Semiparametric Estimation for Integral Projection Models

by

Kyoung Ju Kim

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 8, 2020

Keywords: Integral Projection Models, Sensitivity Analysis, State Space Models, Single Index Models

Approved by

Asheber Abebe, Chair, Professor of Mathematics and Statistics
F. Stephen Dobson, Professor of Biological Sciences
Nedret Billor, Professor of Mathematics and Statistics
Guanqun Cao, Associate Professor of Mathematics and Statistics

Abstract

Understanding the connection between variation in climate and population dynamics of plants and animals is important for predicting the impacts of future climate change. A popular approach for studying population dynamics is integral projection models, in which covariates are easily parameterized by regression models. The main goal of this dissertation is to extend the scope of stochastic integral projection models (IPMs), which have been commonly fitted with linear or generalized linear models in the past.

The state-space models and the stochastic IPMs using linear mixed models are discussed in Chapter 2 and Chapter 3 respectively. In the state-space models, we study climate effects on the population, which is segmented in different age groups. We find that we may need to include climate effects to better understand the population dynamics. The second chapter provides an important basic analysis for the validity of the findings of the IPMs. The third chapter provides the analyses of the stochastic IPMs with linear and generalized linear models in which the LASSO method is applied for variable selection, and perturbation analyses are discussed.

In Chapter 4, by fitting more flexible IPMs, we develop a new method of finding elasticities of population growth rate to climate effects by estimating the derivatives of smooth functions of semi-parametric regression models.

Based on the models studied in this dissertation, we find that the projected population growth is consistent across all models. In addition, we find that climate variables associated with temperature have significant effects on the population growth rate.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Asheber Abebe who has guided my research with expertise and patience. Without his insight and help, this dissertation would have never taken shape.

I am also deeply grateful to Dr. F. Stephen Dobson for generously sharing his time to proofread this dissertation and provide constructive comments.

My appreciation also extends to the rest of my thesis committee: Dr. Nedret Billor and Dr. Guanqun Cao for their time and support to serve on my committee. I would like to also thank Dr. Alan Wilson for his time and contributions as the university reader.

Finally, I would like to thank my family and friends for their love.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Matrix projection models have been commonly used to study population dynamics in the past (Coulson et al. 2001; Saltz et al. 2006; Bakker et al. 2009; and Hunter et al. 2010). Matrix models are easy to compute and have well established methods for estimating sensitivities and elasticities (Caswell 2001). In matrix models, populations are often divided into discrete classes or stages: the most general structure in populations is age.

However, matrix population models do not have a seamless way to include effects of environments, or other biological variables in the estimation of vital rates. One way to overcome the limitations that the matrix population models present is given by state-space models or dynamic linear models, which were introduced in Kalman (1960) and Kalman and Bucy (1961). In addition to accounting for temporal correlations in variables by modeling them as a Markov process, state-space models provide a means to efficiently include covariates that can affect vital rates as well as stochasticity to account for possible unmeasured observation errors.

State-space models, however, do not provide a way to include individual characteristics that are important in determining vital rates. For instance, in addition to age, the body mass of a female may play an important role in her ability to reproduce and survive. For example, Rubach et al. (2016) showed that female Columbian ground squirrels have alternative reproductive options. While most females reproduce annually, some females with low body mass may skip reproduction and invest more strongly in their own weight and body condition in any given year. Thus females of the same age may or may not give birth in a certain year depending on their body condition. A modeling approach that allows one to include such individual characteristics in the estimation of vital rates as well as quantify their effects is the integral projection model (IPM). An IPM is a practical tool for organizing demographic information about populations according to continuous trait variations (Ellner and Rees 2006). This approach avoids the issues caused by classifying the population into discrete stages, such as age groups. Another advantage of IPMs is that parameters can be easily estimated by fitting regression models. For these reasons, IPMs are becoming popular (Ellner et al. 2016 and references therein), and applications extending the basic IPM to deal with stochasticity and additional biological properties have been developed since the introduction of IPMs (Ellner and Rees 2007, Rees and Ellner 2009, and Ellner et al. 2016).

1.2 Contribution

Motivated by a population ecology problem that involves Columbian ground squirrels, our study presents a stochastic integral projection model as a viable method to capture the effect of individual biological traits as well as climate effects on population vital rates. We integrate model selection via the LASSO into the fitting IPMs to provide parsimonious IPM models. This is particularly useful for selecting combinations of climate and environmental variables that influence vital rates. Variable selection was commonly done using stepwise regression with Akaike information criterion (AIC) and Bayesian information criterion (BIC) in the past. However, in addition to being computationally expensive, these methods have a limitation in that they may end up selecting models that are too simplistic. To avoid the limitation, we apply the LASSO shrinkage method to include relevant variables in the regression models within the IPM framework. To our knowledge, this is the first work that combines constrained estimation for model selection with integral projection models.

We first conducted a preliminary analysis of the data using state-space models in order to contrast them with the IPM approach. We expected the approaches to give comparable estimates of the projected population growth, because both are based on the same changes in population size. However, these alternative methods may differ in their estimation of model components. Additionally, our primary goal was to understand the effects of climate variables in changes of population growth.

In Chapter 4, we extended the IPM approach using semi-parametric models: partial linear models and single index models. These approaches combine the flexibility of non-parametric, additive models with interpretability of parametric regression models. The approach allows us to measure the contribution of individual variables to the vital rates that guide population dynamics. To understand the effect of environmental conditions on population growth rate, we calculated the elasticities of climate variables within the proposed semi-parametric IPM framework, which involved evaluations of derivatives of the unknown smooth functions. The computing method of elasticities using the derivatives of smooth functions is new in IPMs, and the new approach is particularly important as it expands the scope of IPM applications.

Chapter 2

State Space Models

2.1 Introduction

An understanding of population dynamics is fundamental to the study of ecology (Elton 1927; Krebs 1985). This is not only a theoretical focus of ecology, but also a practical matter given global change (IPCC 2019). Such research requires a means for describing the life cycles of individuals. In any population, the rates at which the individuals are born, grow, reproduce, and die are termed vital rates. The dynamics of the population are determined by these vital rates. Matrix population models provide a way to link the individual to its population, based on fairly simple descriptions of these vital rates (Caswell 2001). Matrix models allow us to make crude groupings of individuals (usually by age and gender, but also by such groups as body mass classes), but they do not allow us to differentiate between individuals in the same grouping. Vital rates also depend on the environment, in addition to the characteristics of the individuals. Thus, models that make explicit connections between vital rates and environmental factors are essential for studying population dynamics.

In this chapter, we consider state-space models (SSMs), which provide the flexibility in modeling various types of data (continuous, count, binary, and categorical data) with linear or nonlinear processes (Auger-Methe, et al. 2020). Also, the SSMs allow us to model process variation and observation errors separately. Including these two separate sources of stochasticity in the model allows us to differentiate between biological variation and observation errors. This allows fuller characterization of uncertainty in the model, than if only one source of stochasticity were included. Since their first introduction by Kalman (1960) and Kalman and Bucy (1961), SSMs have been extensively used in signal processing and more recently for modeling population dynamics (Newman et al., 2014), metapopulation dynamics (Ward et al., 2010), and fisheries stock assessment (Aeberhard et al., 2018).

We are interested in estimating the population growth rate by projecting future population sizes and also the differential effect of weather on the various age groups of the Columbian ground squirrels (CGS). We considered "normal" SSMs with process variance and observation errors modeled with Gaussian distributions. SSMs make two main assumptions. First, SSMs assume that the underlying state time series evolves as a Markov process. This means that the state at time t, $\mathbf{N}_t$, depends only on the state at the previous time step, $\mathbf{N}_{t-1}$. Second, SSMs assume that given their dependence on the state, each observation $\mathbf{Y}_t$ is independent of all other observations $\mathbf{Y}_s$, $s \neq t$ (Figure 2.1).

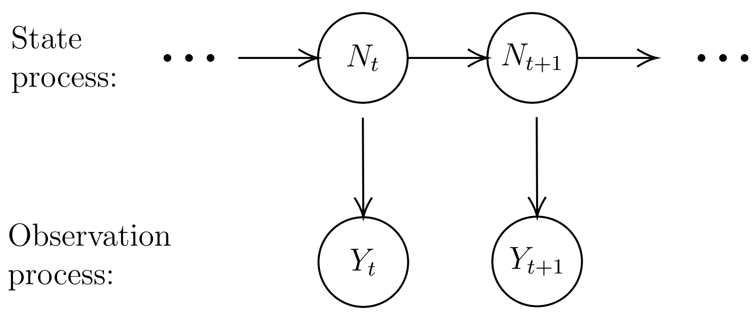


Figure 2.1: Diagram of a state space model

2.2 Data

2.2.1 Population Data

Our population data set that motivated our development were individual-based, and consisted of information for uniquely marked individual in a population of Columbian ground squirrels within the Sheep River Provincial Park in the Rocky Mountains of Alberta, Canada (50.39° 7′ N, 114.37° 27′ W; 1550 m elevation, see Figure 2.2). Each spring, all ground squirrels were permanently marked with numbered metal ear tags and dyed with individual symbols for later visual recognition. In summer, mothers and their known litters were trapped and individually marked during intensive monitoring. The ground squirrels were thus monitored annually during their April to July/August annual activity period in each year between 1992 and 2018 (with 2018 used for survival and growth of ground squirrels monitored in 2017). For more information on how data were collected or further information on the study area, see Dobson et al. (2016). For each individual adult female, the variables we are interested in include:

- **year**: indexed by year t.
- **z**: body mass at emergence from hibernation, in log scale of the individual at year t.
- **z′**: body mass at emergence from hibernation, in log scale of the individual at year $t + 1$.
- **juv.mass**: mean mass at emergence from hibernation, in log scale of one-year-old offspring (young in the past year).
- **ls.wean**: the number of young in a female's litter, when young first emerge from nest burrows near the time of at weaning.
- **surv.juv**: a number of young that survived to the next year from the mother's litter in the previous year.
- **had.lit**: whether or not (1/0) a female reproduced.
- **fy.surv**: whether or not (1/0) a female survived to the next spring.
- **wean.f**: the number of female pups at weaning per mother.

Scatter plots of the population data set are shown in Figure 2.3. These show the expected patterns that higher mass appears to be related to higher rates of survival,

Figure 2.2: Columbian ground squirrel study site

higher reproduction probability, more offspring per reproduction, and higher mass of recruits. Connections between mass and reproduction for Columbian ground squirrels were discussed in the paper Rubach et al. (2016).

2.2.2 Climate Data

In order to examine the effect of environmental variables on CGS population dynamics, we used climate data from Dobson, et al. (2016). Dobson et al. (2016) used the climate data for examining the relations between inter-annual variation in climate variables (rainfall, temperature, snowfall, and snowpack) and annual fitness of Columbian ground squirrels. In particular, they used a "sliding window" search procedure (Lane et al. 2012; Mihoub et al. 2012) to identify relevant periods of climate windows. Original climate data were obtained for Okotoks, Alberta, from the Environment Canada weather archive

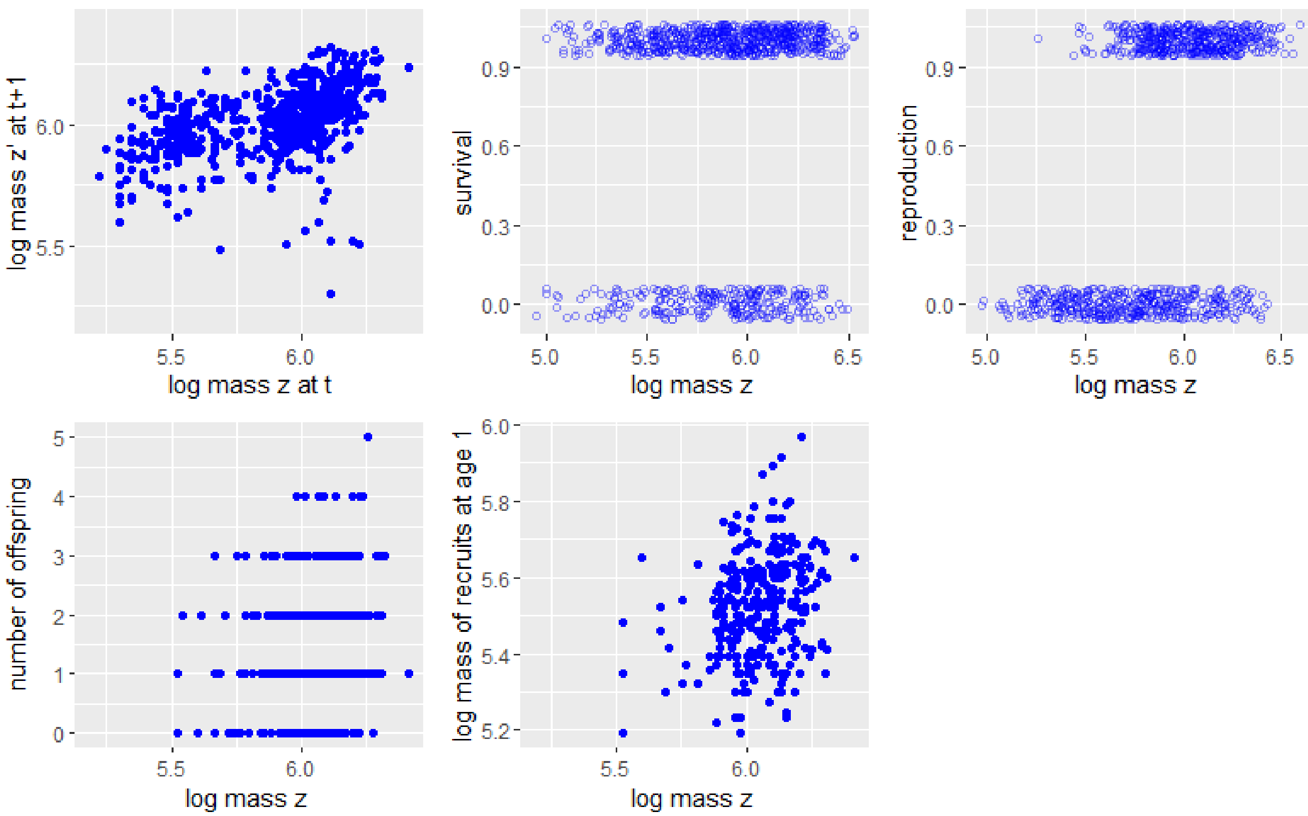


Figure 2.3: Population data for vital rate functions

(http://climate.weather.gc.ca/) (Lane et al. 2012; Dobson et al. 2016). The full list of climate variables we examined include mean annual temperature (TempAnn), mean temperature during the annual emergence from hibernation, from April 12 to May 3 (TempAM1), mean temperature from April 26 to May 7 (TempAM2), mean temperature after most litters were weaned, from June 28 to July 18 (TempJJ), mean annual rainfall (RainAnn), mean rainfall after most litters were weaned, from June 28 to July 11 (RainJJ), mean annual snowfall (SnowAnn), mean annual precipitation (Precip), mean annual snow pack (PackAnn), mean annual daily snow pack from December 5 to December 15 (PackD), mean daily snow as ground squirrels were emerging from hibernation and starting the active season, from April 13 to April 28 (Snow before), mean daily snow after the end of the annual active season when all ground squirrels were hibernating, from November 26 to December 10 (Snow after), mean annual daily snowfall during the following year (Snow following year) (Figure 2.4).

2.3 Fitting State Space Models

Let $\mathbf{N}_t$ be the estimates of the true (hidden) population size at time t , and $\mathbf{Y}_t$ be the observed population size at time t. To construct the SSM, we divided the population of Columbian ground squirrels into $p + 1 = 10$ age classes: $\mathbf{N}_t = (x_{0t}, \ldots, x_{pt})^T$, where x_{it} represents the number of squirrels in the population within the i-th age class in year t. For example, x_{0t} is the number in the population that are age 0, and x_{pt} is the number of population of age $p = 9$ and older at year t. The last class is an absorbing class, since few ground squirrels survive beyond 9 years of age (maximum age = 14 years).

Then our state space model has the following two time series:

$$\mathbf{N}_t = \Phi \mathbf{N}_{t-1} + \Gamma \mathbf{u}_t + \mathbf{w}_t \qquad \text{(state equation)}$$

$$\mathbf{Y}_t = \mathbf{N}_t + \mathbf{v}_t \qquad \text{(observation equation)}$$

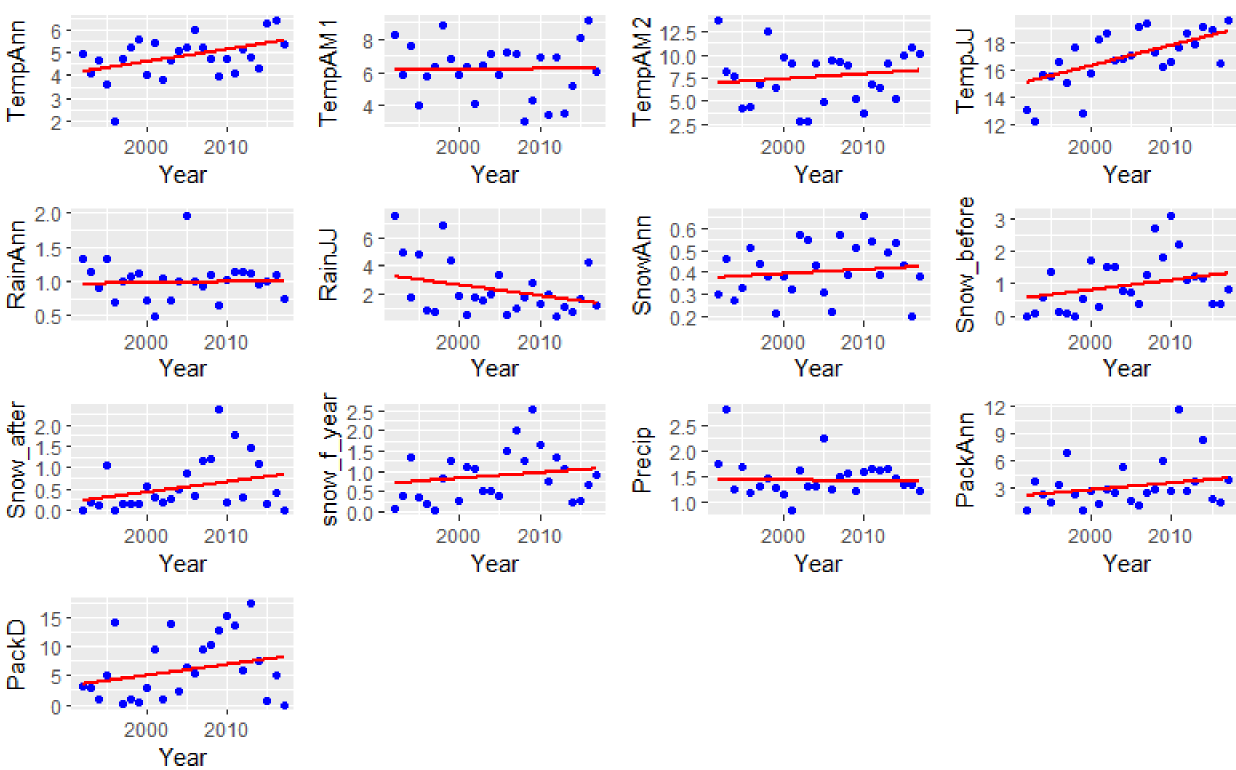


Figure 2.4: Climate data: A linear regression line has been added to each scatter plot.

Here Φ is a $(p+1) \times (p+1)$ unknown transition matrix composed of birth rates and survival rates (equation 2.1), and Γ is an unknown $(p+1) \times q$ coefficient matrix for the $q \times 1$ weather variable vector $\mathbf{u}_t$. Here $\mathbf{w}_t$ are $(p+1)$-dimensional independent and identically distributed, zero-mean normal vectors with covariance matrix Q (we write this as $\mathbf{w}_t \sim_{iid} N_{(p+1)}(\mathbf{0}, Q)$). $\mathbf{Y}_t$ is a $(p+1)$-dimensional observed data vector. The additive observation noise is $\mathbf{v}_t \sim_{iid} N_{(p+1)}(\mathbf{0}, R)$, where R is the covariance matrix of $\mathbf{v}_t$. While in general it is possible to include input variables in the observation equation, we will not discuss that case since that does not apply to the study in question. We also assume that the process starts with a normal vector $\mathbf{N}_0$, such that $\mathbf{N}_0 \sim N_{(p+1)}(\boldsymbol{\mu}_0, \Sigma_0)$.

Since we have $p+1$ age classes in our model, the transition matrix Φ has the following form:

$$\Phi = \begin{pmatrix} 0 & b_1 & b_2 & \cdots & b_{p-2} & b_{p-1} & b_p \\ \phi_{10} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \phi_{21} & 0 & \cdots & 0 & 0 & 0 \\ & & \ddots & \vdots & & & \\ 0 & 0 & 0 & \cdots & \phi_{p-1,p-2} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \phi_{p,p-1} & \phi_{p,p} \end{pmatrix} \quad (2.1)$$

where b_i is a female birth function for the i-th age class and ϕ_{ij} is a female survival function from j-th age class to i-th age class. The diagonals of Φ for the first p classes are 0 since we have single age groups and the post-breeding census is conducted yearly. The $(p+1)$-th diagonal ($\phi_{p,p}$) is non-zero since we have multiple ages (≥ 9 for CGS data) and within-class age transitions can occur.

A primary goal of the state-space model is to estimate and forecast the underlying unobserved, hidden population size $\mathbf{N}_t$, given the observed data $\mathbf{Y}_{1:s} = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots \mathbf{Y}_s\}$. When $s < t$, the problem is called forecasting or prediction. When $s = t$, the problem is called filtering. When $s > t$, the problem is called smoothing (Shumway and Stoffer 2017). If there are no weather variables, then the population growth rate can be estimated directly by finding the principal eigenvalue of $\hat{\Phi}$. If weather variables $\mathbf{u}_t$ are included in the model, then we can approximate the growth rate by studying the rate of change in the estimated signal $\mathbf{N}_t$.

The Kalman filter approach can be used to obtain the filtering and forecasting equations. We will use the notation and description given in Shumway and Stoffer (2017). Let

$$\mathbf{N}_t^s = E(\mathbf{N}_t | \mathbf{Y}_{1:s})$$
$$P_t^s = E\{(\mathbf{N}_t - \mathbf{N}_t^s)(\mathbf{N}_t - \mathbf{N}_t^s)'\}$$

be the conditional mean and covariance of the state process.

For the state space model specified in (state equation) and (observation equation), with initial conditions $\mathbf{N}_0^0 = \mu_0$ and $P_0^0 = \Sigma_0$, for $t = 1, 2, \ldots, n$, we can show that the mean and covariance can be iteratively updated as

$$\mathbf{N}_t^{t-1} = \Phi \mathbf{N}_{t-1}^{t-1} + \Gamma \mathbf{u}_t \tag{2.2}$$
$$P_t^{t-1} = \Phi P_{t-1}^{t-1} \Phi' + Q \tag{2.3}$$

with

$$\mathbf{N}_t^t = \mathbf{N}_t^{t-1} + K_t(\mathbf{Y}_t - \mathbf{N}_t^{t-1})$$
$$P_t^t = [I - K_t] P_t^{t-1}$$

where

$$K_t = P_t^{t-1}[P_t^{t-1} + R]^{-1}$$

is the so-called Kalman gain.

Prediction for $t > n$ is obtained using (2.2) and (2.3) with conditions $\mathbf{N}_n^n$ and P_n^n. Moreover, we have the following prediction errors (innovations)

$$\epsilon_t = \mathbf{Y}_t - E(\mathbf{Y}_t|\mathbf{Y}_{1:t-1}) = \mathbf{Y}_t - \mathbf{N}_t^{t-1}$$

and the corresponding variance-covariance matrices

$$\Sigma_t = \text{var}(\epsilon_t) = \text{var}[(\mathbf{N}_t - \mathbf{N}_t^{t-1}) + \mathbf{v}_t] = P_t^{t-1} + R \quad \text{for} \quad t = 1, \ldots, n \; .$$

This assumes that the model parameters are known and if so, then we can use the `ksmooth1` function of the package `astsa` to estimate $\mathbf{N}_t^s$ using the Kalman filter. In real applications, however, the model parameters are unknown and need to be estimated. When the parameters are unknown, then we can use the maximum likelihood estimation to estimate the parameters which are then plugged into the Kalman filter routine. For this optimization problem we use the method of Byrd et al. (1995) which uses box constraints where each variable can be given a lower and/or upper bound along with the quasi-Newton method. The quasi-Newton method is an optimization method that uses the first derivatives (gradients) and approximated second derivatives (Hessian matrix) of the function being optimized. This is implemented in the method "L-BFGS-B" of the R function `optim`. Initial values are obtained using the EM algorithm, which performs maximum likelihood estimation in the presence of latent variables (Shumway and Stoffer 2017). The details are given below.

2.4 Maximum Likelihood Estimation

Let Θ be the vector of all unknown parameters. That is, Θ contains the initial mean μ_0 and covariance matrix Σ_0, the transition matrix Φ, and the state and observation covariance matrices Q and R, and the input coefficient matrix Γ.

Under the assumptions $\mathbf{N}_0 \sim N_{p+1}(\boldsymbol{\mu}_0, \Sigma_0)$, $\mathbf{w}_t \sim_{iid} N_{p+1}(\mathbf{0}, Q))$, and $\mathbf{v}_t \sim_{iid} N_{p+1}(\mathbf{0}, R)$, we used the method of maximum likelihood to estimate Θ. Under the Gaussian distribution assumption, the negative log likelihood function $L_{\mathbf{Y}}(\Theta)$ to be minimized is given by

$$-\ln L_{\mathbf{Y}}(\Theta) = \frac{1}{2}\sum_{t=1}^{n} \ln|\Sigma_t(\Theta)| + \frac{1}{2}\sum_{t=1}^{n} \epsilon_t(\Theta)'\Sigma_t(\Theta)^{-1}\epsilon_t(\Theta)$$

where $\epsilon_t = Y_t - N_t^{t-1}$ and $\Sigma_t = P_t^{t-1} + R$.

To accomplish maximum likelihood estimation, we can use a Newton-Raphson algorithm:

Step 1 : Select initial values for the parameter, say, $\Theta^{(0)}$. We will use the EM algorithm to estimate $\Theta^{(0)}$.

Step 2 : Run the Kalman filter using the initial parameter values, $\Theta^{(0)}$, to obtain a set of innovations and error covariances, $\{\epsilon_t^{(0)} : t = 1, \cdots, n\}$ and $\{\Sigma_t^{(0)} : t = 1, \cdots, n\}$

Step 3 : Run one iteration of a Newton-Raphson procedure with $-\ln L_{\mathbf{Y}}(\Theta)$ as the criterion function (i.e., the function to be minimized), to obtain a new set of estimates $\Theta^{(1)}$

Step 4 : At iteration j, $(j = 1, 2, \cdots)$, repeat Step 2 using $\Theta^{(j)}$ in place of $\Theta^{(j-1)}$ to obtain a new set of innovation values $\{\epsilon_t^{(j)} : t = 1, \cdots, n\}$ and $\{\Sigma_t^{(j)} : t = 1, \cdots, n\}$

Step 5 : Repeat Step 3 to obtain a new estimate $\Theta^{(j+1)}$. Stop when the values of $\Theta^{(j+1)}$ differ from $\Theta^{(j)}$, or when $L_{\mathbf{Y}}(\Theta^{(j+1)})$ differ from $L_{\mathbf{Y}}(\Theta^{(j)})$, by some predetermined, small amount.

2.5 Results

We first considered the state space model without climate effects on the population. In this case, the input vector $\mathbf{u}_t$ in the state equation is zero. So, the state space model becomes

$$\mathbf{N}_t = \Phi \mathbf{N}_{t-1} + \mathbf{w}_t \tag{2.4}$$

$$\mathbf{Y}_t = \mathbf{N}_t + \mathbf{v}_t \tag{2.5}$$

Then, using the estimated Φ, we obtained $\mathbf{N}_t = (x_{1,t}, \cdots, x_{10,t})$ by the equation

$$\mathbf{N}_t = \Phi \mathbf{N}_{t-1} \tag{2.6}$$

We then used the initial population number in 1992 and projected the population forward for each age class using the estimated Φ. The overall estimated annual population $n_t = \sum_{i=1}^{10} x_{i,t}$ and growth rates are shown as solid blue curves in the Figure 2.5. Depicting the population growth as a monotonic increase, the blue curves clearly fail to detect annual population fluctuations. In particular, a significant decrease of the population between the years 2001 and 2003 is not reflected in the blue curves. So this observation suggests that we may need to include other variables such as climates, which may influence the survival and reproduction probabilities of the population.

Thus, we included climate variables in the state space model. To simplify our estimation process and reduce the number of parameters we need to estimate, we considered two cases with simple climate effect scenarios:

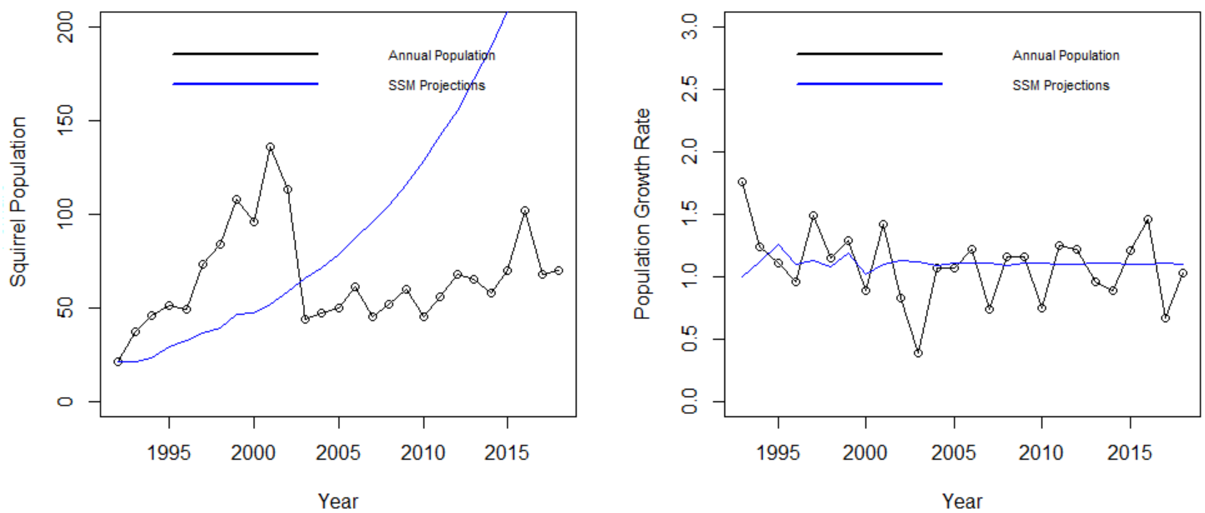


Figure 2.5: **Left**: The population trajectory with no climate effects. **Right**: Population growth rate with no climate effects.

Case 1: Climate effects are the same on all ages. If the non-zero input vector $\mathbf{u}_t$ represents all 13 weather variables at year t, then the input coefficient matrix Γ has the form

$$\Gamma = \begin{pmatrix} \mathbf{c} \\ \mathbf{c} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{c} \end{pmatrix}_{10\times 13}$$

where $\mathbf{c} = (c_1, c_2, \cdots, c_{13})$.

Case 2: Climate effects on age 0 is different from the rest of the age groups, but it is uniform for all over age 0. In this case, the input coefficient matrix Γ has the form

$$\Gamma = \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{d} \end{pmatrix}_{10\times 13}$$

where $\mathbf{c} = (c_1, c_2, \cdots, c_{13})$ and $\mathbf{d} = (d_1, d_2, \cdots, d_{13})$.

Now our state space model becomes as follows.

$$\mathbf{N}_t = \Phi\mathbf{N}_{t-1} + \Gamma\mathbf{u}_t + \mathbf{w}_t$$
$$\mathbf{Y}_t = \mathbf{N}_t + \mathbf{v}_t$$

Using the initial 1992 population, the measured climate variables, and the estimated Φ and Γ in each case, we projected the population forward for each age group. The aggregate total annual population trajectories and growth rates are shown in Figure 2.6. The solid blue curves were drawn with no climate vectors (equation 2.6). On the other hand, the red curves shown were obtained by using the equation that incorporated annual climate vectors

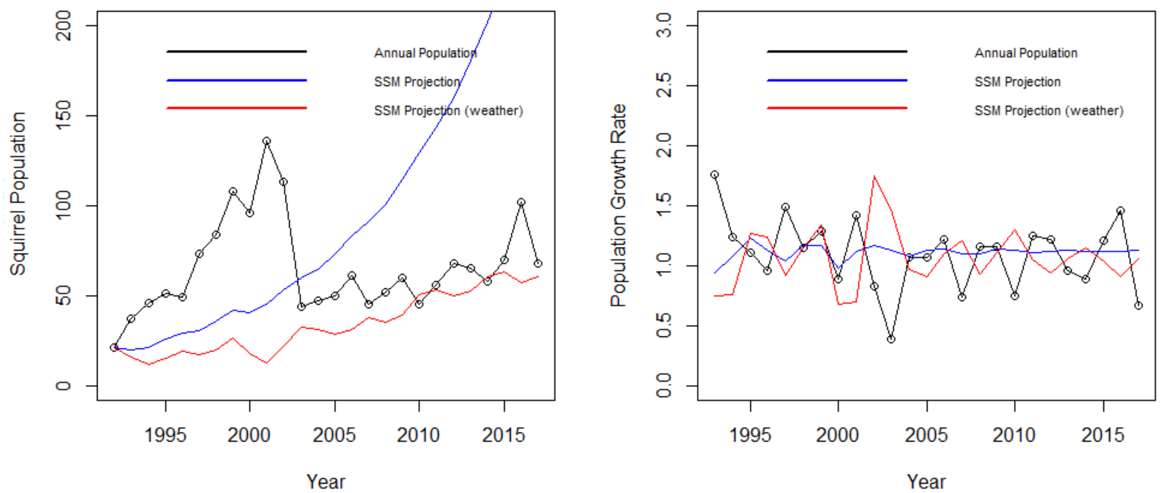
$$\mathbf{N}_t = \Phi\mathbf{N}_{t-1} + \Gamma\mathbf{u}_t \tag{2.7}$$

We see that the blue curves miss the fluctuations of the actual population given by the black line. Also the blue curves on the left-hand panels project that the population is increasing at a much faster rate than the actual population. The red curves indicate that environmental conditions have a large impact on population dynamics. We see that the model with the assumption of the uniform climate effects captures the actual population trajectory best. This could be due to the numerical efficiency of the estimation of a uniform effect. The population growth rate λ from the model was calculated at a mean of 1.13, indicating that the population was projected to increase at 13% per year.

2.6 Discussion

All in all, our results serve as a proof of concept that the stochastic components of CGS population dynamics are better captured when including climate effects. The state-space model includes both process stochasticity and observation errors. If, instead of using only the initial 1992 population, we only made one year ahead forecasts, then we can observe that the state space model gives very accurate forecasts. Figure 2.7 shows one-step ahead annual population predictions using the uniform weather effect model. Few of the observed annual populations are outside error bounds of the predicted populations. This result motivates further research to validate the accuracy of the state estimates, because the state estimates by incorporating process and observation errors can be a closer approximation of the true population dynamics than the observed data. Our study shows that including climate variables as covariates can lead to a better understanding of population dynamics. However, we still need to identify which climate variables are most important to population growth. Also, the analysis was done with simplified age-class segmentation, which may not be the most appropriate approach for describing the population. For example, rather than ages, body mass might better explain the probabilities of survival and reproduction. In fact, as noted by Ruback et al. (2016), for Columbian ground squirrel, it is the combination of age and body condition that affects the probability of successful reproduction. The effect of mass is also apparent in Figure 2.3. Two squirrels in the same age class may or may not reproduce in a

Uniform Climate Effects on Ages



Varying Climate Effects on Ages

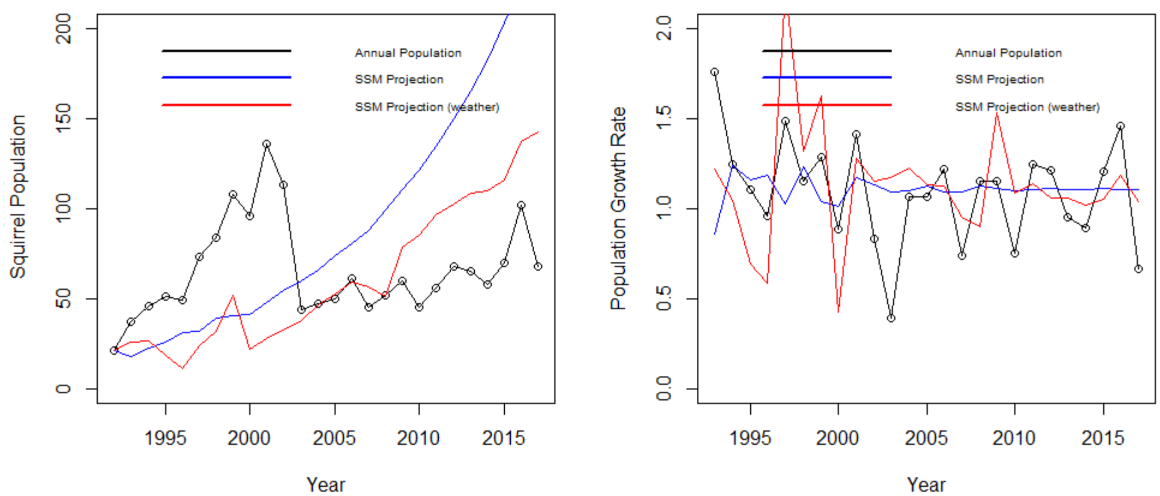


Figure 2.6: **Top Left**: Uniform climate effects on all age groups. **Top Right**: Population growth rate with uniform climate effects. **Bottom Left**: Varying climate effects: Climate effects are assumed to be different on age 0 and the same for the rest of age classes. **Bottom Right**: Population growth rate with varying climate effects.

given year, depending on their body condition. Thus, further research is needed to identify approaches that include such individual-level effects on population dynamics.

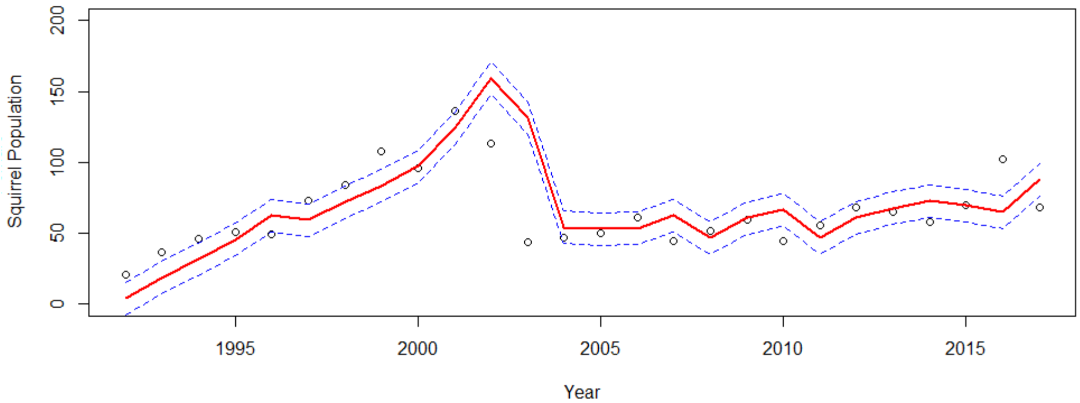


Figure 2.7: The observed annual populations y_t are shown as open dots. Based on the uniform climate effects model, the predicted annual populations n_t are shown as a red line with $\pm 2\sqrt{p_t^{t-1}}$ error bounds as dashed lines in the 95% confidence level.

Chapter 3

Stochastic Integral Projection Models

3.1 Introduction

Climate change is an important influence on the population dynamics of plant or animal species. Understanding and predicting the ecological effects of climate change on plant and animal species has thus become a central objective in ecology (e.g. Simmonds and Coulson 2015; Dalgleish et al. 2011). Study of the effects of environmental factors on populations is especially important for ecosystem conservation and management strategies due to potential chain effects among species within the ecosystem (Fortin et al. 2005). Many studies have identified the statistical association between climate or a specific climatic factor and demographic variables that contribute to population growth (Post et al. 1997; Post and Forchhammer 2002; Ellis and Post 2004; Parmesan 2006; and Ozgul et al. 2010).

In particular, Parmesan (2006) found a general pattern of earlier timing of reproduction in response to climate warming. Dalgleish et al. (2011) modeled effects of precipitation and temperature on all vital rates of three sage-brush steppe plants. Simmonds and Coulson (2015) analyzed how changes in a large-scale climate index, the North Atlantic Oscillation (NAO), might influence population dynamics and phenotypic characters in a population of Soay sheep on St. Kilda in the U.K.

Our objective was to examine how the vital rates of individuals responded to climate and how these vital rates translate into population dynamics. Vital rates are the patterns of survival, growth, reproduction and recruitment that occur in a dynamic biotic population. We used a stochastic IPM to link yearly variability in climate to interannual fluctuations of vital rates, and finally to discern the influences of changes in vital rates on changes in population size. We estimated the importance of vital rates based on a continuous state variable, log body mass. We examined the influences of climatic variables on vital rates with mixed-effects regression models, and then computed their relative influences on population growth. We applied our examination to a long term (27-year) dataset on female Columbian ground squirrels, *Urocitellus columbianus*.

The ground squirrels were an excellent model organism for us to evaluate climatic influences on populations. They are hibernating sciurid rodents that live in mountain environments, and have a very short active season each year. The lifecycle is very well known (e.g., Dobson and Kjelgaard 1985a; Dobson and Murie 1987; Dobson 1988; Dobson et al. 1999) and changes in population size have been studied using matrix and other traditional methods (e.g. Dobson and Kjelgaard 1985b; Dobson 1995; Dobson and Oli 2001). A single reproductive period occurs during their short active season, and adult females support their small litter of young from a combination of stored energy and daily foraging (Dobson et al. 1992; Broussard et al. 2003, 2005; Rubach et al. 2016). Spring and summer climatic conditions are known to influence reproduction and survival of the ground squirrels (Lane et al. 2012; Dobson et al. 2016). The influence of the changing climate in their Rocky Mountain environments on the dynamics of their populations, however, is not well examined. Thus, the species provides a good model for such an evaluation using an Integral Projection Model (IPM) approach.

3.2 Stochastic integral projection models

An IPM begins with a kernel expression, $K(z', z; \mathbf{w}_t)$, which in our case describes all possible transitions from body mass z at time t to body mass z' at time $t+1$, consists of two components, a survival-growth component and a fecundity component. The kernel is given by

$$K(z', z; \mathbf{w}_t) = \underbrace{S(z; \mathbf{w}_t)G(z', z; \mathbf{w}_t)}_{\text{Survival and Growth component } P} + \underbrace{p_b(z; \mathbf{w}_t)b(z; \mathbf{w}_t)p_r C_1(z', z; \mathbf{w}_t)}_{\text{Fecundity component } F}$$

where

- $S(z; \mathbf{w}_t)$: Survival probability function
- $G(z', z; \mathbf{w}_t)$: Growth distribution function
- $p_b(z; \mathbf{w}_t)$: Probability of reproduction function
- $b(z; \mathbf{w}_t)$: The expected litter size per individual for those that reproduce
- p_r: Probability of successful recruitment for a juvenile (from the year of weaning to the next year)
- $C_1(z', z; \mathbf{w}_t)$: Mass distribution of new recruits at age 1
- $\mathbf{w}_t$: A vector of climate variables

Then, the stochastic integral projection model for the number of individuals of size z' at time $t+1$ is given by

$$n(z', t+1) = \int_L^U K(z', z; \mathbf{w}_t) n(z, t)\, dz$$

The integral is evaluated over all possible masses, $[L, U]$ where L is $0.9 \times$ the minimum mass, U is $1.1 \times$ the maximum mass.

3.3 Regressions of vital rate functions with climate

In regressions, identifying relevant predictors for response variables is important to enhance the performance of the model. In our case, one possible way to select variables is to examine the correlations between vital rate parameter and the full list of climate variables. Then we can select climate variables that are correlated with a vital rate parameter with P-value below the chosen significance level. However, this method can be considered arbitrary because selected variables can be very different depending on the chosen level of significance. Another variable selection method is to apply subset selection technique. However, the method suffers from computational complexity when the number of predictors is large, and the high variability with small changes in the data because predictors are either retained or removed (Tibshirani, 1996). However, the LASSO continuously shrinks coefficient estimates towards zero, it provides more stable variable selection procedure. Also the LASSO sets some coefficients exactly equal to zero, thus it performs variable selection and therefore provides more interpretable model.

For this reason, we used the method of LASSO (Tibshirani, 1996) to find the relevant weather variables for each vital rate function (survival, growth, reproduction, litter size, and recruit mass). The LASSO coefficients $\hat{\beta}_\lambda^L$ are found by minimizing the ℓ_1-penalized least squares objective function

$$\sum_{i=1}^{n}\Big(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\Big)^2 + \underbrace{\lambda\sum_{j=1}^{p}|\beta_j|}_{l_1-penalty} = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j| \tag{3.1}$$

The tuning parameter λ controls the relative impact of the two terms RSS (residual sum of squares) and ℓ_1-penalty on the regression coefficient estimates. The value of the tuning parameter λ was chosen using a 10-fold cross-validation to compare the predictive performance of candidate models. Because the ℓ_1-penalty has singularity points, there was a positive probability that some of the regression coefficients would shrink to exactly zero. To select the tuning parameter λ, we used `cv.glmnet()` in R (R Core Team 2016) in a 10-fold cross-validation. Once the value of λ which gives the smallest cross-validation error was chosen, we performed variable selection using `glmnet()` to obtain a list of predictors for each vital rate function.

We modeled vital rate functions with random effects describing yearly variation in the intercepts and fixed effects of body mass, and the LASSO-selected weather variables using `lmer()` and `glmer()` in R (R Core Team 2016).

The survival function, which describes survival probability of individuals from time t to $t+1$, $s(z, \mathbf{w}_t)$, was modeled using logistic regression

$$\text{logit}[s(z, \mathbf{w}_t)] = \beta_{0,t} + \beta_1 z + \beta_{c,1}\mathbf{w}_{1,t} \cdots + \beta_{c,i}\mathbf{w}_{i,t} \ ,$$

where z is the log body mass at time t, $\beta_{0,t}$ is a random, year-specific intercept, β_1 is a common slope for mass, $\mathbf{w}_t$ is a vector of climate variables, $\beta_{c,i}$ is a common slope for the ith climate variable.

The growth function describes a distribution of changes in body mass among surviving individuals from time t to $t+1$. The expected body mass at time $t+1$ for a given body mass at time t was modeled as a linear regression

$$\mu_g(z, \mathbf{w}_t) = \beta_{0,t} + \beta_1 z + \beta_{c,1}\mathbf{w}_{1,t} \cdots + \beta_{c,i}\mathbf{w}_{i,t} + \varepsilon_t \ ,$$

where μ_g is expected body mass at time $t+1$, and $\beta_{0,t}$, β_1, $\beta_{c,i}$ and $\mathbf{w}_t$ are as described above. The error term, ε_t, was assumed to be normally distributed with a mean zero and a constant variance. Then, we modeled the growth function using the normal probability density function

$$g(z', z, \mathbf{w}_t) = \frac{1}{\sqrt{2\pi}\sigma_g} exp\Big(- \frac{(z' - \mu_g(z, \mathbf{w}_t))^2}{2\sigma_g^2}\Big)$$

with $\mu_g(z, \mathbf{w}_t)$ given above and a constant variance σ_g.

The reproduction function, $p_b(z; \mathbf{w}_t)$, which denotes the probability that an individual reproduces, was modeled in the same way as the survival function.

The recruit mass distribution, $C_1(z', z; \mathbf{w}_t)$, described a body mass distribution of one-year olds which enter the population at time $t+1$, given a mother body mass z at time t. The recruit mass distribution was modeled in the same way as the growth function.

To model the litter size function, $b(z; \mathbf{w}_t)$, which describes the expected number of female offspring, given a mom body mass z at time t, we used a Poisson distribution

$$b(z; \mathbf{w}_t) = e^{\beta_{0,t} + \beta_1 z + \beta_{c,1} \mathbf{w}_{1,t} \cdots + \beta_{c,i} \mathbf{w}_{i,t}}$$

The probability of successful recruitment, p_r, can be estimated by the ratio of the number of surviving female offspring to the number of weaned female offspring. Since we didn't have the number of surviving female offspring in the data, we estimated p_r by dividing the total number of juveniles that survived to emergence from their first hibernation by the total number of litters at weaning (assuming male and female offspring have the same probability of survival).

Finally, we constructed a set of 26 yearly kernels using the parameters obtained from the models. Next we projected the population one step forward using a kernel selected at random, of which step was iterated $50,000$ times to calculate growth rate (See equation (3.2) or (3.3)).

3.4 Stochastic growth rate

The stochastic growth rate of the population, $\log \lambda_S$, is defined by (Caswell 2001, Tuljapurkar and Haridas 2006)

$$\log \lambda_S = \lim_{t \to \infty} \frac{1}{t} \log N(t) \,, \tag{3.2}$$

where $N(t)$ is the total population size at time t and λ_S corresponds to the dominant eigenvalue of a time-invariant projection kernel. Ellner and Rees (2007) showed that the stochastic growth rate is equal to the average annual growth rate,

$$\log \lambda_S = E[\log(N(t+1)/N(t))] \tag{3.3}$$

Based on the equation (3.3), the stochastic growth rate, $\log \lambda_S$, was estimated as (Caswell 2001, Ellner and Rees 2006)

$$\widehat{\log \lambda_S} = \frac{1}{T} \sum_{t=0}^{T-1} r_t$$

where $r_t = \log(N(t+1)/N(t))$ and $T = 50,000$. We also calculated an approximate 95% confidence interval on $\log \lambda_S$ using the formula

$$\widehat{\log \lambda_S} \pm 1.96 \sqrt{\frac{V(r_t)}{T}}$$

where $\sqrt{V(r_t)/T}$ is its approximate standard error and $V(r_t)$ is the variance of $\{r_t\}_{t=0}^{T-1}$ (Caswell 2001; Ellner et al. 2016).

3.5 Perturbation analysis

Following the general procedures from Rees and Ellner (2009) (also, see Ellner, Childs, and Rees 2016), a perturbation analysis for the stochastic IPM was carried out at two different levels: the projection kernel and parameters that define the vital rate functions. From each level of perturbations, we calculated the elasticity, the fractional change in the growth rate λ_s relative to the fractional change in the quantity being perturbed (Ellner et al. 2016). By definition, the elasticity is obtained by taking the partial derivative of $\log \lambda_s$ with respect to a specific perturbation (Ellner et al. 2016).

The basic formula for the general perturbation of yearly kernels K_t to $K_t + \epsilon C_t$, where $C_t = C(z', z, \mathbf{w}_t)$ is a sequence of perturbation kernels, is given by

$$\frac{\partial \log \lambda_S}{\partial \epsilon} = \frac{1}{\lambda_S}\frac{\partial \lambda_S}{\partial \epsilon} = E\Big[\frac{\langle v_{t+1}, C_t w_t\rangle}{\langle v_{t+1}, K_t w_t\rangle}\Big] \tag{3.4}$$

where angled bracket is the inner product $\langle f, g\rangle = \int_X f(x)g(x)\,dx$ and v_t and w_t are the sequences of reproductive values and the time-varying population structure. Roughly speaking, reproductive value v_t is the relative, expected total number of descendants that will ever be produced by an individual of size z at time t. w_t represents the relative population structure at time t (Caswell 2001). w_t and v_t are calculated as follows:

$$\tilde{w}_{t+1} = K_t w_t = \int_Z K_t(z', z) w_t(z)\,dz, \qquad w_{t+1} = \tilde{w}_{t+1} / \int_Z \tilde{w}_{t+1}(z)\,dz$$

$$\tilde{v}_{t-1} = v_t K_{t-1} = \int_Z v_t(z') K_{t-1}(z', z)\,dz', \qquad v_{t-1} = \tilde{v}_{t-1} / \int_Z \tilde{v}_{t-1}(z)\,dz$$

In practice, to use the formula (3.4), we generate the sequences $\{w_t\}_{t=0}^T$ and $\{v_t\}_{t=0}^T$ with arbitrary choices of w_0 and v_T. Then, these sequences are used to evaluate the sensitivity or elasticity for any perturbation by using a time average from $t = k$ to $t = T - k$ to approximate the expectation for some large k.

To find the elasticity of λ_S to kernel value $K(z_0', z_0)$, since we have

$$K_t(z', z) \to K_t(z', z) + \epsilon \delta_{z_0', z_0}(z', z) K_t(z_0', z_0),$$

where δ_{z_0', z_0} represents the Dirac delta function, the perturbation kernel becomes $C_t = \delta_{z_0', z_0}(z', z) K_t(z_0', z_0)$. Then, by the formula (3.4), the elasticity is given by

$$e_S(z_0', z_0) = \frac{\partial \log \lambda_S}{\partial \log K_t(z_0', z_0)} = E\Big[\frac{v_{t+1}(z_0') w_t(z_0) K_t(z_0', z_0)}{\langle v_{t+1}, K_t w_t\rangle}\Big] .$$

For the elasticity to parameters, we found the perturbation kernel C_t by using the chain rule

$$C_t = \frac{\partial K_t}{\partial f} \frac{\partial f}{\partial \theta_i} \frac{\partial \theta_i}{\partial \epsilon} \ ,$$

where f is a vital rate function and θ_i is a parameter in f.

3.6 Results

3.6.1 Growth rate and parameter perturbation

The final vital rate models chosen by the Lasso procedure are shown in Tables 3.1 - 3.3.

Vital rates	Function	Parameter (SE) estimates
Survival	logit$[S(z;\boldsymbol{w}_t)]$=$\beta_{0,t}$+ $\beta_{s,1}z$ +$\beta_{s,2}$ TempAnn+$\beta_{s,3}$ TempAM1 +$\beta_{s,4}$ TempAM2 +$\beta_{s,5}$ TempJJ+$\beta_{s,6}$ RainAnn+$\beta_{s,7}$ RainJJ +$\beta_{s,8}$ Snow before +$\beta_{s,9}$ Snow after +$\beta_{s,10}$ Snow following year +$\beta_{s,11}$ Precip+$\beta_{s,12}$ PackD	$\beta_{0,t} = 0.728(2.205)$ $\beta_{s,1} = 0.256(0.287)$ $\beta_{s,2} = -0.276(0.167)$ $\beta_{s,3} = 0.14(0.10)$ $\beta_{s,4} = 0.24(0.049)$ $\beta_{s,5} = -0.176(0.063)$ $\beta_{s,6} = 0.764(0.661)$ $\beta_{s,7} = -0.172(0.084)$ $\beta_{s,8} = 0.211(0.167)$ $\beta_{s,9} = -0.058(0.239)$ $\beta_{s,10} = -0.257(0.188)$ $\beta_{s,11} = 0.109(0.545)$ $\beta_{s,12} = 0.01(0.023)$
Growth	$\mu_g(z,\boldsymbol{w}_t)$=$\beta_{0,t}$+$\beta_{g,1}z$ + $\beta_{g,2}$ TempAnn +$\beta_{g,3}$ TempAM1 +$\beta_{g,4}$ TempAM2 +$\beta_{g,5}$ TempJJ +$\beta_{g,6}$ RainAnn +$\beta_{g,7}$ RainJJ +$\beta_{g,8}$ SnowAnn +$\beta_{g,9}$ Snow before + $\beta_{g,10}$ Snow after +$\beta_{g,11}$ Snow following year +$\beta_{g,12}$ Precip +$\beta_{g,13}$ PackAnn +$\beta_{g,14}$ PackD $\sigma_g = 0.1124$	$\beta_{0,t} = 4.688(0.13)$ $\beta_{g,1} = 0.243(0.017)$ $\beta_{g,2} = -0.027(0.01)$ $\beta_{g,3} = 0.01(0.006)$ $\beta_{g,4} = 0.007(0.003)$ $\beta_{g,5} = -0.007(0.004)$ $\beta_{g,6} = 0.052(0.04)$ $\beta_{g,7} = -0.008(0.005)$ $\beta_{g,8} = 0.193(0.089)$ $\beta_{g,9} = 0.017(0.01)$ $\beta_{g,10} = 0.032(0.019)$ $\beta_{g,11} = -0.03(0.012)$ $\beta_{g,12} = -0.039(0.03)$ $\beta_{g,13} = -0.002(0.003)$ $\beta_{g,14} = -0.003(0.001)$

Table 3.1: Estimated survival and growth functions using linear mixed models. Here, z is log(body mass).

Vital rates	Function	Parameter (SE) estimates
Reproduction	logit$[p_b(z; \boldsymbol{w}_t)]$=$\beta_{0,t}$+ $\beta_{b,1}z$ +$\beta_{b,2}$ TempAnn +$\beta_{b,3}$ TempAM2 +$\beta_{b,4}$ RainAnn+$\beta_{b,5}$ RainJJ +$\beta_{b,6}$ Snow following year +$\beta_{b,7}$ Precip+$\beta_{b,8}$ PackAnn +$\beta_{b,9}$ PackD	$\beta_{0,t} = -50.727(3.524)$ $\beta_{b,1} = 7.654(0.538)$ $\beta_{b,2} = 0.486(0.151)$ $\beta_{b,3} = 0.115(0.047)$ $\beta_{b,4} = 0.818(0.676)$ $\beta_{b,5} = -0.179(0.072)$ $\beta_{b,6} = 0.302(0.182)$ $\beta_{b,7} = 1.101(0.524)$ $\beta_{b,8} = 0.108(0.046)$ $\beta_{b,9} = -0.021(0.021)$
litter size	$\log(b(z; \boldsymbol{w}_t))$=$\beta_{0,t}$+$\beta_{l,1}z$ + $\beta_{l,2}$ TempAnn +$\beta_{l,3}$ RainAnn + $\beta_{l,4}$ Snow after +$\beta_{l,5}$ Precip +$\beta_{l,6}$ PackAnn	$\beta_{0,t} = -7.318(1.903)$ $\beta_{l,1} = 1.052(0.298)$ $\beta_{l,2} = 0.183(0.061)$ $\beta_{l,3} = 0.202(0.228)$ $\beta_{l,4} = -0.0997(0.081)$ $\beta_{l,5} = 0.076(0.185)$ $\beta_{l,6} = 0.046(0.019)$

Table 3.2: Estimated reproduction and litter size functions using linear mixed models. Here, z is log(body mass).

Using the equation (3.3), we obtained the growth rate

$$\log \lambda_S = 0.0784,$$

with 95% confidence interval $(0.0769, 0.0799)$. This value indicated that the population slowly increased in the long run.

We examined how the growth rate might change under different circumstances by calculating elasticities, the fractional change of λ_S to a perturbation of a specific parameter (Tables 3.4 - 3.7).

Vital rates	Function	Parameter (SE) estimates
Recruitment	$p_r = 0.42797$	
recruit mass	$\mu_r(z, \boldsymbol{w}_t)$=$\beta_{0,t}$+$\beta_{r,1}z$ + $\beta_{r,2}$ TempAnn +$\beta_{r,3}$ TempAM1 +$\beta_{r,4}$ TempAM2 +$\beta_{r,5}$ TempJJ +$\beta_{r,6}$ RainAnn +$\beta_{r,7}$ RainJJ +$\beta_{r,8}$ SnowAnn +$\beta_{r,9}$ Snow before + $\beta_{r,10}$ Snow after +$\beta_{r,11}$ Snow following year +$\beta_{r,12}$ Precip +$\beta_{r,13}$ PackAnn +$\beta_{r,14}$ PackD $\sigma_r = 0.1104$	$\beta_{0,t} = 4.453(0.321)$ $\beta_{r,1} = 0.229(0.049)$ $\beta_{r,2} = -0.099(0.014)$ $\beta_{r,3} = 0.032(0.009)$ $\beta_{r,4} = 0.019(0.004)$ $\beta_{r,5} = -0.011(0.005)$ $\beta_{r,6} = 0.153(0.053)$ $\beta_{r,7} = -0.017(0.006)$ $\beta_{r,8} = 0.183(0.114)$ $\beta_{r,9} = 0.017(0.013)$ $\beta_{r,10} = 0.048(0.025)$ $\beta_{r,11} = -0.024(0.016)$ $\beta_{r,12} = -0.11(0.039)$ $\beta_{r,13} = -0.005(0.004)$ $\beta_{r,14} = -0.003(0.002)$

Table 3.3: The probability of successful recruitment, and estimated recruit mass function using linear mixed models. Here, z is log(body mass).

We found that an increased mass slope (rate of growth) had consistent positive effects on the growth rate λ_S in all vital rate functions (Figure 3.1). The growth rate λ_S was more sensitive to the change of the mass slope in growth than in recruit mass. It means that the female adults' growth would affect population size more than how big one-year-olds were as they were recruited to the population. Also, the growth rate λ_S was more sensitive to the change of mass slope in reproduction than in survival.

Vital Rate	Parameter	$e_{S,i}$	$e^{\mu}_{S,i}$	$e^{\sigma}_{S,i}$
Survival	Intercept	0.1377	0.1387	−0.0009
	Mass	0.2884	0.2884	⋯
	TempAnn	−0.2445	−0.2445	⋯
	TempAM1	0.1558	0.1558	⋯
	TempAM2	0.3058	0.3058	⋯
	TempJJ	−0.5728	−0.5728	⋯
	RainAnn	0.1433	0.1433	⋯
	RainJJ	−0.0696	−0.0696	⋯
	Snow before	0.0451	0.0451	⋯
	Snow after	−0.0074	−0.0074	⋯
	Snow following year	−0.0485	−0.0485	⋯
	Precip	0.0302	0.0302	⋯
	PackD	0.0136	0.0136	⋯

Table 3.4: Stochastic elasticities for parameters in survival function; the $e_{S,i}$ are the stochastic elasticities, $e^{\mu}_{S,i}$ are the stochastic elasticities to the mean, and $e^{\sigma}_{S,i}$ are the stochastic elasticities to the standard deviation

Vital Rate	Parameter	$e_{S,i}$	$e^{\mu}_{S,i}$	$e^{\sigma}_{S,i}$
Growth	Intercept	11.6399	11.6394	0.0005
	Mass	3.5729	3.5729	⋯
	TempAnn	−0.3239	−0.3239	⋯
	TempAM1	0.1566	0.1566	⋯
	TempAM2	0.1305	0.1305	⋯
	TempJJ	−0.3023	−0.3023	⋯
	RainAnn	0.1282	0.1282	⋯
	RainJJ	−0.0485	−0.0485	⋯
	SnowAnn	0.1864	0.1864	⋯
	Snow before	0.0330	0.0330	⋯
	Snow after	0.0347	0.0347	⋯
	Snow following year	−0.0741	−0.0741	⋯
	Precip	−0.1421	−0.1421	⋯
	PackAnn	−0.0115	−0.0115	⋯
	PackD	−0.0553	−0.0553	⋯
	standard deviation	0.1444	0.1444	⋯

Table 3.5: Stochastic elasticities for parameters in growth function; the $e_{S,i}$ are the stochastic elasticities, $e^{\mu}_{S,i}$ are the stochastic elasticities to the mean, and $e^{\sigma}_{S,i}$ are the stochastic elasticities to the standard deviation

Vital Rate	Parameter	$e_{S,i}$	$e^{\mu}_{S,i}$	$e^{\sigma}_{S,i}$
Reproduction	Intercept	-2.4178	-2.4178	-0.000073202
	Mass	2.1769	2.1769	$\cdots$
	TempAnn	0.1093	0.1093	$\cdots$
	TempAM2	0.0394	0.0394	$\cdots$
	RainAnn	0.0390	0.0390	$\cdots$
	RainJJ	-0.0204	-0.0204	$\cdots$
	Snow following year	0.0128	0.0128	$\cdots$
	Precip	0.0771	0.0771	$\cdots$
	PackAnn	0.0173	0.0173	$\cdots$
	PackD	-0.0064	-0.0064	$\cdots$
litter size	Intercept	-1.6677	-1.6677	$\cdots$
	Mass	1.4574	1.4574	$\cdots$
	TempAnn	0.2039	0.2039	$\cdots$
	RainAnn	0.0479	0.0479	$\cdots$
	Snow after	-0.0135	-0.0135	$\cdots$
	Precip	0.0262	0.0262	$\cdots$
	PackAnn	0.0367	0.0367	$\cdots$

Table 3.6: Stochastic elasticities for parameters in reproduction and litter size functions; the $e_{S,i}$ are the stochastic elasticities, $e^{\mu}_{S,i}$ are the stochastic elasticities to the mean, and $e^{\sigma}_{S,i}$ are the stochastic elasticities to the standard deviation

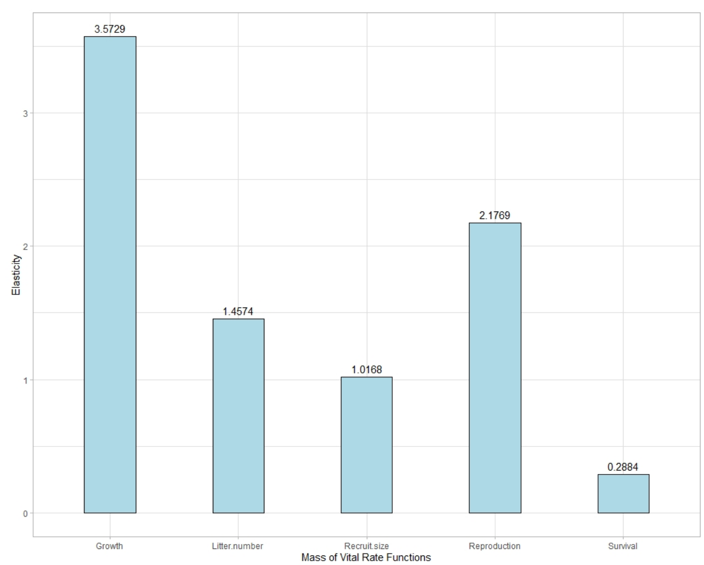


Figure 3.1: Elasticities of λ_S to the mass slope in vital rate functions

Vital Rate	Parameter	$e_{S,i}$	$e^{\mu}_{S,i}$	$e^{\sigma}_{S,i}$
Recruit Mass	Intercept	3.2478	3.2478	$\cdots$
	Mass	1.0168	1.0168	$\cdots$
	TempAnn	-0.3215	-0.3215	$\cdots$
	TempAM1	0.1389	0.1389	$\cdots$
	TempAM2	0.0755	0.0755	$\cdots$
	TempJJ	-0.1332	-0.1332	$\cdots$
	RainAnn	0.1127	0.1127	$\cdots$
	RainJJ	-0.0311	-0.0311	$\cdots$
	SnowAnn	0.0604	0.0604	$\cdots$
	Snow before	0.0130	0.0130	$\cdots$
	Snow after	0.0163	0.0163	$\cdots$
	Snow following year	-0.0171	-0.0171	$\cdots$
	Precip	-0.1328	-0.1328	$\cdots$
	PackAnn	-0.0133	-0.0133	$\cdots$
	PackD	-0.0170	-0.0170	$\cdots$
	standard deviation	0.5053	0.5053	$\cdots$

Table 3.7: Stochastic elasticities for parameters in recruit mass function; the $e_{S,i}$ are the stochastic elasticities, $e^{\mu}_{S,i}$ are the stochastic elasticities to the mean, and $e^{\sigma}_{S,i}$ are the stochastic elasticities to the standard deviation

In the survival function, parameters related to temperature were more important than the ones related to snows or rains to λ_S (Figure 3.2). In particular, parameter increase of mean temperature from June 28 to July 18 was most influential to the growth rate. An increase of mean temperate from April 26 to May 7 increased the survival probability of individuals, but an increase of mean temperature from June 28 to July 18 had a negative effect on the survival probability of individuals. Mean annual daily snow pack from December 5 to December 15 and mean daily snow after the end of the annual active season from November 26 to December 10 showed the least impact on growth rate.

In the growth function, on the other hand, changes of parameters in the mean annual temperature and the mean temperature from June 28 to July 18 were most influential to the growth rate λ_S (Figure 3.3). Increases of mean temperature from April 12 to May 3 and mean temperature from April 26 to May 6 increased the expected size of all individuals

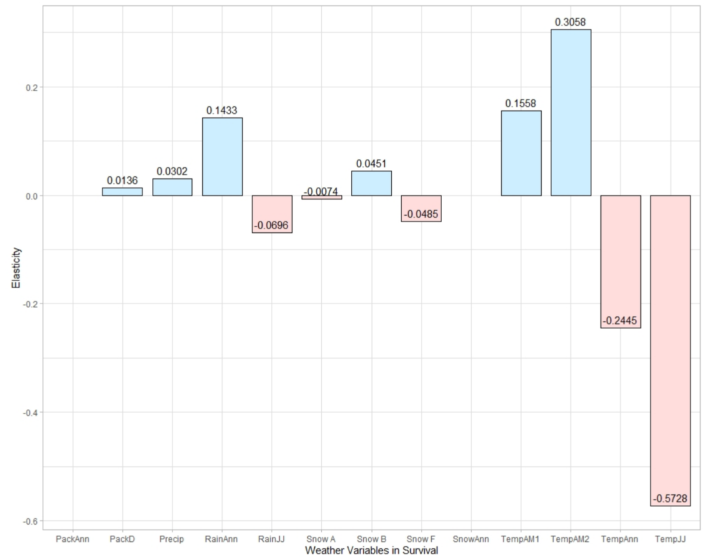


Figure 3.2: Elasticities of λ_S to the weather parameters in the survival function

after growing for one year, but increases of mean annual temperature and mean temperature from June 28 to July 18 had a negative effect on the growths of all individuals. Similarly, parameters for rains or snows showed varied effects. For example, increases of snowfall before and after the active season had a positive effect on the growth of individuals, but increase of mean rainfall from June 28 to July 11 decreased the expected size of individuals after growing for one year.

In the reproduction function, change in TempAnn parameter affected the growth rate λ_S the most (Figure 3.4). Also each parameter for Precipitation, TempAM2, RainAnn, and RainJJ demonstrated a significant effect although RainJJ had a negative effect on the reproduction probability of female adults. Parameters related to temperature and rain were more important to the growth rate λ_S than the parameters related to snow.

In the recruit mass function, TempAnn and TempJJ parameters were most important on λ_S (Figure 3.5). Also increases in TempAnn and TempJJ reduced the expected body mass of one-year olds, which enter the population as recruitments in the next year. On

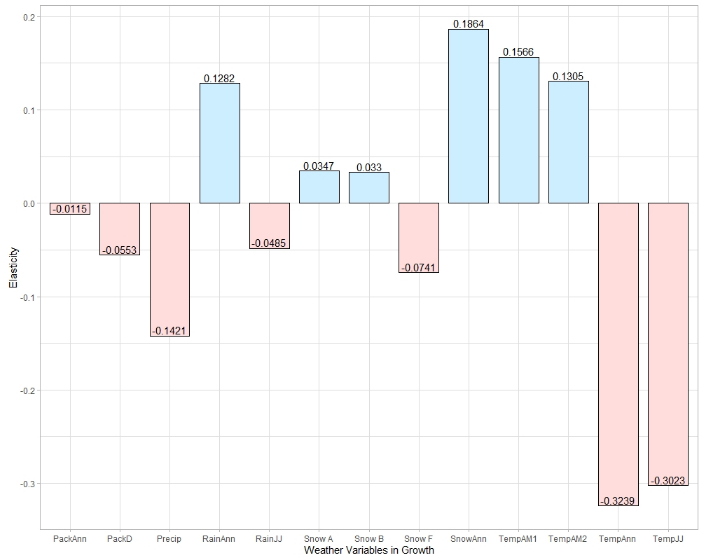


Figure 3.3: Elasticities of λ_S to the weather parameters in the growth function

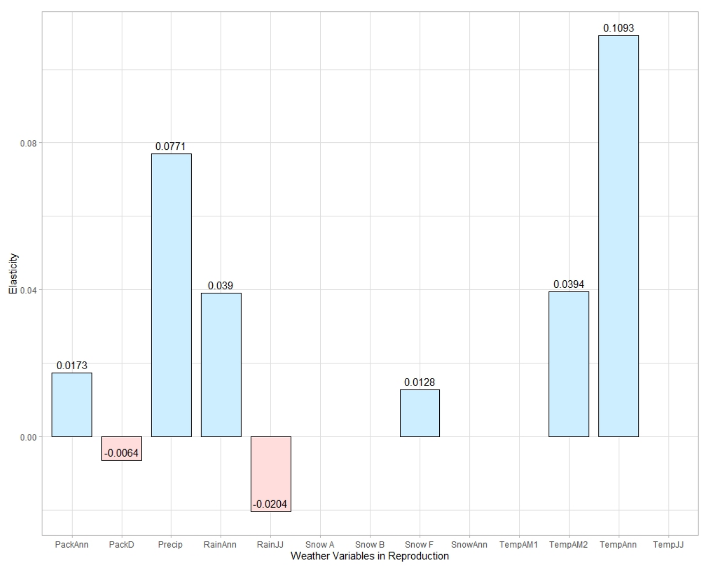


Figure 3.4: Elasticities of λ_S to the weather parameters in the reproduction function

the other hand, increased TempAM1 and TempAM2 had a positive effect on the masses of one-year old recruits.

In the litter size function, annual temperature (TempAnn parameter) showed the strongest effect on population growth rate λ_S (Figure 3.6).

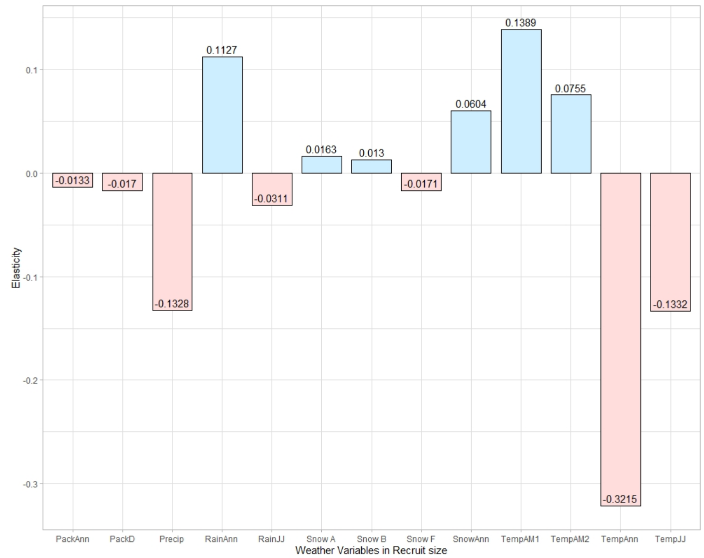


Figure 3.5: Elasticities of λ_S to the weather parameters in the recruit mass function

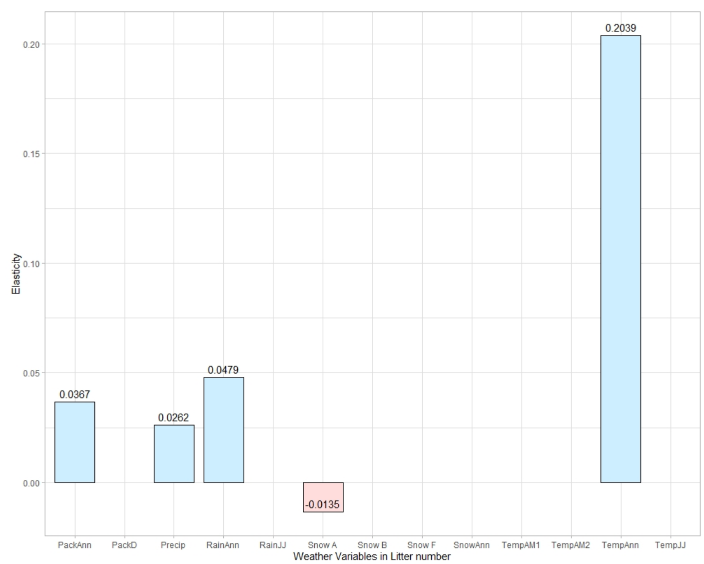
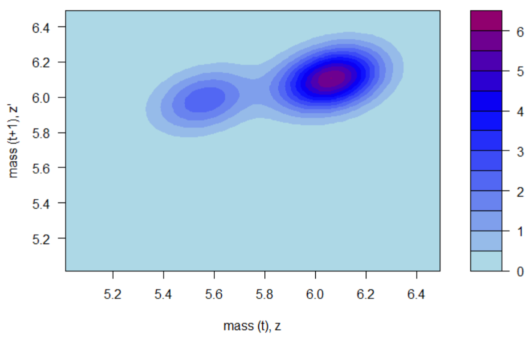


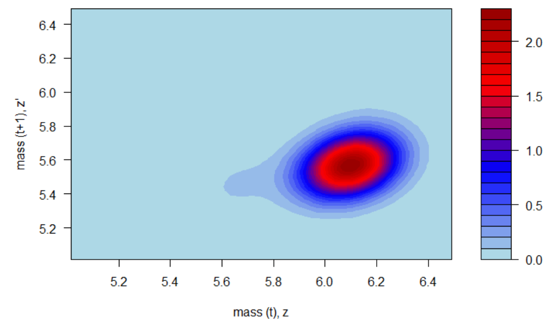
Figure 3.6: Elasticities of λ_S to the weather parameters in the nuber of offspringfunction

3.6.2 Kernel perturbation

The kernel-level perturbation analysis showed that the survival-growth component of the kernel contributed 79.26% of the elasticity (Figure 3.7A) while the fecundity component accounted for the rest of the elasticity (Figure 3.7B).



(a) Survival-growth component by LM



(b) Reproduction component by LM

Figure 3.7: Elasticity functions from kernel perturbation

This result showed that the survival-growth component had the major effect on the population growth rate. We observed that a 1% increase of the survival-growth component of the kernel at the point with transitions from mass 429 grams (≈ 6.06 in log-scale) to 450 grams (≈ 6.11 in log-scale) resulted in 6.01% increase of the growth rate (Figure 3.7A).

On the other hand, a 1% increase of the fecundity component of the kernel at the point (6.09, 5.56) in log-scale, which corresponded with the mother's mass (≈ 439 grams) and its offspring's mass (≈ 260 grams) respectively, showed the greatest effect, resulting in 2.30% increase of the growth rate. In other words, λ_s was most sensitive when offspring, born from the mothers with a mass of 439 grams, grew to be 260 grams next year and recruited to the population.

3.7 Discussion

We considered how the population of Columbian ground squirrels responded to climate changes with stochastic population models. Using the Lasso, we have determined a list of weather variables to be included to each vital rate model and further explored climate influences on population dynamics. To date, we are not aware of any work that combined constrained estimation for model selection with integral projection models. Our study uncovered climate variables that affected vital rates differentially and increased the efficiency of the integral projection model.

Our study found that the population was slowly increasing in the long run with the growth rate $\lambda_s = 1.08$. Climate effects were significant factors on the population of Columbian ground squirrels, and the growth rate λ_s was most sensitive to the changes in temperature parameters (Figure 3.8). Warm summer temperatures had a negative influence on population growth rate via their effects on individual growth and particularly survival. Temperature in spring, however, had lesser but positive influences on growth and particularly survival. These patterns accord well with previous results for this population (Lane et al. 2011; Dobson et al. 2016). As one might expect, this produced a negative influence of annual temperature on growth and survival, but also a positive influence on reproduction. The latter result likely results from the advantage of beneficial spring conditions associated with warmer weather (Lane et al. 2011).

We also observed that transitions from body mass about 446 grams (≈ 6.1 in log-scale) had the largest effect on the growth rate λ_s. These individuals weighed from 403 grams (≈ 6.0 in log-scale) to 469 grams (≈ 6.15 in log-scale), and are the adult females in prime reproductive condition (Rubach et al. 2016). As well, younger females that weigh more than 260 grams may breed in years with early spring conditions, and may have an influence on population growth (Dobson and Murie 1987; Rubach et al. 2016).

The relative contributions from survival-growth and reproduction for kernel perturbation were in agreement with those we calculated for parameter-level perturbations. From the parameter-level perturbation, we found that the sum of the elasticities to parameters in the survival and growth functions was about 77%, which was quite close with the result we obtained from the kernel perturbation.

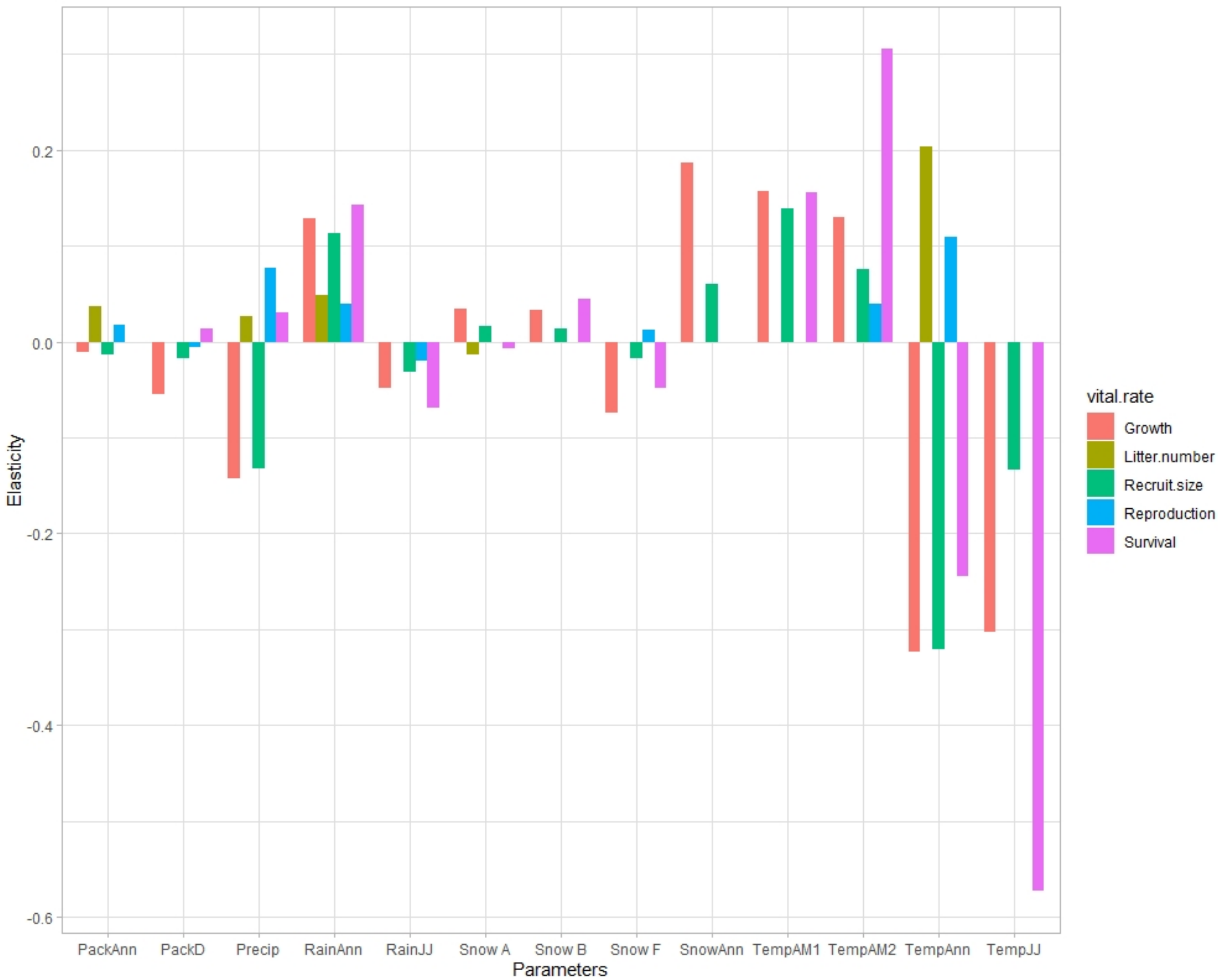


Figure 3.8: Elasticities of λ_s to the weather parameters in vital rate functions

Our vital rate functions had the form

$$\mu_{t+1} = \beta_{0,t} + \beta_1 z + \beta_{c,1}\theta_{1,t} \cdots + \beta_{c,j}\theta_{j,t} + \varepsilon_t \, , \tag{3.5}$$

where μ_{t+1} could represent the expected log body mass or the logit of survival probability for individuals at time $t + 1$. In most literatures on IPMs, parametric models like the equation (3.5) have been used to describe vital rates functions: linear or generalized linear models in deterministic IPMs, and linear mixed effects models, including our study, in the stochastic IPMs. However, parametric models are restrictive with some of the model assumptions although they are easy to use and provide interpretable parameter estimation. Fully nonparametric models are flexible, but do not provide interpretable coefficients. To avoid some drawbacks of parametric or nonparametric models, we can use semiparametric models such as single-index models, which achieve more flexibility than fully parametric models, and greater estimation precision than nonparametric models. We are interested in examining whether the same result persists between the parametric model and the nonparametric model.

Chapter 4

Semi-Parametric Integral Projection Models

4.1 Introduction

The majority of integral projection models (IPMs) in the literature used parametric models to describe the demographic processes. Linear or generalized linear models in IPMs have been commonly used. Although these simpler structures make it easier to obtain estimates of effects of variables on vital rates of population dynamics, they lack the flexibility to capture nonlinearities that are common in relationships of vital rates and covariates in population dynamics. For instance, we often have threshold effects that cannot be captured by linear regression coefficients. Nonlinearities may often be unknown and have to be nonparametrically estimated from the data. At the same time, we want to be able to characterize the effect of variables on vital rates. The ability to fit IPMs that combine the flexibility of nonparametric function estimation and the ability to obtain simple estimates of effects of variables on vital rates is of paramount importance in ecological studies. Moreover, such flexibility in estimating parameters particularly in the presence of outliers as well as departures or errors from model assumptions would be a big step forward in ecological modeling.

Thus, our goal in the present study is to explore the use of more flexible, semi-parametric models to extend the IPM approach. Considering that there is no completely correct model for a given data set, we investigated different models, some of which may be better than others as tools for evaluating the validity of different results obtained. The responses Y_i are assumed to follow an exponential family distribution with $E(Y_i) = \mu_i$. Survival and reproduction are assumed to follow a binomial distribution, the litter size is assumed to follow a Poisson distribution, and size variables are assumed to follow Gaussian distributions. We considered the following two semi-parametric generalized model formulations to model vital rate functions:

$$G(\mu_i) = f(z_i) + \beta_1 w_{i1} \cdots + \beta_p w_{ip} \tag{4.1}$$

$$G(\mu_i) = f(z_i) + h(\beta_1 w_{i1} \cdots + \beta_p w_{ip}) \tag{4.2}$$

The equation (4.1) is a generalized partial linear model where body mass is related to the mean response (up to a known link function G) through a smooth function, but the climate variables are linearly related to the mean response. The equation (4.2) is a generalized model with a known link function G where body mass is again related to the mean response through a smooth function, but climate variables are related to the response as single-index model through an unknown smooth function h. In both cases, we have some flexibility provided by the unknown smooth functions but a linearity assumption on the climate variables to enable us to derive elasticities that represent the effect of individual weather variables on the mean response. The main reason for exploring such models is that in addition to measuring the effect of size on vital rates, we want to factor-out its effect to be able to measure the contribution of weather variables to changes in vital rates. For such purposes, the size variable is treated as a nuisance variable whose effect needs to be accounted for as flexibly as possible. Thus, following the fitting of the IPM using the equations (4.1) and (4.2), we estimated growth rates and conducted perturbation analyses to investigate weather effects.

4.2 Partial Linear Models

A generalized partial linear model consists of two additive components, a linear and a nonparametric part:

$$G(\mu_i) = f(z_i) + \alpha_1 w_{i1} \cdots + \alpha_p w_{ip} \ ,$$

where $\mu_i = E(Y_i)$, G is a known monotonic link function, $w_1, \cdots, w_p$ are climate variables, $f : \mathbb{R} \to \mathbb{R}$ is a smooth, unknown function of the log body mass z (called as a smoother), and $\alpha_1, \cdots, \alpha_p$ are unknown parameters.

To model vital rate functions, we used `gam()` in the R package `mgcv`. A smooth function f is estimated using penalized regression splines, e.g., cubic penalized regression splines for a single predictor (Wood 2017). The smoothing parameter, which controls the smoothness or wiggliness of the smooth function f, is chosen using generalized cross validation (GCV). If f has bounded second derivatives, then we can show that the spline estimator has favorable convergence properties (Schumaker 1981).

4.3 Single Index Models

Our generalized single index model for vital functions has the following form:

$$G(\mu_i) = f(z_i) + h(\beta_1 w_{i1} \cdots + \beta_p w_{ip})$$

where $\mu_i = E(Y_i)$, G is a known monotonic link function, $w_1, \cdots, w_p$ are climate variables, $\beta_1, \cdots, \beta_p$ are unknown parameters, and f and h are unknown smooth functions. This model is not identifiable as presented since changes in the size of $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)'$ can be compensated by changes in the function h. Thus we need an extra assumption for the model to be identifiable. One approach to make the model identifiable is to fix the size of $\boldsymbol{\beta}$. To that end, the assumption we will use is that $\boldsymbol{\beta}$ resides on the surface of the upper half of the unit hypersphere. In other words, we will assume that $\|\boldsymbol{\beta}\| = 1$ and $\beta_1 > 0$.

A single index model (SIM) summarizes the effects of the explanatory variables within a single variable called the index. By summarizing all the information contained in the variables $w_1, \cdots, w_p$ into one "single index" term, we will greatly reduce the dimensionality of a problem. So, the single index model was developed as a possible remedy to overcome the "curse of dimensionality," which arises when using nonparametric multivariate regression methods. The curse of dimensionality is a phenomenon that as the number of predictors increase, the performance of the fitted model decreases due to the sparsity of data in high dimensional space (e.g., James et al. 2013). The single index model is also a formulation that allows us to measure the effects of individual explanatory variables.

We again used the generalized cross-validation framework of Wood (2017) to estimate the smooth functions. We combined this a with constrained optimization subject to $\|\boldsymbol{\beta}\| = 1$ and $\beta_1 > 0$ via the `optim()` function of R. This combination of estimating parameters of climate variables and finding the optimal smoothing parameter using the maximum likelihood estimation procedure were implemented through the `gam()` function of R. An example code that shows how this is performed is given in the Appendix.

4.4 Results

The final vital rate models by partial linear models and single index models are shown in Tables 4.1 - 4.3 and Tables 4.4 - 4.6 respectively. Figures 4.1, 4.2, and 4.3 give the estimated smooth functions f and h.

For the generalized partial linear approach, TempAM2 had a large positive impact on the survival rate while TempJJ and RainJJ both had a moderate but significant ($P < .05$) negative effect on survival. Average growth was negatively affected by increasing levels of

TempAnn, TempJJ, RainJJ, Snow following year, and PackD. Average growth was positively affected by TempAM2 and SnowAnn. The reproduction rate was positively affected by increasing levels of TempAnn, TempAM2, Precip, and PackAnn while it was negatively affected by increasing RainJJ. The average litter size was positively related to increasing TempAnn and PackAnn. Finally the average size of recruits was positively related to TempAM1, TempAM2 and RainAnn while it was negatively related to TempJJ, RainJJ, and PackAnn. As expected, we can see in Figure 4.1 that mass has an increasing relationship with survival, reproduction, the litter size, recruit mass, and obviously, the size of the animal the following year. However, there appears to be a threshold effect in the effect of mass on survival where the effect on survival does not change beyond around mass = 5.7 (298.8 grams).

For the generalized single index model, once again we notice that all vital rates are positively related to mass. However, the interpretation of the effect of climate variables on vital rates is dependent upon the shape of the estimate of the function h. For example, consider the estimation of reproduction probability. We notice from Table 4.5 that the coefficient for RainAnn is -0.88. This means high values of RainAnn lead to low values of the estimated index $\hat{\beta}'\mathbf{w}$. However, from Figure 4.2, we see that the estimate of the function h fluctuates around a constant at low values of the index $\hat{\beta}'\mathbf{w}$, which then becomes a quadratic curve with a deep valley as the index increases. Thus, since increasing values of RainAnn decrease the index $\hat{\beta}'\mathbf{w}$, the high values of RainAnn have no substantial impact on the probability of reproduction and the effect turning to decreasing and then increasing as RainAnn decreases.

As in the general linear model approach, we will further study the elasticities computed from the models to provide an effect direction and size of all variables.

Vital rates	Function	Parameter (SE) estimates
Survival	logit$[S(z;\boldsymbol{w}_t)]$=$f_s(z)$ $+\beta_{s,2}$ TempAnn$+\beta_{s,3}$ TempAM1 $+\beta_{s,4}$ TempAM2 $+\beta_{s,5}$ TempJJ$+\beta_{s,6}$ RainAnn$+\beta_{s,7}$ RainJJ $+\beta_{s,8}$ Snow before $+\beta_{s,9}$ Snow after $+\beta_{s,10}$ Snow following year $+\beta_{s,11}$ Precip$+\beta_{s,12}$ PackD	$(\beta_0 = 2.264(1.168))$ $\beta_{s,2} = -0.274(0.152)$ $\beta_{s,3} = 0.151(0.089)$ $\beta_{s,4} = 0.244(0.044)$ $\beta_{s,5} = -0.182(0.056)$ $\beta_{s,6} = 0.767(0.612)$ $\beta_{s,7} = -0.178(0.076)$ $\beta_{s,8} = 0.220(0.154)$ $\beta_{s,9} = -0.042(0.211)$ $\beta_{s,10} = -0.281(0.167)$ $\beta_{s,11} = 0.079(0.515)$ $\beta_{s,12} = 0.012(0.021)$
Growth	$\mu_g(z,\boldsymbol{w}_t)$=$f_g(z)$ $+\ \beta_{g,2}$ TempAnn $+\beta_{g,3}$ TempAM1 $+\beta_{g,4}$ TempAM2 $+\beta_{g,5}$ TempJJ $+\beta_{g,6}$ RainAnn $+\beta_{g,7}$ RainJJ $+\beta_{g,8}$ SnowAnn $+\beta_{g,9}$ Snow before $+\ \beta_{g,10}$ Snow after $+\beta_{g,11}$ Snow following year $+\beta_{g,12}$ Precip $+\beta_{g,13}$ PackAnn $+\beta_{g,14}$ PackD $\sigma_g = 0.1121$	$(\beta_0 = 6.092(0.060))$ $\beta_{g,2} = -0.026(0.008)$ $\beta_{g,3} = 0.009(0.005)$ $\beta_{g,4} = 0.007(0.003)$ $\beta_{g,5} = -0.006(0.003)$ $\beta_{g,6} = 0.050(0.035)$ $\beta_{g,7} = -0.008(0.004)$ $\beta_{g,8} = 0.205(0.075)$ $\beta_{g,9} = 0.019(0.009)$ $\beta_{g,10} = 0.025(0.016)$ $\beta_{g,11} = -0.027(0.010)$ $\beta_{g,12} = -0.001(0.003)$ $\beta_{g,13} = -0.036(0.026)$ $\beta_{g,14} = -0.003(0.001)$

Table 4.1: Estimated survival and growth functions using partial linear models.

4.4.1 Estimating growth rates

The Columbian ground squirrel population growth rate estimated using the generalized partial linear IPM was

$$\log \lambda_s = 0.0521\ (\lambda_s = 1.0535)$$

Vital rates	Function	Parameter (SE) estimates
Reproduction	logit$[p_b(z;\boldsymbol{w}_t)]$=$f_b(z)$ $+\beta_{b,2}$ TempAnn $+\beta_{b,3}$ TempAM2 $+\beta_{b,4}$ RainAnn$+\beta_{b,5}$ RainJJ $+\beta_{b,6}$ Snow following year $+\beta_{b,7}$ Precip$+\beta_{b,8}$ PackAnn $+\beta_{b,9}$ PackD	$(\beta_0 = -5.689(0.808))$ $\beta_{b,2} = 0.468(0.144)$ $\beta_{b,3} = 0.118(0.044)$ $\beta_{b,4} = 0.869(0.659)$ $\beta_{b,5} = -0.164(0.070)$ $\beta_{b,6} = 0.296(0.172)$ $\beta_{b,7} = 1.011(0.509)$ $\beta_{b,8} = 0.102(0.045)$ $\beta_{b,9} = -0.017(0.019)$
litter size	log$(b(z;\boldsymbol{w}_t))$=$f_l(z)$ + $\beta_{l,2}$ TempAnn $+\beta_{l,3}$ RainAnn + $\beta_{l,4}$ Snow after $+\beta_{l,5}$ Precip $+\beta_{l,6}$ PackAnn	$(\beta_0 = -0.968(0.399))$ $\beta_{l,2} = 0.183(0.061)$ $\beta_{l,3} = 0.202(0.228)$ $\beta_{l,4} = -0.099(0.081)$ $\beta_{l,5} = 0.076(0.185)$ $\beta_{l,6} = 0.046(0.019)$

Table 4.2: Estimated reproduction and litter size functions using partial linear models.

with 95% confidence interval $(0.0505, 0.0537)$. The result is similar to that obtained using the single index IPM

$$\log \lambda_S = 0.0458 \ (\lambda_S = 1.0469)$$

with 95% confidence interval $(0.0443, 0.0473)$.

Considering the growth rate $\log \lambda_S = 0.0784$ ($\lambda_s = 1.0816$) from the linear model, it is reassuring that all three estimates of the growth rates were similar, indicating that the population slowly increased in the long run. Given that there is variation in the estimated population grown rates, we performed a stability analysis to select the most appropriate model to use.

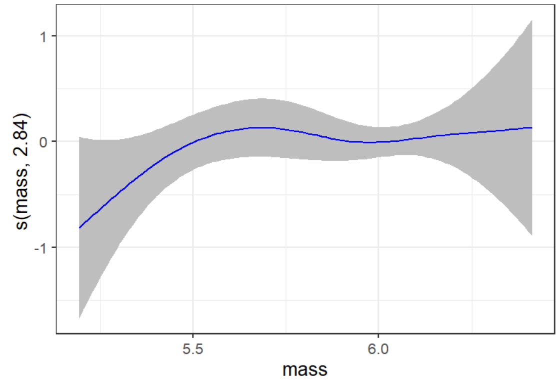
Vital rates	Function	Parameter (SE) estimates
Recruitment	$p_r = 0.42797$	
recruit mass	$\mu_r(z, \boldsymbol{w}_t) = f_r(z) + \beta_{r,2}$ TempAnn $+\beta_{r,3}$ TempAM1 $+\beta_{r,4}$ TempAM2 $+\beta_{r,5}$ TempJJ $+\beta_{r,6}$ RainAnn $+\beta_{r,7}$ RainJJ $+\beta_{r,8}$ SnowAnn $+\beta_{r,9}$ Snow before $+\ \beta_{r,10}$ Snow after $+\beta_{r,11}$ Snow following year $+\beta_{r,12}$ Precip $+\beta_{r,13}$ PackAnn $+\beta_{r,14}$ PackD	$(\beta_0 = 5.838(0.092))$ $\beta_{r,2} = -0.099(0.015)$ $\beta_{r,3} = 0.032(0.009)$ $\beta_{r,4} = 0.019(0.004)$ $\beta_{r,5} = -0.011(0.005)$ $\beta_{r,6} = 0.153(0.054)$ $\beta_{r,7} = -0.017(0.006)$ $\beta_{r,8} = 0.183(0.117)$ $\beta_{r,9} = 0.017(0.014)$ $\beta_{r,10} = 0.048(0.026)$ $\beta_{r,11} = -0.024(0.017)$ $\beta_{r,12} = -0.005(0.004)$ $\beta_{r,13} = -0.109(0.039)$
	$\sigma_r = 0.1104$	$\beta_{r,14} = -0.003(0.002)$

Table 4.3: The probability of successful recruitment, and estimated recruit mass function using partial linear models.

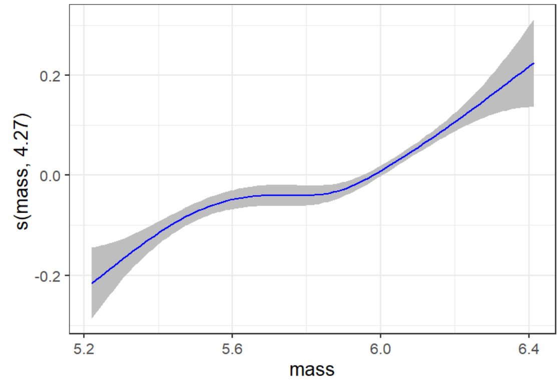
4.4.2 Stability Analysis

We designed a bootstrapping approach to determine how much the estimate of $\log \lambda_S$ is likely to vary from the original data ($n = 1704$) to sub-samples with different sample sizes ($n = 1689$ and $n = 1685$). This is used to understand the approach among the generalized linear, partial linear, and single index models that gave us results that are more stable. The smaller the variation, the more stable the procedure. We conducted the bootstrap samplings with 100 and 1000 iterations to compute the standard deviations. The result found from the bootstrap method is shown in Table 4.7.

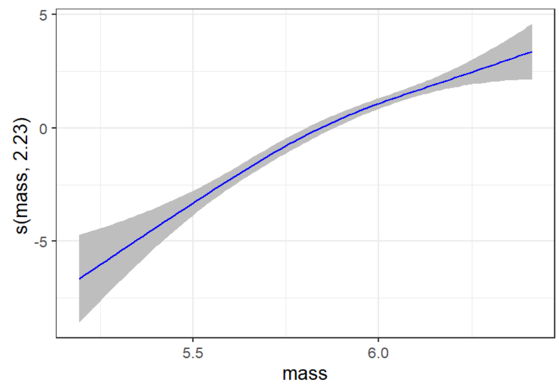
We observed that the partial linear model showed the smallest variation both in sample sizes $n = 1689$ and $n = 1685$ in 100 iterations, and the linear model had the largest variance. In 1000 iterations, the result was consistent in that the partial linear model performed



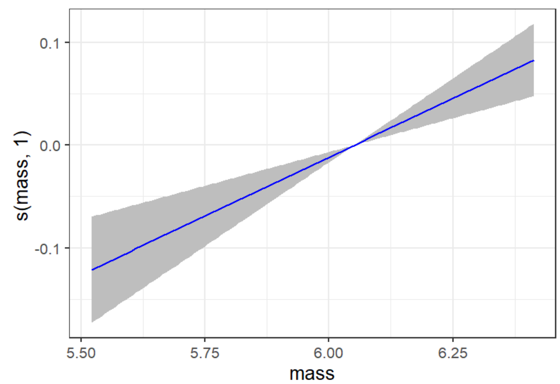
(a) mass in survival



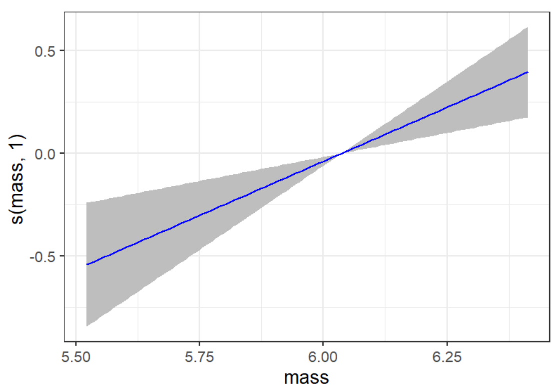
(b) mass in growth



(c) mass in reproduction



(d) mass in recruit



(e) mass in litter size

Figure 4.1: Plots of smooth functions by partial linear models

the best. However, the single index model showed the smallest change between different sample sizes. This relatively small change of the single index model was still persistent when the bootstrap sampling was iterated 1000 times. The result suggests that the proposed

Vital rates	Function	Parameter estimates
Survival	logit$[S(z;\boldsymbol{w}_t)]$=$f_s(z)$ +h_S($\beta_{s,2}$ TempAnn+$\beta_{s,3}$ TempAM1 +$\beta_{s,4}$ TempAM2 +$\beta_{s,5}$ TempJJ+$\beta_{s,6}$ RainAnn+$\beta_{s,7}$ RainJJ +$\beta_{s,8}$ Snow before +$\beta_{s,9}$ Snow after +$\beta_{s,10}$ Snow following year +$\beta_{s,11}$ Precip+$\beta_{s,12}$ PackD)	($\beta_0 = 1.0341$) $\beta_{s,2} = 0.0156$ $\beta_{s,3} = 0.1523$ $\beta_{s,4} = 0.2357$ $\beta_{s,5} = -0.4312$ $\beta_{s,6} = 0.0359$ $\beta_{s,7} = -0.2150$ $\beta_{s,8} = 0.3167$ $\beta_{s,9} = 0.1340$ $\beta_{s,10} = -0.5516$ $\beta_{s,11} = 0.5136$ $\beta_{s,12} = -0.0357$
Growth	$\mu_g(z,\boldsymbol{w}_t)$=$f_g(z)$ + h_g($\beta_{g,2}$ TempAnn +$\beta_{g,3}$ TempAM1 +$\beta_{g,4}$ TempAM2 +$\beta_{g,5}$ TempJJ +$\beta_{g,6}$ RainAnn +$\beta_{g,7}$ RainJJ +$\beta_{g,8}$ SnowAnn +$\beta_{g,9}$ Snow before + $\beta_{g,10}$ Snow after +$\beta_{g,11}$ Snow following year +$\beta_{g,12}$ Precip +$\beta_{g,13}$ PackAnn +$\beta_{g,14}$ PackD) $\sigma_g = 0.1094$	($\beta_0 = 6.0183$) $\beta_{g,2} = 0.0236$ $\beta_{g,3} = 0.1358$ $\beta_{g,4} = -0.0628$ $\beta_{g,5} = 0.0312$ $\beta_{g,6} = 0.0190$ $\beta_{g,7} = -0.0030$ $\beta_{g,8} = 0.9057$ $\beta_{g,9} = -0.1799$ $\beta_{g,10} = -0.1797$ $\beta_{g,11} = 0.0696$ $\beta_{g,12} = -0.1585$ $\beta_{g,13} = 0.2243$ $\beta_{g,14} = 0.1024$

Table 4.4: Estimated survival and growth functions using single index models.

semiparametric IPM approaches provide more stable estimates than the current generalized linear model based IPMs.

Vital rates	Function	Parameter estimates
Reproduction	$\text{logit}[p_b(z;\boldsymbol{w}_t)]=f_b(z)+h_{P_b}(\beta_{b,2}$ TempAnn $+\beta_{b,3}$ TempAM2 $+\beta_{b,4}$ RainAnn$+\beta_{b,5}$ RainJJ $+\beta_{b,6}$ Snow following year $+\beta_{b,7}$ Precip$+\beta_{b,8}$ PackAnn $+\beta_{b,9}$ PackD)	($\beta_0=-0.1962$) $\beta_{b,2}=0.2110$ $\beta_{b,3}=0.0284$ $\beta_{b,4}=-0.8808$ $\beta_{b,5}=0.1111$ $\beta_{b,6}=-0.0418$ $\beta_{b,7}=0.2763$ $\beta_{b,8}=0.1345$ $\beta_{b,9}=0.2653$
litter size	$\log(b(z;\boldsymbol{w}_t))=f_l(z)+h_b$ ($\beta_{l,2}$ TempAnn $+\beta_{l,3}$ RainAnn $+\beta_{l,4}$ Snow after $+\beta_{l,5}$ Precip $+\beta_{l,6}$ PackAnn)	($\beta_0=0.3343$) $\beta_{l,2}=0.6031$ $\beta_{l,3}=0.6648$ $\beta_{l,4}=-0.3289$ $\beta_{l,5}=0.2506$ $\beta_{l,6}=0.1531$

Table 4.5: Estimated reproduction and litter size using single index models.

4.4.3 Parameter perturbation

To find the elasticity of $\log\lambda_S$ to changes in parameters, we need to find the perturbation kernel C_t (Section 3.6) for parameters in vital rate functions. For partial linear models, since the climate variables enter the model linearly, C_t was similar to the one for linear models we considered in Chapter 3. But, for single index integral projection models, we need to additionally evaluate the derivatives of estimated smooth functions h. The full list of C_t for parameters in single index models is listed in Table 4.8.

The derivatives have to be evaluated numerically for the estimated h values. This can be done using the method of finite differences which is implemented in R. To numerically evaluate the derivatives of smooth functions, we used `derivatives()` in the R package `gratia`, which evaluates the derivatives of estimated smooth functions via finite differences.

Vital rates	Function	Parameter estimates
Recruitment	$p_r = 0.42797$	
recruit mass	$\mu_r(z, \boldsymbol{w}_t)$=$f_r(z)$ + h_r ($\beta_{r,2}$ TempAnn +$\beta_{r,3}$ TempAM1 +$\beta_{r,4}$ TempAM2 +$\beta_{r,5}$ TempJJ +$\beta_{r,6}$ RainAnn +$\beta_{r,7}$ RainJJ +$\beta_{r,8}$ SnowAnn +$\beta_{r,9}$ Snow before + $\beta_{r,10}$ Snow after +$\beta_{r,11}$ Snow following year +$\beta_{r,12}$ Precip +$\beta_{r,13}$ PackAnn +$\beta_{r,14}$ PackD)	($\beta_0 = 5.526$) $\beta_{r,2} = 0.1299$ $\beta_{r,3} = 0.2028$ $\beta_{r,4} = 0.1740$ $\beta_{r,5} = 0.1196$ $\beta_{r,6} = 0.2452$ $\beta_{r,7} = 0.2559$
	$\sigma_r = 0.10774$	$\beta_{r,8} = -0.7194$ $\beta_{r,9} = 0.2208$ $\beta_{r,10} = -0.2193$ $\beta_{r,11} = -0.1455$ $\beta_{r,12} = 0.2975$ $\beta_{r,13} = 0.1791$ $\beta_{r,14} = 0.1250$

Table 4.6: The probability of successful recruitment, and estimated recruit mass function using using single index models.

Based on the formula listed in Table 4.8, the elasticities to climate parameters are computed as shown in Table 5.1 - Table 5.4.

In the survival function, both models showed that parameters related to temperature were more important than the ones related to snow or rain to the growth rate λ_S (Figure 5.1). In particular, the parameter change of mean temperature from June 28 to July 18 (TempJJ) had the greatest impact on the population growth rate. These results were consistent with the ones from the linear model.

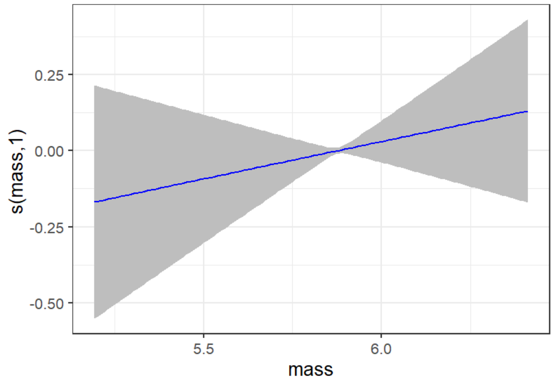
The figure 5.2 of the growth function showed that parameters related to temperature were important to the population growth, which is consistent to the linear mixed model. But, unlike the linear or partial linear models, parameters related to snow pack (PackAnn, PackD) showed large impacts to λ_S in the single index model.

Sample Size	Models	100 iterations Mean	SD	1000 iterations Mean	SD
$n = 1689$	LM	0.07701	0.00475	0.07627	0.00479
	PLM	0.05113	0.00188	0.05071	0.00191
	SIM	0.04406	0.00202	0.04434	0.00202
$n = 1685$	LM	0.07645	0.00660	0.07595	0.00567
	PLM	0.05063	0.00224	0.05069	0.00215
	SIM	0.04395	0.00230	0.04407	0.00224

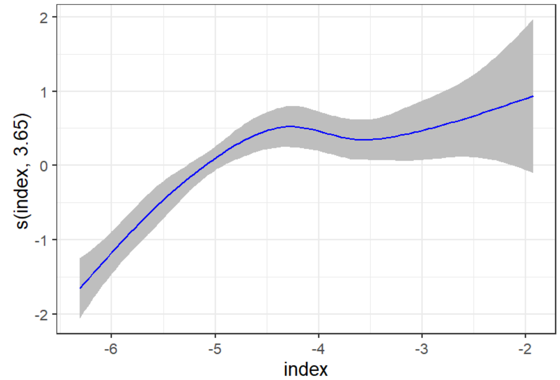
Table 4.7: Mean and standard deviation of $\log \lambda_S$ from samples of sizes n=1689 and n=1685. LM: Linear Models; PLM: Partial Linear Models; SIM: Single Index Models

Vital Rates	C_t
Survival	$g(z', z, \mathbf{w}_t) \cdot s(z, \mathbf{w}_t) \cdot \dfrac{h_s'(\boldsymbol{\beta}'\mathbf{x}) \cdot x_i}{1 + \exp(m_s)} \cdot \beta_{s,i}$
Growth	$g(z', z, \mathbf{w}_t) \cdot s(z, \mathbf{w}_t) \cdot \dfrac{z' - m_g}{\sigma_g^2} \cdot h_g'(\boldsymbol{\beta}'\mathbf{x}) \cdot x_i \cdot \beta_{g,i}$
Reproduction	$p_b(z, \mathbf{w}_t) \cdot b(z, \mathbf{w}_t) \cdot p_r \cdot C_1(z', z, \mathbf{w}_t) \dfrac{h_{P_b}'(\boldsymbol{\beta}'\mathbf{x}) \cdot x_i}{1 + \exp(m_{P_b})} \cdot \beta_{P_b,i}$
litter size	$p_b(z, \mathbf{w}_t) \cdot b(z, \mathbf{w}_t) \cdot p_r \cdot C_1(z', z, \mathbf{w}_t) \cdot h_b'(\boldsymbol{\beta}'\mathbf{x}) \cdot x_i \cdot \beta_{l,i}$
recruit mass	$p_b(z, \mathbf{w}_t) \cdot b(z, \mathbf{w}_t) \cdot p_r \cdot C_1(z', z, \mathbf{w}_t) \dfrac{z' - m_r}{\sigma_r^2} \cdot h_r'(\boldsymbol{\beta}'\mathbf{x}) \cdot x_i \cdot \beta_{r,i}$

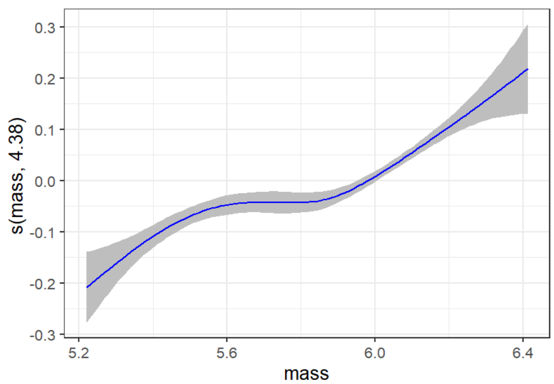
Table 4.8: Perturbation kernels for single index models. $m_j = f_j(z) + h_j(\boldsymbol{\beta}'\mathbf{x})$, $h_j'(\cdot)$ is a derivative of the single index component in each vital rate function, and x_i is a climate variable.



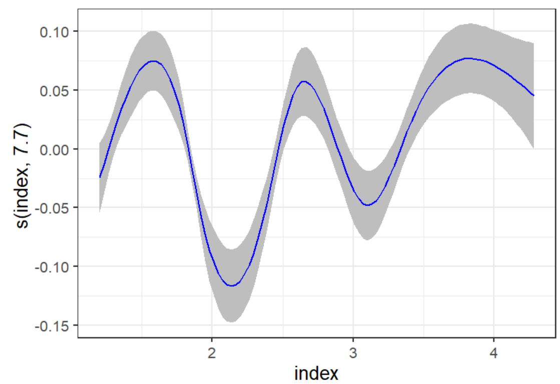
(a) mass in survival



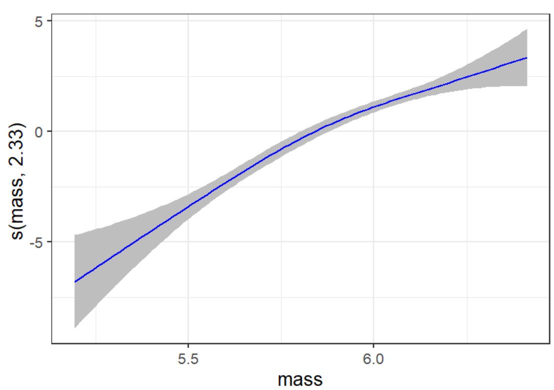
(b) index in survival



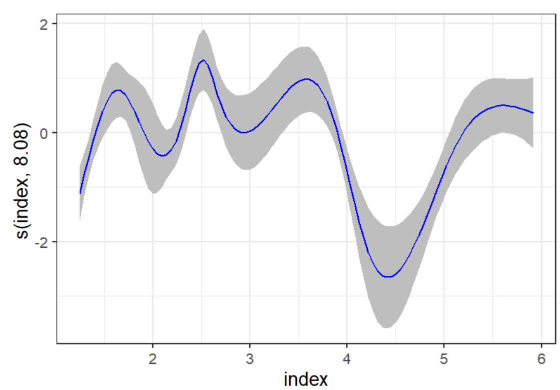
(c) mass in growth



(d) index in growth



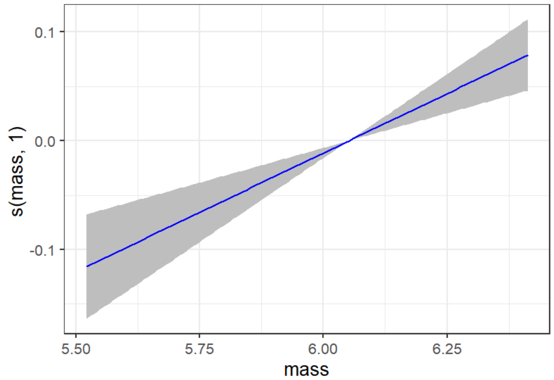
(e) mass in reproduction



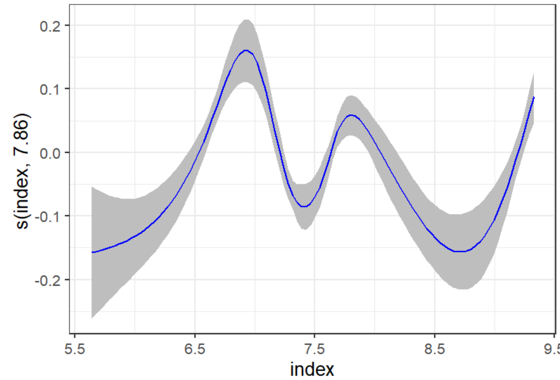
(f) index in reproduction

Figure 4.2: Plots of smooth functions by single index models

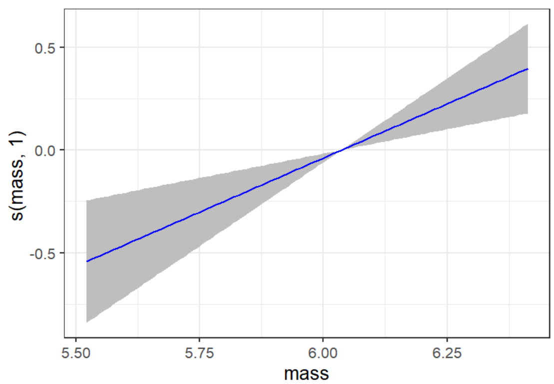
In the reproduction function, the parameter change in TempAnn (mean annual temperature) in the partial linear model had the largest impact on λ_S. However, in the single



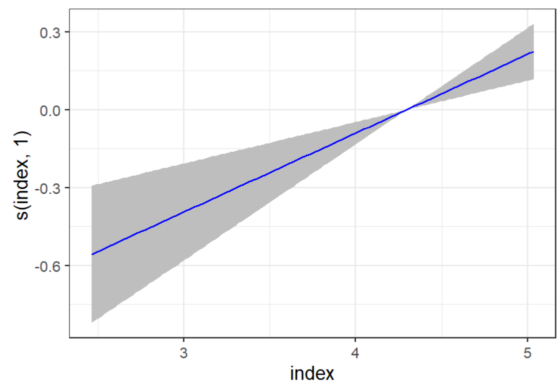
(a) mass in recruit



(b) index in recruit



(c) mass in litter size



(d) index in litter size

Figure 4.3: Plots of smooth functions by single index models

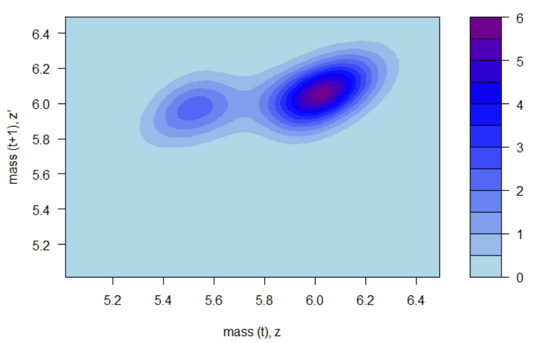
index model, PackD (mean annual daily snow pack from December 5 to December 15) had the largest elasticity (Figure 5.3).

In the litter size function, the elasticity in all three models showed very similar patterns. TempAnn (mean annual temperature) had the largest effect on the population growth rate λ_S (Figure 5.4).

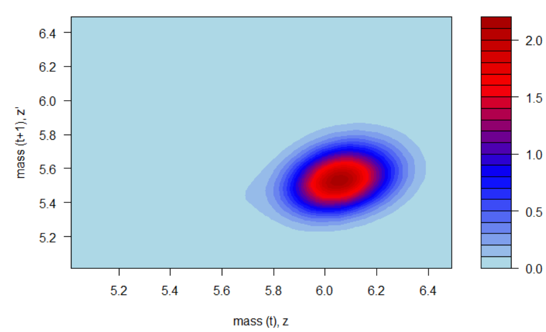
In the recruit mass function, the parameters related to temperature had significant effects on the population growth rate in all three models. The elasticity of TempAnn parameter showed the largest impact on the growth rate in the linear and partial linear models, but TempJJ showed the largest effect in the single index model (Figure 5.5).

4.4.4 Kernel Perturbation

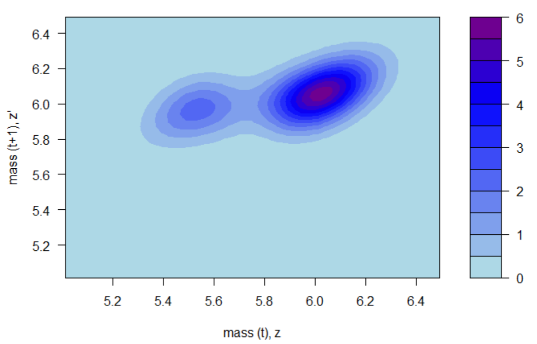
Figure 4.4 shows the contour plots of elasticities of λ_S to kernel perturbations in the partial linear models (PLMs) and single index models (SIMs). We see that there is very little difference in the shape of the contour plots, which means that the local change of kernels in the PLMs and SIMs have similar effects on the population growth rate. For example, a 1% increase of the survival-growth component of the kernel at the point with transitions from mass 419 grams (≈ 6.04 in log-scale) to 429 grams (≈ 6.06 in log-scale) resulted in 5.75% increase of the growth rate in the PLMs (Figure 4.4a) and 5.80% increase of the growth rate in the SIMs (Figure 4.4c). Similarly, a 1% increase of the fecundity component of the kernel at the point (6.06, 5.54) in log-scale, which corresponded with the mother's mass (≈ 429 grams) and its offspring's mass (≈ 254 grams) respectively, had the greatest effect, resulting in 2.18% increase of the growth rate in the PLMs (Figure 4.4b) and 2.15% increase of the growth rate in the SIMs (Figure 4.4d).



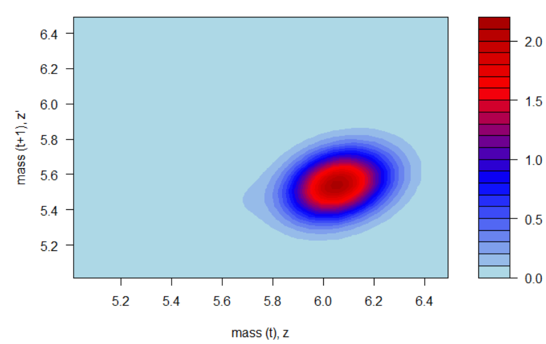
(a) Survival-growth component by PLM



(b) Reproduction component by PLM



(c) Survival-growth component by SIM



(d) Reproduction component by SIM

Figure 4.4: Elasticity functions from kernel perturbation

Chapter 5

Discussion and Future Work

Models help us understand the population dynamics in the data. Since there is no fully right model, it is important to explore different models in search of the best-fit model for the data. In this dissertation, we studied four population dynamics models and modeled the effect of climate as well as individual characteristics on population vital rates. The first one is the traditional state-space model, which is a generalization of matrix population models to include the effect of other covariates as well as stochasticity seamlessly. This, however, does not allow individual characteristics to be included. The last three are variations of the integral projection model (IPM). One is based on parametric generalized linear model formulation. This assumes linear relationships (up to known link functions) between vital rates and covariates. This has two major limitations. The first is that such relationships are rarely (if ever) linear. The second is including obvious influencers like size in a linear regression obfuscates the measurement of the effect of other covariates. In a sense, we want to treat this variable as an unknown, but flexible, nuisance to be integrated out for the purpose of measuring the effect of covariates like climate variables. Thus the last two models are novel semiparametric IPM formulations where we allow the effect of size on vital rates to be non-parametric: one where climate effects are assumed to be linear (PLM: Partial Linear Model) and another where climate effects are assumed to be via a single index model (SIM).

Reassuringly, all four models indicated that the population of Columbian ground squirrels was slowly increasing in the long run. The shape of the kernel perturbation showed little difference in all three IPM models. In the linear model, the largest elasticity was 6.01% at the point transitioning from mass 429 grams (≈ 6.06 in log-scale) to 450 grams (≈ 6.11 in log-scale) as shown in the Figure 3.7A. However, in the PLMs and SIMs, the largest elasticities were 5.75% and 5.80% each, at the point transitioning from 419 grams (≈ 6.04 in log-scale) to 429 grams (≈ 6.06 in log-scale) as shown in Figure 4.4.

In parameter perturbations, we found that all three models revealed that the population growth rate was most sensitive to climate variables associated with temperature. The partial linear model showed a similar pattern to the linear model while the elasticities from the SIMs showed some discrepancies with the linear models and PLMs. The variables associated with snow (PackD and PackAnn) in the SIMs were suggested to be important to the growth rate, while the effects of these variables appeared small in the linear models and PLMs.

In all of our evaluations, we applied the LASSO procedure to select the relevant variables in the model. Previous IPM evaluations either did not have a mechanism for selecting variables or depended on simple correlations among variables combined with subjective expert evaluations to decide on variables to be included. To our knowledge, this is the first work to incorporate constrained optimization in the IPM framework.

When a group of predictors present a collinearity problem, the LASSO tends to select only one predictor from the group and may hide the relevance of one of the highly correlated variables (Zou and Hastie 2005). Ridge regression, on the other hand, shrinks coefficients towards zero to give similar coefficient estimates for the highly correlated variables (Zou and Hastie 2005). If predictors are highly correlated, the prediction performance of ridge regression is better than the LASSO (Tibshirani 1996). However, one disadvantage of ridge regression is that it does not perform variable selection and include all predictors in the final model. When feature selection is a main interest, the LASSO is more desirable because it provides parsimonious modeling. To combine the strengths of the LASSO and ridge regression, Zou and Hastie (2005) proposed the elastic net. The elastic net performs variable selection and once one predictor in the group of highly correlated predictors is selected, all predictors in the group will be included in the model. Therefore, if there are strongly correlated predictors, the elastic net provides a viable option.

Using a simulation experiment to compare different models within the IPM framework is difficult since the IPM kernels consist of several vital rate functions, in which many parameters need to be estimated. After taking all the components of the IPM into account, defining the relationships between the vital rate functions and the population growth rate is not easy. In other words, it is very difficult to specify parameters in vital rate functions with the given growth rate since the relationships among them are unknown due to the variations associated with the complexity of the IPM kernels.

An honest comparison of such models is quite complicated since predictive assessment via cross-validation or information criteria is not very straightforward. This is a current topic of heated discussion in the population modeling community. Further research is needed to investigate how to properly assess the differences among IPMs. We are considering theoretical and numerical investigations of such evaluations as a future work.

Vital Rate	Parameter	e_S^{LM}	e_S^{PLM}	e_S^{SIM}
Survival	TempAnn	-0.2445	-0.2487	0.0087
	TempAM1	0.1558	0.1721	0.2022
	TempAM2	0.3058	0.3172	0.2023
	TempJJ	-0.5728	-0.6075	-0.9298
	RainAnn	0.1433	0.1478	0.0029
	RainJJ	-0.0696	-0.0735	0.0210
	Snow before	0.0451	0.0484	-0.0111
	Snow after	-0.0074	-0.0054	-0.0120
	Snow following year	-0.0485	-0.0537	-0.0371
	Precip	0.0302	0.0227	0.0762
	PackD	0.0136	0.0166	-0.0147

Table 5.1: Stochastic elasticities for parameters in survival function; e_S^{LM} is the stochastic elasticities of linear models, e_S^{PLM} is the stochastic elasticities of partial linear models, and e_S^{SIM} is the stochastic elasticities of single index models.

Vital Rate	Parameter	e_S^{LM}	e_S^{PLM}	e_S^{SIM}
Growth	TempAnn	-0.3239	-0.0972	0.0142
	TempAM1	0.1566	0.0416	0.1160
	TempAM2	0.1305	0.0416	-0.0533
	TempJJ	-0.3023	-0.0798	0.0691
	RainAnn	0.1282	0.0406	0.0023
	RainJJ	-0.0485	-0.0156	-0.0007
	SnowAnn	0.1864	0.0682	0.0512
	Snow before	0.0330	0.0155	-0.0212
	Snow after	0.0347	0.0116	-0.0109
	Snow following year	-0.0741	-0.0183	0.0098
	Precip	-0.1421	-0.0430	-0.0283
	PackAnn	-0.0115	-0.0034	0.0875
	PackD	-0.0553	-0.0172	0.0827

Table 5.2: Stochastic elasticities for parameters in growth function; e_S^{LM} is the stochastic elasticities of linear models, e_S^{PLM} is the stochastic elasticities of partial linear models, and e_S^{SIM} is the stochastic elasticities of single index models.

Vital Rate	Parameter	e_S^{LM}	e_S^{PLM}	e_S^{SIM}
Reproduction	TempAnn	0.1093	0.1134	0.2830
	TempAM2	0.0394	0.0437	0.0561
	RainAnn	0.0390	0.0445	−0.2620
	RainJJ	−0.0204	−0.0200	0.0811
	Snow following year	0.0128	0.0134	−0.0115
	Precip	0.0771	0.0759	0.1265
	PackAnn	0.0173	0.0175	0.1605
	PackD	−0.0064	−0.0057	0.5984
litter size	TempAnn	0.2039	0.1963	0.3793
	RainAnn	0.0479	0.0461	0.0876
	Snow after	−0.0135	−0.0128	−0.0249
	Precip	0.0262	0.0251	0.0478
	PackAnn	0.0367	0.0349	0.0645

Table 5.3: Stochastic elasticities for parameters in reproduction and litter size functions; e_S^{LM} is the stochastic elasticities of linear models, e_S^{PLM} is the stochastic elasticities of partial linear models, and e_S^{SIM} is the stochastic elasticities of single index models.

Vital Rate	Parameter	e_S^{LM}	e_S^{PLM}	e_S^{SIM}
Recruit Mass	TempAnn	−0.3215	−0.0929	0.0103
	TempAM1	0.1389	0.0378	0.0231
	TempAM2	0.0755	0.0272	0.0181
	TempJJ	−0.1332	−0.0357	0.0340
	RainAnn	0.1127	0.0305	0.0030
	RainJJ	−0.0311	−0.0076	0.0032
	SnowAnn	0.0604	0.0141	−0.0058
	Snow before	0.0130	0.0032	0.0039
	Snow after	0.0163	0.0053	−0.0020
	Snow following year	−0.0171	−0.0041	−0.0047
	Precip	−0.1328	−0.0316	0.0061
	PackAnn	−0.0133	−0.0034	0.0065
	PackD	−0.0170	−0.0040	0.0186

Table 5.4: Stochastic elasticities for parameters in recruit mass function; e_S^{LM} is the stochastic elasticities of linear models, e_S^{PLM} is the stochastic elasticities of partial linear models, and e_S^{SIM} is the stochastic elasticities of single index models.

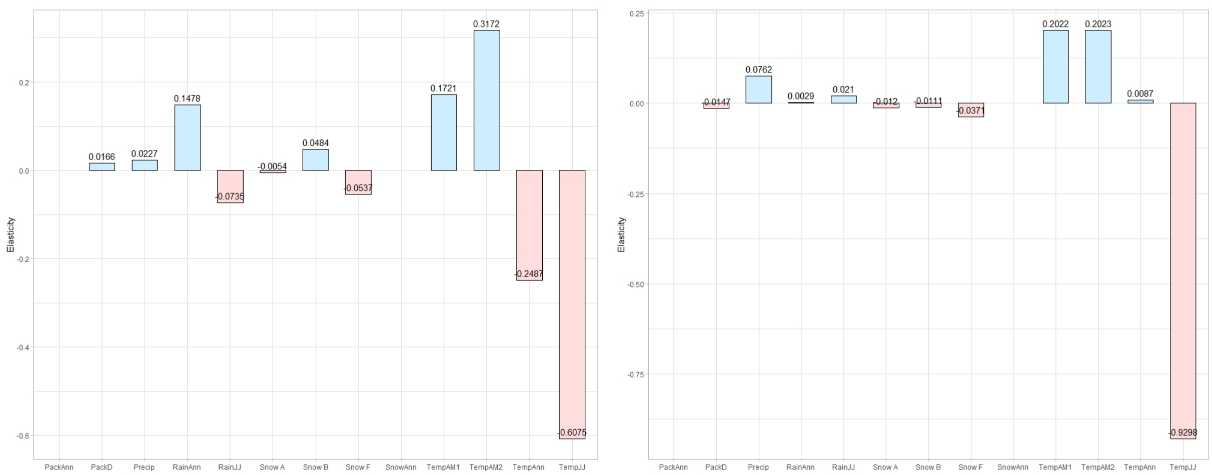


Figure 5.1: **Left**: Elasticities to parameters in survival partial linear models. **Right**: single index models.

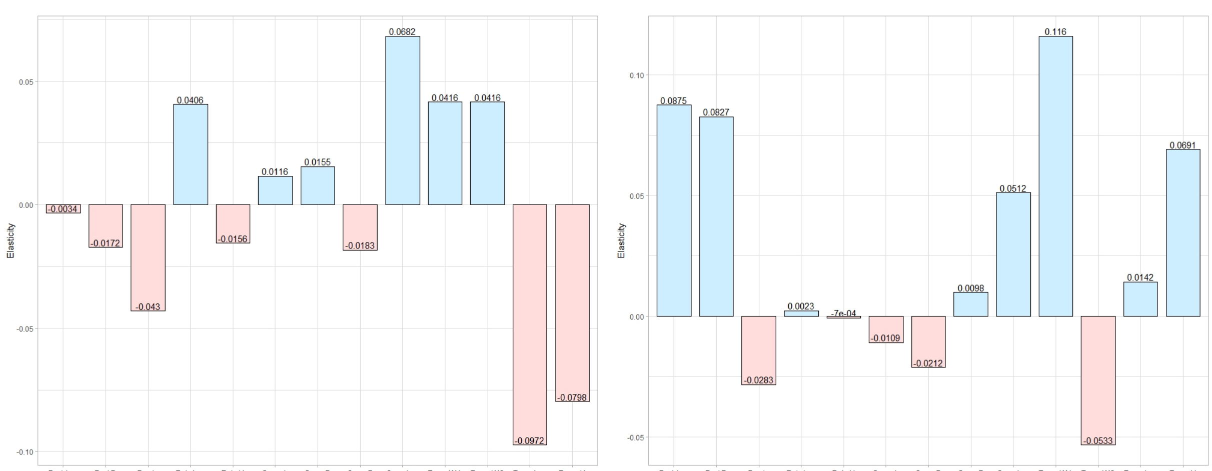


Figure 5.2: **Left**: elasticities to parameters in growth by partial linear models. **Right**: elasticities to parameters in growth by single index models.

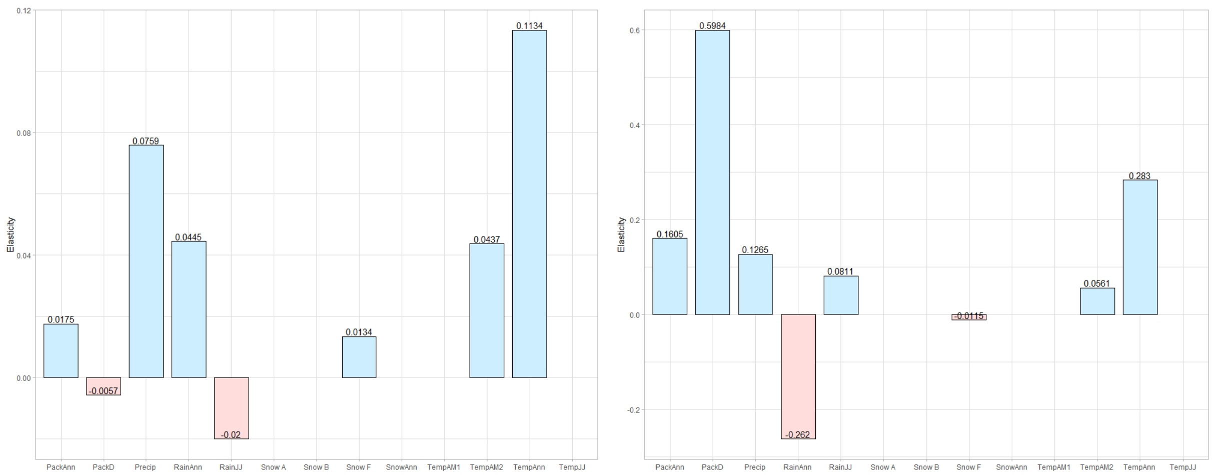


Figure 5.3: **Left**: elasticities to parameters in reproduction by partial linear models. **Right**: elasticities to parameters in reproduction by single index models.

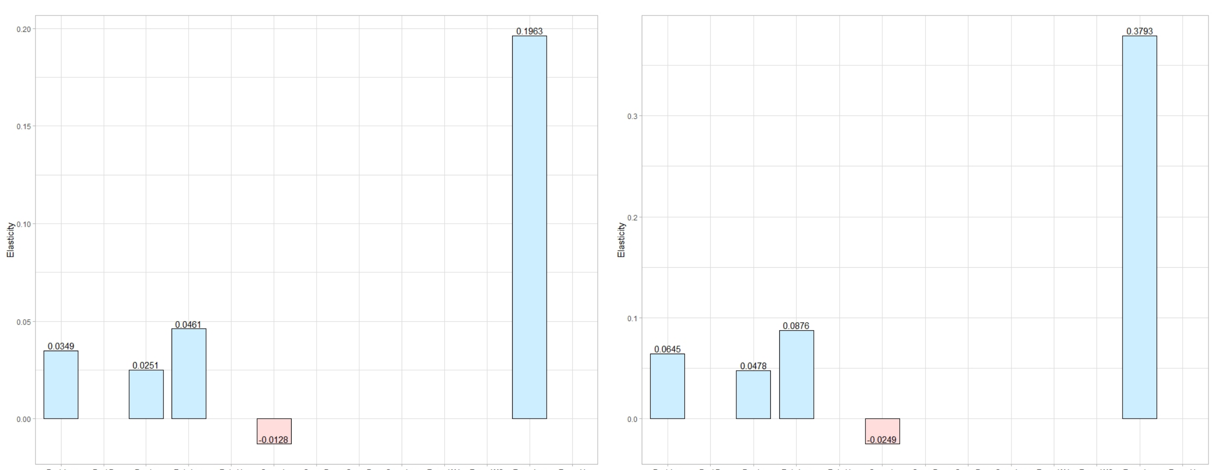


Figure 5.4: **Left**: elasticities to parameters in litter number by partial linear models. **Right**: elasticities to parameters in litter number by single index models.

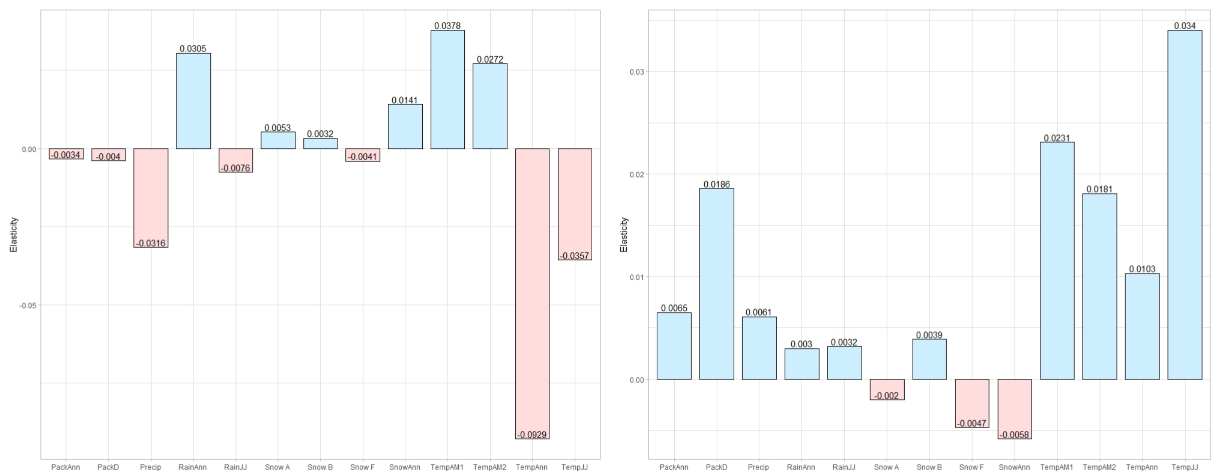


Figure 5.5: **Left**: elasticities to parameters in recruit mass by partial linear models. **Right**: elasticities to parameters in recruit mass by single index models.

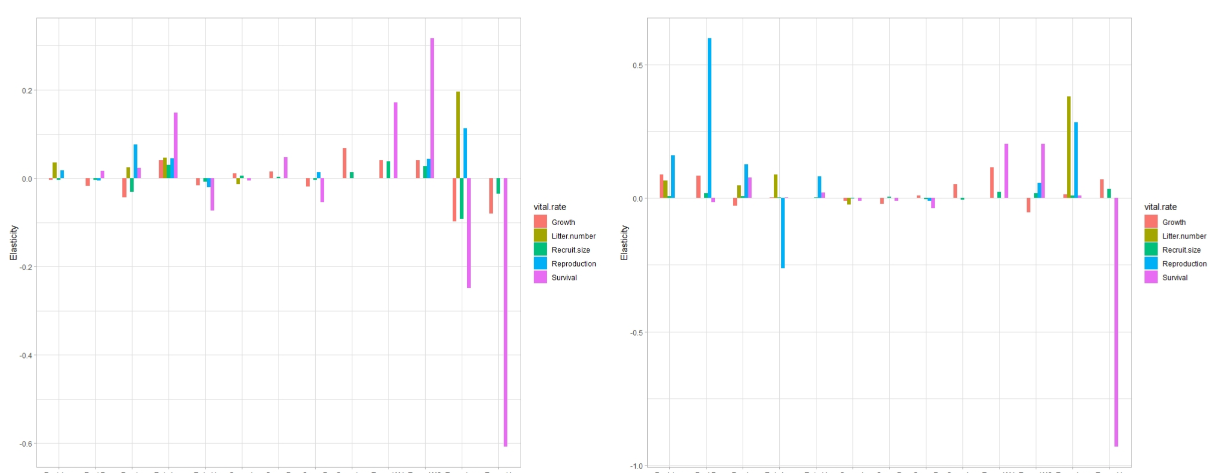


Figure 5.6: **Left**: elasticities to parameters by partial linear models. **Right**: elasticities to parameters by single index models.

Bibliography

[1] Aeberhard, W. H., Mills Flemming, J., and Nielsen, A. (2018), Review of state-space models for fisheries Science, *Annual Review of Statistics and Its Application*, 5:215-235.

[2] Bakker, V. J., Doak, D. F., Roemer, G. W., Garcelon, D. K., Coonan, T. J., Morrison, S. A., Lynch, C., Ralls, K., Shaw, R. (2009), Incorporating ecological drivers and uncertainty into a demographic population viability analysis for the island fox, *Ecological Monographs*, 79(1), 77–108.

[3] Caswell, H. (2001), Matrix population models: construction, analysis, and interpretation, Second edition, Sinauer.

[4] Coulson, T., Catchpole, E.A., Albon, S.D., Morgan, B.J., Pemberton, J.M., Clutton-Brock, T.H., Crawley, M.J., Grenfell, B.T. (2001), Age, sex, density, winter weather, and population crashes in Soay sheep, *Science*, 292(5521), 1528-1531.

[5] Coulson, T. (2012), Integral projections models, their construction and use in posing hypotheses in ecology, *Oikos*, 121, 1337-1350.

[6] Dalgleish, H.J., Koons, D. N., Hooten, M. B., Moffet, C.A. and Adler, P.B. (2011), Climate influences the demography of three dominant sagebrush steppe plants, *Ecology*, 92(1), 75-85.

[7] Dobson, F. S. (1988), The limits of phenotypic plasticity in life histories of Columbian ground squirrels. Pp. 193–210, in M. S. Boyce, ed. Evolution of life histories of mammals. Yale Univ. Press, New Haven, CT.

[8] Dobson, F. S. (1995), Regulation of population size: evidence from Columbian ground squirrels. *Oecologia*, 102, 44–51.

[9] Dobson, F. S., M. J. Badry, and C. Geddes. (1992), Seasonal activity in the Columbian ground squirrel. *Can. J. Zool.*, 70, 1364–1368.

[10] Dobson, F. S., and J. D. Kjelgaard. (1985a), The influence of food resources on population dynamics in Columbian ground squirrels. *Can. J. Zool.*, 63, 2095–2104.

[11] Dobson, F. S., and J. D. Kjelgaard. (1985b), The influence of food resources on life history in Columbian ground squirrels. *Can. J. Zool.*, 63, 2105–2109.

[12] Dobson, F. S., Lane, J. E., Low, M., and Murie, J. O. (2016), Fitness implications of seasonal climate variation in Columbian ground squirrels, *Ecology and Evolution*, 6(16), 5614-5622.

[13] Dobson, F. S., and J. O. Murie (1987), Interpretation of intraspecific life history patterns: evidence from Columbian ground squirrels. *Am. Nat.*, 129, 382–397.

[14] Dobson, F. S. and M. K. Oli (2001), The demographic basis of population regulation in Columbian ground squirrels *The American Naturalist*, 158 (3), 236-247

[15] Dobson, F.S., T. S. Risch, and J. O, Murie (1999), Increasing returns in the life history of Columbian ground squirrels, *Journal of Animal Ecology*, 73-86

[16] Easterling, M.R., Ellner, S.P., and Dixon, P.M. (2000), Size-specific sensitivity: applying a new structured population model, *Ecology*, 81, 694-708.

[17] Ellis, A.M. and Post, E. (2004), Population response to climate change: linear vs non-linear modeling approchaes, *BMC Ecol.*, 4:2.

[18] Ellner, S.T., Childs, D.Z. and Rees, M. (2016), Data-driven Modeling of Structured Populations, Springer

[19] Ellner, S. P. and Rees, M. (2007), Stochastic stable population growth in integral projection models: theory and application, *J. Math. Biol.*, 54, 227-256.

[20] Elton, C. (1927), Animal Ecology, Sidgwick and Jackson, London.

[21] Fortin, D, Beyer, H.L., Boyce, M.S., Smith, D.W., Duchesne, T., and Mao, J.S. (2005), Wolves influence elk movements: behavior shapes a tropic cascade in Yellowstone National Park, *Ecology*, 86, 1320-1330.

[22] Grewal, M. S. and Andrews, A. P. (2010). Applications of Kalman filtering in aerospace 1960 to the present. *IEEE Control Systems Magazine*, 30:69-78.

[23] Hunter, C.M., Caswell, H., Runge, M.C., Regehr, E.V., Amstrup, S.C., Stirling, I. (2010), Climate change threatens polar bear populations: a stochastic demographic analysis, *Ecology*, 91(10), 2883-2897.

[24] Intergovernmental Panel on Climate Change (2019), Special Report on climate change, desertification, land degradation, sustainable land management, food security, and greenhouse gas fluxes in terrestrial ecosystems.

[25] James, G., D. Witten, T. Hastie, and R. Tibshirani (2013), An Introduction to Statistical Learning with Applications in R, Springer.

[26] Kalman, R.E.(1960), A new approach to linear filtering and prediction problems, *Trans ASME J. Basic Eng.*, 82, 35-45.

[27] Kalman, R.E. and R.S. Bucy (1961), New results in filtering and prediction theory, *Trans ASME J. Basic Eng.*, 83, 95-108.

[28] Krebs, C. J. (1994), Ecology, Harper-Collins, New York.

[29] Lane, J. E., L. E. B. Kruuk, A. Charmantier, J. O. Murie, D. W. Coltman, S. Raveh, et al. (2011), Heritability of hibernation emergence date in a wild population of Columbian ground squirrels. *J. Evol. Biol.*, 24, 1949–1959.

[30] Lane, J. E., L. E. B. Kruuk, A. Charmantier, J. O. Murie, and F. S. Dobson. (2012), Delayed phenology and reduced fitness associated with climate change in a wild mammal, *Nature*, 489, 554–557.

[31] Maunder, M. N., Deriso R. B., Hanson C. H. (2015), Use of state-space population dynamics models in hypothesis testing:advantages over simple log-linear regressions for modeling survival,illustrated with application to longfin smelt, *Fisheries Research*, 164, 102-111.

[32] Mihoub, J. B., N. G. Jouawad, P. Pilard, F. Jiguet, M. Low, and C. Teplitsky. (2012), Impact of temperature on the breeding performance and selection patterns in lesser kestrels Falco naumanni. *J. Avian Biol.*, 43, 472–480.

[33] Newman, K. B., Buckland, S. T., Morgan, B. J. T., King, R., Borchers, D. L., Cole, D. J., Besbeas, P., Gimenez, O., and Thomas, L. (2014), Modelling population dynamics: model formulation, fitting and assessment using state-space methods, Springer, New York, NY.

[34] Ozgul, A., Tuljapurkar, S., Benton, T.G., Pemberton, J.M., Clutton-Brock, T.H., and Coulson, T. (2009), The dynamics of phenotypic change and the shrinking sheep of St. Kilda, *Science*, 325, 464-467.

[35] Parmesan, C. (2006), Ecological and evolutionay responses to recent climate change, *Annu. Rev. Ecol. Evol. Syst.*, 37, 637-669.

[36] Post. E. and Forchhammer, M.C. (2002), Synchronization of animal population dynamics by large-scale climate, *Nature*, 420, 168–171.

[37] Post. E., Stenseth, N.C., Langvatn R., and Fromentin, J.-M.(1997), Global climate change and phenotypic variation among red deer cohorts, *Proc. R. Soc. B*, 264, 1317-1324.

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

[39] Rees, M. and Ellner, S. P. (2009), Integral projection models for populations in temporally varying environments, *Ecological Monographs*, 79(4), 575-594.

[40] Rubach, K., Wu, M., Abebe A., Dobson, F. S., Murie, J. O., and Viblanc, V. A. (2016), Testing the reproductive and somatic trade-off in female Columbian ground squirrels, *Ecology and Evolution*, 6(21), 7586–7595.

[41] Saltz, D., Rubenstein, D.I., White, G.C. (2006), The impact of increased environmental stochasticity due to climate change on the dynamics of asiatic wild ass, *Conserv Biol.*, 20(5), 1402-1409.

[42] Schumaker, L. L. (1981), *Spline Functions: Basic Theory*, Wiley, New York.

[43] Shumway, R.H. and Stoffer, D.S. (2017), Time Series Analysis and Its Applications, 4th edition, Springer.

[44] Simmonds, E.G. and Coulson, T (2015), Analysis of phenotypic change in relation to climate drivers in a population of Soay sheep Ovis aries, *Oikos*, 124, 543-552.

[45] Tredennick, A.T., Teller, B.J., Adler, P.B., Hooker, G., and Ellner, S.P. (2018), Size-by-environment interactions: a neglected dimension of species' responses to environmental variation, preprint.

[46] Tuljapurkar, S. and Haridas, C.V.(2006), Temporal autocorrelation and stochastic population growth, *Ecology Letters*, 9, 324-334.

[47] Ward, E. J., Chirakkal, H., Gonzalez-Suarez, M., Aurioles-Gamboa, D., Holmes, E. E., and Gerber, L. (2010), Inferring spatial structure from time-series data using multivariate state-space models to detect metapopulation structure of California sea lions in the Gulf of California, Mexico, *Journal of Applied Ecology*, 47, 47-56.

[48] Wood, S. (2017), Generalized additive models: an introduction with R, Second edition, CRC Press.

[49] Zou, H., Hastie, T. (2005), Regularization and variable selection via the elastic net, *J. R. Statist. Soc. B.*, 67(2), 301-320.

Appendix A

R Codes

In this appendix we provide two example R codes: one for the state space model estimation and the second for the IPM using the generalized single-index model. We will leave the other models out, but all codes will be provided upon request.

A.1 R code for State Space Models

```

##############################################
#State Space Model with varying weather effects
##############################################

library(astsa)
agedata <- read.csv("age_weather.csv")
agedata2 <- as.data.frame(agedata)
y <- yy <- agedata2[,c(-1, -12:-17)]
y <- y[complete.cases(y),]

#num=27(years)
#p=10(age0-age914)

yy <- ts(y[,1:10], start = 1992)
weather <- ts(y[,11:23], start = 1992)
num <- nrow(yy)
p <- ncol(yy)
#num=26(years)
#p=10(age0-age914) and 13 weather variables

###
# State: N(t+1) = Phi*N(t) + Ups*weather(t) + w(t)
# Observation: Y(t) = A(t)N(t) + v(t)
# Parameters: Phi, Q=Var(w(t)), R=Var(v(t))
# A(t) = I


# make array of obs matrices
A <- array(0, dim=c(p,p,num))
```

```
for(k in 1:num){
  A[,,k] <- diag(1,p)
}


# Initial values
mu0 <- matrix(1, p, 1)
Phi <- diag(1, p)
cQ <- cR <-  Sigma0 <- 1*diag(1, p)

em <- EM1(num, yy, A, mu0, Sigma0, Phi, cQ, cR, 1000, .001)

Ups <- 0
Gam <- 0
input <- 0
cR <- chol(em$R)
cQ <- chol(em$Q)
mu0 <- em$mu0
Sigma0 <- em$Sigma0
ks <- Ksmooth1(num, yy, A, em$mu0, em$Sigma0, em$Phi, 0, 0,
               chol(em$Q), chol(em$R), 0)

init.transition <- rep(.5, 19)
init.weather <- rep(0, 26)
Phi <- matrix(0,p,p)

# Estimate SSM
# Repeat a few times for consistency
for(i in 1:1){
  llf.transition <- function(par.t, Ups){
    Phi[1,] <- cbind(0, par.t[1], par.t[2], par.t[3], par.t[4],
                     par.t[5], par.t[6], par.t[7], par.t[8], par.t[9])
    Phi[2,1] <- par.t[10]
    Phi[3,2] <- par.t[11]
    Phi[4,3] <- par.t[12]
    Phi[5,4] <- par.t[13]
    Phi[6,5] <- par.t[14]
    Phi[7,6] <- par.t[15]
    Phi[8,7] <- par.t[16]
    Phi[9,8] <- par.t[17]
    Phi[10,9] <- par.t[18]
    Phi[10,10] <- par.t[19]
    kf <- Kfilter1(num, yy, A, mu0, Sigma0, Phi, Ups, Gam = 0, cQ, cR,
   input=weather)
    return(kf$like)
  }



  est <- optim(init.transition, llf.transition, Ups=Ups, NULL,
               method='L-BFGS-B', hessian=TRUE,
```

```
                lower = rep(.05, 19),
                upper = c(rep(5,9), rep(.99,10)),
                control=list(trace=1, maxit = 10000))
round(cbind(estimate=est$par), 4)
par.t <- init.transition <- est$par


Phi <- matrix(0,p,p)
Phi[1,] <- cbind(0, par.t[1], par.t[2], par.t[3], par.t[4],
                 par.t[5], par.t[6], par.t[7], par.t[8], par.t[9])
Phi[2,1] <- par.t[10]
Phi[3,2] <- par.t[11]
Phi[4,3] <- par.t[12]
Phi[5,4] <- par.t[13]
Phi[6,5] <- par.t[14]
Phi[7,6] <- par.t[15]
Phi[8,7] <- par.t[16]
Phi[9,8] <- par.t[17]
Phi[10,9] <- par.t[18]
Phi[10,10] <- par.t[19]


# Assume climate effect on age 0 different from rest
# but otherwise same. We may need better screening of weather vars
llf.weather <- function(par.w, Phi){
  young <- c(par.w[1],par.w[2],par.w[3],par.w[4], par.w[5], par.w[6],par.w[7],
             par.w[8], par.w[9],par.w[10], par.w[11], par.w[12],par.w[13])
  others <- c(par.w[14],par.w[15],par.w[16],par.w[17], par.w[18], par.w[19],
              par.w[20], par.w[21], par.w[22],par.w[23], par.w[24], par.w[25],par.w[26])
  Ups <- matrix(c(young, rep(others, p-1)), nrow = p, byrow = T)
  kf <- Kfilter1(num, yy, A, mu0, Sigma0, Phi, Ups, Gam = 0, cQ, cR, input=weather)
  return(kf$like)
}

# Use optim to maximize likelihood
est <- optim(init.weather, llf.weather, Phi = Phi, NULL,
             method='L-BFGS-B',
             control=list(trace=1, maxit = 10000))
round(cbind(estimate=est$par), 4)
par.w <- init.weather <- est$par
young <- c(par.w[1],par.w[2],par.w[3],par.w[4], par.w[5], par.w[6],par.w[7],
           par.w[8], par.w[9],par.w[10], par.w[11], par.w[12],par.w[13])
others <- c(par.w[14],par.w[15],par.w[16],par.w[17], par.w[18], par.w[19],
```

```r
                 par.w[20], par.w[21], par.w[22],par.w[23], par.w[24], par.w[25],par.w[26])
  Ups <- matrix(c(young, rep(others, p-1)), nrow = p, byrow = T)
}


ks <- Ksmooth1(num, yy, A, em$mu0, em$Sigma0, Phi, Ups, 0,
                  chol(em$Q), chol(em$R), input=weather)
ys <- ps <- matrix(0, ncol = p, nrow = num)


for(i in 1:p){
  ys[,i] <- ks$xp[i,,]
  ps[,i] <- 2*sqrt(ks$Pp[i,i,])
}

T <- num
x <- matrix(0, nrow = p, ncol = T)
x.w <- matrix(0, nrow = p, ncol = T)
yyy <- as.matrix(yy)
x[,1] <- x.w[,1] <- yyy[1,]
for(i in 2:T){
  x.w[,i] <- Phi%*%cbind(x.w[,i-1]) + Ups%*%weather[i,]
  x[,i] <- Phi%*%cbind(x[,i-1])
}


year.pop <- rowSums(yy)
n.fem <- ts(year.pop, start = 1992)
mod.fem <- ts(colSums(x), start = 1992)
mod.fem.w <- ts(colSums(x.w), start = 1992)
em.fem <- ts(rowSums(ys), start = 1992)


all.ts <- ts.intersect(n.fem, mod.fem, mod.fem.w)
lambda.ts <- ts.intersect(n.fem/lag(n.fem,-1), mod.fem/lag(mod.fem,-1), mod.fem.w/lag(mod.fem.w,-1))

par(mfrow=c(1,2))
plot(all.ts[,1], lwd = 1, type = 'o',ylim = c(0,200), ylab = "Female
     Squirrel Population", xlab="Year")
lines(all.ts[,2], lwd = 1, col="blue")
lines(all.ts[,3], lwd = 1, col="red")
legend("topleft",legend=c("Annual Population", "SSM Projection", "SSM Projection (weather)" ),
       col = c("black", "blue", "red","green"),
       lty = 1,
       lwd = 2,
       #pch = c(NA,NA,NA),
       bty = 'n',
       pt.cex = 15, cex = 0.65, y.intersp=0.55)
```

```r

plot(lambda.ts[,1], lwd = 1, type = 'o',ylim = c(0,2), ylab = "Population
    Growth Rate", xlab="Year")
lines(lambda.ts[,2], lwd = 1, col="blue")
lines(lambda.ts[,3], lwd = 1, col="red")
legend("topleft",legend=c("Annual Population", "SSM Projection", "SSM
    Projection (weather)" ),
       col = c("black", "blue", "red"),
       lty = 1,
       lwd = 2,
       bty = 'n',
       pt.cex = 15, cex = 0.65, y.intersp=0.55)
```

```r

###############################################
#State Space Model with uniform weather effects
###############################################
library(astsa)
agedata <- read.csv("age_weather.csv")
agedata2 <- as.data.frame(agedata)
y <- yy <- agedata2[,c(-1, -12:-17)]
y <- y[complete.cases(y),]

#num=27(years)
#p=10(age0-age914)

yy <- ts(y[,1:10], start = 1992)
weather <- ts(y[,11:23], start = 1992)
num <- nrow(yy)
p <- ncol(yy)

###
# State: N(t+1) = Phi*N(t) + Ups*weather(t) + w(t)
# Observation: Y(t) = A(t)N(t) + v(t)
# Parameters: Phi, Q=Var(w(t)), R=Var(v(t))
# A(t) = I

# make array of obs matrices
A <- array(0, dim=c(p,p,num))
for(k in 1:num){
  A[,,k] <- diag(1,p)
}

# Initial values
mu0 <- matrix(1, p, 1)
Phi <- diag(1, p)
cQ <- cR <-  Sigma0 <- 1*diag(1, p)

em <- EM1(num, yy, A, mu0, Sigma0, Phi, cQ, cR, 1000, .001)

Ups <- 0
Gam <- 0
```

```r
input <- 0
cR <- chol(em$R)
cQ <- chol(em$Q)
mu0 <- em$mu0
Sigma0 <- em$Sigma0
ks <- Ksmooth1(num, yy, A, em$mu0, em$Sigma0, em$Phi, 0, 0,
               chol(em$Q), chol(em$R), 0)

init.transition <- rep(.5, 19)
init.weather <- rep(0, 13)
Phi <- matrix(0,p,p)

# Estimate SSM
# Repeat a few times for consistency
for(i in 1:3){
  llf.transition <- function(par.t, Ups){
    Phi[1,] <- cbind(0, par.t[1], par.t[2], par.t[3], par.t[4],
                     par.t[5], par.t[6], par.t[7], par.t[8], par.t[9])
    Phi[2,1] <- par.t[10]
    Phi[3,2] <- par.t[11]
    Phi[4,3] <- par.t[12]
    Phi[5,4] <- par.t[13]
    Phi[6,5] <- par.t[14]
    Phi[7,6] <- par.t[15]
    Phi[8,7] <- par.t[16]
    Phi[9,8] <- par.t[17]
    Phi[10,9] <- par.t[18]
    Phi[10,10] <- par.t[19]
    kf <- Kfilter1(num, yy, A, mu0, Sigma0, Phi, Ups, Gam = 0, cQ, cR,
   input=weather)
    return(kf$like)
  }

  est <- optim(init.transition, llf.transition, Ups=Ups, NULL,
               method='L-BFGS-B', hessian=TRUE,
               lower = rep(.05, 19),
               upper = c(rep(5,9), rep(.99,10)),
               control=list(trace=1, maxit = 10000))
  round(cbind(estimate=est$par), 4)
  par.t <- init.transition <- est$par

  Phi <- matrix(0,p,p)
  Phi[1,] <- cbind(0, par.t[1], par.t[2], par.t[3], par.t[4],
                   par.t[5], par.t[6], par.t[7], par.t[8], par.t[9])
  Phi[2,1] <- par.t[10]
  Phi[3,2] <- par.t[11]
  Phi[4,3] <- par.t[12]
  Phi[5,4] <- par.t[13]
  Phi[6,5] <- par.t[14]
  Phi[7,6] <- par.t[15]
  Phi[8,7] <- par.t[16]
```

```
  Phi[9,8] <- par.t[17]
  Phi[10,9] <- par.t[18]
  Phi[10,10] <- par.t[19]

  # Assume climate effect the same for all ages
  #We may need better screening of weather vars
  llf.weather <- function(par.w, Phi){
    young <- c(par.w[1],par.w[2],par.w[3],par.w[4], par.w[5], par.w[6],par.w[7],
                 par.w[8], par.w[9],par.w[10], par.w[11], par.w[12],par.w[13])
    Ups <- matrix(rep(young, p), nrow = p, byrow = T)
    kf <- Kfilter1(num, yy, A, mu0, Sigma0, Phi, Ups, Gam = 0, cQ, cR, input=weather)
    return(kf$like)
  }

  # Use optim to maximize likelihood
  est <- optim(init.weather, llf.weather, Phi = Phi, NULL,
               method='L-BFGS-B',
               control=list(trace=1, maxit = 10000))
  round(cbind(estimate=est$par), 4)
  par.w <- init.weather <- est$par
  young <- c(par.w[1],par.w[2],par.w[3],par.w[4], par.w[5], par.w[6],par.w[7],
             par.w[8], par.w[9],par.w[10], par.w[11], par.w[12],par.w[13])
  Ups <- matrix(rep(young, p), nrow = p, byrow = T)
}


ks <- Ksmooth1(num, yy, A, em$mu0, em$Sigma0, Phi, Ups, 0,
               chol(em$Q), chol(em$R), input=weather)
ys <- ps <- matrix(0, ncol = p, nrow = num)

for(i in 1:p){
  ys[,i] <- ks$xp[i,,]
  ps[,i] <- 2*sqrt(ks$Pp[i,i,])
}

T <- num
x <- matrix(0, nrow = p, ncol = T)
x.w <- matrix(0, nrow = p, ncol = T)
yyy <- as.matrix(yy)
x[,1] <- x.w[,1] <- yyy[1,]
for(i in 2:T){
  x.w[,i] <- Phi%*%cbind(x.w[,i-1]) + Ups%*%weather[i,]
  x[,i] <- Phi%*%cbind(x[,i-1])
}

year.pop <- rowSums(yy)
n.fem <- ts(year.pop, start = 1992)
```

```r
mod.fem <- ts(colSums(x), start = 1992)
mod.fem.w <- ts(colSums(x.w), start = 1992)
em.fem <- ts(rowSums(ys), start = 1992)


all.ts <- ts.intersect(n.fem, mod.fem, mod.fem.w)
lambda.ts <- ts.intersect(n.fem/lag(n.fem,-1), mod.fem/lag(mod.fem,-1),
                          mod.fem.w/lag(mod.fem.w,-1))
lambda.em <- log(em.fem/lag(em.fem,-1))
lambda.nm <- log(n.fem/lag(n.fem,-1))

par(mfrow=c(1,2))
plot(all.ts[,1], lwd = 1, type = 'o',ylim = c(0,200), ylab = "Female
     Squirrel Population", xlab="Year")
lines(all.ts[,2], lwd = 1, col="blue")
lines(all.ts[,3], lwd = 1, col="red")
legend("topleft",legend=c("Annual Population", "SSM Projection",
                          "SSM Projection (weather)"),
       col = c("black", "blue", "red"),
       lty = 1,
       lwd = 2,
       bty = 'n',
       pt.cex = 15, cex = 0.7, y.intersp=0.55)

plot(lambda.ts[,1], lwd = 1, type = 'o',ylim = c(0,3),
     ylab = "Population Growth Rate", xlab="Year" )
lines(lambda.ts[,2], lwd = 1, col="blue")
lines(lambda.ts[,3], lwd = 1, col="red")
legend("topleft",legend=c("Annual Population", "SSM Projection", "SSM
   Projection (weather)"),
       col = c("black", "blue", "red"),
       lty = 1,
       lwd = 2,
       #pch = c(NA,NA,NA),
       bty = 'n',
       pt.cex = 15, cex = 0.7, y.intersp=0.55)

########Plot Predictions
all.ts <- ts.intersect(n.fem, em.fem)
plot(all.ts[,1], type='p', ylim = c(0,200), ylab = "Female
     Squirrel Population", xlab="Year")
lines(all.ts[,2], lwd = 2, col="red")
lines(em.fem+2*sqrt(ps.fem), lty = 2, col=4)
lines(em.fem-2*sqrt(ps.fem), lty = 2, col=4)
```

A.2 R code for Stochastic Integral Projection Models by Single Index Models

```r
#####################################
## IPM WITH SINGLE INDEX MODEL
####################################

library(lme4)
library(nlme)
library(mgcv)
library(gratia)

set.seed(53241986)

workr2 <- read.csv("work6_2017.csv")
weather1 <- read.csv("weather1.csv")
source("Standard Graphical Pars.R");

workr2$masst1 <- log(workr2$masst1)
workr2$masst <- log(workr2$masst)
workr2$MEAN_JS_MASS <- log(workr2$MEAN_JS_MASS)

1.1*max(na.omit(workr2$z))
1.1*max(na.omit(workr2$z1))
0.9*min(na.omit(workr2$juv_mass))

# total litter numbers
litters <- na.omit(workr2$LS_WEAN)

# total survived offsprings
surv.juv <- na.omit(workr2$NUMBER_JUV_S)

# probability of successful recruitment
p_r <- sum(surv.juv)/sum(litters)

Yeart <- factor(workr2$YEAR)

######################
#fit survival models
######################

si.surv <- function(theta,dat,opt=TRUE) {
  ## Fit single index model using gam call, given theta (defines alpha).
  ## Return ML if opt==TRUE and fitted gam with theta added otherwise.
  ## Suitable for calling from 'optim' to find optimal theta/alpha.

  alpha.surv <- c(1,theta) ## constrained alpha defined using free theta
  kk <- sqrt(sum(alpha.surv^2))
  alpha.surv <- alpha.surv/kk  ## so now ||alpha||=1
  x <- cbind(dat$TempAnn, dat$TempAM1, dat$TempAM2, dat$TempJJ,
             dat$RainAnn, dat$RainJJ, dat$Snow_before, dat$Snow_after,
```

```r
              dat$snow_f_year, dat$Precip, dat$PackD)
  y.surv <- x%*%alpha.surv                ## argument of smooth
  b1 <- gam(dat$fy_surv ~ s(dat$masst, bs="cr") + s(y.surv, bs="cr"), family=binomial, method="ML")       ## fit model
  if (opt) return(b1$gcv.ubre) else {
    b1$alpha.surv <- alpha.surv  ## add alpha
    b1$J <- outer(alpha.surv,-theta/kk^2) ## compute Jacobian
    for (j in 1:length(theta)) b1$J[j+1,j] <- b1$J[j+1,j] + 1/kk   ## dalpha_i/dtheta_j
    return(b1)
  }
} ## si.surv

alpha.surv <- rep(NA, 13)
sse.si.surv <- rep(NA)
sdhat.si.surv <- rep(NA)

f1 <- optim(rep(1,10), si.surv, method="BFGS", hessian = TRUE, dat=workr2)
apsi.surv <- si.surv(f1$par, workr2, opt = FALSE)
alpha.surv <- apsi.surv$alpha.surv
sse.si.surv <- sum(resid(apsi.surv)^2)
sdhat.si.surv <- sqrt(sse.si.surv/df.residual(apsi.surv))

#####################
#fit growth models
#####################

si.growth <- function(theta,dat,opt=TRUE) {
  ## Fit single index model using gam call, given theta (defines alpha).
  ## Return ML if opt==TRUE and fitted gam with theta added otherwise.
  ## Suitable for calling from 'optim' to find optimal theta/alpha.

  alpha.growth <- c(1,theta) ## constrained alpha defined using free theta
  kk <- sqrt(sum(alpha.growth^2))
  alpha.growth <- alpha.growth/kk  ## so now ||alpha||=1
  x <- cbind(dat$TempAnn, dat$TempAM1, dat$TempAM2, dat$TempJJ, dat$RainAnn, dat$RainJJ,
              dat$SnowAnn, dat$Snow_before, dat$Snow_after, dat$snow_f_year,
              dat$Precip, dat$PackAnn, dat$PackD)
  y.growth <- x%*%alpha.growth                ## argument of smooth
  b1 <- gam(dat$masst1 ~ s(dat$masst, bs="cr")+s(y.growth, bs="cr"), method="ML")      ## fit model
  if (opt) return(b1$gcv.ubre) else {
    b1$alpha.growth <- alpha.growth  ## add alpha
    b1$J <- outer(alpha.growth,-theta/kk^2) ## compute Jacobian
    for (j in 1:length(theta)) b1$J[j+1,j] <- b1$J[j+1,j] + 1/kk   ## dalpha_i/dtheta_j
    return(b1)
  }
} ## si.growth
```

```

alpha.growth <- rep(NA, 13)
sse.si.growth <- rep(NA)
sdhat.si.growth <- rep(NA)

f1 <- optim(rep(1,12), si.growth, method="BFGS", hessian = TRUE, dat=
    workr2)
apsi.growth <- si.growth(f1$par, workr2, opt = FALSE)
alpha.growth <- apsi.growth$alpha.growth
sse.si.growth<- sum(resid(apsi.growth)^2)
sdhat.si.growth <- sqrt(sse.si.growth/df.residual(apsi.growth))

###############################
#fit reproduction models
###############################

si.reproduction <- function(theta,dat,opt=TRUE) {
  ## Fit single index model using gam call, given theta (defines alpha).
  ## Return ML if opt==TRUE and fitted gam with theta added otherwise.
  ## Suitable for calling from 'optim' to find optimal theta/alpha.

  alpha.reproduction <- c(1,theta) ## constrained alpha defined using free
     theta
  kk <- sqrt(sum(alpha.reproduction^2))
  alpha.reproduction <- alpha.reproduction/kk  ## so now ||alpha||=1
  x <- cbind(dat$TempAnn, dat$TempAM2, dat$RainAnn, dat$RainJJ,
             dat$snow_f_year, dat$Precip, dat$PackAnn, dat$PackD)
  y.reproduction <- x%*%alpha.reproduction                ## argument of
    smooth
  b1 <- gam(dat$had_lit ~ s(dat$masst, bs="cr") + s(y.reproduction, bs="cr
    "), family=binomial, method="ML")      ## fit model
  if (opt) return(b1$gcv.ubre) else {
    b1$alpha.reproduction <- alpha.reproduction  ## add alpha
    b1$J <- outer(alpha.reproduction,-theta/kk^2) ## compute Jacobian
    for (j in 1:length(theta)) b1$J[j+1,j] <- b1$J[j+1,j] + 1/kk   ##
    dalpha_i/dtheta_j
    return(b1)
  }
} ## si.reproduction

alpha.reproduction <- rep(NA, 13)
sse.si.reproduction <- rep(NA)
sdhat.si.reproduction <- rep(NA)

f1 <- optim(rep(1,7), si.reproduction, method="BFGS", hessian = TRUE, dat=
    workr2)
apsi.reproduction <- si.reproduction(f1$par, workr2, opt = FALSE)
alpha.reproduction <- apsi.reproduction$alpha.reproduction
sse.si.reproduction<- sum(resid(apsi.reproduction)^2)
sdhat.si.reproduction <- sqrt(sse.si.reproduction/df.residual(apsi.
    reproduction))
```

```r

################################
#fit recruit size models
################################

si.recruit.size <- function(theta,dat,opt=TRUE) {
  ## Fit single index model using gam call, given theta (defines alpha).
  ## Return ML if opt==TRUE and fitted gam with theta added otherwise.
  ## Suitable for calling from 'optim' to find optimal theta/alpha.

  alpha.recruit.size <- c(1,theta) ## constrained alpha defined using free theta
  kk <- sqrt(sum(alpha.recruit.size^2))
  alpha.recruit.size <- alpha.recruit.size/kk  ## so now ||alpha||=1
  x <- cbind(dat$TempAnn, dat$TempAM1, dat$TempAM2, dat$TempJJ, dat$RainAnn, dat$RainJJ,
             dat$SnowAnn, dat$Snow_before, dat$Snow_after, dat$snow_f_year,
             dat$Precip, dat$PackAnn, dat$PackD)
  y.recruit.size <- x%*%alpha.recruit.size                ## argument of smooth
  b1 <- gam(dat$MEAN_JS_MASS ~ s(dat$masst, bs="cr")+ s(y.recruit.size, bs="cr"), method="ML")      ## fit model
  if (opt) return(b1$gcv.ubre) else {
    b1$alpha.recruit.size <- alpha.recruit.size  ## add alpha
    b1$J <- outer(alpha.recruit.size,-theta/kk^2) ## compute Jacobian
    for (j in 1:length(theta)) b1$J[j+1,j] <- b1$J[j+1,j] + 1/kk   ## dalpha_i/dtheta_j
    return(b1)
  }
} ## si.recruit.size

alpha.recruit.size <- rep(NA, 13)
sse.si.recruit.size <- rep(NA)
sdhat.si.recruit.size <- rep(NA)

f1 <- optim(rep(1,12), si.recruit.size, method="BFGS", hessian = TRUE, dat=workr2)
apsi.recruit.size <- si.recruit.size(f1$par, workr2, opt = FALSE)
alpha.recruit.size <- apsi.recruit.size$alpha.recruit.size
sse.si.recruit.size<- sum(resid(apsi.recruit.size)^2)
sdhat.si.recruit.size <- sqrt(sse.si.recruit.size/df.residual(apsi.recruit.size))

########################################
#fit female number of offsprings models
########################################

si.litter <- function(theta,dat,opt=TRUE) {
  ## Fit single index model using gam call, given theta (defines alpha).
  ## Return ML if opt==TRUE and fitted gam with theta added otherwise.
```

```
  ## Suitable for calling from 'optim' to find optimal theta/alpha.

  alpha.litter <- c(1,theta) ## constrained alpha defined using free theta
  kk <- sqrt(sum(alpha.litter^2))
  alpha.litter <- alpha.litter/kk  ## so now ||alpha||=1
  x <- cbind(dat$TempAnn, dat$RainAnn, dat$Snow_after, dat$Precip, dat$PackAnn)
  y.litter <- x%*%alpha.litter                  ## argument of smooth
  b1 <- gam(dat$WEAN_F ~ s(dat$masst,bs="cr" )+ s(y.litter, bs="cr"), family=poisson, method="ML")      ## fit model
  if (opt) return(b1$gcv.ubre) else {
    b1$alpha.litter   <- alpha.litter    ## add alpha
    b1$J <- outer(alpha.litter  ,-theta/kk^2) ## compute Jacobian
    for (j in 1:length(theta)) b1$J[j+1,j] <- b1$J[j+1,j] + 1/kk   ## dalpha_i/dtheta_j
    return(b1)
  }
} ## si.litter

alpha.litter  <- rep(NA, 13)
sse.si.litter   <- rep(NA)
sdhat.si.litter   <- rep(NA)

f1 <- optim(rep(1,4), si.litter  , method="BFGS", hessian = TRUE, dat=workr2)
apsi.litter   <- si.litter  (f1$par, workr2, opt = FALSE)
alpha.litter   <- apsi.litter  $alpha.litter
sse.si.litter   <- sum(resid(apsi.litter  )^2)
sdhat.si.litter   <- sqrt(sse.si.litter/df.residual(apsi.litter  ))

############################################
#Define functions to calculate vital rates
############################################

alpha.surv.1 <- rep(NA, 13)
alpha.surv.1[1:11] <- alpha.surv
alpha.surv.1[12:13] <- 0

alpha.reproduction.1 <- rep(NA, 13)
alpha.reproduction.1[1:8] <- alpha.reproduction
alpha.reproduction.1[9:13] <- 0

alpha.litter.1 <- rep(NA, 13)
alpha.litter.1[1:5] <- alpha.litter
alpha.litter.1[6:13] <- 0

alpha<- cbind(alpha.surv.1, alpha.growth, alpha.reproduction.1, alpha.litter.1, alpha.recruit.size )

save(alpha, file="SIM alpha.Rdata")

```

```
y.surv <- alpha[1,1]*workr2$TempAnn + alpha[2,1]*workr2$TempAM1
+ alpha[3,1]*workr2$TempAM2 + alpha[4,1]*workr2$TempJJ + alpha[5,1]*workr2$RainAnn + alpha[6,1]*workr2$RainJJ
+ alpha[7,1]*workr2$Snow_before + alpha[8,1]*workr2$Snow_after + alpha[9,1]*workr2$snow_f_year
+ alpha[10,1]*workr2$Precip  +alpha[11,1]*workr2$PackD

y.growth <- alpha[1,2]*workr2$TempAnn + alpha[2,2]*workr2$TempAM1
+ alpha[3,2]*workr2$TempAM2 + alpha[4,2]*workr2$TempJJ + alpha[5,2]*workr2$RainAnn + alpha[6,2]*workr2$RainJJ
+ alpha[7,2]*workr2$SnowAnn + alpha[8,2]*workr2$Snow_before + alpha[9,2]*workr2$Snow_after + alpha[10,2]*workr2$snow_f_year
+ alpha[11,2]*workr2$Precip + alpha[12,2]*workr2$PackAnn +alpha[13,2]*workr2$PackD


y.reproduction <- alpha[1,3]*workr2$TempAnn
+ alpha[2,3]*workr2$TempAM2  + alpha[3,3]*workr2$RainAnn + alpha[4,3]*workr2$RainJJ
+ alpha[5,3]*workr2$snow_f_year
+ alpha[6,3]*workr2$Precip + alpha[7,3]*workr2$PackAnn +alpha[8,3]*workr2$PackD

y.litter <- alpha[1,4]*workr2$TempAnn
+ alpha[2,4]*workr2$RainAnn + alpha[3,4]*workr2$Snow_after
+ alpha[4,4]*workr2$Precip + alpha[5,4]*workr2$PackAnn

y.recruit.size <-  alpha[1,5]*workr2$TempAnn + alpha[2,5]*workr2$TempAM1
+ alpha[3,5]*workr2$TempAM2 + alpha[4,5]*workr2$TempJJ + alpha[5,5]*workr2$RainAnn + alpha[6,5]*workr2$RainJJ
+ alpha[7,5]*workr2$SnowAnn + alpha[8,5]*workr2$Snow_before + alpha[9,5]*workr2$Snow_after + alpha[10,5]*workr2$snow_f_year
+ alpha[11,5]*workr2$Precip + alpha[12,5]*workr2$PackAnn +alpha[13,5]*workr2$PackD


workr3 <- cbind(workr2, y.surv, y.growth, y.reproduction, y.litter, y.recruit.size  )

surv.gam <- gam(fy_surv ~ s(z, bs="cr")+ s(y.surv, bs="cr"), family=binomial, data=workr3)

growth.gam <- gam(z1 ~ s(z, bs="cr")+s(y.growth, bs="cr"), data=workr3)

sse.growth <- sum(resid(growth.gam)^2)
sdhat.growth <- sqrt(sse.growth/df.residual(growth.gam))

reproduction.gam <- gam(had_lit ~ s(z, bs="cr")+s(y.reproduction, bs="cr"), family=binomial, data=workr3)
```

```r
litter.gam <- gam(WEAN_F ~ s(z, bs="cr")+s(y.litter, bs="cr"), family=poisson, data=workr3)

recruit.size.gam <- gam(juv_mass ~ s(z, bs="cr")+s(y.recruit.size, bs="cr"), data=workr3)

sse.recruit.size <- sum(resid(recruit.size.gam)^2)
sdhat.recruit.size <- sqrt(sse.recruit.size/df.residual(recruit.size.gam))

########################
# survival probability
########################

s_z <- function(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                Snow_before, Snow_after, snow_f_year, Precip,  PackD, alpha){

  y.surv <- alpha[1,1]*TempAnn + alpha[2,1]*TempAM1
  + alpha[3,1]*TempAM2 + alpha[4,1]*TempJJ + alpha[5,1]*RainAnn + alpha[6,1]*RainJJ
  + alpha[7,1]*Snow_before + alpha[8,1]*Snow_after + alpha[9,1]*snow_f_year
  + alpha[10,1]*Precip  +alpha[11,1]*PackD

  survdata <- data.frame(z=z, y.surv = y.surv)
  mu.surv <- predict(surv.gam, newdata=survdata)

  u <- exp(mu.surv)

  return(u/(1+u))
}

# survival mean

mu_surv <- function(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                    Snow_before, Snow_after, snow_f_year, Precip,  PackD, alpha){

  y.surv <- alpha[1,1]*TempAnn + alpha[2,1]*TempAM1
  + alpha[3,1]*TempAM2 + alpha[4,1]*TempJJ + alpha[5,1]*RainAnn + alpha[6,1]*RainJJ
  + alpha[7,1]*Snow_before + alpha[8,1]*Snow_after + alpha[9,1]*snow_f_year
  + alpha[10,1]*Precip  +alpha[11,1]*PackD


  survdata <- data.frame(z=z, y.surv = y.surv)
  mu.surv <- predict(surv.gam, newdata=survdata)
  return(mu.surv)
}

```

```
# Derivative of survival mean

d_mu_surv <- function(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                          Snow_before, Snow_after, snow_f_year, Precip,  PackD, alpha){

  y.surv <- alpha[1,1]*TempAnn + alpha[2,1]*TempAM1
  + alpha[3,1]*TempAM2 + alpha[4,1]*TempJJ + alpha[5,1]*RainAnn + alpha[6,1]*RainJJ
  + alpha[7,1]*Snow_before + alpha[8,1]*Snow_after + alpha[9,1]*snow_f_year
  + alpha[10,1]*Precip  +alpha[11,1]*PackD

  survdata <- data.frame(z=0, y.surv = y.surv)
  d.mu.surv <- derivatives(surv.gam,newdata=survdata,n=100,eps=1e-07,type = "central", unconditional = FALSE)
  return(d.mu.surv$derivative)
}

#####################
#Growth function
#####################

g_z1z <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                      SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha){


  y.growth <- alpha[1,2]*TempAnn + alpha[2,2]*TempAM1
  + alpha[3,2]*TempAM2 + alpha[4,2]*TempJJ + alpha[5,2]*RainAnn + alpha[6,2]*RainJJ
  + alpha[7,2]*SnowAnn + alpha[8,2]*Snow_before + alpha[9,2]*Snow_after + alpha[10,2]*snow_f_year
  + alpha[11,2]*Precip + alpha[12,2]*PackAnn +alpha[13,2]*PackD


  growthdata <- data.frame(z=z, y.growth = y.growth)

  mu.growth <- predict(growth.gam, newdata=growthdata, type="response")

  return(dnorm(z1, mean=mu.growth, sd=sdhat.growth))

}

mu_growth <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                          SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha){

```

```

  y.growth <- alpha[1,2]*TempAnn + alpha[2,2]*TempAM1
  + alpha[3,2]*TempAM2 + alpha[4,2]*TempJJ + alpha[5,2]*RainAnn + alpha
    [6,2]*RainJJ
  + alpha[7,2]*SnowAnn + alpha[8,2]*Snow_before + alpha[9,2]*Snow_after +
    alpha[10,2]*snow_f_year
  + alpha[11,2]*Precip + alpha[12,2]*PackAnn +alpha[13,2]*PackD


  growthdata <- data.frame(z=z, y.growth = y.growth)

  mu.growth <- predict(growth.gam, newdata=growthdata, type="response")

  return(mu.growth)

}

d_mu_growth <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn,
     RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year,
    Precip, PackAnn, PackD, alpha){


  y.growth <- alpha[1,2]*TempAnn + alpha[2,2]*TempAM1
  + alpha[3,2]*TempAM2 + alpha[4,2]*TempJJ + alpha[5,2]*RainAnn + alpha
    [6,2]*RainJJ
  + alpha[7,2]*SnowAnn + alpha[8,2]*Snow_before + alpha[9,2]*Snow_after +
    alpha[10,2]*snow_f_year
  + alpha[11,2]*Precip + alpha[12,2]*PackAnn +alpha[13,2]*PackD


  growthdata <- data.frame(z=0, y.growth = y.growth)
  d.mu.growth <- derivatives(growth.gam, newdata=growthdata,n=100,
                             eps=1e-07,type = "central", unconditional =
    FALSE)
  return(d.mu.growth$derivative)

}

#########################################
# Reproduction and number of offsprings
#########################################

pb_z <- function(z, TempAnn, TempAM2, RainAnn, RainJJ, Snow_after,
                 snow_f_year, Precip, PackAnn, PackD, alpha){


  y.reproduction <- alpha[1,3]*TempAnn
  + alpha[2,3]*TempAM2  + alpha[3,3]*RainAnn + alpha[4,3]*RainJJ
  + alpha[5,3]*snow_f_year
  + alpha[6,3]*Precip + alpha[7,3]*PackAnn +alpha[8,3]*PackD
```

```
  y.litter <- alpha[1,4]*TempAnn
  + alpha[2,4]*RainAnn + alpha[3,4]*Snow_after
  + alpha[4,4]*Precip + alpha[5,4]*PackAnn


  reproductiondata <- data.frame(z=z, y.reproduction = y.reproduction)

  mu.reproduction <- predict(reproduction.gam, newdata=reproductiondata)

  litterdata <- data.frame(z=z, y.litter = y.litter)

  mu.litter <- predict(litter.gam, newdata=litterdata)

  u <- exp(mu.reproduction)

  litter.number <- exp(mu.litter)
  return(u/(1+u)*litter.number)
}

mu_reproduction <- function(z, TempAnn, TempAM2, RainAnn, RainJJ,
                            snow_f_year, Precip, PackAnn, PackD, alpha){


  y.reproduction <- alpha[1,3]*TempAnn
  + alpha[2,3]*TempAM2  + alpha[3,3]*RainAnn + alpha[4,3]*RainJJ
  + alpha[5,3]*snow_f_year
  + alpha[6,3]*Precip + alpha[7,3]*PackAnn +alpha[8,3]*PackD


  reproductiondata <- data.frame(z=z, y.reproduction = y.reproduction)

  mu.reproduction <- predict(reproduction.gam, newdata=reproductiondata)

  return(mu.reproduction)
}

d_mu_reproduction <- function(z, TempAnn, TempAM2, RainAnn, RainJJ,
                              snow_f_year, Precip, PackAnn, PackD, alpha){


  y.reproduction <- alpha[1,3]*TempAnn
  + alpha[2,3]*TempAM2  + alpha[3,3]*RainAnn + alpha[4,3]*RainJJ
  + alpha[5,3]*snow_f_year
  + alpha[6,3]*Precip + alpha[7,3]*PackAnn +alpha[8,3]*PackD


  reproductiondata <- data.frame(z=0, y.reproduction = y.reproduction)

  d.mu.reproduction <- derivatives(reproduction.gam, newdata=
   reproductiondata,n=100,
```

```
                                    eps=1e-07,type = "central",
    unconditional = FALSE)
  return(d.mu.reproduction$derivative)

}

mu_litter <- function(z, TempAnn, RainAnn, Snow_after, Precip, PackAnn,
    alpha){

  y.litter <- alpha[1,4]*TempAnn
  + alpha[2,4]*RainAnn + alpha[3,4]*Snow_after
  + alpha[4,4]*Precip + alpha[5,4]*PackAnn

  litterdata <- data.frame(z=z, y.litter = y.litter)

  mu.litter <- predict(litter.gam, newdata=litterdata)

  return(mu.litter)
}


d_mu_litter <- function(z, TempAnn, RainAnn, Snow_after, Precip, PackAnn,
    alpha){


  y.litter <- alpha[1,4]*TempAnn
  + alpha[2,4]*RainAnn + alpha[3,4]*Snow_after
  + alpha[4,4]*Precip + alpha[5,4]*PackAnn

  litterdata <- data.frame(z=0, y.litter = y.litter)

  d.mu.litter <- derivatives(litter.gam, newdata=litterdata,n=100,
                              eps=1e-07,type = "central", unconditional =
    FALSE)
  return(d.mu.litter$derivative)
}

#############################
## Recruit Size function
#############################

c_z1z <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn,
    RainJJ,
                  SnowAnn, Snow_before, Snow_after, snow_f_year, Precip,
    PackAnn,
                  PackD, alpha){


  y.recruit.size <- alpha[1,5]*TempAnn + alpha[2,5]*TempAM1
  + alpha[3,5]*TempAM2 + alpha[4,5]*TempJJ + alpha[5,5]*RainAnn + alpha
    [6,5]*RainJJ
```

```
  + alpha[7,5]*SnowAnn + alpha[8,5]*Snow_before + alpha[9,5]*Snow_after + alpha[10,5]*snow_f_year
  + alpha[11,5]*Precip + alpha[12,5]*PackAnn +alpha[13,5]*PackD


  recruitsizedata <- data.frame(z=z, y.recruit.size = y.recruit.size)

  mu.recruit.size <- predict(recruit.size.gam, newdata=recruitsizedata, type="response")

  return(dnorm(z1, mean=mu.recruit.size, sd=sdhat.recruit.size))

}

mu_recruit_size <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                              SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                              PackD, alpha){


  y.recruit.size <- alpha[1,5]*TempAnn + alpha[2,5]*TempAM1
  + alpha[3,5]*TempAM2 + alpha[4,5]*TempJJ + alpha[5,5]*RainAnn + alpha[6,5]*RainJJ
  + alpha[7,5]*SnowAnn + alpha[8,5]*Snow_before + alpha[9,5]*Snow_after + alpha[10,5]*snow_f_year
  + alpha[11,5]*Precip + alpha[12,5]*PackAnn +alpha[13,5]*PackD


  recruitsizedata <- data.frame(z=z, y.recruit.size = y.recruit.size)

  mu.recruit.size <- predict(recruit.size.gam, newdata=recruitsizedata, type="response")

  return(mu.recruit.size)

}

d_mu_recruit_size <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                                SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                                PackD, alpha){


  y.recruit.size <- alpha[1,5]*TempAnn + alpha[2,5]*TempAM1
  + alpha[3,5]*TempAM2 + alpha[4,5]*TempJJ + alpha[5,5]*RainAnn + alpha[6,5]*RainJJ
  + alpha[7,5]*SnowAnn + alpha[8,5]*Snow_before + alpha[9,5]*Snow_after + alpha[10,5]*snow_f_year
  + alpha[11,5]*Precip + alpha[12,5]*PackAnn +alpha[13,5]*PackD
```

```
  recruitsizedata <- data.frame(z=0, y.recruit.size = y.recruit.size)

  d.mu.recruit.size <- derivatives(recruit.size.gam, newdata=recruitsizedata,n=100,
                                   eps=1e-07,type = "central", unconditional = FALSE)
  return(d.mu.recruit.size$derivative)

}

#############
#### Kernel
#############

#Define the survival-growth kernel

surv.growth<- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                       SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha){
  return(s_z(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
             Snow_before, Snow_after, snow_f_year, Precip,  PackD, alpha)
         *g_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha))
}

#Define the reproduction kernel

fecundity<- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                     SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha) {
  return(pb_z(z, TempAnn, TempAM2, RainAnn, RainJJ, Snow_after,
              snow_f_year, Precip, PackAnn, PackD, alpha)
         *p_r*c_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                    SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha))
}

## Build the discretized kernel
mk_k  <- function(m, alpha, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                  SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                  PackD, L, U) {
# mesh width
  h <- (U - L)/m

```

```
  meshpts <- L + ((1:m) - 1/2) * h

  P <- h * (outer(meshpts, meshpts, surv.growth, TempAnn=TempAnn, TempAM1=TempAM1, TempAM2=TempAM2,
                  TempJJ=TempJJ, RainAnn=RainAnn, RainJJ=RainJJ, SnowAnn=SnowAnn, Snow_before=Snow_before,
                  Snow_after=Snow_after, snow_f_year = snow_f_year, Precip=Precip, PackAnn=PackAnn,
                  PackD=PackD, alpha=alpha))
  F <- h * (outer(meshpts, meshpts, fecundity, TempAnn=TempAnn, TempAM1=TempAM1, TempAM2=TempAM2,
                  TempJJ=TempJJ, RainAnn=RainAnn, RainJJ=RainJJ, SnowAnn=SnowAnn, Snow_before=Snow_before,
                  Snow_after=Snow_after, snow_f_year = snow_f_year, Precip=Precip, PackAnn=PackAnn,
                  PackD=PackD, alpha=alpha))
  K <- P + F
  return(list(K = K, h=h, meshpts = meshpts, P = P, F = F))
}

#####################################################

nBigMatrix <- 100
n.est <- 50000
n.runin <- 500
minsize <- 4.67
maxsize <-  7.05
n.years <- 26

############################
#Run stochastic IPM
############################


TempAnn.1 <- weather1$TempAnn
TempAnn <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  TempAnn[i,] <- TempAnn.1
}

TempAM1.1 <- weather1$TempAM1
TempAM1 <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  TempAM1[i,] <- TempAM1.1
}

TempAM2.1 <- weather1$TempAM2
TempAM2 <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  TempAM2[i,] <- TempAM2.1
}
```

```r

TempJJ.1 <- weather1$TempJJ
TempJJ <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  TempJJ[i,] <- TempJJ.1
}

RainAnn.1 <- weather1$RainAnn
RainAnn <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  RainAnn[i,] <- RainAnn.1
}

RainJJ.1 <- weather1$RainJJ
RainJJ <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  RainJJ[i,] <- RainJJ.1
}

SnowAnn.1 <- weather1$SnowAnn
SnowAnn <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  SnowAnn[i,] <- SnowAnn.1
}

Snow_before.1 <- weather1$Snow_before
Snow_before <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  Snow_before[i,] <- Snow_before.1
}

Snow_after.1 <- weather1$Snow_after
Snow_after <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  Snow_after[i,] <- Snow_after.1
}

snow_f_year.1 <- weather1$snow_f_year
snow_f_year <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  snow_f_year[i,] <- snow_f_year.1
}

Precip.1 <- weather1$Precip
Precip <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  Precip[i,] <- Precip.1
}

PackAnn.1 <- weather1$PackAnn
PackAnn <- matrix(NA, nrow=nBigMatrix, ncol=26)
```

```
for(i in 1:nBigMatrix){
  PackAnn[i,] <- PackAnn.1
}

PackD.1 <- weather1$PackD
PackD <- matrix(NA, nrow=nBigMatrix, ncol=26)
for(i in 1:nBigMatrix){
  PackD[i,] <- PackD.1
}


iterate_model<-function(alpha,n.years,n.est) {

  #Construct the yearly kernels

  K.year.i <- array(NA,c( n.years,nBigMatrix,nBigMatrix))

  for(i in 1:n.years){

    year.K<- mk_k(nBigMatrix, alpha,  TempAnn[,i], TempAM1[,i], TempAM2[,i
   ],
                  TempJJ[,i], RainAnn[,i], RainJJ[,i],
                  SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                  snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                  minsize, maxsize)
    K.year.i[i,,] <- year.K$K
  }

  h <- year.K$h;


  #Calculate mean kernel, v and w

  mean.kernel <- apply(K.year.i,2:3,mean)

  w <- Re(eigen(mean.kernel)$vectors[,1]);
  v <- Re(eigen(t(mean.kernel))$vectors[,1]);

  # scale eigenvectors <v,w>=1
  w <- abs(w)/sum(h*abs(w))
  v <- abs(v)
  v <- v/(h*sum(v*w))
  cat(h*sum(v*w)," should = 1","\n")

  v.Ktw <- rep(NA,n.years)

  for(i in 1:n.years) {
    v.Ktw[i] <- sum(v*(K.year.i[i,,] %*% w))*h
  }

  #initialize variables
```

```
  nt<-rep(1/nBigMatrix,nBigMatrix)
  rt.V <- rt.N <- rep(NA,n.est)

  #Iterate model

  for (year.t in 1:n.est){
    if(year.t%%10000==0) cat("iterate: ", year.t,"\n");

    #Select year at random

    year.i <- sample(1:n.years,1)

    #iterate model with year-specific kernel
    nt1<-K.year.i[year.i,,] %*% nt

    sum.nt1<-sum(nt1)

    #Calculate log growth rates

    rt.V[year.t] <- log(sum(nt1*v)/sum(nt*v))
    rt.N[year.t] <- log(sum(nt1)/sum(nt))

    nt <- nt1 / sum.nt1
  }
  return(list(rt.N=rt.N,rt.V=rt.V,meshpts=year.K$meshpts,
              mean.kernel=mean.kernel,v.Ktw=v.Ktw))
  }

iter <- iterate_model(alpha,  n.years ,n.est)

rt.N <- iter$rt.N;
rt.V <- iter$rt.V;

Ls.Nt <- mean(rt.N)
SE.Ls.Nt <- sqrt(var(rt.N)/length(rt.N))
acf(rt.N,plot=FALSE)$acf[2:5];

Ls.Vt <- mean(rt.V)
SE.Ls.Vt <- sqrt(var(rt.V)/length(rt.V))
Vt <- sqrt(var(rt.V)/length(rt.V))
acf(rt.V,plot=FALSE)$acf[2:5];

cat("Log Lambda S using Nt ",Ls.Nt," 95% c.i ",Ls.Nt+2*SE.Ls.Nt," ",Ls.Nt
   -2*SE.Ls.Nt,"\n")
cat("Log Lambda S using Vt ",Ls.Vt," 95% c.i ",Ls.Vt+2*SE.Ls.Vt," ",Ls.Vt
   -2*SE.Ls.Vt,"\n")

lam.1 <- Re(eigen(iter$mean.kernel)$values[1])

var.v.Ktw <- var(iter$v.Ktw)
```

```r

approx.Ls <- log(lam.1) - var.v.Ktw/(2*lam.1*lam.1)

cat("Stochastic Log Lambda = ",mean(rt.N),"    approx=",approx.Ls,"\n")
```

```r
##########################################################
## Parameter Perturbations By Single Index Model
###########################################################

library(mgcv)
library(gratia)

set.seed(234)

load("SIM alpha.Rdata")
source("SIM.R")

######################################################################
#Stochastic perturbation analysis
######################################################################

params <- alpha

nBigMatrix <- 100
n.est <- 50000
n.runin <- 500
minsize <- 4.67
maxsize <-  7.05
n.years <- 26

stoc_pert_analysis<-function(params,n.est,n.runin,C.t,C.t.mean){

  year.i <- sample(1:n.years,n.est+1,replace=TRUE)

  K.year.i <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

  for(i in 1:n.years){

    year.K<- mk_k(nBigMatrix, alpha,  TempAnn[,i], TempAM1[,i], TempAM2[,i
    ],
                  TempJJ[,i], RainAnn[,i], RainJJ[,i],
                  SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                  snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                  minsize, maxsize)
    K.year.i[i,,] <- year.K$K

  }

  h <- year.K$h;
  meshpts <- year.K$meshpts

  #Calculate mean kernel, v and w
```

```r

mean.kernel <- apply(K.year.i,2:3,mean)

w <- Re(eigen(mean.kernel)$vectors[,1]);
v <- Re(eigen(t(mean.kernel))$vectors[,1]);

# scale eigenvectors <v,w>=1
w <- abs(w)/sum(h*abs(w))
v <- abs(v)
v <- v/(h*sum(v*w))
cat(h*sum(v*w)," should = 1","\n")

#Esimate Lambda s
#initialize variables

nt<-rep(1/nBigMatrix,nBigMatrix)
rt.V <- rt.N <- rep(NA,n.est)

#Iterate model

for (year.t in 1:n.est){
  if(year.t%%10000==0) cat("iterate: ", year.t,"\n");


  #iterate model with year-specific kernel
  nt1<-K.year.i[year.i[year.t],,] %*% nt

  sum.nt1<-sum(nt1)

  #Calculate log growth rates

  rt.V[year.t] <- log(sum(nt1*v)/sum(nt*v))
  rt.N[year.t] <- log(sum(nt1)/sum(nt))
  nt <- nt1 / sum.nt1

}

Ls <- exp(mean(rt.V))


### Get wt time series ###
wt<-matrix(1/nBigMatrix, nrow=n.est+1, ncol=nBigMatrix);

for (i in 1:n.est) {

  K              <- K.year.i[year.i[i],,]
  wt[i+1,]   <-K %*% wt[i,]
  wt[i+1,]   <-wt[i+1,]/sum(wt[i+1,]);
  if(i%%10000==0) cat("wt ",i,"\n")

}
```

```r


  ### Get vt time series ###
  vt<-matrix(1/nBigMatrix, nrow=n.est+1, ncol=nBigMatrix);
  for (i in (n.est+1):2) {

    K            <- K.year.i[year.i[i],,]
    vt[i-1,]   <- vt[i,] %*% K
    vt[i-1,]   <- vt[i-1,]/sum(vt[i-1,]);
    if(i%%10000==0) cat("vt  ",i,"\n")

  }

  elas.s <- matrix(0,nBigMatrix,nBigMatrix)
  elas.s.mean <- matrix(0,nBigMatrix,nBigMatrix)

  for (year.t in n.runin:(n.est-n.runin)) {

    #standard calculations needed for the various formulae

    vt1.wt  <- outer(vt[year.t+1,],wt[year.t,],FUN="*")
    vt1.C.wt <- vt1.wt * C.t[year.i[year.t],,]

    vt1.C.wt.mean         <- vt1.wt * C.t.mean[year.i[year.t],,]

    K            <- K.year.i[year.i[year.t],,]

    vt1.K.wt     <- sum(vt[year.t+1,] * (K %*% wt[year.t,]))

    #calculation of the standard elasticities

    elas.s             <-elas.s + (vt1.C.wt) / vt1.K.wt;
    elas.s.mean <-elas.s.mean + (vt1.C.wt.mean) / vt1.K.wt;

  }

  elas.s               <- elas.s/(n.est-2*n.runin+1)
  elas.s.mean  <- elas.s.mean/(n.est-2*n.runin+1)


  return(list(meshpts=year.K$meshpts, h=h, elas.s=elas.s, elas.s.mean=elas
    .s.mean,
              mean.kernel=mean.kernel, Ls=Ls))

}

########################################################################
#Let's do the survival TempAnn slope. Other variables are similar.
########################################################################

set.seed(53241986)
```

```
#Select the parameters to use
params.to.use <- alpha


#First calculate the mean of perturbation kernels

beta_s.2.mean <- mean(params.to.use[1,1])

Ct_z1z <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                   SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha){

  growth <- g_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                  SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha)
  survival <- s_z(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                  Snow_before, Snow_after, snow_f_year, Precip,  PackD, alpha)
  surv.mean <- mu_surv(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                       Snow_before, Snow_after, snow_f_year, Precip, PackD, alpha)
  surv.d <-  d_mu_surv(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                       Snow_before, Snow_after, snow_f_year, Precip, PackD, alpha)

  return(growth * survival *TempAnn /(1+exp(surv.mean)) * surv.d * alpha[1,1])
}

C.pert <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

for(i in 1:n.years){
  year.C <- h * (outer(meshpts, meshpts, Ct_z1z,  TempAnn[,i], TempAM1[,i], TempAM2[,i],
                       TempJJ[,i], RainAnn[,i], RainJJ[,i],
                       SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                       snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                       alpha = params.to.use))
  C.pert[i,,] <- year.C
}

Ct_z1z_mean <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha){
```

```
  growth <- g_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                  SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha)
  survival <- s_z(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                  Snow_before, Snow_after, snow_f_year, Precip,  PackD, alpha)
  surv.mean <- mu_surv(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                       Snow_before, Snow_after, snow_f_year, Precip, PackD, alpha)
  surv.d <-  d_mu_surv(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                       Snow_before, Snow_after, snow_f_year, Precip, PackD, alpha)

  return(growth * survival *TempAnn /(1+exp(surv.mean)) * surv.d *  beta_s.2.mean)
}

C.pert.mean <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

for(i in 1:n.years){
  year.C <- h * (outer(meshpts, meshpts, Ct_z1z_mean,TempAnn[,i], TempAM1[,i], TempAM2[,i],
                       TempJJ[,i], RainAnn[,i], RainJJ[,i],
                       SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                       snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                       alpha = params.to.use))
  C.pert.mean[i,,] <- year.C
}

pert.K <- stoc_pert_analysis(params.to.use, n.est, n.runin, C.pert, C.pert.mean)

elas.s           <- sum(pert.K$elas.s)
elas.s.mean <- sum(pert.K$elas.s.mean)
elas.s.sd      <- elas.s-elas.s.mean
cat("Stochastic elasticity ",elas.s,"\n")
cat("Stochastic elasticity mean ",elas.s.mean,"\n")
cat("Stochastic elasticity sd ",elas.s.sd,"\n")
cat("Stochastic sensitivity  mean ",pert.K$Ls*elas.s.mean/beta_s.2.mean,"\n")



################################
#GROWTH Parameter Perturbation
################################
```

```r

##########################################################################
#Growth function: TempAnn. Other variables are similar.
##########################################################################


set.seed(53241986)

#Select the parameters to use
params.to.use <- alpha


#First calculate the mean of perturbation kernels

beta.g.2.mean <- mean(params.to.use[1,2])

Ct_z1z <- function(z1,z,TempAnn, TempAM1, TempAM2,TempJJ, RainAnn, RainJJ,
                   SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha){

  growth <- g_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                  SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha)
  survival <- s_z(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                  Snow_before, Snow_after, snow_f_year, Precip,  PackD, alpha)
  growth.mean <- mu_growth(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                           SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha)
  growth.d <- d_mu_growth(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                          SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha)


  return( survival*growth*TempAnn* (z1-growth.mean)/(sdhat.growth^2 )* growth.d*alpha[1,2])
}

C.pert <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

for(i in 1:n.years){
  year.C <- h * (outer(meshpts, meshpts, Ct_z1z,  TempAnn[,i], TempAM1[,i], TempAM2[,i],
                       TempJJ[,i], RainAnn[,i], RainJJ[,i],
                       SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                       snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                       alpha = params.to.use))
```

```
  C.pert[i,,] <- year.C
}


Ct_z1z_mean <- function(z1,z,TempAnn, TempAM1, TempAM2,TempJJ, RainAnn, RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha){

  growth <- g_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                  SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha)
  survival <- s_z(z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                  Snow_before, Snow_after, snow_f_year, Precip,  PackD, alpha)
  growth.mean <- mu_growth(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                           SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha)
  growth.d <- d_mu_growth(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                          SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha)


  return( survival*growth*TempAnn* (z1-growth.mean)/(sdhat.growth^2 )* growth.d *beta.g.2.mean)
}

C.pert.mean <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

for(i in 1:n.years){
  year.C <- h * (outer(meshpts, meshpts, Ct_z1z_mean,  TempAnn[,i], TempAM1[,i], TempAM2[,i],
                       TempJJ[,i], RainAnn[,i], RainJJ[,i],
                       SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                       snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                       alpha = params.to.use))
  C.pert.mean[i,,] <- year.C
}


pert.K <- stoc_pert_analysis(params.to.use, n.est, n.runin, C.pert, C.pert.mean)

elas.s            <- sum(pert.K$elas.s)
elas.s.mean <- sum(pert.K$elas.s.mean)
elas.s.sd       <- elas.s-elas.s.mean
cat("Stochastic elasticity ",elas.s,"\n")
```

```
cat("Stochastic elasticity mean ",elas.s.mean,"\n")
cat("Stochastic elasticity sd ",elas.s.sd,"\n")
cat("Stochastic sensitivity  mean ",pert.K$Ls*elas.s.mean/beta.g.2.mean,"\n")



####################################
##Reproduction
####################################

#######################################################################
#Reproduction function: TempAnn. Other variables are similar.
#######################################################################

set.seed(53241986)

#Select the parameters to use
params.to.use <- alpha

#First calculate the mean of perturbation kernels

beta_rp.2.mean <- mean(params.to.use[1,3])

Ct_z1z <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                   SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                   PackD, alpha){

  recruit.size <- c_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                        PackD, alpha)
  reproduction <- pb_z(z, TempAnn, TempAM2, RainAnn, RainJJ, Snow_after,
                       snow_f_year, Precip, PackAnn, PackD, alpha)
  reproduction.mean <- mu_reproduction(z, TempAnn, TempAM2, RainAnn, RainJJ,
                                       snow_f_year, Precip, PackAnn, PackD, alpha)
  reproduction.d <-  d_mu_reproduction(z, TempAnn, TempAM2, RainAnn, RainJJ,
                                       snow_f_year, Precip, PackAnn, PackD, alpha)


  return(p_r*recruit.size * reproduction *TempAnn /(1+exp(reproduction.mean)) * reproduction.d * alpha[1,3])
}

```

```
C.pert <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

for(i in 1:n.years){
  year.C <- h * (outer(meshpts, meshpts, Ct_z1z,  TempAnn[,i], TempAM1[,i], TempAM2[,i],
                       TempJJ[,i], RainAnn[,i], RainJJ[,i],
                       SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                       snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                       alpha = params.to.use))
  C.pert[i,,] <- year.C
}

Ct_z1z_mean <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha){

  recruit.size <- c_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                        PackD, alpha)
  reproduction <- pb_z(z, TempAnn, TempAM2, RainAnn, RainJJ, Snow_after,
                       snow_f_year, Precip, PackAnn, PackD, alpha)
  reproduction.mean <- mu_reproduction(z, TempAnn, TempAM2, RainAnn, RainJJ,
                                       snow_f_year, Precip, PackAnn, PackD, alpha)
  reproduction.d <-  d_mu_reproduction(z, TempAnn, TempAM2, RainAnn, RainJJ,
                                       snow_f_year, Precip, PackAnn, PackD, alpha)
  return(p_r*recruit.size * reproduction *TempAnn /(1+exp(reproduction.mean)) * reproduction.d
         *beta_rp.2.mean)
}

C.pert.mean <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

for(i in 1:n.years){
  year.C <- h * (outer(meshpts, meshpts, Ct_z1z_mean,TempAnn[,i], TempAM1[,i], TempAM2[,i],
                       TempJJ[,i], RainAnn[,i], RainJJ[,i],
                       SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                       snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                       alpha = params.to.use))
  C.pert.mean[i,,] <- year.C
}

```

```r
pert.K <- stoc_pert_analysis(params.to.use, n.est, n.runin, C.pert, C.pert.mean)

elas.s           <- sum(pert.K$elas.s)
elas.s.mean <- sum(pert.K$elas.s.mean)
elas.s.sd       <- elas.s-elas.s.mean
cat("Stochastic elasticity ",elas.s,"\n")
cat("Stochastic elasticity mean ",elas.s.mean,"\n")
cat("Stochastic elasticity sd ",elas.s.sd,"\n")
cat("Stochastic sensitivity  mean ",pert.K$Ls*elas.s.mean/beta_rp.2.mean,"\n")



#####################
####### Litter Number
#####################

#########################################################################
#Litter Number: TempAnn. Other variables are similar.
#########################################################################


set.seed(53241986)

#Select the parameters to use
params.to.use <- alpha

#First calculate the mean of perturbation kernels

beta_litter.2.mean <- mean(params.to.use[1,4])

Ct_z1z <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                   SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                   PackD, alpha){

  recruit.size <- c_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                        PackD, alpha)
  reproduction <- pb_z(z, TempAnn, TempAM2, RainAnn, RainJJ, Snow_after,
                       snow_f_year, Precip, PackAnn, PackD, alpha)
  litter.d <-  d_mu_litter(z, TempAnn, RainAnn, Snow_after, Precip, PackAnn, alpha)


  return(p_r*recruit.size * reproduction *litter.d *TempAnn * alpha[1,4])
}
```

```r

C.pert <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

for(i in 1:n.years){
  year.C <- h * (outer(meshpts, meshpts, Ct_z1z,  TempAnn[,i], TempAM1[,i], TempAM2[,i],
                       TempJJ[,i], RainAnn[,i], RainJJ[,i],
                       SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                       snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                       alpha = params.to.use))
  C.pert[i,,] <- year.C
}

Ct_z1z_mean <- function(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha){

  recruit.size <- c_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                        PackD, alpha)
  reproduction <- pb_z(z, TempAnn, TempAM2, RainAnn, RainJJ, Snow_after,
                       snow_f_year, Precip, PackAnn, PackD, alpha)
  litter.d <-  d_mu_litter(z, TempAnn, RainAnn, Snow_after, Precip, PackAnn, alpha)


  return(p_r*recruit.size * reproduction * litter.d *TempAnn * beta_litter.2.mean)
}

C.pert.mean <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

for(i in 1:n.years){
  year.C <- h * (outer(meshpts, meshpts, Ct_z1z_mean,TempAnn[,i], TempAM1[,i], TempAM2[,i],
                       TempJJ[,i], RainAnn[,i], RainJJ[,i],
                       SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                       snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                       alpha = params.to.use))
  C.pert.mean[i,,] <- year.C
}

pert.K <- stoc_pert_analysis(params.to.use, n.est, n.runin, C.pert, C.pert.mean)

elas.s             <- sum(pert.K$elas.s)
```

```
elas.s.mean <- sum(pert.K$elas.s.mean)
elas.s.sd       <- elas.s-elas.s.mean
cat("Stochastic elasticity ",elas.s,"\n")
cat("Stochastic elasticity mean ",elas.s.mean,"\n")
cat("Stochastic elasticity sd ",elas.s.sd,"\n")
cat("Stochastic sensitivity  mean ",pert.K$Ls*elas.s.mean/beta_litter.2.
    mean,"\n")



####################
#Recruit Size
####################

#########################################################################
#Recruit Size: TempAnn. Other variables are similar.
#########################################################################


set.seed(53241986)

#Select the parameters to use
params.to.use <- alpha


#First calculate the mean of perturbation kernels

beta.rc.2.mean <- mean(params.to.use[1,5])

Ct_z1z <- function(z1,z,TempAnn, TempAM1, TempAM2,TempJJ, RainAnn, RainJJ,
                   SnowAnn, Snow_before, Snow_after, snow_f_year, Precip,
    PackAnn, PackD, alpha){

  reproduction <- pb_z(z, TempAnn, TempAM2, RainAnn, RainJJ, Snow_after,
                       snow_f_year, Precip, PackAnn, PackD, alpha)

  recruit.size <- c_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn,
     RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year,
    Precip, PackAnn,
                        PackD, alpha)
  recruit.size.mean <- mu_recruit_size(z1, z, TempAnn, TempAM1, TempAM2,
    TempJJ, RainAnn, RainJJ,
                                       SnowAnn, Snow_before, Snow_after,
    snow_f_year, Precip, PackAnn,
                                       PackD, alpha)
  recruit.size.d <- d_mu_recruit_size(z1, z, TempAnn, TempAM1, TempAM2,
    TempJJ, RainAnn, RainJJ,
                                      SnowAnn, Snow_before, Snow_after,
    snow_f_year, Precip, PackAnn,
                                      PackD, alpha)
```

```
  return( p_r*reproduction* recruit.size * (z1-recruit.size.mean )/(sdhat.recruit.size^2 )
          *recruit.size.d*TempAnn*alpha[1,5])
}



C.pert <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

for(i in 1:n.years){
  year.C <- h * (outer(meshpts, meshpts, Ct_z1z,  TempAnn[,i], TempAM1[,i], TempAM2[,i],
                       TempJJ[,i], RainAnn[,i], RainJJ[,i],
                       SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                       snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                       alpha = params.to.use))
  C.pert[i,,] <- year.C
}


Ct_z1z_mean <- function(z1,z,TempAnn, TempAM1, TempAM2,TempJJ, RainAnn, RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn, PackD, alpha){

  reproduction <- pb_z(z, TempAnn, TempAM2, RainAnn, RainJJ, Snow_after,
                       snow_f_year, Precip, PackAnn, PackD, alpha)

  recruit.size <- c_z1z(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                        SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                        PackD, alpha)
  recruit.size.mean <- mu_recruit_size(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                                       SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                                       PackD, alpha)
  recruit.size.d <- d_mu_recruit_size(z1, z, TempAnn, TempAM1, TempAM2, TempJJ, RainAnn, RainJJ,
                                      SnowAnn, Snow_before, Snow_after, snow_f_year, Precip, PackAnn,
                                      PackD, alpha)
  return( p_r*reproduction* recruit.size * (z1-recruit.size.mean )/(sdhat.recruit.size^2 )
          *recruit.size.d*TempAnn*alpha[1,5])
}

C.pert.mean <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

```

```
for(i in 1:n.years){
  year.C <- h * (outer(meshpts, meshpts, Ct_z1z_mean,  TempAnn[,i], TempAM1[,i], TempAM2[,i],
                         TempJJ[,i], RainAnn[,i], RainJJ[,i],
                         SnowAnn[,i], Snow_before[,i], Snow_after[,i],
                         snow_f_year[,i], Precip[,i], PackAnn[,i], PackD[,i],
                         alpha = params.to.use))
  C.pert.mean[i,,] <- year.C
}


pert.K <- stoc_pert_analysis(params.to.use, n.est, n.runin, C.pert, C.pert.mean)

elas.s             <- sum(pert.K$elas.s)
elas.s.mean <- sum(pert.K$elas.s.mean)
elas.s.sd        <- elas.s-elas.s.mean
cat("Stochastic elasticity ",elas.s,"\n")
cat("Stochastic elasticity mean ",elas.s.mean,"\n")
cat("Stochastic elasticity sd ",elas.s.sd,"\n")
cat("Stochastic sensitivity  mean ",pert.K$Ls*elas.s.mean/beta.rc.2.mean,"\n")
```

```
############################################
## Kernel Perturbation by Single Index Model
############################################

workr2 <- read_csv("work6_2017.csv")
source("Standard Graphical Pars.R")
weather1 <- read_csv("weather1.csv")
load("SIM alpha.Rdata")
source("SIM.R")


#####################################################################
#Stochastic perturbation analysis
#####################################################################
set.seed(53241986)

nBigMatrix <- 100
n.est <- 50000
n.runin <- 500
minsize <- 4.67
maxsize <- 7.05
n.years <- 26
params <- alpha

stoc_pert_analysis<-function(params,n.est,n.runin){

  year.i <- sample(1:n.years,n.est+1,replace=TRUE)

```

```r
K.year.i <- array(NA,c(n.years,nBigMatrix,nBigMatrix))
P.year.i <- array(NA,c(n.years,nBigMatrix,nBigMatrix))
F.year.i <- array(NA,c(n.years,nBigMatrix,nBigMatrix))

for(i in 1:n.years){

  year.K<- mk_k(nBigMatrix,params[,i], TempAnn[i], TempAM1[i], TempAM2[i],
                TempJJ[i], RainAnn[i], RainJJ[i],
                SnowAnn[i], Snow_before[i], Snow_after[i],
                snow_f_year[i], Precip[i], PackAnn[i], PackD[i], minsize, maxsize)

  K.year.i[i,,] <- year.K$K
  P.year.i[i,,] <- year.K$P
  F.year.i[i,,] <- year.K$F

  }

h <- year.K$h;

#Calculate mean kernel, v and w

mean.kernel <- apply(K.year.i,2:3,mean)
mean.kernel.P <- apply(P.year.i,2:3,mean)
mean.kernel.F <- apply(F.year.i,2:3,mean)

w <- Re(eigen(mean.kernel)$vectors[,1]);
v <- Re(eigen(t(mean.kernel))$vectors[,1]);

# scale eigenvectors <v,w>=1
w <- abs(w)/sum(h*abs(w))
v <- abs(v)
v <- v/(h*sum(v*w))
cat(h*sum(v*w)," should = 1","\n")

#Esimate Lambda s
#initialize variables

nt<-rep(1/nBigMatrix,nBigMatrix)
rt.V <- rt.N <- rep(NA,n.est)

#Iterate model

for (year.t in 1:n.est){
  if(year.t%%10000==0) cat("iterate: ", year.t,"\n");


  #iterate model with year-specific kernel
  nt1<-K.year.i[year.i[year.t],,] %*% nt

```

```
    sum.nt1<-sum(nt1)

    #Calculate log growth rates

    rt.V[year.t] <- log(sum(nt1*v)/sum(nt*v))
    rt.N[year.t] <- log(sum(nt1)/sum(nt))
    nt <- nt1 / sum.nt1

  }

  Ls <- exp(mean(rt.V))


  ### Get wt and Rt time series ###
  wt<-matrix(1/nBigMatrix, nrow=n.est+1, ncol=nBigMatrix);

  for (i in 1:n.est) {

    K            <- K.year.i[year.i[i],,]
    wt[i+1,]  <-K %*% wt[i,]
    wt[i+1,]  <-wt[i+1,]/sum(wt[i+1,]);
    if(i%%10000==0) cat("wt ",i,"\n")

  }


  ### Get vt time series ###
  vt<-matrix(1/nBigMatrix, nrow=n.est+1, ncol=nBigMatrix);
  for (i in (n.est+1):2) {

    K           <- K.year.i[year.i[i],,]
    vt[i-1,]  <- vt[i,] %*% K
    vt[i-1,]  <- vt[i-1,]/sum(vt[i-1,]);
    if(i%%10000==0) cat("vt  ",i,"\n")

  }

  sens.s <- matrix(0,nrow=nBigMatrix,ncol=nBigMatrix);

  elas.s <- sens.s


  for (i in n.runin:(n.est-n.runin)) {

    #standard calculations needed for the various formulae

    K                  <-   K.year.i[year.i[i],,]
    vt1.wt         <-   vt[i+1,]%*%t(wt[i,])
    vt1.K.wt    <- sum(vt[i+1,] * (K %*% wt[i,]))

    #calculation of the standard sensitivities and elasticities
```

```
    sens.s<-sens.s+vt1.wt/vt1.K.wt;

    elas.s<-elas.s+K*(vt1.wt/vt1.K.wt);

  }

  elas.s <- elas.s/(n.est-2*n.runin+1)
  sens.s <- Ls*sens.s/(n.est-2*n.runin+1)

  return(list(meshpts=year.K$meshpts,h=h,sens.s=sens.s,elas.s=elas.s,
              mean.kernel=mean.kernel,Ls=Ls, mean.kernel.P=mean.kernel.P,
              mean.kernel.F=mean.kernel.F))

}


########################################################################

pert.K <- stoc_pert_analysis(wm.par.est.mm,n.est,n.runin)

meshpts <- pert.K$meshpts
h.inv.2 <- 1 / (pert.K$h^2)

Kmean.elas <- pert.K$mean.kernel * pert.K$sens.s / pert.K$Ls
Pmean.elas <- pert.K$mean.kernel.P * pert.K$sens.s / pert.K$Ls
Fmean.elas <- pert.K$mean.kernel.F * pert.K$sens.s / pert.K$Ls

K.sens <- pert.K$sens.s * h.inv.2
K.elas <-  pert.K$elas.s * h.inv.2
K.mean.elas <- Kmean.elas * h.inv.2
P.mean.elas <- Pmean.elas * h.inv.2
F.mean.elas <- Fmean.elas * h.inv.2

K.sd.elas <- K.elas - K.mean.elas

cat("sum elasticities = ",sum(pert.K$elas.s)," should be 1")

## plot these
dev.off()
par(mfrow=c(1,2))
ikeep <- which(meshpts>5 & meshpts<6.5) # use to extract a region to plot
cols = rev(colorRampPalette(c('darkred','red','blue','lightblue'))(24))
filled.contour(meshpts[ikeep], meshpts[ikeep], t(P.mean.elas[ikeep,ikeep])
    ,
               xlab="mass (t), z", ylab="mass (t+1), z\'", col=cols)

filled.contour(meshpts[ikeep], meshpts[ikeep], t(F.mean.elas[ikeep,ikeep])
    ,
               xlab="mass (t), z", ylab="mass (t+1), z\'",  col=cols)
```