classification Tree Models", *Ecological Modeling*, 120, 237-246.

Hastie, T., and Tibshirani, R. J. (1986), "Generalized Additive Models," *Statistical Science*, 1, 297-318.

Hastie, T. J., and Tibshirani, R. J. (1990), *Generalized Additive Models*, New York: Chapman and Hall.

Hockman, J. N., Burger, G. A., and Smith, D. W. (1990). "Classification Model to Predict Frasier Fir Christmas Tree Growth," *Forest Science*, 36, 45-53.

Homer, C. H., Ramsey, R. D., Edwards, T. C., Jr., and Falconer, A. (1997), "Landscape Cover-Type Modeling Using a Multi-Scene TM Mosaic," *Photogrammetric Engineering and Remote Sensing*, 63, 221-231.

Kannellopoulos, I. and Wilkinson, G. G. (1997), "Strategies and Best Practice for Neural Network Image Classification," *International Journal of Remote Sensing*, 18, 711-725.

Lek-Ann, S., Deharveng, L., and Lek, S. (1999), "Predictive Models of Collembolan Diversity and Abundance in a Riparian Habitat," *Ecological Modeling*, 120, 247-260.

Levien, L. M., Roffers, P., Maurizi, B., Suero, J., Fischer, C., and Huang, X. (1999), "Change Detection Using Neural Networks," in *1999 Annual Conference Proceedings of the American Society of Photogrammetry and Remote Sensing*, eds. T. J. Smith and R. Edwards, Portland, Oregon, June 1999, pp 224-237.

Lewis, H. G., Cote, S., and Tatnall, A. R. L. (1997), "Determination of Spatial and Temporal Characteristics as an Aid to Neural Network Cloud Classification," *International Journal of Remote Sensing*, 18, 899–915.

Manel, S., Dias, J., and Omerod, S. J. (1999), "Comparing Discriminant Analysis, Neural Networks and Logistic Regression for Predicting Species Distributions: A Case Study with a Himalayan River Bird," *Ecological Modeling*, 120, 337-347.

McCullagh, P., and Nelder, J. A. (1989), *Generalized Linear Models*, New York: Chapman and Hall.

Michaelson, J., Davis, F., and Borchert, M. (1987), "A Nonparametric Method for Analyzing Hierarchical Relationships in Ecological Data," *Coenoses*, 2, 39-48.

Michaelson, J., Davis, F., and Bochert, M. (1987), "A Nonparametric Method for Analyzing Hierarchical Relationships in Ecological Data," *Coenoses*, 2, 39-48.

Moatar, F., Fessant, F., and Poirel, A. (1999), "pH Modeling by Neural Networks: Application of Control and Validation Data Series in the Middle Loire River," *Ecological Modeling*, 120, 141-156.

Moisen, G. G., and Edwards, T. C., Jr. (1999), "Use of Generalized Linear Models and Digital Data in a Forest Inventory of Northern Utah," *Journal of Agricultural, Biological, and Environmental Statistics*, 4, 372-390.

Moisen, G. G., and Edwards, T. C., Jr., and Cutler, R. D. (1996), "Estimation, Exploration, and Mapping of Forest Attributes Through Extensions to Generalized Additive Models," in *Proceedings of the Meeting on Modeling Longitudinal and Spatially Correlated Data*, Nantucket, MA, 1996.

Morgan, J. N., and Sonquist, J. A. (1963), "Problems in the Analysis of Survey Data and a Proposal," *Journal of the American Statistical Association*, 58, 415-434.

Nelder, J. A., Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society A*, 135, 370-384.

Norton, T. W., and Mitchell, N. D. (1993), "Application of Generalized Additive Models to Wildlife Modeling and Population Viability Analysis," in *Proceedings of the 7th Annual Symposium on Geographic Information Systems in Forestry, Environments and Natural Resources*, Vancouver, Canada, 1993.

Paoloa, J. D., and Schowengerdt, R. A. (1997), "The Effect of Neural Network Structure on a Multispectral Land-Use/Land-Cover Classification, " *Photogrammetric Engineering and Remote Sensing*, 63, 535-544.

Paruelo, J. M., and Tomasel, F. (1997), "Prediction of Functional Characteristics of Ecosystems: A Comparison of Artificial Neural Networks and Regression Models," *Ecological Modeling*, 98, 173-186.

Ripley, B. D. (1994), "Neural Networks and Related Methods for Classification," *Journal of the Royal Statistical Society B*, 56(3), 409-456.

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, New York: Cambridge University Press.

Roberts. D. W., and Cooper, S. V. (1989), "Concepts and Techniques of Vegetation Mapping," in *Land Classifications Based on Vegetation: Applications for Resource Management*, eds. D. Ferguson, P. Morgan, and F. D. Johnson, General Technical Report INT-257, US Department of Agriculture, Forest Service, Intermountain Research Station, Ogden, UT, pp. 90-96.

Scardi, M. and Harding, L. W., Jr. (1999), "Developing an Empirical Model of Phytoplankton Primary Production: A Neural Network Case Study," *Ecological Modeling*, 120, 213-223.

Scott, J. M., Davis, F., Csuti, B., Noss, R., Butterfield, B., Caicco, S., Groves, C., Edwards, T. C., Jr., Ulliman, J., Anderson, H., DErchia, F., and Wright, R. G. (1993), "Gap Analysis: A Geographic Approach to Protection of Biological Diversity," *Wildl. Monogr. No. 123.*

Skidmore, A. K., Turner, B. J., Brinkhof, W., and Knowls, W. (1997), "Performance of a Neural Network: Mapping Forests Using GIS and Remotely Sensed Data," *Photogrammetric Engineering and Remote Sensing*, 63, 501-514.

Stenback, J. M., and Congalton, R. G. (1990), "Using Thematic Mapper Imagery to Examine Forest Understory," *Photogrammetric Engineering and Remote Sensing,* 56, 1285-1290.

Stern, H. S. (1996), "Neural Networks in Applied Statistics," *Technometrics*, 38, 205-220.

Van Wijk, M. T., and Bouten, W. (1999), "Water and Carbon Fluxes above European Coniferous Forests Modeled with Artificial Neural Networks," *Ecological Modeling,* 120, 181-197.

Vogelmann, J. E., Sohl, T., and Howard, S. M. (1998), "Regional characterization of land cover using multiple sources of data," *Photogrammetric Engineering and Remote Sensing*, 64, 45-57.

Wang, Y., and Dong, D. (1997), "Retrieving Forest Stand Parameters from SAR Backscatter Data Using a Neural Network Trained by a Canopy Backscatter Model," *International Journal of Remote Sensing*, 18, 981-990.

Wythoff, B. J. (1993), "Backpropagation Neural Networks: A Tutorial," *Chemometrics and Intelligent Laboratory Systems,* 18, 115-155.

Yee, T. W., and Mitchell, N. D. (1991), "Generalized Additive Models in Plant Ecology," *Journal of Vegetation Science*, 2, 587-602.

Zhang, X., and Yuan, Y. (1997), "Application of Neural Networks to Identifying Vegetation Types from Satellite Images," *AI Applications*, 11, 99–106.

APPENDICES

Appendix A: Data Extraction Procedure

Appendix A-1. Frequently used commands in ArcInfo.

```
## list grids
   eg: lg

## describe grid spec
   eg: describe usa_grid_alb

## list elements in a coverage
   eg: list <coverage>.pat

## delete a grid or coverage
   eg: kill mt2_lz all

## convert grid or coverage to a new projection
   eg: project grid usa_grid_alb usa_grid_lamaz
       gis/projections/utm122lamaz.prj
   eg: project cover az1 az1_lz autm122lamaz.prj

## copy from one coverage to another
   eg: copy st_boundary mtbnd

## build poygon coverage after projection
   eg: build spatial/data/coverages/bound/azbnd_lz poly

## clip grid with boundary and make sep grid file
   Usage: LATTICECLIP <in_lattice> <clip_cover>
            <out_lattice> {MINIMUM|EXTENT}{z_factor}
   eg: latticeclip /fsfiles/unit/fia/spatial/data/factory/usa_grd_lz
            /fsfiles/unit/fia/spatial/data/coverages/bound/azbnd_lz
             azdem1000_lz

## extract point values from grid and add to point cover
   Usage: LATTICESPOT <in_lattice> <in_cover> {spot_item} {z_factor}
   eg: latticespot ../dma/azdem1000_u12 az2_fctl dem1000

## Arc: identity
   Usage: IDENTITY <in_cover> <identity_cover> <out_cover>
                 {POLY | LINE | POINT}
                 {fuzzy_tolerance} {JOIN | NOJOIN}

## display a grid
   ap
   Arcplot: display 9999
   Arcplot: mape utdem1000_lz
   Arcplot: image utdem1000_lz
   q

## generate slope and aspect from elevation grid
   grid
   Grid: ../dma/azslp1000_u12 = slope(../dma/azdem1000_u12,degree)
   Grid: ../dma/azasp1000_u12 = aspect(../dma/azdem1000_u12)
   q
```

```
##make ascii file from a coverage w/ plotid,variables (comma delim)
   tables
   tables: select <coverage>.pat
   UNLOAD <outfile> {$recno,item1,...,itemn..no bracks}
          {DELIMIted | columnar <format_file>}
   q stop

## join item
   Usage: JOINITEM <in_info_file> <join_info_file> <out_info_file>
                   <relate_item>
                {start_item} {LINEAR | ORDERED | LINK}

## change name of attribute in coverage
   tables
   tables: select <coverage>.pat
   tables:items or list
   tables: alter
     itemname: <newname>
     <return what left unchanged>
   q stop

## extract values from multiple gridsat points using various
   interpolation options (or none for discrete grids)
   Grid: sample
   Usage:  (T) SAMPLE (<mask>, {grid, ..., grid})
        (T) SAMPLE (<* | point_file>, {grid, ..., grid},
                    {NEAREST | BILINEAR | CUBIC})

        eg: Grid: ../../../ascii/ut2/map1000/ut2_gap1k.dat =
           sample(../../../ascii/ut2/map1000/ut2_1k.txt,
                  ut2elv90_u12,ut2asp90_u12,ut2slp90_u12,
                  ut2tm_b3_u12,ut2tm_b4_u12,
                  ut2tm_b5_u12,utgap90_u12,nearest)

           (from /spatial/data/ascii)
        transfer/ut2_nlcd5k.dat =
           sample(ut2/responses/ut2_xy5_u12.dat,../factory/nlcd/ut_nlc
           d_u12,nearest)
        transfer/mt2_nlcd5k.dat =
           sample(mt2/responses/mt2_xy5_u12.dat,../factory/nlcd/mt_nlc
           d_u12,nearest)
        transfer/ut1_nlcd1k.dat =
           sample(ut1/map1000/ut1_1k.txt,../factory/nlcd/ut_nlcd_u12,n
           earest)


## generate an intensive grid for mapping then sample
## from other grids to create an ascii file

   LATTICECLIP inlattice clipcover outlattice
   outgrid=RESAMPLE(grid,cellsize)
   outascii=SAMPLE(mask_grid,grid,grid,...,grid)
   ex:latticeclip dma/utdem1000_u12 bound/ut2_u12 dma/ut2dem1k_u12
      grid
        dma/ut2dem100_u12 = resample(dma/ut2dem1k_u12,100,nearest)
        dma/ut2dem500_u12 = resample(dma/ut2dem1k_u12,500,nearest)
                 ../ascii/map.pts/ut2samp500.txt =
            sample(dma/ut2dem500_u12,dma/ut2dem1k_u12)

## project to new projection
  project cover mtbnd_u12 mtbnd_lam
         /fsfiles/unit/fia/gis/projections/utm122lam.prj
  build   mtbnd_lam poly


####################################
# extract data from usa 1000m dems #
####################################

0. describe grid to get correct projection parameters
1. project spatial/data/factory/usa_grd_lz in lamaz
2. project state boundaries in lamaz
3. rebuild boundaries as polygon coverages
4. latticeclip AZ, MT, and UT from demgrid
    and project as u12
5. create slope and aspect grids for each of 3 states
6. latticespot ecoregion point coverages to extract elv1000,
     slp1000, and asp1000
```

Appendix A-2. SAS program to extract plot and tree level variables from SAS data sets from plots given file of UTM coordinates.
(Written by Shirley Waters.)

```
   %Let STATE=arizona;
    %Let SA=all;
    Titlel Arizona;

    *Filename gretchen_az.sas;
    * SAS program to create a file of variables and merge with
    * a file created by gretchen;
    %Let CALPATH2 =%STR(/calcul);
   %Let CALPATH1 =%STR(/fsfiles/unit/fia/data/prelimdat/);
   %Let CALCDIR  =&CALPATH1&STATE&CALPATH2;
   %Let DIRSEP=/;    * Directory separator;
   Libname LIBRARY
             "/fsfiles/unit/fia/data/prelimdat/&STATE/formats";

   Libname TAB  "/fsfiles/unit/fia/data/prelimdat/&STATE/calcul";

   Libname DAT  "/fsfiles/unit/fia/data/prelimdat/&STATE/field";

           Options Linesize=200;
   *OPTIONS OBS=20;

   Data _NULL_; File '?Footnote';
       Length RUNDATE $8;
       RUNDATE = Put(DATE(),MMDDYY8.);
       Put "Footnote H=.9 J=R

'/fsfiles/unit/fia/data/prelimdat/arizona/calcul/
            gretchen_az.sas--"RUNDATE"';";
   Run;


   %Include '?Footnote';
   Footnote2 H=.1;

   Data PLOTS(Drop=DIA TRHIS BA NGRWBA BAACC NVOLTOT NVOLMER
            NGRWCF BIOTOT TPALOC);
     Set TAB.TOTAL&SA(Keep=SA PLOTID CO LOC GRDCOV LNDUSE OWNER
             FOREST DAY
     MONTH YEAR A_FORTYP A_STSZCL CRCOV ELEV PHYSCL QMD ASPECTAZ
             SLOPE
     CURVECL BAACC MAICF A_STAGE SAMPKND CNDPROP LNDUSE GRDCOV
             A_FORTYP
     DIA TRHIS TPALOC BA BIOTOT NVOLTOT NVOLMER NGRWCF NGRWBA);
     Retain STPALOC SBA SBIOTOT SNVOLTOT SNVOLMER SNGRWCF
            SNGRWBA 0;
     By PLOTID;
     If CNDPROP Ge 1;
     If FIRST.PLOTID Then Do;
        STPALOC = 0;
        SBA = 0;
```

```
      SBIOTOT = 0;
      SNVOLTOT = 0;
      SNVOLMER = 0;
      SNGRWCF = 0;
      SNGRWBA = 0;
   End;
   If TRHIS Eq 1 And DIA Ge 1 Then Do;
      STPALOC = STPALOC + TPALOC;
      SBA = SBA + BA;
      SBIOTOT = SBIOTOT + BIOTOT;
      SNVOLTOT = SNVOLTOT + NVOLTOT;
   End;
   If TRHIS Eq 1 And DIA Ge 5 Then Do;
      SNVOLMER = SNVOLMER + NVOLMER;
      SNGRWCF = SNGRWCF + NGRWCF;
      SNGRWBA = SNGRWBA + NGRWBA;
   End;
 If LAST.PLOTID Then Output PLOTS;
 Run;

 Proc Sort Data=PLOTS;
    By CO LOC;
 Run;

 Data GRET1;
   Filename GRET1
     "/fsfiles/unit/fia/gis/factory/anal/az1_fldplts.txt";
   Infile GRET1 DLM=',';
   Input CO LOC UTME UTMN;
 Run;

  Data GRET2;
   Filename GRET2
    "/fsfiles/unit/fia/gis/factory/anal/az2_fldplts.txt";
   Infile GRET2 DLM=',';
   Input CO LOC UTME UTMN;
  Run;

Proc Sort Data=GRET1;
   By CO LOC;
Run;

Proc Sort Data=GRET2;
   By CO LOC;
Run;

 Data VARS;
    Merge DAT.CTRL(Keep=CO LOC GLU GRIDZONE LONGTUDE LATITUDE
           PLOTID GRIDZONE EASTING NORTHING)
          DAT.COND(Keep=PLOTID CNDPROP PCTBARE SZCND)
          DAT.LOC(Keep=PLOTID SZFOR RSCOV1);
    By PLOTID;
    If CNDPROP Ne 1 Then Delete;
    If GLU Ge 96 Then Delete;
 Run;
 Proc Sort Data=VARS;
    By CO LOC;
 Run;

 Data PLOTS1; Merge PLOTS GRET1(In=A) VARS(In=B);
     If A and B;
     By CO LOC;
     If GLU Ge 96 Then Delete;
 Run;

 Data PLOTS2; Merge PLOTS GRET2(In=A) VARS(In=B);
     If A and B;
     By CO LOC;
     If GLU Ge 96 Then Delete;
 Run;

Data _NULL_; Set PLOTS1 END=LAST;
   File
    "/fsfiles/unit/fia/data/prelimdat/arizona/calcul/
          az1_fldplts_sum.txt" ;
   Put (ASPECTAZ A_FORTYP A_STAGE A_STSZCL CNDPROP CO CRCOV
           CURVECL DAY
    EASTING ELEV FOREST GLU GRDCOV GRIDZONE LATITUDE LNDUSE
          LOC LONGTUDE
    MAICF MONTH NORTHING OWNER PHYSCL PLOTID QMD SA
          SAMPKND SLOPE UTME UTMN YEAR)
    (:10. + (-1)',') +(-1) ','
    (SBA SBIOTOT SNGRWBA SNGRWCF SNVOLMER SNVOLTOT STPALOC)
           (:10.2 + (-1)',');
   If LAST Then Put 'END';
 Run;

 Data _NULL_; Set PLOTS2 END=LAST;
   File
          "/fsfiles/unit/fia/data/prelimdat/arizona/calcul/
          az2_fldplts_sum.txt";
   Put (ASPECTAZ A_FORTYP A_STAGE A_STSZCL CNDPROP CO CRCOV
          CURVECL DAY
    EASTING ELEV FOREST GLU GRDCOV GRIDZONE LATITUDE LNDUSE
          LOC LONGTUDE
    MAICF MONTH NORTHING OWNER PHYSCL PLOTID QMD SA SAMPKND
           SLOPE UTME UTMN YEAR)
    (:10. + (-1)',') +(-1) ','
    (SBA SBIOTOT SNGRWBA SNGRWCF SNVOLMER SNVOLTOT STPALOC)
           (:10.2 + (-1)',');
 Run;

 Proc Contents Data=PLOTS1;
 Run;

 Options Obs = 50;

 Proc Print Data=PLOTS1;
 title1  first 50 records of
```

```
   /fsfiles/unit/fia/data/prelimdat/arizona/calcul/
   az2_fldplts_sum.txt;
title2  warning:  Arizona has 2 gridzones;
run;

Proc print Data=PLOTS2;
title1  first 50 records of
         /fsfiles/unit/fia/data/prelimdat/arizona/
         calcul/az2_fldplts_sum.txt;
title2  warning:  Arizona has 2 gridzones;
run;
```

Appendix A-3. AML to generate a point coverage with attributes from Oracle tables (Written by Ron Tymcio and Tracey Frescino)

```
/*####################################################################
/*
/* TITLE:  ORACLE_LLGEN.AML
/* DESCRIPTION:  To generate a point coverage using latitude/longitude
/*                coordinates and plotid
/*                data that resides in Oracle with user-specified
/*                attributes.
/*
/* Written by Ron Tymcio 10/3/95.
/* Seriously modified by Tracey Frescino 6/28/99
/*  Then re-seriously modified by Ron Tymcio 2/26/00.
/*          - Modified to use Geographic Coord.
/*
/*
/* INPUTS:         .table     Oracle user.table that has coordinate
/*           info to build coverage
/*             .cover   Output coverage name

/*               .where Oracle WHERE clause specifying boundary
/*                criteria

/*             .more    Additional variables, if desired

/*             .out           Output filename for ASCII info, if
/*                desired
/*
/* OUTPUTS:    A point coverage including additional variables
/*                if specified
/*             A comma delimited ASCII file if asked for.
/*             (The point coverage is projected to UTM zone 12)
/*
/*####################################################################


/*
/* Set variables
/*
&args .table .where .cover newsel1 newsel2 newsel3  .more atts atts2
.more2 outatts .out


/**********************************************************************
*********************
/* Begin - prompt for Oracle user.tablename
/**********************************************************************
*********************
&do &while [null %.table%]
  &setvar .table [response 'Enter the Oracle user.table that has
coordinate information - fia.rllocs']
&end
/*&do &while [null %.where%]
```

```
  &type Enter WHERE clause to delineate coverage boundary
  &setvar .where [response '(Include WHERE in statement - ex. where
forest = 3)']
/*&end
/*
/* Prompt for point coverage name.
/*
&do &while [null %.cover%]
  &setvar .cover [response 'Enter the name of ouput coverage']
&end

/**********************************************************************
*********************
/* Erase any existing INFO tables or ASCII files
/**********************************************************************
*********************
rm tempcoord.dat
rm newtemp1.dat
rm newtemp2.dat
rm newtemp3.dat
&data arc tables
kill tempcoord
kill tempatts
kill tempatts2
kill newtemp1
kill newtemp2
kill newtemp3
q stop;
&end

/**********************************************************************
*********************
/* Set Oracle SELECT statement using inputted table and where clause
info.
/**********************************************************************
*********************
&if [NULL %.where%] &then
  &setvar .select := [quote SELECT plotid, longtude*-1 longtude,
latitude FROM %.table%]
&if not [NULL %.where%] &then
  &setvar .select := [quote SELECT plotid, longtude*-1 longtude,
latitude FROM %.table% [unquote %.where%]]

    /*********************************************************
    /* Connect to Oracle and define variables for INFO
    /*********************************************************
    &data arc
    connect oracle fia/rre
    dbmsinfo oracle %.select% tempcoord define
      plotid %.cover%-id 8 8 I;
      longtude longtude 11 11 N 6;
      latitude latitude 10 10 N 6;
      end;
    tables
      sel tempcoord;
      reselect longtude = 0;
      purge;y;
      sel tempcoord;
      unload tempcoord.dat %.cover%-id, longtude, latitude, delimited;
      q stop;

    /**********************************************************
    /* Kill old coverage if it exists and create new coverage
    /**********************************************************
    &if [exists %.cover% -cover] &then kill %.cover% all

      generate tempcover
      input tempcoord.dat
      point
      q

    build tempcover point
    /*quit
    /*&end

    /**********************************************************
    /* Project cover to UTM 12
    /**********************************************************

    project cover tempcover %.cover%
/fsfiles/unit/fia/gis/projections/dd2utm12.prj
    build %.cover% point
    kill tempcover all
    quit
    &end

/***********************************************************************
*********************
/* Add the plotid attribute to the point coverage by adding an
additional column
/* called plotid and copying the cover-id values to the plotid column.
/***********************************************************************
*********************

&data arc
  tables
  additem %.cover%.pat plotid 10 10 I
  sel %.cover%.pat
  calc plotid = %.cover%-id
  q stop;
        &if [exists temp1 -cover] &then kill temp1 all
        &if [exists temp2 -cover] &then kill temp2 all
        &if [exists temp3 -cover] &then kill temp3 all
        &if [exists temp1prj -cover] &then kill temp1prj all
        &if [exists temp3prj -cover] &then kill temp3prj all
  quit
&end

/**********************************************************************
/* To add additional attributes to point coverage
```

```
/***********************************************************************
  &do &while [null %atts%]
    &sv atts := [response 'Do you want to add additional attributes to
the point coverage (y/n)? ']
  &end
&if %atts% EQ 'n' OR %atts% EQ 'N' &then
    &return &inform Bye
&if %atts% EQ 'y' OR %atts% EQ 'Y' &then
  &do
    /*
    /* Prompt for added attribute variables
    /*
    &do &while [null %.more%]
      &setvar .more [response 'Enter variables (up to 12) seperated
with commas (ex. fortyp,lba,...)']
    &end

    /**********************************************************
    /* To pull the data from Oracle
    /**********************************************************
    &if [NULL %.where%] &then
      &setvar .select2 := [quote select plotid, [unquote %.more%] from
%.table%]
    &if not [NULL %.where%] &then
      &setvar .select2 := [quote select plotid, [unquote %.more%] from
%.table% [unquote %.where%]]

    &data arc
      connect oracle
      dbmsinfo oracle %.select2% tempatts define
        plotid %.cover%-id 10 10 I;
        end;
      quit
    &end

    /**********************************************************
    /* To extract variable names and change types from binary to
integer
    /**********************************************************
    &data arc tables
    &sv var := one
    &sv left := two
    &sv last := 0
/*    sel tempatts
/*    alter %.cover%-id
/*    ,15,,,,,
      &do &until %last% = 99
        &sv var := [before %.more% ,]
        &if %var% EQ %.more% &then
          &sv last = 99
        &else
          &sv .more := [after %.more% ,]
          sel tempatts
          alter %var%
          %var%;;
          B;0;;
      &end
      q stop
      quit
    &end

    /*********************************************************
    /* To join attributes to the point coverage attribute table
    /*********************************************************
    joinitem %.cover%.pat tempatts %.cover%.pat %.cover%-id plotid
    /*
    /*
    &do &while [null %atts2%]
      &sv atts2 := [response 'Do you want to add any more attributes to
the point coverage (y/n)? ']
    &end
    &if %atts2% EQ 'y' OR %atts2% EQ 'Y' &then
      &do
        &do &while [null %.more2%]
          &setvar .more2 [response 'Enter variables (up to 12)
seperated with commas (ex. fortyp,lba,...)']
        &end
        &setvar .select3 := [quote select plotid, [unquote %.more2%]
from %.table% [unquote %.where%]]
        &data arc
          connect oracle
          dbmsinfo oracle %.select3% tempatts2 define
            plotid plotid 10 10 I;
            end;
          quit
        &end

        /*********************************************************
        /* To extract variable names and change type from binary to
integer
        /*********************************************************
        &data arc tables
        &sv var := one
        &sv left := two
        &sv last := 0
          &do &until %last% = 99
            &sv var := [before %.more2% ,]
            &if %var% EQ %.more2% &then
              &sv last = 99
            &else
              &sv .more2 := [after %.more2% ,]
              sel tempatts2
              alter %var%
              %var%;;
              B;0;;
          &end
          q stop
          &end
          /*
          &data arc
```

```
            joinitem %.cover%.pat tempatts2 %.cover%.pat plotid
            quit
          &end
      &end
  &end


/*********************************************************************
    /* Build the final coverage

/*********************************************************************
    build %.cover% point


/*********************************************************************
    /* To unload the coverage attribute table to a comma delimited
ASCII file

/*********************************************************************
    &do &while [null %outatts%]
      &sv outatts := [response 'Do you want an ASCII file of the
coverage attribute table (y/n)? ']
    &end
    &if %outatts% EQ 'n' OR %outatts% EQ 'N' &then
      &return &inform Bye
    &if %outatts% EQ 'y' OR %outatts% EQ 'Y' &then
      &do

        /*********************************************************
        /* Prompt for output filename
        /*********************************************************
        &do &while [null %.out%]
          &setvar .out [response 'Enter name for output ASCII file']
        &end
          &if [exists %.out%] &then
            &do
            &sv opt := [response 'The file exists. Do you want to
overwrite it (y/n)? ']
                &if %opt% EQ 'n' OR %outatts% EQ 'N' &then
                  &setvar .out [response 'Enter name for output ASCII
file']
                &else
                  &sys rm %.out%
            &end
        &data arc tables
          sel %.cover%.pat
          alter %.cover%-id
          ,15,,,,,
          unload %.out%, delimited;
          q stop;
          quit
        &end
      &end
      &else
        &return &error Must enter y or n
  &end
  &else
    &return &error Must enter y or n
/*
/* END
```

Appendix A-4. ArcView project with multiple scripts for extractin information from Oracle
(Written by Tracey Frescino)

ARCVIEW PROJECT: ORA_EXPLORE.APR

What you can do:

▣ Generate a point shape file/theme from Oracle.

⇧ Project a theme and add it to a projected view.

◆ Add attributes to an existing or generated point theme by exploring the Oracle database.

⁘ Exports a theme's attribute table to an Arc shapefile, a comma-delimited ASCII file, or an INFO file.

A Generate an ASCII file from selected Oracle data.

▣ To generate a point shape file/theme form Oracle referenced in decimal degrees (No projection).

(1) **ORACLE TABLE**: Select table where coordinates are stored.

RILOCS	Region 1 NFS
UTLOCS	Utah (all owners)
MTLOCS_ALL	Montana (all owners)

(2) **WHERE**: Do you want to subset the location data with a where clause?
If yes: Select criteria variable (ex.) glu
Select one or more value (ex.) 20

(3) **Convert to shapefile:**
File Name: theme?.shp in /fsfiles/unit/fia/gis/av_proj/montana/anal/temp/
(DEFAULT)

(Change name keeping .shp extension. You can overwrite an existing shapefile
by selecting it
from the list or typing in the same name.)

The theme (shapefile) will appear in View. Click in box to display.

To PROJECT theme and add it to a PROJECTED view.

Note: You must know projection of the view (or themes in view) (ex.) UTM - 1927; Zone 12

(1) Select theme to project (make active)
(2) Click on icon

(3) **Projector!** Pick output units (ex.) meters UTM, ALBERS
feet STATE PLANE

(4) **Projection Properties:**
Category: Projections of the World - Change to coordinate system (ex.
UTM - 1927)
(check Zone if UTM)
Recalculate area, perimeter, ... using meters? - YES
Add projected shapefile(s) as theme(s) to a View? - YES

(5) **Projector! Add Theme to:** - Select View from list to add projected theme to

(6) **Project:**
File Name: theme?.shp in /fsfiles/unit/fia/gis/av_proj/montana/anal/temp/
(DEFAULT)
(Change name keeping .shp extension. You can overwrite an existing shapefile
by selecting it
from the list or typing in the same name.)

The theme (shapefile) will appear in View. Click in box to display.

To add attributes to an existing or generated point theme.

(1) **LOCATION/TREE DATA:** Select type of data (location or tree).
Location - plot level data
Tree - summed tree level data

(2) **ORACLE TABLE:** Select table(s) where data are found

(3) **WHERE:** Do you want to subset location data with a where clause?
If yes: Select table where criteria variable is found.
Select criteria variable (ex. owner)
Select one or more values. (ex. 11)

Do you want to add anothere where clause?

Note: Any plotid from the selected Oracle table that matches the theme plotid will be joined.

(4) **TABLE VARIABLES:** Do you want to select any variables from the selected table.
If yes: Select variables from list
Do you want to select any more variables from the selected table?

Note: If selecting tree data:
GROUP FUNCTION: Select a group function (SUM, AVG, MAX, MIN, COUNT)
Do you want to subset the tree data with a tree where clause?

(5) **SQL STATEMENT:** Yes/No

(6) **JOIN:** Do you want to join data to a theme?.
If yes: Select theme from list

To export a theme attribute table to an Arc shapefile, a comma-delimited ASCII file, or an INFO file.

(1) **THEME ATTRIBUTE TABLE:** Select the theme attribute table you want to export.

(2) **EXPORT:** Export format: (SHAPE, INFO, Delimited Text)

(3) **EXPORT:**
File Name: theme?.ext in /fsfiles/unit/fia/gis/av_proj/montana/anal/temp/ (DEFAULT)
(Change name keeping the extension. You can overwrite an existing file by selecting it from the list or typing in the same name.)

The theme (shapefile) will appear in View. Click in box to display.

A To generate a comma-delimited ASCII file (only) of selected Oracle data.

(1) **LOCATION/TREE DATA:** Select type of data (location or tree).
Location - plot level data
Tree - summed tree level data

(2) **ORACLE TABLE:** Select table(s) where data are found

(3) **WHERE:** Do you wnat to subset location data with a where clause?
If yes: Select table where criteria variable is found.
Select criteria variable (ex.) owner
Select one or more values. (ex.) 11

Do you want to add anothere where clause?

Note: Any plotid from the selected Oracle table that matches the theme plotid will be joined.

(4) **TABLE VARIABLES:** Do you want to select any variables from selected table.
If yes: Select variables from list
Do you want to select any more variables from selected table?

Note: If selecting tree data:
GROUP FUNCTION: Select a group function for variable (SUM, AVG, MAX, MIN, COUNT)
Do you want to subset the tree data with a tree where clause?

(5) **SQL STATEMENT:** Yes/No

(6) **ASCII FILE**: Do you want an ASCII file of the attribute data?
If yes: Select fields you want to export to ASCII.

(7) **EXPORT TO ASCII:**
File Name: theme?.txt in /fsfiles/unit/fia/gis/av_proj/montana/anal/temp/ (DEFAULT)
(Change name keeping .txt extension. You can overwrite an existing shapefile by selecting it from the list or typing in the same name.)

The theme (shapefile) will appear in View. Click in box to display.

To remove any fields joined to the attribute table of selected theme(s).

(1) **UNJOIN**: Select theme(s) to remove joined fields.

To delete tables from the Project.

(1) If there are any tables in Project
DELETE: Are you sure you want to delete <table name>?

Appendix A-5. Instructions for loading and viewing national 2-week composite AVHRR data

Process:

1. Read the Readme.1st file on the CD to determine the location of the image data files on the CD and to get general information on the AVHRR data sets.
2. From Imagine toolbar, go to MAIN—IMPORT/EXPORT—
   Type: Generic Binary
   Media: cd-rom or file
   Specify input and output files
   Data format: bsq
   Data type: unsigned 8-bit
   Number of rows: 2889
   Number of columns: 4587
   Number of bands: 6
   Select bands in multiple files option then select each image file you want and what band designation it will have
   All other prompts are left at their (0) default settings
3. From Viewer, go to FILE—OPEN—RASTER to view image
4. From Viewer, go to UTILITY—LAYER/INFO, then EDIT—Change map model
   Upper left X: -2050000
   Upper left Y: 752000
   Pixel size X: 1000
   Pixel size Y: 1000
   Units: meters
   Projection: Lambert-Azimuth – equal area
   Indicate yes to change for all bands then EDIT—PROJECTION
   Long of center: -100
   Lat of center: 45
   Apply to all layers

Next, to create statewide grids for each band, see:
"Using and Arc coverage to subset and image and generate grids"

Appendix A-6. Instructions for using an Arc coverage to subset an image in Imagine. (Written by Tracey Frescino.)

######### USING AN ARC COVERAGE TO SUBSET AN IMAGE IN IMAGINE ##########

Open the image to be subsetted:

(1) From Viewer, go to **FILE--OPEN--RASTER LAYER...**

(2) Enter image filename and click on **RASTER OPTIONS** to set view extent (Fit to Frame)

CREATE AN AREA OF INTEREST (AOI) LAYER

## Open the AOI subset file:

(3) First, set up a new AOI layer: **FILE--NEW--AOI LAYER...**

(4) Then, open the subset coverage (vector layer): **FILE--OPEN--VECTOR LAYER...**

Note: the subset coverage must be an Arc coverage having polygon topology and
having compatible projections (To check projections, See Page 2)

## Change vector layer to polygon layer:

Note: As a default, Imagine loads an ARC coverage as a vector (line) layer and not
as a polygon layer. This step turns on the polygon topology

(5) From Viewer, go to **VECTOR--VIEWING PROPERTIES...**

(6) Click on the **POLYGON** button and **APPLY**

(7) Close window (No need to save)

## Select AOI polygon/polygons and add them to the AOI layer:

(8) Use the mouse to click inside the polygon and select it. If you want to select
more than one polygon, hold the SHIFT key while clicking on each polygon.

(9) Then, go to **AOI--COPY SELECTION TO AOI...**

Note: If you have additional polygons you want to include, use the mouse button again to select each AOI polygon (again,using the shift key on the keyboard for multiple polygons). Once they are all selected, they must be grouped together using **AOI--GROUP**

## Save AOI to a file:

(10) From Viewer, go to **FILE--SAVE--SAVE AOI AS...**

<u>Note</u>: ****DO NOT** click on Selected Only box

<u>SUBSET IMAGE USING AOI</u>

## Set subset boundary:

(11) Click on the AOI in the Viewer (This should draw a boundary box around the AOI)

(12) Click right mouse button and select **INQUIRE BOX...** from menu

(13) In window, click on **FIT TO AOI** and **APPLY** (This should adjust coordinates to AOI)

<u>Note</u>: Keep window open

## Subset image:

(14) From the main menu, go to **INTERPRETER--UTILITIES--SUBSET**

(15) Specify your input (.img) and output (.img) files.

(16) Click on **FROM INQUIRE BOX** (Located right of Coordinate Type: Subset Definition)

(This should change the Map coordinates to match the coordinates from the INQUIRE BOX window above. If the coordinates do not match, either try to set boundary again or manually type in the correct coordinates from the INQUIRE BOX window.)

(17) Click on the **AOI** button at the bottom of the window. In the window that pops up, select **FILE** and specify the AOI file that you saved. Then click on **OK.**

(18) Click on: Ignore Zero Output Stats.

(19) Click on **OK**

CONVERT IMAGE TO GRID IN IMAGINE

(1) From main menu, go to **IMPORT**

(2) In window, click on **Export**

(3) Change Type: **GRID**
Media: **FILE**

(4) Enter Input File and Output File names

<u>Note</u>: When you enter an Input File name, the Output File name will default to the
same name having a .grid extension instead of a .img extension (and in the
same directory. You can change the name and directory if desired.

(5) Click **OK**

**Another window should appear titled: Export GRID Data containing info on
the number of layers and rows and columns

If you want to exclude certain layers, click on **EXPORT OPTIONS...**

(6) Click **OK** again

PROJECTIONS IN IMAGINE

## Check arrangement of layers (The layer you want to check projections must be on top)

(1) From Viewer, **VIEW--ARRANGE LAYERS...**

If the layer you are interested in is not on top, click on the layer and click **UP**
until it is on top. Then click **APPLY**. Take note of the arrangement because you
may want to switch the layers back to the way they were.

## Check projection info

(2) From Viewer, go to **UTILITY--LAYER INFO...**

Take note of: Map Info: the units
Projection Info: everything

Note: If you are comparing projections to see if coverages are compatible, keep
this window open and bring up the Layer Info... window from the other
coverage using the same process.

Appendix A-7. Example of extracting predictor variables at 5km and 1km grid points in Montana

```
#######################################
## start arc grid from ascii directory

cd /spatial/data/ascii
arc
grid

## sample 1 and 5k grids from nlcd coverage and put
## in file in transfer directory

transfer/mt1_nlcd1k.dat = sample(mt1/map1000/mt1_1k.txt,
     ../factory/nlcd/mt_nlcd_u12,nearest)
transfer/mt1_nlcd5k.dat = sample(mt1/responses/mt1_xy5_u12.dat,
     ../factory/nlcd/mt_nlcd_u12,nearest)
transfer/mt2_nlcd1k.dat = sample(mt2/map1000/mt2_1k.txt,
     ../factory/nlcd/mt_nlcd_u12,nearest)
transfer/mt2_nlcd5k.dat = sample(mt2/responses/mt2_xy5_u12.dat,
     ../factory/nlcd/mt_nlcd_u12,nearest)

## sample 1 and 5k grids from dma and avhrr coverages and put
## in file in transfer directory (I hope the long paths on the sample
coverages
## don't screww it up...normally I run wthin the coverage directory
itself..)

transfer/mt1_avh1k.dat = sample(mt1/map1000/mt1_1k.txt,
     ../factory2/predictors/montana/mtdem1000_u12,
     ../factory2/predictors/montana/mtasp1000_u12,
     ../factory2/predictors/montana/mtslp1000_u12,
     ../factory2/predictors/montana/mt_avh1_u12,
     ../factory2/predictors/montana/mt_avh2_u12,
     ../factory2/predictors/montana/mt_avh3_u12,
     ../factory2/predictors/montana/mt_avh4_u12,
     ../factory2/predictors/montana/mt_avh5_u12,
     ../factory2/predictors/montana/mt_andvi_u12,
     nearest)

transfer/mt1_avh5k.dat = sample(mt1/responses/mt1_xy5_u12.dat,
     ../factory2/predictors/montana/mtdem1000_u12,
     ../factory2/predictors/montana/mtasp1000_u12,
     ../factory2/predictors/montana/mtslp1000_u12,
     ../factory2/predictors/montana/avh1_u12,
     ../factory2/predictors/montana/avh2_u12,
     ../factory2/predictors/montana/avh3_u12,
     ../factory2/predictors/montana/avh4_u12,
     ../factory2/predictors/montana/avh5_u12,
     ../factory2/predictors/montana/mt_andvi_u12,
     nearest)

## divide 1k files for floppy transfer
```

```
transfer/mt2_avh1k.dat1 = sample(mt2/map1000/mt2_1k.txt,
     ../factory2/predictors/montana/mtdem1000_u12,
     nearest)
transfer/mt2_avh1k.dat2 = sample(mt2/map1000/mt2_1k.txt,
     ../factory2/predictors/montana/mtasp1000_u12,
     ../factory2/predictors/montana/mtslp1000_u12,
     nearest)
transfer/mt2_avh1k.dat3 = sample(mt2/map1000/mt2_1k.txt,
     ../factory2/predictors/montana/mt_andvi_u12,
     ../factory2/predictors/montana/mt_avh1_u12,
     ../factory2/predictors/montana/mt_avh2_u12,
     nearest)
transfer/mt2_avh1k.dat4 = sample(mt2/map1000/mt2_1k.txt,
     ../factory2/predictors/montana/mt_avh3_u12,
     ../factory2/predictors/montana/mt_avh4_u12,
     ../factory2/predictors/montana/mt_avh5_u12,
     nearest)

transfer/mt2_avh5k.dat = sample(mt2/responses/mt2_xy5_u12.dat,
     ../factory2/predictors/montana/mtdem1000_u12,
     ../factory2/predictors/montana/mtasp1000_u12,
     ../factory2/predictors/montana/mtslp1000_u12,
     ../factory2/predictors/montana/avh1_u12,
     ../factory2/predictors/montana/avh2_u12,
     ../factory2/predictors/montana/avh3_u12,
     ../factory2/predictors/montana/avh4_u12,
     ../factory2/predictors/montana/avh5_u12,
     ../factory2/predictors/montana/mt_andvi_u12,
     nearest)

bye
bye

## compress all files and check that each is <1.2M to fit on floppy

cd transfer
compress transfer/mt1_nlcd1k.dat
         transfer/mt1_nlcd5k.dat
         transfer/mt2_nlcd1k.dat
         transfer/mt2_nlcd5k.dat
         transfer/mt1_avh1k.dat
         transfer/mt1_avh5k.dat
         transfer/mt2_avh1k.dat1
         transfer/mt2_avh1k.dat2
         transfer/mt2_avh1k.dat3
         transfer/mt2_avh1k.dat4
         transfer/mt2_avh5k.dat
```

Appendix B: Installation Notes for S-Plus Libraries

Appendix B-1. Installation procedures for mda library of functions.

```
#To install the mda software:
#-----------------------------

# in unix, copy share file here and unpack

      mkdir /export/jerry2/gretchen/diss/s/mda
      mkdir .Data
      mkdir .Data/.Help
      sh ratfor.shar

# in unix, prepare directory as chapter for converted functions

      cd /export/jerry2/gretchen/diss/s/mda/.Data
      mkdir __Meta
      cd ..
      Splus5 CHAPTER
      Splus5 make

# in Splus, convert the old library and help functions

      convertOldLibrary("/export/jerry2/gretchen/diss/s/mda")
      convertOldDoc("/export/jerry2/gretchen/diss/s/mda",
                  "/export/jerry2/gretchen/diss/s/mda")

# in unix, global change "dyn.load" to "dyn.open" in all s-functions
(dumpdata.mda)
#    then source (does convertOldLibrary do this?)

      source("/export/jerry2/gretchen/diss/s/mda/dumpdat2.mda")

# in Splus, create pointers to mars object and help files

      attach("/export/jerry2/gretchen/diss/s/mda")
      # library or CHAPTER might be better ?
      .mars.object_"/export/jerry2/gretchen/diss/s/mda/MARS.o"
      .bruto.object_"/export/jerry2/gretchen/diss/s/mda/BRUTO.o"
```

Appendix B-2. Instalation procedures for FUNFITS.

```
# In UNIX


      mkdir Funfits
      cd Funfits


# now move the tar file funfits51.tar.Z into this directory


     uncompress funfits51.tar.Z

     tar -xvf funfits51.tar



## run make file line by line and troubleshoot

# initialize directories

      mkdir funs
      mkdir data
      mkdir bin
      mkdir lib
      mkdir lib/Gcvpack
      mkdir lib/Nnreg

# set Splus5 work and set home directory for FUNFITS

      setenv S_WORK /export/jerry2/gretchen/s5work
      Splus5 <  dump.header.q

# source functions and data

      cat dfun?.q | Splus5
      cat ddata?.q | Splus5
      cat dextra.q |    Splus5

# there are few functions that have S and Fortran versions
# use the S code ones as the default ( but see fortran item below)
#
      Splus5 < set.scode.S
      chmod og+r .Data
      chmod og+r .Data/.Help
      chmod og+r .Data/.Help/*


#  compile all FORTRAN that will be dynloaded into a shared library
      cd src/dynload
      Splus5 CHAPTER
```

```
# S.so is the created shared library this should be moved into the
# FUNFITS home directory
#
      mv S.so ../..

# rm the .o files and move back up to FUNFTIS home directory
      rm *.o
      cd ../..

Splus5 < set.fort.S
chmod og+r bin
#rm all.f

# These all deal with stand alone FORTRAN program that are executed in
# the shell with the unix command in Splus
# cover:
      touch null.o
      rm *.o
      f77  -O src/Cover/cover.f -o bin/cover
      chmod og+x bin

# nnreglib
      touch null.o
      rm *.o
      f77 -O src/Nnreg/nnreg.f src/Nnreg/lib/*.f -o bin/nnreg
      f77 -O src/Nnreg/nnregci.f src/Nnreg/lib/*.f -o bin/nnregci
      chmod og+x bin/nnreg
      chmod og+x bin/nnregci
# lle
      touch null.o
      rm *.o
      f77  -O src/Lle/*.f -o bin/lle
      touch null.o
      rm  *.o
      chmod og+x bin/lle

## source S-code
## switch to S-functions
```

Appendix C: S-Plus Code for Modeling System

Appendix C-00. p.go

```
#######
# source supplemental functions
#######

source("/export/jerry2/gretchen/diss/s/myfun/f0.functions")

#######
# input, filter, and restructure data
#######

eco_"ut2"
data.path_"/export/jerry2/gretchen/diss/dat/"
results.path_"/export/jerry2/gretchen/diss/output/"
seed1_1
holdout_.3
source("/export/jerry2/gretchen/diss/s/myfun/f1.data",immediate=T)
  dat.map.ut2_dat.map
  datd.all.ut2_datd.all
  datd.mod.ut2_datd.mod
  datd.tst.ut2_datd.tst
  datc.all.ut2_datc.all
  datc.mod.ut2_datc.mod
  datc.tst.ut2_datc.tst

#####
# quick reset for reruns
#####

dat.map_dat.map.ut2
datd.all_datd.all.ut2
datd.mod_datd.mod.ut2
datd.tst_datd.tst.ut2
datc.all_datc.all.ut2
datc.mod_datc.mod.ut2
datc.tst_datc.tst.ut2

######
# model FORTYP.2
######

slow_T
Xname_"all"
Yname_"FORTYP.2"
source("/export/jerry2/gretchen/diss/s/myfun/f2.model",immediate=T)
  FORTYP2.mod.ut2_mars.mod
  FORTYP2.mod_mars.mod

######
# model FORTYP.3
######

Yname_"FORTYP.3"
source("/export/jerry2/gretchen/diss/s/myfun/f2.model",immediate=T)
```

```
   FORTYP3.mod.ut2_mars.mod
   FORTYP3.mod_mars.mod

######
# model BIOTOT
######

Yname_"BIOTOT"
source("/export/jerry2/gretchen/diss/s/myfun/f2.model",immediate=T)
   BIOTOT.mod.ut2_mars.mod
   BIOTOT.mod_mars.mod

######
# model CRCOV
######

Yname_"CRCOV"
source("/export/jerry2/gretchen/diss/s/myfun/f2.model",immediate=T)
   CRCOV.mod.ut2_mars.mod
   CRCOV.mod_mars.mod

######
# model STAGECL
######

Yname_"STAGECL"
source("/export/jerry2/gretchen/diss/s/myfun/f2.model",immediate=T)
   STAGECL.mod.ut2_mars.mod
   STAGECL.mod_mars.mod

######
# model QMDALL
######

Yname_"QMDALL"
source("/export/jerry2/gretchen/diss/s/myfun/f2.model",immediate=T)
   QMDALL.mod.ut2_mars.mod
   QMDALL.mod_mars.mod

######
# generate maps
######

map.path_"/export/jerry2/gretchen/diss/maps/"
source("/export/jerry2/gretchen/diss/s/myfun/f3.map",immediate=T)

######
# produce population totals
######

source("/export/jerry2/gretchen/diss/s/myfun/f4.strat",immediate=T)

######
# display results
######
```

```
source("/export/jerry2/gretchen/diss/s/myfun/f5.results",immediate=T)
```

Appendix C-0. p0.functions

```
#######
# p0.functions: supplemental function sourcefiles
#######

#######
# data functions
#######

## class.ind: generates a class indicator function from a given factor
class.ind_function(cl){
  n_length(cl)
  cl_as.factor(cl)
  x_matrix(0,n,length(levels(cl)))
  x[(1:n)+n*(codes(cl)-1)]_1
  dimnames(x)_list(row.names(cl),levels(cl))
  x
}

## nona.fun: eleiminates NA's from dataframe
nona1.fun_function(x1){sum(match(is.na(x1),T,nomatch=0))}

##f.order: establish plotting order of variables in dataframe
f.order_function(fvar,cvar){
   f.medians_tapply(cvar,fvar,median)
   newvar_ordered(fvar,levels=names(sort(f.medians)))
   return(newvar)
}

## f.zero1: rescale values to 0-1
f.zero1_function(x){
  x01_(x-min(x))/(max(x)-min(x))
  return(x01)
}

## f.nlcdmod: collapse nlcd veg to 0, 40, 50
f.nlcdmod_function(y){
             n_length(y)
             yhat_rep(0,n)
             yhat[match(y,c(40,41,42,43,91))!="NA"]_1
             return(yhat)
          }

## trcos: transform aspect to wetness index
trcos_function(x,deg){
  trasp_(cos((x-deg)*pi/180)+1)/2
  return(trasp)}


#######
# map accuracy functions
#######

## khat (written by T. Frescino)
```

```
f.khat_function(x) {
  N_sum(apply(x, 1, sum))
  total_0
  for(i in 1:(nrow(x))) {
    total_sum(x[i,])*sum(x[,i]) + total
  }
  (N*sum(diag(x))-total)/(N^2-total)
}


## khat.var (written by T. Frescino)
f.khat.var_function(x) {
   N_sum(apply(x, 1, sum))
   theta1_sum(diag(x)/N)
   theta2_0
   for(i in 1:(nrow(x))){
      theta2_((sum(x[i,])*sum(x[,i]))/N^2) + theta2
   }
   theta3_0
   for(i in 1:(nrow(x))){
      theta3_((sum(x[i,]) + sum(x[,i])) * x[i,i])/N^2 + theta3
   }
   theta4_0
   for(i in 1:(nrow(x))){
      theta4j_0
      for(j in 1:(ncol(x))){
         theta4j_(((sum(x[i,])) + (sum(x[,j])))^2 * x[i,j])/N^3 +
theta4j
      }
      theta4_theta4j+theta4
   }
   (((theta1*(1-theta1))/(1-theta2)^2) +
       (2*(1-theta1) * (2*theta1*theta2 - theta3))/((1-theta2)^3) +
       (((1-theta1)^2) * (theta4-4*theta2^2))/(1-theta2)^4)*1/N
}


## mykhat (written by T. Frescin
f.mykhat_function(x){
  N_sum(x)
  xiplus_apply(x,2,sum)
  xplusi_apply(x,1,sum)
  return( (N*sum(diag(x))-sum(xiplus*xplusi)) / (N^2-
sum(xiplus*xplusi)) )
}


## mapacc0
f.mapacc0_function(tru,pred){
      nr1_length(tru)
      confus.mat_table(tru,pred)
      if(ncol(confus.mat)==1){confus.mat_cbind(confus.mat,c(0,0))}
      pcc_round(sum(diag(confus.mat))/nr1,3)
      pcc.se_round(sqrt(pcc*(1-pcc)/nr1),3)
      prod0_round(confus.mat[1,1]/sum(confus.mat[1,]),3)
      prod1_round(confus.mat[2,2]/sum(confus.mat[2,]),3)
      kappa_round(f.khat(t(confus.mat)),3)
      kappa.se_round(sqrt(f.khat.var(confus.mat)),3)
      return(pcc,pcc.se,prod0,prod1,kappa,kappa.se)}


## mapaccl
f.mapaccl_function(tru,pred,span){
      rho_round(cor(tru,pred),3)
      good_pred[abs(tru-pred)<=(span*pred)]
      pwi_round(length(good)/length(pred),3)
      rmse_round(sqrt(sum((pred-tru)**2)/n),3)
      return(rho,pwi,rmse)
}


#######
# stratification functions
#######

## dss
f.dss_function(y,strata,strcodes,nph,N,bin){
   L_length(strcodes)
   np_sum(nph)
   wh_nph/np
   nh_rep(0,L)
   ybh_rep(0,L)
   s2h_rep(0,L)
   for (h in 1:L){
     crit_strata==strcodes[h]
     nh[h]_length(y[crit])
     ybh[h]_mean(y[crit])
     if (bin==1){s2h[h]_nh[h]*ybh[h]*(1-ybh[h])/(nh[h]-1)}
     else{s2h[h]_var(y[crit])}
   }
   ybst_sum(wh*ybh)
   vybst_sum((wh^2)*s2h/nh)-sum(wh*s2h/N)+
         ((N-np)/(np*(N-1)))*sum(wh*(ybh-ybst)^2)
   yb_ybst
   seyb_sqrt(vybst)
   lyb_yb-1.96*seyb
   uyb_yb+1.96*seyb
   allo_round(nh/sum(nh),2)
   proallo_round(wh,2)
   optallo_round(nh*sqrt(s2h)/sum(nh*sqrt(s2h)),2)
   allodif_sum(abs(allo-optallo))
   return(N*yb,N*seyb,N*lyb,N*uyb,allo,proallo,optallo,allodif)
}

## srs
f.srs_function(y,type){
   yb_mean(y)
   n_length(y)
   if (type==0){seyb_sqrt(yb*(1-yb)/n)}
   else {seyb_sqrt(var(y)/n)}
   lyb_yb-1.96*seyb
   uyb_yb+1.96*seyb
   pcntse_seyb/yb
   return(round(c(yb,seyb,lyb,uyb,pcntse),3))
}

## str
f.str_function(y,strata,strcodes,Wh,type){
   n_length(y)
   L_length(strcodes)
   nh_rep(0,L)
   ybh_rep(0,L)
   s2h_rep(0,L)
   for (h in 1:L){
     crit_strata==strcodes[h]
     nh[h]_length(y[crit])
     ybh[h]_mean(y[crit])
     if (type==0){s2h[h]_ybh[h]*(1-ybh[h])}
     else{s2h[h]_var(y[crit])}
   }
   ybst_sum(Wh*ybh)
   vybst_sum(Wh*s2h)/n
   yb_ybst
   seyb_sqrt(vybst)
   lyb_yb-1.96*seyb
   uyb_yb+1.96*seyb
   pcntse_seyb/yb
   return(round(c(yb,seyb,lyb,uyb,pcntse),3))
}
```

Appendix C-1. p1.data

```
#########################################################################
print("###### p1.data: create model, test, and map data frames ######")
#########################################################################

#######
print("# specify inputs and outputs")
#######

in.fld_paste(data.path,eco,"/",eco,"_fld.dat",sep="")
in.avh5k_paste(data.path,eco,"/",eco,"_avh5k.dat",sep="")
in.avh1k_paste(data.path,eco,"/",eco,"_avh1k.dat",sep="")
in.nlcd1k_paste(data.path,eco,"/",eco,"_nlcd1k.dat",sep="")
in.nlcd5k_paste(data.path,eco,"/",eco,"_nlcd5k.dat",sep="")
in.1k_paste(data.path,eco,"/",eco,"_1k.dat",sep="")
out.map_paste(data.path,eco,"/",eco,"_map.dat",sep="")

#######
print("# read data and handle missing values")
#######

dat.fld_read.table(in.fld,header=T,sep=",",as.is=T)
  dat.fld_dat.fld[match(dat.fld$GLU,c(95:99))=="NA",]
  if ((eco=="az1") | (eco=="az2")){
    dat.fld$EASTING_dat.fld$UTME
    dat.fld$NORTHING_dat.fld$UTMN}
  if (eco=="ut1"){dat.fld$EASTING_round(dat.fld$EASTING/1000)*1000}
  for(i in 1:ncol(dat.fld)){dat.fld[,i]_as.numeric(dat.fld[,i])}
  dat.fld$FORTYP[dat.fld$FORTYP=="NA"]_0
  dat.fld$BIOTOT[dat.fld$BIOTOT=="NA"]_0
  dat.fld$STAGECL[dat.fld$STAGECL=="NA"]_0
  dat.fld$QMDALL[dat.fld$QMDALL=="NA"]_0
  dat.fld$CRCOV[dat.fld$CRCOV=="NA"]_0
  dat.fld$NVOLTOT[dat.fld$NVOLTOT=="NA"]_0
  dat.fld$NGRWCF[dat.fld$NGRWCF=="NA"]_0
#  crit_apply(dat.fld,1,nona1.fun)
#  dat.fld_dat.fld[crit<1,]
dat.avh5k_read.table(in.avh5k,header=T,sep=" ",as.is=T)
  dat.avh5k_dat.avh5k[,-1] ## remove mask info
  for(i in 1:ncol(dat.avh5k)){dat.avh5k[,i]_as.numeric(dat.avh5k[,i])}
  crit_apply(dat.avh5k,1,nona1.fun)
  dat.avh5k_dat.avh5k[crit<1,]
dat.nlcd5k_read.table(in.nlcd5k,header=T,sep=" ",as.is=T)
  dat.nlcd5k_dat.nlcd5k[,-1] ## remove mask info
  for(i in
1:ncol(dat.nlcd5k)){dat.nlcd5k[,i]_as.numeric(dat.nlcd5k[,i])}
  crit_apply(dat.nlcd5k,1,nona1.fun)
  dat.nlcd5k_dat.nlcd5k[crit<1,]
if((eco=="az2")|(eco=="az1")){
  dat.nlcd5k_dat.nlcd5k[
    (dat.nlcd5k$EASTING/1000==round(dat.nlcd5k$EASTING/1000))&
    (dat.nlcd5k$EASTING/1000==round(dat.nlcd5k$EASTING/1000)),]
  dat.avh5k_dat.avh5k[
     (dat.avh5k$EASTING/1000==round(dat.avh5k$EASTING/1000))&
     (dat.avh5k$EASTING/1000==round(dat.avh5k$EASTING/1000)),]
   dat.tmp_merge(dat.fld,dat.nlcd5k,all=T)
   dat.tmp[is.na(dat.tmp)]_0
   dat.fld_dat.tmp[,-length(names(dat.tmp))]
}

#######
print("# create mapping data frame")
#######

dat.avh1k_read.table(in.avh1k,header=T,sep="\t ",as.is=T)
  dat.avh1k_dat.avh1k[,-1] ## remove mask info
  for(i in 1:ncol(dat.avh1k)){dat.avh1k[,i]_as.numeric(dat.avh1k[,i])}
dat.nlcd1k_read.table(in.nlcd1k,header=T,sep=" ",as.is=T)
  dat.nlcd1k_dat.nlcd1k[,-1] ## remove mask info
  for(i in
1:ncol(dat.nlcd1k)){dat.nlcd1k[,i]_as.numeric(dat.nlcd1k[,i])}
dat.map_data.frame(dat.avh1k,dat.nlcd1k$NLCD)
      names(dat.map)_c(names(dat.avh1k),"NLCD")
             crit_apply(dat.map,1,nona1.fun)
             dat.map_dat.map[crit<1,]


#######
print("# check proportion of non-5k plots")
#######

# nrow(dat.fld[dat.fld$GRID==6,])/nrow(dat.fld)

#######
print("# create forest type variables")
#######

n_nrow(dat.fld)
dat.fld$FORTYP.2_rep(1,n)
dat.fld$FORTYP.2[dat.fld$FORTYP=="NA" | dat.fld$FORTYP==0]_0
dat.fld$FORTYP.3_rep(1,n)
dat.fld$FORTYP.3[dat.fld$FORTYP.2==0]_0
if ((eco!="mt1")&(eco!="mt2")){
  dat.fld$FORTYP.3[match(dat.fld$FORTYP,c(76:80,88,90,93,97))!="NA"]_0}
if ((eco=="mt1")|(eco=="mt2")){
  dat.fld$FORTYP.3[match(dat.fld$FORTYP,c(1,31:39))!="NA"]_0}

if ((eco!="mt1")&(eco!="mt2")){
  dat.fld$TWN_rep(2,n)
  dat.fld$TWN[dat.fld$FORTYP.2==0]_0
  dat.fld$TWN[match(dat.fld$FORTYP,c(76:80,88,90,93,97))!="NA"]_1}
if ((eco=="mt1")|(eco=="mt2")){
  dat.fld$TWN_rep(1,n)
  dat.fld$TWN[dat.fld$FORTYP.2==0]_0
  dat.fld$TWN[match(dat.fld$FORTYP,c(1,31:39))!="NA"]_2}

#######
print("# transform aspect")
```

```
#######

#dat.avh5k$ASPECT.1K_as.numeric(dat.avh5k$ASPECT.1K)
  dat.avh5k_dat.avh5k[dat.avh5k$ASPECT.1K>=0,]
  dat.avh5k$TRASP.1K_trcos(dat.avh5k$ASPECT.1K,30)
#dat.avh1k$ASPECT.1K_as.numeric(dat.avh1k$ASPECT.1K)
#  dat.avh1k_dat.avh1k[dat.avh1k$ASPECT.1K>=0,]
#  dat.avh1k$TRASP.1K_trcos(dat.avh1k$ASPECT.1K,30)
#dat.map$ASPECT.1K_as.numeric(dat.map$ASPECT.1K)
dat.map_dat.map[dat.avh5k$ASPECT.1K>=0,]
dat.map$TRASP.1K_trcos(dat.map$ASPECT.1K,30)

#######
print("# collapse vegetation classes")
#######

NLCD2_rep(0,nrow(dat.nlcd5k))
NLCD2[match(dat.nlcd5k$NLCD,c(41,42,43,91))!="NA"]_40
NLCD2[match(dat.nlcd5k$NLCD,c(51,52,53))!="NA"]_50
dat.nlcd5k$NLCD_NLCD2

NLCD2_rep(0,nrow(dat.map))
NLCD2[match(dat.map$NLCD,c(40,41,42,43,91))!="NA"]_40
NLCD2[match(dat.map$NLCD,c(50,51,52,53))!="NA"]_50
dat.map$NLCD_NLCD2


#######
print("# Create NDVI")
#######

# dat.gap5k$NDVI_(dat.gap5k$TM.4-dat.gap5k$TM.3)/
#                (dat.gap5k$TM.4+dat.gap5k$TM.3)

# dat.gap1k$NDVI_(dat.gap1k$TM.4-dat.gap1k$TM.3)/
#                (dat.gap1k$TM.4+dat.gap1k$TM.3)


#######
print("# identify factors")
#######

#dat.fld$FORTYP.2_factor(dat.fld$FORTYP.2)
#dat.fld$FORTYP.3_factor(dat.fld$FORTYP.3)
#dat.fld$PILU_factor(dat.fld$PILU)
#dat.fld$PITYP_factor(dat.fld$PITYP)
#dat.fld$PITYP_factor(dat.fld$PHYCL)
#dat.fld$PITYP_factor(dat.fld$CURVECL)

dat.nlcd5k$NLCD_factor(dat.nlcd5k$NLCD)
dat.map$NLCD_factor(dat.map$NLCD)
crit_apply(dat.map,1,nonal.fun)
      dat.map_dat.map[crit<1,]

#######
```

```
print("# merge data frames")
#######

# dat.all_merge(dat.fld,dat.gap5k,by=c("LONGTUDE","LATITUDE"))
dat.all_merge(dat.fld,dat.avh5k,by=c("EASTING","NORTHING"),all=F)
datb.all_merge(dat.all,dat.nlcd5k,by=c("EASTING","NORTHING"),all=F)
if (eco!="mt2"){
  datd.all_datb.all[,c("FORTYP.2","TWN","NVOLTOT","NGRWCF",
        "EASTING","NORTHING","ELEV.1K","TRASP.1K",

"SLOPE.1K","AVH.1","AVH.2","AVH.3","AVH.4","AVH.5","NDVI","NLCD")]
}
if (eco=="mt2"){
  datd.all_datb.all[,c("FORTYP.2","TWN","NVOLTOT","NGRWCF",
        "EASTING","NORTHING","ELEV.1K","TRASP.1K",
        "AVH.1","AVH.2","AVH.3","AVH.4","AVH.5","NDVI","NLCD")]
}
    set.seed(seed1)
    testset_sample(nrow(datd.all),round(holdout*nrow(datd.all)))
    datd.mod_datd.all[-testset,]
    datd.tst_datd.all[testset,]
if (eco!="mt2"){
  datc.all_datb.all[datb.all$FORTYP.2==1,]

datc.all_datc.all[,c("FORTYP.3","BIOTOT","STAGECL","QMDALL","CRCOV",
        "EASTING","NORTHING","ELEV.1K","TRASP.1K","SLOPE.1K",
        "AVH.1","AVH.2","AVH.3","AVH.4","AVH.5","NDVI","NLCD")]
}
if (eco=="mt2"){
  datc.all_datb.all[datb.all$FORTYP.2==1,]

datc.all_datc.all[,c("FORTYP.3","BIOTOT","STAGECL","QMDALL","CRCOV",
        "EASTING","NORTHING","ELEV.1K","TRASP.1K",
        "AVH.1","AVH.2","AVH.3","AVH.4","AVH.5","NDVI","NLCD")]
}
  set.seed(seed1)
  testset_sample(nrow(datc.all),round(holdout*nrow(datc.all)))
  datc.mod_datc.all[-testset,]
  datc.tst_datc.all[testset,]
```

Appendix C-2. p2.model

```
#########################################################################
print("## p2.model: models, diagnostics, predictive results ##")
#########################################################################

#######
print("# specify variable types")
#######

type_1
ftyp_0
if(Yname=="FORTYP.2"){type_0
                      ftyp_0}
if (Yname=="FORTYP.3"){type_0
                       ftyp_1}

#######
print("# specify .ps, .txt and results files")
#######

if (Yname=="TEST"){
   eco_"test"
   Xname_paste("TEST",noise,sep="")}
run.title_paste(eco,":",Yname,"~",Xname)
run.label_paste(eco,"_",Yname,"_",Xname,sep="")
run.ps_paste(results.path,eco,"/","ps/",run.label,".ps",sep="")
run.txt_paste(results.path,eco,"/","txt/",run.label,".txt",sep="")
run.res0_paste(results.path,"results0.txt",sep="")
run.res1_paste(results.path,"results1.txt",sep="")
if (Yname=="TEST"){
   run.res0_paste(results.path,"test0.txt",sep="")
   run.res1_paste(results.path,"test1.txt",sep="")}

#######
print("# define modelling variables")
#######

if (Xname=="all" & eco!="mt2")
{Xlist_c("ELEV.1K","TRASP.1K","SLOPE.1K","EASTING",

"NORTHING","AVH.1","AVH.2","AVH.3","AVH.4","AVH.5",
                     "NDVI","NLCD")
                   gam.form_formula(Y ~ s(ELEV.1K)+s(TRASP.1K)+
                      s(SLOPE.1K)+s(EASTING)+s(NORTHING)+s(AVH.1)+
                      s(AVH.2)+s(AVH.3)+s(AVH.4)+s(AVH.5)+s(NDVI)+NLCD)
}
if (Xname=="all" & eco=="mt2") {Xlist_c("ELEV.1K","TRASP.1K","EASTING",

"NORTHING","AVH.1","AVH.2","AVH.3","AVH.4","AVH.5",
                     "NDVI","NLCD")
                   gam.form_formula(Y ~ s(ELEV.1K)+s(TRASP.1K)+
                      s(EASTING)+s(NORTHING)+s(AVH.1)+
                        s(AVH.2)+s(AVH.3)+s(AVH.4)+s(AVH.5)+s(NDVI)+NLCD)
 }

 if (type==0 &
 ftyp==0){Y.mod_datd.mod[,(match(names(datd.mod),Yname))!="NA"]
              X.mod_datd.mod[,(match(names(datd.mod),Xlist))!="NA"]
            Y.tst_datd.tst[,(match(names(datd.tst),Yname))!="NA"]
              X.tst_datd.tst[,(match(names(datd.tst),Xlist))!="NA"]
              }
 if (((Yname!="TEST")&(type==1)) | (ftyp==1))
             {Y.mod_datc.mod[,(match(names(datc.mod),Yname))!="NA"]
              X.mod_datc.mod[,(match(names(datc.mod),Xlist))!="NA"]
            Y.tst_datc.tst[,(match(names(datc.tst),Yname))!="NA"]
              X.tst_datc.tst[,(match(names(datc.tst),Xlist))!="NA"]
              }
 if (Yname=="TEST"){
       n_1000
       xz_rep(0,n)
       x_matrix(runif(n*10),ncol=10)
       X.all_data.frame(x)
       X.mod_X.all[1:700,]
       X.tst_X.all[701:1000,]
       names(X.mat)_c("x1","x2","x3","x4","x5","x6","x7","x8","x9","x10"
 )
       if (noise!=0){eps_rnorm(n,0,sqrt(noise))}
       if(noise==0){eps_0}
       Y.all_.2*sin(pi*x[,1]*x[,2])+.4*(x[,3].5)^2+.2*x[,4]+.1*x[,5]+eps
       Y.mod_Y.all[1:700]
       Y.tst_Y.all[701:1000]
       gam.form_formula(Y ~ s(x.1)+s(x.2)+
                       s(x.3)+s(x.4)+s(x.5)+s(x.6)+
                       s(x.7)+s(x.8)+s(x.9)+s(x.10))}

if (type==0){YX.mod_data.frame(as.factor(Y.mod),X.mod)}
if (type==1){YX.mod_data.frame(Y.mod,X.mod)}

#######
print("# activate ps file")
#######

postscript(file=run.ps,horizontal=F,append=F,onefile=T)


#######
print("# NLCD")
#######

sink(file=run.txt,append=F)
if(type==0 & Xname=="all"){time0_proc.time()[3]

           Y.nlcd_f.nlcdmod(X.tst$NLCD)
           runtime_proc.time()[3]-time0
           res0_f.mapacc0(Y.tst,Y.nlcd)

res_data.frame(eco,Yname,Xname,"NLCD",runtime,data.frame(res0))
```

```
         write.table(res,run.res0,append=T,dimnames.write=F,sep="\t")
           }

if(type==1 & Yname!="TEST" & Xname=="all"){time0_proc.time()[3]
         nlcd.mod_lm(Y.mod~-1+NLCD,data=X.mod)
           Y.nlcd_predict(nlcd.mod,newdata=X.tst)
           runtime_proc.time()[3]-time0
           res0_f.mapacc1(Y.tst,Y.nlcd,.25)

res_data.frame(eco,Yname,Xname,"NLCD",runtime,data.frame(res0))
         write.table(res,run.res1,append=T,dimnames.write=F,sep="\t")
           par(mfrow=c(2,2),pty="s")
           plot.gam(nlcd.mod,se=T,main=paste(run.title,".....NLCD"))
           plot(Y.tst,Y.nlcd,xlim=c(0,max(Y.tst)),ylim=c(0,max(Y.tst)))
           lines(c(0,max(Y.nlcd)),c(0,max(Y.nlcd)))
           yax_max(abs(Y.tst-Y.nlcd))*1.1
         plot(Y.nlcd,(Y.tst-Y.nlcd),ylim=c((0-yax),yax))
         lines(c(0,max(Y.nlcd)),c(0,0))}
if(Yname=="TEST"){time0_proc.time()[3]
         lm.mod_lm(Y.mod~1+x.1+x.2+x.3+x.4+x.5+x.6+x.7+x.8+x.9+x.10,
                    data=X.mod)
           Y.lm_predict(lm.mod,X.tst)
           Y.lm[Y.lm<0]_0
           Y.lm[Y.lm>max(Y.tst)]_max(Y.tst)
           runtime_proc.time()[3]-time0
           res0_f.mapacc1(Y.tst,Y.lm,.25)

res_data.frame(eco,paste(Yname,noise,sep=""),Xname,"LM",runtime,data.fr
ame(res0))
         write.table(res,run.res1,append=T,dimnames.write=F,sep="\t")
           par(mfrow=c(2,1),pty="s")
           plot(Y.tst,Y.lm,xlim=c(0,max(Y.tst)),ylim=c(0,max(Y.tst)),
                main=paste(run.title,".....LM, noise=",noise))
           lines(c(0,max(Y.lm)),c(0,max(Y.lm)))
           yax_max(abs(Y.tst-Y.lm))
           plot(Y.lm,(Y.tst-Y.lm),ylim=c((0-yax),yax))
           lines(c(0,max(Y.lm)),c(0,0))}

sink()


#######
print("# GAM")
#######

sink(file=run.txt,append=T)
if (type==0){Y_Y.mod
             time0_proc.time()[3]
             gam.mod0_gam(gam.form,data=X.mod,family=binomial)
             if (slow==F){gam.mod_gam.mod0}
             if(slow==T){
                my.scope_gam.scope(data.frame(Y.mod,X.mod))
                gam.mod_step(gam.mod0,my.scope,trace=F)}
             Y.gam_round(predict.gam(gam.mod,X.tst,type="response"))
             runtime_proc.time()[3]-time0
```

```
              res0_f.mapacc0(Y.tst,Y.gam)

 res_data.frame(eco,Yname,Xname,"GAM",runtime,data.frame(res0))
            write.table(res,run.res0,append=T,dimnames.write=F,sep="\t")
              par(mfrow=c(4,3))
              plot.gam(gam.mod,se=T,main=paste(run.title,".....GAM"))
            print("##########################")
              print(paste(run.title,".....GAM"))
              print(summary(gam.mod))
            }

 if (type==1){Y_Y.mod
            time0_proc.time()[3]
            gam.mod0_gam(gam.form,data=X.mod,family=gaussian,
                         control=gam.control(maxit=20,bf.maxit=20))
                          gam.mod_gam.mod0
              if (slow==F){gam.mod_gam.mod0}
              if(slow==T){
                my.scope_gam.scope(data.frame(Y.mod,X.mod))
                gam.mod_step(gam.mod0,my.scope,trace=F,
                  control=gam.control(maxit=20,bf.maxit=20))}
              Y.gam_predict.gam(gam.mod,X.tst,type="response")
              Y.gam[Y.gam<0]_0
              Y.gam[Y.gam>max(Y.tst)]_max(Y.tst)
              runtime_proc.time()[3]-time0
              res0_f.mapacc1(Y.tst,Y.gam,.25)

res_data.frame(eco,Yname,Xname,"GAM",runtime,data.frame(res0))
              if(Yname=="TEST"){
                res_data.frame(eco,paste(Yname,noise,sep=""),
                    Xname,"GAM",runtime,data.frame(res0))}
            write.table(res,run.res1,append=T,dimnames.write=F,sep="\t")
              par(mfrow=c(4,3))
              plot.gam(gam.mod,se=T)
              par(mfrow=c(2,1),pty="s")
              plot(Y.tst,Y.gam,main=paste(run.title,".....GAM"),
                   xlim=c(0,max(Y.tst)),ylim=c(0,max(Y.tst)),pty="s")
               lines(c(0,max(Y.gam)),c(0,max(Y.gam)))
              plot(Y.gam,(Y.tst-Y.gam),ylim=c((0-yax),yax))
               lines(c(0,max(Y.gam)),c(0,0))
            print("##########################")
              print(paste(run.title,".....GAM"))
              print(summary(gam.mod))
            }
sink()

#######
print("# CART")
#######

sink(file=run.txt,append=T)
time0_proc.time()[3]
cart0.mod_tree(YX.mod)
if (slow==F){
  cv.res_cv.tree(cart0.mod)
```

```
   opsz_ceiling(cv.res$size[cv.res$dev==min(cv.res$dev)])
   vote.opsz_opsz
   cart.mod_prune.tree(cart0.mod,best=vote.opsz)}
 if (slow==T){
   opsz_rep(0,20)
   for ( i in 1:20){
      cv.res_cv.tree(cart0.mod)
      opsz[i]_ceiling(cv.res$size[cv.res$dev==min(cv.res$dev)])
   }

 vote.opsz_as.numeric(names(table(opsz)[table(opsz)==max(table(opsz))]))
   cart.mod_prune.tree(cart0.mod,best=vote.opsz)}
 if (type==0){Y.cart_predict.tree(cart.mod,X.tst,type="class")
              runtime_proc.time()[3]-time0
              res0_f.mapacc0(Y.tst,Y.cart)

 res_data.frame(eco,Yname,Xname,"CART",runtime,data.frame(res0))

 write.table(res,run.res0,append=T,dimnames.write=F,sep="\t")}
 if (type==1){Y.cart_predict.tree(cart.mod,X.tst)
              Y.cart[Y.cart<0]_0
              Y.cart[Y.cart>max(Y.tst)]_max(Y.tst)
              runtime_proc.time()[3]-time0
              res0_f.mapacc1(Y.tst,Y.cart,.25)

 res_data.frame(eco,Yname,Xname,"CART",runtime,data.frame(res0))
           if(Yname=="TEST"){
                res_data.frame(eco,paste(Yname,noise,sep=""),
                    Xname,"CART",runtime,data.frame(res0))}
           write.table(res,run.res1,append=T,dimnames.write=F,sep="\t")
             par(mfrow=c(2,1),pty="s")
             plot(Y.tst,Y.cart,main=paste(run.title,".....CART"),
                  xlim=c(0,max(Y.tst)),ylim=c(0,max(Y.tst)),pty="s")
             lines(c(0,max(Y.cart)),c(0,max(Y.cart)))
             plot(Y.cart,(Y.tst-Y.cart),ylim=c((0-yax),yax))
             lines(c(0,max(Y.cart)),c(0,0))}
if
(vote.opsz>1){post.tree(cart.mod,title=run.title,file=paste(run.ps,"tre
e",sep=""))}

print("###########################")
print(paste(run.title,".....CART"))
print(summary(cart.mod))
sink()

#######
print("# MARS")
#######

sink(file=run.txt,append=T)
#source("/export/jerry2/gretchen/diss/s/mda/dumpdat2.mda")
#attach("/export/jerry2/gretchen/diss/s/mda")
#.mars.object_"/export/jerry2/gretchen/diss/s/mda/MARS.o"
#.bruto.object_"/export/jerry2/gretchen/diss/s/mda/BRUTO.o"
```

```
if (Xname=="all"){
   indics_data.frame(class.ind(X.mod$NLCD))
   posit_row(as.matrix(names(X.mod)))[names(X.mod)=="NLCD"]
   X.moda_X.mod[,-posit]
   X.moda_data.frame(X.moda,indics)
   indics_data.frame(class.ind(X.tst$NLCD))
   X.tsta_X.tst[,-posit]
   X.tsta_data.frame(X.tsta,indics)}
if (Xname!="all"){X.moda_X.mod
                  X.tsta_X.tst}
#if (type==0) {Y_as.numeric(Y.mod)-1}
#if (type==1) {Y_Y.mod}
Y_Y.mod
time0_proc.time()[3]
mars.mod_mars(X.moda,Y.mod,degree=2)
if (type==0){Y.mars_round(predict.mars(mars.mod,X.tsta))
             Y.mars[Y.mars<0]_0
             Y.mars[Y.mars>1]_1
             runtime_proc.time()[3]-time0
             res0_f.mapacc0(Y.tst,Y.mars)

res_data.frame(eco,Yname,Xname,"MARS",runtime,data.frame(res0))

write.table(res,run.res0,append=T,dimnames.write=F,sep="\t")}
if (type==1){Y.mars_predict.mars(mars.mod,X.tsta)
           Y.mars[Y.mars<0]_0
             Y.mars[Y.mars>max(Y.tst)]_max(Y.tst)
             runtime_proc.time()[3]-time0
             res0_f.mapacc1(Y.tst,Y.mars,.25)

res_data.frame(eco,Yname,Xname,"MARS",runtime,data.frame(res0))
           if(Yname=="TEST"){
               res_data.frame(eco,paste(Yname,noise,sep=""),
                   Xname,"MARS",runtime,data.frame(res0))}

write.table(res,run.res1,append=T,dimnames.write=F,sep="\t")
           par(mfrow=c(2,1),pty="s")
           plot(Y.tst,Y.mars,main=paste(run.title,".....MARS"),
                  xlim=c(0,max(Y.tst)),ylim=c(0,max(Y.tst)))
             lines(c(0,max(Y.mars)),c(0,max(Y.mars)))
             plot(Y.mars,(Y.tst-Y.mars),ylim=c((0-yax),yax))
             lines(c(0,max(Y.mars)),c(0,0))}

### decipher, print, and plot resulting model

sel.fac_abs(mars.mod$factor[mars.mod$selected.terms,])
n.bs_nrow(sel.fac)
n.col_ncol(sel.fac)
fin.fac_t(matrix(sel.fac[1,]))
n.fin_1
for (i in 2:n.bs){
  adit_T
  for ( j in 1:n.fin){
    if (sum(sel.fac[i,]==fin.fac[j,])==n.col) {adit_F}
  }
```

```
   if (adit==T){fin.fac_rbind(fin.fac,sel.fac[i,])
                n.fin_nrow(fin.fac)}
}
fin.fac_data.frame(fin.fac)


print("##########################")
print(paste(run.title,".....MARS"))
for(i in 1:n.fin){
   print(names(fin.fac)[fin.fac[i,]==1])}
sink()

#######
print("# ANN")
#######

sink(file=run.txt,append=T)
FUNFITS.BIN_"/export/jerry2/gretchen/diss/s/Funfits/bin/"
attach("/export/jerry2/gretchen/diss/s/Funfits")
if (Xname!="all"){X.moda_X.mod
                  X.tsta_X.tst}
#if (type==0) {Y_as.numeric(Y.mod)-1}
#if (type==1) (Y_Y.mod)
Y_Y.mod
time0_proc.time()[3]
if (slow==F){ann.mod_nnreg(X.moda,Y,k1=1,k2=3,fast=T)}
if (slow==T){ann.mod_nnreg(X.moda,Y,k1=1,k2=5,fast=T)}
ann.res_summary(ann.mod)
#ann.mod$best.model_
#            row(as.matrix(ann.res[,5]))[ann.res[,5]==min(ann.res[,5])]
if (type==0){Y.ann_round(predict.nnreg(ann.mod,as.matrix(X.tsta)))
             Y.ann[Y.ann<0]_0
             Y.ann[Y.ann>1]_1
             runtime_proc.time()[3]-time0
             res0_f.mapacc0(Y.tst,Y.ann)

res_data.frame(eco,Yname,Xname,"ANN",runtime,data.frame(res0))

write.table(res,run.res0,append=T,dimnames.write=F,sep="\t")}
if (type==1){Y.ann_predict.nnreg(ann.mod,as.matrix(X.tsta))
           Y.ann[Y.ann<0]_0
             Y.ann[Y.ann>max(Y.tst)]_max(Y.tst)
             runtime_proc.time()[3]-time0
             res0_f.mapacc1(Y.tst,Y.ann,.25)

res_data.frame(eco,Yname,Xname,"ANN",runtime,data.frame(res0))
           if(Yname=="TEST"){
               res_data.frame(eco,paste(Yname,noise,sep=""),
                   Xname,"ANN",runtime,data.frame(res0))}
           write.table(res,run.res1,append=T,dimnames.write=F,sep="\t")
           par(mfrow=c(2,1),pty="s")
           plot(Y.tst,Y.ann,main=paste(run.title,".....ANN"),
                  xlim=c(0,max(Y.tst)),ylim=c(0,max(Y.tst)))
             lines(c(0,max(Y.ann)),c(0,max(Y.ann)))
             plot(Y.ann,(Y.tst-Y.ann),ylim=c((0-yax),yax))
```

```
              lines(c(0,max(Y.ann)),c(0,0))}
par(mfrow=c(1,1))

plot(ann.mod,main=paste(run.title,".....ANN"))
print("###########################")
print(paste(run.title,".....ANN"))
print(summary(ann.mod))
sink()

#######
print("# close out")
#######

dev.off()
```

Appendix C-3. p3.map

```
#######################################################################
print("## p3.map: generate maps ##")
#######################################################################

#######
print("# specify .ps and ascii files")
#######

run.title_paste(eco,":",Xname)
run.label_paste(eco,"_",Xname,"_map",sep="")
run.ps_paste(map.path,eco,"/",run.label,".ps",sep="")
run.ascii_paste(map.path,eco,"/",run.label,".txt",sep="")

#######
print("# predict")
#######

if (eco != "mt2"){
dat.map_dat.map[,c("EASTING","NORTHING","ELEV.1K","TRASP.1K","SLOPE.1K"
,

"AVH.1","AVH.2","AVH.3","AVH.4","AVH.5","NDVI","NLCD")]}
if (eco == "mt2"){
dat.map_dat.map[,c("EASTING","NORTHING","ELEV.1K","TRASP.1K",

"AVH.1","AVH.2","AVH.3","AVH.4","AVH.5","NDVI","NLCD")]}
indics_data.frame(class.ind(dat.map$NLCD))
dat.mapa_data.frame(dat.map,indics)
if (eco != "mt2"){dat.mapa_dat.mapa[,-12]}
if (eco == "mt2"){dat.mapa_dat.mapa[,-11]}


NLCD.map_f.nlcdmod(dat.map$NLCD)
FORTYP2.map_round(predict.mars(FORTYP2.mod,dat.mapa))
      FORTYP2.map[FORTYP2.map<0]_0
      FORTYP2.map[FORTYP2.map>1]_1
FORTYP3.map_round(predict.mars(FORTYP3.mod,dat.mapa))
      FORTYP3.map[FORTYP3.map<0]_0
      FORTYP3.map[FORTYP3.map>1]_1
      FORTYP3.map_FORTYP3.map+1
      FORTYP3.map[FORTYP2.map==0]_0
BIOTOT.map_round(predict.mars(BIOTOT.mod,dat.mapa),2)
      BIOTOT.map[BIOTOT.map<0]_0
      BIOTOT.map[BIOTOT.map>max(datc.mod$BIOTOT)]_max(datc.mod$BIOTOT)
      BIOTOT.map[NLCD.map==0]_0
CRCOV.map_round(predict.mars(CRCOV.mod,dat.mapa),2)
      CRCOV.map[CRCOV.map<0]_0
      CRCOV.map[CRCOV.map>100]_100
      CRCOV.map[NLCD.map==0]_0
STAGECL.map_round(predict.mars(STAGECL.mod,dat.mapa),2)
      STAGECL.map[STAGECL.map<0]_0
      STAGECL.map[STAGECL.map>max(datc.mod$STAGECL)]_max(datc.mod$STAGE
CL)
      STAGECL.map[NLCD.map==0]_0
QMDALL.map_round(predict.mars(QMDALL.mod,dat.mapa),2)
      QMDALL.map[QMDALL.map<0]_0
      QMDALL.map[QMDALL.map>max(datc.mod$QMDALL)]_max(datc.mod$QMDALL)
      QMDALL.map[NLCD.map==0]_0


#######
print("# build levelplots")
#######
#postscript(file=run.ps,horizontal=F,append=F,onefile=T)
#ps.options(height=8,width=8)
#  east_dat.map$EASTING/1000
#  north_dat.map$NORTHING/1000
#  levelplot(NLCD.map~east*north,main=paste(run.title,"NLCD"),
#         col.regions=c(0:1),colorkey=T,region=T)
#  levelplot(FORTYP2.map~east*north,main=paste(run.title,"FORTYP2"),
#         col.regions=c(0:1),colorkey=T,region=T)
#  levelplot(BIOTOT.map~east*north,main=paste(run.title,"BIOTOT"),
#         col.regions=c(0:15)/15,colorkey=T,region=T)
#dev.off()

#######
print("# write asciis")
#######

attr.map_data.frame(dat.map$EASTING,dat.map$NORTHING,dat.map$ELEV.1K,

dat.map$NLCD,NLCD.map,FORTYP2.map,FORTYP3.map,BIOTOT.map,
                    CRCOV.map,STAGECL.map,QMDALL.map)
names(attr.map)_c("EASTING","NORTHING","ELEV.1K","NLCD","NLCD.map",
                   "FORTYP2.map","FORTYP3.map","BIOTOT.map",
                    "CRCOV.map","STAGECL.map","QMDALL.map")
write.table(attr.map,file=run.ascii)
```

Appendix C-4. p4.boot

```
##################
print("### p4.boot: program to produce bootstrap variance estimates
                    dif stratification schemes ###")
##################

#####
# print("# initialize objects")
#####


eco_"mtl"
dat.map_dat.map.mtl
datd.all_datd.all.mtl

#######
print("# specify inputs and outputs")
#######

boot.ps_paste(results.path,"f5/",eco,".boot.ps",sep="")
boot.out_paste(results.path,"resultsboot.txt",sep="")
boot2.out_paste(results.path,"resultsboot2.txt",sep="")
boottst.out_paste(results.path,"resultsboottst.txt",sep="")
boottst2.out_paste(results.path,"resultsboottst2.txt",sep="")

#######
print("# set up data")
#######

dat.toty_datd.all[,c("FORTYP.2","NVOLTOT","NGRWCF","TWN")]
dat.toty$FOREST_dat.toty$FORTYP.2
dat.toty$TIMBER_dat.toty$TWN
dat.toty$TIMBER[dat.toty$TIMBER!=2]_0
dat.toty$TIMBER[dat.toty$TIMBER==2]_1
dat.totx_datd.all[,c("EASTING","NORTHING","ELEV.1K","TRASP.1K",
      "SLOPE.1K","AVH.1","AVH.2","AVH.3","AVH.4","AVH.5","NDVI",
      "NLCD")]


#######
print("# bootstrap variance estimates #")
#######


n_nrow(dat.toty)
bootmat_matrix(0,nrow=bootn,ncol=n)
for ( i in 1:bootn){
  bootmat[i,]_sample(n,n,replace=T)
}

indics_data.frame(class.ind(dat.totx$NLCD))
   posit_row(as.matrix(names(dat.totx)))[names(dat.totx)=="NLCD"]
   dat.totxa_dat.totx[,-posit]
   dat.totxa_data.frame(dat.totxa,indics)
dat.map0_dat.map[,names(dat.totx)]
indics_data.frame(class.ind(dat.map0$NLCD))
   posit_row(as.matrix(names(dat.map0)))[names(dat.map0)=="NLCD"]
   dat.mapa_dat.map0[,-posit]
   dat.mapa_data.frame(dat.mapa,indics)

bootmars_function(bootset,dat.toty,dat.totx,dat.totxa,dat.map,dat.mapa)
{
   dat.booty_dat.toty[bootset,]
   dat.bootx_dat.totxa[bootset,]
   dat.bootnlcd_dat.totx$NLCD[bootset]

   yf_as.numeric(dat.booty$FOREST)
   xf_dat.bootx
   marsf.boot_mars(xf,yf,degree=2)

   y_as.numeric(dat.booty$TWN[dat.booty$FORTYP.2>0])-1
   x_dat.bootx[dat.booty$FORTYP.2>0,]
   mars.boot_mars(x,y,degree=2)

   str2.mars_round(predict.mars(marsf.boot,dat.bootx))
      str2.mars[str2.mars<0]_0
      str2.mars[str2.mars>1]_1
   str.mars_round(predict.mars(mars.boot,dat.bootx))
   str.mars[str.mars<0]_0
   str.mars[str.mars>1]_1
   str.mars_str.mars+1
   str.mars[str2.mars==0]_0

   bootf.map_round(predict.mars(marsf.boot,dat.mapa))
      bootf.map[bootf.map<0]_0
      bootf.map[bootf.map>1]_1
   boot.map_round(predict.mars(mars.boot,dat.mapa))
      boot.map[boot.map<0]_0
      boot.map[boot.map>1]_1
      boot.map_boot.map+1
      boot.map[bootf.map==0]_0
   wts0_as.numeric(table(boot.map))
   wts_wts0/sum(wts0)

   res0_t(as.matrix(f.str(y=dat.booty$FOREST,strata=str.mars,
      strcodes=c(0,1,2),Wh=wts,type=0)))
     res_data.frame(eco,"MARS","FOREST",res0)
     write.table(res,boot.out,append=T,dimnames.write=F,sep="\t")
     for.mean_res[,4]
     for.var_res[,5]

   res0_t(as.matrix(f.str(y=dat.booty$TIMBER,strata=str.mars,
      strcodes=c(0,1,2),Wh=wts,type=0)))
     res_data.frame(eco,"MARS","TIMBER",res0)
     write.table(res,boot.out,append=T,dimnames.write=F,sep="\t")
     tim.mean_res[,4]
     tim.var_res[,5]
   res0_t(as.matrix(f.str(y=dat.booty$NVOLTOT,strata=str.mars,
      strcodes=c(0,1,2),Wh=wts,type=1)))
     res_data.frame(eco,"MARS","NVOLTOT",res0)
     write.table(res,boot.out,append=T,dimnames.write=F,sep="\t")
     vol.mean_res[,4]
     vol.var_res[,5]


   res0_t(as.matrix(f.str(y=dat.booty$NGRWCF,strata=str.mars,
      strcodes=c(0,1,2),Wh=wts,type=1)))
     res_data.frame(eco,"MARS","NGRWCF",res0)
     write.table(res,boot.out,append=T,dimnames.write=F,sep="\t")
     grw.mean_res[,4]
     grw.var_res[,5]


     return(c(for.mean,for.var,tim.mean,tim.var,
              vol.mean,vol.var,grw.mean,grw.var))}


boot.res_apply(bootmat,1,bootmars,dat.toty,dat.totx,dat.totxa,dat.map,d
at.mapa)
 boot.res_data.frame(t(boot.res))
 write.table(boot.res,boot2.out,append=T,dimnames.write=F,sep="\t")
 boot.res.mt1_boot.res

 ####
 print("# plot results #")
 ####

 boot.eco_c(sqrt(var(boot.res[,1])),sqrt(var(boot.res[,3])),
            sqrt(var(boot.res[,5])),sqrt(var(boot.res[,7])))
 if (eco=="ut2"){
   MARS.var_c(.012,.01,31.929,.625)
   NLCD.var_c(.012,.012,36.34,.732)
   SRS.var_c(.013,.014,40.197,.809)
 }
 if (eco=="mt1"){
   MARS.var_c(.012,.011,5.197,.103)
   NLCD.var_c(.013,.013,5.505,.115)
   SRS.var_c(.015,.014,5.969,.124)
 }
 if (eco=="mt1"){
   MARS.var_c(.006,.011,41.85,1.194)
   NLCD.var_c(.008,.012,40.449,1.163)
   SRS.var_c(.009,.012,45.668,1.264)
 }
 postscript(boot.ps,append=F,width=8.5,height=11,horizontal=F)
 par(mfrow=c(2,2))

 bootplot_rep(0,95)
 for(i in 5:100){bootplot[i-4]_sqrt(var(boot.res[1:i,1]))}
 plot(5:100,bootplot,
      xlab="Number of bootstrap samples",
      ylab="Bootstrap standard errors",
     main="FOREST IN MT1",type="l",ylim=c(0.005,.020))
     points(100,MARS.var[1],pch=1)
     points(100,NLCD.var[1],pch=2)
     points(100,SRS.var[1],pch=3)

bootplot_rep(0,95)
for(i in 5:100){bootplot[i-4]_sqrt(var(boot.res[1:i,3]))}
plot(5:100,bootplot,
     xlab="Number of bootstrap samples",
     ylab="Bootstrap standard errors",
     main="TIMBERLAND IN MT1",type="l",ylim=c(0.005,.020))
     points(100,MARS.var[2],pch=1)
     points(100,NLCD.var[2],pch=2)
     points(100,SRS.var[2],pch=3)

bootplot_rep(0,95)
for(i in 5:100){bootplot[i-4]_sqrt(var(boot.res[1:i,5]))}
 plot(5:100,bootplot,
     xlab="Number of bootstrap samples",
     ylab="Bootstrap standard errors",
     main="NVOLTOT IN MT1",type="l",ylim=c(3,7))
     points(100,MARS.var[3],pch=1)
     points(100,NLCD.var[3],pch=2)
     points(100,SRS.var[3],pch=3)

bootplot_rep(0,95)
for(i in 5:100){bootplot[i-4]_sqrt(var(boot.res[1:i,7]))}
 plot(5:100,bootplot,
     xlab="Number of bootstrap samples",
     ylab="Bootstrap standard errors",
     main="NGRWCF IN MT1",type="l",ylim=c(0.05,.2))
     points(100,MARS.var[4],pch=1)
     points(100,NLCD.var[4],pch=2)
     points(100,SRS.var[4],pch=3)

 dev.off()
```

Appendix C-4. p4.strat

```
##################
print("### p4.strat: program to produce population estimates under
            dif stratification schemes ###")
##################


#######
print("# specify inputs and outputs")
#######

out.res_paste(results.path,"resultstot.txt",sep="")

#######
print("# set up data")
#######

dat.toty_datd.all[,c("FORTYP.2","TWN","NVOLTOT","NGRWCF")]
dat.toty$FOREST_dat.toty$FORTYP.2
dat.toty$TIMBER_dat.toty$TWN
dat.toty$TIMBER[dat.toty$TIMBER!=2]_0
dat.toty$TIMBER[dat.toty$TIMBER==2]_1
if(eco !="mt2"){
  dat.totx_datd.all[,c("EASTING","NORTHING","ELEV.1K","TRASP.1K",
      "SLOPE.1K","AVH.1","AVH.2","AVH.3","AVH.4","AVH.5","NDVI",
      "NLCD")]
}
if(eco =="mt2"){
  dat.totx_datd.all[,c("EASTING","NORTHING","ELEV.1K","TRASP.1K",
      "AVH.1","AVH.2","AVH.3","AVH.4","AVH.5","NDVI",
      "NLCD")]
}


#######
print("# srs estimates")
#######

res0_t(as.matrix(f.srs(dat.toty$FOREST,type=0)))
  res_data.frame(eco,"SRS","FOREST",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")

res0_t(as.matrix(f.srs(dat.toty$TIMBER,type=0)))
  res_data.frame(eco,"SRS","TIMBER",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")

res0_t(as.matrix(f.srs(dat.toty$NVOLTOT,type=1)))
  res_data.frame(eco,"SRS","NVOLTOT",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")

res0_t(as.matrix(f.srs(dat.toty$NGRWCF,type=1)))
  res_data.frame(eco,"SRS","NGRWCF",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")


#######
print("# NLCD estimates")
#######

str.nlcd_f.nlcdmod(dat.totx$NLCD)
wts0_as.numeric(table(NLCD.map))
wts_wts0/sum(wts0)

res0_t(as.matrix(f.str(y=dat.toty$FOREST,strata=str.nlcd,
      strcodes=c(0,1),Wh=wts,type=0)))
  res_data.frame(eco,"NLCD","FOREST",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")

res0_t(as.matrix(f.str(y=dat.toty$TIMBER,strata=str.nlcd,
      strcodes=c(0,1),Wh=wts,type=0)))
  res_data.frame(eco,"NLCD","TIMBER",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")

res0_t(as.matrix(f.str(y=dat.toty$NVOLTOT,strata=str.nlcd,
      strcodes=c(0,1),Wh=wts,type=1)))
  res_data.frame(eco,"NLCD","NVOLTOT",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")

res0_t(as.matrix(f.str(y=dat.toty$NGRWCF,strata=str.nlcd,
      strcodes=c(0,1),Wh=wts,type=1)))
  res_data.frame(eco,"NLCD","NGRWCF",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")


#######
print("# MARS estimates")
#######

indics_data.frame(class.ind(dat.totx$NLCD))
   posit_row(as.matrix(names(dat.totx)))[names(dat.totx)=="NLCD"]
   dat.totxa_dat.totx[,-posit]
   dat.totxa_data.frame(dat.totxa,indics)

str2.mars_round(predict.mars(FORTYP2.mod,dat.totxa))
   str2.mars[str2.mars<0]_0
   str2.mars[str2.mars>1]_1
str.mars_round(predict.mars(FORTYP3.mod,dat.totxa))
   str.mars[str.mars<0]_0
   str.mars[str.mars>1]_1
   str.mars_str.mars+1
   str.mars[str2.mars==0]_0

wts0_as.numeric(table(FORTYP3.map))
#wts0_as.numeric(table(str.mars))
wts_wts0/sum(wts0)

res0_t(as.matrix(f.str(dat.toty$FOREST,strata=str.mars,
      strcodes=c(0,1,2),Wh=wts,type=0)))
  res_data.frame(eco,"MARS","FOREST",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")

res0_t(as.matrix(f.str(dat.toty$TIMBER,strata=str.mars,
      strcodes=c(0,1,2),Wh=wts,type=0)))
  res_data.frame(eco,"MARS","TIMBER",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")

res0_t(as.matrix(f.str(y=dat.toty$NVOLTOT,strata=str.mars,
      strcodes=c(0,1,2),Wh=wts,type=1)))
  res_data.frame(eco,"MARS","NVOLTOT",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")

res0_t(as.matrix(f.str(y=dat.toty$NGRWCF,strata=str.mars,
      strcodes=c(0,1,2),Wh=wts,type=1)))
  res_data.frame(eco,"MARS","NGRWCF",res0)
  write.table(res,out.res,append=T,dimnames.write=F,sep="\t")
```

Appendix C-5. p5.results

```
#######################################################################
print("###### p5.results: generate graphical results ######")
#######################################################################

#######
print("# specify inputs and outputs")
#######
f5.path_"/export/jerry2/gretchen/diss/output/f5/"
in1.res_paste(results.path,"results1.txt",sep="")
in0.res_paste(results.path,"results0.txt",sep="")
intot.res_paste(results.path,"resultstot.txt",sep="")
intest.res_paste(results.path,"test.txt",sep="")
res1.ps_paste(f5.path,"fig8_11.ps",sep="")
res0.ps_paste(f5.path,"fig3_5.ps",sep="")
restot.ps_paste(f5.path,"fig15_16.ps",sep="")
test.ps_paste(f5.path,"fig2.ps",sep="")
rank0.ps_paste(f5.path,"fig6.ps",sep="")
rank1.ps_paste(f5.path,"fig13.ps",sep="")

#######
print("# read and reformat data")
#######

res1_read.table(in1.res,header=T,sep="\t",as.is=T)
res0_read.table(in0.res,header=T,sep="\t",as.is=T)
restot_read.table(intot.res,header=T,sep="\t",as.is=T)
#restest_read.table(intest.res,header=T,sep="\t",as.is=T)
restest$tech_c("LM","GAM","CART","MARS","ANN")

###########
# print("# standardize output")
###########
resall_res1

res1$rmse[res1$eco=="az1" & res1$Yname=="BIOTOT"]_
   res1$rmse[res1$eco=="az1" &
res1$Yname=="BIOTOT"]/mean(datc.all.az1$BIOTOT)
res1$rmse[res1$eco=="az1" & res1$Yname=="CRCOV"]_
   res1$rmse[res1$eco=="az1" &
res1$Yname=="CRCOV"]/mean(datc.all.az1$CRCOV)
res1$rmse[res1$eco=="az1" & res1$Yname=="STAGECL"]_
   res1$rmse[res1$eco=="az1" &
res1$Yname=="STAGECL"]/mean(datc.all.az1$STAGECL)
res1$rmse[res1$eco=="az1" & res1$Yname=="QMDALL"]_
   res1$rmse[res1$eco=="az1" &
res1$Yname=="QMDALL"]/mean(datc.all.az1$QMDALL)

res1$rmse[res1$eco=="az2" & res1$Yname=="BIOTOT"]_
   res1$rmse[res1$eco=="az2" &
res1$Yname=="BIOTOT"]/mean(datc.all.az2$BIOTOT)
res1$rmse[res1$eco=="az2" & res1$Yname=="CRCOV"]_
   res1$rmse[res1$eco=="az2" &
res1$Yname=="CRCOV"]/mean(datc.all.az2$CRCOV)
```

```
res1$rmse[res1$eco=="az2" & res1$Yname=="STAGECL"]_
    res1$rmse[res1$eco=="az2" &
res1$Yname=="STAGECL"]/mean(datc.all.az2$STAGECL)
res1$rmse[res1$eco=="az2" & res1$Yname=="QMDALL"]_
    res1$rmse[res1$eco=="az2" &
res1$Yname=="QMDALL"]/mean(datc.all.az2$QMDALL)

res1$rmse[res1$eco=="mt1" & res1$Yname=="BIOTOT"]_
    res1$rmse[res1$eco=="mt1" &
res1$Yname=="BIOTOT"]/mean(datc.all.mt1$BIOTOT)
res1$rmse[res1$eco=="mt1" & res1$Yname=="CRCOV"]_
    res1$rmse[res1$eco=="mt1" &
res1$Yname=="CRCOV"]/mean(datc.all.mt1$CRCOV)
res1$rmse[res1$eco=="mt1" & res1$Yname=="STAGECL"]_
    res1$rmse[res1$eco=="mt1" &
res1$Yname=="STAGECL"]/mean(datc.all.mt1$STAGECL)
res1$rmse[res1$eco=="mt1" & res1$Yname=="QMDALL"]_
    res1$rmse[res1$eco=="mt1" &
res1$Yname=="QMDALL"]/mean(datc.all.mt1$QMDALL)

res1$rmse[res1$eco=="mt2" & res1$Yname=="BIOTOT"]_
    res1$rmse[res1$eco=="mt2" &
res1$Yname=="BIOTOT"]/mean(datc.all.mt2$BIOTOT)
res1$rmse[res1$eco=="mt2" & res1$Yname=="CRCOV"]_
    res1$rmse[res1$eco=="mt2" &
res1$Yname=="CRCOV"]/mean(datc.all.mt2$CRCOV)
res1$rmse[res1$eco=="mt2" & res1$Yname=="STAGECL"]_
    res1$rmse[res1$eco=="mt2" &
res1$Yname=="STAGECL"]/mean(datc.all.mt2$STAGECL)
res1$rmse[res1$eco=="mt2" & res1$Yname=="QMDALL"]_
    res1$rmse[res1$eco=="mt2" &
res1$Yname=="QMDALL"]/mean(datc.all.mt2$QMDALL)

res1$rmse[res1$eco=="ut1" & res1$Yname=="BIOTOT"]_
    res1$rmse[res1$eco=="ut1" &
res1$Yname=="BIOTOT"]/mean(datc.all.ut1$BIOTOT)
res1$rmse[res1$eco=="ut1" & res1$Yname=="CRCOV"]_
    res1$rmse[res1$eco=="ut1" &
res1$Yname=="CRCOV"]/mean(datc.all.ut1$CRCOV)
res1$rmse[res1$eco=="ut1" & res1$Yname=="STAGECL"]_
    res1$rmse[res1$eco=="ut1" &
res1$Yname=="STAGECL"]/mean(datc.all.ut1$STAGECL)
res1$rmse[res1$eco=="ut1" & res1$Yname=="QMDALL"]_
    res1$rmse[res1$eco=="ut1" &
res1$Yname=="QMDALL"]/mean(datc.all.ut1$QMDALL)

res1$rmse[res1$eco=="ut2" & res1$Yname=="BIOTOT"]_
    res1$rmse[res1$eco=="ut2" &
res1$Yname=="BIOTOT"]/mean(datc.all.ut2$BIOTOT)
res1$rmse[res1$eco=="ut2" & res1$Yname=="CRCOV"]_
    res1$rmse[res1$eco=="ut2" &
res1$Yname=="CRCOV"]/mean(datc.all.ut2$CRCOV)
res1$rmse[res1$eco=="ut2" & res1$Yname=="STAGECL"]_
    res1$rmse[res1$eco=="ut2" &
res1$Yname=="STAGECL"]/mean(datc.all.ut2$STAGECL)
```

```
res1$rmse[res1$eco=="ut2" & res1$Yname=="QMDALL"]_
   res1$rmse[res1$eco=="ut2" &
res1$Yname=="QMDALL"]/mean(datc.all.ut2$QMDALL)


#######
print("# res1...plot results")
#######
postscript(file=res1.ps,onefile=T,width=8.5,height=11,horizontal=F)
resall_res1
res1$eco_ordered(res1$eco,rev(c("mt1","mt2","ut1","ut2","az1","az2")))
# res1$tech_ordered(res1$tech,rev(c("ANN","MARS","GAM","CART","NLCD")))

#rmse

res1$tech_f.order(res1$tech,-res1$rmse)
dotplot(tech~rmse|Yname*eco,data=res1,layout=c(4,6,1))

#pwi

res1$tech_f.order(res1$tech,res1$pwi)
dotplot(tech~pwi|Yname*eco,data=res1,layout=c(4,6,1))


#rho

res1$tech_f.order(res1$tech,res1$rho)
dotplot(tech~rho|Yname*eco,data=res1,layout=c(4,6,1))


#runtime

res1$tech_f.order(res1$tech,-res1$runtime)
dotplot(tech~runtime|Yname*eco,data=res1,layout=c(4,6,1))


dev.off()

#######
print("# res0...plot results")
#######
postscript(file=res0.ps,onefile=T,width=8.5,height=11,horizontal=F)
resall_res0
res0$eco_ordered(res0$eco,rev(c("mt1","mt2","ut1","ut2","az1","az2")))
#res0$tech_ordered(res0$tech,rev(c("ANN","MARS","GAM","CART","NLCD")))

#pcc

res0$tech_f.order(res0$tech,res0$pcc)
dotplot(tech~pcc|Yname*eco,data=res0,layout=c(2,6,1),aspect=.5)


#prod0

#res0$tech_f.order(res0$tech,res0$prod0)
```

```
#dotplot(tech~prod0|Yname*eco,data=res0,layout=c(2,6,1),aspect=.5)


#prod1

#res0$tech_f.order(res0$tech,res0$prod1)
#dotplot(tech~prod1|Yname*eco,data=res0,layout=c(2,6,1),aspect=.5)


#kappa

res0$tech_f.order(res0$tech,res0$kappa)
dotplot(tech~kappa|Yname*eco,data=res0,layout=c(2,6,1),aspect=.5)

#runtime

res0$tech_f.order(res0$tech,-res0$runtime)
dotplot(tech~runtime|Yname*eco,data=res0,layout=c(2,6,1),aspect=.5)


dev.off()


#######
print("# create and plot rank files - 1")
#######

postscript(file=rank1.ps,onefile=T,width=8.5,height=11,horizontal=F)

n_nrow(res1)
rrho_rep(0,n)
rpwi_rep(0,n)
rrmse_rep(0,n)
rruntime_rep(0,n)
for (i in 1:(n/5)){
  rrho[(i*5-4):(i*5)]_rank(res1$rho[(i*5-4):(i*5)])
  rpwi[(i*5-4):(i*5)]_rank(res1$pwi[(i*5-4):(i*5)])
  rrmse[(i*5-4):(i*5)]_rank(-(res1$rmse[(i*5-4):(i*5)]))
  rruntime[(i*5-4):(i*5)]_rank(-(res1$runtime[(i*5-4):(i*5)]))
}
rhomat_matrix(rrho,ncol=5,byrow=T)
rankrho_apply(rhomat,2,mean)
pwimat_matrix(rpwi,ncol=5,byrow=T)
rankpwi_apply(pwimat,2,mean)
rmsemat_matrix(rrmse,ncol=5,byrow=T)
rankrmse_apply(rmsemat,2,mean)
runmat_matrix(rruntime,ncol=5,byrow=T)
rankrun_apply(runmat,2,mean)

tech_rep(c("NLCD","GAM","CART","MARS","ANN"),4)
perfm_c(rep("RHO",5),rep("PWI",5),rep("RMSE",5),rep("RUNTIME",5))
rank1_data.frame(tech,perfm,c(rankrho,rankpwi,rankrmse,rankrun))
names(rank1)_c("tech","perfm","rank")
rank1$tech_f.order(rank1$tech,rank1$rank)
```

```
#rank1$tech_ordered(rank1$tech,rev(c("ANN","MARS","GAM","CART","NLCD"))
)
dotplot(tech~rank|perfm,data=rank1,layout=c(4,1,1),aspect=.75)

dev.off()

#######
print("# create and plot rank files - 0")
#######

postscript(file=rank0.ps,onefile=T,width=8.5,height=11,horizontal=F)

n_nrow(res0)
rpcc_rep(0,n)
rkappa_rep(0,n)
rprod0_rep(0,n)
rprod1_rep(0,n)
rruntime_rep(0,n)
for (i in 1:(n/5)){
  rpcc[(i*5-4):(i*5)]_rank(res0$pcc[(i*5-4):(i*5)])
  rkappa[(i*5-4):(i*5)]_rank(res0$kappa[(i*5-4):(i*5)])
#  rprod0[(i*5-4):(i*5)]_rank(res0$prod0[(i*5-4):(i*5)])
#  rprod1[(i*5-4):(i*5)]_rank(res0$prod1[(i*5-4):(i*5)])
  rruntime[(i*5-4):(i*5)]_rank(-(res0$runtime[(i*5-4):(i*5)]))
}
pccmat_matrix(rpcc,ncol=5,byrow=T)
rankpcc_apply(pccmat,2,mean)
kappamat_matrix(rkappa,ncol=5,byrow=T)
rankkappa_apply(kappamat,2,mean)
#prod0mat_matrix(rprod0,ncol=5,byrow=T)
#rankprod0_apply(prod0mat,2,mean)
#prod1mat_matrix(rprod1,ncol=5,byrow=T)
#rankprod1_apply(prod1mat,2,mean)
runmat_matrix(rruntime,ncol=5,byrow=T)
rankrun_apply(runmat,2,mean)

#tech_rep(c("NLCD","GAM","CART","MARS","ANN"),5)
tech_rep(c("NLCD","GAM","CART","MARS","ANN"),3)
#perfm_c(rep("PCC",5),rep("KAPPA",5),rep("PROD0",5),rep("PROD1",5),
#                   rep("RUNTIME",5))
perfm_c(rep("PCC",5),rep("KAPPA",5),rep("RUNTIME",5))
rank0_data.frame(tech,perfm,c(rankpcc,rankkappa,rankrun))
names(rank0)_c("tech","perfm","rank")
rank0$tech_f.order(rank0$tech,rank0$rank)
#rank0$tech_ordered(rank0$tech,rev(c("ANN","MARS","GAM","CART","NLCD"))
)
#dotplot(tech~rank|perfm,data=rank0,layout=c(5,1,1))
dotplot(tech~rank|perfm,data=rank0,layout=c(3,1,1))

dev.off


#######
print("# restot...plot results")
#######
```

```
postscript(file=restot.ps,onefile=T,width=8.5,height=11,horizontal=F)
restot$sesrs_rep(0,nrow(restot))
for (i in 1:6){
   se_restot$se[((i*12)-11):((i*12)-8)]
   restot$sesrs[((i*12)-11):(i*12)]_rep(se,3)}
restot$ratio_restot$sesrs/restot$se
restot$eco_ordered(restot$eco,rev(c("mt1","mt2","ut1","ut2","az1","az2"
)))
restot$popmean_ordered(restot$popmean,c("FOREST","TIMBER","NVOLTOT","NG
RWCF"))
dotplot(strat~pcntse|popmean*eco,data=restot,layout=c(4,6,1))
dotplot(strat~ratio|popmean*eco,data=restot,layout=c(4,6,1))
dev.off()

#######
print("# res0...plot results")
#######

postscript(test.ps,onefile=T,width=8.5,height=11,horizontal=F)
par(mfrow=c(2,2))
restest$tech_f.order(restest$tech,-restest$rmse)
dotplot(tech~rmse,data=restest,main="RMSE")
restest$tech_f.order(restest$tech,-restest$runtime)
dotplot(tech~runtime,data=restest,main="Run Time")
restest$tech_f.order(restest$tech,restest$pwi)
dotplot(tech~pwi,data=restest,main="PWI")
restest$tech_f.order(restest$tech,restest$rho)
dotplot(tech~rho,data=restest,main="RHO")
dev.off()
```

Appendix D: Results Tables

Table D-1. Results from Predictive Mapping of the Discrete Variables.

Eco	Response	Technique	RunTime	PCC	(se)	Kappa	(se)
az1	FORTYP.2	NLCD	1	0.711	0.025	0.412	0.05
az1	FORTYP.2	GAM	723	0.719	0.024	0.426	0.05
az1	FORTYP.2	CART	260	0.658	0.026	0.302	0.053
az1	FORTYP.2	MARS	76	0.716	0.024	0.42	0.051
az1	FORTYP.2	ANN	194	0.696	0.025	0.374	0.053
az1	FORTYP.3	NLCD	0	0.603	0.035	0.167	0.09
az1	FORTYP.3	GAM	579	0.854	0.025	0.706	0.05
az1	FORTYP.3	CART	233	0.824	0.027	0.644	0.055
az1	FORTYP.3	MARS	56	0.864	0.024	0.726	0.049
az1	FORTYP.3	ANN	178	0.869	0.024	0.737	0.048
az2	FORTYP.2	NLCD	0	0.894	0.017	0.446	0.078
az2	FORTYP.2	GAM	683	0.912	0.015	0.642	0.06
az2	FORTYP.2	CART	250	0.894	0.017	0.592	0.061
az2	FORTYP.2	MARS	102	0.903	0.016	0.529	0.072
az2	FORTYP.2	ANN	265	0.876	0.018	0.508	0.066
az2	FORTYP.3	NLCD	0	0.66	0.067	0.248	0.187
az2	FORTYP.3	GAM	661	0.94	0.034	0.694	0.166
az2	FORTYP.3	CART	224	0.9	0.042	0	0.283
az2	FORTYP.3	MARS	8	0.96	0.028	0.811	0.129
az2	FORTYP.3	ANN	17	0.92	0.038	0.31	0.275
mt1	FORTYP.2	NLCD	0	0.879	0.015	0.643	0.042
mt1	FORTYP.2	GAM	829	0.913	0.013	0.673	0.046
mt1	FORTYP.2	CART	272	0.924	0.012	0.712	0.044
mt1	FORTYP.2	MARS	119	0.913	0.013	0.663	0.047
mt1	FORTYP.2	ANN	324	0.901	0.013	0.646	0.046
mt1	FORTYP.3	NLCD	0	0.421	0.024	0.025	0.085
mt1	FORTYP.3	GAM	573	0.615	0.024	-0.016	0.096
mt1	FORTYP.3	CART	284	0.634	0.024	0.073	0.077
mt1	FORTYP.3	MARS	120	0.615	0.024	0.027	0.078
mt1	FORTYP.3	ANN	210	0.624	0.024	0	0.098

mt2	FORTYP.2	NLCD	0	0.856	0.011	0.702	0.022
mt2	FORTYP.2	GAM	632	0.888	0.009	0.772	0.019
mt2	FORTYP.2	CART	463	0.856	0.011	0.702	0.022
mt2	FORTYP.2	MARS	359	0.884	0.01	0.762	0.02
mt2	FORTYP.2	ANN	463	0.886	0.01	0.766	0.019
mt2	FORTYP.3	NLCD	0	0.502	0.023	0.03	0.064
mt2	FORTYP.3	GAM	389	0.62	0.022	0.233	0.045
mt2	FORTYP.3	CART	204	0.502	0.023	0.03	0.064
mt2	FORTYP.3	MARS	107	0.645	0.022	0.285	0.044
mt2	FORTYP.3	ANN	204	0.602	0.022	0.207	0.045
ut1	FORTYP.2	NLCD	0	0.697	0.027	0.393	0.054
ut1	FORTYP.2	GAM	484	0.686	0.027	0.364	0.055
ut1	FORTYP.2	CART	258	0.697	0.027	0.393	0.054
ut1	FORTYP.2	MARS	91	0.703	0.027	0.401	0.054
ut1	FORTYP.2	ANN	190	0.69	0.027	0.373	0.055
ut1	FORTYP.3	NLCD	0	0.686	0.037	0.299	0.079
ut1	FORTYP.3	GAM	605	0.818	0.031	0.574	0.07
ut1	FORTYP.3	CART	227	0.818	0.031	0.595	0.067
ut1	FORTYP.3	MARS	32	0.818	0.031	0.574	0.07
ut1	FORTYP.3	ANN	183	0.836	0.029	0.623	0.067
ut2	FORTYP.2	NLCD	1	0.72	0.023	0.416	0.047
ut2	FORTYP.2	GAM	421	0.71	0.023	0.347	0.05
ut2	FORTYP.2	CART	277	0.72	0.023	0.416	0.047
ut2	FORTYP.2	MARS	109	0.712	0.023	0.35	0.051
ut2	FORTYP.2	ANN	230	0.702	0.023	0.338	0.05
ut2	FORTYP.3	NLCD	1	0.627	0.031	0.152	0.075
ut2	FORTYP.3	GAM	666	0.876	0.021	0.737	0.044
ut2	FORTYP.3	CART	248	0.884	0.02	0.756	0.042
ut2	FORTYP.3	MARS	73	0.896	0.019	0.777	0.041
ut2	FORTYP.3	ANN	216	0.863	0.022	0.705	0.047

Table D-2. Results from Predictive Mapping of the Continuous Variables

Eco	Resonse	Technique	Run Time	Rho	PWI 25%	RMSE
az1	BIOTOT	NLCD	1	0.319	0.251	2583.157
az1	BIOTOT	GAM	284	0.592	0.307	2228.626
az1	BIOTOT	CART	235	0.459	0.276	2509.072
az1	BIOTOT	MARS	39	0.562	0.296	2278.476
az1	BIOTOT	ANN	130	0.631	0.276	2124.114
az1	CRCOV	NLCD	1	0.302	0.412	7.763
az1	CRCOV	GAM	375	0.568	0.467	6.679
az1	CRCOV	CART	235	0.516	0.442	6.977
az1	CRCOV	MARS	52	0.56	0.482	6.709
az1	CRCOV	ANN	140	0.549	0.442	6.788
az1	STAGECL	NLCD	1	0.058	0.482	17.77
az1	STAGECL	GAM	342	0.262	0.528	17.585
az1	STAGECL	CART	245	0.27	0.543	17.248
az1	STAGECL	MARS	47	0.259	0.538	17.882
az1	STAGECL	ANN	151	0.206	0.558	17.948
az1	QMDALL	NLCD	1	0.158	0.477	1.441
az1	QMDALL	GAM	512	0.249	0.477	1.408
az1	QMDALL	CART	263	0.162	0.497	1.448
az1	QMDALL	MARS	56	0.139	0.482	1.478
az1	QMDALL	ANN	135	0.265	0.482	1.401
az2	BIOTOT	NLCD	2	0.403	0.28	1887.105
az2	BIOTOT	GAM	388	0.777	0.2	1267.915
az2	BIOTOT	CART	210	0.605	0.26	1600.747
az2	BIOTOT	MARS	16	0.32	0.18	2662.053
az2	BIOTOT	ANN	20	0.611	0.36	2733.205
az2	CRCOV	NLCD	0	0.411	0.24	4.356
az2	CRCOV	GAM	396	0.483	0.28	4.18
az2	CRCOV	CART	214	0.296	0.18	4.591
az2	CRCOV	MARS	12	0.241	0.28	5.651
az2	CRCOV	ANN	20	0.561	0.24	3.978
az2	STAGECL	NLCD	0	-0.077	0.56	13.388
az2	STAGECL	GAM	419	0	0.36	14.627
az2	STAGECL	CART	223	0	0.54	13.241

az2	STAGECL	MARS	16	0.025	0.44	14.919
az2	STAGECL	ANN	24	0.021	0.5	16.008
az2	QMDALL	NLCD	1	0.278	0.52	0.634
az2	QMDALL	GAM	452	0.265	0.42	0.637
az2	QMDALL	CART	236	.278	0.5	0.651
az2	QMDALL	MARS	12	0.201	0.54	0.682
az2	QMDALL	ANN	23	0.303	0.58	0.651
mt1	BIOTOT	NLCD	1	0.276	0.23	7.3415
mt1	BIOTOT	GAM	358	0.335	0.251	7.1921
mt1	BIOTOT	CART	262	0.242	0.215	7.4585
mt1	BIOTOT	MARS	94	0.339	0.256	7.1934
mt1	BIOTOT	ANN	215	0.291	0.239	7.3073
mt1	CRCOV	NLCD	1	0.302	0.23	11.769
mt1	CRCOV	GAM	450	0.386	0.268	11.377
mt1	CRCOV	CART	272	0.382	0.256	11.402
mt1	CRCOV	MARS	114	0.302	0.23	11.769
mt1	CRCOV	ANN	213	0.404	0.249	11.305
mt1	STAGECL	NLCD	1	0.175	0.179	27.465
mt1	STAGECL	GAM	458	0.406	0.196	25.44
mt1	STAGECL	CART	283	0.255	0.182	27.346
mt1	STAGECL	MARS	97	0.391	0.194	25.835
mt1	STAGECL	ANN	215	0.291	0.146	27.355
mt1	QMDALL	NLCD	3	0.049	0.419	2.178
mt1	QMDALL	GAM	729	0.248	0.431	2.117
mt1	QMDALL	CART	323	.049	0.419	2.179
mt1	QMDALL	MARS	87	0.189	0.428	2.146
mt1	QMDALL	ANN	300	-0.014	0.39	2.971
mt2	BIOTOT	NLCD	1	0.392	0.28	8.903
mt2	BIOTOT	GAM	282	0.433	0.29	8.709
mt2	BIOTOT	CART	261	0.347	0.294	9.216
mt2	BIOTOT	MARS	74	0.411	0.304	8.814
mt2	BIOTOT	ANN	211	0.439	0.296	8.676
mt2	CRCOV	NLCD	0	0.365	0.378	7.029
mt2	CRCOV	GAM	204	0.416	0.369	6.861
mt2	CRCOV	CART	269	0.375	0.363	7.009
mt2	CRCOV	MARS	97	0.382	0.392	7.01

mt2	CRCOV	ANN	214	0.383	0.38	6.977
mt2	STAGECL	NLCD	1	0.218	0.224	26.032
mt2	STAGECL	GAM	319	0.527	0.22	22.645
mt2	STAGECL	CART	260	0.441	0.208	24.126
mt2	STAGECL	MARS	80	0.503	0.229	23.091
mt2	STAGECL	ANN	204	0.511	0.188	22.892
mt2	QMDALL	NLCD	1	0.046	0.457	1.345
mt2	QMDALL	GAM	309	0.261	0.478	1.303
mt2	QMDALL	CART	279	0.067	0.424	1.385
mt2	QMDALL	MARS	114	0.241	0.418	1.339
mt2	QMDALL	ANN	248	0.149	0.476	1.406
ut1	BIOTOT	NLCD	1	0.025	0.27	11.71
ut1	BIOTOT	GAM	329	0.059	0.22	11.902
ut1	BIOTOT	CART	230	0.029	0.296	12.075
ut1	BIOTOT	MARS	27	0.075	0.283	12.226
ut1	BIOTOT	ANN	121	0.081	0.264	12.659
ut1	CRCOV	NLCD	0	0.249	0.233	8.075
ut1	CRCOV	GAM	384	0.358	0.277	7.709
ut1	CRCOV	CART	235	0.216	0.283	8.161
ut1	CRCOV	MARS	32	0.298	0.258	7.981
ut1	CRCOV	ANN	131	0.272	0.22	8.048
ut1	STAGECL	NLCD	1	0.094	0.403	20.014
ut1	STAGECL	GAM	385	0.424	0.384	18.411
ut1	STAGECL	CART	254	0.267	0.34	21.918
ut1	STAGECL	MARS	36	0.381	0.384	19.366
ut1	STAGECL	ANN	132	0.385	0.377	19.003
ut1	QMDALL	NLCD	2	0.072	0.327	1.595
ut1	QMDALL	GAM	640	0.205	0.365	1.564
ut1	QMDALL	CART	299	0.168	0.327	1.585
ut1	QMDALL	MARS	32	0.131	0.321	1.679
ut1	QMDALL	ANN	142	0.252	0.421	1.555

ut2	BIOTOT	NLCD	1	0.201	0.225	10.763
ut2	BIOTOT	GAM	327	0.551	0.281	9.131
ut2	BIOTOT	CART	236	0.456	0.261	9.807
ut2	BIOTOT	MARS	77	0.556	0.221	9.106
ut2	BIOTOT	ANN	171	0.586	0.257	8.884
ut2	CRCOV	NLCD	1	0.294	0.329	9.201
ut2	CRCOV	GAM	374	0.482	0.361	8.455
ut2	CRCOV	CART	242	0.339	0.313	9.302
ut2	CRCOV	MARS	70	0.441	0.357	8.718
ut2	CRCOV	ANN	150	0.483	0.369	8.462
ut2	STAGECL	NLCD	1	0.038	0.47	22.172
ut2	STAGECL	GAM	363	0.106	0.458	22.723
ut2	STAGECL	CART	235	0.007	0.43	23.176
ut2	STAGECL	MARS	57	0.027	0.458	25.201
ut2	STAGECL	ANN	186	-0.047	0.45	28.242
ut2	QMDALL	NLCD	2	0.208	0.498	1.825
ut2	QMDALL	GAM	370	0.38	0.522	1.73
ut2	QMDALL	CART	237	0.294	0.546	1.926
ut2	QMDALL	MARS	71	0.373	0.502	1.744
ut2	QMDALL	ANN	161	0.274	0.474	1.973

CURRICULUM VITAE

GRETCHEN GENGENBACH MOISEN

80 North Wolcott Street, SLC, UT 84103
PH: 801/363-7288, EMAIL: gmoisen@fs.fed.us

EDUCATION

2000. Ph.D. (Mathematical Sciences), Utah State University, Logan, UT.

1990. M.S. (Statistics), Utah State University, Logan, UT.

1982. B.S. (Forestry), University of New Hampshire, Durham, N.H.

PROFESSIONAL EXPERIENCE

1989 - Present. Research Scientist, US Forest Service, Ogden, UT.

1982 - 1989. Forester, US Forest Service, Ogden, UT.

TEACHING EXPERIENCE

1995 - 1997. Co-lecturer, Utah State University, Logan, UT.
Topics in Spatial Ecology Seminar
Statistical Models in Ecology Seminar
Seminar in Spatial Analysis of Ecological Data

AWARDS AND OTHER HONORS

2000. USDA National Director's Award for excellence in FIA.

2000. USDA Merit Award for developing cost-effective estimation strategies for forest inventories of the Interior West, Ogden, UT.

1994. Graduate Student Researcher of the Year, College of Science, Utah State University, Logan, UT.

1994. Outstanding Student Award for Research, Department of Mathematics and Statistics, Utah State University, Logan, UT.

1991. Outstanding Student Award for Research, Department of Mathematics and Statistics, Utah State University, Logan, UT.

1990. Outstanding Student Award for Academics, Department of Mathematics and Statistics, Utah State University, Logan, UT.

1990. USDA Merit Award for contributions to the 1990 RPA assessment, Ogden, UT.

1985. USDA Merit Award for exceptional supervisory performance for the Arizona and New Mexico forest inventories, Ogden, UT.

1982. Graduated Magna Cum Laude, University of New Hampshire, Durham, N.H.

GRANTS

1999-2003 Spatial products from forest inventories: development of a modelling, display, and analysis environment (PI) , US Forest Service, $225,000.

1993-1996 Refinement of the Great Basin Gap Analysis base vegetation map for use in vegetation inventories (Co-PI), US Forest Service, $94,500

PROFESSIONAL SERVICE

1999 - Present Member of the US Forest Service National Statistical Estimation Band.

1999 - Present. Participant in development phase of US Forest Service Northern Region Ecological System Inventory, Analysis, and Monitoring effort , Missoula, MT.

1999. Participant in the North Central Remote Sensing Workshop designed to coordinate research efforts conducted under US Forest Service Remote Sensing grant funding, Minneapolis, MN.

1999. Participant in the Biometrician's Workshop to design estimation strategies to meet 1998 Farm Bill legislative requirements, New Orleans, LA.

1998. Participant in review of US Forest Service Northern Region Strategic Ecological System Inventory, Analysis, and Monitoring Charter , Missoula, MT.

1998. Panelist for the University Consortium on Geographic Information Science, Research Priority Panel on Uncertainty, Park City, UT.

1997 - Present. Member of the US Forest Service National Remote Sensing Band.

1996. Reviewer of statistical methods proposed for the Annual Forest Inventory System pilot study in Aspen-Birch Unit of Minnesota, Fort Collins, CO.

1995 - Present Reviewer for Ann Arbor Press, Biometrics, Cnanadian Journal of Forest Research, Forest Science, Journal of Computational and Graphical Statistics, Remote Sensing and Environment.

1994. Facilitator at National Analysts Conference, Forest Inventory and Analysis, Denver, CO.

1994. Participant in National Biological Service Gap Analysis Program workshop on accuracy assessment of broad-scale maps, Santa Barbara, CA.

1992 - Present. Participant in 6 peer review panels of other federal scientists.

1992 - 1993. Member of the US Forest Service Committee on Sampling Condition Boundaries.

1989 - Present. Reviewer of numerous US Forest Service Research committee recommendations and program charters.

PUBLICATIONS

2001 Frescino, T.S., G.G. Moisen, and T.C. Edwards. In Press. Development and use of Forest Inventory and Analysis (FIA) map products. Proceedings of the 8th Biennial Remote Sensing Applications Conference, CD-ROM format.

2001 Reutebuch, S., D. Patterson, B. Bollenbacher, and G. G. Moisen. In press. Large-scale, GPS/IMU-controlled aerial photography for forest inventory and monitoring. Procedings of the 8th Biennial Remote Sensing Applications Conference , CD-ROM format.

2001 Frescino, T. S., T. C. Edwards, Jr., and G. G. Moisen. In press. Modelling spatially explicit structural attributes using generalized additive models. Journal of Vegetation Science.

2000 Moisen, G.G. 2000. Comparing nonlinear and nonparametric modelling techniques for stratification and mapping in forest inventories of the Interior Western USA. Ph.D. dissertation, Utah State University.

2000. Moisen, G. G., R. D. Cutler, and T. C. Edwards, Jr. Generalized linear mixed models for analyzing error in a satellite-based vegetation map of Utah. *In* H. T. Mowrer and R. Congalton, editors. Quantifying Spatial Uncertainty in Natural Resources: Theory and Application for GIS and Remote Sensing. Ann Arbor Press, Chelsea, Michigan, p. 37-44.

1999. Moisen, G. G., T. C. Edwards, Jr., and T. S. Frescino. Expanded data, models and applications in a forest inventory of Utah, USA. In C. Aguirre-Bravo and C. R. Franco, compilers. North American Science Symposium: Toward a Unified Framework for Inventorying and Monitoring Forest Ecosystem Resources. Guadalajara, Mexico (November 2-6, 1998). Proceedings RMRS-P-12. US Department of Agriculture, Forest Service, Rocky Mountain Research Station, Fort Collins, CO USA. p 212-218.

1999. Moisen, G. G. and T. C. Edwards, Jr. Use of generalized linear models and digital data in a forest inventory of Utah. Journal of Agricultural, Biological and Environmental Statistics 4(4):372-390.

1998. Edwards, T. C., Jr., G. G. Moisen, and D. R. Cutler. Assessing map uncertainty in ecoregion-scale cover-maps. Remote Sensing of Environment 63:73-83.

1998. Moisen, G. G., T. C. Edwards, Jr., and D. Van Hooser. Merging regional forest inventory data with satellite-based information through nonlinear regression methods. In Thierry Ranchin and Lucien Wald, eds., proceedings of the Second International Conference on the Fusion of Earth Data, Sophia Antipolis, France, January 1998. p. 123-128.

1996. Moisen, G. G., R. D. Cutler, and T. C. Edwards, Jr. Generalized linear mixed models for analyzing error in a satellite-based vegetation map of Utah. Pages 459-466 *in* H. T. Mowrer, R. L. Czaplewski, and R. H. Hamre, editors. Spatial accuracy assessment in natural resources and environmental sciences, USDA Forest Service General Technical Report RM-GTR-277.

1996. Edwards, T. C., Jr., E. Deshler, D. Foster, and G. G. Moisen. Adequacy of wildlife habitat relation models for estimating spatial distributions of terrestrial vertebrates. Conservation Biology 10:263-270.

1995. Moisen, G. G., A. R. Stage, and J. D. Born. 1995. Point sampling with disjunct support near population boundaries. Forest Science Monograph 31:62-82.

1994. Moisen, G. G., T. C. Edwards, Jr., and D. R. Cutler. Spatial sampling to assess classification accuracy of remotely sensed data. Pages 161-178 *in* J. Brunt, S. S. Stafford, and W. K. Michener, editors. Environmental Information Management and Analysis: Ecosystem to Global Scales, Taylor and Francis, Limited.

1993. Chojnacky, D. C. and G. G. Moisen. Converting wood volume to biomass for pinyon and juniper. Research Note INT-411. Ogden, Utah: USDA Forest Service, Intermountain Research Station. 5p.

1990. Moisen, G. G. 1990. Total and merchantable cubic-foot volume equations for six timber species in Western Montana and northern Idaho. Masters Report, Utah State University. 35p.

1988. Green, A. and G. G. Moisen. Timberland and woodland resources outside National Forests in northeastern Wyoming, 1984. Resource Bulletin INT-54. Ogden, UT: USDA Forest Service, Intermountain Research Station. 46p.

INVITED PRESENTATIONS

2000. Moisen, G. G, T. C. Edwards, Jr., and T. S. Frescino. An ecoregion mapping and delivery system in the Interior West. Presentation given at the NFS Forest and Range Directors Meeting, Phoenix, AZ.

2000. Frescino, T.S. and G.G. Moisen. Development, display, and analysis of forest inventory spatial products. Invited presentation to be given at the ESRI SouthWest User Group Conference 2000 in Moab, Utah.

2000. Moisen, G. G, T. S. Frescino, and R. P. Tymcio. Mapping and stratification efforts in the Interior West. Presentation to be given at the FIA - Remote Sensing Band Technical Workshop, Salt Lake City, UT.

2000. Moisen, G. G. Discussion of non-parametric and semi-parametric methods in surveys. Presentation to be given at the 2000 Joint Statistical Meetings of the American Statistical Association, Indianapolis, IN.

2000. Moisen, G. G. Comparing linear, nonlinear, and nonparametric models for stratification and prediction in forest inventories of the Intermountain West. Presentaton to be given at the 2000 Meetings of the Institute for Operations Research and the Management Sciences, Salt Lake City, UT.

1999. Edwards, T. C., Jr., G. G. Moisen, T. S. Frescino, and J. J. Lawler. Linking remotely-sensed environmental information with regional forest inventories to model forest sructure: methods and applications. Invited paper, International Union of Forest Research Organizations Conference on Remote Sensing and Forest Monitoring, Rogow, Poland, 6/2/99.

1999. Moisen, G. G. Development of spatial products from forest inventory data in the Interior West. Presentation before the US Forest Service North Central Remote Sensing Workshop, Minneapolis, MN.

1999. Moisen, G. G., R. P. Tymcio, and T. C. Edwards, Jr. Stratification and modelling experiences in the Interior West. Presentation before the Technical Review Committee on the status of the annual inventory system with focus on the South, New Orleans, LA.

1998. Moisen, G. G., T. C. Edwards, Jr., and T. S. Frescino. Expanding applications, data, and models in a forest inventory of Northern Utah, USA. Presentation at the North American symposium: Toward a Unified Framework for Inventorying and Monitoring Forest Ecosystem Resources, Guadalajara, Mexico.

1998. Edwards, T. C., Jr., and G. G. Moisen. Linking remotely-sensed environmental information with regional forest inventories to model forest structure. Presentation before the Department of Geography, San Diego State University, San Diego California

1998. Moisen, G. G., T. C. Edwards, Jr., and T. S. Frescino. Merging forest inventory data with satellite-based information in Utah. Presentation at the Conference on Environmental Monitoring Surveys Over Time, Seattle, WA.

1998. Moisen, G. G., T. C. Edwards, Jr., and T. S. Frescino. Experiences modelling forest inventory data in Utah. Seminar sponsored by the Department of Fisheries and Wildlife, Utah State University, Logan, UT.

1997. Moisen, G. G., T. C. Edwards, Jr., and D. R. Cutler. Broad-scale prediction of forest structure through generalized additive models. Poster presented at the Joint Statitical Meetings of the American Statistical Association, Anaheim, CA.

1997. Moisen, G. G., T. C. Edwards, Jr., and R. D. Cutler. Linking regional forest inventory data with satellite-based information: the broad-scale prediction of forest structure. Seminar sponsored by Utah Chapter of the Society of American Foresters, Ogden, UT.

1997. Moisen, G. G., D. Van Hooser, T. C. Edwards, Jr., and R. D. Cutler. Adding an 'in-place' dimension to grid inventories: linking regional forest inventory data with satellite-based information for prediction of forest structure. Seminar presented before US Forest Service, Forest Inventory, Economics, and Recreation Research Staff, Washington, D.C.

1997. Edwards, T. C., Jr., G. G. Moisen, D. Van Hooser, and R. D. Cutler. Linking digital cover maps with regional forest inventory data: the broad-scale prediction of forest structure. Seminar presented before the USGS Biological Resources Division Staff, Reston, Virginia, 4/3/97.

1997. Moisen, G. G., T. C. Edwards, Jr., and R. D. Cutler. Linking digital cover maps with regional forest inventory data: the broad-scale prediction of forest structure. Seminar sponsored by the Forest and Rangeland Ecosystem Science Center, Corvallis, OR.

1996. Moisen, G. G. Improved statistical models for estimating, exploring and mapping forest attributes. Presentation at the Second Annual US Forest Service Timber Staff Officers Workshop, Ogden, UT.

1996. Moisen, G. G., D. R. Cutler, and T. C. Edwards, Jr. Generalized linear mixed models for analyzing error in a satellite-based vegetation map of Utah. Paper presented at the Second International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, Ft. Collins, CO.

1993. Edwards, T. C., Jr., G. G. Moisen, and D. R. Cutler. Spatial sampling to assess classification accuracy of remotely sensed data. Paper presented at The First Environmental Information Management and Analysis Symposium, National Science Foundation, Albuquerque, N.M.

1992. Moisen, G. G. Efficiency of sample designs to assess map accuracy. Paper presented at the Second Annual National Gap Analysis Meeting, Logan, UT.

OFFERED PRESENTATIONS

2000. Moisen, G. G, T. S. Frescino, and R. P. Tymcio. An ecoregion mapping system in the Interior West. Presentation given at the 2nd annual FIA Symposium, Salt Lake City, UT.

2000. Moisen, G. G. Comparing linear, nonlinear, and nonparametric models for stratification and prediction in forest inventories of the Interior Western USA. Presentation given at the 2000 Joint Statistical Meetings of the American Statistical Association, Indianapolis, IN.

2000 Frescino, T.S., G.G. Moisen, and T.C. Edwards. Development and use of Forest Inventory and Analysis (FIA) map products. Eighth Biennial Remote Sensing Applications Conference, Albuquerque, NM, April 2000.

2000 Reutebuch, S., D. Patterson, B. Bollenbacher, and G. G. Moisen. Large-scale, GPS/IMU-controlled aerial photography for forest inventory and monitoring. Paper to be presented at the 8th Biennial Remote Sensing Applications Conference, Albuquerque, N.M.

1999. T. C. Edwards, Jr., G. G. Moisen, and T. S. Frescino. Modelling landscape-level forest attributes using modern regression tools: methods and applications. Poster presented at the 5th World Congress of the International Association of Landscape Ecology, Snowmass, Colorado.

1999. T. S. Frescino, T. C. Edwards, Jr., and G. G. Moisen. Development and validation of forest habitat models in the Uinta Mountains, Utah. Poster presented at the 5th World Congress of the International Association of Landscape Ecology, Snowmass, Colorado.

1999. Moisen, G. G., R. P. Tymcio, and T. C. Edwards, Jr. Stratification and modelling experiences in the Interior West. Presentation before the Technical Review Committee on the status of the annual inventory system with focus on the South, New Orleans, LA.

1998. Frescino, T. S., G. G. Moisen, and T. C. Edwards, Jr. Development and validation of forest habitat models in the Uintah mountains, Utah. Poster presented at the Symposium on Integrated Tools for Natural Resources in the 21st Century, Boise, ID.

1998. Frescino, T. S., G. G. Moisen, and T. C. Edwards, Jr. Development and validation of forest habitat models in the Uintah mountains, Utah. Poster presented at the 83rd Annual Meeting of the Ecological Society of America, Providence, R.I.

1998. Moisen, G. G. and T. C. Edwards, Jr. Comparing modern regression techniques for broad-scale prediction of forest structure. Presentation at the Third International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, Quebec City, Canada.

1998. Edwards, T. C., Jr., G. G. Moisen, J. J. Lawler, and T. S. Frescino. Developing probability based models of regional patterns in animal diversity. Presentation at the Third International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, Quebec City, Canada.

1998. Edwards, T. C., Jr. and G. G. Moisen. The seven faces of Dr. Accuracy (with apologies to Lao!) Presentation at the Third International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, Quebec City, Canada.

1998. Moisen, G. G., T. C. Edwards, Jr., and T. S. Frescino. Merging forest inventory data with satellite-based information in Utah. Seminar sponsored by the Department of Fisheries and Wildlife, Utah State University, Logan, UT.

1998. Moisen, G. G., T. C. Edwards, Jr., and D. Van Hooser. Merging regional forest inventory data with satellite-based information through nonlinear regression methods. Poster presented at the Second International Conference on the Fusion of Earth Data, Sophia Antipolis, France.

1997. Moisen, G. G., T. C. Edwards, Jr., and D. R. Cutler. Broad-scale prediction of forest structure through generalized additive models. Poster presented at the USDA Forest Service Rocky Mountain Research Station Leadership Team Meeting, Ogden, UT.

1997. Moisen, G. G., T. C. Edwards, Jr., and D. R. Cutler. Broad-scale prediction of forest structure through generalized additive models. Poster presented at the Annual Meeting of the Ecological Society of America, Albuquerque, N.M.

1997. Moisen, G. G., T. C. Edwards, Jr., and R. D. Cutler. Why I want a GAMM for Christmas. Seminar sponsored by the Department of Mathematics and Statistics, Utah State University, Logan, UT.

1997. Moisen, G. G., T. C. Edwards, Jr., and R. D. Cutler. Estimation, exploration and mapping of forest attributes through extensions to generalized additive models. Poster displayed at the Association for Women in Science, Northern Utah Chapter annual poster display, 4/4/97.

1997. Moisen, G. G., T. C. Edwards, Jr., and R. D. Cutler. Sampling and analysis strategies for assessing map accuracy in ecoregion-scale cover-maps. Poster displayed at the Association for Women in Science, Northern Utah Chapter annual poster display, Logan,UT.

1996. Moisen, G. G., T. C. Edwards, Jr., and D. R. Cutler. Estimation, exploration and mapping of forest attributes through extensions to generalized additive models. Poster presented at the Meeting on Modelling Longitudinal and Spatially Correlated Data, Nantucket, MA.

1996. Moisen, G. G., T. C. Edwards, Jr., and D. R. Cutler. Sampling and analysis strategies for assessing map accuracy in ecoregion-scale cover-maps. Poster presented at the Sixth Annual National Gap Analysis Meeting, Key Largo, FL.

1996. Edwards, T. C., Jr., E. Deshler, D. Foster, and G. G. Moisen. Adequacy of wildlife habitat models for estimating spatial distributions of terrestrial vertebrates. Poster presented at the Sixth Annual National Gap Analysis Meeting, Key Largo, FL..

1996. Edwards, T. C., Jr., G. G. Moisen, and R. D. Cutler. Sampling and analysis strategies for assessing map accuracy in ecoregion-scale cover-maps. Poster presented at the Second International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, Ft. Collins, CO.

1996. Moisen, G. G., T. C. Edwards, Jr., and R. D. Cutler. Accuracy assessment for the Gap Analysis cover-map of Utah. Poster presented at the Sixth Biennial Forest Service Remote Sensing Conference, Denver, CO.

1996. Moisen, G. G., D. R. Cutler, and T. C. Edwards, Jr. Generalized linear mixed models for analyzing error in a satellite-based vegetation map of Utah. Poster presented at the Third International Conference on Integrating GIS and Environmental Modelling, Santa Fe, N.M..

1995. Edwards, T. C., Jr., E. Deshler, D. Foster, and G. G. Moisen. Adequacy of wildlife habitat models for estimating spatial distributions of terrestrial vertebrates. Paper presented at the 1995 Annual Conference of The Wildlife Society, Portland, OR.

1995. Edwards, T. C., Jr., G. G. Moisen, and R. D. Cutler. Use of cluster sampling to assess map accuracy of remotely sensed cover-maps. Paper presented at the Eighth Annual Meeting of the Ecological Society of America, Snowbird, UT.

1994. Edwards, T. C., Jr., E. Deshler, D. Foster, and G. G. Moisen. Adequacy of wildlife habitat relation models for estimating spatial distributions of terrestrial vertebrates. Poster presented at the Fourth Annual National Gap Analysis Meeting, Seattle, WA.

1994. Moisen, G. G., T. C. Edwards, Jr., and D. R. Cutler. Assessing map uncertainty in Gap Analysis cover maps: incorporation of a cost function. Poster presented at the Fourth Annual National Gap Analysis Meeting, Seattle, WA.

1992. Obrien, R. A. and G. G. Moisen. Expanded applications of large-scale integrated forest inventory data. Poster presented at the Workshop on Taking an Ecological Approach to Management Using Integrated Resource Maps and Inventories, Salt Lake City, UT.