Statistical Modeling, Exploration, and Visualization of Snow Water Equivalent Data

James Beguah Odei
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/etd

Part of the Mathematics Commons, and the Statistics and Probability Commons

Recommended Citation
https://digitalcommons.usu.edu/etd/3871

This Dissertation is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.
STATISTICAL MODELING, EXPLORATION, AND VISUALIZATION OF
SNOW WATER EQUIVALENT DATA

by

James Beguah Odei

A dissertation submitted in partial fulfillment
of the requirements for the degree
of
DOCTOR OF PHILOSOPHY
in
Mathematical Sciences
(Statistics)

Approved:

Dr. Jürgen Symanzik
Major Professor

Dr. Piotr S. Kokoszka
Committee Member

Dr. John R. Stevens
Committee Member

Dr. D. Richard Cutler
Committee Member

Dr. Jiming Jin
Committee Member

Dr. Mark R. McLellan
Vice President for Research and
Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah
2014
ABSTRACT

Statistical Modeling, Exploration, and Visualization of Snow Water Equivalent Data

by

James Beguah Odei, Doctor of Philosophy
Utah State University, 2014

Major Professor: Dr. Jürgen Symanzik
Department: Mathematics and Statistics

Due to a continual increase in the demand for water as well as an ongoing regional drought, there is an imminent need to monitor and forecast water resources in the Western United States. In particular, water resources in the Intermountain West rely heavily on snow water storage. Thus, the need to improve seasonal forecasts of snowpack and considering new techniques would allow water resources to be more effectively managed throughout the entire water-year. Many available models used in forecasting snow water equivalent (SWE) measurements require delicate calibrations. In contrast to the physical SWE models most commonly used for forecasting, we offer a statistical model. We present a data-based statistical model that characterizes seasonal snow water equivalent in terms of a nested time-series, with the large scale focusing on the inter-annual periodicity of dominant signals and the small scale accommodating seasonal noise and autocorrelation. This model provides a framework for independently estimating the temporal dynamics of SWE for the various snow telemetry (SNOTEL) sites. We use SNOTEL data from ten stations in Utah over 34 water-years to implement and validate this model. This dissertation has three main goals: (i) developing a new statistical model to forecast SWE; (ii) bridging existing R packages
into a new R package to visualize and explore spatial and spatio-temporal SWE data; and (iii) applying the newly developed R package to SWE data from Utah SNOTEL sites and the Upper Sheep Creek site in Idaho as case studies.
PUBLIC ABSTRACT

Due to a continual increase in the demand for water as well as an ongoing regional drought, there is an imminent need to monitor and forecast water resources in the Western United States. In particular, water resources in the Intermountain West rely heavily on snow water storage. Thus, the need to improve seasonal forecasts of snowpack and considering new techniques would allow water resources to be more effectively managed throughout the entire water-year. Many available models used in forecasting snow water equivalent (SWE) measurements require delicate calibrations.

In contrast to the physical SWE models most commonly used for forecasting, we offer a statistical model. We present a data-based statistical model that characterizes seasonal snow water equivalent in terms of a nested time-series, with the large scale focusing on the inter-annual periodicity of dominant signals and the small scale accommodating seasonal noise and autocorrelation. This model provides a framework for independently estimating the temporal dynamics of SWE for the various snow telemetry (SNOTEL) sites. We use SNOTEL data from ten stations in Utah over 34 water-years to implement and validate this model.

This dissertation has three main goals: (i) developing a new statistical model to forecast SWE; (ii) bridging existing R packages into a new R package to visualize and explore spatial and spatio-temporal SWE data; and (iii) applying the newly developed R package to SWE data from Utah SNOTEL sites and the Upper Sheep Creek site in Idaho as case studies.

James Beguah Odei
I dedicate this work to my entire family, especially my late father, Samuel Kwasi Odei, and children, Alvin and Kenneth.
ACKNOWLEDGMENTS

I am sincerely grateful to my advisor, Dr. Jürgen Symanzik, for his extensive mentoring and endless enthusiasm. I am also thankful for him always being an example of academic excellence and professionalism for me. His constant support, guidance, and devotion to research and teaching kept me focused and productive throughout my study despite obvious and hidden obstacles. Without Jürgen’s generous supervision, this dissertation and my whole career as a statistician would be impossible.

I am also grateful to Professors Richard Cutler, John Steven, Piotr Kokoszka, and Mevin Hooten for many helpful comments and useful suggestions, and for valuable technical assistance, mathematical insights, and resources. I sincerely appreciate Mevin’s patience with me at certain stages of my scientific career. The excellent courses they offered significantly extended the horizons of my scientific interests.

I would like to thank Professor Adele Cutler, Cindy Moulton, and the rest of the staff at the Department of Mathematics and Statistics. This dissertation would not have been complete without their positive encouragements and support.

James Beguah Odei
CONTENTS

ABSTRACT ................................................................. iii
PUBLIC ABSTRACT ....................................................... v
ACKNOWLEDGMENTS ...................................................... vii
LIST OF TABLES ........................................................... xii
LIST OF FIGURES ......................................................... xiii

1 GENERAL INTRODUCTION ............................................. 1
  1.1 Background ....................................................... 1
  1.2 Snow Modeling ................................................... 2
    1.2.1 Types of Snow Models Used ............................... 2
    1.2.2 Problems Associated with Current Snow Models ....... 4
  1.3 Goals and Objectives of this Dissertation .................... 5
    1.3.1 Goal 1: Development of a New Statistical Model to Forecast Snow
          Water Equivalent Data ....................................... 5
    1.3.2 Goal 2: Development of a New R Package for the Visualization and
          Exploration of Spatial and Spatio-Temporal SWE Data .... 6
    1.3.3 Goal 3: Application of the Newly Developed R Package in Two Case
          Studies ......................................................... 6

2 A BAYESIAN HIERARCHICAL MODEL FOR FORECASTING INTERMOUNTAIN SNOW DYNAMICS .... 7
  2.1 Introduction ..................................................... 7
  2.2 Material and Methods ........................................... 10
    2.2.1 SNOTEL Data Sources ..................................... 10
    2.2.2 Dimension Reduction ........................................ 13
    2.2.3 Nested Time–Series Models ............................... 15
  2.3 Results ............................................................ 20
  2.4 General Assessments ............................................. 31
  2.5 Conclusions ...................................................... 33

3 SWEVIS: AN R PACKAGE FOR EXPLORING AND VISUALIZING SPATIAL AND SPATIO-TEMPORAL DATA ........................................... 36
  3.1 Introduction ..................................................... 36
  3.2 Why Use R? ...................................................... 37
  3.3 Existing Exploratory Spatial Data Analysis and Visualization Tools for Spatial and Spatio-Temporal Data ................................. 38
  3.4 Development of the SWEVIS R Package for Visualizing SWE Data .... 43
    3.4.1 The SWEVIS R Package ..................................... 46
4 APPLICATION OF THE SWEVIS R PACKAGE USING UTAH SNOTEL SITES AND THE UPPER SHEEP CREEK SITE AS CASE STUDIES

4.1 Introduction ................................................. 53
4.2 Data ....................................................... 53
  4.2.1 Utah SNOTEL Data ..................................... 53
  4.2.2 Upper Sheep Creek Watershed Data .................... 54
4.3 Single SNOTEL Site SWE Visualization .................... 57
4.4 Multiple SNOTEL Sites SWE Visualization ............... 65
4.5 Upper Sheep Creek SWE Visualization .................... 78

5 SUMMARY AND CONCLUSIONS .............................. 86

REFERENCES .................................................... 89

APPENDICES .................................................... 100

APPENDIX A SUMMARY OF COMPUTER R CODE ............ 101
APPENDIX B DERIVATION OF THE FULL-CONDITIONAL DISTRIBUTIONS .................................................... 102

APPENDIX C PROCEDURAL PARADIGM #1 (R CODE) .... 105
  C.1 Function to Read SNOTEL SWE Data .................. 105
  C.2 Inverse Gamma Start Function ......................... 105
  C.3 SWE SVD-MCMC Function ................................ 106
  C.4 Posterior Prediction Plot Functions ................. 111
    C.4.1 Functions for Tony Grove SNOTEL Site ............ 112
    C.4.2 Functions for Little Bear SNOTEL Site .......... 121
    C.4.3 Functions for Horse Ridge SNOTEL Site ........ 129
  C.5 Function to Determine Proportion Of SWE Measurements Outside the Credible Intervals ....................... 131
  C.6 Function to Estimate Missing SWE Measurements Using Kriging .............................................. 132
  C.7 Function to Download Utah SNOTEL Historical SWE Measurements for 90 Sites ............................. 133
  C.8 Function to Download Utah SNOTEL Current SWE Measurements for 90 Sites .............................. 137

APPENDIX D PROCEDURAL PARADIGM #2 (R CODE) ...... 140
  D.1 To Import Needed Functions and Data ................. 140
  D.2 Creating Figure 2.1 .................................. 143
  D.3 Creating Figure 2.2 .................................. 145
  D.4 Combining Augmented Historical and Current SWE Data and Selecting the Ten SNOTEL Sites ....................... 146
  D.5 Performing Singular Value Decomposition and Creating Table 2.1 .............................................. 147
  D.6 Choosing a Reasonable q-Value for Dimension Reduction .................................................... 148
  D.7 Creating Figure 2.3 .................................. 155
  D.8 Creating Figure 2.4 .................................. 156
  D.9 Creating Figure 2.5 .................................. 157
  D.10 Applying the SVD/MCMC Function to the 2008 Water-Year for the Ten Selected UTAH SNOTEL Sites ....................... 158
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.11 Creating the Tables for the Proportion of Observed SWE After Selected Initiation Dates in the 2008 Water-Year</td>
<td>160</td>
</tr>
<tr>
<td>D.12 Obtaining the Plots of SWE Measurements and Posterior Distributions As of the Selected Initiation Dates in the 2008 Water-Year</td>
<td>162</td>
</tr>
<tr>
<td>D.13 Creating Figure 2.6</td>
<td>167</td>
</tr>
<tr>
<td>D.14 Creating Figure 2.7</td>
<td>169</td>
</tr>
<tr>
<td>D.15 Creating Figure 2.8</td>
<td>169</td>
</tr>
<tr>
<td>D.16 Applying the SVD/MCMC Function to the 2009 Water-Year for the Ten Selected UTAH SNOTEL Sites</td>
<td>169</td>
</tr>
<tr>
<td>D.17 Creating the Tables for the Proportion of Observed SWE After Selected Initiation Dates in the 2009 Water-Year</td>
<td>172</td>
</tr>
<tr>
<td>D.18 Obtaining the Plots of SWE Measurements and Posterior Distributions As of the Selected Initiation Dates in the 2009 Water-Year</td>
<td>173</td>
</tr>
<tr>
<td>D.19 Applying the SVD/MCMC Function to the 2010 Water-Year for the Ten Selected UTAH SNOTEL Sites</td>
<td>178</td>
</tr>
<tr>
<td>D.20 Creating the Tables for the Proportion of Observed SWE After Selected Initiation Dates in the 2010 Water-Year</td>
<td>180</td>
</tr>
<tr>
<td>D.21 Obtaining the Plots of SWE Measurements and Posterior Distributions As of the Selected Initiation Dates in the 2010 Water-Year</td>
<td>182</td>
</tr>
<tr>
<td>D.22 Applying the SVD/MCMC Function to the 2011 Water-Year for the Ten Selected UTAH SNOTEL Sites</td>
<td>186</td>
</tr>
<tr>
<td>D.23 Creating the Tables for the Proportion of Observed SWE After Selected Initiation Dates in the 2011 Water-Year</td>
<td>189</td>
</tr>
<tr>
<td>D.24 Obtaining the Plots of SWE Measurements and Posterior Distributions As of the Selected Initiation Dates in the 2011 Water-Year</td>
<td>190</td>
</tr>
<tr>
<td>D.25 Applying the SVD/MCMC Function to the 2012 Water-Year for the Ten Selected UTAH SNOTEL Sites</td>
<td>195</td>
</tr>
<tr>
<td>D.26 Creating the Tables for the Proportion of Observed SWE After Selected Initiation Dates in the 2012 Water-Year</td>
<td>197</td>
</tr>
<tr>
<td>D.27 Obtaining the Plots of SWE Measurements and Posterior Distributions As of the Selected Initiation Dates in the 2012 Water-Year</td>
<td>199</td>
</tr>
<tr>
<td>D.28 Computing and Obtaining Barplots of the Ranked Probability Skill Scores</td>
<td>204</td>
</tr>
<tr>
<td>D.28.1 Creating Figure 2.11</td>
<td>210</td>
</tr>
</tbody>
</table>

**APPENDIX E SWEVIS R PACKAGE PARADIGM (R CODE)**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.1 ReadSweData</td>
<td>212</td>
</tr>
<tr>
<td>E.2 CalcSweSumStat</td>
<td>215</td>
</tr>
<tr>
<td>E.3 SimSweMCMCData</td>
<td>216</td>
</tr>
<tr>
<td>E.4 SwePlotData</td>
<td>217</td>
</tr>
<tr>
<td>E.5 SweRawDataPlot</td>
<td>220</td>
</tr>
<tr>
<td>E.6 SweHistPlot</td>
<td>222</td>
</tr>
<tr>
<td>E.7 SweBoxPlot</td>
<td>225</td>
</tr>
<tr>
<td>E.8 SweVariogPlot</td>
<td>227</td>
</tr>
<tr>
<td>E.9 SweRgoogleMap</td>
<td>228</td>
</tr>
<tr>
<td>E.10 SwePostPlot</td>
<td>228</td>
</tr>
<tr>
<td>E.11 iSweBrushMapSingle</td>
<td>229</td>
</tr>
<tr>
<td>E.12 iSwePlot</td>
<td>231</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>22</td>
</tr>
<tr>
<td>2.2</td>
<td>30</td>
</tr>
</tbody>
</table>

2.1 Showing the proportion of variance explained by the orthogonal components of each decomposition (signal).

2.2 Proportion of the observed SWE after the initiation dates: January 8, February 7, March 9, and April 8 in the five selected water-years, that lies outside the 50% and 95% credible intervals (CI) obtained for the posterior predictive distribution for the Tony Grove and Little Bear SNOTEL sites.
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Locations of 90 active SNOTEL sites in Utah. The black shaded diamonds and the triangle are the 10 sites with at least 30 years of SNOTEL measurements selected for model validation. The dashed lines represent county boundaries. The sites are: (1) Tony Grove, (2) Ben Lomond Peak, (3) Horse Ridge, (4) Little Bear, (5) Bug Lake, (6) Dry Bread Pond, (7) Monte Cristo, (8) Farmington, (9) Timpanogos Divide, and (10) Parley’s Summit. The Tony Grove SNOTEL site is further discussed in Section 2.2.1.</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Plots of SWE measurements (in inches) for the Tony Grove site (1979–2008). Each of these plots represents the amount of water stored during the water-year.</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>Boxplots for proportion of SWE that fell outside the 50% (a, c) and 95% (b, d) credible intervals of the posterior predictive distribution in forecasting as of January 8 in the 2008 water-year (a, b) and 2009 water-year (c, d) with important signals, ( q = 2, 4 ) and 6 over the ten selected SNOTEL sites. The spread is given in the boxplots, in which the median, lower and upper quartiles, and minimum and maximum values are given.</td>
<td>21</td>
</tr>
<tr>
<td>2.4</td>
<td>Plot of the orthonormal vectors, ( v_{tg} (t = 1, 2, \ldots, 365; g = 1, 2, 3, 4) ) containing the first four annual dominant signals. These plots may vary by year to be forecast and also depend on the data matrix set up in equation (2.1).</td>
<td>23</td>
</tr>
<tr>
<td>2.5</td>
<td>Plot of the left singular vectors (or latent time series for the dominant signals), ( \phi_{gt} (g = 1, 2, 3, 4; t = 1, 2, \ldots, 30) ) containing the inter-annual time series for the Tony Grove SNOTEL site based on the four important signals ( g ) obtained from the SWE data reduction (1979 – 2008).</td>
<td>24</td>
</tr>
<tr>
<td>2.6</td>
<td>Plot of the estimates of the model parameter, ( \alpha_{st} (s = 1, 2, \ldots, 10; t = 1, 2, \ldots, 30) ), over the ten selected SNOTEL sites based on the four important signals ( g ) obtained from the SWE data reduction (1979 – 2008). TG = Tony Grove, BLP = Ben Lomond Peak, HR = Horse Ridge, LB = Little Bear, BL = Bug Lake, DBP = Dry Bread Pond, MC = Monte Cristo, FAR = Farmington, TD = Timpanogos Divide, PS = Parley’s Summit.</td>
<td>25</td>
</tr>
<tr>
<td>2.7</td>
<td>Boxplots for the estimates of the model parameter, ( \alpha ), based on the four important signals ( g = 1, 2, 3, 4 ) obtained from the SWE data reduction in the 2008 water-year for the (a) Tony Grove and (b) Little Bear SNOTEL sites. The spread is given in the boxplots, in which the median, lower and upper quartiles, and minimum and maximum values are given.</td>
<td>26</td>
</tr>
</tbody>
</table>
2.8 Observed SWEs for 2008 water-year and posterior prediction as of January 8, 2008 (day 100; vertical brown dashed-dotted line) for the Tony Grove SNOTEL site. The horizontal axis shows the day (in water-year) and the vertical axis shows the SWE (in inches) in each plot. The previous data envelope (prev. data env., black dashed line) indicates the minimum and maximum SWE for each day from the past 29 water-years. The dark grey shaded region (pred. CI: 50%) and light grey shaded region (pred. CI: 95%) are estimates of the 50% and 95% credible intervals, respectively, of the posterior predictive distribution. The solid salmon line (prev. data mean) shows the average amount of SWE based on the past water-years and the solid orange line (current data) indicates the actual observed SWE for the current water year. The dotted black line (5th & 95th perc.) represents the 5th and 95th percentile of the SWE measured for each day from the past 29 water-years.

2.9 Observed SWEs and posterior predictions for the Tony Grove SNOTEL site. Rows 1 (top) through 5 (bottom) show plots for the 2008, 2009, 2010, 2011, and 2012 water-years, respectively. Columns 1 (left) through 4 (right) show posterior predictions based on the initiation dates (January 8, February 7, March 9, and April 8, respectively). See Figure 2.8 for additional details.

2.10 Observed SWEs and posterior predictions for the Little Bear SNOTEL site. Rows 1 (top) through 5 (bottom) show plots for the 2008, 2009, 2010, 2011, and 2012 water-years, respectively. Columns 1 (left) through 4 (right) show posterior predictions based on the initiation dates (January 8, February 7, March 9, and April 8, respectively). See Figure 2.8 for additional details.

2.11 RPSS for the ten selected SNOTEL sites for the (a) 2008, (b) 2009, (c) 2010, and (d) 2011 water-years with the January 8 initiation date used in the posterior predictive forecasting. TG = Tony Grove, BLP = Ben Lomond Peak, HR = Horse Ridge, LB = Little Bear, BL = Bug Lake, DBP = Dry Bread Pond, MC = Monte Cristo, FAR = Farmington, TD = Timpanogos Divide, PS = Parley’s Summit.

3.1 A choropleth map in grey scale showing the proportion of non-white births in North Carolina, 1974–1978. (Previously published as Fig 11.10 in Bivand et al. (2008), page 327).

3.2 Squared-differences variogram cloud for the scallops data. (Previously published as Figure 4.6 in Kaluzny et al. (1998), page 77).

3.3 A satellite image of Utah using the RgoogleMaps package in R. The orange shaded circles are the 10 selected SNOTEL sites used for our modeling efforts in Chapter 2. Further details of the locations of all 90 active SNOTEL sites in Utah can be seen in Figure 2.1.
3.4 Histograms of daily minimums, daily maximums, and daily averages of SWE measurements for the Farmington SNOTEL site for 30 water years. It is not intended to directly compare these plots; thus, no common scale has been used.

3.5 Boxplots of daily minimums, daily maximums, and daily averages of SWE measurements for the Farmington SNOTEL site for 30 water years. Again, it is not intended to directly compare these plots.

3.6 Variogram cloud plot of SWE for the ten selected SNOTEL sites discussed in Chapter 2 on December 8, 2009 (day 68 in the water year).

3.7 Snapshot of the created interactive environment (using iSwePlot) connecting the map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2, the variogram cloud plot of SWE (Figure 3.6), and plot of the selected SNOTEL site numbers.

4.1 The red shaded square shows the approximate location of the Upper Sheep Creek Watershed in Idaho in Owyhee County. The dashed lines represent county boundaries.

4.2 Topography and instrument locations within the Upper Sheep Creek Watershed (Previously published as Figure 1 in Flerchinger and Cooley (2000), page 87).

4.3 Disaggregation of the Upper Sheep Creek Watershed into three zones based on vegetation and snow accumulation. The low sagebrush, mountain big sagebrush, and aspen zones comprise 58.9%, 26.6% and 14.5% of the watershed, respectively (Previously published as Figure 2 in Flerchinger and Cooley (2000), page 89).

4.4 Snow water equivalent measured in Upper Sheep Creek March 3, 1993. The boundary represents the Upper Sheep Creek watershed. The dots represent locations of the grid stations where measurements of snow water equivalent (SWE) were taken. No grid stations are available at points 9N and 25D and point L10 was not measured (Previously published as Figure 2 in Luce and Tarboton (2004), page 1411).

4.5 Histograms of the minimum, maximum, and average of observed SWE measurements for each day of the 365 days in the water-year over the past 30 years for the Horse Ridge SNOTEL site. It is not intended to directly compare these plots; thus, no common scale has been used.

4.6 Boxplots of the minimum, maximum, and average of observed SWE measurements for each day of the 365 days in the water-year over the past 30 years for the Horse Ridge SNOTEL site. Again, it is not intended to directly compare these plots; thus, no common scale has been used.
4.7 Histogram and boxplot of the observed SWE measurements for the Horse Ridge SNOTEL site. The histogram shows the observed SWE measurements for each day of the 365 days in the 2009 water-year. The boxplot shows comparable plots for the observed SWE measurements for each day of the 365 days (in a water-year) for the previous 30 water-years (1979 – 2008). The two highest amounts of SWE were recorded in the water-years 6 (1984) and 19 (1997). .......................... 62

4.8 Observed SWEs and posterior predictions for the 2009 water-year for the Horse Ridge SNOTEL site based on the initiation date January 8. See Figure 2.8 for additional details. .......................... 63

4.9 Snapshot of the created interactive environment (using iSwePlot) connecting the map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2 and the six statistical plots. Here the Horse Ridge is selected on the map view, yielding the corresponding histograms, boxplots, and posterior predictive plot of its SWE measurements. The approximate horizontally brushed highest 15% of the SWE values in the posterior predictive plot is displayed immediately in all the other statistical plots. The SWE boxplot window shows 6 boxplots: 3 for the brushed data (in red color) and 3 for the non-brushed data (in white color). .......................... 66

4.10 Comparing boxplots of daily minimums, daily maximums, and daily averages of SWE measurements over $T = 30$ years (with 2009 as the current water-year) for the four selected SNOTEL sites. Rows 1 (top) through 2 (bottom) show plots for the Ben Lomond Peak (top left), Little Bear (top right), Monte Cristo (bottom left), and Timpanogos Divide (bottom right) SNOTEL sites, respectively. The SWE daily minimums, daily maximums, and daily averages are obtained by finding the minimum, maximum, and average SWE for each day of the 365 days in the water-year over the past $T = 30$ years (with 2009 as the current water-year). .................. 69

4.11 Comparing histograms of daily minimums, daily maximums, and daily averages of SWE measurements over $T = 30$ years (with 2009 as the current water-year) for the four selected SNOTEL sites. Rows 1 (top) through 4 (bottom) show plots for the Ben Lomond Peak, Little Bear, Monte Cristo, and Timpanogos Divide SNOTEL sites, respectively. The SWE daily minimums, daily maximums, and daily averages are obtained by finding the minimum, maximum, and average SWE for each day of the 365 days in the water-year over the past $T = 30$ years (with 2009 as the current water-year). .................. 70

4.12 Comparing histograms and boxplots of the observed SWE measurements for four selected SNOTEL sites. The histograms show observed SWE measurements for each day of the 365 days in the 2009 water-year. The boxplots shows comparable plots for the observed SWE measurements for each day of the 365 days (in a water-year) for the previous 30 water-years. Rows 1 (top) through 4 (bottom) show plots for the Ben Lomond Peak, Little Bear, Monte Cristo, and Timpanogos Divide SNOTEL sites, respectively. ........ 71
4.13 Comparing observed SWEs and posterior predictions for the 2009 water-year for the four selected SNOTEL sites based on the initiation date January 8. Rows 1 (top) through 2 (bottom) show plots for the Ben Lomond Peak (top left), Little Bear (top right), Monte Cristo (bottom left), and Timpanogos Divide (bottom right) SNOTEL sites, respectively. See Figure 2.8 for additional details.

4.14 Snapshot of the created interactive environment (using iSwePlot) connecting a map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2, the variogram cloud plot (Figure 3.6), and a plot of the selected SNOTEL site numbers. Here, in the the variogram cloud plot, the vertical and horizontal axes represent variogram estimates (gamma) and pairwise distance (in miles) respectively. In the plot of the selected SNOTEL site numbers, the axes represent numbers corresponding to the SNOTEL sites in the map view. One data point in the variogram cloud plot (bottom right) is selected, highlighting a data point in the plot of the SNOTEL site numbers (top right) and also, a corresponding pair of SNOTEL sites in the map view.

4.15 Snapshot of the created interactive environment (using iSwePlot) connecting a map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2, the variogram cloud plot (Figure 3.6), and a plot of the selected SNOTEL site numbers. Here, in the the variogram cloud plot, the vertical and horizontal axes represent variogram estimates (gamma) and pairwise distance (in miles) respectively. In the plot of the selected SNOTEL site numbers, the axes represent numbers corresponding to the SNOTEL sites in the map view. Two data points in the variogram cloud plot (bottom right) are selected, highlighting two data points in the plot of the SNOTEL site numbers (top right) and also, the corresponding three SNOTEL sites in the map view.

4.16 Snapshot of the created interactive environment (using iSwePlot) connecting a map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2, the variogram cloud plot (Figure 3.6), and a plot of the selected SNOTEL site numbers. Here, in the the variogram cloud plot, the vertical and horizontal axes represent variogram estimates (gamma) and pairwise distance (in miles) respectively. In the plot of the selected SNOTEL site numbers, the axes represent numbers corresponding to the SNOTEL sites in the map view. Three data points in the variogram cloud plot (bottom right) are selected, highlighting three data points in the plot of the SNOTEL site numbers (top right) and also, the corresponding five SNOTEL sites in the map view.
4.17 Snapshot of the created interactive environment (using iSwePlot) connecting map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2, the variogram cloud plot (Figure 3.6), and a plot of the selected SNOTEL site numbers. Here, in the the variogram cloud plot, the vertical and horizontal axes represent variogram estimates (gamma) and pairwise distance (in miles) respectively. In the plot of the selected SNOTEL site numbers, the axes represent numbers corresponding to the SNOTEL sites in the map view. Six SNOTEL sites are selected in the map view, highlight 15 data points in the variogram cloud plot (bottom right) as well as the plot of the SNOTEL site numbers (top right). ......................... 77

4.18 Snow water equivalent measured in the Upper Sheep Creek in 1993 on February 10, March 3, March 23, April 8, April 15, April 29, May 12, May 19, and May 25 (from left to right and top to bottom). Point L10 (black shaded square) was not measured – see Figure 4.4. ......................... 81

4.19 Difference in SWE measurements for the consecutive dates in Figure 4.18. Here, “1st–Difference n” represents the first difference in SWE measurements between the nth and (n+1)th dates. Point L10 (black shaded square) was not measured – see Figure 4.4. ......................... 82

4.20 Snapshot of the created interactive environment (using iSweAsciiPlot) connecting a map view of an image plot of SWE with SNOTEL stations within the Upper Sheep Creek watershed, boxplots of SWE measurements on March 3, 1993, and boxplots of elevations (in feet). Two of the boxplots are for brushed data values (in red color) and the other two are for non-brushed data values (in white color). Here, highlighted elevations above 6500ft create and highlight a boxplot in the SWE window with corresponding SNOTEL stations in the northeast that are also highlighted in the image plot. ...... 83

4.21 Snapshot of the created interactive environment (using iSweAsciiPlot) connecting a map view of an image plot of SWE with SNOTEL stations within the Upper Sheep Creek watershed, boxplots of SWE measurements on March 3, 1993, and boxplots of elevations (in feet). Two of the boxplots are for brushed data values (in red color) and the other two are for non-brushed data values (in white color). Here, highlighted elevations below 6200ft create and highlight a boxplot in the SWE window with corresponding SNOTEL stations in the northwest that are also highlighted in the image plot. ...... 84

4.22 Snapshot of the created interactive environment (using iSweAsciiPlot) connecting a map view of an image plot of SWE with SNOTEL stations within the Upper Sheep Creek watershed, boxplot of SWE measurements on March 3, 1993, and boxplot of elevations (in feet). Two of the boxplots are for brushed data values (in red color) and the other two are for non-brushed data values (in white color). Here, highlighted SWEs greater than 1 inch create and highlight a boxplot in the elevation boxplot window with eleven corresponding SNOTEL stations that are also highlighted in the image plot. ...... 85
1.1 Background

Snowmelt runoff is a major water resource in the western United States, as 50%–70% of the annual precipitation in the mountainous regions (Serreze et al., 1999) is snowfall, and 75%–85% of the annual stream-flow required for agricultural irrigation and other human activities is from snowmelt runoff (Grant and Kahan, 1974). It is also estimated that seasonal snow cover is the primary source of water supply for over 60 million people in the western United States, and that melting snow is responsible for 80% or more of soil moisture and streamflow in semi-arid mountain basins in this region (Bales et al., 2006; Marks et al., 2001).

In the western United States, accurate forecasting of snow water equivalent (SWE) remains a challenging problem due to insufficient observations and the strong heterogeneity of SWE distributions based on complex topography. Current observed droughts, and the general water crisis in the state of California, are a motivating example of why informed decision making, which ensures reliable and equitable distribution of this limited water resource, needs to be motivated by an understanding of the physical snowmelt process. Factors such as the quantity of available snow, and rate of snowmelt are important indicators to water resource officials who wish to forecast water availability.

Informed decision making, which ensures reliable and equitable distribution of this limited water resource, thus needs to be motivated by an understanding of the physical snowmelt process. Estimation of distributed SWE is challenging because of the many factors that affect its distribution, and the small correlation length of the SWE spatial distribution. Further, difficulties associated with accurately determining the time of maximum accumulation present a problem for snowmelt runoff forecasters.
1.2 Snow Modeling

1.2.1 Types of Snow Models Used

Understanding the linkages between processes controlling accumulation and melting of snow is critical in developing a predictive ability to describe a basin’s response to changes in water, energy and chemicals. Both conceptual and physical approaches have been employed when it come to snowmelt-runoff modeling. Conceptual models propose a mathematical relationship between snowmelt and measured quantities; thus melt can be obtained without factoring in all the the physical processes that affect snowmelt.

There are various approaches to estimating snow pack characteristics that differ in spatial scale, reliability, and accuracy. These include snow pack telemetry stations, direct remote sensing, and spatial simulation using landscape characteristics. Snow pack is typically measured using snow water equivalents (i.e., the amount of water in a column of snow; SWE). This is seen as a more accurate integrator of snow pack characteristics than snow pack depth because it combines both snow pack depth and density. The U.S. Department of Agriculture operates the Snow Telemetry (SNOTEL) system, which is a network of remote, automatic, monitoring stations that yield online daily measurements of SWE, precipitation, temperature and, more recently, snow depth.

A variety of several empirical statistical models have also been developed as alternative to approaches that directly measure dynamic snow pack characteristics, to map the spatial distribution of snow pack properties in terms of more-easily measurable influences such as terrain, vegetation, precipitation and, in some cases, wind exposure.

The use of direct remote sensing of snow-covered area, is another commonly used method of describing snow pack, although it is primarily a binary measure (snow cover vs no snow cover) of limited utility to wildlife analyses when compared with snow pack water equivalent Simic et al. (2004).

To forecast water resources, the National Weather Service (NWS) maintains a set of conceptual, continuous, hydrologic simulation models used to generate extended streamflow outlooks, and flood forecasts (Day, 1985; Hudlow, 1988). An integral part of the hydrologic
simulation models is a snow accumulation and ablation model that uses observed temperature and precipitation data to simulate the average snow water equivalent over a river basin. The estimates simulated by the snow accumulation and ablation model are then updated using observed ground-based and airborne snow water equivalent estimates when these data are available. Ground-based estimates are routinely made at manual snow course and automated snow telemetry (SNOTEL) sites throughout the West by the Natural Resources Conservation Service (NRCS). Airborne estimates are obtained by the NWS using low-flying aircraft to measure natural terrestrial gamma radiation emitted by the soil over 1700 flight lines in the United States and Canada.

A combination of regression of elevation, vegetation, and insolation with geostatistical interpolation of residuals has been used to map SWE distributions; elevation exerted the greatest influence on SWE distribution (Hosang and Dettwiler, 1991). For operational purposes, empirical approaches using combining remote sensing data to estimate snow-covered area, and snow depth networks to estimate SWE has improved (Martinec and Rango, 1991; Martinec et al., 1991). For example, McManamon et al. (1993) have combined airborne and ground based measurements to produce gridded SWE estimates for the upper Colorado River region.

To obtain precise water resource forecasts, it is essential that snow water equivalent estimates used to update the hydrologic simulation models be accurate. Recently, NWS has developed a spatial statistics model that uses the ground-based and airborne snow water equivalent data to estimate the snow water equivalent in areas where no observed measurements are available (Carroll et al., 1995; Day, 1990). Here, estimates of the snow water equivalent are obtained on a 30 arc second grid in river basins throughout the western United States. The estimates of the 30 arc second snow water equivalent in each of the river basins are the averaged to obtain areal snow water equivalent estimates for each basin, which are used as near real-time updates to the snow water equivalent estimates generated by the hydrologic simulation models.

Two current snow models included in terrestrial ecosystem models are the Terrestrial
Observation and Prediction System (TOPS) model (Nemani et al., 2003; White and Nemani, 2004) and the Biome-BGC model (Thornton, 1998; Thornton et al., 2002). The TOPS process-based snow melting and accumulation model is similar in principle to other process-based snow models. This snow model runs at daily time step using minimum, maximum, and average temperature precipitation, vapour pressure, and solar radiation. The Biome-BGC snow model is based on empirical temperature index model with radiation-driven melting. The model uses daily air temperature, precipitation, and solar radiation to simulate daily snow accumulation and melting processes (Thornton, 1998; Thornton et al., 2002).

The Variable Infiltration Capacity (VIC) hydrologic model is another physically based hydrologic model (Liang et al., 1994) used in estimating snow water equivalent. The VIC model accounts for fluxes of water and energy at the land surface and includes three soil layers and detailed representations of vegetation to simulate movement of soil moisture upward through plants and downward through the soil by infiltration and baseflow processes.

### 1.2.2 Problems Associated with Current Snow Models

Snow pack simulation models have generally been developed and used for purposes other than applications to ecological investigations. They are typically used in hydrological and climatological contexts, often at fairly coarse spatial resolutions designed to yield area-averaged estimates of snow pack properties and runoff over very large areas (Carroll et al., 1999; Etchevers et al., 2004; Melloh, 1999). Snow pack models developed for fine-scale ecological purposes to date are limited in spatial extent (Hiemstra et al., 2006).

To date, most of these empirical statistical models developed, lack some of the characteristics required for evaluating large mammal dynamics, such as: a means of scaling up to large study areas (Erickson et al., 2005), a means of incorporating temporal dynamics (Elder et al., 1998), deterministic results (Marchand and Killingtveit, 2005), or an objective, validated basis (Farnes et al., 1999).

One of the main obstacles of physically based modeling is the accumulation of of the necessary meteorological and snow-cover data to run, calibrate, and validate such models. While the study of hydrologic phenomena is a relatively mature eld, the hydrologic processes
pertaining to water and energy fluxes in mountainous regions, where basins are largely dominated by snowmelt, are yet not well captured by hydrologic models (Bales et al., 2006).

Though conceptual models benefit from less informational input, it suffers from the uncertainty that, parameters estimated under one set of model conditions are applicable to other conditions. In comparison to linear models, non-linear models may improve the prediction snowmelt but their use is limited due to the failure of transformed data satisfying the condition of nonlinearity (Dey et al., 1992).

Several efforts have been made using general circulation models (GCMs) to simulate and predict the SWE, making numerical models promising tools for SWE forecasting (Marshall and Oglesby, 2003). However, GCMs are unable to adequately capture snow-related atmospheric processes in mountainous areas because of overparameterized physics, coarse spatial resolution, and in many models, simplified land surface schemes. Improvements in the atmospheric water and energy transport were introduced to regional climate models (RCMs) by Grell et al. (1994) to better understand the physical processes affecting SWE evolution in mountainous regions (Kim et al., 2000; Liston et al., 1999). Nevertheless, coarse treatment of land surface snow physics in these models results in poorly simulated SWE and related processes (Leung and Qian, 2003).

1.3 Goals and Objectives of this Dissertation

This dissertation has three specific goals.

1.3.1 Goal 1: Development of a New Statistical Model to Forecast Snow Water Equivalent Data

The aforementioned shortcomings of the physical models, utilized in a conventional manner, impose a significant challenge in the modeling of SWE systems. However, we can effectively address such issues by utilizing statistically based snow models that rely heavily on observational data.

A general spatio-temporal statistical model was introduced in Odei et al. (2009). In this dissertation, we focus on a simplified version of the general spatio-temporal statistical
model that treats the SNOTEL sites independently. Using a Bayesian approach, we develop a hierarchical statistical model to forecast SWE data. Details can be found in Chapter 2.

1.3.2 Goal 2: Development of a New R Package for the Visualization and Exploration of Spatial and Spatio-Temporal SWE Data

We complement our modeling efforts from Goal 1 by developing a new SWEVIS R package for the visualization and exploration of spatial and spatio-temporal SWE data in a user-friendly interactive environment. All of the programming is done using R (Chapter 3). R is a free software environment for statistical computing and graphics, \url{http://www.r-project.org}. We bundle all of the code into an R package, which is a set of utility methods for managing, storing, and visualizing data, and results. With this SWEVIS package, one can explore and visualize SWE measurements for single or multiple SNOTEL sites. It also allows the end user to perform linked brushing between RGoogleMaps and statistical plots.

1.3.3 Goal 3: Application of the Newly Developed R Package in Two Case Studies

Many end users of the R package described in Goal 2 can be expected to have a background from environmental agencies or they might be individuals interested in the daily amount of snow measurements. These users want to focus on the results and graphics, and not on running computer code. Thus, we demonstrate the functions of the newly developed R package using SWE measurements from Utah SNOTEL sites and the Upper Sheep Creek site in Idaho as case studies (Chapter 4).
CHAPTER 2
A BAYESIAN HIERARCHICAL MODEL FOR FORECASTING INTERMOUNTAIN SNOW DYNAMICS

Summary

Due to a continual increase in the demand for water as well as an ongoing regional
drought, there is an imminent need to monitor and forecast water resources in the Western
United States. In particular, water resources in the Intermountain West rely heavily on snow
water storage. Thus, the need to improve seasonal forecasts of snowpack and considering
new techniques would allow water resources to be more effectively managed throughout
the entire water-year. Many available models used in forecasting snow water equivalent
(SWE) measurements require delicate calibrations. In contrast to the physical SWE models
most commonly used for forecasting, we offer a statistical model. We present a data-based
statistical model that characterizes seasonal snow water equivalent in terms of a nested
time-series, with the large scale focusing on the inter-annual periodicity of dominant signals
and the small scale accommodating seasonal noise and autocorrelation. This model provides
a framework for independently estimating mainly the temporal dynamics of SWE for the
various snow telemetry (SNOTEL) sites. We use SNOTEL data from 10 stations in Utah
over 34 water-years to implement and validate this model.

2.1 Introduction

Snow is considered to play significant roles in climate, terrestrial biosphere, and hy-
drology. As an integral part of the annual water budget, during the winter snowpack, water
is accumulated and then released in the spring snowmelt season (Ichii et al., 2007).

The Intermountain West region of the semi-arid Western United States (WUS) com-
prises of varying ecological and economic systems (Bailey, 1995) where over 75% of water

---

1Co-Authored by J. B. Odei, J. Symanzik, and M. B. Hooten. Reproduced from Environmetrics Journal,
resources result from snowmelt water. In this region, many different systems including snow courses, snow telemetry (SNOTEL), aerial markers, and airborne gamma radiation (Cowles et al., 2002), are used to measure the amount of water in the snow. The term Intermountain West generally comprises of six states including Utah. We use the term “intermountain” because our underlying principle can be extended to other states from this region.

A report from the National Climatic Data Center indicated that, since 1998, multi-year droughts have seriously affected the supply of water in the Southwest (http://www.wcc.nrcs.usda.gov/snotel/Utah/utah.html) and these droughts are some of the major risks residents and ecosystems of this region are facing (Mote et al., 2005). Thus, effective long-term management decisions must be made based on predictions concerning accumulation and melt of intermountain snow, a major water resource for the region. Modeling can offer a useful tool to perform such prediction, and physically based numerical computer models are broadly used to generate snow and related forecasts (Jin et al., 1999). The majority of these physical models are able to produce reasonable forecasts during the snow accumulation phase, but some perform poorly during the snowmelt stage due to inadequate representation of snowmelt regimes (Jin et al., 1999; Jin and Wen, 2012) and spring climate forcing (Saha et al., 2006).

With the growing concern about the effects of change in climate on snowpack accumulation (Lapp et al., 2005; Mote, 2006; Mote et al., 2005) and societal seasonal water supply requirements (Brekke et al., 2004; Miller et al., 2003; Stewart et al., 2005; Van Rheenen et al., 2004), accurate forecasting of water supplies has become increasingly urgent. Previous hydrologic simulation models and a spatial statistical model developed by the National Weather Service (NWS) enables us to address some of these questions. The NWS hydrologic simulation model generates extended streamflow predictions, water supply and spring flood outlooks, and flood forecasts (Day, 1985; Hudlow, 1988). With the spatial model, estimation of snow water equivalent (SWE) across the WUS river basins are obtained on a 30 arc second grid (Carroll and Cressie, 1996). An important part of the NWS spatial model is that its covariance function produces the covariance between the SWE at two sites.
In situations where areas have no observed measurements available, the NWS spatial statistical model uses ground-based and airborne SWE data (Carroll et al., 1995, 1999; Day, 1990).

Despite some successes in statistical snow modeling, there still remains a need for accurate seasonal forecasts. For example, Seidou et al. (2006) used Bayesian models for the interpolation of field measurements of SWE. Cowles et al. (2002) presented a Bayesian spatio-temporal analysis of the combined SWE data from four sources (snow courses, SNOTEL, aerial markers, and airborne gamma radiation) that allows for systematic differences in their accuracy and reliability. Their work also focused on identifying long-term temporal trends spatially. Additionally, Leisenring and Moradkhani (2011) described a statistical approach to “calibrating” a mechanistic snow model. This model was used to produce longer range forecasts, but it did not allow that some of the current water-year had already occurred. There are also alternative approaches for estimating SWE when it is not measured directly, for example, by linking it to auxiliary data sources such as snow depth and bulk snow density (Jonas et al., 2009).

Given the need for improved forecasts, we offer a new modeling direction, which is complimentary to other efforts. Some of the benefits of statistical models are: they are simple, very efficient in computation, and easy to use. Most importantly, statistically based snow models rely heavily on observational data, which greatly increases their reliability in forecasting. All these benefits are missing in physical models. In this article, we present a nested time-series statistical model for SWE data arising from the SNOTEL monitoring stations. Our objective here is to use an empirical Bayesian hierarchical statistical model to analyze our economically important variable: distribution of SWE. Our hierarchical model incorporates both spatial structure and measurement uncertainty.

Section 2.2 provides an overview on how the SWE data were obtained, dimension reduction, and a presentation on the formulation of the hierarchical model. In Section 2.3, we look at results after implementing the model from Section 2.2. We briefly discuss some important general assessments in Section 2.4 and state our conclusions in Section 2.5.
Appendix A provides an overview of the R (R Core Team, 2012) code used for the modeling. Appendix B provides the derivations of the full conditional distributions in this article.

2.2 Material and Methods

2.2.1 SNOTEL Data Sources

In this article, we provide a proof-of-concept application of our model to mountain ranges in the state of Utah. This state has active SNOTEL monitoring stations that provide various snow and environmental measurements (i.e., SWE, air temperature, precipitation, snow depth, soil moisture/temperature). In this article, 10 SNOTEL stations, each with 34 water-years (1979-2012) and 365 days per water-year yielding 124100 \((34 \times 10 \times 365)\) total SWE observations, were used. For simplicity, February 29 in leap years was ignored. These SNOTEL systems, installed, operated and maintained by the Natural Resources Conservation Service (NRCS), are designed to collect snowpack and other related climatic data in the WUS and Alaska. An illustration of the state-wide active SNOTEL sites of Utah is shown in Figure 2.1.

In near real time, SNOTEL uses meteor burst communications technology to communicate data where very high frequency (VHF) radio signals are reflected at a steep angle off the band of ionized meteors existing above the earth surface from about 50 to 75 miles (NRCS, 2009). Generally, these SNOTEL sites are located remotely on high watersheds where accessibility is difficult or restricted. For maintenance purposes, access involves the use of helicopters, skiing, snowmobiles, hiking, and snowshoeing. Overall, there are six NRCS data collection offices that monitor daily site statistics. When data are received, they are converted to engineering units and screened properly for errors and then stored in a database. The data (and additional information) are made available to the public through the National Water and Climate Center (NWCC) web site (http://www.wcc.nrcs.usda.gov).

Figure 2.2 shows plots of SWE measurements for the Tony Grove SNOTEL site from 1979 to 2008. The time of SWE measurements is in days of a water-year (Oct. 1 – Sept. 30). For example, in the 1979 water-year, SWE measurements were taken between October 1, 1978, and September 30, 1979. Years with large (e.g., 1997) and small (e.g., 1981, 1987)
Fig. 2.1: Locations of 90 active SNOTEL sites in Utah. The black shaded diamonds and the triangle are the 10 sites with at least 30 years of SNOTEL measurements selected for model validation. The dashed lines represent county boundaries. The sites are: (1) Tony Grove, (2) Ben Lomond Peak, (3) Horse Ridge, (4) Little Bear, (5) Bug Lake, (6) Dry Bread Pond, (7) Monte Cristo, (8) Farmington, (9) Timpanogos Divide, and (10) Parley’s Summit. The Tony Grove SNOTEL site is further discussed in Section 2.2.1.
Fig. 2.2: Plots of SWE measurements (in inches) for the Tony Grove site (1979–2008). Each of these plots represents the amount of water stored during the water-year.
measurements of overall SWE are readily apparent in Figure 2.2. Also, there were years (e.g., 1982, 2006) with relatively high SWE measurements early in the water-year and sharp decreases by the middle of the water-year. In addition to these observations, there are other factors or important (dominant) signals underlying the SWE measurements at the SNOTEL sites that cannot be seen directly, but nonetheless could be modeled. We will discuss in Section 2.2.2 how we can capture the important signals underlying the SWE measurements.

2.2.2 Dimension Reduction

Spatial and spatio-temporal data and processes of interest are naturally high dimensional, and when considered in a statistical context, it is common for model specifications to have a large number of parameters and state variables. A popular approach to dealing with high dimensional parameter spaces in spatial and spatio-temporal models (e.g. Shumway and Stoffer, 2006, state-space models) is to project the process onto a lower dimensional manifold for further modeling. This technique, sometimes referred to as fixed- or low-rank approximation (Wikle, 2010, Ch. 8), or fixed rank kriging in geostatistics (Cressie and Johannesson, 2008; Shi and Cressie, 2007), can be readily achieved through the use of a transformation involving a set of basis functions that map the latent lower-dimensional dynamic process to the scale at which the data are collected. It should be noted that, in theory, dimension reduction procedures reveal a set of latent, underlying processes. The data can then be fully utilized to learn about an underlying process which is governed by a parsimonious set of parameters. Many forms of basis functions are possible (e.g., wavelets, Fourier, splines, empirical orthogonal functions) and their utility varies depending on the goals of the study.

The predictor variables (e.g., SWE on a given date at several snowcourses) used in operational water supply forecasting are typically correlated. A satisfactory and statistically rigorous manner to deal with these inter-correlations is the use of principal component analysis (Garen, 1992). In the spatio-temporal literature, principal components are commonly used for such dimension reduction and are referred to as empirical orthogonal functions (EOFs) (Bjornsson and Venegas, 1997). The two most common ways to obtain these basis
functions are to decompose the space-time covariance matrix spectrally (eigen decomposition) (Golub and van Loan, 1996) or to use the singular value decomposition (SVD) (Golub and Reinsch, 1970) on a matrix of the data directly. In this article, we take the latter approach.

Based on the SNOTEL measurements, we obtain our SWE data for site \( s \) in matrix form, say \( \mathbf{Y}_{T \times M} = [\mathbf{y}_{s1}, \mathbf{y}_{s2}, \ldots, \mathbf{y}_{sT}]' \), where \( \mathbf{y}_{st} \) is of dimension \( 365 \times 1 \) and represents SWE measurements at site \( s \) in year \( t \) for all 365 days in a water-year, \( M = 365 \) days and \( T \) is determined by prior water-years. For example, for 2008 water-year, we only use data from 1979 through 2007 (29 years), for 2009 water-year, we use data from 1979 through 2008 (30 years), etc. So, \( T = 29 \) for 2008 water-year, 30 for 2009 water-year, etc. The sample average of SWE at site \( s \) for the past \( T \) years for day \( d \), \( \bar{\mathbf{y}}_{..d} \) is \( \frac{1}{T} \sum_{t=1}^{T} \mathbf{y}_{sdt} \) where the scalar \( y_{std} \) represents the observed SWE at site \( s \) in year \( t \) for day \( d \). The obtained SWE sample average, \( \bar{\mathbf{y}}_{..d} \), is then repeated at each site \( s \) for day \( d \) in all the \( T \) water-years to form a matrix \( \bar{\mathbf{y}} \) that has the same dimension as \( \mathbf{Y} \) \( (T \times M) \) with each row made up of the average SWE observed for each day at site \( s \) for \( T \) water-years. Thus, \( \bar{\mathbf{y}}_{T \times M} = [\bar{\mathbf{y}}_{..1}, \bar{\mathbf{y}}_{..2}, \ldots, \bar{\mathbf{y}}_{..d}, \ldots, \bar{\mathbf{y}}_{..365}] \), where \( \bar{\mathbf{y}}_{..d} \) is of dimension \( T \times 1 \) and contains \( T \) repetitions of the obtained SWE average, \( \bar{\mathbf{y}}_{..d} \), for day \( d \).

Applying SVD to our SNOTEL data, we decompose the SWE data after removing the day-of-year sample mean for site \( s \) independently as

\[
\mathbf{Y} - \bar{\mathbf{y}} = \hat{\mathbf{U}}\hat{\mathbf{D}}\hat{\mathbf{V}}' \approx \mathbf{U}\mathbf{D}\mathbf{V}'.
\]  

(2.1)

The approximation in equation (2.1) is the mechanism for reducing dimensionality in the parameter space. The \( \mathbf{U}\mathbf{D}\mathbf{V}' \) representation on the right hand side of equation (2.1) is a truncation of the \( \hat{\mathbf{U}}\hat{\mathbf{D}}\hat{\mathbf{V}}' \) decomposition based on the first \( q \) important signals (EOFs). It should be noted here that, \( q \) is much smaller than \( M \). The matrix of orthonormal vectors \( \mathbf{V} \) (first \( q \) columns of \( \hat{\mathbf{V}} \)) is of dimension \( M \times q \), and contains the important signals (EOFs). The matrix of left singular vectors, \( \mathbf{U} \) (first \( q \) columns of \( \hat{\mathbf{U}} \)), has dimension \( T \times q \) and contains the latent times series. The singular values are contained in \( \mathbf{D} \), a diagonal matrix of dimension \( q \times q \) with the eigenvalues on the diagonal which is obtained from the first \( q \)
eigenvalues (on the diagonal) of $\tilde{D}$.

Since $U$ contains the set of time series corresponding to each signal in $V$ for site $s$,

$$U = [u_{s1}, u_{s2}, ..., u_{sT}]'$$

where each $u_{st}$ is of dimension $q \times 1$, we can obtain approximations of the SWE for a particular year $t$, at site $s$, $y_{st}$ of dimension $M \times 1$ by adding the day-of-year sample mean to the product of $V$, $D$, and the time series corresponding to that selected year as

$$y_{st} \approx \tilde{y}_.. + VD u_{st}$$  \hspace{1cm} (2.2)

where $\tilde{y}_.$ (also of dimension $M \times 1$) consists of the averages of SWE measured for each day in the water-year at site $s$ for the past $T$ years. That is, $\tilde{y}_. = [\tilde{y}_{.1}, \tilde{y}_{.2}, ..., \tilde{y}_{.d}, ..., \tilde{y}_{.M}]'$. There may be an underlying physical process that may operate on a lower-dimensional manifold.

If we can parse the model into a data component that accounts for additional noise that cannot be explained and a process component represented by an underlying latent dynamic process, then we have defined a hierarchical model. For more details on this basic approach and its strengths, see Wikle and Hooten (2010). Thus, this general equation (2.2) provides motivation for a hierarchical statistical model.

### 2.2.3 Nested Time-Series Models

In a more general sense, by considering the temporal dependence of SWE among the SNOTEL sites, we can specify a model for all of the SWE data by partitioning the temporal structure into a large and a small temporal resolution. The specifications of the large and small temporal resolution are done using vector autoregressive (VAR) and conditional autoregressive (CAR) models, respectively. Thus, a nested time-series model with a spatio-temporal structure results.

**The Bayesian Hierarchical Statistical Model**

A general spatio-temporal statistical model was introduced in Odei et al. (2009). In this article, we focus on a simplified version of the general spatio-temporal statistical model that treats the SNOTEL sites independently. Using a Bayesian approach, we propose a
hierarchical statistical model for SWE. Letting the dynamics of SWE for any site $s$ be represented by a dimensionally reduced vector autoregression, for prediction (forecasting), we model $y_{st}$ as:

$$y_{st} = \bar{y}_s + VD A_{st} u_{s,t-1} + \varepsilon_{st},$$

(2.3)

with $\varepsilon_{st} \sim N(0, \Sigma_y)$, $t = 1, 2, ..., T$, and where $\bar{y}_s$ (of dimension $M \times 1$) consists of the averages of SWE measured for each day in the water-year at site $s$ for the past $T$ years. Thus, we have $\bar{y}_s = [\bar{y}_{s,1}, \bar{y}_{s,2}, ..., \bar{y}_{s,d}, ..., \bar{y}_{s,M}]'$. The covariance structure, $\Sigma_y$, handles the small scale temporal (daily) SWE correlation, which allows for latent autocorrelation in the days of the water-year. The latent time series for site $s$ in the $t^{th}$ year, $u_{st}$, in equation (2.2) is postulated as the product of a propagation matrix, $A_{st}$ and $u_{s,t-1}$, the latent time series in year $t-1$ for site $s$. The propagation matrix $A_{st}$ of dimension $q \times q$, handles the inter-annual scale temporal SWE correlation. This matrix could be specified in various ways depending on the desired amount of generality in the dynamics; it could also be linked to temporally varying covariates such as Pacific Decadal Oscillation (PDO) and El Niño/Southern Oscillation (ENSO). Here we treat $y_{st}$ as a joint multivariate normal distribution, which would be discussed further in Section 2.2.3.

The remaining parameters in our hierarchical model are defined as:

$$A_{st} = \text{diag}(\alpha_{st}),$$

$$\alpha_{st} \sim N(\alpha, \sigma^2_\alpha I),$$

$$\alpha \sim N(0, \sigma^2_\alpha I),$$

$$\Sigma_y = \sigma^2_y (I - \rho W)^{-1}.$$  

The CAR model specification we used for $W$ and $\rho$ are standard in both the spatial and time series statistical literature (Schabenberger and Gotway, 2005). $W$ itself in our case is fixed and thus doesn’t have a prior. It merely provides a feeling for which time points are neighbors of which other time points. $W$ is an $M \times M$ temporal proximity matrix that allows for structure between neighboring days. The elements of the matrix $W$, $w_{kd}$
(k, d = 1, 2, ..., M) are 1/2 if the absolute difference between the k\textsuperscript{th} and d\textsuperscript{th} days is one and it is 0 otherwise. Furthermore, recent literature in Markov random field modeling suggests that small \( \rho \) parameters are not very useful in describing dependence structures. Thus, the recommendation has been to use either a very strong prior on \( \rho \) that puts it close to 1, or simply remove it altogether (Banerjee et al., 2004). The latter choice yields an intrinsic conditional autoregressive (ICAR) model that is technically improper (in the probability sense) because it doesn’t integrate to one. However, we alleviate this problem by informing \( \rho \) to be less than one but very large (Banerjee et al., 2004). Thus, \( \rho \) is estimated by sampling from a truncated normal distribution with mean and standard deviation of 0.99 and 0.01 respectively on the interval \([-1, 1]\).

The variance component \( \sigma_\hat{y}^2 \) was assumed to have an inverse gamma distribution \( \sigma_\hat{y}^2 \sim IG(\nu_1, \nu_2) \) with initial values of the scale \( \nu_1 \) and shape \( \nu_2 \) parameters as 3 and 0.05, respectively. Here, the autoregression coefficients \( \alpha_{st} \), can be considered mixed effects that depend on the global coefficients, \( \alpha \). In order to make our priors on the \( \alpha_{st} \) and \( \alpha \) parameters vague, we specified the initial values of \( \sigma_\alpha^2 \) and \( \sigma_\alpha^2 \) as 10 and 100, respectively. With these types of models, variances are notoriously sensitive to prior specifications (Gelman, 2006). Thus, how well the model behaves would depend on good choices of initial values for the variances. In this article, we chose values that were relatively diffused (spread out) and conservative for what we thought the variances might likely be. Further detailed descriptions and applications of how a hierarchical model works can be found in Davis and Seaman (2002), Suess et al. (2002), and Wikle et al. (2001).

**Implementation**

As part of our Bayesian hierarchical statistical modeling, given the SWE measurements for any site \( s \), \( y_{st} \), we obtain the posterior distribution as proportional to the product of the likelihood of the data given the latent process models and parameter models as

\[
\left[\{\alpha_{st}\}, \alpha, \sigma_\hat{y}^2, \rho\right|\{z_{st}\}] \propto \left(\prod_{t=1}^{T} [z_{st}|\alpha_{st}, \sigma_\hat{y}^2] \times [\alpha_{st}|\alpha] \times [\alpha] \times [\sigma_\hat{y}^2] \times [\rho]\right),
\]

where \( z_{st} = y_{st} - \bar{y}_s \). The bracket notation \([ . \] denotes the probability distribution of the
argument contained within the brackets. We also obtain the full conditional distributions of \( \alpha_{st}, \alpha, \sigma^2_y \) which are straightforward to sample from, and they depend on the centered response variable \( z_{st} \). The full conditional distribution for both \( \alpha_{st} \) and \( \alpha \) is multivariate normal and that of \( \sigma^2_y \) is inverse gamma (see Appendix B).

Analytically, the posterior obtained in equation (2.4) is not tractable. Thus, with the estimate of the parameter \( \rho \), we then implemented the model using a hybrid Metropolis-Hastings and Gibbs Markov Chain Monte Carlo (MCMC) Sampling algorithm (Banerjee et al., 2004; Carlin and Lewis, 2009) using the R software (R Core Team, 2012). The R packages used as part of the model implementation include MCMCpack (Martin et al., 2011), mvtnorm (Genz and Bretz, 2009; Genz et al., 2011), msm (Jackson, 2011), lattice (Sarkar, 2008), Mass (Venables and Ripley, 2002), and spdep (Pebesma and Bivand, 2005; Bivand et al., 2008).

The MCMC algorithm was run for 5,000 iterations with burn-in periods of 1,250 iterations. Convergence, which was assessed visually, occurred rapidly and the posterior distribution appeared well characterized. Summaries of the post-convergence MCMC samples provide posterior inference for model parameters. The resulting MCMC samples were used to calculate summary statistics for all the latent processes and model parameters \( (\alpha, \alpha_{st}, \Sigma_y) \). Many statistical methods (e.g., simple and multiple linear regression models, and logistic regression models) provide confidence intervals, and can (in cross-validation mode) provide prediction intervals. The advantage of the simulation approach is that it provides estimates of the entire posterior distribution which permits us to obtain interval estimates as well. For example, the 2.5\(^{th}\) and the 97.5\(^{th}\) quantiles of the (post-convergence) sampled values for a model parameter provide a 95% interval estimate of the parameter. In the Bayesian framework, such an interval is termed credible interval or credible set to differentiate it from the confidence interval in frequentist statistics (Schabenberger and Gotway, 2005). Unlike confidence intervals, credible intervals have a direct probabilistic interpretation: A 95% credible interval defines an interval having a 0.95 posterior probability of containing the parameter of interest.
For the purpose of forecasting, we consider measurements of SWE for a water-year at site $s$, $y_{st}$ (for $t \in \{2, 3, ..., T\}$) partitioned into observed ($obs.$) and unobserved ($unobs.$) SWE measurements. Thus, we can think of both as a joint multivariate normal random vector

$$y_{st} = \begin{bmatrix} y_{st,obs.} \\ y_{st,unobs.} \end{bmatrix} \sim N \left( \begin{bmatrix} \bar{y}_{st-1,obs.} \\ \bar{y}_{st-1,unobs.} \end{bmatrix}, \begin{bmatrix} \Sigma_{obs.,obs.} & \Sigma_{obs.,unobs.} \\ \Sigma_{unobs.,obs.} & \Sigma_{unobs.,unobs.} \end{bmatrix} \right),$$

where for forecasting after day $d$, $\bar{y}_{st-1,obs.}$ is the average SWE over $t - 1$ previous years corresponding to days 1 to $d$ and $\bar{y}_{st-1,unobs.}$ is the average SWE over $t - 1$ previous years corresponding to days $d + 1$ to $M = 365$. Then, in the MCMC algorithm, we can use composition sampling (a technique for computing an integral) (Carlin and Lewis, 2009; Gelman et al., 2003) to obtain samples from the posterior predictive distribution for the unobserved SWE as

$$y_{st,unobs.|y_{st,obs.}} \sim N(\mu^*, \Sigma^*),$$

where we use multivariate normal results to obtain the conditional mean and variance

$$\mu^* = \bar{y}_{st-1,unobs.} + \Sigma_{unobs.,obs.}\Sigma_{obs.,obs}^{-1}(y_{st,obs.} - \bar{y}_{st-1,obs.}),$$

and

$$\Sigma^* = \Sigma_{unobs.,unobs.} - \Sigma_{unobs.,obs.}\Sigma_{obs.,obs}^{-1}\Sigma_{obs.,unobs.}.$$

**Model Validation**

For the purpose of validation, the model was implemented using different forecast initiation dates/days in a water-year and selected SNOTEL sites with at least 34 years of data for our posterior predictions. The initiation dates/days were chosen such that they were 30 days apart with the first initiation day being January 8 (when we have a reasonable amount of SWE data for meaningful predictions). Posterior predictive credible intervals
were also obtained for the “unobserved” hold-out SWE data. Here, the hold-out SWE data are the data obtained by splitting the original SWE data into two groups, the training and the testing sets. The choice of the train/test split was done using the day $d$ after which the forecasting would be made. Thus, SWE measurements from day 1 to day $d$ (training set) in the current water-year are considered observed and SWE measurements from day $d + 1$ to day $M = 365$ (testing set) in the current water-year are considered as unobserved when the model is implemented. The SVD is calculated on $t − 1$ previous years of SWE data prior to the forecast in the current water-year.

The probability skill of the SWE forecasts is assessed by means of the ranked probability skill score (RPSS; Wilks (1995)). The RPSS is used to evaluate a model’s skill in capturing categorical probabilities relative to climatology. The SWE values were divided into three categories: values below the 25th percentile, above the 75th percentile, and the rest between the 25th and 75th percentiles. The RPSS ranges from $-\infty$ to 1.0; the highest value indicating a perfect forecast and a negative value implying that the forecast has lesser accuracy as compared to that of climatology.

2.3 Results

Our Bayesian hierarchical model presented in Section 2.2.3 was fitted to the SWE data described in Section 2.2.1. SVD was performed on the data matrix in equation (2.1), the scaled SWE data ($Y - \bar{y}$), and truncated to obtain the important seasonal signals and inter-annual time-series.

To choose the value of $q$ for our truncation, we selected three different values (2, 4, and 6) and then constructed boxplots for the proportion of the actual SWE that fell outside the obtained 50% and 95% credible interval for the posterior predictive distribution to forecast SWE after 100 days (January 8) with $q = 2$, 4, and 6 in the 2008 and 2009 water-years based on the ten selected SNOTEL sites in this article. Overall, from Figure 2.3, with the exception of a few outliers, the boxplots for $q = 4$ showed a much shorter range of the proportion of the actual SWE that fell outside the obtained 50% and 95% credible intervals in both water-years. The initial SVD of the data matrix ($Y - \bar{y}$) for $T = 29$ and for $T = 30$
Fig. 2.3: Boxplots for proportion of SWE that fell outside the 50% (a, c) and 95% (b, d) credible intervals of the posterior predictive distribution in forecasting as of January 8 in the 2008 water-year (a, b) and 2009 water-year (c, d) with important signals, $q = 2, 4$ and 6 over the ten selected SNOTEL sites. The spread is given in the boxplots, in which the median, lower and upper quartiles, and minimum and maximum values are given.

years showed that four of the signals contributed about 95.5% of the variability in the data (see Table 2.1) and thus, provided a reasonable reduction in the dimension of the parameter set to facilitate estimation while retaining the dominant system dynamics. Based on these results, $q = 4$ was used for all following years. A plot of the first four dominant seasonal signals can be seen in Figure 2.4. The first dominant signal (solid black line, Signal 1), which accounts for approximately 81% of the variability in the data, reveals the overall average SWE measurements with the maximum around day 230 (May 18th) of the water-year which begins on October 1 in the previous calendar year. The second most important signal (dashed line, Signal 2), which explains an additional 10% of the variability in the
Table 2.1: Showing the proportion of variance explained by the orthogonal components of each decomposition (signal)

<table>
<thead>
<tr>
<th>Signal (g)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV</td>
<td>80.68%</td>
<td>9.89%</td>
<td>3.01%</td>
<td>1.92%</td>
<td>0.95%</td>
<td>0.67%</td>
</tr>
<tr>
<td>Cumulative PV</td>
<td>80.68%</td>
<td>90.57%</td>
<td>93.58%</td>
<td>95.50%</td>
<td>96.45%</td>
<td>97.12%</td>
</tr>
</tbody>
</table>

data, gives an indication of the discrepancy between the middle and the end of the water year.

Figure 2.5 shows a latent time series plot (say $\phi_{gt}$) from $U$ in equation (2.1) for the Tony Grove SNOTEL site ($s = 1$). In this figure, we consider the first four latent time series, which correspond to the expressions of the first four dominant signals through the years 1979–2008. We note here that, for this site, $\phi_{gt}$ is a $1 \times T$ matrix that consists of the $g^{th}$ row entries of $[u_{s1}, u_{s2}, u_{s3}, ..., u_{sT}]$. In the first latent time series ($g = 1$) plot, we observe that the smallest value of almost zero occurs in years numbered 3 (1981), 9 (1987), and 14 (1992) and the largest value occurs in year numbered 19 (1997). The years with the smallest values indicate a very low amount of snow and the year with the largest value had a large amount of snow. In case of the second latent time series ($g = 2$), a lower value indicates a smaller amount of snow at the beginning of the snow year (from October 1 to November 30) and a higher value indicates a large amount of snow. This time series shows a discrepancy between the middle (from February 1 to April 30) and late portion (between May 1 and June 30) of the year. Combining these first two time series, we get a general shape of the overall amount of snow for a particular snow season with approximately 90% of the dynamics represented.

In terms of the estimation of model parameters ($\alpha, \alpha_{st}, \Sigma_y$) in Section 2.2.3 based on the posterior predictive distributions obtained for the parameters of the propagation matrix $\alpha, \alpha_{st},$ and $\sigma_y^2$, using composition sampling, we found that although some of the posterior $\alpha_{st}$ were quite different from zero especially in the cases of the first two important signals ($g = 1, 2$), the global autoregression coefficients $\alpha$ were not (see Figures 2.6 and 2.7). This is likely due to the limited amount of inter-annual data ($T = 29$) or it could also be that...
Fig. 2.4: Plot of the orthonormal vectors, \( \mathbf{v}_{tg} \) \( t = 1, 2, ..., 365; g = 1, 2, 3, 4 \) containing the first four annual dominant signals. These plots may vary by year to be forecast and also depend on the data matrix set up in equation (2.1).
Fig. 2.5: Plot of the left singular vectors (or latent time series for the dominant signals), $\phi_{gt}$ ($g = 1, 2, 3, 4; \ t = 1, 2, ..., 30$) containing the inter-annual time series for the Tony Grove SNOTEL site based on the four important signals ($g$) obtained from the SWE data reduction (1979 – 2008).
Fig. 2.6: Plot of the estimates of the model parameter, $\alpha_{st}$ ($s = 1, 2, \ldots, 10; t = 1, 2, \ldots, 30$), over the ten selected SNOTEL sites based on the four important signals ($g$) obtained from the SWE data reduction (1979 – 2008). TG = Tony Grove, BLP = Ben Lomond Peak, HR = Horse Ridge, LB = Little Bear, BL = Bug Lake, DBP = Dry Bread Pond, MC = Monte Cristo, FAR = Farmington, TD = Timpanogos Divide, PS = Parley’s Summit.

there is no temporal autocorrelation at these time scales. The covariance matrix for the short time scale had a very large posterior range parameter $\sigma_y^2$, indicating a high degree of smoothness in the daily SWE measurements.

The posterior predictive credible intervals always performed well in terms of capturing the “unobserved” hold-out data as part of our model validation. We obtained posterior predictions for four different initiation dates/days (January 8, February 7, March 9, and April 8). The posterior predictions in the calendar days showed poor results for forecasting. This is due to the fact that we have little or no snow for that period of time in the water-year (especially from October 1 to November 30). Figure 2.8 shows an enlarged example of a
Fig. 2.7: Boxplots for the estimates of the model parameter, $\alpha$, based on the four important signals ($g = 1, 2, 3, 4$) obtained from the SWE data reduction in the 2008 water-year for the (a) Tony Grove and (b) Little Bear SNOTEL sites. The spread is given in the boxplots, in which the median, lower and upper quartiles, and minimum and maximum values are given.

plot for the posterior prediction of SWE as a posterior predictive mean with 50% and 95% credible intervals. Here we considered the Tony Grove SNOTEL site using the January 8 (day 100) initiation date for our posterior predictions.

The graphs of the posterior predictions for SWE based on the four different initiation dates in five consecutive water-years (2008, 2009, 2010, 2011, and 2012) with 50% and 95% credible intervals for the Tony Grove and Little Bear SNOTEL sites are displayed in Figures 2.9 and 2.10. Table 2.2 contains a summary of the percentages of the actual SWE measurements that fall outside the 50% and 95% credible intervals for the Tony Grove and Little Bear SNOTEL sites.

In Figure 2.9, we notice that in the 2011 water-year, for all the initiation dates chosen, the 95% credible interval for the posterior prediction was not able to capture well a large proportion of the actual SWE. Specifically, from the end of March towards the beginning of July, we notice SWE measures higher than the maximum from the previous 32 water-years. Similar extreme observations were also recorded at the Timpanogos Divide, Farmington, and Monte Cristo SNOTEL sites. This resulted in under-prediction due to our model being incapable of predicting outside the previous data envelope, i.e., the historic minimum and
Fig. 2.8: Observed SWEs for 2008 water-year and posterior prediction as of January 8, 2008 (day 100; vertical brown dashed-dotted line) for the Tony Grove SNOTEL site. The horizontal axis shows the day (in water-year) and the vertical axis shows the SWE (in inches) in each plot. The previous data envelope (prev. data env., black dashed line) indicates the minimum and maximum SWE for each day from the past 29 water-years. The dark grey shaded region (pred. CI: 50%) and light grey shaded region (pred. CI: 95%) are estimates of the 50% and 95% credible intervals, respectively, of the posterior predictive distribution. The solid salmon line (prev. data mean) shows the average amount of SWE based on the past water-years and the solid orange line (current data) indicates the actual observed SWE for the current water year. The dotted black line (5th & 95th perc.) represents the 5th and 95th percentile of the SWE measured for each day from the past 29 water-years.
Fig. 2.9: Observed SWEs and posterior predictions for the Tony Grove SNOTEL site. Rows 1 (top) through 5 (bottom) show plots for the 2008, 2009, 2010, 2011, and 2012 water-years, respectively. Columns 1 (left) through 4 (right) show posterior predictions based on the initiation dates (January 8, February 7, March 9, and April 8, respectively). See Figure 2.8 for additional details.
Fig. 2.10: Observed SWEs and posterior predictions for the Little Bear SNOTEL site. Rows 1 (top) through 5 (bottom) show plots for the 2008, 2009, 2010, 2011, and 2012 water-years, respectively. Columns 1 (left) through 4 (right) show posterior predictions based on the initiation dates (January 8, February 7, March 9, and April 8, respectively). See Figure 2.8 for additional details.
Table 2.2: Proportion of the observed SWE after the initiation dates: January 8, February 7, March 9, and April 8 in the five selected water-years, that lies outside the 50% and 95% credible intervals (CI) obtained for the posterior predictive distribution for the Tony Grove and Little Bear SNOTEL sites

| Water-Year | Tony Grove Site | | | | | | Little Bear Site | | | | |
|------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|            | Percentage for 50% CI | Percentage for 95% CI | Percentage for 50% CI | Percentage for 95% CI |
|            | Jan.8 | Feb.7 | Mar.9 | Apr.8 | Jan.8 | Feb.7 | Mar.9 | Apr.8 | Jan.8 | Feb.7 | Mar.9 | Apr.8 | Jan.8 | Feb.7 | Mar.9 | Apr.8 |
| 2008       | 25.6  | 29.8  | 32.8  | 31.7  | 0.0   | 1.5   | 11.3  | 1.5   | 34.3  | 19.2  | 16.2  | 13.2  | 7.2   | 5.7   | 0.4   | 1.9   |
| 2009       | 35.9  | 26.4  | 26.8  | 21.5  | 2.3   | 2.3   | 17.4  | 7.9   | 29.4  | 27.5  | 23.0  | 8.3   | 6.0   | 10.9  | 6.8   | 3.4   |
| 2010       | 36.6  | 30.2  | 28.7  | 18.1  | 8.3   | 7.5   | 3.8   | 4.9   | 26.0  | 24.2  | 8.7   | 9.8   | 0.4   | 6.0   | 4.5   | 6.8   |
| 2011       | 65.7  | 43.4  | 44.2  | 40.4  | 38.9  | 34.0  | 34.0  | 33.2  | 49.1  | 24.2  | 21.1  | 11.3  | 34.3  | 4.5   | 6.8   | 3.8   |
| 2012       | 40.4  | 43.4  | 36.6  | 18.9  | 4.5   | 30.9  | 29.8  | 12.5  | 26.8  | 27.2  | 28.3  | 7.5   | 20.4  | 7.9   | 16.6  | 1.1   |
maximum SWE. Overall, the 2011 water-year rather seemed to be an unusual year for many of the 10 selected SNOTEL sites as a result of the record high measurements of SWE. In the 2012 water-year, for the initiation dates February 7 (day 130) and March 8 (day 160), the unseemly over-prediction is due to the high amount of snow recorded in the prior months.

In Figure 2.10, we notice that our credible interval for the January 8 initiation date of the 2011 water-year captures only a small proportion of the actual SWE. This noticeable over-prediction was as a result of an unexpectedly high amount of SWE measured towards the end of December (above the expected average). Our obtained posterior predictions showed that the three other initiation dates in 2011 water-year yielded useful forecasts for the rest of the season. The January 8 and March 9 initiation dates of the 2012 water-year also posed some problems for our model, given the steep SWE increase in mid-January and the steep SWE decline shortly after March 9. Overall, the 95% credible intervals capture, on average, a large proportion of the actual SWE measurements in the other water-years (and initiation dates) for this site.

The probabilistic forecast skills, expressed by RPSS using the January 8 posterior predictions, are presented in Figure 2.11. In the 2008 water-year, with the exception of the Farmington site, all the SNOTEL sites show a positive RPSS. In addition, for the 2009 and 2010 water-years, all ten sites show a positive RPSS and, thus, outscore the climatology forecast. For the 2011 water-year, all the SNOTEL sites show a negative RPSS, an indication of lesser accuracy from our model compared to that of climatology forecast. This is in agreement with what was found using the predictive credible interval.

2.4 General Assessments

We used our Bayesian hierarchical model to successfully incorporate the SNOTEL SWE data in a likelihood and create posterior predictive distributions for the process of interest. Our model is comparable to other developed empirical statistical models; except those were restricted to making statistical inferences on distributions while our model relies heavily on observational data. Specifically, the statistical approach by itself does not explicitly incorporate knowledge about the atmospheric process, but provides a heavily data based
method for prediction. Using a fully-rigorous statistical model for the phenomenological behavior of SWE over space and time, we are able to properly learn about the uncertainty in our predictions of SWE for the near-future.

A critical part of modeling efforts is evaluating model performance. A number of cross-validation approaches are available when it comes to model evaluation. However, model fit within the prediction credible interval is assumed to be homogeneous for all traditional model-fit metrics. A benefit of the Bayesian approach is that inference is made based on the posterior distribution. For example, we considered plots of the posterior prediction of SWE as a posterior predictive mean with 50% and 95% credible intervals (see Figure 2.8). The results shown in Table 2.2 indicate that the credible intervals for the posterior
predictive of the non-spatial Bayesian hierarchical model proposed does well by capturing larger proportions of the actual SWE measured. For example in the case of the 95% credible interval, larger proportion means 80% or more of the SWE measured. In addition, we observed that the model performed weakly for the Tony Grove site for all four initiation dates in 2011 and for the Little Bear site for the January 8 initiation date in 2011 with respect to the 95% credible interval. This was mainly due to discrepancies in the SWE measurements compared to previous years; large amounts of snow were recorded in the month of December of the 2011 water-year for both sites. There were readily visible under-estimation, over-estimation and data running out of the past min-max envelopes, especially in the 2011 water-year.

2.5 Conclusions

The statistical model presented in this article provides a framework for independently estimating the temporal dynamics of SWE for various SNOTEL sites. Using a parsimonious set of important signals, we can reduce the dimension of the SWE process and obtain meaningful posterior forecasts for the remainder of the water-year, beginning on various starting dates in a year. Except for the 2011 water-year, we noticed our model predictive skills were positive. This is an indication that our model has some higher accuracy than that of climatology forecast.

Better knowledge about the uncertainty associated with snow water prediction could be put into a GIS framework to accumulate runoff volumes and provide information to farmers and land managers about future irrigation availability. Similarly, likely future water level ranges for reservoirs and lakes could be obtained in a similar fashion and provide information about fishery resources and recreational opportunities. Locally and regionally dry conditions contribute to wildfire likelihood, thus SWE predictions from our model could be aggregated at relevant spatial scales to provide an improved understanding of future soil moisture conditions at various time points in the late spring.

The statistical model presented in this article provides a framework for independently estimating the temporal dynamics of SWE for various SNOTEL sites. Using a parsimonious
set of important signals, we can reduce the dimension of the SWE process and obtain
meaningful posterior forecasts for the remainder of the water-year, beginning on various
starting dates in a year. Except for the 2011 water-year, we noticed our model predictive
skills were positive. This is an indication that our model has some higher accuracy than
that of climatology forecast. Our approach differs considerably from some of the predictive
SWE models, in that, it focuses on phenomenological modeling (rather than mechanistic)
and it produces within-season forecasts of SWE. In addition, our model is purely data-based
and does not rely on extensive mechanistic SWE models and their associated forcings and
parameterizations.

Better knowledge about the uncertainty associated with snow water prediction could
be put into a GIS framework to accumulate runoff volumes and provide information to
farmers and land managers about future irrigation availability. Similarly, likely future
water level ranges for reservoirs and lakes could be obtained in a similar fashion and provide
information about fishery resources and recreational opportunities. Locally and regionally
dry conditions contribute to wildfire likelihood, thus SWE predictions from our model could
be aggregated at relevant spatial scales to provide an improved understanding of future soil
moisture conditions at various time points in the late spring.

We noted that the statistical framework we presented could be modified to accom-
modate missing and misaligned data sources. Our model is currently not able to handle
locations where historical SWE measurements contain data gaps (missing or unobserved
values). However, we would be able to do meaningful forecasting if an appropriate imputa-
tion technique is first applied to fill these data gaps. This could be as simple as a moving
average if only a few observations are missing or as complex as co-kriging if SWE measure-
ments and other observations from nearby locations are available. New SWE stations are
being added regularly and though this increases our spatial resolution, it does not help us
learn about historical SWE patterns. These kinds of issues will not decrease our learning,
but they may enhance it in other aspects (e.g., spatial information instead of temporal
information) that we have not yet considered.
Our model was developed to best predict the remainder of a water-year given early information. It is able to properly account for the uncertainty in prediction so that it could be used to provide managers and scientists with an idea of how predictable a certain year might be. However, if one were interested in predicting a different quantity (like peak SWE) then a different model may need to be constructed and evaluated.

Ichii et al. (2007) discussed snow models in terrestrial biosphere models and they pointed out that snow dynamics, snow depth, snow water storage, and spring run off are factors that directly affect intermountain ecosystem and economic function. Knapp (1998) discussed wildfire prevalence, modeling, and prevention. In all of these scenarios, predicted SWE might be another useful component to further enhance these models. Moreover, predicted SWE might be useful for reservoir level forecasting for summer water availability, irrigation potential for agriculture, and avalanche research.

Future work might include focusing on relationships between the SWE and other response variables of interest (water supply, fire frequency, etc.). This way, we can forecast meaningful seasonal runoffs that would be used to make sound water management decisions. The relationship between SWE and fire frequency could assist land managers with the assessment of post-fire water quality, reforestation, and animal habitat issues. For the problem of historical SWE measurements with data gaps, one could also borrow the strengths of physical process models by combining model results using a method similar to Bayesian model averaging (Hoeting et al., 1999). Utilizing the components together, however, would be both advantageous and scientifically thorough and should provide better forecasts and understanding of snowpack dynamics. We will also consider the spatial dependence of all 90 active SNOTEL sites in Utah and we will include other spatial covariates (e.g., latitude, longitude, temperature, precipitation, and elevation) in our modeling efforts. Finally, we plan to implement the general statistical model that was introduced in Odei et al. (2009).
CHAPTER 3
SWEVIS: AN R PACKAGE FOR EXPLORING AND VISUALIZING SPATIAL AND SPATIO-TEMPORAL DATA

3.1 Introduction

This chapter consists of two main parts: the first part discusses existing exploration and visualization tools for spatial and spatio-temporal data, and the second part presents the classes and functionalities for the SWEVIS R package that bundles several of these tools. The SWEVIS R package consists of a set of utility methods for managing, storing, importing and exporting snow water equivalent (SWE) data and results. It also includes the statistical model for forecasting SWE discussed in Chapter 2. In addition, it also implements multivariate exploratory data analysis (EDA) and exploratory spatial data analysis (ESDA) techniques, which will be discussed in Section 3.3. Three main techniques will be introduced: (i) data manipulation methods, (ii) forecasting, and (iii) linkage between map views and interactive statistical graphics.

The implementation of the exploration and visualization functions for SWE data is done using R (R Core Team, 2012), which is a free software environment for statistical computing and graphics, and available at http://www.r-project.org. R compiles and runs on a wide variety of UNIX platforms, Windows, and MacOS. Over the last ten years, R has become one of the most widely used statistical software packages among statisticians and researchers. The R package repositories, and, specifically, the Comprehensive R Archive Network (CRAN, http://CRAN.R-project.org), currently provides more than 5000 specialized packages. In Section 3.2, we discuss why we chose R to implement the exploration and visualization techniques. There exist several R packages that are used to explore and visualize a variety of data types. In Section 3.3, we summarize the functionality of some of these existing packages. Then, in Section 3.4, we talk about how the SWEVIS R package
was developed and implemented in R.

3.2 Why Use R?

The R software environment (R Core Team, 2012) is widely used among statisticians as a procedural programming language. It is a tool for conducting statistical computing and graphics that has become increasingly popular in the fields of statistics, ecology, and fisheries science, among others. Its freely available source code and active community of scientists contributing extensions, called packages, to the base functionality of R make it a flexible tool for conducting statistical research. It is unique because it gives the developer the power to do three major things in one single tool:

- Data Manipulation: R allows the data scientist to shape the dataset into a format that could be easily accessed and analyzed by slicing large multivariate datasets. It is also one of the few tools that has great indexing techniques.

- Data Analysis: Any kind of statistical data analysis could be found in R. R is an open source development tool that is supported by a large community of statisticians, computer scientists, and applied scientists from all disciplines. It has over 5000 packages that implement various statistical analysis tools related to hypothesis testing, model fitting, clustering techniques, machine learning, and so on.

- Data Visualization: R is “the software” for visualization. There are many on-the-shelf graphic functions and packages in R that are ready to be used. The best part of R is that it gives the developer capabilities to implement any visualization idea for any dataset. In addition, animated and interactive graphs can be implemented easily in R.

In addition to the above mentioned advantages, R is also free, runs on any operating system, and can read data in any format (text, Excel, xml, CSV, etc.). It gives developers complete control over the system, and thus, they can build statistical systems that fit their users’ needs, and they can even make R interact with other programming languages such as Java, C, C++, Fortran, etc. Besides R’s graphical user interface (GUI) limitation to
command line interactions, some web interfaces exist. These R web interfaces enables users
to not write down the commands to load data, perform statistical analyses and create plots.
The interface RExcel for example, is an add-in for Microsoft Excel which allows access to
R from within Excel. Here Excel is used as a GUI for R, making R functionality accessible
through menus and dialog boxes instead of a command line oriented programming style.
Other R web interfaces include Rweb, Rcgi, webbioc, Rwui, and Rook.

In R, a large number of packages provide spatial statistical methods or access to GIS
data, and many of them provide data structures and plotting methods for spatial data.
A more flexible way to analyze spatial data is to use the spatial classes provided by the
sp package (Pebesma and Bivand, 2005). The sp package provides methods for projecting,
gridding, overlaying, and converting spatial data to and from data frames and matrices, and
to new or existing spatial statistics R packages that use or depend on it, such as splancs

3.3 Existing Exploratory Spatial Data Analysis and Visualization Tools for
Spatial and Spatio-Temporal Data

“Visualization refers not only to a set of graphical images but also to the iterative
process of visual thinking and interaction with the images” (Edsal et al., 2000). One goal of
data exploration is the recognition of patterns and the abstraction of structure and meaning
from data. Visualization can play an important role in bringing to light subtle patterns that
may not be immediately apparent in strictly quantitative data analysis methods. This is of
high relevance for complex spatial and spatio-temporal data sets.

Several visualization techniques are available today to visualize a variety of data types.
These data sets may be characterized as spatial (in 2 or 3 dimensions), spatio-temporal,
and multivariate. Multivariate data relates to more than one variable represented at each
location. In this dissertation proposal, we will focus on spatially continuous data and areal
data.

Spatially continuous data (also called geostatistical data) are data which have been
sampled at fixed point locations with spatial variation in a variable varying continuously
over the study area (Bailey and Gatrell, 1995). Typical examples include concentration of some pollutants, rainfall measurements, or geological measures on an ore deposit such as mineral grade. Additional examples of spatially continuous data are SWE, precipitation, and temperature measurements.

For areal data (also called lattice data), the variable of interest does not vary continuously, but has values only within a fixed set of areas or zones covering the study area. These fixed areas may either constitute a regular lattice (such as plots used in agricultural field trials, or pixels in remote sensing data) or they may consist of irregular areal units (such as health districts or census tracts).

Other types of spatial data, such as spatial point patterns and spatial interaction data, also exist and have been discussed in more detail in Cressie (1993) and Fischer and Wang (2011, Part II), respectively. We speak of spatio-temporal data when spatial data are collected over time (that is, when an additional temporal component is present).

Due to large databases available to researchers, new tools are needed for the analysis of this information that match the sophistication of storage, retrieval, and display provided by the rapidly evolving technology of Geographical Information Systems (GIS). In many instances, techniques are needed to aid in discovering patterns and to suggest potential relationships and hypotheses. A large number of such techniques exist, following the initial ideas of Tukey (1977) on EDA, which emphasize on the interaction between an individual and the data, by means of innovative graphics, summary displays, and other highly computational tools. EDA techniques such as boxplots, histograms, and scatterplot matrices are commonly used in studies involving spatial analyses. However, these techniques ignore special characteristics of spatial data like spatial dependence and spatial heterogeneity (Anselin, 1990).

ESDA (Anselin, 1994; Bivand, 2010; Symanzik, 2014), which is an extension of EDA, provides a set of robust tools for exploring spatial data, which do not require a knowledge of advanced statistics for their use. ESDA methods are used to detect spatial patterns of the data, formulate hypotheses based on the geography of the data, and assess spatial models.
To a considerable extent, based on the type of the spatial data set, one can determine which ESDA techniques are most appropriate to use. Many of the existing ESDA techniques are suitable for more than one type of spatial data.

To explore spatial data, we often display the data on a map and then produce several variations of the initial map (via different colors, symbols, and symbol sizes). In this dissertation, two broad classes of visualization techniques are of particular relevance: visualizing spatial distributions and visualizing spatial association. These methods have received attention in the literature during the last 25 years and have also been implemented in several software systems such as REGARD (Wills, 1992; Wills et al., 1991), SpaceStat (Anselin, 1992), SAGE (Haining et al., 1998), DynESDA (Anselin and Smirnov, 1999a,b) and ArcView/XGobi (Symanzik et al., 2000).

For areal/lattice data, several types of map displays exist and can be used for exploring and visualizing the spatial variation of a variable of interest. The most widely used visualization techniques are based on choropleth maps. These maps are thematic maps based on predefined aerial units. Here, each of the areas is colored (or shaded) according to a discrete scale based on the value of the attribute of interest within that area. One can think of it as a multi-colored checkerboard map. However, our choice of color in such a map can have a major effect on how we perceive the data. There are excellent color choice options for such maps and other statistical plots from the ColorBrewer software tool (Harrower and Brewer, 2003), accessible at http://colorbrewer2.org. An example of a choropleth map is shown in Figure 3.1. This figure shows the spatial variation of the proportion of non-white births in each county of North Carolina.

In the case of spatially continuous data, a general exploratory technique that is useful for gaining insight into the covariance structure and for visualizing the spatial association is the variogram cloud plot (Haslett et al., 1991). This is a plot of the dissimilarity between any two observations as a function of their separation in geographical space. The variogram cloud plot can be used in ESDA to identify data pairs that differ more than what is observed on average over the entire study area. If several “outlying” data pairs involve
the same observation, this indicates that this observation is substantially different from the neighboring ones, and may be a spatial outlier.

Figure 3.2 shows a variogram cloud plot of the scallops data (Kaluzny et al., 1998). In this plot, one can observe that the variogram cloud is extremely dense and shows many existing similar values across all distances. The variability at smaller distances appears lesser as compared to the variability at larger distances. A researcher working with the scallops data for example may want to take a closer look at these data points with a gamma value of about 40 and a distance between 0.1 and 0.5. This can be done via linked brushing. By connecting multiple visualizations through interactive linking and brushing would provide more information than considering the component visualizations independently.

The idea of linking and brushing is to combine different visualization methods to overcome the shortcomings of single techniques (Symanzik et al., 2000). Interactive changes made in one visualization are automatically reflected in the other visualizations. This method of interaction is used heavily by statistical tools such as GGobi (Cook and Swayne, 2007) and Mondrian (Theus and Urbanek, 2009). By linking, we mean showing how a point, or set of points, behaves in each of the plots. This is accomplished by highlighting these points in some fashion. For example, the highlighted points could be drawn as a filled circle.
Fig. 3.2: Squared-differences variogram cloud for the scallops data. (Previously published as Figure 4.6 in Kaluzny et al. (1998), page 77).

while the remaining points could be drawn as unfilled circles. A typical application of this would be to show how an outlier shows up in each of the individual plots. Brushing extends this concept a bit further. In brushing, the points to be highlighted are interactively selected by a mouse and the plots are dynamically updated (ideally in real time). That is, we can select a rectangular region of points in one plot and see how those points are reflected in the other plots. Overall, linked brushing is one of the most powerful interactive tools for doing exploratory data analysis using visualization. When we brush, i.e., highlight locations in a map view, we speak of geographic brushing (Monmonier, 1989). An example might be a three part display consisting of a map view showing data points and a histogram of zinc measurements together with a variogram cloud plot. Brushing and linking would allow the user to assign a color, green for instance, to two or more bars of the histogram corresponding to data points on the map view, thus causing the corresponding points in the variogram cloud plot to also be highlighted in green.
In R (R Core Team, 2012), there are two plotting systems: the traditional plotting system and the Trellis Graphics system. The Trellis Graphics system builds upon the grid graphics model (Murrell, 2006). Traditional graphics are typically built incrementally: graphic elements are added in several consecutive function calls. Trellis graphics allow plotting of high-dimensional data by providing conditioning plots: organized lattices of plots with shared axes (Cleveland, 1993, 1994). This feature is particularly useful when multiple maps need to be compared, for example in case of a spatial time series, comparison across a number of species or variables, or comparison of different modeling scenarios or approaches. The R lattice package (Sarkar, 2008) is particularly useful for visualizing spatial, spatio-temporal, and multivariate data sets. There are several R packages today that provide support for ESDA. These are discussed in detail in Bivand et al. (2008, pp. 192, pp. 195–200).

3.4 Development of the SWEVIS R Package for Visualizing SWE Data

An interactive visualization environment, in which the user may choose to display the data in many different ways, encourages data exploration. Here, our general approach in developing the new R package for the visualization of spatial and spatio-temporal data (e.g. SWE measurements) is to use as much of the capability of existing R packages as possible, thereby preserving conventions that many users are familiar with, and avoiding duplication of efforts. Thus, as part of this dissertation chapter, we considered a linkage between maps from RgoogleMaps (Loecher, 2012) and/or googleVis (Gesmann and de Castillo, 2012) and interactive statistical graphics from iPlots (Theus and Urbanek, 2003; Urbanek, 2013), which is already related to Mondrian (Theus and Urbanek, 2009). This linkage will support linking, brushing, and linked brushing in the statistical plots and the map views. The main idea is to create an interactive graphics package that will link multiple graphical displays and allow the user to brush (or highlight) subsets of observations in these linked displays. This new package will particularly rely on interactive graphics, lattice graphics, and spatial classes provided by the sp package where RgoogleMaps will serve as basis for the geographical visualization and exploration of our available data (see Figure 3.3). The
Fig. 3.3: A satellite image of Utah using the RgoogleMaps package in R. The orange shaded circles are the 10 selected SNOTEL sites used for our modeling efforts in Chapter 2. Further details of the locations of all 90 active SNOTEL sites in Utah can be seen in Figure 2.1.
RgoogleMaps package provides a comfortable R interface to query the Google server for static maps and it also uses the map as a background image to overlay plots within R which requires proper coordinate scaling. The iPlots package, written in Java, provides high interaction statistical graphics. A wide variety of plots is supported by this package, such as histograms, barcharts, scatter plots, boxplots, and parallel coordinates plots. All these plots support interactive features, such as querying, linked highlighting, color brushing, and interactive changing of parameters.

For our new SWEVIS R package, some of the functionalities for a single selected site on an RgoogleMap would include linkage to a plot showing forecasting/predicting SWE based on the statistical model described in Chapter 2. We will also consider histograms and boxplots of the available SWE data and other summary statistics for selected multiple sites (e.g., latitude, longitude, elevation, average temperature, minimum, maximum, and average SWE measurements, etc.). In the case of multiple sites selection via brushing, we would be able to generate a variogram cloud plot. By using this new package, one would be able to select sites on the map from RgoogleMap that corresponds to the points on the variogram cloud plot and vice-versa.

At the conceptual level, our goal is to create a visual data exploration system that is inductive and flexible, a system that provides the user with a variety of exploratory approaches including those mentioned in Section 3.3. At the operational level, these conceptual goals would be realized by a visualization environment that is dynamic and interactive. At the implementation level, dynamic and interactive system features would include real-time change, basic user navigation, and possibly color manipulation. This newly developed R package could also be applied to lattice data to produce image plots or heat maps.

In summary, the newly developed R package will provide the following features and plots not available in the underlying software:

- Spatial data manipulation and utilities: input of SWE data in a designed matrix format
- Forecasting: using the statistical model discussed in Chapter 2
• Mapping: maps from RgoogleMaps, heat maps, and image plots in a linked environment. A heat map is a graphical representation of data where the individual values contained in a matrix are represented as colors. Image plots are gridded colored or gray-scale rectangles with colors corresponding to the values in z (statistic or variable of interest). This can be used to display three-dimensional or spatial data

• EDA and ESDA: statistical graphics like histograms, boxplots, scatter plots, and variogram cloud plots linked to a map view

• Linked Brushing: connecting map displays from RgoogleMaps and EDA/ESDA graphics from iPlots

• Variogram cloud plot: one-to-two linking/brushing between statistical plots and map view

3.4.1 The SWEVIS R Package

The newly developed SWEVIS R package consists of 11 main functions. These functions are outlined as follows:

• ReadSweData: This function is responsible for reading, storing, and manipulating SWE data in text file format. This function stores the observed SWE measurements for the SNOTEL sites and necessary additional information (like elevation, site name, site number, county, Station ID, year of historical data, latitude, longitude, etc.) in a data frame.

• CalcSweSumStat: This function is responsible for calculating and storing summary statistics for daily minimum (minDay), daily maximum (maxDay) and daily average (aveDay) SWE measurement in a water-year for the past T water-years. In addition, we have minimum (minPast), maximum (maxPast) and average (avePast) SWE measurements over all the past T water-years. As a reminder, a water-year is from October 1 to September 30 (see Section 2.2.1 for more details). It also combines the
obtained minPast, maxPast, and avePast with the data stored via the `ReadSweData` function into another data frame to be used later.

- **SimSweMCMCData**: This function is responsible for performing singular value decomposition (SVD) on the data matrix discussed in equation (2.1). It also computes and saves the sampling estimates of the model parameters $\alpha, \alpha_{st}$, and $\Sigma_y$ in Section 2.2.3 for a specified number of simulations.

- **SwePlotData**: This function is responsible for manipulating and storing data for visualizing the SWE data. It uses the data stored via the `ReadSweData` function. All of the following functions inherit from the `SwePlotData`.

- **SweRawDataPlot, SweHistPlot, SweBoxPlot, SweVariogPlot, SweRgoogleMap**: These five functions are for visualizing single-site and multiple-sites SWE measurements. `SweRawDataPlot` produces histogram and boxplots for the raw SWE measurements for T water years. `SweHistPlot` produces a histogram of summary statistics like min-max and average SWE measurements over time for both single and multiple sites (Figure 3.4); `SweBoxPlot` produces boxplots of summary statistics (min-max and average) of SWE measurements over time for both single and multiple sites (Figure 3.5); `SweVariogPlot` produces a variogram cloud plot of SWE measurements for multiple sites (Figure 3.6). This function has a maximum of six selected timepoints (which can be changed). For a specified water year, these timepoints are: December 8 (day 68), January 8 (day 100), February 9 (day 132), March 12 (day 164), April 13 (day 196), and May 15 (day 228). And, finally, `SweRgoogleMap` creates maps using the RgoogleMaps package and stores them as a png file in the current working directory to be used later by `iSwePlot` (Figure 3.3).

- **SwePostPlot**: This is the function used to produce posterior predictive plots based on the model described in Chapter 2 (Figure 2.8). It uses the data stored by the `SimSweMCMCData` function.
Fig. 3.4: Histograms of daily minimums, daily maximums, and daily averages of SWE measurements for the Farmington SNOTEL site for 30 water years. It is not intended to directly compare these plots; thus, no common scale has been used.

Fig. 3.5: Boxplots of daily minimums, daily maximums, and daily averages of SWE measurements for the Farmington SNOTEL site for 30 water years. Again, it is not intended to directly compare these plots.
Fig. 3.6: Variogram cloud plot of SWE for the ten selected SNOTEL sites discussed in Chapter 2 on December 8, 2009 (day 68 in the water year).

- **iSwePlot**: This function is responsible for creating an interactive environment that connects map displays from RgoogleMaps and EDA/ESDA graphics from iPlots, i.e., one-to-two linking/brushing between statistical plots and map view (Figure 3.7). It uses the data stored by the ReadSweData and SwePlotData functions.

For purposes of linked brushing in the created interactive environment, the following functions are used.

- **iSweBrushMapSingle**: This function is used to select or highlight a point (location of SNOTEL site) on a map view to produce six main windows of corresponding statistical plots for further exploration of the data.

- **iSweBrushMap**: This function is used to select or highlight at least two points (location of SNOTEL sites) on a map view which highlights corresponding points in the statistical plots.
Fig. 3.7: Snapshot of the created interactive environment (using *iSwePlot*) connecting the map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2, the variogram cloud plot of SWE (Figure 3.6), and plot of the selected SNOTEL site numbers.

- *iSweBrushPlot*: This function is used to select or highlight at least one point in any of the statistical plots which highlights corresponding points (location of SNOTEL sites) in the map view.

The newly developed SWEVIS R package could also be used to visualize SWE measurements that are saved in *ascii* format. The following three functions can be used to achieve this goal.

- *ReadSweAsciiData*: This function is responsible for reading, storing, and manipulating SWE data in ascii file format. This function stores the observed SWE measurements (for specified dates) and elevations for the SNOTEL stations in a data frame.

- *SweAsciiImagePlot*: This function produces an image plot of the SWE measurements for specific dates. It uses the data stored via the *ReadSweAsciiData* function.
• *iSweAsciiPlot*: This function is responsible for creating an interactive environment that connects the image plot and EDA/ESDA graphics from iPlots, i.e., one-to-two linking/brushing between statistical plots and map view (Figure 3.7). It uses the data stored by the *ReadSweAsciiData* function and any necessary data (e.g., elevation) for the SNOTEL station.

The SWEVIS package developed requires the following R packages: MCMCpack (Martin et al., 2011), mvtnorm (Genz and Bretz, 2009; Genz et al., 2011), msm (Jackson, 2011), lattice (Sarkar, 2008), Mass (Venables and Ripley, 2002), spdep (Bivand et al., 2008; Pebesma and Bivand, 2005), geoR (Ribeiro Jr. and Diggle, 2001), maps (Becker et al., 2013), and iPlots (Urbanek, 2013). In addition to these R packages, other functionalities were created to make the SWEVIS R package a success: “read.snotel.R”, “krig.swe.R”, “utah.snotel.historical.data.R”, “utah.snotel.current.data.R”, “invgammastrt.R”, “swe.svd.mcmc.R”, and “plot.swe.post.R”.

For each SNOTEL site, the *ReadSweData* functionality uses the “read.Snotel.R” function to clean up the historical and current water-year daily SWE measurements by reading the downloaded SNOTEL data in text format one year at a time and output vectors of values with NAs in place of blanks and “- - -”. The “utah.snotel.current.data.R” and “utah.snotel.historical.data.R” functions use the “read.Snotel.R” function to put the already downloaded historical and current text files of SWE measurements for the ninety active Utah SNOTEL sites in a matrix form. The function “krig.swe.R” augments the historical SWE data for SNOTEL sites with missing values using a Kriging approach. The “swe.svd.mcmc.R” function which forms part of the *SimSweMCMCData* function, performs the SVD on the data matrix discussed in equation (2.1), computes, and saves the sampling estimates of the model parameters $\alpha, \alpha_{st}$, and $\Sigma_y$ in Section 2.2.3 for a specified number of simulations. It also produces samples of the posterior distributions for the unobserved SWE after a chosen day $d$ in our forecasting of SWE. In addition, the “invgammastrt.R” function computes the initial values of the shape and scale parameters of the inverse gamma distribution for $\sigma^2_y$ (discussed in Section 2.2.3). The function “plot.swe.post.R” in *SwePost-
Plot, uses output from the “swe.svd.mcmc.R” from SimSweMCMCDData to enable us to reproduce Figures 2.8, 2.9, and 2.10.
CHAPTER 4
APPLICATION OF THE SWEVIS R PACKAGE USING UTAH SNOTEL SITES AND THE UPPER SHEEP CREEK SITE AS CASE STUDIES

4.1 Introduction

Most likely, many end users of the R package discussed in Chapter 3 are from environmental agencies, or individuals interested in the daily amount of snow measurements. Those users are unlikely to know how to deal with computer code written in R. These users want to focus on the results and graphics, and not on running computer code. In this chapter, we present two case studies that make use of the functions from our newly developed SWEVIS R package from Chapter 3.

Thus, this chapter serves as a short tutorial for using the SWEVIS R package directly from R. It is divided into three parts: (1) single snow telemetry (SNOTEL) site snow water equivalent (SWE) visualizations (Section 4.3), (2) multiple SNOTEL sites SWE visualizations (Section 4.4), and (3) Upper Sheep Creek (USC) SWE visualizations (Section 4.5).

4.2 Data

To demonstrate our newly developed SWEVIS R package, we use SWE data from (i) the SNOTEL sites in Utah and (ii) the USC Watershed in Idaho.

4.2.1 Utah SNOTEL Data

In Section 2.2.1, some description of the Utah SNOTEL data was provided (also see Figure 2.1). The basic SNOTEL station provides SWE data via a pressure-sensing snow pillow. It also collects data on snow depth, all-season precipitation accumulation, and air temperature with daily maximums, minimums, and averages. Many of the enhanced SNOTEL stations are also equipped to take soil moisture and temperature measurements at various depths. The atmospheric and, where installed, soil moisture and soil temperature
measurements are generally reported multiple times per day with some stations reporting hourly. Other sensors, such as water quality sensors, can be added to any of the enhanced SNOTEL stations.

4.2.2 Upper Sheep Creek Watershed Data

The Upper Sheep Creek Watershed is a 26-ha semi-arid mountainous sub-basin located within the Reynolds Creek Experimental Watershed in the Owyhee Mountains of southwest Idaho (see Figure 4.1). In 1984, a detailed study of the Upper Sheep Creek Watershed was initiated by the USDA-ARS Northwest Watershed Research Center and expanded in 1989 (Flerchinger and Cooley, 2000). Locations on the watershed are referenced by a grid system as illustrated in Figure 4.2.

Three distinct vegetation types can be identified in the Upper Sheep Creek Watershed: low sagebrush, mountain big sagebrush, and aspen (see Figure 4.3). Low sagebrush areas are located predominantly on the west-facing slopes and are bare of snow for much of the winter. North-facing slopes are covered with mainly mountain big sagebrush and, during the winter, typically accumulate about a meter of snow. Aspen thickets are established on the upper portions of the north-facing slopes where large snow drifts form annually (Flerchinger and Cooley, 2000). The Upper Sheep Creek Watershed has more heterogeneity (Luce and Tarboton, 2004) in snow accumulation than one might expect at this scale due to drifting (Luce and Tarboton, 2004) caused by strong winds (see Figure 4.4).

The primary source of runoff from the basin is spring snowmelt and it provides the driving hydrologic force for runoff and sub-surface flow. Nearly all water reaching the stream is sub-surface flow; overland flow is seldom observed in the basin. The geology of the Upper Sheep Creek consists of variably fractured and altered basalt underlain by a thick dense basalt at a depth of 20 to 30 m. Geophysical studies of the area show that the surface of the dense basalt closely follows surface topography (Mock, 1988; Stevens, 1991; Winkelmaier, 1987) so that the watershed boundary for the ground water flow is approximately the same as for the surface. In addition, climatic information such as wind speed, temperature, humidity, solar radiation, and precipitation were monitored within the
Fig. 4.1: The red shaded square shows the approximate location of the Upper Sheep Creek Watershed in Idaho in Owyhee County. The dashed lines represent county boundaries.
Fig. 4.2: Topography and instrument locations within the Upper Sheep Creek Watershed (Previously published as Figure 1 in Flerchinger and Cooley (2000), page 87).

Fig. 4.3: Disaggregation of the Upper Sheep Creek Watershed into three zones based on vegetation and snow accumulation. The low sagebrush, mountain big sagebrush, and aspen zones comprise 58.9%, 26.6% and 14.5% of the watershed, respectively (Previously published as Figure 2 in Flerchinger and Cooley (2000), page 89).
Fig. 4.4: Snow water equivalent measured in Upper Sheep Creek March 3, 1993. The boundary represents the Upper Sheep Creek watershed. The dots represent locations of the grid stations where measurements of snow water equivalent (SWE) were taken. No grid stations are available at points 9N and 25D and point L10 was not measured (Previously published as Figure 2 in Luce and Tarboton (2004), page 1411).

The SWE were measured on a grid of 255 stations within Upper Sheep Creek (see Figure 4.4). These measurements were taken several times each year over the course of nine years (Cooley, 1988). For our newly developed SWEVIS R package described in Chapter 3, some of the functions include the display of plots of SWE measurements on different dates in a specified year and plots for the difference in SWE measurements for two consecutive dates at the USC.

4.3 Single SNOTEL Site SWE Visualization

In order to read and store SWE data for each SNOTEL site, the user needs to first specify the path to the SWE data files and source the SWEVIS functions. This is done using the following R code:

```r
# Specify path of the SWE Data files
setwd("path/goes/here")
```
Read and store SWE measurements for Utah SNOTEL sites for a specified current water year (e.g., 2009, for demonstration purposes)

```r
swedata <- ReadSweData(n.sites = 90, current.year = 2009, loc = "UtahSnotelSites.csv")
```

- `n.sites` = the total number of SNOTEL sites
- `current.year` = specified current water-year
- `loc` = additional information about each site (like elevation, site name, latitude, longitude, etc.) in a csv file

The `swedata` is a data frame with components including:
- Total number of SNOTEL sites
- Number of water-years before the current water-year
- Number of days in a water-year
- Names of SNOTEL sites
- Site number
- Station IDs
- Year for historical SWE data measurements
- SNOTEL site counties
- Elevation (in feet)
- Latitude
- Longitude
- SWE measurements for each site

To read the value of the components in `swedata`, a user can use the `$` notation.

The following R code will print the number of years (T) for past SWE measurements, number of days (m) in the water-year, the number of SNOTEL sites (n.sites) data stored, and the first six values of SWE data for the first fourteen water-years for a single site, say Horse Ridge (#42).

```r
swedata$T
[1] 30

swedata$m
[1] 365

swedata$n.sites
[1] 90

head(sweedatatutah.list[[42]][, 1:13])
[1,] 0 0 0 0 1.9 0.0 0 0 2.4 0 0 0 0
[2,] 0 0 0 0 2.2 0.0 0 0 2.4 0 0 0 0
[3,] 0 0 0 0 2.2 0.0 0 0 2.9 0 0 0 0
[4,] 0 0 0 0 1.9 0.2 0 0 2.9 0 0 0 0
[5,] 0 0 0 0 1.6 0.0 0 0 2.9 0 0 0 0
[6,] 0 0 0 0 1.6 0.0 0 0 2.7 0 0 0 0
```

When SWE measurements for all SNOTEL sites have been loaded via the `SweData` function, we can obtain summary statistics and additional information of a selected single site.
using the `CalSweSumStat` function. For example, if we select site #42 we obtain the following results:

```r
swesumstat <- CalSweSumStat(num = 42, dat = swedata)

# dat = data to be used
# num = a vector of values representing selected SNOTEL sites for
#       single or multiple visualization purposes
```

```
swesumstat$sweinfo
```

<table>
<thead>
<tr>
<th>Number</th>
<th>42</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNOTEL.Site</td>
<td>HORSE RIDGE</td>
</tr>
<tr>
<td>State</td>
<td>Utah</td>
</tr>
<tr>
<td>Site.Number</td>
<td>533</td>
</tr>
<tr>
<td>Station.ID</td>
<td>11h21s</td>
</tr>
<tr>
<td>County</td>
<td>Morgan</td>
</tr>
<tr>
<td>Longitude</td>
<td>-111.4333</td>
</tr>
<tr>
<td>Latitude</td>
<td>41.3</td>
</tr>
<tr>
<td>Elevation</td>
<td>8160</td>
</tr>
<tr>
<td>Historical.Data.Year</td>
<td>1979</td>
</tr>
<tr>
<td>Past.min.SWE</td>
<td>0</td>
</tr>
<tr>
<td>Past.max.SWE</td>
<td>40.2</td>
</tr>
<tr>
<td>Past.ave.SWE</td>
<td>7.052779</td>
</tr>
</tbody>
</table>

In order to use the visualizations from the SWEVIS package, the user needs to set up data for the graphs. The `SwePlotData` function is used to achieve this objective. This generates data to be used for the histograms, boxplots, forecasting SWE, and variogram cloud plots (in the case of multiple sites). The following R code generates the required data for these plots.

```r
# Generate and store data for the plot functions
sweplotdata <- SwePlotData(dat = swedata, n.sites = 90,
                           num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66))

# dat = data to be used
# n.sites = the number of all SNOTEL sites
# num = a vector of values representing selected SNOTEL sites for
#       single and multiple visualization purposes
```

Using the data generated by the `SwePlotData` function, we can create the histograms and boxplots of the minimum, maximum, and average SWE for each day in the water-year for the past say $T = 30$ years (in the case of 2009 being our current water-year). These plots
can be created using the `SweHistPlot` and `SweBoxPlot` functions. In addition, we can graph histograms and boxplots of observed SWE measurements for the past say $T = 30$ years for the selected site using the `RawSweDataPlot` function. The following chunk of R code creates these plots. The results for site #42 (Horse Ridge) can be seen in Figures 4.5, 4.6, and 4.7.

```r
# Show the histogram of a single site daily minimums, daily maximums, # and daily averages of SWE over $T = 30$ water years for site #42
SweHistPlot(id = 42, sig = TRUE, dat = sweplotdata, col = "light grey",
            cex.lab = 1.5, cex.axis = 1.5, main = "")

# id = a vector of values representing the SNOTEL sites
# sig = logical. If TRUE, provides plots for single SNOTEL sites
# dat = data to be used
# col = a color to be used to fill the bars
# cex.lab = the magnification to be used for x and y labels
# cex.axis = the magnification to be used for axis annotation
# main = this augments to the main title. By default, there
#        is no main title

# Show the boxplot of a single site daily minimums, daily maximums, # and daily averages of SWE over $T = 30$ water years for site #42
SweBoxPlot(id = 42, sig = TRUE, dat = sweplotdata, col = "light grey",
           cex.lab = 1.5, cex.axis = 1.6, boxlwd = 1.5, main = "")

# id = a vector of values representing the SNOTEL sites
# sig = logical. If TRUE, provides plots for single SNOTEL sites
# dat = data to be used
# col = a color to be used to fill the bars
# cex.lab = the magnification to be used for x and y labels
# cex.axis = the magnification to be used for axis annotation
# boxlwd = the box line width
# main = this augments to the main title. By default, there
#        is no main title

# Show the histogram and box plot of a single site raw SWE measurements # over $T = 30$ water years for site #42
SweRawDataPlot(id = 42, sig = TRUE, dat = swedata, col = "light grey",
               cex.lab = 1.5, cex.axis = 1.5, boxlwd = 1.5, main = "")

# id = a vector of values representing the SNOTEL sites
# sig = logical. If TRUE, provides plots for single SNOTEL sites
# dat = data to be used
# col = a color to be used to fill the bars
Fig. 4.5: Histograms of the minimum, maximum, and average of observed SWE measurements for each day of the 365 days in the water-year over the past 30 years for the Horse Ridge SNOTEL site. It is not intended to directly compare these plots; thus, no common scale has been used.

Fig. 4.6: Boxplots of the minimum, maximum, and average of observed SWE measurements for each day of the 365 days in the water-year over the past 30 years for the Horse Ridge SNOTEL site. Again, it is not intended to directly compare these plots; thus, no common scale has been used.
Fig. 4.7: Histogram and boxplot of the observed SWE measurements for the Horse Ridge SNOTEL site. The histogram shows the observed SWE measurements for each day of the 365 days in the 2009 water-year. The boxplot shows comparable plots for the observed SWE measurements for each day of the 365 days (in a water-year) for the previous 30 water-years (1979 – 2008). The two highest amounts of SWE were recorded in the water-years 6 (1984) and 19 (1997).

A goal of this dissertation was to produce posterior predictive plots based on the model discussed in Chapter 2. These posterior predictive plots are plots showing forecasting of future SWE measurements after a specified initiation date based on the posterior predictive distributions obtained in Chapter 2 (see Section 2.2.3 and Appendix B for further information). For our selected SNOTEL site, we perform singular value decomposition (SVD) on the data matrix discussed in equation (2.1). For 1000 simulations, sampling estimates of the model parameters $\alpha$, $\alpha_{st}$, and $\Sigma_y$ discussed in Section 2.2.3 are then computed and saved via the $\text{SweMCMCData}$ function. The $\text{SwePostPlot}$ function is then used to produce the posterior predictive plot for the selected SNOTEL site. The following R code is used to obtain this result (see Figure 4.8).

```r
# Computes and stores sampling estimates of the model parameters
# for a specified number of simulations for a selected SNOTEL site
# (e.g., #42)
swemcmcdata <- SimSweMCMCData(id = 42, dat = swedata, q = 4,
               n.mcmc = 1000, mm = 100, outdat = 100,
               intdate = "Jan.8", current.year = 2009)
```
Fig. 4.8: Observed SWEs and posterior predictions for the 2009 water-year for the Horse Ridge SNOTEL site based on the initiation date January 8. See Figure 2.8 for additional details.

```r
# id = a vector of values representing the SNOTEL sites
# dat = data to be used
# q = number of important signals chosen for dimension reduction as
#     discussed in Chapter 2
# n.mcmc = number of MCMC simulations to be done
# mm = the number of days of observed data in the current water-year
# outdat = the day after which forecasting is done based on the
#     Posterior Distribution
# intdate = initiation date (date corresponding to the outdat)
# current.year = specified current water-year

# Show the posterior predictive plot for site #42
SwePostPlot(dat = swemcmcmdata, sig = TRUE, intdate = "Jan.8",
             current.year = 2009, labnum = 1, date = "Jan. 8, 2009 --")

# dat = data to be used
# sig = logical. If TRUE, provides plots for single SNOTEL sites
# intdate = initiation data (date corresponding to the outdat)
# current.year = specified current water-year
# labnum = number assigned to a specified initiation date
# date = date to be used in the legend for single plot
#        (i.e. when sig = TRUE)
```
The *iSwePlot* function of the SWEVIS package introduced in Chapter 3 (Section 3.4), can be used to view and interactively query the SWEs. This function provides several capabilities, including linked brushing between the RgoogleMap view window and interactive statistical graphics from iPlots. For a single SNOTEL site, we can do linked brush between the RgoogleMap view window and the SWE boxplots, histograms, and posterior predictive plot based on the model discussed in Chapter 2.

An example of an analysis using linked brushing is shown in Figure 4.9. This figure shows the RgoogleMaps map view for the state of Utah with the ten selected SNOTEL sites discussed in Chapter 2, and six statistical plots in an interactive environment. Figure 4.9 also gives an example of the brushing capabilities of the *iSwePlot* function. A selected (highlighted) point on the map view produces the six corresponding statistical plots. This selected point in the map view, a large filled red circle, indicates the location of a SNOTEL site. Among these statistical plots is a window for three boxplots of the non-brushed data (in white color). When brushing is performed, three additional boxplots (in red color) are created for the brushed data. By moving the brush up and down in any of the linked statistical plots, various quantiles of the data can be explored. In this case, a horizontal rectangular shaped brush has been used to brush approximately the highest 15% of the SWE values in the posterior predictive plot. These points on the posterior plot are shown as large red filled circles (appear as a solid red line due to overplotting), indicating days in the water-year containing high values of SWE measurements. Alternatively, a vertical rectangular shaped brush could be used to brush a specific range of values of observed SWE in the histogram. In order to obtain the plots in Figure 4.9 in an interactive environment, the following chunk of R code is used.

```r
## To create the map view needed in the interactive environment
iSwePlot(dat1 = swedata, dat2 = sweplotdata,
        num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66),
        x1 = -114.2, y1 = 37, x2 = -108.9, y2 = 42.1,
        destfile = "UTAH.png", maptype = "hybrid")

# dat1 = 1st data to be used
# dat2 = 2nd data to be used
```
# num = a vector of values representing selected SNOTEL sites for
# single and multiple visualization purposes
# x1 = lower limit of the Longitude coordinates
# x2 = upper limit of the Longitude coordinates
# y1 = lower limit of the Latitude coordinates
# y2 = upper limit of the Latitude coordinate
# destfile = the file to load the map image from or save to
# maptype = defines the type of map from the Google server to construct

## Brushing a single SNOTEL site on the map view to produce the plots
## discussed

tSweBrushMapSingle(i = 1, dat1 = swedata, dat2 = sweplotdata,
num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66),
q = 4, n.mcmc = 1000, mm = 100, outdat = 100,
intdate = "Jan.8", current.year = 2009,
date = "Jan. 8, 2009")

# i = iset selected for brushing in iPlot (map with the RgoogleMap
# as background)
# dat1 = 1st dataset to be used
# dat2 = 2nd dataset to be used
# num = a vector of values representing selected SNOTEL sites for
# single and multiple visualization purposes
# q = number of important signals chosen for dimension reduction as
# discussed in Chapter 2
# n.mcmc = number of MCMC simulations to be done
# mm = the number of days of observed data in the current water-year
# outdat = the day after which forecasting is done based on the
# Posterior Distribution
# intdate = initiation date (date corresponding to the outdat)
# current.year = specified current water-year
# date = date to be used in the legend for single SWE posterior plot

4.4 Multiple SNOTEL Sites SWE Visualization

In Section 4.3, we demonstrated how to read and store the raw SWE data for all
SNOTEL sites. In order to obtain summary statistics and additional information of selected
multiple sites using the SweSumStat function, we make a slight modification to the line of
R code used for a single SNOTEL site. To visualize, say, four SNOTEL sites at the same
time, we simply provide a vector of length 4 (instead of length 1) as the num argument
in the SweSumStat function. For example, if we select SNOTEL sites Ben Lomond Peak
(#4), Little Bear (#53), Monte Cristo (#64), and Timpanogos Divide (#82), we obtain
the following results:
Fig. 4.9: Snapshot of the created interactive environment (using iSwePlot) connecting the map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2 and the six statistical plots. Here the Horse Ridge is selected on the map view, yielding the corresponding histograms, boxplots, and posterior predictive plot of its SWE measurements. The approximate horizontally brushed highest 15% of the SWE values in the posterior predictive plot is displayed immediately in all the other statistical plots. The SWE boxplot window shows 6 boxplots: 3 for the brushed data (in red color) and 3 for the non-brushed data (in white color).
swesumstat <- CalSweSumStat(num = c(4, 53, 64, 82), dat = swedata)
swesumstat$sweinfo

<table>
<thead>
<tr>
<th>Number</th>
<th>SNOTEL.Site</th>
<th>4</th>
<th>53</th>
<th>64</th>
<th>82</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BEN LOMOND PEAK</td>
<td>LITTLE BEAR</td>
<td>MONTE CRISTO</td>
<td>TIMPANOGOS DIVIDE</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>Utah</td>
<td>Utah</td>
<td>Utah</td>
<td>Utah</td>
<td></td>
</tr>
<tr>
<td>Site.Number</td>
<td>332</td>
<td>582</td>
<td>634</td>
<td>820</td>
<td></td>
</tr>
<tr>
<td>Station.ID</td>
<td>11h08s</td>
<td>11h25s</td>
<td>11h57s</td>
<td>11j21s</td>
<td></td>
</tr>
<tr>
<td>County</td>
<td>Weber</td>
<td>Cache</td>
<td>Weber</td>
<td>Wasatch</td>
<td></td>
</tr>
<tr>
<td>Longitude</td>
<td>-111.9333</td>
<td>-111.8167</td>
<td>-111.4833</td>
<td>-111.6000</td>
<td></td>
</tr>
<tr>
<td>Latitude</td>
<td>41.36667</td>
<td>41.40000</td>
<td>41.45000</td>
<td>40.41667</td>
<td></td>
</tr>
<tr>
<td>Elevation</td>
<td>8000</td>
<td>6544</td>
<td>8960</td>
<td>8140</td>
<td></td>
</tr>
<tr>
<td>Past.min.SWE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Past.max.SWE</td>
<td>74.0</td>
<td>25.5</td>
<td>49.0</td>
<td>43.0</td>
<td></td>
</tr>
<tr>
<td>Past.ave.SWE</td>
<td>13.566637</td>
<td>3.237525</td>
<td>9.997084</td>
<td>7.239470</td>
<td></td>
</tr>
</tbody>
</table>

As we demonstrated in Section 4.3, to visualize SNOTEL SWE data, we first have to use the `SwePlotData` function to generate data for the plots based on the raw SWE measurements stored in R via the `ReadSweData` function for the selected SNOTEL sites. When this is done, we can visualize SWE data from two or more SNOTEL sites at the same time. To visualize together SWE data from the SNOTEL sites Ben Lomond Peak (#4), Little Bear (#53), Monte Cristo (#64), and Tinpanogos Divide (#82), we provide a vector of length 4 (instead of length 1) as the `num` argument in the `SwePlotData` function and as the `id` argument in the `SweHistPlot`, `SweBoxPlot`, and `RawSweDataPlot` functions. The results, using the following chunk of R code, can be seen in Figures 4.10, 4.11, and 4.12.

# Generate and store data for the plot functions for selected SNOTEL sites
sweplotdata <- SwePlotData(dat = swedata, n.sites = 90,
                           num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66))

# dat = data to be used
# n.sites = the number of all SNOTEL sites
# num = a vector of values representing selected SNOTEL sites
# for single and multiple visualization purposes

# Show the boxplots of a multiple sites daily minimums, daily maximums,
# and daily averages of SWE over T = 30 water years for sites #4, #53,
# #64, and #82
SweBoxPlot(id = c(4, 53, 64, 82), sig = FALSE, dat = sweplotdata, common.scale = TRUE, col = "light grey", ylim = c(0, 75), cex.lab = 1.5, cex.axis = 1.6, boxlwd = 1.5, main = "")
Show the histograms of multiple sites daily minimums, daily maximums, and daily averages of SWE over T = 30 water years for sites #4, #53, #64, and #82

SweHistPlot(id = c(4, 53, 64, 82), sig = FALSE, dat = sweplotdata, main = "", common.scale = TRUE, col = "light grey", xlim1 = c(0, 20), xlim2 = c(0, 80), xlim3 = c(0, 40), ylim1 = c(0, 300), ylim2 = c(0, 200), ylim3 = c(0, 220), cex.lab = 1.5, cex.axis = 1.5, main = "")

Show the histogram and box plot of a single site raw SWE measurements over T = 30 water years for site #42

SweRawDataPlot(id = c(4, 53, 64, 82), sig = FALSE, dat = swedata, main = "", common.scale = TRUE, col = "light grey", xlim1 = c(0, 60), ylim1 = c(0, 150), ylim2 = c(0, 80), cex.lab = 1.5, cex.axis = 1.5, boxlwd = 1.5, main = "")
Fig. 4.10: Comparing boxplots of daily minimums, daily maximums, and daily averages of SWE measurements over $T = 30$ years (with 2009 as the current water-year) for the four selected SNOTEL sites. Rows 1 (top) through 2 (bottom) show plots for the Ben Lomond Peak (top left), Little Bear (top right), Monte Cristo (bottom left), and Timpanogos Divide (bottom right) SNOTEL sites, respectively. The SWE daily minimums, daily maximums, and daily averages are obtained by finding the minimum, maximum, and average SWE for each day of the 365 days in the water-year over the past $T = 30$ years (with 2009 as the current water-year).

In order to compare posterior predictive plots based on the model discussed in Chapter 2 for our selected SNOTEL sites, we perform singular value decomposition (SVD) on the data matrix discussed in equation (2.1) for each site. Again, for 1000 simulations, sampling estimates of the model parameters $\alpha, \alpha_st,$ and $\Sigma_y$ in Section 2.2.3 are then computed and saved for each site via the $SweMCMCData$ function. This is done using the $SweMCMCData$
Fig. 4.11: Comparing histograms of daily minimums, daily maximums, and daily averages of SWE measurements over $T = 30$ years (with 2009 as the current water-year) for the four selected SNOTEL sites. Rows 1 (top) through 4 (bottom) show plots for the Ben Lomond Peak, Little Bear, Monte Cristo, and Timpanogos Divide SNOTEL sites, respectively. The SWE daily minimums, daily maximums, and daily averages are obtained by finding the minimum, maximum, and average SWE for each day of the 365 days in the water-year over the past $T = 30$ years (with 2009 as the current water-year).
Fig. 4.12: Comparing histograms and boxplots of the observed SWE measurements for four selected SNOTEL sites. The histograms show observed SWE measurements for each day of the 365 days in the 2009 water-year. The boxplots shows comparable plots for the observed SWE measurements for each day of the 365 days (in a water-year) for the previous 30 water-years. Rows 1 (top) through 4 (bottom) show plots for the Ben Lomond Peak, Little Bear, Monte Cristo, and Timpanogos Divide SNOTEL sites, respectively.
functionality. We then use the `SwePostPlot` function again to produce comparable posterior predictive plots for the selected SNOTEL sites. The following lines of R code are used to obtain this result (see Figure 4.13).

```r
# Computing and storing sampling estimates of the model parameters
# for a specified number of simulations for each selected SNOTEL site
swemcmcdata1 <- SimSweMCMCData(id = 4, dat = swedata, q = 4,
n.mcmc = 1000, mm = 100, outdat = 100,
intdate = "Jan.8", current.year = 2009)
swemcmcdata2 <- SimSweMCMCData(id = 53, dat = swedata, q = 4,
n.mcmc = 1000, mm = 100, outdat = 100,
intdate = "Jan.8", current.year = 2009)
swemcmcdata3 <- SimSweMCMCData(id = 64, dat = swedata, q = 4,
n.mcmc = 1000, mm = 100, outdat = 100,
intdate = "Jan.8", current.year = 2009)
swemcmcdata4 <- SimSweMCMCData(id = 82, dat = swedata, q = 4,
```
n.mcmc = 1000, mm = 100, outdat = 100, intdate = "Jan.8", current.year = 2009)

# id = a vector of values representing the selected SNOTEL sites
# dat = data to be used
# q = number of important signals chosen for dimension reduction as discussed in Chapter 2
# n.mcmc = number of MCMC simulations to be done
# mm = the number of days of observed data in the current water-year
# outdat = the day after which forecasting is done based on the Posterior Distribution
# intdate = initiation date (date corresponding to the outdat)
# current.year = specified current water-year

# Show the posterior predictive plots for multiple sites selected
SwePostPlot(dat = list(swemcmcdatal, swemcmcdatb, swemcmcdatc, swemcmcdatd),
              sig = FALSE, intdate = "Jan.8", current.year = 2009,
              labnum = 1, date = "Jan. 8, 2009 --")

# dat = data to be used for the multiple sites
# sig = logical. If FALSE, provides plots for multiple SNOTEL sites
# intdate = initiation date (date corresponding to the outdat)
# current.year = specified current water-year
# labnum = number assigned to a specified initiation date
# date = date to be used in the legend for single plot
# (i.e. when sig = TRUE)

As introduced in Chapter 3, for multiple sites, one can create an interactive environment that connects map displays from RgoogleMaps and EDA/ESDA graphics from iPlots, i.e., one-to-two linking/brushing between statistical plots and a map view. For a single data point highlighted in the variogram cloud plot, with a gamma value of about 6.2 and a distance between 30 and 40 miles, we can determine the pair of SNOTEL sites on the map view that produced it (see Figure 4.14). In Figure 4.15, when two data points are selected/highlighted in the variogram cloud plot, three SNOTEL sites are also highlighted on the map view. Here, one site is affected by two locations, as seen in the scatterplot of SNOTEL sites. Similarly, in Figure 4.16, when three data points are selected in the variogram cloud plot, these correspond to five highlighted SNOTEL sites in the map view. In Figures 3.7 and 4.17, one can observe that, when six SNOTEL sites are selected in the map view, this selection highlights \( \binom{6}{2} = 15 \) data points in the variogram cloud plot (bottom
Fig. 4.14: Snapshot of the created interactive environment (using \textit{iSwePlot}) connecting a map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2, the variogram cloud plot (Figure 3.6), and a plot of the selected SNOTEL site numbers. Here, in the variogram cloud plot, the vertical and horizontal axes represent variogram estimates (gamma) and pairwise distance (in miles) respectively. In the plot of the selected SNOTEL site numbers, the axes represent numbers corresponding to the SNOTEL sites in the map view. One data point in the variogram cloud plot (bottom right) is selected, highlighting a data point in the plot of the SNOTEL site numbers (top right) and also, a corresponding pair of SNOTEL sites in the map view.

The following chunk of R code enables us to produce an interactive environment that connects map displays from RgoogleMaps for the State of Utah with ten selected SNOTEL sites and a variogram cloud plot based on the observed SWE measurements and vice versa.

```r
# Obtaining RgoogleMap and save it as a png file in the current working directory to be used as a map view background later for the iSweplot functionality
SweRgoogleMap(lat = c(37, 42), lon = c(-114.2, -109), destfile = "UTAH.png", maptype = "hybrid")
```
Fig. 4.15: Snapshot of the created interactive environment (using `iSwePlot`) connecting a map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2, the variogram cloud plot (Figure 3.6), and a plot of the selected SNOTEL site numbers. Here, in the variogram cloud plot, the vertical and horizontal axes represent variogram estimates (gamma) and pairwise distance (in miles) respectively. In the plot of the selected SNOTEL site numbers, the axes represent numbers corresponding to the SNOTEL sites in the map view. Two data points in the variogram cloud plot (bottom right) are selected, highlighting two data points in the plot of the SNOTEL site numbers (top right) and also, the corresponding three SNOTEL sites in the map view.

```r
# Creating an interactive environment that connects map displays from RgoogleMaps and EDA/ESDA graphics from iPlots
iSwePlot(dat1 = swedata, dat2 = sweplotdata, destfile = "UTAH.png",
          datenum = 1, num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66),
          x1 = -114.2, y1 = 37, x2 = -108.9, y2 = 42.1, maptype = "hybrid")
```

# lat = latitude range of the map view needed
# lon = longitude range for the map view needed
# destfile = stored file name (in png format) of the map view
# maptype = defines the type of map from the Google server to construct
Fig. 4.16: Snapshot of the created interactive environment (using iSwePlot) connecting a map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2, the variogram cloud plot (Figure 3.6), and a plot of the selected SNOTEL site numbers. Here, in the variogram cloud plot, the vertical and horizontal axes represent variogram estimates (gamma) and pairwise distance (in miles) respectively. In the plot of the selected SNOTEL site numbers, the axes represent numbers corresponding to the SNOTEL sites in the map view. Three data points in the variogram cloud plot (bottom right) are selected, highlighting three data points in the plot of the SNOTEL site numbers (top right) and also, the corresponding five SNOTEL sites in the map view.

```r
# datenum = selected timepoint number for a specified date
# num = a vector of values representing selected SNOTEL sites
# for single and multiple visualization purposes
# x1, x2 = lower and upper limits of longitude to be used
# for the map view in ipilots
# y1, y2 = lower and upper limits of latitude to be used
# for the map view in ipilots
# maptype = defines the type of map from the Google server to construct

# Brushing/Linking Map View to EDA/ESDA graphics from iPlots
iSweBrushMap(i = 1, j = 2, datenum = 1,
dat1 = swedata, dat2 = sweplotdata,
num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66))
```
Fig. 4.17: Snapshot of the created interactive environment (using iSwePlot) connecting map of Utah with the ten (interactively) selected SNOTEL sites discussed in Chapter 2, the variogram cloud plot (Figure 3.6), and a plot of the selected SNOTEL site numbers. Here, in the the variogram cloud plot, the vertical and horizontal axes represent variogram estimates (gamma) and pairwise distance (in miles) respectively. In the plot of the selected SNOTEL site numbers, the axes represent numbers corresponding to the SNOTEL sites in the map view. Six SNOTEL sites are selected in the map view, highlight 15 data points in the variogram cloud plot (bottom right) as well as the plot of the SNOTEL site numbers (top right).

# i = iset in iplots to select data points from
# j = iset in iplots to highlight corresponding data point(s)
#   based on selected data points from i
# datenum = selected timepoint number for a specified date
# dat1 = 1st dataset to be used
# dat2 = 2nd dataset to be used
# num = a vector of values representing selected SNOTEL sites
#   for single and multiple visualization purposes

# Brushing/Linking EDA/ESDA graphics from iPlots to Map View
iSweBrushPlot(i = 2, j = 1, datenum = 1,
              dat1 = swedata, dat2 = sweplotdata,
              num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66))
# i = iset in iplots to select data points from
# j = iset in iplots to highlight corresponding data point(s)
#   based on selected data points from i
# datenum = selected timepoint number for a specified date
# dat1 = 1st dataset to be used
# dat2 = 2nd dataset to be used
# num = a vector of values representing selected SNOTEL sites
#   for single and multiple visualization purposes

4.5 Upper Sheep Creek SWE Visualization

One of the functions (SweAsciiImagePlot) for a single site in our SWEVIS R package in Chapter 3 creates a display of plots of SWE measurements for different dates and also plots for the difference in SWE measurements for two consecutive dates. We can demonstrate this with SWE data from the Upper Sheep Creek. Figure 4.18 shows a display of plots of SWE measurements on nine different dates in 1993 at the Upper Sheep Creek. The plots for the difference in SWE measurements for two consecutive dates are shown in Figure 4.19. In the plots for differences, we determine the difference between SWE measurements for two consecutive dates at each station and plot them conditioned on seven categories. This is similar to comparative micromaps, a sequence of time-specific maps of values accompanied by a corresponding series of maps of the class or value differences of consecutive maps (Carr and Pickle, 2010). Like the comparative micromaps, the plot of SWE differences allows the reader to see explicitly the amount and location of changes. With our SWEVIS package, the following R code produces these plots.

# Reading and storing SWE data in ASCII format
sweasciidata <- ReadSweAsciiData(filename = c("swe021093.asc", "swe030393.asc", "swe032393.asc", "swe040893.asc", "swe041593.asc", "swe042993.asc", "swe051293.asc", "swe051993.asc", "swe052593.asc"), n.dates = 9)

# filename = a character string giving the name(s) of the file(s)
# saved in ASCII format
# n.dates = number of dates when SWE measurements were taken
Producing Image plots of SWE data in ASCII format

SweAsciiImagePlot(dat = sweasciidata, n.dates = 9, diffplot = FALSE,
    breaks = c(-0.5, 0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0),
    zlim = c(-0.50, 3.0),
    legend = c("0", "(0, 0.5]", "(0.5, 1.0]", "(1.0, 1.5]",
        "(1.5, 2.0]", "(2.0, 2.5]", "(2.5, 3.0]"))

# dat = SWE data in ASCII format
# n.dates = number of dates when SWE measurements were taken
# diffplot = logical. If TRUE, provides image plots of the SWE differences
# breaks = a set of breakpoints for the colors: must give one more breakpoint than color and be sorted in increasing order
# zlim = the minimum and maximum z values for which colors should be plotted, defaulting to the range of the finite values of z. Each of the given colors will be used to color an equispaced interval of this range. The midpoints of the intervals cover the range, so that values just outside the range will be plotted.
# legend = used to add legends to the image plots

SweAsciiImagePlot(dat = sweasciidata, n.dates = 9, diffplot = TRUE,
    breaks = c(-1.2, -0.8, -0.4, -0.0004, 0.0004, 0.4, 0.8, 1.2),
    zlim = c(-1.2, 1.2),
    legend = c("(-1.2, -0.8]", "(-0.8, -0.4]", "(-0.4, 0.0]", " 0.0 ", "(0.0, 0.4]",
        "(0.4, 0.8]", "(0.8, 1.20]"))

# dat = SWE data in ASCII format
# n.dates = number of dates when SWE measurements were taken
# diffplot = logical. If TRUE, provides image plots of the SWE differences
# breaks = a set of breakpoints for the colors: must give one more breakpoint than color and be sorted in increasing order
# zlim = the minimum and maximum z values for which colors should be plotted, defaulting to the range of the finite values of z. Each of the given colors will be used to color an equispaced interval of this range. The midpoints of the intervals cover the range, so that values just outside the range will be plotted.
# legend = used to add legends to the image plots

The iSweAsciiPlot function of the SWEVIS package introduced in Chapter 3 (Section 3.4), can be used to also view and interactively query the SWEs for selected dates of measurements. This function provides capabilities, including linked brushing between the image plot window and interactive statistical graphics (e.g., boxplot) from iPlots. For a
single selected date of SWE measurements, we can do linked brushing between the image plot window and the SWE boxplots.

Examples of analyses using this link are shown in Figures 4.20, 4.21, and 4.22. These figures show the image plot for the SWE measurements on March 3, 1993, and boxplots for these SWE measurements and elevations of the SNOTEL stations at the Upper Sheep Creek in an interactive environment. These figures also give an example of the brushing capabilities of the iSweAsciiPlot function. Selected (highlighted) point(s) on the image plot produce corresponding boxplots in the SWE and elevation boxplot windows. Selected point(s) in the image plot, large filled red circle(s), indicate the locations of SNOTEL station(s). By moving the brush up and down in any of the linked statistical plots, various quantiles of the data can be explored. In Figures 4.20 and 4.21, a horizontal rectangular shaped brush has been used to brush approximately the elevations above 6500 feet and below 6200 feet respectively in the elevation boxplot window. We then notice corresponding points (indicating SNOTEL stations) and boxplots of their corresponding SWE measurements. In Figure 4.22, we first brush vertically in the SWE boxplot window measurements greater than 1 inch. This highlights the corresponding boxplot of observed elevations and 11 red large filled circles in the image plot window. In order to obtain the plots in Figures 4.20, 4.21, and 4.22 in an interactive environment, the following R code is used.

```r
# To produce image plot and boxplots of SWE and elevation of SNOTEL # stations in an interactive environment
iSweAsciiPlot(dat = "USCdata", pngmap = "uscw.png", datenum = 2,
               x1 = 522300, y1 = 4774114, x2 = 523130, y2 = 4774490)

# dat = USC data array containing SWE measurements on different # dates and observed elevation for each SNOTEL station # pngmap = stored file name (in png format) of the image plot # datenum = selected timepoint number for a specified date # x1, x2 = lower and upper limits of the horizontal axis # in the image plot window # y1, y2 = lower and upper limits of the vertical axis # in the image plot window
```
Fig. 4.18: Snow water equivalent measured in the Upper Sheep Creek in 1993 on February 10, March 3, March 23, April 8, April 15, April 29, May 12, May 19, and May 25 (from left to right and top to bottom). Point L10 (black shaded square) was not measured – see Figure 4.4.
Fig. 4.19: Difference in SWE measurements for the consecutive dates in Figure 4.18. Here, “1st–Difference n” represents the first difference in SWE measurements between the nth and (n+1)th dates. Point L10 (black shaded square) was not measured – see Figure 4.4.
Fig. 4.20: Snapshot of the created interactive environment (using iSweAsciiPlot) connecting a map view of an image plot of SWE with SNOTEL stations within the Upper Sheep Creek watershed, boxplots of SWE measurements on March 3, 1993, and boxplots of elevations (in feet). Two of the boxplots are for brushed data values (in red color) and the other two are for non-brushed data values (in white color). Here, highlighted elevations above 6500ft create and highlight a boxplot in the SWE window with corresponding SNOTEL stations in the northeast that are also highlighted in the image plot.
Fig. 4.21: Snapshot of the created interactive environment (using iSweAsciiPlot) connecting a map view of an image plot of SWE with SNOTEL stations within the Upper Sheep Creek watershed, boxplots of SWE measurements on March 3, 1993, and boxplots of elevations (in feet). Two of the boxplots are for brushed data values (in red color) and the other two are for non-brushed data values (in white color). Here, highlighted elevations below 6200ft create and highlight a boxplot in the SWE window with corresponding SNOTEL stations in the northwest that are also highlighted in the image plot.
Fig. 4.22: Snapshot of the created interactive environment (using \textit{iSweAsciiPlot}) connecting a map view of an image plot of SWE with SNOTEL stations within the Upper Sheep Creek watershed, boxplot of SWE measurements on March 3, 1993, and boxplot of elevations (in feet). Two of the boxplots are for brushed data values (in red color) and the other two are for non-brushed data values (in white color). Here, highlighted SWEs greater than 1 inch create and highlight a boxplot in the elevation boxplot window with eleven corresponding SNOTEL stations that are also highlighted in the image plot.
CHAPTER 5
SUMMARY AND CONCLUSIONS

In this dissertation, we presented a data-based statistical model that characterizes seasonal snow water equivalent (SWE) in terms of a nested time-series, with the large scale focusing on the inter-annual periodicity of dominant signals and the small scale accommodating seasonal noise and autocorrelation. This model provides a framework for independently estimating mainly the temporal dynamics of SWE for the various snow telemetry (SNOTEL) sites. We also presented visual techniques for exploring SWE data in an interactive environment. These techniques were bundled into the “open source” R package SWEVIS which will be made available on the CRAN website sometime soon.

The advantage of taking a purely statistical approach for predicting SWE is that it is completely data driven and will yield formal statistical inference and predictions. Long term predictions of SWE will of course contain noise due to an imperfect knowledge of the dynamics but they will be useful in accounting for general trends in snow accumulation.

One of the limitations we faced in our modeling efforts is the inability to handle SNOTEL locations where historical SWE measurements contained missing or unobserved values. However, we would be able to do meaningful forecasting if an appropriate imputation technique is first applied to fill these data gaps. If only a few observations are missing, this could be as simple as a moving average or as complex as co-kriging if SWE measurements and other observations from nearby locations are available.

The SWEVIS package developed encourages data exploration in many different ways in an interactive visualization environment. This package allows for analyses of SWE measurements for single and multiple SNOTEL locations. It also allows for linked brushing connecting map views and statistical graphics from iPlots. To make use of the functions, the SWEVIS package is applied to the SWE data from (i) the SNOTEL sites in Utah and (ii) the Upper Sheep Creek (USC) Watershed in Idaho. In the case of the SWE data from
Utah SNOTEL sites, linked brushing is done between an RGoogleMap view of Utah and a variogram cloud plot for a specific water-year at ten locations. With the USC, the SWE-VIS package is used to visualize SWE measurements saved in ASCII format by connecting image plots and statistical graphics.

Future work might include generating snow accumulation forecasts using a state-of-the-art physical process model and then utilizing these forecasted data in a statistical setting to produce snowmelt forecasts with formal measures of uncertainty. We might assimilate the statistically forecasted snow data into the physical process model and then perform new snow accumulation forecasts. We believe that this iterative process will yield better snow dynamics forecasts than either model could alone. The general modeling approach could contain two major components that would ultimately be linked in a statistically rigorous fashion. The first component would be a novel statistical mixture model that combines both inter-annual and seasonal dynamics, and the second component would involve state-of-the-art physical models for atmospheric processes. The two components would be constructed, calibrated, and implemented independently to evaluate their effectiveness for modeling local snowpack dynamics. Then both could be combined, in an iterative fashion, using statistical data assimilation (e.g., Wikle and Berliner, 2006) to yield quality inter-annual and seasonal inference and prediction for the study region. This assimilation process would also allow us to better understand how and to what extent snow affects the local climate and hydrological cycle as compared to conventional forecasts.

The goal would be to allow the two modeling approaches to borrow strength from each other by combining model results using a method similar to Bayesian model averaging (Hoeting et al., 1999). Specifically, the statistical approach by itself does not explicitly incorporate knowledge about the atmospheric process, but would provide a heavily data-based method for prediction, whereas the physical process models would need to be calibrated and would not have formal mechanisms for accounting for uncertainty. Physical models are based on theory, and therefore SWE can be understood according to the principles and fundamental rules. However, portions of SWE which are phenomena that are not physically
analyzeable cannot be modeled. A drawback in physical models is that they need much time to use past data for tuning parameters in physical model formulas. But combining the physical model and statistical model, which configures a model based on past data using statistical methods, may improve the estimation accuracy of the statistical model. Utilizing the components together, however, would be both advantageous and scientifically thorough and should provide better forecasts and understanding of snowpack dynamics.

Finally, we would extend our R package such that it can be linked to a database with SWE data at the operational level. Interaction tools like zooming and animation also would be considered. Spatially lagged scatterplots to access local instability in a spatial association could also be incorporated. In addition, lines would be added in the map view. These lines would connect each pair of locations related to a point that has been brushed in the variogram cloud plot. Also to be considered is cut-off distance in the variogram cloud plot. This would allow us to select pairs of SNOTEL locations that satisfy a given cut-off condition. For example, we would be interested selecting/brushing pairs of SNOTEL locations with distances greater or less than the cut-off distance. We would consider making the USC plots and difference plots interactive, e.g., brush the 10% highest SWE increases and see on which SNOTEL locations and time points they occur. Last but not the least, we would upgrade to iPlots eXtreme (also known as Acinonyx) (Urbanek, 2011). This would allow for speed optimization in order to explore and visualize large data sets. One question we would consider for future work is, if it is worthwhile to quantify prediction quality. In other words, we would like to determine if it is better to predict earlier or later in the water-year by comparing different initiation dates, e.g., compare predictions from initiation date January 15 to January 22 to January 29, etc. We would also consider applying our statistical model to a different region of similar SWE distribution, e.g., Swiss alps and Canada. If possible, we would address multi-modal SWE patterns as well as non normal-based (skewed) distribution modeling of SWE.
REFERENCES


Anselin, L., 1992. SpaceStat, a Software Package for the Analysis of Spatial Data. National Center for Geographic Information and Analysis, University of California, Santa Barbara, California.


   URL http://www.spacestat.com/utilities.htm

Anselin, L., Smirnov, O., 1999b. The DynESDA Extension for ArcView.
   URL http://www.spacestat.com/utilities.htm


   URL http://CRAN.R-project.org/package=maps/


URL http://CRAN.R-project.org/package=mvtnorm

URL http://cran.r-project.org/web/packages/googleVis/


URL http://www.jstatsoft.org/v38/i08/


URL http://cran.r-project.org/web/packages/PBSmapping/


URL http://cran.r-project.org/web/packages/splancs/


URL http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/


URL http://www.rosuda.org/iplots/


APPENDICES
APPENDIX A

SUMMARY OF COMPUTER R CODE

The provided R code enables readers to apply the proposed methods to their own data and to reproduce all figures in this paper, a key requirement in terms of reproducible research (Baggerly and Berry, 2011). Datasets for this paper can be obtained from the freely accessible web archives of NWCC (http://www.wcc.nrcs.usda.gov/snow/).

For each SNOTEL site, the “read.Snotel.R” function is used to clean up the historical and current water-year daily SWE measurements by reading the downloaded SNOTEL data in text format one year at a time and output vectors of values with NAs in place of blanks and “- - -”. The “utah.snotel.current.data.R” and “utah.snotel.historical.data.R” functions use the “read.Snotel.R” function to put the already downloaded historical and current text files of SWE measurements for the ninety active Utah SNOTEL sites in a matrix form.

The “swe.svd.mcmc.R” function performs the SVD on the data matrix discussed in equation (2.1), computes, and saves the sampling estimates of the model parameters $\alpha$, $\alpha_{st}$, and $\Sigma_y$ in Section 2.2.3 for a specified number of simulations. This code requires the R packages: MCMCpack (Martin et al., 2011), mvtnorm (Genz and Bretz, 2009; Genz et al., 2011), msm (Jackson, 2011), lattice (Sarkar, 2008), Mass (Venables and Ripley, 2002) and spdep (Pebesma and Bivand, 2005; Bivand et al., 2008). It also produces samples of the posterior distributions for the unobserved SWE after a chosen day $d$ in our forecasting of SWE. The function “plot.swe.post.R” uses output from the “swe.svd.mcmc.R” to enable us to reproduce Figures 2.8, 2.9, and 2.10. To reproduce Figures 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, and 2.11 as well as Tables 2.1 and 2.2, we used some of the commands in the “Codes-Used.R” file. The function “prop.pred.R” is used to generate Table 2.2.

The “invgammastrt.R” function computes the initial values of the shape and scale parameters of the inverse gamma distribution for $\sigma^2_y$. The function “krig.swe.R” augments the historical SWE data for SNOTEL sites with missing values using a Kriging approach. The text file “xcoord.txt” contains latitude and longitude coordinates of the ninety active SNOTEL sites in this article. All R code and data have been combined in a single zip archive (odei-et-al-R-Code-Data.zip) and can be downloaded from http://www.math.usu.edu/~symanzik/papers/odei-et-al-R-Code-Data.zip. Our results were produced with R version 2.15.1.
APPENDIX B

DERIVATION OF THE FULL-CONDITIONAL DISTRIBUTIONS

Considering the definition of $A_{st}$ in section 2.2.3, we can rewrite equation (2.3) as

$$z_{st} = Q_{s,t-1} \alpha_{st} + \epsilon_{st},$$  \hspace{1cm} (B.1)

where $z_{st} = y_{st} - \bar{y}_s$ and $Q_{s,t-1} = V D \text{diag}(u_{s,t-1})$.

The joint posterior distribution of the parameters in the full model, given the data, is given by

$$[\{\alpha_{st}\}, \alpha, \sigma^2_y, \rho | \{z_{st}\}] \propto \prod_{t=1}^{T} \left[ z_{st} | \alpha_{st}, \sigma^2_y \right] \times \left[ \alpha | \sigma^2_y \times [\rho] \right].$$  \hspace{1cm} (B.2)

To use the computationally efficient Gibbs sampler, we need to analytically specify the full-conditional distribution for each parameter in the model. The derivations of these full-conditionals are given here.

From the joint posterior distribution above, the full-conditional distribution for $\alpha_{st}$, the autoregression coefficients related to the global coefficients, $\alpha$, is

$$[\alpha_{st} | . ] \propto \left[ z_{st} | \alpha_{st}, \sigma^2_y \right] \times [\alpha | \sigma^2_y].$$  \hspace{1cm} (B.3)

The prior distribution of $\alpha_{st}$ is $N(\alpha, \sigma^2_\alpha I)$, and the likelihood is the process model:

$$z_{st} | \alpha_{st}, \sigma^2_y \sim N(Q_{s,t-1} \alpha_{st}, \Sigma_y)$$

Further simplification yields:

$$[\alpha_{st} | . ] \propto \exp\left\{ -\frac{1}{2} (z_{st} - Q_{s,t-1} \alpha_{st})' \Sigma_y^{-1} (z_{st} - Q_{s,t-1} \alpha_{st}) \right\}$$

$$\times \exp\left\{ -\frac{1}{2} (\alpha_{st} - \alpha)' (\sigma^2_\alpha I)^{-1} (\alpha_{st} - \alpha) \right\}$$

$$\times \exp\left\{ -\frac{1}{2} (-2z_{st}' \Sigma_y^{-1} Q_{s,t-1} \alpha_{st}) \right\}$$

$$\times \exp\left\{ -\frac{1}{2} (\alpha_{st}' Q_{s,t-1} \Sigma_y^{-1} Q_{s,t-1} \alpha_{st}) \right\}$$

$$\times \exp\left\{ -\frac{1}{2} (-2\alpha' (\sigma^2_\alpha I)^{-1} \alpha_{st}) \right\}$$

$$\times \exp\left\{ -\frac{1}{2} (\alpha_{st}' (\sigma^2_\alpha I)^{-1} \alpha_{st}) \right\}$$
\[ \propto \exp \left\{ \frac{1}{2} \left( -2 \left( z'_{st} \Sigma^{-1}_{y} Q_{s,t-1} + \alpha' \left( \sigma^2_{\alpha} I \right)^{-1} \right) \alpha_{st} \right) \right\} \]
\[ \times \exp \left\{ \frac{1}{2} \left( \alpha'_{st} \left( Q'_{s,t-1} \Sigma^{-1}_{y} Q_{s,t-1} + \left( \sigma^2_{\alpha} I \right)^{-1} \right) \alpha_{st} \right) \right\} \]

(B.4)

Then the full conditional distribution for \( \alpha_{st} \) is

\[ \alpha_{st} \sim N(A^{-1}b, A^{-1}) \]  

(B.5)

where \( A = Q'_{s,t-1} \Sigma^{-1}_{y} Q_{s,t-1} + \left( \sigma^2_{\alpha} I \right)^{-1} \) and \( b = \left( z'_{st} \Sigma^{-1}_{y} Q_{s,t-1} + \alpha' \left( \sigma^2_{\alpha} I \right)^{-1} \right)' \).

The full conditional distributions for \( \alpha \) and \( \sigma^2_{\alpha} \) can be found in a similar way. Consider \( \alpha \):

\[
[\alpha|.] \propto \left( \prod_{t=1}^{T} [\alpha_{st}|\alpha] \right) \times [\alpha],
\]

(B.6)

and the prior distribution of \( \alpha \) is \( N(0, \sigma^2_{\alpha} I) \). Simplifying equation (B.6) yields:

\[
[\alpha|.] \propto \left\{ \prod_{t=1}^{T} \exp \left( \frac{1}{2} \left( \alpha_{st} - \alpha \right)' \left( \sigma^2_{\alpha} I \right)^{-1} \left( \alpha_{st} - \alpha \right) \right) \right\} \times \exp \left\{ \frac{1}{2} \left( \alpha - 0 \right)' \left( \sigma^2_{\alpha} I \right)^{-1} \left( \alpha - 0 \right) \right\}
\]
\[ \propto \exp \left\{ \frac{1}{2} \sum_{t=1}^{T} \left( \alpha_{st} - \alpha \right)' \left( \sigma^2_{\alpha} I \right)^{-1} \left( \alpha_{st} - \alpha \right) \right\} \times \exp \left\{ \frac{1}{2} \left( \alpha' \left( \sigma^2_{\alpha} I^{-1} \alpha \right) \right) \right\}
\]
\[ \propto \exp \left\{ \frac{1}{2} \left( \sum_{t=1}^{T} \left( -2 \alpha'_{st} \left( \sigma^2_{\alpha} I^{-1} \right) \right) \right) \right\} \times \exp \left\{ \frac{1}{2} \left( \sum_{t=1}^{T} \left( \alpha' \left( \sigma^2_{\alpha} I^{-1} \right) \right) \right) \right\}
\]
\[ \times \exp \left\{ \frac{1}{2} \left( \alpha' \left( \sigma^2_{\alpha} I^{-1} \right) \right) \right\}
\]
\[ \propto \exp \left\{ \frac{1}{2} \left( \sum_{t=1}^{T} \left( \alpha_{st} - \alpha \right)' \left( \sigma^2_{\alpha} I^{-1} \right) \right) \right\} \times \exp \left\{ \frac{1}{2} \left( \sum_{t=1}^{T} \left( \alpha' \left( \sigma^2_{\alpha} I^{-1} \right) \right) \right) \right\}
\]
\[ \times \exp \left\{ \frac{1}{2} \left( \alpha' \left( \sum_{t=1}^{T} \left( \sigma^2_{\alpha} I^{-1} \right) \right) \right) \right\}
\]

(B.7)

Then the full conditional distribution for \( \alpha \) is

\[ \alpha \sim N(A^{-1}b, A^{-1}) \]  

(B.8)
where $A_\ast = \left( \sum_{t=1}^{T} (\sigma^2_\alpha I)^{-1} \right) + (\sigma^2_\alpha I)^{-1}$ and $b_\ast = \left( \sum_{t=1}^{T} \alpha'_{st} (\sigma^2_\alpha I)^{-1} \right)'$.

The full-conditional distribution in the case of $\sigma^2_y$ is

$$[\sigma^2_y]\propto \left( \prod_{t=1}^{T} [z_{st}|\alpha_{st},\sigma^2_y] \right) \times [\sigma^2_y], \quad \text{(B.9)}$$

and the prior distribution of $\sigma^2_y$ is $IG(\nu_1, \nu_2)$. Here, it must be noted that

$$[\sigma^2_y|\nu_1, \nu_2] = \frac{1}{\nu_1^2 \Gamma(\nu_2)} (\sigma^2_y)^{-\nu_2+1} \exp \left\{ -\frac{1}{\nu_1 \sigma^2_y} \right\}. \quad \text{(B.10)}$$

Letting $R_{st} = z_{st} - Q_{s,t-1} \alpha_{st}$ and further simplifying equation (B.9) yields:

$$[\sigma^2_y]\propto \frac{1}{\Sigma_y^{-\frac{T}{2}}} (\sigma^2_y)^{-\nu_2+1} \exp \left\{ -\frac{1}{\nu_1 \sigma^2_y} \right\}$$

$$\times \exp \left\{ -\frac{1}{2} \sum_{t=1}^{T} \left( R_{st}' \Sigma_y^{-1} R_{st} \right) \right\}$$

$$\times \exp \left\{ -\frac{1}{2} \nu_2 \sum_{t=1}^{T} R_{st}' \left( \sigma^2_y (I - \rho W)^{-1} \right)^{-1} R_{st} \right\}$$

$$\propto \left( \sigma^2_y \right)^{-\nu_1 + \nu_2 + 1} \exp \left\{ -\frac{1}{\nu_1 \sigma^2_y} \right\} \Sigma_y^{-\frac{T}{2}}$$

$$\times \exp \left\{ -\frac{1}{\nu_1} + \frac{1}{2} \sum_{t=1}^{T} \left( R_{st}' (I - \rho W) R_{st} \right) \right\} \quad \text{(B.11)}$$

Then the full conditional distribution for $\sigma^2_y$ is

$$\sigma^2_y \sim IG(r^*, q^*) \quad \text{(B.12)}$$

where $r^* = \frac{1}{\nu_1} + \frac{1}{2} \sum_{t=1}^{T} \left( R_{st}' (I - \rho W) R_{st} \right)$ and $q^* = \frac{M \times T}{2} + \nu_2$.

We have derived analytic full-conditional distributions for all parameters in the model, and the joint posterior distribution of the parameters in the model can be found using hybrid Metropolis-Hastings and Gibbs Markov Chain Monte Carlo (MCMC) Sampling algorithm (Banerjee et al., 2004; Carlin and Lewis, 2009).
APPENDIX C

PROCEDURAL PARADIGM #1 (R CODE)

In this appendix, titles match code names, which are case sensitive. To be readable, the code is structured. The following codes reflects the content of the R package. To run the code outside the R package, copy and paste into an R GUI taking care to correct double quotes ” as necessary. Sometimes the double quotes in Word are misinterpreted in the R GUI.

C.1 Function to Read SNOTEL SWE Data

# Reads in SNOTEL SWE data in "txt" file format downloaded from the
# SNOTEL website one year at a time and output vector of values with
# NAs in place of blanks and ‘---’.

read.snotel <- function(f = NULL, nskip = 1){
  skip <- 7 + 43 * (nskip-1)
  if (!is.null(f)) {
    swe.vec <- as.real(as.matrix(read.fwf(f, skip = skip,
      widths = c(4, rep(6, 12)),
      na.strings = c("---", " "))[1:31, -1])
  }
  swe.vec
}

C.2 Inverse Gamma Start Function

# Inverse Gamma function to obtain the initial values of the scale and shape
# parameters, r and q, respectively, for the posterior distribution of the
# variance component

invgammastrt <- function(igmn, igvar){
  q <- 2 + (igmn^2)/igvar
  r <- 1/(igmn * (q - 1))
  list(r = r, q = q)
}
C.3 SWE SVD-MCMC Function

# Function to perform singular value decomposition (SVD) and MCMC
# simulations to obtain parameter estimates for the posterior distributions

swe.svd.mcmc <- function(Y, q, n.mcmc, mm = NULL, outdat = 100) {

  # Y is a T by m matrix with T = number of years, m = number of days
  # q is number of important signals chosen for dimension reduction
  # n.mcmc is number of simulations to be done
  # outdat is the day after which forecasting is done based on the
  #     posterior distribution

  ## Subroutines and Libraries
  ##
  require(MCMCpack)
  require(mvtnorm)
  require(spdep)
  require(msm)
  source("invgammastrt.R")

  ## Create the proximity matrix

  makeW <- function(m) {
    W <- as.matrix(as_dsTMatrix_listw(similar.listw(nb2listw(knn2nb(knearneigh(
      cbind(rep(1, m), 1:m)), sym = TRUE), style = "W"))))
    W
  }

  ## Setup Variables
  ##
  T <- dim(Y)[1] - 1
  m <- dim(Y)[2]
  Y.c <- scale(Y[-(T + 1), ], scale = FALSE)
  mu <- attributes(Y.c)[[2]]
  Y.svd <- svd(Y.c)
  U <- Y.svd$u
  D <- diag(Y.svd$d)
  V <- Y.svd$v[, 1:q]
  X <- (U%*%D)[, 1:q]

  Y.Tplus1 <- Y[T + 1, ]

  if (is.null(mm)) {
    mm <- max((1:m)[!is.na(Y.Tplus1)])
  }

s2.save <- rep(0, n.mcmc)
rho.save <- rep(0, n.mcmc)
alpha.mean <- matrix(0, q, T + 1)
alpha.save <- matrix(0, q, n.mcmc)
alpha.Tplus1.save <- matrix(0, q, n.mcmc)
Y.pred <- matrix(Y[T + 1, ], m, n.mcmc)
Y.max <- apply(Y[1:T, ], 2, max)

##
## Priors and Starting Values
##

s2alpha <- 10
s2rho <- 4
taulpha <- 100
s2mn <- 10
s2var <- 100
rr <- invgammastrt(s2mn, s2var)$r
qq <- invgammastrt(s2mn, s2var)$q

alphat <- matrix(0, q, T+1)
alpha <- rep(0, q)
s2 <- 1
rho <- 0.99
rho.tune <- 0.01
Phi <- diag(m)
R <- solve(diag(m) - rho * makeW(m))%*%Phi
Rinv <- solve(R)
Sig <- s2 * R
Siginv <- solve(Sig)

MU <- mu + V%*%t(X)

##
## Initialize Timing Variables
##

time1 <- proc.time()
time2 <- time1
timeidx <- 0

##
## Begin MCMC Loop
##

for (k in 1:n.mcmc) {
  #cat(k, " ")

  ##
  ## Timing Calculations
  ##

if (k==12) {
  tentime <- (proc.time()-time1)[3]
  #cat("n", tentime * (n.mcmc/600), "expected minutes", "n")
}
if (k%%100==0) {
  timeidx <- timeidx + 1
  elapsetime <- (proc.time()-time2)[3]
  #cat("n", elapsetime/60, "elapsed minutes", "n")
  leftidx <- n.mcmc/100 - timeidx
  #cat("n", (elapsetime/timeidx) * leftidx/60, "remaining minutes", "n")
}

##
## Sample alpha_t
##
#cat("a_t", "n")
for (t in 2:T) {
  alpha.var <- solve(diag(X[t - 1, ])*t(V)*Siginv*V*diag(X[t - 1, ])) + diag(q)/s2alpha
  alpha.mn <- alpha.var*t(c(Y[t] - mu)*Siginv*V*diag(X[t - 1, ])) + alpha/s2alpha
  alphat[, t] <- rmvnorm(1, alpha.mn, alpha.var, method = "chol")
}
alpha.var <- solve(diag(X[T, ])*t(V[1:mm, ]))%*%solve(Sig[1:mm, 1:mm])%*%V[1:mm, ]%*%diag(X[T, ])) + diag(q)/s2alpha
alpha.mn <- alpha.var*t(c(Y.Tplus1[1:mm] - mu[1:mm]))%*%solve(Sig[1:mm, 1:mm])%*%V[1:mm, ]%*%diag(X[T, ])) + alpha/s2alpha
alphat[, T + 1] <- rmvnorm(1, alpha.mn, alpha.var, method = "chol")

##
## Sample alpha
##
#cat("a", "n")
alpha.var <- solve(T * diag(q)/s2alpha + diag(q)/taualpha)
alpha.mn <- alpha.var*%apply(alphat, 1, sum)/s2alpha
alpha <- rmvnorm(1, alpha.mn, alpha.var, method = "chol")

##
## Sample s2
##
#cat("s2", "n")
tmpsum <- sum(diag((Y[1:T, ] - t(MU[, 1:T]))*%Rinv%*t(Y[1:T, ] - t(MU[, 1:T])))
rstar <- 1/(tmpsum/2 + 1/rr)
qstar <- (m * T)/2 + qq
s2 <- 1/rgamma(1, qstar, , rstar)

Sig <- s2 * R
Siginv <- solve(Sig)

##
## Sample rho
##

#cat("rho", " ")
rho.star <- rtnorm(1, rho, rho.tune, -1, 1)

#cat(rho, rho.star, " ")
R.star <- solve(diag(m) - rho.star * makeW(m))%*%Phi
Sig.star <- s2 * R.star

tmpsum1 <- -sum(log(eigen(Sig.star, symmetric = TRUE,
    only.values = TRUE)$values)) -
    sum(diag((Y[2:T,] - t(MU[, 2:T]))%*
        solve(Sig.star)%*%t(Y[2:T,] - t(MU[, 2:T]))))

tmpsum2 <- -sum(log(eigen(Sig, symmetric = TRUE,
    only.values = TRUE)$values)) -
    sum(diag((Y[2:T,] - t(MU[, 2:T]))%*
        Siginv%*%t(Y[2:T,] - t(MU[, 2:T]))))

mh1 <- tmpsum1 + dtnorm(rho.star, 0, sqrt(s2rho), -1, 1, log = TRUE) +
    dtnorm(rho, rho.star, sqrt(s2rho), -1, 1, log = TRUE)

mh2 <- tmpsum2 + dtnorm(rho, 0, sqrt(s2rho), -1, 1, log = TRUE) +
    dtnorm(rho.star, rho, sqrt(s2rho), -1, 1, log = TRUE)

mh <- exp(mh1- mh2)
if (mh > runif(1)) {
    rho <- rho.star
    Sig <- Sig.star
    R <- R.star
    Rinv <- solve(R.star)
    Siginv <- solve(Sig)
}

##
## Save Samples
##

alpha.mean <- alpha.mean + alphat/n.mcmc
alpha.save[, k] <- alpha
rho.save[k] <- rho
s2.save[k] <- s2
## Get Predictions

```r
#cat("pred", "\n")

Sig.obs.inv <- solve(Sig[1:mm, 1:mm])

Y.pred.mn <- V%*%diag(c(alphat[, T + 1]))%*%X[T, ] + mu

Y.pp.mn <- Y.pred.mn[(mm + 1):m] + Sig[(mm + 1):m, 1:mm]%*%Sig.obs.inv%*% (Y[T + 1, 1:mm] - Y.pred.mn[1:mm])

Y.pp.Sig <- Sig[(mm + 1):m, (mm + 1):m] - Sig[(mm + 1):m, 1:mm]%*% Sig.obs.inv%*%Sig[1:mm, (mm + 1):m]

Y.pred[(mm + 1):m, k] <- rmvnorm(1, Y.pp.mn, Y.pp.Sig, method = "chol")
Y.pred[, k][Y.pred[, k] < 0] <- 0
Y.pred[, k][Y.pred[, k] > Y.max] <- Y.max[Y.pred[, k] > Y.max]
```

## Write Temporary Output

```r
if (k%%outdat==0) {
  tmpout <- list(n.mcmc = n.mcmc, alpha.save = alpha.save,
                s2.save = s2.save, Y = Y, Y.pred = Y.pred, mu = mu,
                mm = mm, rho.save = rho.save, alpha.mean = alpha.mean,
                Y.svd = Y.svd)
  save(tmpout, file = "tmpout.RData")
}
```

## Write Output

```r
list(n.mcmc = n.mcmc, alpha.save = alpha.save, s2.save = s2.save, Y = Y,
     mu = mu, mm = mm, Y.pred = Y.pred, rho.save = rho.save,
     alpha.mean = alpha.mean, Y.svd = Y.svd)
```
C.4 Posterior Prediction Plot Functions

plot.swe.post <- function(dat, date = NULL, head = NULL){
    # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
    # Working Directory)
    
    ## Setup Variables
    ##
    n.burn <- round(dat$n.mcmc/4)
    T <- dim(dat$Y)[1]
    m <- dim(dat$Y)[2]
    mos <- cumsum(modays)
    #y.max <- max(dat$Y, na.rm = TRUE)
    y.max <- 75
    
    ## Get Prediction 50% and 95% Credible Intervals
    ##
    CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
    CI.95 <- c(CI.95[, 1], rev(CI.95[, 2]))
    CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
    CI.50 <- c(CI.50[, 1], rev(CI.50[, 2]))
    
    ## Plot the Graphics
    ##
    matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),
            type = "n",
            col = 3,
            lty = 1,
            ylab = "SWE",
            ylim = c(0, y.max),
            xlab = "Day (in water-year)",
            cex.lab = 1.65,
            main = paste(head),
            cex.main = 2.4,
            axes = FALSE)
    axis(1, c(0, 100, 200, 300, 365),
         c("Oct. 1", "Jan. 8", "Apr. 18", "Jul. 27", "Sep. 30"),
         cex.axis = 1.7)
    axis(2, cex.axis = 1.7)
    box()
C.4.1 Functions for Tony Grove SNOTEL Site

plot.swe.post_jan_Fig8_TGsite <- function(dat, date = NULL, head = NULL){

    # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
    # Working Directory)

    ##
    ## Setup Variables
    ##

    n.burn <- round(dat$n.mcmc/4)
    T <- dim(dat$Y)[1]
    m <- dim(dat$Y)[2]
    mos <- cumsum(modays)
    y.max <- max(dat$Y, na.rm = TRUE)
    y.max <- 75

    abline(v = c(0, mos), col = 8)
    lines(apply(dat$Y[-T, ], 2, mean), col = "salmon", lwd = 11)
    matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.001, 0.999))),
           type = "l",
           lty = 2,
           col = 1,
           lwd = 9,
           add = TRUE)
    polygon(c(1:m, m:1), CI.95, col = gray(.5), border = NA)
    polygon(c(1:m, m:1), CI.50, col = gray(.25), border = NA)
    lines(dat$Y[T, ], lwd = 7, col = 2)
    abline(v = 100, col = "brown", lty = 4, lwd = 10)
    text(mos - 15, rep(y.max, 12), labels = c(10:12, 1:9))

    legend("right",
           col = c(1, gray(0.25), gray(0.5), 4, 2, 8),
           lty = c(2, 1, 1, 1, 1, 1),
           cex = 1.8,
           lwd = c(6, 8, 8, 6, 6, 2),
           legend = c("prev. data env.", "pred. CI: 50%", "pred. CI: 95%",
                       "prev. data mean", "current data", "months"),
           bg = "white")
}
## Get Prediction 50% and 95% Credible Intervals

CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
CI.95 <- c(CI.95[1, ], rev(CI.95[2, ]))
CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
CI.50 <- c(CI.50[1, ], rev(CI.50[2, ]))

## Plot the Graphics

matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),
    type = "n",
    col = 3,
    lty = 1,
    # ylab = "SWE",
    ylab = "",
    ylim = c(0, y.max),
    # xlab = "Day (in water-year)",
    # cex.lab = 1.65,
    # main = paste(head),
    # cex.main = 2.4,
    axes = FALSE)

axis(1, c(0, 92, 182, 273),
    c("Oct. 1", "Jan. 1", "Apr. 1", "Jul. 1"),
    cex.axis = 1.8)

axis(2, cex.axis = 1.8, las = 0, tck = -0.023, line = -0.03)
box()

abline(v = c(0, mos), col = 8)
text(mos - 15, rep(y.max, 12), labels = c(10:12, 1:9), cex = 1.6)
polygon(c(1:m, m:1), CI.95, col = gray(.5), border = NA, lwd = 5)
polygon(c(1:m, m:1), CI.50, col = gray(.25), border = NA, lwd = 5)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.001, 0.999))),
    type = "l",
    lty = 2,
    col = 1,
    lwd = 6,
    add = TRUE)

lines(apply(dat$Y[-T, ], 2, mean), col = "salmon", lwd = 11)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.05, 0.95))),
    type = "l",
    lty = 3,
# pch = 3,
col = 1,
lwd = 7,
add = TRUE)

lines(dat$Y[T, ], lwd = 11, col = "orange")

abline(v = 100, col = "darkred", lty = 4, lwd = 10)
text(x = 100, y = 62, labels = "Jan. 8, 2008 --", pos = 2,
col = "darkred", cex = 2.4, font = 1)

legend(x = c(283, 377), y = c(49, 72),
col = c(1, gray(0.25), gray(0.5), "salmon", "orange", 8, 1),
lty = c(2, 1, 1, 1, 1, 1, 3),
cex = 1.3,
lwd = c(4, 8, 8, 5, 5, 3, 4),
legend = c("prev. data env.", "pred. CI: 50%", "pred. CI: 95%",
"prev. data mean", "current data", "months",
"5th & 95th perc.")
bg = "white")

plot.swe.post_jan_TGsite <- function(dat, date = NULL, head = NULL){

    # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
    # Working Directory)

    ## Setup Variables
    ##
    n.burn <- round(dat$n.mcmc/4)
    T <- dim(dat$Y)[1]
    m <- dim(dat$Y)[2]
    mos <- cumsum(modays)
    #y.max <- max(dat$Y, na.rm = TRUE)
    y.max <- 75

    ##
    ## Get Prediction 50% and 95% Credible Intervals
    ##
    CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
    CI.95 <- c(CI.95[1, ], rev(CI.95[2, ]))
    CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
    CI.50 <- c(CI.50[1, ], rev(CI.50[2, ]))

    ##}
## Plot the Graphics

```r
matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),
        type = "n",
        col = 3,
        lty = 1,
        ylab = "SWE",
        ylab = " ",
        ylim = c(0, y.max),
        xlab = "Day (in water-year)",
        cex.lab = 1.65,
        main = paste(head),
        cex.main = 2.4,
        axes = FALSE)

axis(1, c(0, 92, 182, 273),
      c("Oct. 1", "Jan. 1", "Apr. 1", "Jul. 1"),
      cex.axis = 2.5, font = 2)

axis(2, cex.axis = 2.5, las = 0, tck = -0.023, line = -0.03, font = 2)

box()

abline(v = c(0, mos), col = 8)

text(mos - 15, rep(y.max, 12), labels = c(10:12, 1:9), cex = 2.3)
polygon(c(1:m, m:1), CI.95, col = gray(.5), border = NA, lwd = 5)
polygon(c(1:m, m:1), CI.50, col=gray(.25), border = NA, lwd = 5)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.001, 0.999))),
        type = "l",
        lty = 2,
        col = 1,
        lwd = 9,
        add = TRUE)

lines(apply(dat$Y[-T, ], 2, mean), col = "salmon", lwd = 11)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.05, 0.95))),
        type = "l",
        lty = 3,
        pch = 3,
        col = 1,
        lwd = 7,
        add = TRUE)

lines(dat$Y[T, ], lwd = 11, col = "orange")

abline(v = 100, col = "darkred", lty = 4, lwd = 10)
text(x = 100, y = 62, labels = "Mar. 9 --", pos = 2,
      col = "darkred", cex = 2.6, font = 2)
```
plot.swe.post_feb_TGsite <- function(dat, date = NULL, head = NULL){
  # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
  # Working Directory)

  ## Setup Variables
  ##
n.burn <- round(dat$n.mcmc/4)
  T <- dim(dat$Y)[1]
  m <- dim(dat$Y)[2]
  mos <- cumsum(modays)
  #y.max <- max(dat$Y, na.rm = TRUE)
  y.max <- 75

  ## Get Prediction 50% and 95% Credible Intervals
  ##
  CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
  CI.95 <- c(CI.95[1, ], rev(CI.95[2, ]))
  CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
  CI.50 <- c(CI.50[1, ], rev(CI.50[2, ]))

  ## Plot the Graphics
  ##
  matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),
          type = "n",
          col = 3,
          lty = 1,
          # ylab = "SWE",
          ylab = "",
          ylim = c(0, y.max),
          # xlab = "Day (in water-year)",
          xlab = "Months",
          bg = "white")
}
plot.swe.post_mar_TGsite <- function(dat, date = NULL, head = NULL) {

    # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
    # Working Directory)

    ##
    ## Setup Variables
    ##

    n.burn <- round(dat$n.mcmc/4)
    T <- dim(dat$Y)[1]
    m <- dim(dat$Y)[2]
    mos <- cumsum(modays)
    y.max <- max(dat$Y, na.rm = TRUE)
    y.max <- 75

    ##
    ## Get Prediction 50% and 95% Credible Intervals
    ##

    CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
    CI.95 <- c(CI.95[1, ], rev(CI.95[2, ]))
    CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
    CI.50 <- c(CI.50[1, ], rev(CI.50[2, ]))

    ##
    ## Plot the Graphics
    ##

    matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),
            type = "n",
            col = 3,
            lty = 1,
            ylab = "SWE",
            xlab = "Day (in water-year)",
            cex.lab = 1.65,
            cex.main = 2.4,
            axes = FALSE)

    axis(1, c(0, 92, 182, 273),
          c("Oct. 1", "Jan. 1", "Apr. 1", "Jul. 1"),
          cex.axis = 2.5, font = 2)

    axis(2, cex.axis = 2.5, las = 0, tck = -0.023, line = -0.03, font = 2)
    box()

    abline(v = c(0, mos), col = 8)
```r
plot.swe.post_apr_TGsite <- function(dat, date = NULL, head = NULL){
  
  # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
  # Working Directory)
  
  ##
  ## Setup Variables
  ##
  
  n.burn <- round(dat$n.mcmc/4)
```
T <- dim(dat$Y)[1]
m <- dim(dat$Y)[2]
mos <- cumsum(modays)
#y.max <- max(dat$Y, na.rm = TRUE)
y.max <- 75

##
## Get Prediction 50% and 95% Credible Intervals
##

CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
CI.95 <- c(CI.95[1, ], rev(CI.95[2, ]))
CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
CI.50 <- c(CI.50[1, ], rev(CI.50[2, ]))

##
## Plot the Graphics
##

matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),

   type = "n",
   col = 3,
   lty = 1,
   # ylab = "SWE",
   ylab = "",
   ylim = c(0, y.max),
   # xlab = "Day (in water-year)",
   # cex.lab = 1.65,
   # main = paste(head),
   # cex.main = 2.4,
   axes = FALSE)

axis(1, c(0, 92, 182, 273),
     c("Oct. 1", "Jan. 1", "Apr. 1", "Jul. 1"),
     cex.axis = 2.5, font = 2)

axis(2, cex.axis = 2.5, las = 0, tck = -0.023, line = -0.03, font = 2)
box()

abline(v = c(0, mos), col = 8)
text(mos - 15, rep(y.max, 12), labels = c(10:12, 1:9), cex = 2.3)
polygon(c(1:m, m:1), CI.95, col = gray(.5), border = NA, lwd = 5)
polygon(c(1:m, m:1), CI.50, col = gray(.25), border = NA, lwd = 5)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.001, 0.999)))),
   type = "l",
lty = 2,
   col = 1,
lwd = 9,
add = TRUE)

lines(apply(dat$Y[-T, ], 2, mean), col = "salmon", lwd = 11)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.05, 0.95))),
       type = "l",
       lty = 3,
       # pch = 3,
       col = 1,
       lwd = 7,
       add = TRUE)

lines(dat$Y[T, ], lwd = 11, col = "orange")

abline(v = 190, col = "darkred", lty = 4, lwd = 10)

text(x = 190, y = 62, labels = "Apr. 8 --", pos = 2,
     col = "darkred", cex = 2.6, font = 2)

# legend(x = c(283, 377), y = c(49, 72),
#       col = c(1, gray(0.25), gray(0.5), "salmon", "orange", 8, 1),
#       lty = c(2, 1, 1, 1, 1, 3),
#       cex = 1.3,
#       lwd = c(4, 8, 8, 5, 5, 3, 4),
#       legend = c("prev. data env.", "pred. CI: 50\%", "pred. CI: 95\%",
#                  "prev. data mean", "current data","months",
#                  "5th & 95th perc."),
#       bg = "white")
}

C.4.2 Functions for Little Bear SNOTEL Site

plot.swe.post_jan_LBsite <- function(dat, date = NULL, head = NULL){

    # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
    # Working Directory)

    ##
    ## Setup Variables
    ##

    n.burn <- round(dat$n.mcmc/4)
    T <- dim(dat$Y)[1]
    m <- dim(dat$Y)[2]
    mos <- cumsum(modays)
    #y.max <- max(dat$Y, na.rm = TRUE)
    y.max <- 33
}
## Get Prediction 50% and 95% Credible Intervals

CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
CI.95 <- c(CI.95[1,], rev(CI.95[2,]))
CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
CI.50 <- c(CI.50[1,], rev(CI.50[2,]))

## Plot the Graphics

matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),
        type = "n",
        col = 3,
        lty = 1,
# ylab = "SWE",
        ylab = "",
# ylim = c(0, y.max),
        ylim = c(0, y.max),
# xlab = "Day (in water-year)",
        xlab = "",
# cex.lab = 1.65,
        cex.lab = 1.65,
# main = paste(head),
        main = paste(head),
# cex.main = 2.4,
        cex.main = 2.4,
        axes = FALSE)

axis(1, c(0, 92, 182, 273),
      c("Oct. 1", "Jan. 1", "Apr. 1", "Jul. 1"),
      cex.axis = 2.5, font = 2)

axis(2, cex.axis = 2.5, las = 0, tck = -0.023, line = -0.03, font = 2)
box()

abline(v = c(0, mos), col = 8)
text(mos - 15, rep(y.max, 12), labels = c(10:12, 1:9), cex = 2.3)

polygon(c(1:m, m:1), CI.95, col = gray(.5), border = NA, lwd = 5)
polygon(c(1:m, m:1), CI.50, col=gray(.25), border = NA, lwd = 5)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.001, 0.999))),
        type = "1",
        lty = 2,
        col = 1,
        lwd = 9,
        add = TRUE)

lines(apply(dat$Y[-T, ], 2, mean), col = "salmon", lwd = 11)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.05, 0.95))),
        type = "1",
        lty = 3,
plot.swe.post_feb_LBsite <- function(dat, date = NULL, head = NULL) {
  # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
  # Working Directory)

  ## Setup Variables

  n.burn <- round(dat$n.mcmc/4)
  T <- dim(dat$Y)[1]
  m <- dim(dat$Y)[2]
  mos <- cumsum(modays)
  y.max <- max(dat$Y, na.rm = TRUE)
  y.max <- 33

  ## Get Prediction 50% and 95% Credible Intervals

  CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
  CI.95 <- c(CI.95[1, ], rev(CI.95[2, ]))
  CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
  CI.50 <- c(CI.50[1, ], rev(CI.50[2, ]))
## Plot the Graphics

```r
matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),
        type = "n",
        col = 3,
        lty = 1,
        ylab = "SWE",
        ylab = "",
        ylim = c(0, y.max),
        xlab = "Day (in water-year)",
        cex.lab = 1.65,
        main = paste(head),
        cex.main = 2.4,
        axes = FALSE)

axis(1, c(0, 92, 182, 273),
     c("Oct. 1", "Jan. 1", "Apr. 1", "Jul. 1"),
     cex.axis = 2.5, font = 2)

axis(2, cex.axis = 2.5, las = 0, tck = -0.023, line = -0.03, font = 2)
box()

abline(v = c(0, mos), col = 8)

polygon(c(1:m, m:1), CI.95, col = gray(.5), border = NA, lwd = 5)

polygon(c(1:m, m:1), CI.50, col = gray(.25), border = NA, lwd = 5)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.001, 0.999))),
        type = "l",
        lty = 2,
        col = 1,
        lwd = 9,
        add = TRUE)

lines(apply(dat$Y[-T, ], 2, mean), col = "salmon", lwd = 11)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.05, 0.95))),
        type = "l",
        lty = 3,
        # pch = 3,
        col = 1,
        lwd = 7,
        add = TRUE)

lines(dat$Y[T, ], lwd = 11, col = "orange")

abline(v = 130, col = "darkred", lty = 4, lwd = 10)

text(x = 130, y = 27, labels = "Feb. 7 --", pos = 2,

```
col = "darkred", cex = 2.6, font = 2)

# legend(x = c(283, 377), y = c(49, 72),
# col = c(1, gray(0.25), gray(0.5), "salmon", "orange", 8, 1),
# lty = c(2, 1, 1, 1, 1, 1, 3),
# cex = 1.3,
# lwd = c(4, 8, 8, 5, 5, 3, 4),
# legend = c("prev. data env.", "pred. CI: 50\%", "pred. CI: 95\%",
# "prev. data mean", "current data", "months",
# "5th & 95th perc.")
# bg = "white")

plot.swe.post_mar_LBsite <- function(dat, date = NULL, head = NULL){
    # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
    # Working Directory)
    ##
    ## Setup Variables
    ##
    n.burn <- round(dat$n.mcmc/4)
    T <- dim(dat$Y)[1]
    m <- dim(dat$Y)[2]
    mos <- cumsum(modays)
    y.max <- max(dat$Y, na.rm = TRUE)
    y.max <- 33
    ##
    ## Get Prediction 50\% and 95\% Credible Intervals
    ##
    CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
    CI.95 <- c(CI.95[1, ], rev(CI.95[2, ]))
    CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
    CI.50 <- c(CI.50[1, ], rev(CI.50[2, ]))
    ##
    ## Plot the Graphics
    ##
    matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),
            type = "n",
            col = 3,
            lty = 1,
            # ylab = "SWE",
            ylab = "",
            #
ylim = c(0, y.max),
# xlab = "Day (in water-year)",
# cex.lab = 1.65,
# main = paste(head),
# cex.main = 2.4,
# axes = FALSE)
axis(1, c(0, 92, 182, 273),
    c("Oct. 1", "Jan. 1", "Apr. 1", "Jul. 1"),
    cex.axis = 2.5, font = 2)
axis(2, cex.axis = 2.5, las = 0, tck = -0.023, line = -0.03, font = 2)
box()
abline(v = c(0, mos), col = 8)
text(mos - 15, rep(y.max, 12), labels = c(10:12, 1:9), cex = 2.3)
polygon(c(1:m, m:1), CI.95, col = gray(.5), border = NA, lwd = 5)
polygon(c(1:m, m:1), CI.50, col=gray(.25), border = NA, lwd = 5)
matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.001, 0.999))),
    type = "1",
    lty = 2,
    col = 1,
    lwd = 9,
    add = TRUE)
lines(apply(dat$Y[-T, ], 2, mean), col = "salmon", lwd = 11)
matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.05, 0.95))),
    type = "1",
    lty = 3,
    # pch = 3,
    col = 1,
    lwd = 7,
    add = TRUE)
lines(dat$Y[T, ], lwd = 11, col = "orange")
abline(v = 160, col = "darkred", lty = 4, lwd = 10)
text(x = 160, y = 27, labels = "Mar. 9 --", pos = 2,
    col = "darkred", cex = 2.6, font = 2)
# legend(x = c(283, 377), y = c(49, 72),
#     col = c(1, gray(0.25), gray(0.5), "salmon", "orange", 8, 1),
#     lty = c(2, 1, 1, 1, 1, 1, 3),
#     cex = 1.3,
#     lwd = c(4, 8, 8, 5, 5, 3, 4),
#     legend = c("prev. data env.", "pred. CI: 50%", "pred. CI: 95%",
#     "prev. data mean", "current data","months",
#     "5th & 95th perc."),
plot.swe.post_apr_LBsite <- function(dat, date = NULL, head = NULL){

  # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
  # Working Directory)

  ## Setup Variables
  ##
  n.burn <- round(dat$n.mcmc/4)
  T <- dim(dat$Y)[1]
  m <- dim(dat$Y)[2]
  mos <- cumsum(modays)
  y.max <- max(dat$Y, na.rm = TRUE)
  y.max <- 33

  ## Get Prediction 50% and 95% Credible Intervals
  ##
  CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
  CI.95 <- c(CI.95[1,], rev(CI.95[2,]))
  CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
  CI.50 <- c(CI.50[1,], rev(CI.50[2,]))

  ## Plot the Graphics
  ##
  matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),
          type = "n",
          col = 3,
          lty = 1,
          # ylab = "SWE",
          ylab = "",
          ylim = c(0, y.max),
          # xlab = "Day (in water-year)",
          # cex.lab = 1.65,
          # main = paste(head),
          # cex.main = 2.4,
          axes = FALSE)
  axis(1, c(0, 0, 92, 182, 273),
    c("Oct. 1", "Jan. 1", "Apr. 1", "Jul. 1"),
    cex.axis = 2.5, font = 2)
C.4.3 Functions for Horse Ridge SNOTEL Site

plot.swe.post_jan_Fig8_HRsite <- function(dat, date = NULL, head = NULL){

    # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
    # Working Directory)

    ## Setup Variables
    ##
    n.burn <- round(dat$n.mcmc/4)
    T <- dim(dat$Y)[1]
    m <- dim(dat$Y)[2]
    mos <- cumsum(modays)
    y.max <- max(dat$Y, na.rm = TRUE) + 1
    # y.max <- 75

    ## Get Prediction 50% and 95% Credible Intervals
    ##
    CI.95 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.025, 0.975))
    CI.95 <- c(CI.95[1, ], rev(CI.95[2, ]))
    CI.50 <- apply(dat$Y.pred[, n.burn:dat$n.mcmc], 1, quantile, c(0.25, 0.75))
    CI.50 <- c(CI.50[1, ], rev(CI.50[2, ]))

    ## Plot the Graphics
    ##
    matplot((dat$Y.pred[, sample(n.burn:dat$n.mcmc, 20)]),
            type = "n",
            col = 3,
            lty = 1,
            ylab = "SWE",
            ylab = "",
            ylim = c(0, y.max),
            xlab = "Day (in water-year)",
            cex.lab = 1.65,
            main = paste(head),
            cex.main = 2.4,
            axes = FALSE)

    axis(1, c(0, 92, 182, 273),
         c("Oct. 1", "Jan. 1", "Apr. 1", "Jul. 1"),
         cex.axis = 1.8)

    axis(2, cex.axis = 1.8, las = 0, tck = -0.023, line = -0.03)
    title(ylab = "SWE (inches)", line = 2.7, cex.lab = 1.8)
box()

abline(v = c(0, mos), col = 8)
text(mos - 15, rep(y.max, 12), labels = c(10:12, 1:9), cex = 1.6)

polygon(c(1:m, m:1), CI.95, col = gray(.5), border = NA, lwd = 5)
polygon(c(1:m, m:1), CI.50, col = gray(.25), border = NA, lwd = 5)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.001, 0.999))),
        type = "l",
        lty = 2,
        col = 1,
        lwd = 6,
        add = TRUE)

lines(apply(dat$Y[-T, ], 2, mean), col = "salmon", lwd = 11)

matplot(t(apply(dat$Y[-T, ], 2, quantile, c(0.05, 0.95))),
        type = "l",
        lty = 3,
        # pch = 3,
        col = 1,
        lwd = 7,
        add = TRUE)

lines(dat$Y[T, ], lwd = 11, col = "orange")

abline(v = 100, col = "darkred", lty = 4, lwd = 10)
text(x = 100, y = 62, labels = "Jan. 8, 2009 --", pos = 2,
     col = "darkred", cex = 2.5, font = 1)

legend(x = c(283, 377), y = c(49, 72),
       col = c(1, gray(0.25), gray(0.5), "salmon", "orange", 8, 1),
       lty = c(2, 1, 1, 1, 1, 1, 3),
       cex = 1.3,
       lwd = c(4, 8, 8, 5, 5, 3, 4),
       legend = c("prev. data env.", "pred. CI: 50%", "pred. CI: 95%",
                 "prev. data mean", "current data","months",
                 "5th & 95th perc."),
       bg = "white")
}
C.5 Function to Determine Proportion Of SWE Measurements Outside the
Credible Intervals

prop.pred <- function(tmpswe, n = 10, outdat = 100, ncol = 6){
  # Uses Output from swe.svd.mcmc (SWE SVD-MCMC Function Stored in R
  # Working Directory)
  #
  # n is the number of selected sites

  ## Setup Variables
  prop.mat <- matrix(NA, n, ncol)
colnames(prop.mat) <- c("50%CL.LL", "50%CL.UL", "50%CL.LLUL",
    "95%CL.LL", "95%CL.UL", "95%CL.LLUL")
rownames(prop.mat) <- c("TGswe", "BLPswe", "HRswe", "LBswe", "BLswe",
    "DBPswe", "MCswe", "FARMswe", "TDswe", "PSswe")

  ## Get Prediction Credible Intervals and Proportion of SWE measurements
  ## Outside these Intervals
  for (i in 1:n) {
    n.burn <- round(tmpswe[[i]]$n.mcmc/4)
    T <- dim(tmpswe[[i]]$Y)[1]
    m <- dim(tmpswe[[i]]$Y)[2]

    CI.50 <- apply(tmpswe[[i]]$Y.pred[, n.burn:tmpswe[[i]]$n.mcmc], 1,
      quantile, c(0.25, 0.75))
    CI.95 <- apply(tmpswe[[i]]$Y.pred[, n.burn:tmpswe[[i]]$n.mcmc], 1,
      quantile, c(0.025, 0.975))

    nc <- c(1:ncol)
    prop.mat[i, nc[1]] <- 100 * sum(tmpswe[[i]]$Y[T, (outdat + 1):m] <
      CI.50[1, (outdat + 1):m])/(m - outdat)
    prop.mat[i, nc[2]] <- 100 * sum(tmpswe[[i]]$Y[T, (outdat + 1):m] >
      CI.50[2, (outdat + 1):m])/(m - outdat)
    prop.mat[i, nc[3]] <- 100 * sum((tmpswe[[i]]$Y[T, (outdat + 1):m] <
      CI.50[1, (outdat + 1):m]), (tmpswe[[i]]$Y[T,
      (outdat + 1):m] > CI.50[2, (outdat + 1):m]))/(m - outdat)
    prop.mat[i, nc[4]] <- 100 * sum(tmpswe[[i]]$Y[T, (outdat + 1):m] <
      CI.95[1, (outdat + 1):m])/(m - outdat)
    prop.mat[i, nc[5]] <- 100 * sum(tmpswe[[i]]$Y[T, (outdat + 1):m] >
      CI.95[2, (outdat + 1):m])/(m - outdat)
    prop.mat[i, nc[6]] <- 100 * sum((tmpswe[[i]]$Y[T, (outdat + 1):m] <
      CI.95[1, (outdat + 1):m]),(tmpswe[[i]]$Y[T,
      (outdat + 1):m] > CI.95[2, (outdat + 1):m]))/(m - outdat)
  }
}
C.6 Function to Estimate Missing SWE Measurements Using Kriging

**krig.swe** <- function(Y, X){

library(geoR)

T <- dim(Y)[2]
n <- dim(Y)[3]
m <- dim(Y)[1]

Y.aug <- Y
for (t in 1:T) {
  for (i in 1:m) {
    #cat(t,i,"
    tmp.idx <- !is.na(Y[i, t, ])
    if (!all(tmp.idx)) {
      yo.tmp <- Y[i, t, tmp.idx]
      xo.tmp <- X[tmp.idx, ]
      xu.tmp <- X[!tmp.idx, ]
      tmp.dat <- cbind(xo.tmp[, 1], xo.tmp[, 2], yo.tmp)
      tmp.pred.dat <- cbind(xu.tmp[, 1], xu.tmp[, 2])
      o.gd <- as.geodata(tmp.dat)
      tmp.v <- variog(o.gd, messages = FALSE)
      do.idx <- (max(tmp.v$v) >= .5)
      if (do.idx) {
        # tmp.f <- variofit(tmp.v, ini = c(max(tmp.v$v), 1),
        # cov.model = "exponential", fix.nug = TRUE,
        # wei = "cressie", max.dist = 2, messages = FALSE)
        tmp.f <- variofit(tmp.v, ini = c(max(tmp.v$v), 1),
                         cov.model = "exponential", fix.nug = TRUE,
                         wei = "cressie", messages = FALSE)
        tmp.k <- krige.conv(o.gd, loc = xu.tmp,
                             krig = krige.control(type.krig = "SK", obj.m = tmp.f),
                             output = output.control(messages = FALSE))
        yu.tmp <- tmp.k$predict
      }
      if (!do.idx) {
        Y.aug[i, t, ] <- 0
      }
    }
  }
  }
}
Y.aug[Y.aug < 0] <- 0
Y.aug

C.7 Function to Download Utah SNOTEL Historical SWE Measurements for 90 Sites

utah.snotel.historical.data <- function(n.sites, current.year){

  ## Example:
  ## swe.utah <- utah.snotel.historical.data(n.sites = 90, current.year = 2009)

  u <- rep(NA, n.sites)

  u[1] <- "Historical_Data/12m26s_s.txt"
  u[2] <- "Historical_Data/11k13s_s.txt"
  u[3] <- "Historical_Data/11j46s_s.txt"
  u[4] <- "Historical_Data/11h08s_s.txt"
  u[5] <- "Historical_Data/11h30s_s.txt"
  u[6] <- "Historical_Data/12l07s_s.txt"
  u[7] <- "Historical_Data/11l04s_s.txt"
  u[8] <- "Historical_Data/12l04s_s.txt"
  u[9] <- "Historical_Data/11j57s_s.txt"
  u[10] <- "Historical_Data/10j30s_s.txt"
  u[11] <- "Historical_Data/11k31s_s.txt"
  u[12] <- "Historical_Data/11h37s_s.txt"
  u[13] <- "Historical_Data/09m02s_s.txt"
  u[14] <- "Historical_Data/11j70s_s.txt"
  u[15] <- "Historical_Data/12m13s_s.txt"
  u[16] <- "Historical_Data/11j01s_s.txt"
  u[17] <- "Historical_Data/11j02s_s.txt"
  u[18] <- "Historical_Data/10j43s_s.txt"
  u[19] <- "Historical_Data/11m06s_s.txt"
  u[20] <- "Historical_Data/11k21s_s.txt"
  u[21] <- "Historical_Data/11k22s_s.txt"
  u[22] <- "Historical_Data/11j32s_s.txt"
  u[23] <- "Historical_Data/11j23s_s.txt"
  u[24] <- "Historical_Data/11k15s_s.txt"
  u[25] <- "Historical_Data/11l05s_s.txt"
  u[26] <- "Historical_Data/11h55s_s.txt"
  u[27] <- "Historical_Data/12j09s_s.txt"
  u[28] <- "Historical_Data/09k01s_s.txt"
  u[29] <- "Historical_Data/11j11s_s.txt"
  u[30] <- "Historical_Data/11j12s_s.txt"
  u[31] <- "Historical_Data/11l01s_s.txt"
  u[32] <- "Historical_Data/10j26s_s.txt"
u[84] <- "Historical_Data/10j52s_s.txt"
u[85] <- "Historical_Data/09j16s_s.txt"
u[86] <- "Historical_Data/11h60s_s.txt"
u[87] <- "Historical_Data/12k01s_s.txt"
u[88] <- "Historical_Data/12m03s_s.txt"
u[89] <- "Historical_Data/10k02s_s.txt"
u[90] <- "Historical_Data/11m03s_s.txt"

s <- rep(NA, n.sites)

s[1] <- 1995
s[3] <- 1979
s[4] <- 1979
s[8] <- 1980
s[9] <- 1987
s[10] <- 1979
s[12] <- 1979
s[14] <- 2003
s[16] <- 1979
s[17] <- 1979
s[18] <- 1982
s[19] <- 2001
s[20] <- 1980
s[21] <- 1980
s[22] <- 1979
s[23] <- 1979
s[25] <- 1986
s[26] <- 1979
s[27] <- 1995
s[28] <- 1986
s[29] <- 1979
s[30] <- 2004
s[31] <- 1981
s[32] <- 1982
s[33] <- 2005
s[34] <- 1981 # missing water-years: 1979 and 1981
s[35] <- 2005
s[36] <- 1994
s[37] <- 1980
s[38] <- 1979
s[39] <- 1986
s[40] <- 1986
s[41] <- 1986
| $s[42]$ | 1979 |
| $s[43]$ | 1980 |
| $s[44]$ | 2008 |
| $s[45]$ | 1981 |
| $s[46]$ | 1980 |
| $s[47]$ | 1980 |
| $s[48]$ | 1980 |
| $s[49]$ | 1990 |
| $s[50]$ | 1982 |
| $s[51]$ | 2005 |
| $s[52]$ | 1982 |
| $s[53]$ | 1979 |
| $s[54]$ | 1986 |
| $s[55]$ | 1982 |
| $s[56]$ | 1986 |
| $s[57]$ | 1988 |
| $s[58]$ | 2000 |
| $s[59]$ | 1980 |
| $s[60]$ | 1981 |
| $s[61]$ | 1982 |
| $s[62]$ | 1989 |
| $s[63]$ | 1989 |
| $s[64]$ | 1979 |
| $s[65]$ | 1979 |
| $s[66]$ | 1979 |
| $s[67]$ | 2000 |
| $s[68]$ | 1981 |
| $s[69]$ | 1979 |
| $s[70]$ | 1986 |
| $s[71]$ | 1981 # missing water-years: 1979 and 1981 |
| $s[72]$ | 1981 |
| $s[73]$ | 1982 |
| $s[74]$ | 1982 |
| $s[75]$ | 1979 |
| $s[76]$ | 1990 |
| $s[77]$ | 1979 |
| $s[78]$ | 1979 |
| $s[79]$ | 2002 |
| $s[80]$ | 2008 |
| $s[81]$ | 1988 |
| $s[82]$ | 1979 |
| $s[83]$ | 1979 |
| $s[84]$ | 1979 |
| $s[85]$ | 1980 |
| $s[86]$ | 2007 |
| $s[87]$ | 1979 |
| $s[88]$ | 1981 |
| $s[89]$ | 1980 # missing water-year: 1983 |
| $s[90]$ | 1979 |
#source("read.snotel.R")
snotel.list <- vector("list", length = n.sites)
T.vec <- rep(0, n.sites)
idx.no0 <- -c(62, 153, 154, 155, 217, 279, 372)

for (i in 1:n.sites) {
  #cat(i, "\n")
  T.vec[i] <- current.year - s[i]
  swe.mat <- matrix(NA, 372, T.vec[i])
  data <- u[i]
  for (j in 1:T.vec[i]) {
    #cat(j, " ")
    swe.mat[, j] <- read.snotel(f = data, j)
    #cat("\n")
  }
  snotel.list[[i]] <- swe.mat[idx.no0, ]
  #cat("\n")
}
snotel.list

C.8 Function to Download Utah SNOTEL Current SWE Measurements for 90 Sites

utah.snotel.current.data <- function(n.sites){
  ## Example: swe.utah.current <- utah.snotel.current.data(n.sites = 90)

  u <- rep(NA, n.sites)

  u[1] <- "Current_Data/12m26s_s.txt"
  u[2] <- "Current_Data/11k13s_s.txt"
  u[3] <- "Current_Data/11j46s_s.txt"
  u[4] <- "Current_Data/11h08s_s.txt"
  u[5] <- "Current_Data/11h30s_s.txt"
  u[6] <- "Current_Data/12107s_s.txt"
  u[7] <- "Current_Data/11104s_s.txt"
  u[8] <- "Current_Data/12104s_s.txt"
  u[9] <- "Current_Data/11j57s_s.txt"
  u[10] <- "Current_Data/10j30s_s.txt"
  u[11] <- "Current_Data/11k31s_s.txt"
  u[12] <- "Current_Data/11h37s_s.txt"
  u[13] <- "Current_Data/09m02s_s.txt"
  u[14] <- "Current_Data/11j70s_s.txt"
  u[15] <- "Current_Data/12m13s_s.txt"
  u[16] <- "Current_Data/11j01s_s.txt"
  u[17] <- "Current_Data/11j02s_s.txt"
u[18] <- "Current_Data/10j43s_s.txt"
u[19] <- "Current_Data/11m06s_s.txt"
u[20] <- "Current_Data/11k21s_s.txt"
u[21] <- "Current_Data/11k22s_s.txt"
u[22] <- "Current_Data/11j32s_s.txt"
u[23] <- "Current_Data/11j23s_s.txt"
u[24] <- "Current_Data/11k15s_s.txt"
u[25] <- "Current_Data/11105s_s.txt"
u[26] <- "Current_Data/11h55s_s.txt"
u[27] <- "Current_Data/12j09s_s.txt"
u[28] <- "Current_Data/09k01s_s.txt"
u[29] <- "Current_Data/11j11s_s.txt"
u[30] <- "Current_Data/11j12s_s.txt"
u[31] <- "Current_Data/11101s_s.txt"
u[32] <- "Current_Data/10j26s_s.txt"
u[33] <- "Current_Data/13m07s_s.txt"
u[34] <- "Current_Data/11112s_s.txt"
u[35] <- "Current_Data/13m06s_s.txt"
u[36] <- "Current_Data/11j37s_s.txt"
u[37] <- "Current_Data/12m05s_s.txt"
u[38] <- "Current_Data/10j44s_s.txt"
u[39] <- "Current_Data/10j04s_s.txt"
u[40] <- "Current_Data/09j08s_s.txt"
u[41] <- "Current_Data/10j01s_s.txt"
u[42] <- "Current_Data/11h21s_s.txt"
u[43] <- "Current_Data/10k01s_s.txt"
u[44] <- "Current_Data/12120s_s.txt"
u[45] <- "Current_Data/12106s_s.txt"
u[46] <- "Current_Data/09j01s_s.txt"
u[47] <- "Current_Data/13m05s_s.txt"
u[48] <- "Current_Data/10j10s_s.txt"
u[49] <- "Current_Data/10j25s_s.txt"
u[50] <- "Current_Data/09103s_s.txt"
u[51] <- "Current_Data/11h59s_s.txt"
u[52] <- "Current_Data/10j35s_s.txt"
u[53] <- "Current_Data/11h25s_s.txt"
u[54] <- "Current_Data/13m04s_s.txt"
u[55] <- "Current_Data/13m02s_s.txt"
u[56] <- "Current_Data/12m06s_s.txt"
u[57] <- "Current_Data/11j64s_s.txt"
u[58] <- "Current_Data/11j69s_s.txt"
u[59] <- "Current_Data/11k03s_s.txt"
u[60] <- "Current_Data/12112s_s.txt"
u[61] <- "Current_Data/12m23s_s.txt"
u[62] <- "Current_Data/11j65s_s.txt"
u[63] <- "Current_Data/12j07s_s.txt"
u[64] <- "Current_Data/11h57s_s.txt"
u[65] <- "Current_Data/09j05s_s.txt"
u[66] <- "Current_Data/11j52s_s.txt"
u[67] <- "Current_Data/11j68s_s.txt"
u[68] <- "Current_Data/11k52s_s.txt"
u[69] <- "Current_Data/11k39s_s.txt"
u[70] <- "Current_Data/12115s_s.txt"
u[71] <- "Current_Data/11k28s_s.txt"
u[72] <- "Current_Data/10j18s_s.txt"
u[73] <- "Current_Data/12j06s_s.txt"
u[74] <- "Current_Data/11k09s_s.txt"
u[75] <- "Current_Data/11j53s_s.txt"
u[76] <- "Current_Data/11j42s_s.txt"
u[77] <- "Current_Data/10j20s_s.txt"
u[78] <- "Current_Data/11j08s_s.txt"
u[79] <- "Current_Data/11h58s_s.txt"
u[80] <- "Current_Data/10k06s_s.txt"
u[81] <- "Current_Data/11j56s_s.txt"
u[82] <- "Current_Data/11j21s_s.txt"
u[83] <- "Current_Data/11h36s_s.txt"
u[84] <- "Current_Data/10j52s_s.txt"
u[85] <- "Current_Data/09j16s_s.txt"
u[86] <- "Current_Data/11h60s_s.txt"
u[87] <- "Current_Data/12k01s_s.txt"
u[88] <- "Current_Data/12m03s_s.txt"
u[89] <- "Current_Data/10k02s_s.txt"
u[90] <- "Current_Data/11m03s_s.txt"

#source("read.snotel.R")
snotel.list <- vector("list", length = n.sites)
idx.no0 <- -c(62, 153, 154, 155, 217, 279, 372)

for (i in 1:n.sites) {
  #cat(i, "\n")
  data <- u[i]
  swe.mat <- read.snotel(f = data)
  #cat("\n")
  snotel.list[[i]] <- swe.mat[idx.no0]
  #cat("\n")
}
snotel.list
APPENDIX D

PROCEDURAL PARADIGM #2 (R CODE)

In this appendix, we provide readable and structured R codes used to produce the figures and tables in Chapter 2.

D.1 To Import Needed Functions and Data

## IMPORTANT: Users Must Make Sure To Set Working Directory To Read The Data
## Files and Source Codes
##
## For example:
## setwd('/Users/jamesbeg/Desktop/....')
##
## The Following Functions As Well As Libraries Should Be Installed

source("read.snotel.R")
source("invgammastrt.R")
source("swe.svd.mcmc.R")
source("plot.swe.post.R")
source("prop.pred.R")
source("krig.swe.R")
source("utah.snotel.historical.data.R")
source("utah.snotel.current.data.R")

source("plot.swe.post_jan_Fig8_TGsite.R")
source("plot.swe.post_jan_TGsite.R")
source("plot.swe.post_feb_TGsite.R")
source("plot.swe.post_mar_TGsite.R")
source("plot.swe.post_apr_TGsite.R")
source("plot.swe.post_jan_LBsite.R")
source("plot.swe.post_feb_LBsite.R")
source("plot.swe.post_mar_LBsite.R")
source("plot.swe.post_apr_LBsite.R")
source("plot.swe.post_jan_Fig8_HRsite.R")

require(MCMCpack)
require(mvtnorm)
require(spdep)
require(msm)
require(lattice)
require(geoR)
require(maps)
require(grid)
n.sites <- 90  ## Number of Active Utah SNOTEL Sites
m <- 365  ## Number of days in a Water-Year
T <- 34  ## Number of Historical Water-Years considered (1979 to 2012)

## 1. Read The Latitude And Longitude Coordinates Of All The 90 Active
##  UTAH SNOTEL Sites in txt format ("xcoord.txt")
LatLonCoord <- read.table("xcoord.txt", header = TRUE)

## 2. Read The Already Downloaded Historical And Current Water-Year SWE
##  txt Files For The 90 Active UTAH SNOTEL Sites

## Historical SWE Data from 1979 to 2012 Water-Years
past.yr <- 2012
swe.utah <- utah.snotel.historical.data(n.sites = 90, current.year =
    past.yr + 1)

## Current Year SWE Data (2013 Water-Year)
swe.utah.current <- utah.snotel.current.data(n.sites = 90)

## 3. Augmenting The SWE Data (Y)

## Original Historical SWE Data With Missing Values
Y <- array(NA, c(m, T, n.sites))

Y[, 17:T, 1] <- swe.utah[[1]]
Y[, 3:T, 2] <- swe.utah[[2]]
Y[, 1:T, 3] <- swe.utah[[3]]
Y[, 1:T, 4] <- swe.utah[[4]]
Y[, 3:T, 5] <- swe.utah[[5]]
Y[, 2:T, 6] <- swe.utah[[6]]
Y[, 4:T, 7] <- swe.utah[[7]]
Y[, 2:T, 8] <- swe.utah[[8]]
Y[, 9:T, 9] <- swe.utah[[9]]
Y[, 1:T, 10] <- swe.utah[[10]]
Y[, 2:T, 11] <- swe.utah[[11]]
Y[, 1:T, 12] <- swe.utah[[12]]
Y[, 8:T, 13] <- swe.utah[[13]]
Y[, 25:T, 14] <- swe.utah[[14]]
Y[, 3:T, 15] <- swe.utah[[15]]
Y[, 1:T, 16] <- swe.utah[[16]]
Y[, 1:T, 17] <- swe.utah[[17]]
Y[, 4:T, 18] <- swe.utah[[18]]
Y[, 23:T, 19] <- swe.utah[[19]]
Y[, 2:T, 20] <- swe.utah[[20]]
Y[, 2:T, 21] <- swe.utah[[21]]
Y[, 1:T, 22] <- swe.utah[[22]]
Y[, 1:T, 23] <- swe.utah[[23]]
Y[, 1, 24] <- swe.utah[[24]][, 1]
Y[, 3:T, 24] <- swe.utah[[24]][, -1]

Y[, 8:T, 25] <- swe.utah[[25]]
Y[, 1:T, 26] <- swe.utah[[26]]
Y[, 17:T, 27] <- swe.utah[[27]]
Y[, 8:T, 28] <- swe.utah[[28]]
Y[, 1:T, 29] <- swe.utah[[29]]
Y[, 26:T, 30] <- swe.utah[[30]]
Y[, 3:T, 31] <- swe.utah[[31]]
Y[, 4:T, 32] <- swe.utah[[32]]
Y[, 27:T, 33] <- swe.utah[[33]]

Y[, 2, 34] <- swe.utah[[34]][, 1]
Y[, 4:T, 34] <- swe.utah[[34]][, -1]

Y[, 27:T, 35] <- swe.utah[[35]]
Y[, 16:T, 36] <- swe.utah[[36]]
Y[, 2:T, 37] <- swe.utah[[37]]
Y[, 1:T, 38] <- swe.utah[[38]]
Y[, 8:T, 39] <- swe.utah[[39]]
Y[, 8:T, 40] <- swe.utah[[40]]
Y[, 8:T, 41] <- swe.utah[[41]]
Y[, 1:T, 42] <- swe.utah[[42]]
Y[, 2:T, 43] <- swe.utah[[43]]
Y[, 30:T, 44] <- swe.utah[[44]]
Y[, 3:T, 45] <- swe.utah[[45]]
Y[, 2:T, 46] <- swe.utah[[46]]
Y[, 2:T, 47] <- swe.utah[[47]]
Y[, 2:T, 48] <- swe.utah[[48]]
Y[, 12:T, 49] <- swe.utah[[49]]
Y[, 4:T, 50] <- swe.utah[[50]]
Y[, 27:T, 51] <- swe.utah[[51]]
Y[, 4:T, 52] <- swe.utah[[52]]
Y[, 1:T, 53] <- swe.utah[[53]]
Y[, 8:T, 54] <- swe.utah[[54]]
Y[, 4:T, 55] <- swe.utah[[55]]
Y[, 8:T, 56] <- swe.utah[[56]]
Y[, 10:T, 57] <- swe.utah[[57]]
Y[, 22:T, 58] <- swe.utah[[58]]
Y[, 2:T, 59] <- swe.utah[[59]]
Y[, 3:T, 60] <- swe.utah[[60]]
Y[, 4:T, 61] <- swe.utah[[61]]
Y[, 11:T, 62] <- swe.utah[[62]]
Y[, 11:T, 63] <- swe.utah[[63]]
Y[, 1:T, 64] <- swe.utah[[64]]
Y[, 1:T, 65] <- swe.utah[[65]]
Y[, 1:T, 66] <- swe.utah[[66]]
Y[, 22:T, 67] <- swe.utah[[67]]
Y[, 3:T, 68] <- swe.utah[[68]]
Y[, 1:T, 69] <- swe.utah[[69]]
Y[, 8:T, 70] <- swe.utah[[70]]

Y[, 2, 71] <- swe.utah[[71]][, 1]
Y[, 4:T, 71] <- swe.utah[[71]][, -1]

Y[, 3:T, 72] <- swe.utah[[72]]
Y[, 4:T, 73] <- swe.utah[[73]]
Y[, 4:T, 74] <- swe.utah[[74]]
Y[, 1:T, 75] <- swe.utah[[75]]
Y[, 12:T, 76] <- swe.utah[[76]]
Y[, 1:T, 77] <- swe.utah[[77]]
Y[, 1:T, 78] <- swe.utah[[78]]
Y[, 24:T, 79] <- swe.utah[[79]]
Y[, 30:T, 80] <- swe.utah[[80]]
Y[, 10:T, 81] <- swe.utah[[81]]
Y[, 1:T, 82] <- swe.utah[[82]]

Y[, 1:T, 83] <- swe.utah[[83]]
Y[61, 30, 83] <- 1.5

Y[, 1:T, 84] <- swe.utah[[84]]
Y[, 2:T, 85] <- swe.utah[[85]]
Y[, 29:T, 86] <- swe.utah[[86]]
Y[, 1:T, 87] <- swe.utah[[87]]
Y[, 3:T, 88] <- swe.utah[[88]]

Y[, 1:4, 89] <- swe.utah[[89]][, 1:4]
Y[, 6:T, 89] <- swe.utah[[89]][, -(1:4)]

Y[, 1:T, 90] <- swe.utah[[90]]

## Augmenting Historical SWE Data With Missing Values Using Kriging
Y.aug <- krig.swe(Y, LatLonCoord)

D.2 Creating Figure 2.1

## Figure 2.1: Map Of UTAH With The Highlighted SNOTEL Sites

layout(matrix(1:2, 1, 2))

map("state", "utah", interior = FALSE, ylim = c(35.8, 44), lwd = 2.5,
     mar=c(0, 1, 0, 0))
map("county", "utah", boundary = FALSE, lty = 2, add = TRUE)
points(LatLonCoord[-c(4, 12, 26, 29, 42, 53, 64, 66, 82, 83), ],
       pch = 16,
       cex = 1.3,
       col = "gray55")
points(LatLonCoord[c(4, 12, 26, 29, 42, 53, 64, 66, 82), ],
       pch = 16,
pch = 18,
cex = 1.6,
col = 1)

points(LatLonCoord[83, ],
pch = 17,
cex = 1.2,
col = 1)

loc <- LatLonCoord[c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66), ]
lonadd <- c(-0.06, -0.07, 0.08, 0.06, 0.08, -0.07, 0.08, 0.08, 0.08, 0.10)
latadd <- c(0, 0, 0, 0.02, 0, 0, 0, 0, 0, 0)

map.scale(relwidth = 0.5, metric = FALSE, font = 2)

rect(xleft = -112.2,
ybottom = 40,
xright = -110.8,
ytop = 42.1,
border = "steelblue4",
lwd = 3)

map("state","utah", interior = FALSE,
xlim = c(-112.2, -110.8),
ylim = c(40, 42.3),
lwd = 2.5,
mar=c(0, 0, 0, 0))

map("county", "utah", boundary = FALSE, lty = 2, add = TRUE)

points(LatLonCoord[-c(4, 12, 26, 29, 42, 53, 64, 66, 82, 83), ],
pch = 16,
cex = 1.6,
col = "gray55")

points(LatLonCoord[c(4, 12, 26, 29, 42, 53, 64, 66, 82), ],
pch = 18,
cex = 1.6,
col = 1)

points(LatLonCoord[83, ],
pch = 17,
cex = 1.2,
col = 1)

loc <- LatLonCoord[c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66), ]
lonadd <- c(-0.06, -0.07, 0.08, 0.06, 0.08, -0.07, 0.08, 0.08, 0.08, 0.10)
latadd <- c(0, 0, 0, 0.02, 0, 0, 0, 0, 0, 0)
text(loc[, 1] + lonadd, loc[, 2] - latadd, cex = 1.7, lwd = 1, col = 1)

rect(xleft = -112.2,
D.3 Creating Figure 2.2

## Figure 2.2: SWE Measurements For TONY GROVE SNOTEL Site (1979 to 2008)

TGdata <- swe.utah[[83]]  # TONY GROVE SITE
TGdata[61, 30] <- 1.5
T <- 2008 - 1979 + 1
TGdata <- TGdata[, 1:T]
day.mat <- matrix(rep(1:365, T))
year.mat <- matrix(1979:2008, m, T, byrow = TRUE)
year.mat2 <- matrix(NA, m, T)

for (i in 1:5){
  year.mat2[, (6*i - 5):(6*i)] = year.mat[, (25 - 6*(i - 1)):30 - 6*(i - 1))]
}

year.series <- as.factor(matrix(year.mat2))
TGdata2 <- data.frame(rev(TGdata), rev(day.mat), rev(year.series))
colnames(TGdata2) <- c("SWE", "Day", "Year")


my.strip <- function(which.given, which.panel, ...) {
  strip.labels <- labels
  panel.rect(x = 0.5, y = 0.5, adj = c(0.5, 0.55), cex = 1.2,
...
lwd = 2,
font = 2,
lab = strip.labels(which.panel[which.given]))
}

print(xyplot(SWE ~ Day | Year, data = TGdata2,
scales = list(alternating = c(1), tck = c(1.0, 0), cex = 1.4),
strip = my.strip,
outer = TRUE,
layout = c(6, 5),
ylab = list("SWE", cex = 1.5),
xlab = list("Day", cex = 1.8),
main = "\n",
panel = function(x, y, ...){
  panel.xyplot(x, y, ..., type = "l", col = "black",
pch = 16, lwd = 2)
}))

D.4 Combining Augmented Historical and Current SWE Data and Selecting the Ten SNOTEL Sites

```r
## 4. Combining the Augmented Historical SWE Data and Current SWE Data For Each UTAH SNOTEL Site In A List Form

swe.utah.list <- vector("list", length = n.sites)
for (i in 1:n.sites){
  #cat(i, "\n")
  swe.utah.list[[i]] <- cbind(Y.aug[, , i], as.matrix(swe.utah.current[[i]]))
  #cat("\n")
}
```

```r
## 5. Obtaining The SWE Measurements For The Ten Selected UTAH SNOTEL Sites For Model Implementation Purposes

TGswe <- swe.utah.list[[83]]  ## TONY GROVE
TGswe[61, 30] <- 1.5
BLPswe <- swe.utah.list[[4]]  ## BEN LOMOND PEAK
HRswe <- swe.utah.list[[42]]  ## HORSE RIDGE
LBswe <- swe.utah.list[[53]]  ## LITTLE BEAR
BLSwe <- swe.utah.list[[12]]  ## BUG LAKE
DBPswe <- swe.utah.list[[26]]  ## DRY BREAD POND
MCswe <- swe.utah.list[[9]]   ## MONTE CRISTO
```
D.5 Performing Singular Value Decomposition and Creating Table 2.1

## 6. Creating The Data Matrix And Performing The SVD For the Dimension Reduction Described In Section 2.2.2

n <- n.sites  
T <- 30  
## This Value is Dependent on the Past Water-Years. For Example,  
## in 2008 Water-Year, T = 29, For 2009 Water-Year, T = 30,  
## For 2010 Water-Year, T = 31, For 2011 Water-Year, T = 32, etc.

## Creating The Data Matrix Described In Section 2.2.2
Ymat <- matrix(Y.aug[, 1:T], m, n*T)

## Scaling The Data Matrix Described In Section 2.2.2 (Equation 2.1)
Y.c2 <- scale(t(Ymat), scale = FALSE)
mu2 <- attributes(Y.c2)[[2]]

M <- matrix(mu2, m, n*T)

## Performing The SVD On The Scaled Data Matrix Described In Equation 2.1
Y.svd <- svd(Y.c2)
U <- Y.svd$u
D <- diag(Y.svd$d)
V <- Y.svd$v
UD <- U%*%D

## Writing the Table for Cumulative Proportion Of Variance Explained By the Orthogonal Components of Each Decomposition (Signal) As .csv File

## Proportion Of Variance Explained By The Orthogonal Components of Each Decomposition (Signal)
pvar <- ((Y.svd$d)^2)/sum((Y.svd$d)^2)

## Cumulative Proportion Of Variance Explained By The Orthogonal Components of Each Decomposition (Signal). See Table 2.1 For The First Six

cumpvar <- cumsum(pvar)
Table1_full <- data.frame(cumpvar)
write.csv(Table1_full, file="Table1.csv"
D.6 Choosing a Reasonable q-Value for Dimension Reduction

## 7. Choosing A Reasonable q-Value (from 2, 4, 6) for the Dimension Reduction Described In Section 2.2.2

```r
set.seed(12345)

tmp2008.out1 <- swe.svd.mcmc.2(t(TGswe[, -c(31, 32, 33, 34, 35)]), 2, 1000, 100, )
tmp2008.out2 <- swe.svd.mcmc.2(t(TGswe[, -c(31, 32, 33, 34, 35)]), 4, 1000, 100, )
tmp2008.out3 <- swe.svd.mcmc.2(t(TGswe[, -c(31, 32, 33, 34, 35)]), 6, 1000, 100, )

tmp2008.out4 <- swe.svd.mcmc.2(t(BLPswe[, -c(31, 32, 33, 34, 35)]), 2, 1000, 100, )
tmp2008.out5 <- swe.svd.mcmc.2(t(BLPswe[, -c(31, 32, 33, 34, 35)]), 4, 1000, 100, )
tmp2008.out6 <- swe.svd.mcmc.2(t(BLPswe[, -c(31, 32, 33, 34, 35)]), 6, 1000, 100, )

tmp2008.out7 <- swe.svd.mcmc.2(t(HRswe[, -c(31, 32, 33, 34, 35)]), 2, 1000, 100, )
tmp2008.out8 <- swe.svd.mcmc.2(t(HRswe[, -c(31, 32, 33, 34, 35)]), 4, 1000, 100, )
tmp2008.out9 <- swe.svd.mcmc.2(t(HRswe[, -c(31, 32, 33, 34, 35)]), 6, 1000, 100, )

tmp2008.out10 <- swe.svd.mcmc.2(t(LBswe[, -c(31, 32, 33, 34, 35)]), 2, 1000, 100, )
tmp2008.out11 <- swe.svd.mcmc.2(t(LBswe[, -c(31, 32, 33, 34, 35)]), 4, 1000, 100, )
tmp2008.out12 <- swe.svd.mcmc.2(t(LBswe[, -c(31, 32, 33, 34, 35)]), 6, 1000, 100, )

tmp2008.out13 <- swe.svd.mcmc.2(t(BLswe[, -c(31, 32, 33, 34, 35)]), 2, 1000, 100, )
tmp2008.out14 <- swe.svd.mcmc.2(t(BLswe[, -c(31, 32, 33, 34, 35)]), 4, 1000, 100, )
tmp2008.out15 <- swe.svd.mcmc.2(t(BLswe[, -c(31, 32, 33, 34, 35)]), 6, 1000, 100, )

tmp2008.out16 <- swe.svd.mcmc.2(t(DBPswe[, -c(31, 32, 33, 34, 35)]), 2, 1000, 100, )
tmp2008.out17 <- swe.svd.mcmc.2(t(DBPswe[, -c(31, 32, 33, 34, 35)]), 4, 1000, 100, )
tmp2008.out18 <- swe.svd.mcmc.2(t(DBPswe[, -c(31, 32, 33, 34, 35)]), 6, 1000, 100, )

tmp2008.out19 <- swe.svd.mcmc.2(t(MCswe[, -c(31, 32, 33, 34, 35)]), 2, 1000, 100, )
```
tmp2008.out20 <- swe.svd.mcmc.2(t(MCswe[, -c(31, 32, 33, 34, 35)]), 4, 1000, 100, )
tmp2008.out21 <- swe.svd.mcmc.2(t(MCswe[, -c(31, 32, 33, 34, 35)]), 6, 1000, 100, )
tmp2008.out22 <- swe.svd.mcmc.2(t(FARMswe[, -c(31, 32, 33, 34, 35)]), 2, 1000, 100, )
tmp2008.out23 <- swe.svd.mcmc.2(t(FARMswe[, -c(31, 32, 33, 34, 35)]), 4, 1000, 100, )
tmp2008.out24 <- swe.svd.mcmc.2(t(FARMswe[, -c(31, 32, 33, 34, 35)]), 6, 1000, 100, )
tmp2008.out25 <- swe.svd.mcmc.2(t(TDswe[, -c(31, 32, 33, 34, 35)]), 2, 1000, 100, )
tmp2008.out26 <- swe.svd.mcmc.2(t(TDswe[, -c(31, 32, 33, 34, 35)]), 4, 1000, 100, )
tmp2008.out27 <- swe.svd.mcmc.2(t(TDswe[, -c(31, 32, 33, 34, 35)]), 6, 1000, 100, )
tmp2008.out28 <- swe.svd.mcmc.2(t(PSswe[, -c(31, 32, 33, 34, 35)]), 2, 1000, 100, )
tmp2008.out29 <- swe.svd.mcmc.2(t(PSswe[, -c(31, 32, 33, 34, 35)]), 4, 1000, 100, )
tmp2008.out30 <- swe.svd.mcmc.2(t(PSswe[, -c(31, 32, 33, 34, 35)]), 6, 1000, 100, )

n <- 30
tmpswe <- list(NA, n)
tmpswe[[1]] <- tmp2008.out1
tmpswe[[2]] <- tmp2008.out2
tmpswe[[3]] <- tmp2008.out3
tmpswe[[4]] <- tmp2008.out4
tmpswe[[5]] <- tmp2008.out5
tmpswe[[6]] <- tmp2008.out6
tmpswe[[7]] <- tmp2008.out7
tmpswe[[8]] <- tmp2008.out8
tmpswe[[9]] <- tmp2008.out9
tmpswe[[10]] <- tmp2008.out10
tmpswe[[11]] <- tmp2008.out11
tmpswe[[12]] <- tmp2008.out12
tmpswe[[13]] <- tmp2008.out13
tmpswe[[14]] <- tmp2008.out14
tmpswe[[15]] <- tmp2008.out15
tmpswe[[16]] <- tmp2008.out16
tmpswe[[17]] <- tmp2008.out17
prop.pred <- function(tmpswe, n = 30, day = 100, ncol = 6) {

  prop.mat <- matrix(NA, n, ncol)
  colnames(prop.mat) <- c("50%CL.LL", "50%CL.UL", "50%CL.LLUL",
                         "95%CL.LL", "95%CL.UL", "95%CL.LLUL")
  rownames(prop.mat) <- c("TGsweQ2", "TGsweQ4", "TGsweQ6", "BLPsweQ2",
                           "BLPsweQ4", "BLPsweQ6", "HRsweQ2", "HRsweQ4",
                           "HRsweQ6", "LBsweQ2", "LBsweQ4", "LBsweQ6",
                           "BLsweQ2", "BLsweQ4", "BLsweQ6", "DBPsweQ2",
                           "DBPsweQ4", "DBPsweQ6", "MCsweQ2", "MCsweQ4",
                           "MCsweQ6", "FARMsweQ2", "FARMsweQ4", "FARMsweQ6",
                           "TDsweQ2", "TDsweQ4", "TDsweQ6", "PSsweQ2",
                           "PSsweQ4", "PSsweQ6")

  for (i in 1:n) {
    n.burn <- round(tmpswe[i]$n.mcmc/4)
    T <- dim(tmpswe[i]$Y)[1]
    m <- dim(tmpswe[i]$Y)[2]

    CI.50 <- apply(tmpswe[i]$Y.pred[, n.burn:tmpswe[i]$n.mcmc], 1, quantile, c(0.25, 0.75))
    CI.95 <- apply(tmpswe[i]$Y.pred[, n.burn:tmpswe[i]$n.mcmc], 1, quantile, c(0.025, 0.975))

    nc <- c(1:ncol)
    prop.mat[i, nc[1]] <- 100*sum(tmpswe[i]$Y[T, (day+1):m] < CI.50[1, (day+1):m])/(m-day)
    prop.mat[i, nc[2]] <- 100*sum(tmpswe[i]$Y[T, (day+1):m] > CI.50[2, (day+1):m])/(m-day)
    prop.mat[i, nc[4]] <- 100*sum(tmpswe[i]$Y[T, (day+1):m] < CI.95[1, (day+1):m])/(m-day)
  }
}

prop.swe[[18]] <- tmp2008.out18
prop.swe[[19]] <- tmp2008.out19
prop.swe[[20]] <- tmp2008.out20
prop.swe[[21]] <- tmp2008.out21
prop.swe[[22]] <- tmp2008.out22
prop.swe[[23]] <- tmp2008.out23
prop.swe[[24]] <- tmp2008.out24
prop.swe[[25]] <- tmp2008.out25
prop.swe[[26]] <- tmp2008.out26
prop.swe[[27]] <- tmp2008.out27
prop.swe[[28]] <- tmp2008.out28
prop.swe[[29]] <- tmp2008.out29
prop.swe[[30]] <- tmp2008.out30
151
(prop.mat[i, nc[5]] <- 100*sum(tmpswe[[i]]$Y[T, (day+1):m] > CI.95[2, (day+1):m])/(m-day)

(prop.mat[i, nc[6]] <- 100*sum((tmpswe[[i]]$Y[T,(day+1):m] < CI.95[1, (day+1):m]), (tmpswe[[i]]$Y[T, (day+1):m] > CI.95[2, (day+1):m]))/(m-day)

)

list(prop.mat = prop.mat)

env <- prop.pred(tmpswe)

propmat <- cbind(env$prop.mat[, 3], env$prop.mat[, 6])

s <- 10

qprop <- matrix(NA, s, 6)

colnames(qprop) <- c("50%-CI.(q = 2)", "50%-CI.(q = 4)", "50%-CI.(q = 6)", "95%-CI.(q = 2)", "95%-CI.(q = 4)", "95%-CI.(q = 6)"

for (j in 1:s) {
qprop[j, 1] <- propmat[(3*j-2), 1]
qprop[j, 2] <- propmat[(3*j-1), 1]
qprop[j, 3] <- propmat[3*j, 1]
qprop[j, 4] <- propmat[(3*j-2), 2]
qprop[j, 5] <- propmat[(3*j-1), 2]
qprop[j, 6] <- propmat[3*j, 2]
}
tmp2009.out9 <- swe.svd.mcmc.2(t(HRswe[, -c(32, 33, 34, 35)]),
6, 1000, 100, )

tmp2009.out10 <- swe.svd.mcmc.2(t(LBswe[, -c(32, 33, 34, 35)]),
2, 1000, 100, )

tmp2009.out11 <- swe.svd.mcmc.2(t(LBswe[, -c(32, 33, 34, 35)]),
4, 1000, 100, )

tmp2009.out12 <- swe.svd.mcmc.2(t(LBswe[, -c(32, 33, 34, 35)]),
6, 1000, 100, )

tmp2009.out13 <- swe.svd.mcmc.2(t(BLswe[, -c(32, 33, 34, 35)]),
2, 1000, 100, )

tmp2009.out14 <- swe.svd.mcmc.2(t(BLswe[, -c(32, 33, 34, 35)]),
4, 1000, 100, )

tmp2009.out15 <- swe.svd.mcmc.2(t(BLswe[, -c(32, 33, 34, 35)]),
6, 1000, 100, )

tmp2009.out16 <- swe.svd.mcmc.2(t(DBPswe[, -c(32, 33, 34, 35)]),
2, 1000, 100, )

tmp2009.out17 <- swe.svd.mcmc.2(t(DBPswe[, -c(32, 33, 34, 35)]),
4, 1000, 100, )

tmp2009.out18 <- swe.svd.mcmc.2(t(DBPswe[, -c(32, 33, 34, 35)]),
6, 1000, 100, )

tmp2009.out19 <- swe.svd.mcmc.2(t(MCswe[, -c(32, 33, 34, 35)]),
2, 1000, 100, )

tmp2009.out20 <- swe.svd.mcmc.2(t(MCswe[, -c(32, 33, 34, 35)]),
4, 1000, 100, )

tmp2009.out21 <- swe.svd.mcmc.2(t(MCswe[, -c(32, 33, 34, 35)]),
6, 1000, 100, )

tmp2009.out22 <- swe.svd.mcmc.2(t(FARMswe[, -c(32, 33, 34, 35)]),
2, 1000, 100, )

tmp2009.out23 <- swe.svd.mcmc.2(t(FARMswe[, -c(32, 33, 34, 35)]),
4, 1000, 100, )

tmp2009.out24 <- swe.svd.mcmc.2(t(FARMswe[, -c(32, 33, 34, 35)]),
6, 1000, 100, )

tmp2009.out25 <- swe.svd.mcmc.2(t(TDswe[, -c(32, 33, 34, 35)]),
2, 1000, 100, )

tmp2009.out26 <- swe.svd.mcmc.2(t(TDswe[, -c(32, 33, 34, 35)]),
4, 1000, 100, )

tmp2009.out27 <- swe.svd.mcmc.2(t(TDswe[, -c(32, 33, 34, 35)]),
6, 1000, 100, )

tmp2009.out28 <- swe.svd.mcmc.2(t(PSswe[, -c(32, 33, 34, 35)]),
2, 1000, 100, )

tmp2009.out29 <- swe.svd.mcmc.2(t(PSswe[, -c(32, 33, 34, 35)]),
4, 1000, 100, )

tmp2009.out30 <- swe.svd.mcmc.2(t(PSswe[, -c(32, 33, 34, 35)]),
6, 1000, 100, )
n <- 30
tmpswe2 <- list(NA, n)
tmpswe2[[1]] <- tmp2009.out1
tmpswe2[[2]] <- tmp2009.out2
tmpswe2[[3]] <- tmp2009.out3
tmpswe2[[4]] <- tmp2009.out4
tmpswe2[[5]] <- tmp2009.out5
tmpswe2[[6]] <- tmp2009.out6
tmpswe2[[7]] <- tmp2009.out7
tmpswe2[[8]] <- tmp2009.out8
tmpswe2[[9]] <- tmp2009.out9
tmpswe2[[10]] <- tmp2009.out10
tmpswe2[[11]] <- tmp2009.out11
tmpswe2[[12]] <- tmp2009.out12
tmpswe2[[13]] <- tmp2009.out13
tmpswe2[[14]] <- tmp2009.out14
tmpswe2[[15]] <- tmp2009.out15
tmpswe2[[16]] <- tmp2009.out16
tmpswe2[[17]] <- tmp2009.out17
tmpswe2[[18]] <- tmp2009.out18
tmpswe2[[19]] <- tmp2009.out19
tmpswe2[[20]] <- tmp2009.out20
tmpswe2[[21]] <- tmp2009.out21
tmpswe2[[22]] <- tmp2009.out22
tmpswe2[[23]] <- tmp2009.out23
tmpswe2[[24]] <- tmp2009.out24
tmpswe2[[25]] <- tmp2009.out25
tmpswe2[[26]] <- tmp2009.out26
tmpswe2[[27]] <- tmp2009.out27
tmpswe2[[28]] <- tmp2009.out28
tmpswe2[[29]] <- tmp2009.out29
tmpswe2[[30]] <- tmp2009.out30

prop.pred <- function(tmpswe, n = 30, day = 100, ncol = 6){
  prop.mat <- matrix(NA, n, ncol)
  colnames(prop.mat) <- c("50%CL.LL", "50%CL.UL", "50%CL.LLUL",
                          "95%CL.LL", "95%CL.UL", "95%CL.LLUL")
  rownames(prop.mat) <- c("TGsweQ2", "TGsweQ4", "TGsweQ6", "BLP sweQ2",...."
"BLPsweQ4", "BLPsweQ6", "HRsweQ2", "HRsweQ4", "HRsweQ6", "LBsweQ2", "LBsweQ4", "LBsweQ6", "BLsweQ2", "BLsweQ4", "BLsweQ6", "DBPsweQ2", "DBPsweQ4", "DBPsweQ6", "MCsweQ2", "MCsweQ4", "MCsweQ6", "FARMsweQ2", "FARMsweQ4", "FARMsweQ6", "TDsweQ2", "TDsweQ4", "TDsweQ6", "PSsweQ2", "PSsweQ4", "PSsweQ6")

for (i in 1:n) {
  n.burn <- round(tmpswe[[i]]$n.mcmc/4)
  T <- dim(tmpswe[[i]]$Y)[1]
  m <- dim(tmpswe[[i]]$Y)[2]

  CI.50 <- apply(tmpswe[[i]]$Y.pred[, n.burn:tmpswe[[i]]$n.mcmc], 1,
                 quantile, c(0.25, 0.75))
  CI.95 <- apply(tmpswe[[i]]$Y.pred[, n.burn:tmpswe[[i]]$n.mcmc], 1,
                 quantile, c(0.025, 0.975))

  nc <- c(1:ncol)
  prop.mat[i, nc[1]] <- 100*sum(tmpswe[[i]]$Y[T, (day+1):m] < CI.50[1, (day+1):m])/(m-day)
  prop.mat[i, nc[2]] <- 100*sum(tmpswe[[i]]$Y[T, (day+1):m] > CI.50[2, (day+1):m])/(m-day)
  prop.mat[i, nc[3]] <- 100*sum((tmpswe[[i]]$Y[T, (day+1):m] < CI.50[1, (day+1):m]), (tmpswe[[i]]$Y[T, (day+1):m] > CI.50[2, (day+1):m]))/(m-day)
  prop.mat[i, nc[4]] <- 100*sum(tmpswe[[i]]$Y[T, (day+1):m] < CI.95[1, (day+1):m])/(m-day)
  prop.mat[i, nc[5]] <- 100*sum(tmpswe[[i]]$Y[T, (day+1):m] > CI.95[2, (day+1):m])/(m-day)
  prop.mat[i, nc[6]] <- 100*sum((tmpswe[[i]]$Y[T, (day+1):m] < CI.95[1, (day+1):m]), (tmpswe[[i]]$Y[T, (day+1):m] > CI.95[2, (day+1):m]))/(m-day)
}

list(prop.mat = prop.mat)

env <- prop.pred(tmpswe2)
propmat2 <- cbind(env$prop.mat[,3], env$prop.mat[,6])

s <- 10

qprop2 <- matrix(NA, s, 6)
# colnames(qprop2) <- c("50%-CI.(q = 2)", "50%-CI.(q = 4)", "50%-CI.(q = 6)", "95%-CI.(q = 2)", "95%-CI.(q = 4)", "95%-CI.(q = 6)")
colnames(qprop2) <- c("q = 2", "q = 4", "q = 6", "q = 2", "q = 4", "q = 6")
for (j in 1:s) {
    qprop2[j, 1] <- propmat2[(3*j-2), 1]
    qprop2[j, 2] <- propmat2[(3*j-1), 1]
    qprop2[j, 3] <- propmat2[3*j, 1]
    qprop2[j, 4] <- propmat2[(3*j-2), 2]
    qprop2[j, 5] <- propmat2[(3*j-1), 2]
    qprop2[j, 6] <- propmat2[3*j, 2]
}

D.7 Creating Figure 2.3

## Figure 2.3: Plot of the Proportion of SWE That Fell Outside the 50% and 95% Credible Interval for q = 2, 4, and 6

layout (matrix(1:4, 2, 2))

par(mar = c(0, 4, 2, 0))
boxplot(qprop[, 1:3], col = "lightgrey", main = "", axes = FALSE, lwd = 2.3, cex.axis = 2, ylim = c(0, 55))
axis(2, cex.axis = 2)
axis(1, at = c(1, 2, 3), labels = FALSE)
box()
text(x = 0.565, y = 54, labels = "(a)", cex = 1.8, font = 2)
rect(xleft = 0, ybottom = 51, xright = 0.74, ytop = 57.5)

par(mar = c(3, 4, 1, 0))
boxplot(qprop2[, 1:3], col = "lightgrey", main = "", lwd = 2.3, cex.axis = 2, ylim = c(0, 55))
text(x = 0.565, y = 53.5, labels = "(c)", cex = 1.8, font = 2)
rect(xleft = 0, ybottom = 50, xright = 0.74, ytop = 57.5)

par(mar = c(0, 1, 2, 2))
boxplot(qprop[, 4:6], col = "lightgrey", main = "", axes = FALSE, lwd = 2.3, cex.axis = 2, ylim = c(0, 55))
axis(2, cex.axis = 2, labels = FALSE)
axis(1, at = c(1, 2, 3), labels = FALSE)
box()
text(x = 0.565, y = 54, labels = "(b)", cex = 1.8, font = 2)
rect(xleft = 0, ybottom = 51, xright = 0.74, ytop = 57.5)

par(mar = c(3, 1, 1, 2))
boxplot(qprop2[, 4:6], col = "lightgrey", main = "", axes = FALSE, lwd = 2.3, cex.axis = 2, ylim = c(0, 55))
axis(2, cex.axis = 2, labels = FALSE)
axis(1, at = c(1, 2, 3), labels = c("q = 2", "q = 4", "q = 6"), cex.axis = 2)
box()
text(x = 0.565, y = 53.5, labels = "(d)", cex = 1.8, font = 2)
rect(xleft = 0, ybottom = 50, xright = 0.74, ytop = 57.5)
D.8 Creating Figure 2.4

## Figure 2.4: Plot Of The Orthogonal Vectors Containing The First Four Dominant Signals

q <- 4  ## Number of Important Signals Chosen For SWE Data Dimension Reduction

snowday <- c(1, 100, 200, 300, 365)
waterday <- c("Oct. 1 (Day 1)", "Jan. 8 (Day 100)", "Apr. 18 (Day 200)", "Jul. 27 (Day 300)", "Sep. 30 (Day 365)"
linetype <- c(1:4)
legendText <- c("Signal 1", "Signal 2", "Signal 3", "Signal 4")
lineColor <- c("gray0", "gray10", "gray25", "gray40")
plotchar <- c(17, 8, 15, 9)

xypoints1 <- cbind(c(3, 43, 83, 123, 163, 198, 243, 263),
                   Y.svd$v[, 1:q][c(3, 43, 83, 123, 163, 198, 243, 263),
                                  c(1, 3)])

xypoints2 <- cbind(c(10, 50, 80, 130, 177, 230, 243, 280),
                   Y.svd$v[, 1:q][c(10, 50, 80, 130, 177, 230, 243, 280),
                                  c(2, 4)])

matplot(Y.svd$v[, 1:q],
        type = "l",
        lty = linetype,
        axes = FALSE, col = lineColor,
        lwd = 5,
        ylab = NA,
        # ylab = expression(italic(upsilon)[ts]),
        xlab = "Day (in water-year)",
        cex.lab = 2.1)

axis(1, snowday, waterday, cex.axis = 1.7)
axis(2, cex.axis = 1.5)
mtext(side = 2, expression(italic(upsilon)[tg]), line = 1.85, cex = 3)
box(lwd = 2)

for (i in 1:2) {
  points(xypoints1[, 1], xypoints1[, i + 1], pch = plotchar[(2*i) - 1],
          cex = 2, lwd = 1.5)
}

for (i in 1:2) {
  points(xypoints2[, 1], xypoints2[, i + 1], pch = plotchar[2*i],
          cex = 2, lwd = 1.5)
}

legend("topright", legendText, lty = linetype, lwd = 2.2, cex = 2.0, pch = plotchar)
D.9 Creating Figure 2.5

## Figure 2.5: Plot Of Left Singular Vectors Containing The Inter-Annual Time Series For TONY GROVE SNOTEL Site

k <- 83  ## UTAH SNOTEL Site Number Selected
T <- 30  ## Number Of Years Consider For Historical SWE Data
q <- 4   ## Number Of Important Signals Selected

u.dat <- matrix(round(U[(T*(k - 1) + 1):(T*k), 1:q], 3))
year.num <- matrix(rep(1979:2008, q))
series.num <- matrix(c(rep(1, T), rep(2, T), rep(3, T), rep(4, T)))

series.f <- factor(series.num, levels = c(1:q),
                   labels = c("Latent Time Series 1", "Latent Time Series 2",
                            "Latent Time Series 3", "Latent Time Series 4"))

series.data <- data.frame(rev(u.dat), rev(year.num), rev(series.num))

my.strip2 <- function(which.given, which.panel, ...){
                      "Latent Time Series 2", "Latent Time Series 1")
    panel.rect(0, 0, 1, 1, col = "grey85", border = 1)
    panel.text(x = 0.5, 
                y = 0.5, 
                adj = c(0.5, 0.55),
                cex = 1.1,
                lwd = 2,
                font = 2,
                lab = strip.labels[which.panel[which.given]]
}

## Define X and Y Axis Range:
xlim <- c(1978, 2009)
ylim <- c(-0.08, 0.09)

## Define Annual Quarters for Plot Grid Line Markers:
d <- seq(from = 1978, to = 2009, by = 1)

## Create Multipanel Plot:
print(xyplot(u.dat~year.num|rev(series.f), data = series.data,
            scales = list(y = "free", rot = 0, cex = 1.5),
            xlim = xlim,
            ylim = ylim,
            strip = my.strip2,
            outer = TRUE,
            layout = c(1, 4, 1),
            ylab = list(expression(italic(phi[gt])), cex = 2.0),
            xlab = list("Year", cex = 1.6),
main = " ",
panel = function(x, y, ...){
  panel.grid(h = -1, v = 0)
    # plot default horizontal gridlines
  panel.abline(h = 0, lty = 2, col = "brown", lwd = 1.5)
    # custom horizontal gridline
  panel.abline(v = d, col = "grey90", lwd = 1.5)
    # custom vertical gridlines
  panel.xyplot(x, y, ..., type = "b", col = 1, pch = 16,
    lwd = 1)  # raw data
},))

D.10 Applying the SVD/MCMC Function to the 2008 Water-Year for the Ten Selected UTAH SNOTEL Sites

## 8. Applying The SVD/MCMC ("swe.svd.mcmc.R") Function To The 2008 Water-Year For The Ten Selected UTAH SNOTEL Sites Based on the 4 Selected Important Signals (q = 1, 2, 3, 4)

## 2008 Water-Year MCMC Forecasting After day 100 (January 8)
set.seed(12345)
#source("swe.svd.mcmc.R")
q <- 4  # Number Of Important Signals Selected
n.mcmc <- 5000  # Number of MCMC Simulations To Be Done
outdat <- 100  # The Day After Which Forecasting Is Done based on The Posterior Distribution
mm <- 100

tmp2008.out1 <- swe.svd.mcmc(t(TGswe[, -c(31, 32, 33, 34, 35)]),
  q, n.mcmc, mm, outdat)
tmp2008.out2 <- swe.svd.mcmc(t(BLPswe[, -c(31, 32, 33, 34, 35)]),
  q, n.mcmc, mm, outdat)
tmp2008.out3 <- swe.svd.mcmc(t(HRswe[, -c(31, 32, 33, 34, 35)]),
  q, n.mcmc, mm, outdat)
tmp2008.out4 <- swe.svd.mcmc(t(LBswe[, -c(31, 32, 33, 34, 35)]),
  q, n.mcmc, mm, outdat)
tmp2008.out5 <- swe.svd.mcmc(t(BLswe[, -c(31, 32, 33, 34, 35)]),
  q, n.mcmc, mm, outdat)
tmp2008.out6 <- swe.svd.mcmc(t(DBPswe[, -c(31, 32, 33, 34, 35)]),
  q, n.mcmc, mm, outdat)
tmp2008.out7 <- swe.svd.mcmc(t(MCswe[, -c(31, 32, 33, 34, 35)]),
  q, n.mcmc, mm, outdat)
tmp2008.out8 <- swe.svd.mcmc(t(FARMswe[, -c(31, 32, 33, 34, 35)]),
  q, n.mcmc, mm, outdat)
tmp2008.out9 <- swe.svd.mcmc(t(TDswe[, -c(31, 32, 33, 34, 35)]),
  q, n.mcmc, mm, outdat)
tmp2008.out10 <- swe.svd.mcmc(t(PSswe[, -c(31, 32, 33, 34, 35)]),
  q, n.mcmc, mm, outdat)
## 2008 Water-Year MCMC Forecasting After day 130 (February 7)
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 130
mm <- 130

tmp2008a.out1 <- swe.svd.mcmc(t(TGswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008a.out2 <- swe.svd.mcmc(t(BLPswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008a.out3 <- swe.svd.mcmc(t(HRswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008a.out4 <- swe.svd.mcmc(t(LBswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008a.out5 <- swe.svd.mcmc(t( BLswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008a.out6 <- swe.svd.mcmc(t(DBPswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008a.out7 <- swe.svd.mcmc(t(MCswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008a.out8 <- swe.svd.mcmc(t(FARMswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008a.out9 <- swe.svd.mcmc(t(TDswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008a.out10 <- swe.svd.mcmc(t(PSswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)

## 2008 Water-Year MCMC Forecasting After day 160 (March 9)
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 160
mm <- 160

tmp2008b.out1 <- swe.svd.mcmc(t(TGswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008b.out2 <- swe.svd.mcmc(t(BLPswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008b.out3 <- swe.svd.mcmc(t(HRswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008b.out4 <- swe.svd.mcmc(t(LBswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008b.out5 <- swe.svd.mcmc(t( BLswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008b.out6 <- swe.svd.mcmc(t(DBPswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008b.out7 <- swe.svd.mcmc(t(MCswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008b.out8 <- swe.svd.mcmc(t(FARMswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
q, n.mcmc, mm, outdat)
tmp2008b.out9 <- swe.svd.mcmc(t(TDswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008b.out10 <- swe.svd.mcmc(t(PSswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)

## 2008 Water-Year MCMC Forecasting After day 190 (April 8)
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 190
mm <- 190
tmp2008c.out1 <- swe.svd.mcmc(t(TGswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008c.out2 <- swe.svd.mcmc(t(BLPswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008c.out3 <- swe.svd.mcmc(t(HRswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008c.out4 <- swe.svd.mcmc(t(LBswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008c.out5 <- swe.svd.mcmc(t(BLswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008c.out6 <- swe.svd.mcmc(t(DBPswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008c.out7 <- swe.svd.mcmc(t(MCswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008c.out8 <- swe.svd.mcmc(t(FARMswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008c.out9 <- swe.svd.mcmc(t(TDswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)
tmp2008c.out10 <- swe.svd.mcmc(t(PSswe[, -c(31, 32, 33, 34, 35)]),
q, n.mcmc, mm, outdat)

D.11 Creating the Tables for the Proportion of Observed SWE After Selected
Initiation Dates in the 2008 Water-Year

## 9. Creating the Tables For the Proportion of Observed SWE After days
## 100 (January 8), 130 (February 7), 160 (March 9), and 190 (April 8)
## In the 2008 Water-Year That Lies Outside The 50% and 95% Credible
## Intervals Obtained For the Posterior Predictive Distribution

## NOTE: To proceed, we need to create a folder and label it as "Output"
## in the current working directory in R

n <- 10 ## Number Of Selected Sites
## Table for Day 100

tmpswe2008 <- list(NA, n)
tmpswe2008[[1]] <- tmp2008.out1
tmpswe2008[[2]] <- tmp2008.out2
tmpswe2008[[3]] <- tmp2008.out3
tmpswe2008[[4]] <- tmp2008.out4
tmpswe2008[[5]] <- tmp2008.out5
tmpswe2008[[6]] <- tmp2008.out6
tmpswe2008[[7]] <- tmp2008.out7
tmpswe2008[[8]] <- tmp2008.out8
tmpswe2008[[9]] <- tmp2008.out9
tmpswe2008[[10]] <- tmp2008.out10

Table2008 <- prop.pred(tmpswe2008)
Table2008 <- data.frame(Table2008)
write.csv(Table2008, file = "Output/Table2008.csv")

## Table for Day 130

## Table for Day 160
Table2008b <- prop.pred(tmpswe2008b)
Table2008b <- data.frame(Table2008b)
write.csv(Table2008b, file = "Output/Table2008b.csv")

## Table for Day 190

tmpswe2008c <- list(NA, n)
tmpswe2008c[[1]] <- tmp2008c.out1
tmpswe2008c[[2]] <- tmp2008c.out2
tmpswe2008c[[3]] <- tmp2008c.out3
tmpswe2008c[[4]] <- tmp2008c.out4
tmpswe2008c[[5]] <- tmp2008c.out5
tmpswe2008c[[6]] <- tmp2008c.out6
tmpswe2008c[[7]] <- tmp2008c.out7
tmpswe2008c[[8]] <- tmp2008c.out8
tmpswe2008c[[9]] <- tmp2008c.out9
tmpswe2008c[[10]] <- tmp2008c.out10

Table2008c <- prop.pred(tmpswe2008c)
Table2008c <- data.frame(Table2008c)
write.csv(Table2008c, file = "Output/Table2008c.csv")

D.12 Obtaining the Plots of SWE Measurements and Posterior Distributions
As of the Selected Initiation Dates in the 2008 Water-Year

## 10. Plots of The Ten SNOTEL Site SWE Measurements and Posterior
## Prediction As Of January 8 (Day 100), February 7 (Day 130),
## March 9 (Day 160), and April 8 (Day 190) In The 2008 Water-Year

## NOTE: To proceed, we need to create a additional folders and label it as
## "Output1", "Output2", "Output3", "Output4", "Output5", "Output6",
## "Output7", "Output8", "Output9", and "Output10" in the
## current working directory in R

## Plots for Forecasting After Day 100

source("plot.swe.post_jan_TGsite.R")

pdf(file = "Output/Output1/SWE2008TGswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2008.out1)
dev.off()

pdf(file = "Output/Output2/SWE2008BLPswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2008.out2)
dev.off()

pdf(file = "Output/Output3/SWE2008HRswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2008.out3)
dev.off()

source("plot.swe.post_jan_LBsite.R")
pdf(file = "Output/Output4/SWE2008LBswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_LBsite(tmp2008.out4)
dev.off()

source("plot.swe.post_jan_TGsite.R")
pdf(file = "Output/Output5/SWE2008BLswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2008.out5)
dev.off()

pdf(file = "Output/Output6/SWE2008DBPswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2008.out6)
dev.off()

pdf(file = "Output/Output7/SWE2008MCswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2008.out7)
dev.off()

pdf(file = "Output/Output8/SWE2008FARMswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2008.out8)
dev.off()

pdf(file = "Output/Output9/SWE2008TDswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2008.out9)
dev.off()

pdf(file = "Output/Output10/SWE2008PSswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2008.out10)
dev.off()

## Plots for Forecasting After Day 130

source("plot.swe.post_feb_TGsite.R")
pdf(file = "Output/Output1/SWE2008aTGswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2008a.out1)
dev.off()

pdf(file = "Output/Output2/SWE2008aBLPswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2008a.out2)
dev.off()

pdf(file = "Output/Output3/SWE2008aHRswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2008a.out3)
dev.off()

source("plot.swe.post_feb_LBswe.R")
pdf(file = "Output/Output4/SWE2008aLBswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_LBswe(tmp2008a.out4)
dev.off()

source("plot.swe.post_feb_TGswe.R")
pdf(file = "Output/Output5/SWE2008aBLswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGswe(tmp2008a.out5)
dev.off()

pdf(file = "Output/Output6/SWE2008aDBPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGswe(tmp2008a.out6)
dev.off()

pdf(file = "Output/Output7/SWE2008aMCswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGswe(tmp2008a.out7)
dev.off()

pdf(file = "Output/Output8/SWE2008aFARMswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGswe(tmp2008a.out8)
dev.off()

pdf(file = "Output/Output9/SWE2008aTDswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGswe(tmp2008a.out9)
dev.off()

pdf(file = "Output/Output10/SWE2008aPSswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGswe(tmp2008a.out10)
dev.off()

## Plots for Forecasting After Day 160

source("plot.swe.post_mar_TGswe.R")
pdf(file = "Output/Output1/SWE2008bTGswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGswe(tmp2008b.out1)
dev.off()

pdf(file = "Output/Output2/SWE2008bBLPswe.pdf", width = 12, height = 9,  
   pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2008b.out2)
dev.off()

pdf(file = "Output/Output3/SWE2008bHRswe.pdf", width = 12, height = 9,  
   pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2008b.out3)
dev.off()

source("plot.swe.post_mar_LBswe.R")

pdf(file = "Output/Output4/SWE2008bLBswe.pdf", width = 12, height = 9,  
   pointsize = 12, bg = "white")
plot.swe.post_mar_LBswe(tmp2008b.out4)
dev.off()

source("plot.swe.post_mar_TGsite.R")

pdf(file = "Output/Output5/SWE2008bBLswe.pdf", width = 12, height = 9,  
   pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2008b.out5)
dev.off()

pdf(file = "Output/Output6/SWE2008bDBPswe.pdf", width = 12, height = 9,  
   pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2008b.out6)
dev.off()

pdf(file = "Output/Output7/SWE2008bMCswe.pdf", width = 12, height = 9,  
   pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2008b.out7)
dev.off()

pdf(file = "Output/Output8/SWE2008bFARMswe.pdf", width = 12, height = 9,  
   pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2008b.out8)
dev.off()

pdf(file = "Output/Output9/SWE2008bTDswe.pdf", width = 12, height = 9,  
   pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2008b.out9)
dev.off()

pdf(file = "Output/Output10/SWE2008bPSswe.pdf", width = 12, height = 9,  
   pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2008b.out10)
dev.off()
## Plots for Forecasting After Day 190

source("plot.swe.post_apr_TGsite.R")

pdf(file = "Output/Output1/SWE2008cTGswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2008c.out1)
dev.off()

pdf(file = "Output/Output2/SWE2008cBLPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2008c.out2)
dev.off()

pdf(file = "Output/Output3/SWE2008cHRswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2008c.out3)
dev.off()

source("plot.swe.post_apr_LBsite.R")

pdf(file = "Output/Output4/SWE2008cLBswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_LBsite(tmp2008c.out4)
dev.off()

source("plot.swe.post_apr_TGsite.R")

pdf(file = "Output/Output5/SWE2008cBLswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2008c.out5)
dev.off()

pdf(file = "Output/Output6/SWE2008cDBPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2008c.out6)
dev.off()

pdf(file = "Output/Output7/SWE2008cMCswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2008c.out7)
dev.off()

pdf(file = "Output/Output8/SWE2008cFARMswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2008c.out8)
dev.off()

pdf(file = "Output/Output9/SWE2008cTDswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2008c.out9)
dev.off()
### D.13 Creating Figure 2.6

## Figure 2.6: XYPlot for the Estimates of the Model Parameter Alpha_st

```r
alphatmean1 <- c(tmp2008.out1$alpha.mean[,1], tmp2008.out2$alpha.mean[,1],
                   tmp2008.out3$alpha.mean[,1], tmp2008.out4$alpha.mean[,1],
                   tmp2008.out5$alpha.mean[,1], tmp2008.out6$alpha.mean[,1],
                   tmp2008.out7$alpha.mean[,1], tmp2008.out8$alpha.mean[,1],
                   tmp2008.out9$alpha.mean[,1], tmp2008.out10$alpha.mean[,1])
```

```r
alphatmean2 <- c(tmp2008.out1$alpha.mean[,2], tmp2008.out2$alpha.mean[,2],
                   tmp2008.out3$alpha.mean[,2], tmp2008.out4$alpha.mean[,2],
                   tmp2008.out5$alpha.mean[,2], tmp2008.out6$alpha.mean[,2],
                   tmp2008.out7$alpha.mean[,2], tmp2008.out8$alpha.mean[,2],
                   tmp2008.out9$alpha.mean[,2], tmp2008.out10$alpha.mean[,2])
```

```r
alphatmean3 <- c(tmp2008.out1$alpha.mean[,3], tmp2008.out2$alpha.mean[,3],
                   tmp2008.out3$alpha.mean[,3], tmp2008.out4$alpha.mean[,3],
                   tmp2008.out5$alpha.mean[,3], tmp2008.out6$alpha.mean[,3],
                   tmp2008.out7$alpha.mean[,3], tmp2008.out8$alpha.mean[,3],
                   tmp2008.out9$alpha.mean[,3], tmp2008.out10$alpha.mean[,3])
```

```r
alphatmean4 <- c(tmp2008.out1$alpha.mean[,4], tmp2008.out2$alpha.mean[,4],
                   tmp2008.out3$alpha.mean[,4], tmp2008.out4$alpha.mean[,4],
                   tmp2008.out5$alpha.mean[,4], tmp2008.out6$alpha.mean[,4],
                   tmp2008.out7$alpha.mean[,4], tmp2008.out8$alpha.mean[,4],
                   tmp2008.out9$alpha.mean[,4], tmp2008.out10$alpha.mean[,4])
```

```r
alphatmean <- round(cbind(alphatmean1, alphatmean2, alphatmean3,
                         alphatmean4), 3)
```

```r
q <- 4
T <- 30
n <- 10
year.num <- matrix(rep(1:(n*T), q))
series.num <- matrix(c(rep(1, n*T), rep(2, n*T), rep(3, n*T), rep(4, n*T)))
series.f <- factor(series.num, levels = c(1:q),
                   labels = c("Signal 1", "Signal 2", "Signal 3", "Signal 4"))
series.data <- data.frame(rev(alphatmean), rev(year.num), rev(series.num))
```
my.strip2 <- function(which.given, which.panel, ...){
  strip.labels <- c("Signal 4 (g = 4)", "Signal 3 (g = 3)",
                   "Signal 2 (g = 2)", "Signal 1 (g = 1)"
  )
  panel.rect(0, 0, 1, 1, col = "grey85", border = 1)
  panel.text(x = 0.5,
             y = 0.5,
             adj = c(0.5, 0.55),
             cex = 1.1,
             lwd = 2,
             font = 2,
             lab = strip.labels[which.panel[which.given]])
}

## Define X and Y Axis Range:
xlim <- c(1, n*T)
ylim <- c(-16, 15.5)

## Define Annual Quarters for Plot Grid Line Markers:
d <- seq(from = 1, to = n*T, by = 1)
d2 <- seq(from = 1, to = n*T, by = 30)

## Create Multipanel Plot:
x.at <- c(16, 46, 76, 106, 136, 166, 196, 226, 256, 286)
print(xyplot(alphatmean~year.num|rev(series.f), data = series.data,
             scales = list(y = "free",
                           x = list(cex=1.5 , at = x.at, labels =c("TG", "BLP", "HR",
                           "LB", "BL", "DBP", "MC", "FAR", "TD", "PS")),
                           rot = 0, cex = 1.5, tck = c(1, 0)),
             xlim = xlim,
ylim = ylim,
strip = my.strip2,
outer = TRUE,
layout = c(1, 4, 1),
ylab = list(expression(italic(alpha[st])), cex = 2.0),
xlab = list("", cex = 1.6),
main = "",
panel = function(x, y, ...) {panel.grid(h = -1, v = 0)
                           # plot default horizontal gridlines
                           panel.abline(h = 0, lty = 2, col = "brown", lwd = 3)
                           # custom horizontal gridline
                           # panel.abline(v = d, col = "grey90", lwd = 1.5)
                           # custom vertical gridlines
                           panel.abline(v = d2, col = "grey5", lwd = 1.5)
                           # custom vertical gridlines
                           panel.xyplot(x, y, ..., type = "l", col = 1, pch = 16,
                                         lwd = 1.5) # raw data
                           },))
D.14 Creating Figure 2.7

### Figure 2.7: Boxplots for the Estimates of the Model Parameter Alpha Based on the Four Important Signals

```r
nmcmc <- 2000
layout(matrix(1:2, 1, 2))
boxplot(list(tmp2008.out1$alpha.save[1,1:nmcmc],
         tmp2008.out1$alpha.save[2,1:nmcmc],
         tmp2008.out1$alpha.save[3,1:nmcmc],
         tmp2008.out1$alpha.save[4,1:nmcmc]),
       names = (expression(alpha[1], alpha[2], alpha[3], alpha[4])),
       main = "", col = "lightgrey", lwd = 2, cex.axis = 3, font = 2)
abline(h = 0, lty = 4, lwd = 3, col = 1)
text(x = 0.57, y = 2.3, labels = "(a)", cex = 2.5, font = 2)
rect(xleft = 0, ybottom = 2.05, xright = 0.8, ytop = 2.7)
boxplot(list(tmp2008.out4$alpha.save[1,1:nmcmc],
         tmp2008.out4$alpha.save[2,1:nmcmc],
         tmp2008.out4$alpha.save[3,1:nmcmc],
         tmp2008.out4$alpha.save[4,1:nmcmc]),
       names = (expression(alpha[1], alpha[2], alpha[3], alpha[4])),
       main = "", col = "lightgrey", lwd = 2, cex.axis = 3, font = 2)
abline(h = 0, lty = 4, lwd = 3, col = 1)
text(x = 0.57, y = 2.2, labels = "(b)", cex = 2.5, font = 2)
rect(xleft = 0, ybottom = 1.98, xright = 0.8, ytop = 2.5)
```

D.15 Creating Figure 2.8

### Figure 2.8: Plot of Posterior Prediction As of January 8, 2008 for the Tony Grove SNOTEL Site

```r
source("plot.swe.post_jan_Fig8_TGsite.R")
plot.swe.post_jan_Fig8_TGsite(tmp2008.out1)
```

D.16 Applying the SVD/MCMC Function to the 2009 Water-Year for the Ten Selected UTAH SNOTEL Sites

### 11. Applying The SVD/MCMC ("swe.svd.mcmc.R") Function To The 2009 Water-Year For The Ten Selected UTAH SNOTEL Sites Based on the 4 Selected Important Signals (q = 1, 2, 3, 4)

```r
set.seed(12345)
source("swe.svd.mcmc.R")
q <- 4  ## Number Of Important Signals Selected
n.mcmc <- 5000  ## Number of MCMC Simulations To Be Done
outdat <- 100  ## The Day After Which Forecasting Is Done based on The Posterior Distribution
```
```
mm <- 100

tmp2009.out1 <- swe.svd.mcmc(t(TGswe[, -c(32, 33, 34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2009.out2 <- swe.svd.mcmc(t(BLswe[, -c(32, 33, 34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2009.out3 <- swe.svd.mcmc(t(HRswe[, -c(32, 33, 34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2009.out4 <- swe.svd.mcmc(t(LBswe[, -c(32, 33, 34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2009.out5 <- swe.svd.mcmc(t(BLSwe[, -c(32, 33, 34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2009.out6 <- swe.svd.mcmc(t(DBPswe[, -c(32, 33, 34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2009.out7 <- swe.svd.mcmc(t(MCswe[, -c(32, 33, 34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2009.out8 <- swe.svd.mcmc(t(FARMswe[, -c(32, 33, 34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2009.out9 <- swe.svd.mcmc(t(TDswe[, -c(32, 33, 34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2009.out10 <- swe.svd.mcmc(t(PSswe[, -c(32, 33, 34, 35)]),
                          q, n.mcmc, mm, outdat)

## 2009 Water-Year MCMC Forecasting After day 130
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 130
mm <- 130

tmp2009a.out1 <- swe.svd.mcmc(t(TGswe[, -c(32, 33, 34, 35)]),
                        q, n.mcmc, mm, outdat)
tmp2009a.out2 <- swe.svd.mcmc(t(BLswe[, -c(32, 33, 34, 35)]),
                        q, n.mcmc, mm, outdat)
tmp2009a.out3 <- swe.svd.mcmc(t(HRswe[, -c(32, 33, 34, 35)]),
                        q, n.mcmc, mm, outdat)
tmp2009a.out4 <- swe.svd.mcmc(t(LBswe[, -c(32, 33, 34, 35)]),
                        q, n.mcmc, mm, outdat)
tmp2009a.out5 <- swe.svd.mcmc(t(BLSwe[, -c(32, 33, 34, 35)]),
                        q, n.mcmc, mm, outdat)
tmp2009a.out6 <- swe.svd.mcmc(t(DBPswe[, -c(32, 33, 34, 35)]),
                        q, n.mcmc, mm, outdat)
tmp2009a.out7 <- swe.svd.mcmc(t(MCswe[, -c(32, 33, 34, 35)]),
                        q, n.mcmc, mm, outdat)
tmp2009a.out8 <- swe.svd.mcmc(t(FARMswe[, -c(32, 33, 34, 35)]),
                        q, n.mcmc, mm, outdat)
tmp2009a.out9 <- swe.svd.mcmc(t(TDswe[, -c(32, 33, 34, 35)]),
                        q, n.mcmc, mm, outdat)
tmp2009a.out10 <- swe.svd.mcmc(t(PSswe[, -c(32, 33, 34, 35)]),
                        q, n.mcmc, mm, outdat)
```

## 2009 Water-Year MCMC Forecasting After day 160

```r
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 160
mm <- 160

tmp2009b.out1 <- swe.svd.mcmc(t(TGswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009b.out2 <- swe.svd.mcmc(t(BLPswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009b.out3 <- swe.svd.mcmc(t(HRswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009b.out4 <- swe.svd.mcmc(t(LBswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009b.out5 <- swe.svd.mcmc(t(BLswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009b.out6 <- swe.svd.mcmc(t(DBPswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009b.out7 <- swe.svd.mcmc(t(MCswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009b.out8 <- swe.svd.mcmc(t(FARMswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009b.out9 <- swe.svd.mcmc(t(TDswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009b.out10 <- swe.svd.mcmc(t(PSswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
```

## 2009 Water-Year MCMC Forecasting After day 190

```r
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 190
mm <- 190

tmp2009c.out1 <- swe.svd.mcmc(t(TGswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009c.out2 <- swe.svd.mcmc(t(BLPswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009c.out3 <- swe.svd.mcmc(t(HRswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009c.out4 <- swe.svd.mcmc(t(LBswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009c.out5 <- swe.svd.mcmc(t(BLswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009c.out6 <- swe.svd.mcmc(t(DBPswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009c.out7 <- swe.svd.mcmc(t(MCswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
tmp2009c.out8 <- swe.svd.mcmc(t(FARMswe[, -c(32, 33, 34, 35)]),
                           q, n.mcmc, mm, outdat)
```
D.17 Creating the Tables for the Proportion of Observed SWE After Selected Initiation Dates in the 2009 Water-Year

## 12. Creating The Tables For The Proportion Of Observed SWE After days 100 (January 8), 130 (February 7), 160 (March 9), and 190 (April 8) In The 2009 Water-Year That Lies Outside The 50% and 95% Credible Intervals Obtained For The Posterior Predictive Distribution

n <- 10 ## Number Of Selected Sites

## Table for Day 100

tmpswe2009 <- list(NA, n)
tmpswe2009[[1]] <- tmp2009.out1
tmpswe2009[[2]] <- tmp2009.out2
tmpswe2009[[3]] <- tmp2009.out3
tmpswe2009[[4]] <- tmp2009.out4
tmpswe2009[[5]] <- tmp2009.out5
tmpswe2009[[6]] <- tmp2009.out6
tmpswe2009[[7]] <- tmp2009.out7
tmpswe2009[[8]] <- tmp2009.out8
tmpswe2009[[9]] <- tmp2009.out9
tmpswe2009[[10]] <- tmp2009.out10

Table2009 <- prop.pred(tmpswe2009)
Table2009 <- data.frame(Table2009)
write.csv(Table2009, file = "Output/Table2009.csv")

## Table for Day 130

tmpswe2009a <- list(NA, n)
tmpswe2009a[[1]] <- tmp2009a.out1
tmpswe2009a[[2]] <- tmp2009a.out2
tmpswe2009a[[3]] <- tmp2009a.out3
tmpswe2009a[[4]] <- tmp2009a.out4
tmpswe2009a[[5]] <- tmp2009a.out5
tmpswe2009a[[6]] <- tmp2009a.out6
tmpswe2009a[[7]] <- tmp2009a.out7
tmpswe2009a[[8]] <- tmp2009a.out8
tmpswe2009a[[9]] <- tmp2009a.out9
tmpswe2009a[[10]] <- tmp2009a.out10

Table2009a <- prop.pred(tmpswe2009a)
Table2009a <- data.frame(Table2009a)
write.csv(Table2009a, file = "Output/Table2009a.csv")
### Table for Day 160

tmpswe2009b <- list(NA, n)
tmpswe2009b[[1]] <- tmp2009b.out1
tmpswe2009b[[2]] <- tmp2009b.out2
tmpswe2009b[[3]] <- tmp2009b.out3
tmpswe2009b[[4]] <- tmp2009b.out4
tmpswe2009b[[5]] <- tmp2009b.out5
tmpswe2009b[[6]] <- tmp2009b.out6
tmpswe2009b[[7]] <- tmp2009b.out7
tmpswe2009b[[8]] <- tmp2009b.out8
tmpswe2009b[[9]] <- tmp2009b.out9
tmpswe2009b[[10]] <- tmp2009b.out10

Table2009b <- prop.pred(tmpswe2009b)
Table2009b <- data.frame(Table2009b)
write.csv(Table2009b, file = "Output/Table2009b.csv")

### Table for Day 190

tmpswe2009c <- list(NA, n)
tmpswe2009c[[1]] <- tmp2009c.out1
tmpswe2009c[[2]] <- tmp2009c.out2
tmpswe2009c[[3]] <- tmp2009c.out3
tmpswe2009c[[4]] <- tmp2009c.out4
tmpswe2009c[[5]] <- tmp2009c.out5
tmpswe2009c[[6]] <- tmp2009c.out6
tmpswe2009c[[7]] <- tmp2009c.out7
tmpswe2009c[[8]] <- tmp2009c.out8
tmpswe2009c[[9]] <- tmp2009c.out9
tmpswe2009c[[10]] <- tmp2009c.out10

Table2009c <- prop.pred(tmpswe2009c)
Table2009c <- data.frame(Table2009c)
write.csv(Table2009c, file = "Output/Table2009c.csv")

D.18 Obtaining the Plots of SWE Measurements and Posterior Distributions As of the Selected Initiation Dates in the 2009 Water-Year

## 13. Plots of The Ten SNOTEL Site SWE Measurements and Posterior Prediction As Of January 8 (Day 100), February 7 (Day 130), March 9 (Day 160), and April 8 (Day 190) In The 2009 Water-Year

## Plots for Forecasting After Day 100

source("plot.swe.post_jan_TGsite.R")
pdf(file = "Output/Output1/SWE2009TGswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2009.out1)
dev.off()

pdf(file = "Output/Output2/SWE2009BLPswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2009.out2)
dev.off()

df(file = "Output/Output3/SWE2009HRswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2009.out3)
dev.off()

source("plot.swe.post_jan_LBswe.R")
pdf(file = "Output_paper/Output4/SWE2009LBswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_LBswe(tmp2009.out4)
dev.off()

source("plot.swe.post_jan_TGswe.R")
pdf(file = "Output/Output5/SWE2009BLswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGswe(tmp2009.out5)
dev.off()

pdf(file = "Output/Output6/SWE2009DBPswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGswe(tmp2009.out6)
dev.off()

pdf(file = "Output/Output7/SWE2009MCswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGswe(tmp2009.out7)
dev.off()

pdf(file = "Output/Output8/SWE2009FARMswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGswe(tmp2009.out8)
dev.off()

pdf(file = "Output/Output9/SWE2009TDSwe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGswe(tmp2009.out9)
dev.off()

pdf(file = "Output/Output10/SWE2009PSswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGswe(tmp2009.out10)
dev.off()
## Plots for Forecasting After Day 130

source("plot.swe.post_feb_TGsite.R")

pdf(file = "Output/Output1/SWE2009aTGswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2009a.out1)
dev.off()

pdf(file = "Output/Output2/SWE2009aBLPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2009a.out2)
dev.off()

pdf(file = "Output/Output3/SWE2009aHRswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2009a.out3)
dev.off()

source("plot.swe.post_feb_LBsite.R")

pdf(file = "Output/Output4/SWE2009aLBswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_LBsite(tmp2009a.out4)
dev.off()

source("plot.swe.post_feb_TGsite.R")

pdf(file = "Output/Output5/SWE2009aBLswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2009a.out5)
dev.off()

pdf(file = "Output/Output6/SWE2009aDBPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2009a.out6)
dev.off()

pdf(file = "Output/Output7/SWE2009aMCswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2009a.out7)
dev.off()

pdf(file = "Output/Output8/SWE2009aFARMswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2009a.out8)
dev.off()

pdf(file = "Output/Output9/SWE2009aTDswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2009a.out9)
dev.off()
## Plots for Forecasting After Day 160

source("plot.swe.post_mar_TGsite.R")

pdf(file = "Output/Output1/SWE2009bTGswe.pdf", width = 12, height = 9, 
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2009b.out1)
dev.off()

pdf(file = "Output/Output2/SWE2009bBLPswe.pdf", width = 12, height = 9, 
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2009b.out2)
dev.off()

pdf(file = "Output/Output3/SWE2009bHRswe.pdf", width = 12, height = 9, 
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2009b.out3)
dev.off()

source("plot.swe.post_mar_LBsite.R")

pdf(file = "Output/Output4/SWE2009bLBswe.pdf", width = 12, height = 9, 
     pointsize = 12, bg = "white")
plot.swe.post_mar_LBsite(tmp2009b.out4)
dev.off()

source("plot.swe.post_mar_TGsite.R")

pdf(file = "Output/Output5/SWE2009bBLswe.pdf", width = 12, height = 9, 
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2009b.out5)
dev.off()

pdf(file = "Output/Output6/SWE2009bDBPswe.pdf", width = 12, height = 9, 
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2009b.out6)
dev.off()

pdf(file = "Output/Output7/SWE2009bMCswe.pdf", width = 12, height = 9, 
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2009b.out7)
dev.off()

pdf(file = "Output/Output8/SWE2009bFARMswe.pdf", width = 12, height = 9, 
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2009b.out8)
dev.off()
```r
pdf(file = "Output/Output9/SWE2009bTDswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2009b.out9)
dev.off()

df(file = "Output/Output10/SWE2009bPSswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2009b.out10)
dev.off()

## Plots for Forecasting After Day 190

source("plot.swe.post_apr_TGsite.R")
pdf(file = "Output/Output1/SWE2009cTGswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2009c.out1)
dev.off()

pdf(file = "Output/Output2/SWE2009cBLPswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2009c.out2)
dev.off()

pdf(file = "Output/Output3/SWE2009cHRswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2009c.out3)
dev.off()

source("plot.swe.post_apr_LBswe.R")
pdf(file = "Output_paper/Output4/SWE2009cLBswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_apr_LBswe(tmp2009c.out4)
dev.off()

source("plot.swe.post_apr_TGsite.R")
pdf(file = "Output/Output5/SWE2009cBLswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2009c.out5)
dev.off()

pdf(file = "Output/Output6/SWE2009cDBPswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2009c.out6)
dev.off()

pdf(file = "Output/Output7/SWE2009cMCswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2009c.out7)
dev.off()
```
pdf(file = "Output_paper/Output8/SWE2009cFARMswe.pdf", width = 12, 
    height = 9, pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2009c.out8)
dev.off()

dpdf(file = "Output/Output9/SWE2009cTDswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2009c.out9)
dev.off()

dpdf(file = "Output/Output10/SWE2009cPSswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2009c.out10)
dev.off()

D.19 Applying the SVD/MCMC Function to the 2010 Water-Year for the Ten 
Selected UTAH SNOTEL Sites

## 14. Applying The SVD/MCMC ("swe.svd.mcmc.R") Function To The 2010 
## Water-Year For The Ten Selected UTAH SNOTEL Sites Based On The 4 
## Selected Important Signals (q = 1, 2, 3, 4) 

## 2010 Water-Year MCMC Forecasting After day 100 
set.seed(12345)
#source("swe.svd.mcmc.R")
q <- 4  ## Number Of Important Signals Selected 
n.mcmc <- 5000  ## Number of MCMC Simulations To Be done 
outdat <- 100  ## The Day After Which Forecasting Is Done Based On The 
## Posterior Distribution

mm <-100

tmp2010.out1 <- swe.svd.mcmc(t(TGswe[, -c(33, 34, 35)]), 
    q, n.mcmc, mm, outdat)
tmp2010.out2 <- swe.svd.mcmc(t(BLPswe[, -c(33, 34, 35)]), 
    q, n.mcmc, mm, outdat)
tmp2010.out3 <- swe.svd.mcmc(t(HRswe[, -c(33, 34, 35)]), 
    q, n.mcmc, mm, outdat)
tmp2010.out4 <- swe.svd.mcmc(t(LBswe[, -c(33, 34, 35)]), 
    q, n.mcmc, mm, outdat)
tmp2010.out5 <- swe.svd.mcmc(t(BLswe[, -c(33, 34, 35)]), 
    q, n.mcmc, mm, outdat)
tmp2010.out6 <- swe.svd.mcmc(t(DBPswe[, -c(33, 34, 35)]), 
    q, n.mcmc, mm, outdat)
tmp2010.out7 <- swe.svd.mcmc(t(MCswe[, -c(33, 34, 35)]), 
    q, n.mcmc, mm, outdat)
tmp2010.out8 <- swe.svd.mcmc(t(FARMswe[, -c(33, 34, 35)]), 
    q, n.mcmc, mm, outdat)
tmp2010.out9 <- swe.svd.mcmc(t(TDswe[, -c(33, 34, 35)]), 
    q, n.mcmc, mm, outdat)
tmp2010.out10 <- swe.svd.mcmc(t(PSswe[, -c(33, 34, 35)]),
## 2010 Water-Year MCMC Forecasting After day 130

```r
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 130
mm <- 130

tmp2010a.out1 <- swe.svd.mcmc(t(TGswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010a.out2 <- swe.svd.mcmc(t(BLPswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010a.out3 <- swe.svd.mcmc(t(HRswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010a.out4 <- swe.svd.mcmc(t(LBswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010a.out5 <- swe.svd.mcmc(t(BLSwe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010a.out6 <- swe.svd.mcmc(t(DBPswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010a.out7 <- swe.svd.mcmc(t(MCswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010a.out8 <- swe.svd.mcmc(t(FARMswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010a.out9 <- swe.svd.mcmc(t(TDswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010a.out10 <- swe.svd.mcmc(t(PSswe[, -c(33, 34, 35)]),
                            q, n.mcmc, mm, outdat)
```

## 2010 Water-Year MCMC Forecasting After day 160

```r
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 160
mm <- 160

tmp2010b.out1 <- swe.svd.mcmc(t(TGswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010b.out2 <- swe.svd.mcmc(t(BLPswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010b.out3 <- swe.svd.mcmc(t(HRswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010b.out4 <- swe.svd.mcmc(t(LBswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010b.out5 <- swe.svd.mcmc(t(BLSwe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010b.out6 <- swe.svd.mcmc(t(DBPswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010b.out7 <- swe.svd.mcmc(t(MCswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010b.out8 <- swe.svd.mcmc(t(FARMswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010b.out9 <- swe.svd.mcmc(t(TDswe[, -c(33, 34, 35)]),
                             q, n.mcmc, mm, outdat)
tmp2010b.out10 <- swe.svd.mcmc(t(PSswe[, -c(33, 34, 35)]),
                            q, n.mcmc, mm, outdat)
```
D.20 Creating the Tables for the Proportion of Observed SWE After Selected Initiation Dates in the 2010 Water-Year

## 15. Creating The Tables For The Proportion Of Observed SWE After days 100 (January 8), 130 (February 7), 160 (March 9), and 190 (April 8) In The 2010 Water-Year That Lies Outside The 50% and 95% Credible Intervals Obtained For The Posterior Predictive Distribution

n <- 10 ## Number Of Selected Sites

## Table for Day 100

tmpswe2010 <- list(NA, n)
tmpswe2010[[1]] <- tmp2010b.out8
tmpswe2010[[2]] <- tmp2010.out2
tmpswe2010[[3]] <- tmp2010.out3
tmpswe2010[[4]] <- tmp2010.out4
tmpswe2010[[5]] <- tmp2010.out5
tmpswe2010[[6]] <- tmp2010.out6
tmpswe2010[[7]] <- tmp2010.out7
tmpswe2010[[8]] <- tmp2010.out8
tmpswe2010[[9]] <- tmp2010.out9
tmpswe2010[[10]] <- tmp2010.out10

Table2010 <- prop.pred(tmpswe2010)
Table2010 <- data.frame(Table2010)
write.csv(Table2010, file = "Output/Table2010.csv")

## Table for Day 130

tmpswe2010a <- list(NA, n)
tmpswe2010a[[1]] <- tmp2010a.out1
tmpswe2010a[[2]] <- tmp2010a.out2
tmpswe2010a[[3]] <- tmp2010a.out3
tmpswe2010a[[4]] <- tmp2010a.out4
tmpswe2010a[[5]] <- tmp2010a.out5
tmpswe2010a[[6]] <- tmp2010a.out6
tmpswe2010a[[7]] <- tmp2010a.out7
tmpswe2010a[[8]] <- tmp2010a.out8
tmpswe2010a[[9]] <- tmp2010a.out9
tmpswe2010a[[10]] <- tmp2010a.out10

Table2010a <- prop.pred(tmpswe2010a)
Table2010a <- data.frame(Table2010a)
write.csv(Table2010a, file = "Output/Table2010a.csv")

## Table for Day 160

tmpswe2010b <- list(NA, n)
tmpswe2010b[[1]] <- tmp2010b.out1
tmpswe2010b[[2]] <- tmp2010b.out2
tmpswe2010b[[3]] <- tmp2010b.out3
tmpswe2010b[[4]] <- tmp2010b.out4
tmpswe2010b[[5]] <- tmp2010b.out5
tmpswe2010b[[6]] <- tmp2010b.out6
tmpswe2010b[[7]] <- tmp2010b.out7
tmpswe2010b[[8]] <- tmp2010b.out8
tmpswe2010b[[9]] <- tmp2010b.out9
tmpswe2010b[[10]] <- tmp2010b.out10

Table2010b <- prop.pred(tmpswe2010b)
Table2010b <- data.frame(Table2010b)
write.csv(Table2010b, file = "Output/Table2010b.csv")
## Table for Day 190

tmpswe2010c <- list(NA, n)
tmpswe2010c[[1]] <- tmp2010c.out1
tmpswe2010c[[2]] <- tmp2010c.out2
tmpswe2010c[[3]] <- tmp2010c.out3
tmpswe2010c[[4]] <- tmp2010c.out4
tmpswe2010c[[5]] <- tmp2010c.out5
tmpswe2010c[[6]] <- tmp2010c.out6
tmpswe2010c[[7]] <- tmp2010c.out7
tmpswe2010c[[8]] <- tmp2010c.out8
tmpswe2010c[[9]] <- tmp2010c.out9
tmpswe2010c[[10]] <- tmp2010c.out10

Table2010c <- prop.pred(tmpswe2010c)
Table2010c <- data.frame(Table2010c)
write.csv(Table2010c, file = "Output/Table2010c.csv")

D.21 Obtaining the Plots of SWE Measurements and Posterior Distributions
As of the Selected Initiation Dates in the 2010 Water-Year

## 16. Plots of The Ten SNOTEL Site SWE Measurements and Posterior
## Prediction As Of January 8 (Day 100), February 7 (Day 130),
## March 9 (Day 160), and April 8 (Day 190) In The 2010 Water-Year

## Plots for Forecasting After Day 100

source("plot.swe.post_jan_TGsite.R")
pdf(file = "Output/Output1/SWE2010TGswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2010.out1)
dev.off()

dpdf(file = "Output/Output2/SWE2010BLPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2010.out2)
dev.off()

dpdf(file = "Output/Output3/SWE2010HRswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2010.out3)
dev.off()

source("plot.swe.post_jan_LBsite.R")
pdf(file = "Output/Output4/SWE2010LBswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_jan_LBsite(tmp2010.out4)
dev.off()
source("plot.swe.post_jan_TGsite.R")
pdf(file = "Output/Output5/SWE2010BLswe.pdf", width = 12, height = 9,
pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2010.out5)
dev.off()

pdf(file = "Output/Output6/SWE2010DBPswe.pdf", width = 12, height = 9,
pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2010.out6)
dev.off()

pdf(file = "Output/Output7/SWE2010MCswe.pdf", width = 12, height = 9,
pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2010.out7)
dev.off()

pdf(file = "Output/Output8/SWE2010FARMswe.pdf", width = 12, height = 9,
pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2010.out8)
dev.off()

pdf(file = "Output/Output9/SWE2010TDswe.pdf", width = 12, height = 9,
pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2010.out9)
dev.off()

pdf(file = "Output/Output10/SWE2010PSswe.pdf", width = 12, height = 9,
pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2010.out10)
dev.off()

## Plots for Forecasting After Day 130

source("plot.swe.post_feb_TGsite.R")
pdf(file = "Output/Output1/SWE2010aTGswe.pdf", width = 12, height = 9,
pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2010a.out1)
dev.off()

pdf(file = "Output/Output2/SWE2010aBLPswe.pdf", width = 12, height = 9,
pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2010a.out2)
dev.off()

pdf(file = "Output/Output3/SWE2010aHRswe.pdf", width = 12, height = 9,
pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2010a.out3)
dev.off()

source("plot.swe.post_feb_LBsite.R")
pdf(file = "Output/Output4/SWE2010aLBswe.pdf", width = 12, height = 9,
plot.swe.post_feb_LBsite(tmp2010a.out4)
dev.off()

source("plot.swe.post_feb_TGsite.R")
pdf(file = "Output/Output5/SWE2010aBLswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2010a.out5)
dev.off()

pdf(file = "Output/Output6/SWE2010aDBPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2010a.out6)
dev.off()

pdf(file = "Output/Output7/SWE2010aMCswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2010a.out7)
dev.off()

pdf(file = "Output/Output8/SWE2010aFARMswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2010a.out8)
dev.off()

pdf(file = "Output/Output9/SWE2010aTDswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2010a.out9)
dev.off()

## Plots for Forecasting After Day 160

source("plot.swe.post_mar_TGsite.R")
pdf(file = "Output/Output1/SWE2010bTGswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2010b.out1)
dev.off()

pdf(file = "Output/Output2/SWE2010bBLPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2010b.out2)
dev.off()

pdf(file = "Output/Output3/SWE2010bHRswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2010b.out3)
dev.off()

source("plot.swe.post_mar_LBsite.R")
pdf(file = "Output/Output4/SWE2010bLBswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_LBsite(tmp2010b.out4)
dev.off()

source("plot.swe.post_mar_TGsite.R")
pdf(file = "Output/Output5/SWE2010bBLswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2010b.out5)
dev.off()

dpdf(file = "Output/Output6/SWE2010bDBPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2010b.out6)
dev.off()

dpdf(file = "Output/Output7/SWE2010bMCswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2010b.out7)
dev.off()

dpdf(file = "Output/Output8/SWE2010bFARMswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2010b.out8)
dev.off()

dpdf(file = "Output/Output9/SWE2010bTDswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2010b.out9)
dev.off()

dpdf(file = "Output/Output10/SWE2010bPSswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2010b.out10)
dev.off()

## Plots for Forecasting After Day 190

source("plot.swe.post_apr_TGsite.R")
pdf(file = "Output/Output1/SWE2010cTGswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2010c.out1)
dev.off()

pdf(file = "Output/Output2/SWE2010cBLPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2010c.out2)
dev.off()
D.22 Applying the SVD/MCMC Function to the 2011 Water-Year for the Ten Selected UTAH SNOTEL Sites

## 17. Applying The SVD/MCMC ("swe.svd.mcmc.R") Function To The 2011 Water-Year For The Ten Selected UTAH SNOTEL Sites Based on the 4 Selected Important Signals (q = 1, 2, 3, 4)
## 2011 Water-Year MCMC Forecasting After day 100
set.seed(12345)
source("swe.svd.mcmc.R")
q <- 4    # Number Of Important Signals Selected
n.mcmc <- 5000    # Number of MCMC Simulations To Be Done
outdat <- 100    # The Day After Which Forecasting Is Done based on The
                # Posterior Distribution
mm <- 100

tmp2011.out1 <- swe.svd.mcmc(t(TGswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011.out2 <- swe.svd.mcmc(t(BLPswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011.out3 <- swe.svd.mcmc(t(HRswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011.out4 <- swe.svd.mcmc(t(LBswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011.out5 <- swe.svd.mcmc(t(BLswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011.out6 <- swe.svd.mcmc(t(DBPswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011.out7 <- swe.svd.mcmc(t(MCswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011.out8 <- swe.svd.mcmc(t(FARMswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011.out9 <- swe.svd.mcmc(t(TDswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011.out10 <- swe.svd.mcmc(t(PSswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)

## 2011 Water-Year MCMC Forecasting After day 130
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 130
mm <- 130

tmp2011a.out1 <- swe.svd.mcmc(t(TGswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011a.out2 <- swe.svd.mcmc(t(BLPswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011a.out3 <- swe.svd.mcmc(t(HRswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011a.out4 <- swe.svd.mcmc(t(LBswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011a.out5 <- swe.svd.mcmc(t(BLswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011a.out6 <- swe.svd.mcmc(t(DBPswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
tmp2011a.out7 <- swe.svd.mcmc(t(MCswe[, -c(34, 35)]),
                          q, n.mcmc, mm, outdat)
## 2011 Water-Year MCMC Forecasting After day 160

```r
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 160
mm <- 160

tmp2011b.out1 <- swe.svd.mcmc(t(TGswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011b.out2 <- swe.svd.mcmc(t(BLPswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011b.out3 <- swe.svd.mcmc(t(HRswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011b.out4 <- swe.svd.mcmc(t(LBswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011b.out5 <- swe.svd.mcmc(t(BLswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011b.out6 <- swe.svd.mcmc(t(DBPswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011b.out7 <- swe.svd.mcmc(t(MCswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011b.out8 <- swe.svd.mcmc(t(FARMswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011b.out9 <- swe.svd.mcmc(t(TDswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011b.out10 <- swe.svd.mcmc(t(PSswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
```

## 2011 Water-Year MCMC Forecasting After day 190

```r
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 190
mm <- 190

tmp2011c.out1 <- swe.svd.mcmc(t(TGswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011c.out2 <- swe.svd.mcmc(t(BLPswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011c.out3 <- swe.svd.mcmc(t(HRswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)
tmp2011c.out4 <- swe.svd.mcmc(t(LBswe[, -c(34, 35)]),
                               q, n.mcmc, mm, outdat)```
D.23 Creating the Tables for the Proportion of Observed SWE After Selected Initiation Dates in the 2011 Water-Year

## 18. Creating The Tables For The Proportion Of Observed SWE After days 100 (January 8), 130 (February 7), 160 (March 9), and 190 (April 8) In The 2011 Water-Year That Lies Outside The 50% and 95% Credible Intervals Obtained For The Posterior Predictive Distribution

n <- 10  ## Number Of Selected Sites

## Table for Day 100

tmpswe2011 <- list(NA, n)
tmpswe2011[[1]] <- tmp2011c.out1
tmpswe2011[[2]] <- tmp2011c.out2
tmpswe2011[[3]] <- tmp2011c.out3
tmpswe2011[[4]] <- tmp2011c.out4
tmpswe2011[[5]] <- tmp2011c.out5
tmpswe2011[[6]] <- tmp2011c.out6
tmpswe2011[[7]] <- tmp2011c.out7
tmpswe2011[[8]] <- tmp2011c.out8
tmpswe2011[[9]] <- tmp2011c.out9
tmpswe2011[[10]] <- tmp2011c.out10

Table2011 <- prop.pred(tmpswe2011)
Table2011 <- data.frame(Table2011)
write.csv(Table2011, file = "Output/Table2011.csv")

## Table for Day 130

tmpswe2011a <- list(NA, n)
tmpswe2011a[[1]] <- tmp2011a.out1
tmpswe2011a[[2]] <- tmp2011a.out2
tmpswe2011a[[3]] <- tmp2011a.out3
tmpswe2011a[[4]] <- tmp2011a.out4
tmpswe2011a[[5]] <- tmp2011a.out5
tmpswe2011a[[6]] <- tmp2011a.out6
D.24 Obtaining the Plots of SWE Measurements and Posterior Distributions
As of the Selected Initiation Dates in the 2011 Water-Year

## 19. Plots of The Ten SNOTEL Site SWE Measurements and Posterior Prediction As Of January 8 (Day 100), February 7 (Day 130),
March 9 (Day 160), and April 8 (Day 190) In The 2011 Water-Year

Plots for Forecasting After Day 100

```r
source("plot.swe.post_jan_TGsite.R")
pdf(file = "Output/Output1/SWE2011TGswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2011.out1)
dev.off()

df(file = "Output/Output2/SWE2011BLPswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2011.out2)
dev.off()

df(file = "Output/Output3/SWE2011HRswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2011.out3)
dev.off()

source("plot.swe.post_jan_LBsite.R")
pdf(file = "Output/Output4/SWE2011LBswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_LBsite(tmp2011.out4)
dev.off()

source("plot.swe.post_jan_TGsite.R")
pdf(file = "Output/Output5/SWE2011BLswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2011.out5)
dev.off()

pdf(file = "Output/Output6/SWE2011DBPswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2011.out6)
dev.off()

pdf(file = "Output/Output7/SWE2011MCswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2011.out7)
dev.off()

pdf(file = "Output/Output8/SWE2011FARMswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2011.out8)
dev.off()

pdf(file = "Output/Output9/SWE2011TDswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2011.out9)
dev.off()
```
## Plots for Forecasting After Day 130

source("plot.swe.post_feb_TGsite.R")

pdf(file = "Output/Output1/SWE2011aTGswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2011a.out1)
dev.off()

pdf(file = "Output/Output2/SWE2011aBLPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2011a.out2)
dev.off()

pdf(file = "Output/Output3/SWE2011aHRswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2011a.out3)
dev.off()

source("plot.swe.post_feb_LBsite.R")

pdf(file = "Output/Output4/SWE2011aBLswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_LBsite(tmp2011a.out4)
dev.off()

source("plot.swe.post_feb_TGsite.R")

pdf(file = "Output/Output5/SWE2011aDBPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2011a.out5)
dev.off()

pdf(file = "Output/Output6/SWE2011aDBPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2011a.out6)
dev.off()

pdf(file = "Output/Output7/SWE2011aMCswwe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2011a.out7)
dev.off()

pdf(file = "Output/Output8/SWE2011aFARMswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2011a.out8)
dev.off()
pdf(file = "Output/Output9/SWE2011aTDswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2011a.out9)
dev.off()

pdf(file = "Output/Output10/SWE2011aPSswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2011a.out10)
dev.off()

## Plots for Forecasting After Day 160

source("plot.swe.post_mar_TGsite.R")
pdf(file = "Output/Output1/SWE2011bTGswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2011b.out1)
dev.off()

pdf(file = "Output/Output2/SWE2011bBLPswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2011b.out2)
dev.off()

pdf(file = "Output/Output3/SWE2011bHRswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2011b.out3)
dev.off()

source("plot.swe.post_mar_LBswe.pdf")
pdf(file = "Output/Output4/SWE2011bBLswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_LBswe(tmp2011b.out4)
dev.off()

source("plot.swe.post_mar_TGsite.R")
pdf(file = "Output/Output5/SWE2011bDBPswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2011b.out5)
dev.off()

pdf(file = "Output/Output6/SWE2011bDBPswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2011b.out6)
dev.off()

pdf(file = "Output/Output7/SWE2011bMCswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2011b.out7)
dev.off()
pdf(file = "Output/Output8/SWE2011bFARMswe.pdf", width = 12, height = 9,  
  pointsize = 12, bg = "white")  
plot.swe.post_mar_TGsite(tmp2011b.out8)  
dev.off()  

pdf(file = "Output/Output9/SWE2011bTDSwe.pdf", width = 12, height = 9,  
  pointsize = 12, bg = "white")  
plot.swe.post_mar_TGsite(tmp2011b.out9)  
dev.off()  

pdf(file = "Output/Output10/SWE2011bPSswe.pdf", width = 12, height = 9,  
  pointsize = 12, bg = "white")  
plot.swe.post_mar_TGsite(tmp2011b.out10)  
dev.off()  

## Plots for Forecasting After Day 190  

source("plot.swe.post_apr_TGsite.R")  
pdf(file = "Output/Output1/SWE2011cTGswe.pdf", width = 12, height = 9,  
  pointsize = 12, bg = "white")  
plot.swe.post_apr_TGsite(tmp2011c.out1)  
dev.off()  

pdf(file = "Output/Output2/SWE2011cBLPswe.pdf", width = 12, height = 9,  
  pointsize = 12, bg = "white")  
plot.swe.post_apr_TGsite(tmp2011c.out2)  
dev.off()  

pdf(file = "Output/Output3/SWE2011cHRswe.pdf", width = 12, height = 9,  
  pointsize = 12, bg = "white")  
plot.swe.post_apr_TGsite(tmp2011c.out3)  
dev.off()  

source("plot.swe.post_apr_LBsite.R")  
pdf(file = "Output/Output4/SWE2011cLBswe.pdf", width = 12, height = 9,  
  pointsize = 12, bg = "white")  
plot.swe.post_apr_LBsite(tmp2011c.out4)  
dev.off()  

source("plot.swe.post_apr_TGsite.R")  
pdf(file = "Output/Output5/SWE2011cBLswe.pdf", width = 12, height = 9,  
  pointsize = 12, bg = "white")  
plot.swe.post_apr_TGsite(tmp2011c.out5)  
dev.off()  

pdf(file = "Output/Output6/SWE2011cDBPswe.pdf", width = 12, height = 9,  
  pointsize = 12, bg = "white")  
plot.swe.post_apr_TGsite(tmp2011c.out6)  
dev.off()
D.25 Applying the SVD/MCMC Function to the 2012 Water-Year for the Ten Selected UTAH SNOTEL Sites

```r
## 20. Applying The SVD/MCMC ("swe.svd.mcmc.R") Function To The 2012 Water-Year For The Ten Selected UTAH SNOTEL Sites Based on the 4 Selected Important Signals (q = 1, 2, 3, 4)
## 2012 Water-Year MCMC Forecasting After day 100
set.seed(12345)
q <- 4  ## Number Of Important Signals Selected
n.mcmc <- 5000  ## Number of MCMC Simulations To Be Done
outdat <- 100  ## The Day After Which Forecasting Is Done based on The Posterior Distribution
mm <- 100

tmp2012.out1 <- swe.svd.mcmc(t(TGswe[, -35]), q, n.mcmc, mm, outdat)
tmp2012.out2 <- swe.svd.mcmc(t(BLPswe[, -35]), q, n.mcmc, mm, outdat)
tmp2012.out3 <- swe.svd.mcmc(t(HRswe[, -35]), q, n.mcmc, mm, outdat)
tmp2012.out4 <- swe.svd.mcmc(t(LBswe[, -35]), q, n.mcmc, mm, outdat)
tmp2012.out5 <- swe.svd.mcmc(t(BLswe[, -35]), q, n.mcmc, mm, outdat)
tmp2012.out6 <- swe.svd.mcmc(t(DBPswe[, -35]), q, n.mcmc, mm, outdat)
tmp2012.out7 <- swe.svd.mcmc(t(MCswe[, -35]), q, n.mcmc, mm, outdat)
tmp2012.out8 <- swe.svd.mcmc(t(FARMswe[, -35]), q, n.mcmc, mm, outdat)
```
tmp2012.out9 <- swe.svd.mcmc(t(TDswe[, -35]),
  q, n.mcmc, mm, outdat)

tmp2012.out10 <- swe.svd.mcmc(t(PSswe[, -35]),
  q, n.mcmc, mm, outdat)

## 2012 Water-Year MCMC Forecasting After day 130
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 130
mm <- 130

tmp2012a.out1 <- swe.svd.mcmc(t(TGswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012a.out2 <- swe.svd.mcmc(t(BLPswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012a.out3 <- swe.svd.mcmc(t(HRswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012a.out4 <- swe.svd.mcmc(t(LBswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012a.out5 <- swe.svd.mcmc(t(BLswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012a.out6 <- swe.svd.mcmc(t(DBPswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012a.out7 <- swe.svd.mcmc(t(MCswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012a.out8 <- swe.svd.mcmc(t(FARMswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012a.out9 <- swe.svd.mcmc(t(TDswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012a.out10 <- swe.svd.mcmc(t(PSswe[, -35]),
  q, n.mcmc, mm, outdat)

## 2012 Water-Year MCMC Forecasting After day 160
set.seed(12345)
q <- 4
n.mcmc <- 5000
outdat <- 160
mm <- 160

tmp2012b.out1 <- swe.svd.mcmc(t(TGswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012b.out2 <- swe.svd.mcmc(t(BLPswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012b.out3 <- swe.svd.mcmc(t(HRswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012b.out4 <- swe.svd.mcmc(t(LBswe[, -35]),
  q, n.mcmc, mm, outdat)
tmp2012b.out5 <- swe.svd.mcmc(t(BLswe[, -35]),
  q, n.mcmc, mm, outdat)
D.26 Creating the Tables for the Proportion of Observed SWE After Selected Initiation Dates in the 2012 Water-Year

## 21. Creating The Tables For The Proportion Of Observed SWE After 100 (January 8), 130 (February 7), 160 (March 9), and 190 (April 8) In The 2012 Water-Year That Lies Outside The 50% and 95% Credible Intervals Obtained For The Posterior Predictive Distribution

n <- 10  ## Number Of Selected Sites
## Table for Day 100

tmpswe2012 <- list(NA, n)
tmpswe2012[[1]] <- tmp2012.out1
tmpswe2012[[2]] <- tmp2012.out2
tmpswe2012[[3]] <- tmp2012.out3
tmpswe2012[[4]] <- tmp2012.out4
tmpswe2012[[5]] <- tmp2012.out5
tmpswe2012[[6]] <- tmp2012.out6
tmpswe2012[[7]] <- tmp2012.out7
tmpswe2012[[8]] <- tmp2012.out8
tmpswe2012[[9]] <- tmp2012.out9
tmpswe2012[[10]] <- tmp2012.out10

Table2012 <- prop.pred(tmpswe2012)
Table2012 <- data.frame(Table2012)
write.csv(Table2012, file = "Output/Table2012.csv")

## Table for Day 130

tmpswe2012a <- list(NA, n)
tmpswe2012a[[1]] <- tmp2012a.out1
tmpswe2012a[[2]] <- tmp2012a.out2
tmpswe2012a[[3]] <- tmp2012a.out3
tmpswe2012a[[4]] <- tmp2012a.out4
tmpswe2012a[[5]] <- tmp2012a.out5
tmpswe2012a[[6]] <- tmp2012a.out6
tmpswe2012a[[7]] <- tmp2012a.out7
tmpswe2012a[[8]] <- tmp2012a.out8
tmpswe2012a[[9]] <- tmp2012a.out9
tmpswe2012a[[10]] <- tmp2012a.out10

Table2012a <- prop.pred(tmpswe2012a)
Table2012a <- data.frame(Table2012a)
write.csv(Table2012a, file = "Output/Table2012a.csv")

## Table for Day 160

tmpswe2012b <- list(NA, n)
tmpswe2012b[[1]] <- tmp2012b.out1
tmpswe2012b[[2]] <- tmp2012b.out2
tmpswe2012b[[3]] <- tmp2012b.out3
tmpswe2012b[[4]] <- tmp2012b.out4
tmpswe2012b[[5]] <- tmp2012b.out5
tmpswe2012b[[6]] <- tmp2012b.out6
tmpswe2012b[[7]] <- tmp2012b.out7
tmpswe2012b[[8]] <- tmp2012b.out8
tmpswe2012b[[9]] <- tmp2012b.out9
tmpswe2012b[[10]] <- tmp2012b.out10
Table2012b <- prop.prd(tmpswe2012b)
Table2012b <- data.frame(Table2012b)
write.csv(Table2012b, file = "Output/Table2012b.csv")

## Table for Day 190

tmpswe2012c <- list(NA, n)
tmpswe2012c[[1]] <- tmp2012c.out1
tmpswe2012c[[2]] <- tmp2012c.out2
tmpswe2012c[[3]] <- tmp2012c.out3
tmpswe2012c[[4]] <- tmp2012c.out4
tmpswe2012c[[5]] <- tmp2012c.out5
tmpswe2012c[[6]] <- tmp2012c.out6
tmpswe2012c[[7]] <- tmp2012c.out7
tmpswe2012c[[8]] <- tmp2012c.out8
tmpswe2012c[[9]] <- tmp2012c.out9
tmpswe2012c[[10]] <- tmp2012c.out10

Table2012c <- prop.prd(tmpswe2012c)
Table2012c <- data.frame(Table2012c)
write.csv(Table2012c, file = "Output/Table2012c.csv")

D.27 Obtaining the Plots of SWE Measurements and Posterior Distributions
As of the Selected Initiation Dates in the 2012 Water-Year

## 22. Plots of The Ten SNOTEL Site SWE Measurements and Posterior
## Prediction As Of January 8 (Day 100), February 7 (Day 130),
## March 9 (Day 160), and April 8 (Day 190) In The 2012 Water-Year

## Plots for Forecasting After Day 100

source("plot.swe.post_jan_TGsite.R")
pdf(file = "Output/Output1/SWE2012TGswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2012.out1)
dev.off()

pdf(file = "Output/Output2/SWE2012BLPswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2012.out2)
dev.off()

pdf(file = "Output/Output3/SWE2012HRswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2012.out3)
dev.off()

source("plot.swe.post_jan_LBsite.R")
pdf(file = "Output/Output4/SWE2012LBswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_LBsite(tmp2012.out4)
dev.off()

source("plot.swe.post_jan_TGsite.R")
pdf(file = "Output/Output5/SWE2012BLswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2012.out5)
dev.off()

pdf(file = "Output/Output6/SWE2012DBPswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2012.out6)
dev.off()

pdf(file = "Output/Output7/SWE2012MCswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2012.out7)
dev.off()

pdf(file = "Output/Output8/SWE2012FARMswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2012.out8)
dev.off()

pdf(file = "Output/Output9/SWE2012TDswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2012.out9)
dev.off()

pdf(file = "Output/Output10/SWE2012PSswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_jan_TGsite(tmp2012.out10)
dev.off()

## Plots for Forecasting After Day 130

source("plot.swe.post_feb_TGsite.R")
pdf(file = "Output/Output1/SWE2012aTGswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2012a.out1)
dev.off()

pdf(file = "Output/Output2/SWE2012aBLPswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2012a.out2)
dev.off()

pdf(file = "Output/Output3/SWE2012aHRswe.pdf", width = 12, height = 9,
    pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2012a.out3)
dev.off()
source("plot.swe.post_feb_LBsite.R")
pdf(file = "Output/Output4/SWE2012aLBswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_LBsite(tmp2012a.out4)
dev.off()

source("plot.swe.post_feb_TGsite.R")
pdf(file = "Output/Output5/SWE2012aBLswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2012a.out5)
dev.off()

pdf(file = "Output/Output6/SWE2012aDBPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2012a.out6)
dev.off()

pdf(file = "Output/Output7/SWE2012aMCswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2012a.out7)
dev.off()

pdf(file = "Output/Output8/SWE2012aFARMswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2012a.out8)
dev.off()

pdf(file = "Output/Output9/SWE2012aTDswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2012a.out9)
dev.off()

pdf(file = "Output/Output10/SWE2012aPSswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_feb_TGsite(tmp2012a.out10)
dev.off()

## Plots for Forecasting After Day 160

source("plot.swe.post_mar_TGsite.R")
pdf(file = "Output/Output1/SWE2012bTGswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2012b.out1)
dev.off()

pdf(file = "Output/Output2/SWE2012bBLPswe.pdf", width = 12, height = 9,
     pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2012b.out2)
dev.off()
pdf(file = "Output/Output3/SWE2012bHRswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2012b.out3)
dev.off()

source("plot.swe.post_mar_LBsite.R")
pdf(file = "Output/Output4/SWE2012bLBswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_LBsite(tmp2012b.out4)
dev.off()

source("plot.swe.post_mar_TGsite.R")
pdf(file = "Output/Output5/SWE2012bBLswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2012b.out5)
dev.off()

pdf(file = "Output/Output6/SWE2012bDBPswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2012b.out6)
dev.off()

pdf(file = "Output/Output7/SWE2012bMCswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2012b.out7)
dev.off()

pdf(file = "Output/Output8/SWE2012bFARMswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2012b.out8)
dev.off()

pdf(file = "Output/Output9/SWE2012bTDswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2012b.out9)
dev.off()

pdf(file = "Output/Output10/SWE2012bPSswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_mar_TGsite(tmp2012b.out10)
dev.off()

## Plots for Forecasting After Day 190

source("plot.swe.post_apr_TGsite.R")
pdf(file = "Output/Output1/SWE2012cTGswe.pdf", width = 12, height = 9, 
    pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2012c.out1)
dev.off()
pdf(file = "Output/Output2/SWE2012cBLPswe.pdf", width = 12, height = 9, 
  pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2012c.out2)
dev.off()

pdf(file = "Output/Output3/SWE2012cHRswe.pdf", width = 12, height = 9, 
  pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2012c.out3)
dev.off()

source("plot.swe.post_apr_LBsite.R")
pdf(file = "Output/Output4/SWE2012cLBswe.pdf", width = 12, height = 9, 
  pointsize = 12, bg = "white")
plot.swe.post_apr_LBsite(tmp2012c.out4)
dev.off()

source("plot.swe.post_apr_TGsite.R")
pdf(file = "Output/Output5/SWE2012cBLswe.pdf", width = 12, height = 9, 
  pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2012c.out5)
dev.off()

pdf(file = "Output/Output6/SWE2012cDBPswe.pdf", width = 12, height = 9, 
  pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2012c.out6)
dev.off()

pdf(file = "Output/Output7/SWE2012cMCswe.pdf", width = 12, height = 9, 
  pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2012c.out7)
dev.off()

pdf(file = "Output/Output8/SWE2012cFARMswe.pdf", width = 12, height = 9, 
  pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2012c.out8)
dev.off()

pdf(file = "Output_paper/Output9/SWE2012cTDswe.pdf", width = 12, height = 9, 
  pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2012c.out9)
dev.off()

pdf(file = "Output/Output10/SWE2012cPSswe.pdf", width = 12, height = 9, 
  pointsize = 12, bg = "white")
plot.swe.post_apr_TGsite(tmp2012c.out10)
dev.off()
D.28 Computing and Obtaining Barplots of the Ranked Probability Skill Scores


\[ T = 30 \]
\[ \text{dat} \leftarrow \text{vector("list", n.sites)} \]
\[ \text{dat}[1] \leftarrow \text{tmp2008.out1} \]
\[ \text{dat}[2] \leftarrow \text{tmp2008.out2} \]
\[ \text{dat}[3] \leftarrow \text{tmp2008.out3} \]
\[ \text{dat}[4] \leftarrow \text{tmp2008.out4} \]
\[ \text{dat}[5] \leftarrow \text{tmp2008.out5} \]
\[ \text{dat}[6] \leftarrow \text{tmp2008.out6} \]
\[ \text{dat}[7] \leftarrow \text{tmp2008.out7} \]
\[ \text{dat}[8] \leftarrow \text{tmp2008.out8} \]
\[ \text{dat}[9] \leftarrow \text{tmp2008.out9} \]
\[ \text{dat}[10] \leftarrow \text{tmp2008.out10} \]

\[ m = 365 \]
\[ n\text{.sites} = 10 \]
\[ n\text{.mcmc} = 5000 \]
\[ n\text{.burn} = \text{round}(n\text{.mcmc}/4) \]

\[ \text{rpss\text{.ver}} \leftarrow \text{matrix(NA, n.sites)} \]
\[ \text{colnames(rpss\text{.ver})} \leftarrow \text{c("RPSS-Value")} \]
\[ \text{rownames(rpss\text{.ver})} \leftarrow \text{c("TG", "BLP", "HR", "LB", "BL", "DBP", "MC", "FAR", "TD", "PS")} \]
\[ q1 = 0.25 \]
\[ q2 = 0.75 \]
\[ \text{cat} = 3 \]

for (k in 1:n.sites) {
    a <- \text{matrix(apply(dat}[k]$Y$pred[, n\text{.burn}:dat}[k]$n\text{.mcmc}, 1, mean))}
    nr <- \text{nrow(a)}
    b <- \text{matrix(0, nr)}
    d <- \text{as.numeric(quantile(a, probs = c(q1, q2)))}

    for (i in 1:nr) {
        if (a[i,] <= d[1]) {
            b[i,] = 1
        }
        else{
            if (a[i,] > d[1] & a[i,] <= d[2]) {

b[i, ] = 2
}
else{
    b[i, ] = 3
}
}

prob <- matrix(NA, cat)
for (i in 1:cat) {
    prob[i, ] = round(length(which(b==i))/nr, cat)
}
pred <- matrix(rep(prob, nr), nr, byrow = TRUE)

data <- matrix(dat[[k]]$Y[T, ])
obs <- matrix(0, nr)
for (i in 1:nr) {
    if (data[i, ] <= d[1]) {
        obs[i, ] = 1
    }
    else{
        if (data[i, ] > d[1] & data[i, ] <= d[2]) {
            obs[i, ] = 2
        }
        else{
            obs[i, ] = 3
        }
    }
}
ver <- rps(obs = c(obs), pred = pred, baseline = c(0.33, 0.42, 0.25))
rpss.ver[k, ] = round(ver$rpss, 4)
}
rpss2008 <- rpss.ver

T <- 31
dat<- vector("list", n.sites)
dat[[1]] <- tmp2009.out1
dat[[2]] <- tmp2009.out2
dat[[3]] <- tmp2009.out3
dat[[4]] <- tmp2009.out4
dat[[5]] <- tmp2009.out5
dat[[6]] <- tmp2009.out6
dat[[7]] <- tmp2009.out7
dat[[8]] <- tmp2009.out8
dat[[9]] <- tmp2009.out9
dat[[10]] <- tmp2009.out10

m <- 365
n.sites <- 10
n.mcmc <- 5000
n.burn <- round(n.mcmc/4)

rpss.ver <- matrix(NA, n.sites)
colnames(rpss.ver) <- c("RPSS-Value")
rownames(rpss.ver) <- c("TG", "BLP", "HR", "LB", "BL",
  "DBP", "MC", "FAR", "TD", "PS")

q1 <- 0.25
q2 <- 0.75
cat <- 3

for (k in 1:n.sites){
a <- matrix(apply(dat[[1]]$Y.pred[, n.burn:dat[[1]]$n.mcmc], 1, mean))
nr <- nrow(a)
b <- matrix(0, nr)
d <- as.numeric(quantile(a, probs = c(q1, q2)))

for (i in 1:nr) {
  if (a[i, ] <= d[1]) {
    b[i, ] = 1
  }
  else{
    if (a[i, ] > d[1] & a[i, ] <= d[2]) {
      b[i, ] = 2
    }
  else{
    b[i, ] = 3
  }
}
}

prob <- matrix(NA, cat)
for (i in 1:cat) {
  prob[i, ] = round(length(which(b==i))/nr, cat)
}
pred <- matrix(rep(prob, nr), nr, byrow = TRUE)

data <- matrix(dat[[k]]$Y[T, ])
obs <- matrix(0, nr)
for (i in 1:nr) {
  if (data[i, ] <= d[1]) {
    obs[i, ] = 1
  }
  else{
    if (data[i, ] > d[1] & data[i, ] <= d[2]) {
      obs[i, ] = 2
    }
  else{
    obs[i, ] = 3
  }
}
ver <- rps(obs = c(obs), pred = pred, baseline = c(0.30, 0.45, 0.25))

rpss.ver[k, ] = round(ver$rpss, 4)

rpss2009 <- rpss.ver

T <- 32
dat<- vector("list", n.sites)
dat[[1]] <- tmp2010.out1
dat[[2]] <- tmp2010.out2
dat[[3]] <- tmp2010.out3
dat[[4]] <- tmp2010.out4
dat[[5]] <- tmp2010.out5
dat[[6]] <- tmp2010.out6
dat[[7]] <- tmp2010.out7
dat[[8]] <- tmp2010.out8
dat[[9]] <- tmp2010.out9
dat[[10]] <- tmp2010.out10

m <- 365
n.sites <- 10
n.mcmc <- 5000
n.burn <- round(n.mcmc/4)

rpss.ver <- matrix(NA, n.sites)
colnames(rpss.ver) <- c("RPSS-Value")
rownames(rpss.ver) <- c("TG", "BLP", "HR", "LB", "BL",
                          "DBP", "MC", "FAR", "TD", "PS")

q1 <- 0.25
q2 <- 0.75
cat <- 3

for (k in 1:n.sites) {
  a <- matrix(apply(dat[[1]]$Y.pred[, n.burn:dat[[1]]$n.mcmc], 1, mean))
  nr <- nrow(a)
  b <- matrix(0, nr)
  d <- as.numeric(quantile(a, probs = c(q1, q2)))

  for (i in 1:nr) {
    if (a[i, ] <= d[1]) {
      b[i, ] = 1
    } else{
      if (a[i, ] > d[1] & a[i, ] <= d[2]) {
        b[i, ] = 2
      } else{
        b[i, ] = 3
      }
    }
  }
}
prob <- matrix(NA, cat)
for (i in 1:cat) {
  prob[i, ] = round(length(which(b==i))/nr, cat)
}
pred <- matrix(rep(prob, nr), nr, byrow = TRUE)
data <- matrix(dat[[k]]$Y[T, ])
obs <- matrix(0, nr)
for (i in 1:nr) {
  if (data[i, ] <= d[1]) {
    obs[i, ] = 1
  }
  else{
    if (data[i, ] > d[1] & data[i, ] <= d[2]) {
      obs[i, ] = 2
    }
    else{
      obs[i, ] = 3
    }
  }
}
ver <- rps(obs = c(obs), pred = pred, baseline = c(0.25, 0.50, 0.25))
rpss.ver[k, ] = round(ver$rpss, 4)
}

rpss2010 <- rpss.ver

T <- 33
dat<- vector("list", n.sites)
dat[[1]] <- tmp2011.out1
dat[[2]] <- tmp2011.out2
dat[[3]] <- tmp2011.out3
dat[[4]] <- tmp2011.out4
dat[[5]] <- tmp2011.out5
dat[[6]] <- tmp2011.out6
dat[[7]] <- tmp2011.out7
dat[[8]] <- tmp2011.out8
dat[[9]] <- tmp2011.out9
dat[[10]] <- tmp2011.out10

m <- 365
n.sites <- 10
n.mcmc <- 5000
n.burn <- round(n.mcmc/4)

rpss.ver <- matrix(NA, n.sites)
colnames(rpss.ver) <- c("RPSS-Value")
rownames(rpss.ver) <- c("TG", "BLP", "HR", "LB", "BL", "BL", "BL", "BL", "BL", "BL")
q1 <- 0.25
define q2 <- 0.75
cat <- 3

for (k in 1:n.sites) {
  a <- matrix(apply(dat[[1]]$Y.pred[, n.burn:dat[[1]]$n.mcmc], 1, mean))
nr <- nrow(a)
b <- matrix(0, nr)
d <- as.numeric(quantile(a, probs = c(q1, q2)))

  for (i in 1:nr) {
    if (a[i, ] <= d[1]) {
      b[i, ] = 1
    } else{
      if (a[i, ] > d[1] & a[i, ] <= d[2]) {
        b[i, ] = 2
      } else{
        b[i, ] = 3
      }
    }
  }

  prob <- matrix(NA, cat)
  for (i in 1:cat) {
    prob[i, ] = round(length(which(b==i))/nr, cat)
  }
pred <- matrix(rep(prob, nr), nr, byrow = TRUE)

  data <- matrix(dat[[k]]$Y[T, ])
  obs <- matrix(0, nr)
  for (i in 1:nr) {
    if (data[i, ] <= d[1]) {
      obs[i, ] = 1
    } else{
      if (data[i, ] > d[1] & data[i, ] <= d[2]) {
        obs[i, ] = 2
      } else{
        obs[i, ] = 3
      }
    }
  }

  ver <- rps(obs = c(obs), pred = pred, baseline = c(0.27, 0.48, 0.25))
rpss.ver[k, ] = round(ver$rpss, 4)
}
rpss2011 <- rpss.ver
D.28.1 Creating Figure 2.11

```r
library(ggplot2)

# Figure 2.11: Ranked Probability Skill Scores (RPSS) Barplots for the
# Ten Selected SNOTEL Sites for the 2008, 2009, 2010, and
# 2011 Water-Years with the January 8 Initiation Date Used
# in the Posterior Predictive Forecasting

data<-read.csv("Output/RPSS-Barplots(Fig_11).csv")

pdf(file = "Output/RPSS-Barplots(Fig_11).pdf", width = 18, height = 14,
pointsize = 12, bg = "white")

layout (matrix(1:4, 2, 2))

par(mar=c(1.5, 4.5, 2, 0.1))
barplot(t(data[1,]), space = 0.25, col = 1, ylim = c(-0.00135, 0.0148),
       lwd = 2.3, cex.axis = 2, cex.lab = 2, cex.names = 2, ylab = "RPSS",
       axisnames = FALSE)
abline(h=0, lwd = 2)
axis(1, at=c(0.75, 1.95, 3.2, 4.45, 5.72, 7, 8.2, 9.46, 10.74, 12),
     labels = FALSE, font = 2, lwd = 2)
box(lwd = 2)
text(x = 0.28, y = 0.0142, labels = "(a)", cex = 2.5, font = 2)
rect(xleft = -0.25, ybottom = 0.0136, xright = 0.8, ytop = 0.0151)

par(mar=c(3, 4.5, 0, 0.1))
barplot(t(data[2,]), space = 0.25, col = 1, ylim = c(-0.001, 0.024),
       lwd = 2.3, cex.axis = 2, cex.lab = 2, cex.names = 2, ylab = "RPSS")
abline(h=0, lwd = 2)
axis(1, at=c(0.75, 1.95, 3.2, 4.45, 5.72, 7, 8.2, 9.46, 10.74, 12),
     labels = FALSE, font = 2, lwd = 2)
box(lwd = 2)
text(x = 0.28, y = 0.023, labels = "(b)", cex = 2.5, font = 2)
rect(xleft = -0.25, ybottom = 0.022, xright = 0.8, ytop = 0.024)

par(mar=c(1.5, 4, 2, 1))
barplot(t(data[3,]), space = 0.25, col = 1, ylim = c(-0.00021, 0.0045),
       lwd = 2.3, cex.axis = 2, cex.lab = 2, cex.names = 2, ylab = "",
       axisnames = FALSE)
abline(h=0, lwd = 2)
axis(1, at=c(0.75, 1.95, 3.2, 4.45, 5.72, 7, 8.2, 9.46, 10.74, 12),
     labels = FALSE, font = 2, lwd = 2)
box(lwd = 2)
text(x = 0.28, y = 0.00432, labels = "(c)", cex = 2.5, font = 2)
rect(xleft = -0.25, ybottom = 0.00415, xright = 0.8, ytop = 0.0055)

par(mar=c(3, 4, 0, 1))
barplot(t(data[4,]), space = 0.25, col = 1, ylim = c(-0.0026, 0.0003),
       lwd = 2.3, cex.axis = 2, cex.lab = 2, cex.names = 2, ylab = "")
abline(h=0, lwd = 2)
axis(1, at=c(0.75, 1.95, 3.2, 4.45, 5.72, 7, 8.2, 9.46, 10.74, 12),
     labels = FALSE, font = 2, lwd = 2)
box(lwd = 2)
text(x = 0.28, y = 0.00018, labels = "(d)", cex = 2.5, font = 2)
```

rect(xleft = -0.25, ybottom = 0.000073, xright = 0.8, ytop = 0.00035)

dev.off()
APPENDIX E
SWEVIS R PACKAGE PARADIGM (R CODE)

In this appendix, we provide readable and structured R codes for the functions of the developed SWEVIS R package (see Chapter 3). The titles match function names in the developed SWEVIS R package, which are case sensitive.

E.1 ReadSweData

ReadSweData <- function(n.sites = 90, current.year = 2009, loc = "xcoord.txt"){  
## Example: swedata <- ReadSweData(n.sites = 90, current.year = 2009,  
## loc = "xcoord.txt")  
## Example: swedata <- ReadSweData(n.sites = 90, current.year = 2009,  
## loc = "UtahSnotelSites.csv")  
source("read.snotel.R")  
source("krig.swe.R")  
source("utah.snotel.historical.data.R")  
source("utah.snotel.current.data.R")  
require(MCMCpack)  
require(mvtnorm)  
require(spdep)  
require(msm)  
require(lattice)  
require(geoR)  
require(maps)  
require(grid)  

#LatLonCoord <- read.table(loc, header = TRUE)  
LatLonCoord <- data.frame(read.csv(loc, header = TRUE))  

swe.utah <- utah.snotel.historical.data(n.sites, current.year)  
swe.utah.current <- utah.snotel.current.data(n.sites)  

m <- 365 ## Number of days in a Water-Year  
T <- current.year - 1979 ## Number of Historical Water-Years Considered  
## (1979 to the year before the current water-year)  
Y <- array(NA, c(m, T, n.sites)) ## Original Historical Data with Missing  
## Values of SWE  
Y[, 17:T, 1] <- swe.utah[[1]]  
Y[, 3:T, 2] <- swe.utah[[2]]  
Y[, 1:T, 3] <- swe.utah[[3]]
\[
Y[, 1:T, 4] \leftarrow \text{swe.utah}[4]
\]
\[
Y[, 3:T, 5] \leftarrow \text{swe.utah}[5]
\]
\[
Y[, 2:T, 6] \leftarrow \text{swe.utah}[6]
\]
\[
Y[, 4:T, 7] \leftarrow \text{swe.utah}[7]
\]
\[
Y[, 2:T, 8] \leftarrow \text{swe.utah}[8]
\]
\[
Y[, 9:T, 9] \leftarrow \text{swe.utah}[9]
\]
\[
Y[, 1:T, 10] \leftarrow \text{swe.utah}[10]
\]
\[
Y[, 2:T, 11] \leftarrow \text{swe.utah}[11]
\]
\[
Y[, 1:T, 12] \leftarrow \text{swe.utah}[12]
\]
\[
Y[, 8:T, 13] \leftarrow \text{swe.utah}[13]
\]
\[
Y[, 25:T, 14] \leftarrow \text{swe.utah}[14]
\]
\[
Y[, 3:T, 15] \leftarrow \text{swe.utah}[15]
\]
\[
Y[, 1:T, 16] \leftarrow \text{swe.utah}[16]
\]
\[
Y[, 1:T, 17] \leftarrow \text{swe.utah}[17]
\]
\[
Y[, 4:T, 18] \leftarrow \text{swe.utah}[18]
\]
\[
Y[, 23:T, 19] \leftarrow \text{swe.utah}[19]
\]
\[
Y[, 2:T, 20] \leftarrow \text{swe.utah}[20]
\]
\[
Y[, 2:T, 21] \leftarrow \text{swe.utah}[21]
\]
\[
Y[, 1:T, 22] \leftarrow \text{swe.utah}[22]
\]
\[
Y[, 1:T, 23] \leftarrow \text{swe.utah}[23]
\]
\[
Y[, 1, 24] \leftarrow \text{swe.utah}[24][, 1]
\]
\[
Y[, 3:T, 24] \leftarrow \text{swe.utah}[24][, -1]
\]
\[
Y[, 8:T, 25] \leftarrow \text{swe.utah}[25]
\]
\[
Y[, 1:T, 26] \leftarrow \text{swe.utah}[26]
\]
\[
Y[, 17:T, 27] \leftarrow \text{swe.utah}[27]
\]
\[
Y[, 8:T, 28] \leftarrow \text{swe.utah}[28]
\]
\[
Y[, 1:T, 29] \leftarrow \text{swe.utah}[29]
\]
\[
Y[, 26:T, 30] \leftarrow \text{swe.utah}[30]
\]
\[
Y[, 3:T, 31] \leftarrow \text{swe.utah}[31]
\]
\[
Y[, 4:T, 32] \leftarrow \text{swe.utah}[32]
\]
\[
Y[, 27:T, 33] \leftarrow \text{swe.utah}[33]
\]
\[
Y[, 2, 34] \leftarrow \text{swe.utah}[34][, 1]
\]
\[
Y[, 4:T, 34] \leftarrow \text{swe.utah}[34][, -1]
\]
\[
Y[, 27:T, 35] \leftarrow \text{swe.utah}[35]
\]
\[
Y[, 16:T, 36] \leftarrow \text{swe.utah}[36]
\]
\[
Y[, 2:T, 37] \leftarrow \text{swe.utah}[37]
\]
\[
Y[, 1:T, 38] \leftarrow \text{swe.utah}[38]
\]
\[
Y[, 8:T, 39] \leftarrow \text{swe.utah}[39]
\]
\[
Y[, 8:T, 40] \leftarrow \text{swe.utah}[40]
\]
\[
Y[, 8:T, 41] \leftarrow \text{swe.utah}[41]
\]
\[
Y[, 1:T, 42] \leftarrow \text{swe.utah}[42]
\]
\[
Y[, 2:T, 43] \leftarrow \text{swe.utah}[43]
\]
\[
Y[, 30:T, 44] \leftarrow \text{swe.utah}[44]
\]
\[
Y[, 3:T, 45] \leftarrow \text{swe.utah}[45]
\]
\[
Y[, 2:T, 46] \leftarrow \text{swe.utah}[46]
\]
\[
Y[, 2:T, 47] \leftarrow \text{swe.utah}[47]
\]
\[
Y[, 2:T, 48] \leftarrow \text{swe.utah}[48]
\]
\[
Y[, 1:T, 49] \leftarrow \text{swe.utah}[49]
\]
\[
Y[, 2:T, 50] \leftarrow \text{swe.utah}[50]
\]
\[
Y[, 1:T, 51] \leftarrow \text{swe.utah}[51]
\]
\[
Y[, 2:T, 52] \leftarrow \text{swe.utah}[52]
\]
\[
Y[, 1:T, 53] \leftarrow \text{swe.utah}[53]
\]
\[
Y[, 2:T, 54] \leftarrow \text{swe.utah}[54]
\]
\[
Y[, 3:T, 55] \leftarrow \text{swe.utah}[55]
\]
\[
Y[, 8:T, 56] \leftarrow \text{swe.utah}[56]
\]
\[
Y[, 1:T, 57] \leftarrow \text{swe.utah}[57]
\]
\[
Y[, 2:T, 58] \leftarrow \text{swe.utah}[58]
\]
\[
Y[, 3:T, 59] \leftarrow \text{swe.utah}[59]
\]
\[
Y[, 1:T, 60] \leftarrow \text{swe.utah}[60]
\]
Y[, 12:T, 49] <- swe.utah[[49]]
Y[, 4:T, 50] <- swe.utah[[50]]
Y[, 27:T, 51] <- swe.utah[[51]]
Y[, 4:T, 52] <- swe.utah[[52]]
Y[, 1:T, 53] <- swe.utah[[53]]
Y[, 8:T, 54] <- swe.utah[[54]]
Y[, 4:T, 55] <- swe.utah[[55]]
Y[, 8:T, 56] <- swe.utah[[56]]
Y[, 10:T, 57] <- swe.utah[[57]]
Y[, 22:T, 58] <- swe.utah[[58]]
Y[, 2:T, 59] <- swe.utah[[59]]
Y[, 3:T, 60] <- swe.utah[[60]]
Y[, 4:T, 61] <- swe.utah[[61]]
Y[, 11:T, 62] <- swe.utah[[62]]
Y[, 11:T, 63] <- swe.utah[[63]]
Y[, 1:T, 64] <- swe.utah[[64]]
Y[, 1:T, 65] <- swe.utah[[65]]
Y[, 1:T, 66] <- swe.utah[[66]]
Y[, 22:T, 67] <- swe.utah[[67]]
Y[, 3:T, 68] <- swe.utah[[68]]
Y[, 1:T, 69] <- swe.utah[[69]]
Y[, 8:T, 70] <- swe.utah[[70]]

Y[, 2, 71] <- swe.utah[[71]][, 1]
Y[, 4:T, 71] <- swe.utah[[71]][, -1]

Y[, 3:T, 72] <- swe.utah[[72]]
Y[, 4:T, 73] <- swe.utah[[73]]
Y[, 4:T, 74] <- swe.utah[[74]]
Y[, 1:T, 75] <- swe.utah[[75]]
Y[, 12:T, 76] <- swe.utah[[76]]
Y[, 1:T, 77] <- swe.utah[[77]]
Y[, 1:T, 78] <- swe.utah[[78]]
Y[, 24:T, 79] <- swe.utah[[79]]
Y[, 30:T, 80] <- swe.utah[[80]]
Y[, 10:T, 81] <- swe.utah[[81]]
Y[, 1:T, 82] <- swe.utah[[82]]

Y[, 1:T, 83] <- swe.utah[[83]]
Y[61, 30, 83] <- 1.5

Y[, 1:T, 84] <- swe.utah[[84]]
Y[, 2:T, 85] <- swe.utah[[85]]
Y[, 29:T, 86] <- swe.utah[[86]]
Y[, 1:T, 87] <- swe.utah[[87]]
Y[, 3:T, 88] <- swe.utah[[88]]

Y[, 1:4, 89] <- swe.utah[[89]][, 1:4]
Y[, 6:T, 89] <- swe.utah[[89]][, -(1:4)]

Y[, 1:T, 90] <- swe.utah[[90]]
LatLonData <- data.frame(cbind(LatLonCoord$Longitude, LatLonCoord$Latitude))
colnames(LatLonData) <- c("Longitude", "Latitude")

## Augmenting Historical Data with Missing Values of SWE Using Kriging
Y.aug <- krig.swe(Y, LatLonData)
swe.utah.list <- vector("list", length = n.sites)
for (i in 1:n.sites) {
  #cat(i, "\n")
  swe.utah.list[[i]] <- cbind(Y.aug[, , i], as.matrix(swe.utah.current[[i]]))
  #cat("\n")
}
swe.utah.list[[8]][287:365, 31] <- 0

## Write Output
##
list(n.sites = n.sites, T = T, m = m, LatLonCoord = LatLonCoord,
     LatLonData = LatLonData, Y = Y, Y.aug = Y.aug,
     swe.utah.list = swe.utah.list)
}

E.2 CalcSweSumStat
CalcSweSumStat <- function(num, dat){

  n.select <- length(num)
  swe.past.min.max.ave <- matrix(NA, nr = n.select, nc = 3)
  for (i in 1:n.select) {
    swe.data <- dat$swe.utah.list[[num[i]]] ## SWE Measurements for Selected
    swe.data[287:365, 31] <- 0
    swe.past.min.max.ave[i, ] <- cbind(min(swe.data), max(swe.data), mean(swe.data))
  }
  colnames(swe.past.min.max.ave) <- c("Past.min.SWE", "Past.max.SWE",
                                        "Past.ave.SWE")

  site.summary <- data.frame(cbind(dat$LatLonCoord[num, ], swe.past.min.max.ave))
sweinfo <- data.frame(t(site.summary))

list(sweinfo = sweinfo)  ## Write Output

E.3 SimSweMCMCData

SimSweMCMCData <- function(id = 83, dat = swedata, q = 4, n.mcmc = 10,
                          mm = 100, outdat = 100, intdate = "Jan.8",
current.year = 2009) {

    ## Example:
    ## swemcmcdata <- SimSweMCMCData(id = 83, dat = swedata, q = 4,
    ##                                n.mcmc = 10, mm = 100, outdat = 100,
    ##                                intdate = "Jan.8", current.year = 2009)

    require(MCMCpack)
    require(mvtnorm)
    require(spdep)
    require(msm)
    require(lattice)
    require(geoR)
    require(maps)
    require(grid)
    require(fields)

    source("invgammastrt.R")
    source("swe.svd.mcmc.R")

    swe.data <- dat$swe.utah.list[[id]]
    swe.data[287:365, 31] <- 0

    m <- dim(swe.data)[1]
    T <- dim(swe.data)[2]

    # swe.mcmc <- swe.svd.mcmc(t(swe.data[, -T]), q = q, n.mcmc = n.mcmc,
    #                          mm = mm, outdat = outdat)
    swe.mcmc <- swe.svd.mcmc(t(swe.data), q = q, n.mcmc = n.mcmc,
                              mm = mm, outdat = outdat)

    ## Write Output
    ##
    list(n.mcmc = n.mcmc, alpha.save = swe.mcmc$alpha.save,
         s2.save = swe.mcmc$s2.save, Y = swe.mcmc$Y, mu = swe.mcmc$mu,
         mm = mm, Y.pred = swe.mcmc$Y.pred, rho.save = swe.mcmc$rho.save,
         alpha.mean = swe.mcmc$alpha.mean, Y.svd = swe.mcmc$Y.svd,
         n.sites = dat$n.sites, T = T, m = m)
}
E.4 SwePlotData

SwePlotData <- function(dat, n.sites = 90,
num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66)){

## Example:
## sweplotdata <- SwePlotData(dat = swedata, n.sites = 90,
## num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66))

require(MCMCpack)
require(mvtnorm)
require(spdep)
require(msm)
require(lattice)
require(geoR)
require(maps)
require(grid)

T <- dat$T

swe.min.max.ave <- vector("list", length = n.sites)

for (i in 1:n.sites) {
  swe.data <- dat$swe.utah.list[[i]] ## SWE for Selected SNOTEL Site
  swe.data[287:365, 31] <- 0
  swe.min.max.ave[[i]] <- cbind(apply(swe.data, 1, min),
                              apply(swe.data, 1, max),
                              apply(swe.data, 1, mean))
  colnames(swe.min.max.ave[[i]]) <- c("minDay", "maxDay", "aveDay")
}

n.select <- length(num)
swe.min.max.ave.select <- vector("list", length = n.select)

for (i in 1:n.select) {
  swe.min.max.ave.select[[i]] <- swe.min.max.ave[[num[i]]]
}

swe.past.min.max.ave <- matrix(NA, nr = n.select, nc = 3)

for (i in 1:n.select) {
  swe.data <- dat$swe.utah.list[[num[i]]] ## SWE for Selected SNOTEL Site
  swe.data[287:365, 31] <- 0
  swe.past.min.max.ave[i, ] <- cbind(min(swe.data), max(swe.data),
                                     mean(swe.data))
}

colnames(swe.past.min.max.ave) <- c("minPast", "maxPast", "avePast")

site.summary <- data.frame(cbind(dat$LatLonCoord[num, ],
                                swe.past.min.max.ave))

n.days <- dat$m
n.select <- length(num) # 10 Selected SNOTEL Sites
swe.yr <- T
swe.days <- 6

swe.select.list <- vector("list", length = n.select)

for (i in 1:n.select) {
swe.select.list[[i]] <- dat$swe.utah.list[[num[i]]]
swe.select.list[[i]][287:365, 31] <- 0
}

datmat <- matrix(NA, n.days, n.select)

for (i in 1:n.select) {
datmat[, i] <- swe.select.list[[i]][, swe.yr]
}

swemat <- t(datmat)

timepoint <- c("Dec.8 (day 68)", "Jan.8 (day 100)", "Feb.9 (day 132)",
              "Mar.12 (day 164)", "Apr.13 (day 196)", "May.15 (day 228)")

swesite.loc <- dat$LatLonData[num, ]
colnames(swesite.loc) <- c("Lon", "Lat")
swesite.loc <- data.frame(swesite.loc)

VARIOGRAMdata.list <- vector("list", n.days)

for (i in 1:n.days) {
data <- cbind(swesite.loc, swemat[, i])
colnames(data) <- c("x", "y", "swe")
VARIOGRAMdata.list[[i]] <- data[!is.na(data[, 3]), ]
}

library(fields)

swevgdata.list <- vector("list", n.days)

for (i in 1:swe.days) {
loc <- cbind(VARIOGRAMdata.list[[68+32*(i-1)]][, 1],
            VARIOGRAMdata.list[[68+32*(i-1)]][, 2])
swe.vg <- vgram(loc, VARIOGRAMdata.list[[68+32*(i-1)]][, 3],
                N = 20, dmax = 300, lon.lat = T)
swevgdata.list[[i]] <- data.frame(swe.vg$d, swe.vg$vgram)
colnames(swevgdata.list[[i]]) <- c("d", "vgram")
}

locmarkers <- swesite.loc
loc <- data.frame(locmarkers)
rownames(loc) <- c(1:10)
out <- rdist.earth(loc)
upper <- col(out) > row(out)

swe_vgram_loc_mat <- matrix(NA, length(out[upper]), 8)
colnames(swe_vgram_loc_mat) <- c("loc1", "loc2", "loc1Lon", "loc1Lat",
"loc2Lon", "loc2Lat", "d", "vgram")

swevgramdata.list <- vector("list", n.days)
for (k in 1:swe.days) {
    swe_vgram_loc_mat[, c("d", "vgram")]
        <- c(swevgdata.list[[k]]$d, swevgdata.list[[k]]$vgram)
    swevgramdata.list[[k]] <- swe_vgram_loc_mat
}

matrix.index <- function(a, value){
    idx <- which(data.frame(a) == value)
    col.num <- ceiling(idx/nrow(a))
    row.num <- idx - (col.num - 1) * nrow(a)
    return(c(row.num, col.num))
}

l1 <- c(1:9)
l2 <- c(2:10)

for (i in 1:9) {
    for (j in i:9) {
        for (k in 1:swe.days) {
            dist_value <- swevgramdata.list[[k]][swevgramdata.list[[k]][, "d"] ==
                rdist.earth(locmarkers[c(l1[i], l2[j]), ], )][1, 2], "d"]
            rnum <- matrix.index(swevgramdata.list[[k]], dist_value)[1]
            swevgramdata.list[[k]][rnum, c("loc1","loc2", "loc1Lon", "loc1Lat",
                "loc2Lon", "loc2Lat")]
                <- c(l1[i], l2[j], loc[l1[i], "Lon"],
                loc[l1[i], "Lat"], loc[l2[j], "Lon"],
                loc[l2[j], "Lat"])
        }
    }
}

## Write Output
##
list(swe.data = swe.data, n.sites = dat$n.sites, T = dat$T, m = dat$m,
    site.summary = site.summary, LatLonCoord = dat$LatLonCoord,
    swe.select.list = swe.select.list, swevgramdata.list = swevgramdata.list,
    swe.min.max.ave = swe.min.max.ave, locmarkers = locmarkers,
    swe.past.min.max.ave = swe.past.min.max.ave,
    swe.min.max.ave.select = swe.min.max.ave.select)
E.5 SweRawDataPlot

SweRawDataPlot <- function(id = c(4, 53, 64, 82), dat, sig = FALSE, common.scale = TRUE, col = "light grey", xlim1 = c(0, 60), ylim1 = c(0, 150), ylim2 = c(0, 80), cex.lab = 1.6, cex.axis = 1.7, boxlwd = 1.5, main = ""){

## Example: SweRawDataPlot(id = 42, dat = swedata, sig = TRUE, col = "light grey", cex.lab = 1.6, cex.axis = 1.7, boxlwd = 1.5, main = "")
## Example: SweRawDataPlot(id = c(4, 53, 64, 82), dat = sweplotdata, sig = FALSE, common.scale = TRUE, col = "light grey", xlim1 = c(0, 60), ylim1 = c(0, 150), ylim2 = c(0, 80), cex.lab = 1.6, cex.axis = 1.7, boxlwd = 1.5, main = "")

if (sig = TRUE) {
  ## Single Site Selection
  swe.data <- dat$swe.utah.list[[id]] ## SWE for Selected SNOTEL Site
  swe.data[287:365, 31] <- 0
  m <- dim(swe.data)[1]
  T <- dim(swe.data)[2]
  layout (matrix(1:2, 1, 2, byrow = TRUE))
  # hist(swe.data[, -T], xlab = "SWE (inches)",
  # col = col, main = main, cex.lab = cex.lab, cex.axis = cex.axis)
  hist(swe.data[, T], xlab = "SWE (inches)",
        col = col, main = main, cex.lab = cex.lab, cex.axis = cex.axis)
  boxplot(swe.data[, -T], xlab = "Water-Year", ylab = "SWE (inches)",
          main = "", col = col, cex.lab = cex.lab, cex.axis = cex.axis,
          boxlwd = boxlwd)
}
else {
  ## Multiple Sites Selection
  swe.data <- dat$swe.utah.list[[id[1]]] ## SWE for Selected SNOTEL Site
  swe.data[287:365, 31] <- 0
  m <- dim(swe.data)[1]
  T <- dim(swe.data)[2]
  nt <- length(id)
  nr <- nt
  nc <- 2
  if (common.scale = TRUE) {

  } else {
    if (sig = FALSE) {
      ## Single Site Selection
      swe.data[287:365, 31] <- 0
      m <- dim(swe.data)[1]
      T <- dim(swe.data)[2]
      nt <- length(id)
      nr <- nt
      nc <- 2
      if (common.scale = TRUE) {

      } else {
        if (sig = FALSE) {
          ## Multiple Sites Selection
          swe.data[287:365, 31] <- 0
          m <- dim(swe.data)[1]
          T <- dim(swe.data)[2]
          nt <- length(id)
          nr <- nt
          nc <- 2
          if (common.scale = TRUE) {

          }
layout(matrix(1:(nr*nc), nrow = nr, ncol = nc, byrow = TRUE))

for (i in 1:(nt-1)) {
  par(mar = c(4, 5, 0.8, 1))
  hist((dat$swe.utah.list[[id[i]]])[, T], xlab = " ", main = " ",
       col = col, cex.lab = cex.lab, cex.axis = cex.axis,
       xlim = xlim1, ylim = ylim1)
  par(mar = c(4, 5, 0.5, 1))
  boxplot((dat$swe.utah.list[[id[i]]])[, -T], xlab = " ",
           ylab = "SWE (inches)", main = " ", col = col,
           cex.lab = cex.lab, cex.axis = cex.axis,
           boxlwd = boxlwd, ylim = ylim2)
}

par(mar = c(4.5, 5, 0, 1))
hist((dat$swe.utah.list[[id[nt]]])[, T], xlab = "SWE (inches)",
     main = " ", col = col, cex.lab = cex.lab, cex.axis = cex.axis,
     xlim = xlim1, ylim = ylim1)

par(mar = c(4.5, 5, 0, 1))
boxplot((dat$swe.utah.list[[id[nt]]])[, -T], xlab = "Water-Year",
       ylab = "SWE (inches)", main = " ", col = col,
       cex.lab = cex.lab, cex.axis = cex.axis,
       boxlwd = boxlwd, ylim = ylim2)
}

else{
  if (common.scale = FALSE) {
    layout(matrix(1:(nr*nc), nrow = nr, ncol = nc, byrow = TRUE))

    for (i in 1:(nt-1)) {
      par(mar = c(4, 5, 0.8, 1))
      hist((dat$swe.utah.list[[id[i]]])[, T], xlab = " ", main = " ",
           col = col, cex.lab = cex.lab, cex.axis = cex.axis)
      par(mar = c(4, 5, 0.5, 1))
      boxplot((dat$swe.utah.list[[id[i]]])[, -T], xlab = " ",
               ylab = "SWE (inches)", main = " ", col = col,
               cex.lab = cex.lab, cex.axis = cex.axis,
               boxlwd = boxlwd)
    }
    par(mar = c(4.5, 5, 0, 1))
    hist((dat$swe.utah.list[[id[nt]]])[, T], xlab = "SWE (inches)",
         main = " ", col = col, cex.lab = cex.lab, cex.axis = cex.axis)
    par(mar = c(4.5, 5, 0, 1))
    boxplot((dat$swe.utah.list[[id[nt]]])[, -T], xlab = "Water-Year",
             ylab = "SWE (inches)", main = " ", col = col,
             cex.lab = cex.lab, cex.axis = cex.axis,
             boxlwd = boxlwd)
  }
}

E.6 SweHistPlot

SweHistPlot <- function(id = c(4, 53, 64, 82), dat, sig = FALSE,
  common.scale = TRUE, col = "light grey",
  xlim1 = c(0, 20), xlim2 = c(0, 80),
  xlim3 = c(0, 40), ylim1 = c(0, 300),
  ylim2 = c(0, 200), ylim3 = c(0, 220),
  cex.lab = 1.5, cex.axis = 1.5, main = " "){

  if (sig = TRUE) {
    layout (matrix(1:3, 1, 3, byrow = TRUE))
    hist((dat$swe.min.max.ave[[id]])[, "minDay"], xlab = "Daily Minimum SWE",
         main = " ", col = col, cex.lab = cex.lab, cex.axis = cex.axis)
    hist((dat$swe.min.max.ave[[id]])[, "maxDay"], xlab = "Daily Maximum SWE",
         ylab = " ", col = col, main = main, cex.lab = cex.lab, cex.axis = cex.axis)
    hist((dat$swe.min.max.ave[[id]])[, "aveDay"], xlab = "Daily Average SWE",
         ylab = " ", main = " ", col = col, cex.lab = cex.lab, cex.axis = cex.axis)
  }

  else {
    if (sig = FALSE) {
      layout (matrix(1:3, 1, 3, byrow = TRUE))
      hist((dat$swe.min.max.ave[[id]])[, "minDay"], xlab = "Daily Minimum SWE",
           main = " ", col = col, cex.lab = cex.lab, cex.axis = cex.axis)
      hist((dat$swe.min.max.ave[[id]])[, "maxDay"], xlab = "Daily Maximum SWE",
           ylab = " ", col = col, main = main, cex.lab = cex.lab, cex.axis = cex.axis)
      hist((dat$swe.min.max.ave[[id]])[, "aveDay"], xlab = "Daily Average SWE",
           ylab = " ", main = " ", col = col, cex.lab = cex.lab, cex.axis = cex.axis)
    }
  }
}

## Example: SweHistPlot(id = 42, dat = sweplotdata,
##                        sig = TRUE, col = "light grey",
##                        cex.lab = 1.5, cex.axis = 1.5, main = " ")
##
## Example: SweHistPlot(id = c(4, 53, 64, 82), dat = sweplotdata,
##                        sig = FALSE, common.scale = TRUE,
##                        col = "light grey", cex.lab = 1.5,
##                        cex.axis = 1.5,main = " ")

if (sig = TRUE) {
  ## Single Site Selection
  layout (matrix(1:3, 1, 3, byrow = TRUE))
  hist((dat$swe.min.max.ave[[id]])[, "minDay"], xlab = "Daily Minimum SWE",
       main = " ", col = col, cex.lab = cex.lab, cex.axis = cex.axis)
  hist((dat$swe.min.max.ave[[id]])[, "maxDay"], xlab = "Daily Maximum SWE",
       ylab = " ", col = col, main = main, cex.lab = cex.lab, cex.axis = cex.axis)
  hist((dat$swe.min.max.ave[[id]])[, "aveDay"], xlab = "Daily Average SWE",
       ylab = " ", main = " ", col = col, cex.lab = cex.lab, cex.axis = cex.axis)
}

else {
  if (sig = FALSE) {
    ## Multiple Sites Selection
    nt <- length(id)
    nr <- nt
    nc <- 3

    if (common.scale = TRUE) {
      layout (matrix(1:(nr*nc), nrow = nr, ncol = nc, byrow = TRUE))
      for (i in 1:(nt-1)) {
        par(mar = c(5, 5, 1, 1))
      }
    }
hist((dat$swe.min.max.ave[[id[i]]])[, "minDay"], xlab = " ",
main = " ", col = col, cex.lab = cex.lab, cex.axis = cex.axis,
xlim = xlim1, ylim = ylim1)
par(mar = c(5, 3, 1, 1))
hist((dat$swe.min.max.ave[[id[i]]])[, "maxDay"], xlab = " ",
ylab = " ", col = col, main = main, cex.lab = cex.lab,
cex.axis = cex.axis, xlim = xlim2, ylim = ylim2)
par(mar = c(5, 3, 1, 1))
hist((dat$swe.min.max.ave[[id[i]]])[, "aveDay"], xlab = " ",
ylab = " ", main = " ", col = col, cex.lab = cex.lab,
cex.axis = cex.axis, xlim = xlim3, ylim = ylim3)
}

par(mar = c(5, 5, 1, 1))
hist((dat$swe.min.max.ave[[id[nt]]])[, "minDay"],
xlab = "Daily Minimum SWE", main = " ", col = col,
cex.lab = cex.lab, cex.axis = cex.axis,
xlim = xlim1, ylim = ylim1)
par(mar = c(5, 3, 1, 1))
hist((dat$swe.min.max.ave[[id[nt]]])[, "maxDay"],
lab = "Daily Maximum SWE", ylab = " ",
col = col, main = main,
cex.lab = cex.lab, cex.axis = cex.axis,
xlim = xlim2, ylim = ylim2)
par(mar = c(5, 3, 1, 1))
hist((dat$swe.min.max.ave[[id[nt]]])[, "aveDay"],
lab = "Daily Average SWE", ylab = " ",
col = col, cex.lab = cex.lab, cex.axis = cex.axis,
xlim = xlim3, ylim = ylim3)
}
else {
  if (common.scale = FALSE) {
    layout (matrix(1:(nr*nc), nrow = nr, ncol = nc, byrow = TRUE))

    for (i in 1:(nt-1)) {
      par(mar = c(5, 5, 1, 1))
      hist((dat$swe.min.max.ave[[id[i]]])[, "minDay"], xlab = " ",
main = " ", col = col, cex.lab = cex.lab, cex.axis = cex.axis)
      par(mar = c(5, 3, 1, 1))
      hist((dat$swe.min.max.ave[[id[i]]])[, "maxDay"], xlab = " ",
ylab = " ", col = col, main = main, cex.lab = cex.lab,
cex.axis = cex.axis)
      par(mar = c(5, 3, 1, 1))
      hist((dat$swe.min.max.ave[[id[i]]])[, "aveDay"], xlab = " ",
ylab = " ", main = " ", col = col, cex.lab = cex.lab,
cex.axis = cex.axis)
    }

    par(mar = c(5, 5, 1, 1))
    hist((dat$swe.min.max.ave[[id[nt]]])[, "minDay"],

xlab = "Daily Minimum SWE", main = " ", col = col,
cex.lab = cex.lab, cex.axis = cex.axis)
par(mar = c(5, 3, 1, 1))
hist((dat$swe.min.max.ave[[id[nt]]])[, "maxDay"],
xlab = "Daily Maximum SWE", ylab = " ",
col = col, main = main,
cex.lab = cex.lab, cex.axis = cex.axis)
par(mar = c(5, 3, 1, 1))
hist((dat$swe.min.max.ave[[id[nt]]])[, "aveDay"],
xlab = "Daily Average SWE", ylab = " ", main = " ",
col = col, cex.lab = cex.lab, cex.axis = cex.axis)
E.7  SweBoxPlot

SweBoxPlot <- function(id = c(4, 53, 64, 82), dat, sig = FALSE,
  common.scale = TRUE, col = "light grey",
  ylim = c(0, 75), cex.lab = 1.7,
  cex.axis = 1.7, boxlwd = 2, main = " "){

  ## Example: SweBoxPlot(id = 83, dat = sweplotdata, sig = TRUE,
  ##   col = "light grey", cex.lab = 1.5,
  ##   cex.axis = 1.5, boxlwd = 2, main = " ")
  ## Example: SweBoxPlot(id = c(4, 53, 64, 82), dat = sweplotdata,
  ##   sig = FALSE, common.scale = TRUE,
  ##   col = "light grey", ylim = c(0, 75),
  ##   cex.lab = 1.5, cex.axis = 1.5,
  ##   boxlwd = 2, main = " ")

  if (sig = TRUE) {
    ## Single Site Selection
    layout (matrix(1:3, 1, 3, byrow = TRUE))
    boxplot((dat$swe.min.max.ave[[id]])[, "minDay"], xlab = "Daily Minimum",
      ylab = "SWE (inches)", main = " ", col = col, cex.lab = cex.lab,
      cex.axis = cex.axis, boxlwd = boxlwd)
    boxplot((dat$swe.min.max.ave[[id]])[, "maxDay"], xlab = "Daily Maximum",
      ylab = " ", col = col, main = main, cex.lab = cex.lab,
      cex.axis = cex.axis, boxlwd = boxlwd)
    boxplot((dat$swe.min.max.ave[[id]])[, "aveDay"], xlab = "Daily Average",
      ylab = " ", main = " ", col = col, cex.lab = cex.lab,
      cex.axis = cex.axis, boxlwd = boxlwd)
  }
  else {
    if (sig = FALSE) {
      ## Multiple Sites Selection
      nt <- length(id)  ## Should be a multiple of 2
      nr <- nt/2
      nc <- 2

      if (common.scale = TRUE) {

        layout (matrix(1:(nr*nc), nrow = nr, ncol = nc, byrow = TRUE))

        for (i in 1:(nr - 1)) {
          par(mar = c(1, 5, 1, 0))
          boxplot((dat$swe.min.max.ave[[id[2*i - 1]]]), xlab = " ",
            ylab = "SWE (inches)", main = " ", ylim = ylim,
            col = col, cex.lab = cex.lab, cex.axis = cex.axis,
            boxlwd = boxlwd, lwd = 2, axes = FALSE)
          axis(1, at = c(0.995, 1.995, 2.995), labels = c(" ", " ", " "),
            cex.axis = cex.axis)
          axis(2, cex.axis = cex.axis)
          box(lwd = 2)
        }
      }
      }}
par(mar = c(1, 0.8, 1, 3))
boxplot((dat$swe.min.max.ave[[id[2*i]]]), xlab = "", ylab = " ",
main = " ", ylim = ylim, col = col, cex.lab = cex.lab,
cex.axis = cex.axis, boxlwd = boxlwd, lwd = 2, axes = FALSE)
axis(1, at = c(0.995, 1.995, 2.995), labels = c(" ", " ", " "),
cex.axis = cex.axis)
axis(2, cex.axis = cex.axis, labels = FALSE)
box(lwd = 2)
}

par(mar = c(2.5, 5, 0, 0))
boxplot((dat$swe.min.max.ave[[id[nt - 1]]]), xlab = " ",
ylab = "SWE (inches)", main = " ", ylim = ylim,
col = col, cex.lab = cex.lab, cex.axis = cex.axis,
boxlwd = boxlwd, lwd = 2, axes = FALSE)
axis(1, at = c(0.995, 1.995, 2.995), cex.axis = cex.axis,
labels = c("Daily Minimum", "Daily Maximum", "Daily Average"))
axis(2, cex.axis = cex.axis)
box(lwd = 2)
par(mar = c(2.5, 0.8, 1, 3))
boxplot((dat$swe.min.max.ave[[id[nt]]]), xlab = " ", ylab = " ",
main = " ", ylim = ylim, col = col, cex.lab = cex.lab,
cex.axis = cex.axis, boxlwd = boxlwd, lwd = 2, axes = FALSE)
axis(1, at = c(0.995, 1.995, 2.995), cex.axis = cex.axis,
labels = c("Daily Minimum", "Daily Maximum", "Daily Average"))
axis(2, cex.axis = cex.axis, labels = FALSE)
box(lwd = 2)
}
else {
  if (common.scale = FALSE) {
    layout (matrix(1:(nr*nc), nrow = nr, ncol = nc, byrow = TRUE))
    for (i in 1:(nr - 1)) {
      par(mar = c(1, 5, 1, 0))
      boxplot((dat$swe.min.max.ave[[id[2*i - 1]]]), xlab = " ",
ylab = "SWE (inches)", main = " ",
col = col, cex.lab = cex.lab, cex.axis = cex.axis,
boxlwd = boxlwd, lwd = 2, axes = FALSE)
      axis(1, at = c(0.995, 1.995, 2.995), labels = c(" ", " ", " "),
cex.axis = cex.axis)
      axis(2, cex.axis = cex.axis)
      box(lwd = 2)
      par(mar = c(1, 0.8, 1, 3))
      boxplot((dat$swe.min.max.ave[[id[2*i]]]), xlab = " ", ylab = " ",
main = " ", col = col, cex.lab = cex.lab,
cex.axis = cex.axis, boxlwd = boxlwd, lwd = 2, axes = FALSE)
      axis(1, at = c(0.995, 1.995, 2.995), labels = c(" ", " ", " "),
cex.axis = cex.axis)
SweVariogPlot <- function(dat, tpnum = 1, cex = 1.4, cex.lab = 1.3, 
cex.axis = 1.3, lwd = 1.7, col = "black", 
main = paste("Variogram.Cloud - ", timepoint[tpnum])){

  ## Example:
  ## SweVariogPlot(dat = sweplotdata, tpnum = 1, cex = 1.4, cex.lab = 1.3, 
  ## cex.axis = 1.3, lwd = 1.7, col = "black", 
  ## main = paste("Variogram.Cloud - ", timepoint[tpnum]))

  timepoint <- c("Dec.8 (day 68)", "Jan.8 (day 100)", "Feb.9 (day 132)", 
              "Mar.12 (day 164)", "Apr.13 (day 196)", "May.15 (day 228)")

  plot(dat$swevgramdata.list[[tpnum]][, c("d", "vgram")], 
       cex = cex, cex.lab = cex.lab, cex.axis = cex.axis, 
       lwd = lwd, col = col, 
       ylab = "Semivariance", 
       xlab = "Distance (in miles)", 
       main = main)
}
E.9 SweRgoogleMap

SweRgoogleMap <- function(lat = c(37, 42), lon = c(-114.2, -109),
                        destfile = "UTAH.png", maptype = "hybrid"){

    ## Example: SweRgoogleMap(lat = c(37, 42), lon = c(-114.2, -109),
    ## destfile = "UTAH.png", maptype = "hybrid")

    library(RgoogleMaps)
    library(PBSmapping)
    library(ReadImages)  ## For jpg format
    ##library(rgdal)  ## For png format
    library(png)  ## For png format

    ## This is needed while RgoogleMaps is being updated on CRAN

    bb <- qbbox(lat = lat, lon = lon)

    MyMap <- GetMap.bbox(bb$lonR, bb$latR, destfile = destfile,
                         maptype = maptype)

    #PlotOnStaticMap(MyMap)
    #lat = bb$latR; lon = bb$lonR
}

E.10 SwePostPlot

SwePostPlot <- function(dat = swemcmcdata, sig = TRUE, intdate = "Jan.8",
                        current.year = 2009, labnum = 1, 
                        date = "Jan. 8, 2009 --"){

    ## Example: SwePostPlot(dat = swemcmcdata, sig = TRUE, 
    ## intdate = "Jan.8", current.year = 2009, labnum = 1, 
    ## date = "Jan. 8, 2009 --")

    if (sig = TRUE) {
        ## Single Site Selection
        # source("plot.swe.post_jan_Fig8_HRsite.R")
        swe.post.function <- c("plot.swe.post_jan_Fig8_TGsite.R",
                                "plot.swe.post_feb_Fig8_TGsite.R",
                                "plot.swe.post_mar_Fig8_TGsite.R",
                                "plot.swe.post_apr_Fig8_TGsite.R")

        source(swe.post.function[labnum])
    }
plot.swe.post(dat = dat, date = date)
}

else {
  if (sig = FALSE) {
    # Multiple Sites Selection
    nt <- length(dat)  # Should be a multiple of 2
    nr <- nt/2
    nc <- 2

    dates <- c("Jan. 8 --", "Feb. 7 --", "Mar. 9 --", "Apr. 8 --")

    swe.post.function <- c("plot.swe.post_jan_TGsite.R", 
                           "plot.swe.post_feb_TGsite.R", 
                           "plot.swe.post_mar_TGsite.R", 
                           "plot.swe.post_apr_TGsite.R")

    layout (matrix(1:(nr*nc), nrow = nr, ncol = nc, byrow = TRUE))

    source(swe.post.function[labnum])

    for (i in 1:nt){
      par(mar = c(2, 3, 2, 1))
      plot.swe.post(dat = dat[[i]], date = dates[labnum])
    }
  }
}

E.11 iSweBrushMapSingle

iSweBrushMapSingle <- function(i = 1, dat1 = swedata, dat2 = sweplotdata, 
num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66), 
q = 4, n.mcmc = 1000, mm = 100, outdat = 100, 
intdate = "Jan.8", current.year = 2009, 
date = "Jan. 8, 2009"){
  # Example:
  # iSweBrushMapSingle(i = 1, dat1 = swedata, dat2 = sweplotdata, 
  # num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66), 
  # q = 4, n.mcmc = 200, mm = 100, outdat = 100, 
  # intdate = "Jan.8", current.year = 2009, 
  # date = "Jan. 8, 2009")

  library(png)

  while (!is.null(ievent.wait())) {
    iset.set(i)
if (iset.sel.changed()) {
  s <- iset.selected()
  if (length(s) == 1)
    sitenum = num[s]
  m <- dim(dat1$swe.utah.list[[sitenum]])[1]
  T <- dim(dat1$swe.utah.list[[sitenum]])[2]
  source("SimSweMCMCData.R")
  swemcmcdata <- SimSweMCMCData(id = sitenum, dat = dat1, q = q,
      n.mcmc = n.mcmc, mm = mm, outdat = outdat,
      intdate = intdate, current.year = current.year)
  swe.post.function <- c("plot.swe.post_jan_Fig8_TGsite.R",
    "plot.swe.post_jan_Fig8_BLPsite.R",
    "plot.swe.post_jan_Fig8_HRsite.R",
    "plot.swe.post_jan_Fig8_LBsite.R",
    "plot.swe.post_jan_Fig8_BLsite.R",
    "plot.swe.post_jan_Fig8_DBPsite.R",
    "plot.swe.post_jan_Fig8_MCsite.R",
    "plot.swe.post_jan_Fig8_FARsite.R",
    "plot.swe.post_jan_Fig8_TDsite.R",
    "plot.swe.post_jan_Fig8_PSsite.R")
  png(file = "plotswepost.png", width = 1100, height = 700,
      units = "px", pointsize = 12)
  source(swe.post.function[sitenum])
  plot.swe.post(dat = swemcmcdata, date = date)
  dev.off()
  pngswepost <- readPNG("plotswepost.png", native = TRUE)
  xcoord <- c(1:m)
  ycoord <- c(dat2$swe.select.list[[s]][, T])
  xycoord <- data.frame(xcoord, ycoord)
  colnames(xycoord) <- c("Days.in.Water-Year", "SWE")
  iplotswepost <- function(pngswepost, x1, y1, x2, y2){
    iplot(xycoord[, "Days.in.Water-Year"], xycoord[, "SWE"],
      main = "Posterior Predictive Plot",
      xname = "Days.in.Water-Year", yname = "SWE")
    iplot.opt(xlim = c(x1, x2), ylim = c(y1, y2))
    iplot.opt(col = c(5, 5), ptDiam = 10)
    ilines(xycoord[, "SWE"])
    iraster(x1, y1, x2, y2, pngswepost, layer = -2)
  }
  y.max <- max(swemcmcdata$Y, na.rm = TRUE)
  iplotswepost(pngswepost = pngswepost, x1 = -12, y1 = -1.2,
    x2 = 375, y2 = y.max + 1)
  min.max.ave.data <- data.frame(dat2$swe.min.max.ave[[sitenum]])
  colnames(min.max.ave.data) <- c("Daily Minimum", "Daily Maximum",
    "Daily Average")
col_names <- c("Daily Minimums", "Daily Maximums", "Daily Averages")

## Interactive Boxplot
ibox(min.max.ave.data, ylab = "SWE (inches)", main = "SWE Boxplots")

## Interactive Histogram
for (k in 1:3) {
    ihist(min.max.ave.data[, k], xlab = col_names[k], ylab = "Frequency",
          main = paste("SWE Histogram for", col_names[k]))
}

ihist(dat2$swe.select.list[[s]][, T], main = paste("SWE Histogram --",
                                                current.year, "Water-Year"))
else
    iobj.opt(visible = TRUE)
}

## NOTE: To discontinue any brushing tool, click on "file" and then "break".

E.12 iSwePlot

iSwePlot <- function(dat1 = swedata, dat2 = sweplotdata, datenum = 1,
                       num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66),
                       x1 = -114.2, y1 = 37, x2 = -108.9, y2 = 42.1,
                       destfile = "UTAH.png", maptype = "hybrid"){

    ## Example: iSwePlot(dat1 = swedata, dat2 = sweplotdata, datenum = 1,
    ## num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66),
    ## x1 = -114.2, y1 = 37, x2 = -108.9, y2 = 42.1,
    ## destfile = "UTAH.png", maptype = "hybrid")

    library(RgoogleMaps)
    library(PBSmapping)
    library(ReadImages) ## For jpg format
    #library(rgdal) ## For png format
    library(png) ## For png format

    ## This is needed while RgoogleMaps is being updated on CRAN

    lat <- c(y1, y2)
    lon <- c(x1, x2)

    bb <- qbbox(lat = lat, lon = lon)

    MyMap <- GetMap.bbox(bb$lonR, bb$latR, destfile = destfile, maptype = maptype)
install.packages("iplots", ,"http://rforge.net", type = "source")

pngmap <- readPNG(destfile, native = TRUE)

locmarkers <- dat1$LatLonData[num, ]
colnames(locmarkers) <- c("Lon", "Lat")
locmarkers <- data.frame(locmarkers)

ibitmap <- function(pngmap, x1, y1, x2, y2){
  iplot(locmarkers[, "Lon"], locmarkers[, "Lat"],
        main = "Plotting Over an Image (Latitude vs Longitude)",
        xname = "Longitude", yname = "Latitude")
  iplot.opt(xlim = c(x1, x2), ylim = c(y1, y2))
  iplot.opt(col = c(5, 5), ptDiam = 18)
  iraster(x1, y1, x2, y2, pngmap, layer = -2)
}

## Plot 1: Bitmap Plot (RgoogleMap as Background)
ibitmap(pngmap, x1 = x1, y1 = y1, x2 = x2, y2 = y2)

## Plot 2: Plot of Location Numbers
iplot(c(rep(dat2$swevgramdata.list[[datenum]][, "loc1"], 1)),
      c(rep(dat2$swevgramdata.list[[datenum]][, "loc2"], 1)),
      cex=1.2, lwd=1.6, xlab = "Location_1", ylab = "Location_2",
      main = paste("Plot_of_SNOTEL_Sites"))
iplot.opt(col = c(5, 5), ptDiam = 16)

## Plot 3: Variogram Cloud Plot Based on the Selected SNOTEL Sites
## SWE Measurements

timepoint = c("Dec.8 (day 68)", "Jan.8 (day 100)", "Feb.9 (day 132)",
            "Mar.12 (day 164)", "Apr.13 (day 196)", "May.15 (day 228)"
            )

dat2$swevgramdata.list[[datenum]] <-
data.frame(dat2$swevgramdata.list[[datenum]])
attach(dat2$swevgramdata.list[[datenum]])
iplot(c(rep(dat2$swevgramdata.list[[datenum]][, "d"], 1)),
      c(rep(dat2$swevgramdata.list[[datenum]][, "vgram"], 1)),
      cex=1.2, lwd=1.6, xlab = "Semivariance", ylab = "Distance (in miles)",
      main = paste("Variogram.Cloud - ",timepoint[datenum]))
iplot.opt(col = c(5, 5), ptDiam = 16)
}
iSweBrushMap <- function(i = 1, j = 2, datenum = 1, 
  dat1 = swedata, dat2 = sweplotdata, 
  num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66)) {

  ## Example: 
  ## iSweBrushMap(i = 1, j = 2, datenum = 1, 
  ##    dat1 = swedata, dat2 = sweplotdata, 
  ##    num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66))

  matrix.index <- function(a, value) {
    idx <- which(data.frame(a) == value)
    col.num <- ceiling(idx/nrow(a))
    row.num <- idx - (col.num - 1) * nrow(a)
    return(c(row.num, col.num))
  }

  locmarkers <- dat1$LatLonData[num, ]
  colnames(locmarkers) <- c("Lon", "Lat")
  locmarkers <- data.frame(locmarkers)

  while (!is.null(ievent.wait())) {
    iset.set(i)
    if (iset.sel.changed()) {
      s <- iset.selected()
      loc <- data.frame(locmarkers)
      if (length(s) >= 2)
        iloc <- data.frame(locmarkers[s, ])
        out <- rdist.earth (iloc)
        out <- as.vector(as.dist(out))
        snew <- matrix(NA, length(out))
        for (n in 1:length(out)) {
          dist_value <- out[n]
          snew[n, ] <- matrix.index(dat2$swevgramdata.list[[datenum]], dist_value)[1]
        }
        snew <- as.vector(snew)
        iset.set(j)
        iset.select(what = snew)
    } else
      iobj.opt(visible = TRUE)
  }

  ## NOTE: To discontinue any brushing tool, click on "file" and then "break".
E.14 iSweBrushPlot

iSweBrushPlot <- function(i = 2, j = 1, datenum = 1,
    dat1 = swedata, dat2 = sweplotdata,
    num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66)){

    ## Example:
    ## iSweBrushMap(i = 1, j = 2, datenum = 1,
    ##    dat1 = swedata, dat2 = sweplotdata,
    ##    num = c(83, 4, 42, 53, 12, 26, 64, 29, 82, 66))

    matrix.index <- function(a, value) {
        idx <- which(data.frame(a) == value)
        col.num <- ceiling(idx/nrow(a))
        row.num <- idx - (col.num - 1) * nrow(a)
        return(c(row.num, col.num))
    }

    locmarkers <- dat1$LatLonData[num, ]
    colnames(locmarkers) <- c("Lon", "Lat")
    locmarkers <- data.frame(locmarkers)

    while (!is.null(ievent.wait())) {
        iset.set(i)
        if (iset.sel.changed()) {
            s <- iset.selected()
            if (length(s) >= 1){
                ivardata <- data.frame(dat2$swevgramdata.list[[datenum]][s, ])
                snew <- as.vector(as.matrix(ivardata[, c("loc1", "loc2")]))
                iset.set(j)
                # ilines(lowess(x = c(-114, -114.2), y = c(37, 37)), col = "#0000c0")
                # ilines(lowess(x = c(0, 0), y = c(0, 0)), col = "marked", visible = FALSE)
                iset.select(what = snew)
                # for(n in 1:(nrow(ivardata))){
                #     iobj.opt(x = lowess(c(ivardata[n, "loc1Lon"], ivardata[n, "loc2Lon"])),
                #             c(ivardata[n, "loc1Lat"], ivardata[n, "loc2Lat"])),
                #     visible = FALSE)
                # }
                else
                iobj.opt(visible = FALSE)
            } else
                iobj.opt(visible = FALSE)
        }
    }

    ## NOTE: To discontinue any brushing tool, click on "file" and then "break".
E.15 ReadSweAsciiData

ReadSweAsciiData <- function(filename = c("swe021093.asc", "swe030393.asc",
"swe032393.asc", "swe040893.asc",
"swe041593.asc", "swe042993.asc",
"swe051293.asc", "swe051993.asc",
"swe052593.asc"), elev, n.dates = 9){
## Example:
## sweasciidata <- ReadSweAsciiData(filename = c("swe021093.asc",
## "swe030393.asc", "swe032393.asc",
## "swe040893.asc", "swe041593.asc",
## "swe042993.asc", "swe051293.asc",
## "swe051993.asc", "swe052593.asc"),
## elev = "uscfoot.asc",
## n.dates = 9)

library(RgoogleMaps)
library(PBSmapping)
library(ReadImages) ## For jpg format
library(png) ## For png format
library(RSAGA)

filename <- filename
RCEWdata.list <- vector("list", n.dates)
for (i in 1:n.dates){
  RCEWdata.list[[i]] <- read.ascii.grid(filename[i])
}

Diffdata.list <- vector("list", (n.dates - 1))
for (i in 1:(n.dates - 1)){
  Diffdata.list[[i]] <- RCEWdata.list[[i+1]]$data - RCEWdata.list[[i]]$data
}

RCEWelevation <- read.ascii.grid(elev)
datenum = 1 # Fixed

xcoord <- RCEWdata.list[[datenum]]$header$xllcorner +
  (0:30)*RCEWdata.list[[datenum]]$header$cellsize
ycoord <- RCEWdata.list[[datenum]]$header$yllcorner +
  (0:14)*RCEWdata.list[[datenum]]$header$cellsize

#matrix(rev(ycoord))
#t(matrix(xcoord))

xborder <- c(522278.2, 522308.7, 522308.7, 522400.2, 522400.2, 522430.6,
  522430.6, 522461.1, 522461.1, 522522.1, 522522.1, 522765.9,
  522765.9, 522796.4, 522796.4, 522826.9, 522826.9, 523070.7,
  523070.7, 523131.7, 523131.7, 523101.2, 523101.2, 523070.7,
  523070.7, 523040.2, 523040.2, 522979.3, 522979.3, 522735.4,
  522735.4, 522705, 522705, 522613.5, 522613.5, 522522.1,
522522.1, 522430.6, 522430.6, 522369.7, 522369.7, 522308.7, 522308.7, 522278.2)
yborder <- c(4774311, 4774311, 4774281, 4774281, 4774220, 4774220, 
4774159, 4774159, 4774128, 4774128, 4774098, 4774098, 
4774128, 4774128, 4774159, 4774159, 4774189, 4774189, 
4774159, 4774159, 4774281, 4774281, 4774342, 4774342, 
4774372, 4774372, 4774433, 4774433, 4774464, 4774464, 
4774494, 4774494, 4774464, 4774464, 4774433, 4774433, 
4774464, 4774464, 4774494, 4774494, 4774464, 4774464, 
4774403, 4774403)
xout <- c(522552.6, 522583, 522583, 522552.6)
yout <- c(4774159, 4774159, 4774189, 4774189)

## Write Output

list(filename = filename, RCEWdata.list = RCEWdata.list, 
RCEWelevation = RCEWelevation, Diffdata.list = Diffdata.list, 
xcoord = xcoord, ycoord = ycoord, xborder = xborder, 
yborder = yborder, xout = xout, yout = yout)

E.16 SweAsciiImagePlot

SweAsciiImagePlot <- function(dat = sweasciidata, n.dates = 9, 
diffplot = FALSE, 
breaks = c(-0.5, 0, 0.5, 1.0, 1.5, 
2.0, 2.5, 3.0), 
zlim = c(-0.50, 3.0), 
legend = c("0","(0, 0.5]", "(0.5, 1.0]", 
"(1.0, 1.5]", "(1.5, 2.0]", 
"(2.0, 2.5]", "(2.5, 3.0]")){

## Example1: SweAsciiImagePlot(dat = sweasciidata, n.dates = 9, 
## diffplot = FALSE, 
## breaks = c(-0.5, 0, 0.5, 1.0, 1.5, 
## 2.0, 2.5, 3.0), 
## zlim = c(-0.50, 3.0), 
## legend = c("0","(0, 0.5]", "(0.5, 1.0]", 
## "(1.0, 1.5]", "(1.5, 2.0]", 
## "(2.0, 2.5]", "(2.5, 3.0]"))

## Example2: SweAsciiImagePlot(dat = sweasciidata, n.dates = 9, 
## diffplot = TRUE, 
## breaks = c(-1.2, -0.8, -0.4, -0.0005, 
## 0.0005, 0.4, 0.8, 1.2),


if (diffplot = FALSE) {
    layout (matrix(1:n.dates, 3, 3, byrow = TRUE))
    library("RColorBrewer")
    mycolor <- brewer.pal(7, "Greys")

    for (i in 1:n.dates){
        x <- dat$RCEWdata.list[[i]]$header$xllcenter +
            (0:30)*dat$RCEWdata.list[[i]]$header$cellsize
        y <- dat$RCEWdata.list[[i]]$header$yllcenter +
            (0:14)*dat$RCEWdata.list[[i]]$header$cellsize

        image(x, y, t(dat$RCEWdata.list[[i]]$data)[, nrow(dat$RCEWdata.list[[i]]$data):1],
            zlim = zlim, col = mycolor, breaks = breaks,
            main = paste(dat$timepoint[i], cex.main = 2)

        polygon(dat$xborder, dat$yborder, border = "black", lwd = 1)
        polygon(dat$xout, dat$yout, border = "black", col = "black",
            lwd = 1, density = 25, angle = -45)

        legend("bottomleft", legend = legend, fill = mycolor,
            bg = "white", cex = 0.9)
    }
}
else{
    if (diffplot = TRUE) {
        layout (matrix(1:n.dates, 3, 3, byrow = TRUE))
        library("RColorBrewer")
        mypalette <- brewer.pal(n = 7, "PRGn")

        for (i in 1:(n.dates - 1)){
            x <- dat$RCEWdata.list[[1]]$header$xllcenter +
                (0:30)*dat$RCEWdata.list[[1]]$header$cellsize
            y <- dat$RCEWdata.list[[1]]$header$yllcenter +
                (0:14)*dat$RCEWdata.list[[1]]$header$cellsize

            image(x, y, t(dat$Diffdata.list[[i]])[, nrow(dat$Diffdata.list[[i]]):1],
                zlim = zlim, col = mypalette, breaks = breaks,
                main = paste("1st-Difference", i), cex.main = 2)

            polygon(dat$xborder, dat$yborder, border = "black", lwd = 1)
            polygon(dat$xout, dat$yout, border = "black", col = "black",
                lwd = 1, density = 25, angle = -45)

            legend("bottomleft", legend = c("(-1.2, -0.8)", "(-0.8, -0.4)",
                "(-0.4, 0.0)", " 0.0 ", "(0.0, 0.4)", "(0.4, 0.8)",
                "(0.8, 1.2)")
        }
"(0.8, 1.2]"), fill = mypalette, bg = "white", cex = 0.8)

}
}
}
}

E.17 iSweAsciiPlot

iSweAsciiPlot <- function(dat = sweasciidata, pngmap = "uscw.png",
                          datenum = 2, x1 = 522300, y1 = 4774114,
                          x2 = 523130, y2 = 4774490){

  ## Example: iSwePlot(dat = sweasciidata, pngmap = "uscw.png", datenum = 2,
  ## x1 = 522245, y1 = 4774067, x2 = 523163, y2 = 4774496)
  ## or x1 = 522247.8, y1 = 4774067, x2 = 523162.2, y2 = 4774494

  swe.data <- dat$RCEWdata.list[[datenum]]$data
  swe_data <- matrix(t(swe.data))

  site.elevation <- dat$RCEWelevation$data
  site_elevation <- matrix(t(site.elevation))

  xval <- dat$RCEWdata.list[[datenum]]$header$xllcenter +
          (0:30)*dat$RCEWdata.list[[datenum]]$header$cellsize
  yval <- dat$RCEWdata.list[[datenum]]$header$yllcenter +
          (0:14)*dat$RCEWdata.list[[datenum]]$header$cellsize

  #matrix(rev(yval))
  #t(matrix(xval))

  xmat <- matrix(t(matrix(xval)), 15, 31, byrow = TRUE)
  xcoord <- matrix(t(xmat))

  ymat <- matrix(matrix(rev(yval)), 15, 31, byrow = FALSE)
  ycoord <- matrix(t(ymat))

  plotdata <- cbind(xcoord, ycoord, swe_data, site_elevation)
  colnames(plotdata) <- c("x", "y", "SWE", "Elevation")
  plotdata2 <- data.frame(plotdata[!is.na(plotdata[, "SWE")], ])
  colnames(plotdata2) <- c("x", "y", "SWE", "Elevation")

  dates <- c("February 10", "March 03", "March 23", "April 08",
            "April 15", "April 29", "May 12", "May 19", "May 25")

  png(filename = "uscw.png", width = 700, height = 480,
       units = "px", pointsize = 12, bg = "white")

  library("RColorBrewer")
  mycolor <- brewer.pal(7, "Greys")
breaks <- c(-0.5, 0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0)
zlim <- c(-0.50, 3.0)

par(mar = c(0, 0, 0, 0))
image(xval, yval,
t(dat$RCEWdata.list[[datenum]]$data)[,
nrow(dat$RCEWdata.list[[datenum]]$data):1],
zlim = zlim, col = mycolor, breaks = breaks,
 xlab = " ", ylab = " ", axes = FALSE)
polygon(dat$xborder, dat$yborder, border = "black", lwd = 2)
polygon(dat$xout, dat$yout, border = "black", col = "black",
 lwd = 2, density = 25, angle = -45)
dev.off()
install.packages("iplots", "http://rforge.net", type = "source")

ibitmap <- function(pngmap, x1, y1, x2, y2){
  iplot(plotdata2[, "x"], plotdata2[, "y"],
     main = "Plot of SNOTEL Stations Over an Image-Plot",
     xname = "Longitude", yname = "Latitude")
  iplot.opt(xlim = c(x1, x2), ylim = c(y1, y2))
  iplot.opt(col = c(5, 5), ptDiam = 10)
  iraster(x1, y1, x2, y2, pngmap, layer = -2)
}
ibitmap(pngmap, x1, y1, x2, y2)

ibox(plotdata2[, "SWE"], ylab = "SWE (inches)",
     main = paste("Boxplot of SWE -", dates[datenum]))

ibox(plotdata2[, "Elevation"], main = "Boxplot for Elevations (in ft)")
}

## NOTE: For horizontal boxplot, click on "view" and then "rotate"
APPENDIX F

CO-AUTHOR AND REPRINT PERMISSION LETTERS
08 August 2014

James B. Odei  
Department of Mathematics and Statistics  
3900 Old Main Hill  
Logan, UT 84322-3900  
(435) 764-5783

Mevin B. Hooten  
Dept. of Fish, Wildlife & Conservation Biology  
Colorado State University  
1484 Campus Delivery  
Fort Collins  
CO 80523-1484

Dear Dr. Hooten,

I am in the process of preparing my dissertation in the Department of Mathematics and Statistics at Utah State University. I hope to complete my degree in August of 2014.

I am requesting your permission to include our co-authored paper as a chapter in my dissertation. You will be cited as a co-author on the title page. Please advise me of any changes you require.

Please indicate your approval of this request by signing in the space provided, attaching any other form or instruction necessary to confirm permission. If you have any questions, please call me at the number above.

Thank you for your cooperation.

James B. Odei

______________________________________________________________________

I Mevin B. Hooten hereby give permission to James B. Odei to reprint the following material in chapter 2 of his dissertation.


Signed ________________________________  
Date _________________________________
**JOHN WILEY AND SONS LICENSE**

This is a License Agreement between James B Odei (“You”) and John Wiley and Sons (“John Wiley and Sons”) provided by Copyright Clearance Center (“CCC”).

<table>
<thead>
<tr>
<th>License Number</th>
<th>3447211354819</th>
</tr>
</thead>
<tbody>
<tr>
<td>License date</td>
<td>Aug 13, 2014</td>
</tr>
<tr>
<td>Licensed content publisher</td>
<td>John Wiley and Sons</td>
</tr>
<tr>
<td>Licensed content publication</td>
<td>Environmetrics</td>
</tr>
<tr>
<td>Licensed content title</td>
<td>A Bayesian hierarchical model for forecasting intermountain snow dynamics</td>
</tr>
<tr>
<td>Licensed copyright line</td>
<td>Copyright © 2014 John Wiley &amp; Sons, Ltd.</td>
</tr>
<tr>
<td>Licensed content author</td>
<td>James B. Odei, Jürgen Symanzik, Mevin B. Hooten</td>
</tr>
<tr>
<td>Licensed content date</td>
<td>May 8, 2014</td>
</tr>
<tr>
<td>Start page</td>
<td>324</td>
</tr>
<tr>
<td>End page</td>
<td>340</td>
</tr>
<tr>
<td>Type of use</td>
<td>Dissertation/Thesis</td>
</tr>
<tr>
<td>Requestor type</td>
<td>Author of this Wiley article</td>
</tr>
<tr>
<td>Format</td>
<td>Print and electronic</td>
</tr>
<tr>
<td>Portion</td>
<td>Full article</td>
</tr>
<tr>
<td>Will you be translating?</td>
<td>No</td>
</tr>
</tbody>
</table>
CURRICULUM VITAE
James Beguah Odei
Curriculum Vitae
August, 2014

CONTACT
368 East Center Street, Apt. 2
Logan, UT 84321-4784
USA

Phone: (435) 764-5783
Email: jamesbeg@gmail.com

EDUCATION

Ph.D., Statistics
Utah State University (USU), Logan, UT, USA
Department of Mathematics & Statistics
Dissertation: “Statistical Modeling, Exploration and Visualization of
Snow Water Equivalent (SWE) Data”
Advisor: Professor Jürgen Symanzik

2007–2014

MS., Mathematical Sciences (Statistics Concentration)
University of Nevada Las Vegas (UNLV), Las Vegas, NV, USA
Department of Mathematics & Statistics
Thesis: “Diffusion Model For Lightning Radiative Transfer Using Spherical Geometry”
Advisor: Professor Dieudonné D. Phanord

2004–2007

BSc. (1st Class), Mathematics (Statistics Minor)
Kwame Nkrumah University of Sci. & Tech. (KNUST), Kumasi, Ghana
Department of Mathematics
Project Title: “Seasonal Variation In The Incidence Of Malaria Cases – A Case Study
At Mamprobi Polyclinic, Accra”
Advisor: Samuel Kwasi Appiah

1997–2001

Certification

Society of Actuaries: Actuarial Course P/1

MAY 2005

RESEARCH INTERESTS

Statistical Data Visualization
Bayesian Hierarchical Modeling

Environmental & Ecological Statistics
Spatial, Temporal, and Spatio–Temporal Statistics
Computer Skills

Experience and Proficiency in C, SAS, SPSS, R, and \LaTeX

Teaching and Working Experience

Graduate Student Teacher at Utah State University

STAT 5100, Linear Regression and Time Series (Summer 2014)
STAT 1040, Introduction to Statistics (Fall 2009, Spring 2010, Fall 2010, Spring 2011)
STAT 1040, Introduction to Statistics – Recitation (Fall 2011, Spring 2012, Spring 2013)
STAT 2300, Business Statistics (Summer 2009, Summer 2010, Summer 2011, Summer 2012, Summer 2013)
MATH 1010, Intermediate Algebra (Fall 2013, Spring 2014)
MATH 1050, College Algebra (Fall 2007)

Research Assistant at Utah State University

Graduate Student Teacher at University of Nevada, Las Vegas

MATH 120, College Algebra (Fall 2009, Spring 2010, Fall 2010, Spring 2011)
MATH 126 & 127, PreCalculus (Spring 2007, Summer 2007)
MATH 132, Finite Mathematics (Spring 2005, Summer 2005, Fall 2005)

Research Assistant at Kwame Nkrumah University of Science & Technology (KNUST)
Institute of Renewable and Natural Resources (IRNR), Ghana (Fall 2002 – Summer 2003)

Papers in Journals


Papers in Proceedings


---

^2 Featured on the BBC news web page in the big picture section 14 January 2010, and on the journal cover.
Professional Conferences and Workshops


Honors and Awards

*Robins Award for Graduate Teaching Assistant of the Year*  
Utah State University  
Spring 2014

*Graduate Student Teacher of the Year Award*  
College of Science, Utah State University  
Spring 2014

*Graduate Student Teacher of the Year Award*  
Department of Mathematics and Statistics, Utah State University  
Spring 2014

*Golden Key International Honour Society Member*  
Utah State University Chapter  
Fall 2012

*School of Graduate Studies Dissertation Fellowship Award*  
Utah State University  
Spring 2012

*Graduate Student Senate Enhancement Award*  
Utah State University  
Spring 2012

*Community Good Samaritan Award*  
American Red Cross (Northern Utah Chapter)  
Spring 2012

*Official Citation of Honor*  
Utah State Legislature  
Spring 2012
Official Citation of Honor
City of Logan-Utah Legislature

Fall 2011

Hero of the Year Award
People Magazine (November 7, 2011 Edition)

Innovative Teacher Award
Department of Mathematics and Statistics, Utah State University

Spring 2011

Best Graduate Teaching Assistant Award
University of Nevada, Las Vegas

Fall 2006

First Class Honor (BSc. Mathematics)
Department of Mathematics – KNUST

Spring 2002

Grants

School of Graduate Studies Travel Grant ($600)
Utah State University

2009 – 2010

College of Science Travel Award ($600)
Utah State University

2009 – 2010

Department of Mathematics and Statistics Travel Award ($1730)
Utah State University

2009 – 2010

Department of Mathematics and Statistics Research Grant ($4000)
Utah State University

Summer 2008

Professional Memberships

American Statistical Association (ASA)
Spring 2005 – Present

ASA Utah Chapter
Fall 2007 – Summer 2014

ASA Nevada Chapter
Spring 2005 – Summer 2007

Academic Service
1. Joshua J. Thoms, Department of Languages, Philosophy, and Speech Communication, Utah State University (Fall 2011). Assisted with quantitative analysis on a study about Comparison of the Differences In Writing and Speaking Gains Between Two Lower-Level Spanish Language Classes (Faculty Project)

2. Janette Smith Kudin, Department of Nutrition and Food Science, Utah State University (Fall 2011). Assisted with quantitative analysis on The Impact Of One-On-One Education on Breast Feeding Initiation Rates (Faculty/MS Thesis Project)

Media Interviews

ABC (The Ellen DeGeneres Show), People Magazine, CBS (The Early Show), 2011

CNN (Piers Morgan Tonight, Anderson Cooper 360°, and Good Morning America)

MSNBC (Caught On Camera) and the Associated Press – USA

Multi Media TV – Ghana

2011