Conservation Conflict: Rare Plant Conservation and Energy Development and Potential in the Colorado Plateau

Joshua Carrell
Utah State University

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

Part of the Environmental Sciences Commons

Recommended Citation
https://digitalcommons.usu.edu/etd/8545

This Thesis 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.
CONSERVATION CONFLICT: RARE PLANT CONSERVATION AND ENERGY
DEVELOPMENT AND POTENTIAL IN THE COLORADO PLATEAU

by

Joshua Carrell

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

MASTER OF SCIENCE

in

Ecology

Approved:

Thomas C. Edwards, Ph.D. Edd Hammill, Ph.D.
Major Professor Committee Member

Eric Thacker, Ph.D. D. Richard Cutler, Ph.D.
Committee Member Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2022
Conservation Conflict: Rare Plant Conservation and Energy Development and Potential in the Colorado Plateau

by

Joshua Carrell, Master of Science
Utah State University, 2022

Major Professor: Dr. Thomas C. Edwards
Department: Wildland Resources

The Colorado Plateau has abundant oil, gas, and alternative energy potential. This energy potential is scattered among a patchwork of land ownership, with private, tribal, and public lands being actively developed for energy extraction. Elements of biodiversity (e.g., listed and sensitive plant and animal species) are distributed among all land tenures, yet the laws protecting them can vary as a function of land tenure. Therefore, it is imperative to understand the spatial distributions of threatened, endangered, and sensitive species in relation to land tenure to preserve habitat and conserve species populations in areas undergoing energy development. I seek to explore the interactions and relationships among land ownerships, existing and potential energy extraction, and sensitive, threatened, and endangered plant species in the Colorado Plateau region of Western North America. To achieve this, I modeled the spatial distribution and habitat of selected threatened, endangered, and sensitive plant species using species distribution models (SDMs), which portray likely spatial locations of modelled species. SDMs were linked
with land tenure in the Colorado Plateau in a geographic information system (GIS) and potential energy extraction probability was overlaid with the SDMs. Last, spatially explicit optimization models were developed using Marxan, a spatial optimization tool, that depicts various land management strategies designed to minimize impacts on private lands and plants and maximize energy extraction. Results indicate that incorporating risk and land tenure in spatially optimized frameworks, it is possible to achieve solutions that suggest the long-term viability of rare plant species. The scenarios developed here represent a different attitude towards the value of rare plants and the risk of energy development. Comparing the results of this study gives insight into the financial consequences of plant species protection and quantify biodiversity costs of energy development across landscapes.

(119 pages)
Conservation Conflict: Rare Plant Conservation and Energy Development and Potential in the Colorado Plateau

Joshua Carrell

The Colorado Plateau provides numerous sources of energy, particularly natural oil and gas. This energy potential is being actively extracted for consumption on both public and private lands. Elements of biodiversity (e.g., listed and sensitive plant and animal species) are distributed among all land tenures, yet the laws protecting them can vary depending if they occur on public, private, or tribal land. Therefore, it is necessary to understand the geographic ranges of threatened, endangered, and sensitive species and their relationship to land ownership to preserve habitat and conserve species populations. I developed species distribution models (SDMs) to examine the predicted geographic range of 29 rare flowering plant species in the Colorado Plateau. In addition, I used a geographic information system (GIS) and conservation planning software, Marxan, to examine how financial cost, oil and gas development, oil and gas potential, and land ownership affect plant conservation and energy extraction targets. I found that predicted rare flowering plant habitat significantly overlaps oil and gas development and potential. In this study, it was found that proper conservation targets regarding these plants could be achieved on public and tribal lands and in locations that minimize impacts on oil and gas development and extraction.
ACKNOWLEDGMENTS

I cannot thank the following individuals enough. This accomplishment is ours together.

Tom Edwards. Thank you for taking a chance on me and being my mentor. It was truly a pleasure and I wish you luck in your adventures of retirement.

To my committee members, Eric Thacker and Edd Hammill. Thank you for teaching me conservation planning can be such an exciting and confusing topic of study. I appreciate your help throughout my time at USU.

To Thad Nicholls. You were the friend and support I needed.

To Doug Ramsey. LSB 225A was not just the GIS lab but a classroom for me. I learned more about maps from you outside of class than I ever did inside. Thank you.

To Shannon Belmont. You gave me every opportunity to learn and grow as a person and professional. You truly deserve the best going forward. Thank you for believing in me!

To Mom and Dad. Thank you for believing in me when I couldn’t. Your love has always been with me, and you always have mine.

To Potato and Hana. You’re both my best friends and the best dogs I could ask for. I only wish you both knew how much you have helped me.

Lastly to my wife, April. I hope you know I could not have done this without you. You are my joy and my adventure. I love you very much.

Josh Carrell
CONTENTS

Page

ABSTRACT ....................................................................................................................... iii
PUBLIC ABSTRACT ......................................................................................................... v
ACKNOWLEDGMENTS ................................................................................................. vi
LIST OF TABLES ........................................................................................................... viii
LIST OF FIGURES ........................................................................................................... ix
INTRODUCTION ............................................................................................................... 1
RESEARCH OBJECTIVES ........................................................................................... 5
METHODS ....................................................................................................................... 6
RESULTS ......................................................................................................................... 18
DISCUSSION ................................................................................................................... 27
CONCLUSION ................................................................................................................. 31
LITERATURE CITED ...................................................................................................... 32
APPENDICES ................................................................................................................... 38
  Species Distribution Modeling Protocol ................................................................. 39
  Marxan Analysis Protocol ....................................................................................... 40
  Species Distribution Modeling Code ................................................................... 41
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1. Scientific and common name, species code, and conservation status of modelled plant species</td>
<td>10</td>
</tr>
<tr>
<td>Table 2. Four Marxan scenarios representing a variety of objectives, targets, and risks</td>
<td>17</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>The Colorado Plateau and associated land tenures</td>
<td>7</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Spatial depictions of data in the Colorado Plateau</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Species distribution modeling metrics for the 29 rare flowering species of interest</td>
<td>19</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Species distribution modeling output density classifications derived from the number of overlapping spatial predictions of plants per grid cell</td>
<td>21</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Species distribution model spatial associations with oil and gas development and potential</td>
<td>22</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Marxan analysis results for financial cost distributions across target objective percentages</td>
<td>24</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Selected Marxan scenario spatially depicting the number of times a unit was selected for conservation action over 100 runs</td>
<td>25</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Best solution of scenario 4 with 30% target objective with geographic locations of interest</td>
<td>26</td>
</tr>
</tbody>
</table>
INTRODUCTION

Oil and gas exploration and extraction are recognized threats to global biodiversity and often negatively impact species conservation (Finer et al. 2008; Jones et al. 2015). Its necessary infrastructure has been documented to increase habitat fragmentation while reducing habitat quality and availability (Copeland et al. 2009; Brittingham et al. 2014; Harfoot et al. 2018). While vagile species may relocate from areas of high energy development, evidence of a decreased survivorship exists (Wilson et al. 2013; Garman 2018). However, plant species are static in their ability to avoid ecological impacts, which raises significant interest in the relationship between plant species conservation and oil and gas development. With the high number of plant species qualifying for listed status under the Endangered Species Act or are considered of concern by NGOs like NatureServe, and the increasing global demand for oil and gas resources (Copeland et al. 2009), we question how these conflicting conservation and energy demand objectives may be mutually satisfied. This study explores the spatial and economic relationships among plant species of concern, oil and gas development, land ownership, and the potential of concurrently meeting conservation objectives among conflicting values (i.e., continued energy exploration and development; species habitat conservation).

This study focuses on the Colorado Plateau, a biodiverse and energy-rich ecoregion located in the Intermountain West of the United States (Nabhan et al. 2002). Due to unique geologic and edaphic features, the Colorado Plateau boasts profuse oil and gas resources. Due to the same features, plant biodiversity hotspots occur but are severely limited in their distribution to small, discrete ranges (Comstock & Ehleringer 1992;
McCaffery et al. 2014; Baker et al. 2016). As a result, many species are placed under federal and state management plans due to their rarity. In addition to limited range size, plants undergo significant disturbance from anthropogenic activity that accompany oil and gas development. Two primary disturbances are unpaved road usage and dust dispersal. Unpaved road use has been linked to not only reducing available habitat for plant species, but creating barriers to dispersal, spreading exotic and harmful species, and increasing dust loads on plants that reduce growth and diminish reproductive potential (Pickering & Hill 2007; Lewis et al. 2017). In addition, previous studies have examined an abundance of coevolved plant and pollinator species in the Colorado Plateau (Althoff et al. 2001; Godsoe et al. 2009). Dust, noise, and turbulence that accompany oil and gas development affect the distribution and density of pollinators which result in indirect impacts to plant health (Grodsky et al. 2021; Phillips et al. 2021).

Societal interests regarding biodiversity conservation and natural resource extraction are diverse and often conflicting. Whereas the prospect of maximizing oil and gas development and extraction while simultaneously conserving the total landscape of rare species is desirable, the outcome is simply not possible. However, solutions abound in spatial prioritization and conservation planning. Spatial conservation prioritization and planning is the science of systematically deciding where to implement certain actions regarding biodiversity conservation (Watts et al. 2017). Here, we implement Marxan (Ball & Possingham 2000), the most widely used conservation planning software in the world, as a means of identifying spatial locations where conservation actions that best protect plants given a background of energy extinction are located. Marxan is a simulated annealing algorithm designed for solving complex conservation planning problems.
through systematically selecting area (often referred to as units or zones) for conservation action (Ball et al. 2009; Watts et al. 2009). This selection is derived from the defined proportion of a conservation target (i.e., conserving x% of a species habitat and range), a financial cost associated with the action for each unit, and a calculated probability of risk relating to the conservation action failing on a unit per unit basis (Studwell et al. 2021). It has been observed that conserving units that cover at least 30% of a species spatial distribution and habitat may sufficiently protect that species in the long-term (Betts 2009). Here, we seek to acquire the 30% threshold for a community of flowering plant species co-located in a region of high and ongoing energy development.

The interaction between humans and ecosystems is global, affecting not only singular species, but ecological communities. Therefore, exploration of these conflicts is not well served by single species approaches. To maximize our actions in plant conservation, we implement a community-based approach in this study. Developing robust conservation scenarios for a single species remains a major endeavor; however, conservation biology emphasis has shifted from species representation within communities to communities within their environment (Frankel et al. 1995). Community-level approaches to biological conservation are a major advance in most current single-species conservation and management practices (Olden 2003). Initiatives that establish species rarity as an indicator of conservation priority, a significant focus of this study, may be biased if they disregard important evolutionary and adaptive processes taking place in lower diversity communities (Scarano 2009). Studies suggest that reductions in biodiversity result in a decreased stability in ecosystem function and productivity (Frankel et al. 1995). Therefore, rare flowering plant species of the Colorado Plateau,
while differing in individual conservation categorization or listed status, are treated with the same conservation objectives in this study to maximize the ecological function of biodiversity.

We evaluated methods of mutual satisfaction in the Colorado Plateau using Species Distribution Modeling (SDM) and Spatial Conservation Prioritization (SCP). Using a variety of SDM algorithms, we constructed spatially explicit habitat suitability models for a community of 29 flowering plant species of concern. We also examined the distribution of current oil and gas infrastructure, oil and gas development potential, and land tenure associated with predicted plant habitat suitability. This framework also includes the financial cost tied to both well-head restoration and land ownership as a function of optimization. We developed 4 scenarios to identify spatially optimized locations with object functions that maximized both energy development and species conservation. Given the constraints of risk and economic cost, neither energy development nor plant conservation can achieve maximum goal satisfaction. Hence our approach simply optimizes conservation of 29 flowering plant species by selecting the minimum number of units to cover 30% of each species distribution and habitat at the lowest financial cost. By minimizing unit selection and cost, areas of high energy development and potential is maximized. This study provides not only general options for land managers but identifies specific spatial zones for conservation action in a highly complex network of land tenures, risks, and conflicting conservation values.
RESEARCH OBJECTIVES

This study examines the spatial relationship between rare plant conservation and oil and gas energy development and potential as well as mutually reaching conservation targets that differ in objective. To achieve these goals, I constructed species distribution models (SDMs) of 29 threatened, endangered, and sensitive plant species that are distributed throughout the Colorado Plateau. Developing these models constitutes the basis for assessing spatial management needs and allow me to quantify the relative importance of land ownership and distribution of threatened endangered, and sensitive plant species ranges. I then explored these relationships between plant distributions, ownership, and energy development and energy potential which defines a real-world current example of the management issues confronting extraction of energy resources, listed species, and public and tribal and private land ownership. After SDMs were developed and examined, I developed optimal spatial depictions that minimize conflict between energy extraction and listed species locations. This final objective seeks to highlight opportunities that emphasize protection on public and tribal lands while minimizing those required on private and tribal lands.
METHODS

Study area

The Colorado Plateau is a high elevation, cold desert located in western North America’s arid continental interior. The region has an area of approximately 182,000 km², spanning four degrees of latitude and four degrees of longitude from the Utah Wasatch mountains in the west (~111°W) to the Colorado Rockies in the east (~107°W), and from the Arizona/New Mexico Plateaus in the south (~36°N) to the Wyoming Basin in the North (~40°N). With a unique geologic history and rich diversity of soils, hot spots for plant species endemism and oil and gas potential occur throughout the Colorado Plateau (Welsh 1978; Stohlgren et al. 2005). The soils, however, are sensitive to disturbance and the patchwork of land tenure poses a challenge to comprehensive range-wide management of the unique biota across the Colorado Plateau. Land tenure consists of a patchwork of federal (63%), state (5%), tribal (14%), and private (17%) lands (Figure 1). Remaining land tenures, including lands owned by local governments and NGOs, account for <1%. 
Figure 1 The Colorado Plateau and associated land tenures.

Rare plants

Species of interest for conservation prioritization consisted of all flowering plant species within the Colorado Plateau boundaries that are categorized as G1-Critically Imperiled; G2-Imperiled; G3-Vulnerable; T1-Critically Imperiled Variety; and T2-Imperiled Variety under NatureServe (www.explorer.natureserve.org). Species of interest also included those listed under the U.S. Fish and Wildlife Service’s ESA as Threatened or Endangered (www.fws.gov/program/listing-and-classification). Meeting these criteria, 40 flowering plant species of concern occur in the Colorado Plateau. Eleven of these plant species were eliminated from this study due to limited data availability and quantity. The remaining 29 species are the focus of this study. These 29 species are
distributed across the spatial extent of the Colorado Plateau and are found primarily in the state of Utah (Figure 2A). Several species are synonymous with other taxonomic designations (Table 1). Under NatureServe categorizations, 10 species are G-1 Critically Imperiled, 10 species as G-2 Imperiled, 3 as G-3 Vulnerable, 2 T-1 Critically Imperiled Variety, 3 as T-2 Imperiled Variety, and 1 Not listed under any categorization. Under ESA listing status, 6 species as Endangered, 7 species as Threatened, 3 species under review, 1 delisted species, and 12 species not listed. It is important to note that species listed under ESA are not directly afforded protection by the ESA. Plant species occurrence spatially overlaps private and public and tribal land tenures, and the laws protecting them may vary as function of land tenure.
Figure 2 Spatial depictions of data in the Colorado Plateau. A) Geographic locations of flowering plant occurrences distributed across private and public land tenures; B) Oil and gas development potential probability model C) Geographic locations of oil and gas well pads; D) Spatial extent of major oil and gas basins of the Colorado Plateau. See Figure 1. for the spatial location of the Colorado Plateau in North America.
<table>
<thead>
<tr>
<th>Scientific Name</th>
<th>Common Name</th>
<th>Species Code&lt;sup&gt;a&lt;/sup&gt;</th>
<th>NS&lt;sup&gt;b&lt;/sup&gt;</th>
<th>ESA&lt;sup&gt;c&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Aliciella caespitosa</em> (A. Gray) J.M. Porter</td>
<td>Rabbit Valley gilia</td>
<td>ALCA23</td>
<td>G2</td>
<td>N</td>
</tr>
<tr>
<td>(=<em>Gilia caespitosa</em> A. Gray)</td>
<td></td>
<td></td>
<td></td>
<td>(=GICA8)</td>
</tr>
<tr>
<td><em>Astragalus equisolensis</em> Neese &amp; S.L. Welsh</td>
<td>Horseshoe milkvetch</td>
<td>ASEQ</td>
<td>T2</td>
<td>N</td>
</tr>
<tr>
<td>(=<em>Astragalus desperatus</em> M.E. Jones var. <em>neeseae</em> Barneby) (=Elizabeth’s Milkvetch)</td>
<td></td>
<td></td>
<td></td>
<td>(=ASDEN)</td>
</tr>
<tr>
<td><em>Astragalus hamiltonii</em> Ced. Porter</td>
<td>Hamilton’s milkvetch</td>
<td>ASHA3</td>
<td>G1</td>
<td>N</td>
</tr>
<tr>
<td>(=<em>Astragalus lonchocarpus</em> Torr. var. <em>hamiltonii</em> (Ced. Porter) Isely)</td>
<td></td>
<td></td>
<td></td>
<td>(=ASLOH)</td>
</tr>
<tr>
<td><em>Astragalus iselyi</em> S.L. Welsh</td>
<td>Isely’s milkvetch</td>
<td>ASIS</td>
<td>G1</td>
<td>U</td>
</tr>
<tr>
<td><em>Astragalus montii</em></td>
<td>Heliotrope milk-vetch</td>
<td>ASMO11</td>
<td>G3</td>
<td>T</td>
</tr>
<tr>
<td>(=<em>Astragalus limnocharis</em> Barneby var. <em>montii</em> (S.L. Welsh) Isely)</td>
<td></td>
<td></td>
<td></td>
<td>(=ASLIM)</td>
</tr>
<tr>
<td><em>Astragalus sabulosus</em> M.E. Jones var. <em>sabulosus</em></td>
<td>Cisco milkvetch</td>
<td>ASSAS</td>
<td>T2</td>
<td>U</td>
</tr>
<tr>
<td><em>Astragalus sabulosus</em> M.E. Jones var. <em>vehiculus</em></td>
<td>Cisco milkvetch</td>
<td>ASSAV</td>
<td>T1</td>
<td>U</td>
</tr>
<tr>
<td><em>Cryptantha barnebyi</em> I.M. Johnst.</td>
<td>Oilshale cryptantha</td>
<td>CRBA6</td>
<td>G2</td>
<td>N</td>
</tr>
<tr>
<td><em>Cryptantha grahamii</em> I.M. Johnst.</td>
<td>Fragrant Cryptantha</td>
<td>CRGR4</td>
<td>G2</td>
<td>N</td>
</tr>
<tr>
<td><em>Cycladenia humilis</em> Benth. var. <em>jonesii</em> (Eastw.) S.L. Welsh Jones’ waxydogbane</td>
<td></td>
<td>CYHUJ</td>
<td>T2</td>
<td>T</td>
</tr>
<tr>
<td>(=<em>Cycladenia jonesii</em> Eastw.)</td>
<td></td>
<td></td>
<td></td>
<td>(=CYJO2)</td>
</tr>
<tr>
<td><em>Erigeron maguirei</em> Cronquist</td>
<td>Maguire’s fleabane</td>
<td>ERMA8</td>
<td>G2</td>
<td>D</td>
</tr>
<tr>
<td>(=<em>Erigeron maguirei</em> Cronquist var. <em>harrisonii</em> S.L. Welsh)</td>
<td></td>
<td></td>
<td></td>
<td>(=ERMAH)</td>
</tr>
<tr>
<td><em>Eriogonum smithii</em></td>
<td>Flat-top buckwheat</td>
<td>ERSM</td>
<td>G1</td>
<td>N</td>
</tr>
</tbody>
</table>
- *Eriogonum corymbosum* Benth. var. *smithii* (Reveal) S.L. Welsh  
  (=ERCOS2)

- *Glaucocarpum suffrutescens* (Rollins) Rollins  
  Uinta Basin waxfruit  
  GLSU  
  G1  
  E

- *Schoenocrambe suffrutescens* (Rollins) S.L. Welsh & Chatterley  
  (=SCSU2)

- *Thelypodium suffrutescens* Rollins  
  (=THSU4)

- *Hesperidanthus suffrutescens* (Rollins) Al-Shehbaz  
  (=Toad-flax cress)

- *Lepidium barnebyanum* Reveal  
  Barneby’s Pepperweed  
  LEBA  
  G1  
  E

- *Oreoxis trotteri* S.L. Welsh & Goodrich  
  Trotter’s alpineparsley  
  ORTR  
  G1  
  N

- *Cymopterus trotteri* (S.L. Welsh & Goodrich) Cronquist  
  (=CYTR13)

- *Pediocactus despainii* S.L. Welsh & Goodrich  
  Despán’s Pincushion cactus  
  PEDE17  
  G2  
  E

- *Pediocactus winkleri* K.D. Heil  
  Winkler’s Pincushion cactus  
  PEW12  
  G2  
  T

- *Penstemon albifluvis* England  
  White River beardtongue  
  PEAL80  
  T1  
  N

- *Penstemon scariosus* Pennell var. *albifluvis* (England) N.H. Holmgren  
  (=PESCA)

- *Penstemon flowersii* Neese & S.L. Welsh  
  Flowers’ beardtongue  
  PEFL8  
  G1  
  N

- *Penstemon gibbensii* Dorn  
  Gibbens’ beardtongue  
  PEGI4  
  G1  
  N

- *Penstemon grahamii* D.D. Keck  
  Uinta Basin beardtongue  
  PEGR6  
  G2  
  N

- *Phemeranthus thompsonii* (N.D. Atwood & S.L. Welsh)  
  Cedar Mountain fameflower  
  PTHH6  
  N  
  N

- *Talinum thompsonii* N.D. Atwood & S.L. Welsh  
  (=TATH)
<table>
<thead>
<tr>
<th>Species</th>
<th>Common Name</th>
<th>Species Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Physaria tumulosa</strong> (=Lesquerella rubicundula Rollins)</td>
<td>Kodachrome bladderpod (=Tum bladderpod)</td>
<td>G3 E</td>
</tr>
<tr>
<td>(=Lesquerella hitchcockii Munz ssp. rubicundula (Rollins) Maguire &amp; A.H. Holmgren) (=LEHIR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(=Lesquerella tumulosa (Barneby) Reveal)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Schoenocrambe argillacea (S.L. Welsh &amp; N.D. Atwood)</strong></td>
<td>Uinta Basin plainsmustard SCAR5</td>
<td>G1 T</td>
</tr>
<tr>
<td>(=Thelypodiopsis argillacea S.L. Welsh &amp; N.D. Atwood)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sclerocactus brevispinus K.D. Heil &amp; J.M. Porter</strong></td>
<td>Pariette cactus SCBR12</td>
<td>G1 E</td>
</tr>
<tr>
<td>(=Sclerocactus whipplei (Engelm. &amp; J.M. Bigelow) Britton &amp; Rose var. ilseae)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sclerocactus wetlandicus Hochstätter</strong></td>
<td>Uinta Basin Hookless cactus SCWE</td>
<td>G3 T</td>
</tr>
<tr>
<td><strong>Sclerocactus wrightiae L.D. Benson</strong></td>
<td>Wright Fishhook cactus SCWR</td>
<td>G2 E</td>
</tr>
<tr>
<td><strong>Spiranthes diluvialis</strong></td>
<td>Ute Ladies’-tresses SPD16</td>
<td>G2 T</td>
</tr>
<tr>
<td>(=Spiranthes romanzoffiana Cham. var. diluvialis (Sheviak) S.L. Welsh) (=SPROD)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Townsendia aprica S.L. Welsh &amp; Reveal</strong></td>
<td>Last Chance townsendia TOAP</td>
<td>G2 T</td>
</tr>
<tr>
<td>(=Townsendia jonesii (Beaman) Reveal var. lutea S.L. Welsh)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Species 4 letter code for each species from USDA NRCS Plants Database.
Species status categorization from NatureServe (G1 = Critically Imperiled; G2 = Imperiled; G3 = Vulnerable; T1 = Critically Imperiled Variety; T2 = Imperiled Variety).
Species listing status from Endangered Species Act (Not Listed = N; Under Review = U; Threatened = T, Endangered = E; Delisted = D)
Oil and gas

The Intermountain West of the United States doubled oil and gas development between 1990 and 2007 (Naugle et al. 2010). With global oil demand projected to increase by 50 percent between 2007 and 2030 (Copeland et al. 2009). Within the boundaries of the Colorado Plateau, energy rich landscapes are scattered among a patchwork of land ownership and are being actively developed for extraction on both public and private lands. The Colorado Plateau currently has an approximate 63,000 active and 26,000 abandoned oil and gas well pads (USGS 2019) which are distributed throughout the entirety of the Colorado Plateau (Figure 3C). Oil and gas wellheads, with an approximate lifespan of 30 years, have an estimated restoration cost that averages $24,000 USD (Statistica 2020).

With the reasonably short life cycle of oil and gas wellheads, regions like the Colorado Plateau are likely to have increased interactions between human activity and biodiversity through increased development. A model of oil and gas development potential (Copeland et al. 2009) is used in this study to analyze the factor of risk in conservation planning (Figure 2B). This model is overlaid the planning units of this study and represents the probability a unit may fail under conservation action.

Species distribution modeling

Species distribution models (SDMs) of the 29-plant species constitute the spatial information by which the baseline conservation targets are defined. Five SDM classifiers were implemented to develop species-specific SDMs for the 29 plant species of interest. These models include: 1) a generalized linear model (GLM) with a logit link (logistic model) (Nelder & Wedderburn 1972); 2) maximum entropy (Phillips et al. 2006; Elith et
al. 2011); 3) random forests (Breiman 2001; Cutler et al. 2007); 4) boosted regression trees (Friedman 2001; De'ath 2007; Elith et al. 2008); and 5) Bayesian additive regression trees (Chipman et al. 2010; Tan & Roy 2019). These five selected classifiers represent a range of statistical algorithms that are parametric and non-parametric in scope, and that are commonly applied to studies involving SDM.

Models were fit with species-specific predictor variables, reduced in number from a common pool. Because all data were collected from targeted sampling approaches (Schreuder et al. 2001) without underlying designs, selection of thresholds for model classification is problematic. Consequently, we implemented max kappa as the classification threshold (Freeman & Moisen 2008) for each species-specific model. Model accuracy was assessed using the area under the curve (AUC) (Hanley & McNeil 1982) and the true skill statistic (TSS) (Allouche et al. 2006). Individual SDMs were evaluated within an ensemble models context (Araújo & New 2007; Grenouillet et al. 2011; Ramirez-Reyes et al. 2021). Ensemble models bolster defensibility of SDM applications as no single SDM is considered “correct”. Procedurally, we stacked the five individual SDMs for each plant species and calculated a mean average, standard deviation (SD), and upper and lower range of likelihood for each modeled pixel. These outputs were converted to spatially explicit products representing the ensemble model prediction (mean) and variability (1 SD, range) at a 1-km² spatial resolution.

The 29 species distribution ensemble models were then cropped to the Colorado Plateau’s spatial extent. Each distribution model was overlayed to a larger grid of 1km² cells that cover the entire range of the 29 species. Grid cells were then assigned a value that correlates with the number of species that have predicted distribution occupying that
cell. The highest number of species distributions occupying a single cell was 10 while the lowest was 1. These values were then categorized in five classes: I) 1-2 species; II) 3-4 species; III) 5-6 species; IV) 7-8 species; and V) 9-10 species (Figure 3A). Oil and gas development potential, a probability model producing an output of values that range from 0 to 100, was overlaid on the 5 classes of species per cell to determine the probability of development risk among species density classifications. Current well pads were also overlaid to determine amount in each species density classification.

**Spatial conservation optimization**

Our objective in applying the Marxan algorithm is to optimize the selection of planning units for the 29 species of interest. The underlying goals are to: 1) identify the least number of units suitable for flowering plant species conservation, 2) do so at the lowest financial cost possible. The number of units selected is derived by the specified conservation target percentage, or, how much of each species distribution we would like to conserve. As mentioned, a 30% target is optimal in the long-term conservation of species (Betts 2009; Hammill et al. 2016; Studwell et al. 2021), although targets 10%-90% were applied to analyze other possibilities. The quantity of selected units for conservation action directly impacts current oil and gas activity, the potential oil and gas development, and the financial cost. The spatial extent of the units possible for selection covered the entirety of the 29 SDMs and consisted of 55,538 1-km2 cells. The cost to either maintain public lands with no oil and gas infrastructure presence or to restore lands with currently occupied oil and gas infrastructure was the baseline deciding factor in unit selection.
The financial cost designated to each cell was dependent upon the oil and gas infrastructure presence. Cells that contained no oil and gas well presence were assigned a value of $1,460, which is the value associated with public land maintenance. While specific information on oil and gas production would be ideal, information on well pad development and production is largely considered proprietary by oil and gas business entities and specific information is consequently unavailable. Thus, we decided to evaluate the financial costs based on well-head restoration costs as a surrogate. Restoration drives the analysis of this study as restoring current and deprecated oil and gas infrastructure would provide the land needed for species habitat. Cells containing oil and gas well pads were assigned the value of $24,500 multiplied to the number of well pads occupying that cell. Analysis for each conservation objective (10% to 90%) was run 100 times with 10,000 iterations each. Analysis for conservation objective at 100% was not included as the values are fixed at upper limits containing all units or units on public lands only.

To better understand the consequences of prioritizing energy development and the conservation of rare plants, analyses were run under a set of different scenarios. Each scenario represents a different management perspective towards the value of rare plants and the risk of energy development. Comparing these results provides insight into the financial consequences of plant species protection and quantifies biodiversity costs of energy development across landscapes. To inform future land management decisions regarding flowering plant conservation in the Colorado Plateau, we developed 4 scenarios depicting a variety of conservation objectives and limiting real-world factors (Table 2). These factors include financial cost, oil and gas development, oil and gas potential
development, and land tenure (i.e., public and private lands). For this study, land tenure is
categorized by either public or private and tribal lands. Public lands are owned by federal
and state government agencies while private lands include ownership from private, local
government, non-governmental organizations, and tribal nations.

Tribal lands are technically categorized as federal lands held in trust by the federal government. However, there is a complex relationship between the two entities when it (Tubb 2018). The Endangered Species Act of 1973 (ESA), for example, differs in its application on tribal land versus non-tribal federal lands (Marren 2007). Secretarial Order #3206 (SO-3206) recognizes that, “Indian lands are not subject to the same controls as federal public lands” and that “tribes are appropriate governmental entities to manage their lands and tribal trust resources, and, as trustees, support tribal measures that preclude the need for conservation restrictions” (States 1995). For this reason, private and tribal lands, although technically separate in their dominion and classification often function similarly when it comes to endangered species management. Hence, we considered them operationally identical in our optimizations.

**Table 2** Four Marxan scenarios representing a variety of objectives, targets, and risks.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Target %</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>10% - 90%</td>
<td>All planning units: No risk included</td>
</tr>
<tr>
<td>S2</td>
<td>10% - 90%</td>
<td>All planning units: Risk included</td>
</tr>
<tr>
<td>S3</td>
<td>10% - 90%</td>
<td>Public land units only: No risk included</td>
</tr>
<tr>
<td>S4</td>
<td>10% - 90%</td>
<td>Public land units only: Risk included</td>
</tr>
</tbody>
</table>
RESULTS

Species distribution modeling

Twenty-nine flowering species distribution ensemble models produced values for four model metrics: The Area Under Curve of the Receiver Operating Characteristic (AUC), Sensitivity, Specificity, and the True Skill Statistic (TSS) (Figure 3). These metrics represent important values in measuring the performance, accuracy, uncertainty of each species-specific model and are commonly analyzed in studies incorporating SDM. AUC values range from 0 to 1 and the higher the AUC value, the better predicting performance of that model. AUC values of the 29 plants ranged from 0.737 to 0.997 with a mean average of 0.931, which suggests a high average modeling prediction performance. Sensitivity and specificity represent the model’s ability to correctly predict suitable habitat and unsuitable habitat, respectively (Parikh et al. 2008). Model sensitivity values ranged from 0.741 to 0.996 with an average value of 0.932 which specificity values ranged from 0.584 to 0.982 with a mean average of 0.843. TSS values, derived from Sensitivity+Specificity-1, ranged from 0.383 to 0.974 with a mean average of 0.781. Model metrics varied across all SDMs with values that suggest high predictive capability.
Figure 3 Species distribution modeling metrics for the 29 rare flowering species of interest.

Higher species predicted suitability overlap occurred in the northern region of the Colorado Plateau (Figure 4). The spatial extent of density classifications generally became smaller as classification value increased from 1 to 10 species per unit cell, meaning there are fewer cells with a higher density of predicted species suitability. Classification values were distributed on both public and private and tribal lands (Figure 5A) and overwhelmingly located on public lands. There was a slight increase of the spatial distribution of plants on public lands as density classification values increase. Values were also distributed throughout two major sedimentary basins known for oil and gas development, the Uinta-Piceance basin and the Paradox basin (Figure 5B). Cells representing 5 to 10 species all occurred within the Uinta-Piceance basin or outside of
either basin. Cells of all other values subsequently occurred in similar locations as well as throughout the Paradox basin. Cells containing 7-10 species overlapped with the highest density of energy development potential while cells with 9-10 species contained all probability values over 50%. Cells with 1-6 species had a greater range of energy potential probability values with a higher density of under 25% (Figure 5C). Oil and gas well pads were within the spatial extent of each density classification value with the highest amount in cells of 3-4 species (Figure 5D). The number of well pads decreases as density classification values increase. These results suggest that there is significant overlap in the density of suitable species habitat and the overall occurrences of oil and gas development and potential. While SDM allows us to visually examine predicted suitable habitat of the 29 species of interest, the use of Marxan in this study will allow us to systematically optimize locations for conservation, which inversely optimize locations for oil and gas development.
Figure 4 Species distribution modeling output density classifications derived from the number of overlapping spatial predictions of plants per grid cell.
Figure 5 Species distribution model spatial associations with oil and gas development and potential. A) Land tenure values across public and private and tribal lands per species density classifications; B) Species density classifications distributed across major sedimentary basins; C) Oil and gas development probability values distributed across species density classifications; D) The number of oil and gas well pads occupying the spatial extent of each species density classification.

Spatial Conservation Prioritization

Four SCP scenarios (from here on referred to as S1, S2, S3, and S4; refer to Table 2 for descriptions) produced specific units for selection at a wide range of financial cost (Figure 6). Target objectives ranged from 10% to 90% of each plant’s predicted spatial...
distribution. 100% of each species distribution was not included for scenarios of both public and private and tribal land tenures as financial cost is fixed regardless of analysis output as species distributions occupy 100% of available cells for both scenarios. The number of selected units per conservation objective ranged from 2,831 units to 42,814 units, always increasing as conservation objective increases. As a higher conservation objective is desired, financial cost increases exponentially with ranges across all scenarios with values from $4.1 million USD the lowest to $394.4 million USD at the highest. All scenarios have approximately the same financial cost for conservation objectives that range 10% to 30%. S3 and S4 financial costs are highest through 40-90% objective targets. S1 and S2 calculated possibilities of unit selection from all 55,538 units, both on public and private and tribal land and cost less per objective targets 20-90% than S3 and S4. No species were missing from S1 and S2 throughout the entire range of conservation objectives. S3 and S4 cost approximately the same amounts per objective target of all percentages. These scenarios both calculated missing species (that the conservation objective could not be met given risk and available units) in objective targets 60-90% with a missing species in S4 at a conservation objective of 40%. S3 and S4 resulted in high financial cost and missing species due to selecting only public units and calculating risk of the unit failing from oil and gas development potential in S4.
Figure 6 Marxan analysis results for financial cost distributions across target objective percentages. Blue color depicts scenario runs that have reach conservation target objectives for all species; Tan color depicts scenario runs in which at least 1 species target objective percentage is not obtainable due to risk, unit availability, and financial cost.

A 30% target objective (Betts 2009) was selected as the best outcome for each scenario. At 30% each scenario produces approximately the same financial cost for those on all units and public lands only (Figure 6). S4 at 30% represents potentially the best scenario to select for plant conservation as it meets objective targets with no missing species, has a moderate price compared with other scenarios, and accounts for oil and gas development potential in its unit selections (Figure 7). S4 has strongly selected units of
where 7-10 species occupy a single unit, often in wild and rough terrain that is unoccupied with oil and gas development (Figure 8A). A majority units with either high oil and gas potential or current oil and gas infrastructure have been clearly excluded from selection (Figure 8B). Due to the expansive range of both the plant species and oil and gas development, selecting locations that entirely exclude oil and gas development is not possible for a 30% target. Units have been selected for conservation action that overlap or are spatially neighboring current well pads and infrastructure (Figure 8C).

Figure 7 Selected Marxan scenario spatially depicting the number of times a unit was selected for conservation action over 100 runs.
Figure 8 Best solution of scenario 4 with a 30% target objective with geographic locations of interest. A) Satellite Imagery of an area of high unit selection depicting rough and wild landscapes; B) Satellite imagery of an area of no unit selection depicting oil and gas infrastructure; C) Satellite imagery of an area with high unit selection and oil and gas well pads.
We began by explicitly acknowledging that no solution meets 100% of both plant species conservation and energy extraction objectives. We achieved a solution (Scenario 4 at 30%) through prioritization that identified a set of spatially explicit outcomes. This solution included 9,255 1-km² units that covered 30% of each plant species predicted distribution while accounting for oil and gas development risk, while selecting units solely on public lands. Our scenario focus was on public lands only given that ESA-listed plant species are not afforded protection private and tribal lands (Section 9, ESA) except where State law is in effect. This public lands solution achieves a sufficient target of 30% that has been suggested to provide significant value in species conservation over time (Betts 2009). While the 30% conservation objective was the highest percentage of objectives before scenarios of missing species, there may be no need to increase objective percentage for conservation. Conserving 30% of a species distribution has become a general rule of thumb in the long-term conservation of terrestrial species and applied in several studies (Betts 2009; Hammill et al. 2016; Studwell et al. 2021).

Land tenure and oil and gas development potential play a significant role in shaping the selection of units, which was only examined in 1 of the 4 scenarios (S4). While perhaps being more cost effective, S1-S3 solutions do not account for real-world limiting factors that shape real-world decision by combining land tenure and development potential risk. For biodiversity conservation planning in energy rich landscapes to be effective, it must explicitly examine human factors such as land tenure and future development probability to reduce conflict between nature and society (Luck et al. 2004; Karimi & Hockings 2018). It has been observed that complications regarding
conservation efforts are often more prevalent on private and tribal lands than public lands (Paloniemi & Tikka 2008). Although options exist that provide incentives for private and tribal landowners to cooperate with species conservation policy (e.g., conservation easements, conservation banking) (Rissman et al. 2007; Parker & Thurman 2019), they too often result in inefficient species conservation on private lands (Polasky & Doremus 1998). In addition to the discussion on public and private lands, the Endangered Species Act of 1973, Section 7 requires federal agencies to consult with the U.S. Fish and Wildlife Service (FWS) or National Oceanic and Atmospheric Administration (NOAA) Fisheries to ensure that any proposed activity on public lands will not adversely modify or destroy critical habitat of a listed species (United 1983). By contrast, private lands (although under the same legal obligations of the ESA Section 7) may compromise habitat through incidental take permits that allow for land modification under a personal habitat conservation plan submitted by the property landowner (Service 2013). Our scenario selection mitigates any potential conservation mishaps by removing unit selection on private lands all together. However, solely excluding private lands from conservation frameworks, although easier and ideal in theory, is not a panacea when confronting global conservation. In order to have effective biodiversity conservation in the current state of land ownership, efforts must be made to have public and private land entities work alongside each other. Otherwise, financial cost will increase and the ability to fully conserve species habitat (i.e., conserving 100% of species habitat) will be unattainable. By include public and private and tribal land tenures in spatially optimized plans will allow us to honestly examine the needs and restrictions of effective conservation.
That is not to say that conservation planning frameworks that incorporate financial cost, risk, and real-world limiting factor (i.e., land tenure) will ever produce a truly “perfect” scenario (Langford et al. 2009; Popov et al. 2022). Spatial conservation prioritization provides opportunities for us to maximize outcomes among conflicting objectives to create a “best” scenario to our knowledge and modeling parameters, but we lack the ability to fully represent reality when examining the interactions among nature and society. As this uncertainty occurs in modeling and conservation planning in general, other limiting factors were found that may affect the outcomes of spatially optimized scenarios for rare plant species. A lack of data availability and sharing among agencies and political entities may restrict the number of species to examine defensibly, lack of data access was the primary reason 11 species were excluded from this study.

Studies that examine ecoregional study areas (i.e., the Colorado Plateau) often overlap societal and political boundaries. Ecoregional studies are intuitive to conservation planning as similar species of status and taxa occupy transcend boundaries (Jepson & Whittaker 2002). However, political governance often inhibits conservation efficacy in transboundary regions (Schoon 2013). Rare species, like the 29 plants examined here, often have restrictions on their data availability and public sharing designated by state, regional, or federal organizations, which may negatively impact research that spatially covers political trans boundaries (Aitken et al. 2008; Reichman et al. 2011; Roche et al. 2015). In addition to lawful restriction, there is surveyed reasoning as to why ecologists limit data sharing that ranges from insufficient data management, concerns for data misinterpretation, and jealousy (Michener 2015).
Biodiversity occurrence databases and citizen science data collection allows for species occurrences to be publicly shared and examined, which may fill data gaps when data sharing neglect among agencies occurs (Matheson 2014; Robertson et al. 2014). Citizen science facilitates mass data collection that closes spatial gaps on global biodiversity occurrence data in seasons rather than years (Freeman et al. 2007; Amano et al. 2016). While success in citizen science is apparent, issues in widespread taxonomic misidentification have occurred with plant species as the rate of increase in global natural history collections outpaces the ability to process, evaluate and name species correctly (Goodwin et al. 2015). Likewise, there is an observed taxonomic bias and societal preference in citizen science data collection, which excludes species endemic to a single region (Troudet et al. 2017) like the Colorado Plateau. In addition, species like our 29 rare plant species often have morphologically similar counterparts and may be very difficult to discriminate without electron microscopy or a determination based on geography (Barkworth & Committee 2003). The need for data availability and sharing among agencies of transboundary locations is apparent for the overall conservation of the worlds rare plant species.
CONCLUSION

In Summary, we have outlined a framework that could sufficiently conserve rare plant species while simultaneously maximizing oil and gas development and accounting for energy development potential risk. We also have identified issues regarding data availability and sharing that may arise in the global conservation of rare species, particularly those that are spatially distributed across transboundary regions. Future studies that adhere to these considerations address real-world challenges in the conservation planning of biodiversity in energy-rich landscapes.
LITERATURE CITED


Betts MGV, M.-A 2009. Landscape thresholds in species occurrence as quantitative targets in forest management: generality in space and time? Setting conservation targets for managed forest landscapes. Cambridge University Press.


Service USCR. 2013. The Endangered Species Act (ESA) and Claims of Property Rights 'Takings'


APPENDICES
1.1 Species Distribution Modeling Protocol

The process for the development of the species distribution models in this study is found below. Specifics on coding structure and coding script is found in appendix 1.3.

1. Species occurrence presence and absence data and expert opinion bounding boxes were gathered from the Utah Department of Natural Resources.
2. Species occurrence and absence data were organized into formatting structure appropriate with modeling within RStudio using the R programming language.
3. Expert opinion bounding boxes were used at the initial modeling domain and a 10km buffer was created around each bounding box. Species without a provided expert opinion bounding box had a buffered 10km buffer around each point provided as the modeling domain.
4. Statistical classifiers were implemented, and spatially explicit outputs were developed for each classifier’s probability outcome, classified threshold, and binned probability (0-20%, 21-40%, 41-60%, 61-80%, 81-100%).
5. Species probability model outputs were then implemented in Marxan analysis. Specifics found in appendix 1.2.
1.2 Marxan Analysis Protocol

Marxan version 2.43 was implemented for the spatial optimization section of this study. Specific on formatting structure and Marxan software-specifics are provided below.

1. Each modeling domain for the 29 species of interest were overlaid and merge into a single polygon within ArcGIS Pro 2.1 software using the Merge and Dissolve functions.

2. Within the spatial boundaries of the single polygon, a 1km grid was developed using ArcGIS Pro 2.1 software using the Make Grid function. This grid constitutes the planning units used in the Marxan Analysis.

3. General parameters for Marxan version 2.43 are as follows:
   a. VERISON 0.1
   b. BLM 25
   c. PROP .9
   d. RANDSEED -1
   e. NUMREPS 100

4. Annealing parameters for Marxan version 2.43 are as follows:
   a. NUMITNS 10000000
   b. STARTTEMP -1
   c. NUMTEMP 10000
   d. COOLFAC 0.00000000000000E+0000

5. Cost thresholds for Marxan version 2.43 are as follows:
   a. COSTTHRESH 0.00000000000000E+0000
   b. THRESHPEN1 1.40000000000000E+0001
   c. THRESHPEN2 1.00000000000000E+0000

6. Input files were labeled as follows:
   a. INPUTDIR input
   b. PUNAME pu_updated.dat
   c. SPECNAME spec.dat
   d. PUVSPRNAME puvsp2.dat
1.3 Species Distribution Modeling Code

The code needed to develop species distribution models tailored to the 29 flowering species of interest are found below.

*Pathways beginning in "F:/Josh" would need to be changed to the directory in which you are working and/or have stored the species data.

```r
# Libraries
library(dplyr)
library(raster)
library(sf)
library(dismo)
library(plyr)
library(ggplot2)
library(terra)
library(fasterize)
library(rgdal)

# EPSG
# ALBERS NORTH AMERICA NAD83
# https://epsg.org/crs_5070/NAD83-Conus-Albers.html?sessionkey=1bl7864cyv
prj.aeaN83 <- "+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=23 +lon_0=-96 +x_0=0 +y_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs"
epsg.aeaN83 <- 5070

# Load UT/COP Frames
path.ut_cop <- "F:/Josh/gis/ut_cop_rasters"
setwd(path.ut_cop)
load("F:/Josh/DATA/UT/COP_fnet.RData", verbose = T) # COP
fishnet
load("ut_copRT.RData", verbose = T)
```

```r
ut_copRT <= as(ut_copR, "SpatRaster")
terra::plot(ut_copRT)
```
# path to folder of species shapefiles
path.species <- "F:/Export2021_06_07"
setwd(path.species)
list.files()

# Load species raw data
load("spp.RData") # Change to species
spec.raw <- spp
dim(spec.raw) # occurrences
head(spec.raw, 2) # check data
names(spec.raw) # names
table(spec.raw$Com_Name); table(spec.raw$Sci_Name) # check name counts,
#should be 1 per column

# Change CRS to epsg 5070
spec.raw <- st_transform(spec.raw, 5070)
st_crs(spec.raw) # correct
spp.pres <- spec.raw # name change

# clean up dataset
spp.presabs <- spp.pres[c(5)]

# Create a set of raw coordinates
spp.coords <- st_coordinates(spec.raw)

# create simply set
spp.presPA <- cbind(spp.presabs, spp.coords)
spp.presPA$PA <- 1
spp.presPA <- spp.presPA[c(2:3, 5, 4)]
spp.presPA <- spp.presPA[c(2:5)]

# DF
spp.presPA.DF <- st_drop_geometry(spp.presPA)

# Save files in presence folder
path.pres <- "F:/Josh/Plants/spp/presence"
save(spp.pres, spp.coords, spp.presPA, spp.presPA.DF,
     file = "F:/Josh/Plants/spp/presence/spp_pres.RData")

# Check saved file
load("F:/Josh/Plants/spp/presence/spp_pres.RData", verbose = T)

# END#

# before absences added
dim(spp.pres) # total data
dim(spp.coords) # coords
dim(spp.presPA) # coords and PA column and geometry
dim(spp.presPA.DF) # coords and PA column

# Create Species BBox 4

# load pres
load("F:/Josh/Plants/spp/presence/spp_pres.RData", verbose = T)

# Buffer points and union
spp.bufptSF <- st_buffer(spp.pres, dist = 10000)
spp.bufptSF <- st_union(spp.bufptSF)

# examine
plot(spp.bufptSF, add = T)

# create sf buffer of species bbox at 10km
spec.bboxbuf.SF <- st_buffer(spp.bufptSF, 10000)

# examine
plot(spec.bboxbuf.SF, add = T)

# create bbox raster
spec.bbox.RT <- terra::crop(ut_copRT, spec.bboxbuf.SF)

# Check with plots
#plot(ut_copRT)
terra::plot(spec.bbox.RT, col = "red") # RT species bbox in RED
plot(spec.bboxbuf.SF, col = "blue", add = T) # 10km Buffered bbox
plot(spp.bufptSF, col = "green", add = T) # species SF bbox in GREEN

# pseudo absences
sfbox <- terra::as.polygons(spec.bbox.RT) # raster extent to polygon sfc
sfbox <- st_as_sf(sfbox) # to sf
test <- sf::st_sample(sfbox, 100, type = "random") # create n random points
plot(test, add = T) # plot points
absence <- st_as_sf(test) # point to sf points
absence$PA <- 0 # add absence column
absence <- st_coordinates(absence)

# change names
spp.bbox.RT <- spec.bbox.RT
spp.bbox.SF <- spp.bufptSF
spp.bboxbuf.SF <- spec.bboxbuf.SF

# Save species bbox
save(spp.bboxbuf.SF, spp.bbox.SF, 
   file = "F:/Josh/Plants/spp/frames/bbox/spp.bbox.RDATA")
terra::writeRaster(spp.bbox.RT, 
   file = "F:/Josh/Plants/spp/frames/bbox/spp.bbox.RT.tif")
# view saved data
load("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RDATA", verbose = T)

spp.bbox.SF <- st_sf(spp.bbox.SF) %>% st_cast()
st_write(spp.bbox.SF, "F:/Josh/Plants/spp", driver = "ESRI Shapefile")

#END#

#################################################################
################ Crop Pres and add Abs
#########################################

# add absence
spp.abs <- st_sample(st_geometry(spec.bboxbuf.SF), type = "random", 
   size = (nrow(spp.pres)*.5)) %>%
   st_sf() %>%
   st_transform(epsg.aeaN83)
plot(spp.abs$geometry, add = T, col = "yellow") # plot abs in blue inside buffer
plot(spp.pres$geometry, add = T, col = "red") # pres on top in red

head(spp.pres); names(spp.pres)
dim(spp.pres) # presence points
dim(spp.abs) # new absence points

spp.coords.p <- st_coordinates(spp.pres) %>% as.data.frame();
dim(spp.coords.p) # pres coordinates
spp.coords.a <- st_coordinates(spp.abs) %>% as.data.frame();
dim(spp.coords.a) # absence coords
spp.coords.p$PA <- 1
spp.coords.a$PA <- 0

spp.pres <- rbind.data.frame(spp.coords.p, spp.coords.a)
dim(spp.pres); table(spp.pres$PA) # new presence absence dataframe

spp.presabs <- st_as_sf(spp.pres, coords = c("X", "Y"), 
   crs = prj.aeaN83, remove = F) # new presence absence
# Create a set of raw coordinates
spp.coords <- st_coordinates(spp.presabs) # new presence absence coords

# DF
spp.presabs.DF <- st_drop_geometry(spp.presabs) # new presence absence

# Save files in presence folder
path.pres <- "F:/Josh/Plants/spp/presence"
save(spp.pres, spp.coords, spp.presabs.DF, spp.presabs,
    file = "F:/Josh/Plants/spp/presence/spp_pres.RData")

# Check saved file
load("F:/Josh/Plants/spp/presence/spp_pres.RData", verbose = T)

#END#

#############################################################
########## FISHNET
#####################################################

# COP fishnet path
path.ut_cop <- "F:/Josh/gis/ut_cop_rasters"
setwd(path.ut_cop)
load("F:/Josh/DATA/Frames/COP_fnet.RData", verbose = T)

# view sf
head(COP_fnetSF, 2)

# crop to species bbox
load("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RDATA", verbose = T)
spec.bbox.fnet <- st_crop(COP_fnetSF, spp.bbox.SF)

# check with plots
plot(COP_fnetSF$geometry)
plot(spec.bbox.fnet$geometry, add = T, col = "red")

# change names
spp.bbox.fnet <- spec.bbox.fnet

# save species fnetid SF
save(spp.bbox.fnet,
    file = "F:/Josh/Plants/spp/frames/fnet/spp.bbox.fnet.RData")

# view saved file
load("F:/Josh/Plants/spp/frames/fnet/spp.bbox.fnet.RData",
    verbose = T)

#END#
# load presence data
load("F:/Josh/Plants/spp/presence/spp_pres.RData", verbose = T)
tru.xy <- spp.coords

# ut/cop raster
path.ut_cop <- "F:/Josh/gis/ut_cop_rasters"
setwd(path.ut_cop)
load("F:/Josh/DATA/Frames/COP_fnet.RData", verbose = T) # COP fishnet
load("ut_copRT.RData", verbose = T)
ut_copRT <- as(ut_copR, "SpatRaster")
plot(ut_copRT)

# process
m1 <- as.matrix(tru.xy) # creates tru.xy as matrix
ex2 <- terra::cellFromXY(ut_copRT, m1) %>% as.data.frame()
spec.pres.fnet <- cbind(tru.xy, ex2) # combine true pres x,y,and FNETIDs

# combine with pres abs
spec.pres.fnet <- cbind(spec.pres.fnet, spp.presabs.DF$PA)

# change names
names(spec.pres.fnet) <- c("X", "Y", "FNETID", "PA")
# Reorder
spec.pres.fnet <- spec.pres.fnet[c(3,1,2, 4)]

# save as species
spp.pres.fnetDF <- spec.pres.fnet

# check
head(spp.pres.fnetDF)

# make the PA pres fnet back into SF
spp.pres.fnetSF <- cbind(spp.presabs, spp.presabs.DF)
spp.pres.fnetSF <- spp.pres.fnetSF[c(1,2, 3, 7)]

# save
save(spp.pres.fnetDF, spp.pres.fnetSF,
     file = "F:/Josh/Plants/spp/presence/spp.pres.fnet.RData")

#END#
Load Predictor Variables

spp.bbox.RT <- terra::rast("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RT.tif")
load("F:/Josh/Plants/spp/presence/spp_pres.RData", verbose = T)

# Load predictors
preds.dom <-
  terra::rast("F:/Josh/DATA/Predictor_layers/All_Stack/
    preds.dom.final.tif")
preds.dom <- terra::crop(preds.dom, spp.bbox.RT) # crop preds
plot(preds.dom)

# Load presence points with FNETIDS
load("F:/Josh/Plants/spp/presence/spp.pres.fnet.RData", verbose = T)

# Load species presence
load("F:/Josh/Plants/spp/presence/spp.pres.fnet.RData", verbose = T)

# extract predictors to species presence
#spp.pres.fnetSF <- sf::st_crop(spp.pres.fnetSF, spp.bbox.SF)
#spp.presPA.SF <- sf::st_crop(spp.presabs, spp.bbox.SF)
spec.pres.sv <- as(spp.pres.fnetSF, "SpatVector") # presence must be spatvector
spec.pres.extract1 <- terra::extract(preds.dom, spec.pres.sv)

spec.trained.data <- cbind(spec.pres.extract1, spp.presabs) # combine true pres
head(spec.trained.data) # check

# organize
names(spec.trained.data)

# reorder
spec.trained.data <- spec.trained.data[\(c(110:112, 2:109)\)]

# rename
spp.trained.data <- spec.trained.data[\(c(3:111)\)]
names(spp.trained.data)

# save trained data
save(spp.trained.data,
  file = "F:/Josh/Plants/spp/trained_data/spp.trained.data.RData")

# save this for maxent model
spp.x.y.pa <- spp.presabs[1:3]
spp.x.y.pa <- st_drop_geometry(spp.x.y.pa)
save(spp.x.y.pa, file = "F:/Josh/Plants/spp/trained_data/spp.x,y.pa.RData")
dim(spp.x.y.pa)

### Exploring Trained Data

# This section will be different for each species. The goal is to
# narrow down our 83 predictors to a set of 10-20 with at least 1
# predictor layer from each grouping. The groupings are as
# follows:
# Weather (at least 1 from each subset)
# Topo
# Soils

library(modeest)
load("F:/Josh/Plants/spp/trained_data/spp.trained.data.RData",
verbose = T)

# Only need to run once prior to code below

# panel.cor function to determine correlations among predictors
panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
{ usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y, use = "pairwise.complete.obs"))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex <- 0.8/strwidth(txt)
  test <- cor.test(x,y)
  # borrowed from printCoefmat
  Signif <- symnum(test$p.value, corr = FALSE, na = FALSE,
  cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
  symbols = c("***", "**", "+", ".", " "))
  text(0.5, 0.5, txt, cex = cex * r)
  text(.8, .8, Signif, cex=cex, col=2)
}

### START function variable importance
varimp.glm <- function(tr.spp, tr.var, pres, pf, pl) {
  tmp.mat <- matrix(ncol = 2, nrow = (pl - pf + 1))
  for (i in pf:pl) {
    # option: linear+quadratic; linear only
    tmp <- glm(tr.spp[, pres] ~ tr.var[, i] + I((tr.var[, i])^2),
      na.action = na.omit,
      family = binomial)
    # linear only glm
    tmp.mat[(i - pf + 1), 1] <- tmp$aic
  }
}

### START function variable importance
tmp.mat[(i - pf + 1), 2] <- (1 - (tmp$deviance/tmp$null.deviance))
}
return(tmp.mat)
}

########################################################################
############### Weather
########################################################################

########################################################################
############## Precipitation
########################################################################

# Precipitation descriptive stats

# mean by PA
aggregate(spp.trained.data[, 2:18], list(presabs =
    spp.trained.data$PA),
    FUN = mean, na.rm = T)
# median by PA
aggregate(spp.trained.data[, 2:18], list(presabs =
    spp.trained.data$PA),
    FUN = median, na.rm = T)
# mode by PA; requires package modeest
aggregate(spp.trained.data[, 2:18], list(presabs =
    spp.trained.data$PA),
    FUN = mfv, na_rm = T)

# Precipitation graphs

windows(22,12)
par(mfrow = c(1, 12))
boxplot(spp.trained.data$prec_01 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "Prec_01", ylim = c(10,40))
boxplot(spp.trained.data$prec_02 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "Prec_02", ylim = c(10,40))
boxplot(spp.trained.data$prec_03 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "Prec_03", ylim = c(10,40))
boxplot(spp.trained.data$prec_04 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "Prec_04", ylim = c(10,40))
boxplot(spp.trained.data$prec_05 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "Prec_05", ylim = c(10,40))
boxplot(spp.trained.data$prec_06 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "Prec_06", ylim = c(10,40))
boxplot(spp.trained.data$prec_07 ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_07", ylim = c(10,40))
boxplot(spp.trained.data$prec_08 ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_08", ylim = c(10,40))
boxplot(spp.trained.data$prec_09 ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_09", ylim = c(10,40))
boxplot(spp.trained.data$prec_10 ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_10", ylim = c(10,40))
boxplot(spp.trained.data$prec_11 ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_11", ylim = c(10,40))
boxplot(spp.trained.data$prec_12 ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_12", ylim = c(10,40))

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/precipitation/monthly/precip_monthly_boxplots.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/precipitation/monthly/precip_monthly_boxplots.jpeg", type = "jpeg")

dev.off()

# prec seasons
#########################################################
windows(22,12)
par(mfrow = c(1, 5))
boxplot(spp.trained.data$prec_winter ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_winter", ylim = c(10,40))
boxplot(spp.trained.data$prec_spring ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_spring", ylim = c(10,40))
boxplot(spp.trained.data$prec_summer ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_summer", ylim = c(10,40))
boxplot(spp.trained.data$prec_fall ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_fall", ylim = c(10,40))
boxplot(spp.trained.data$prec_year ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "Prec_year", ylim = c(10,40))

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/precipitation/seasonal/precip_seasonal_boxplots.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/precipitation/seasonal/precip_seasonal_boxplots.jpeg", type = "jpeg")
Correlation among precip monthly predictors

# numeric correlations

```r
cut.point <- 0.7 # set cutpoint for correlation
c1 <- cor(spp.trained.data[, c(2:13)], use = "pairwise.complete.obs",
    method = "spearman") # est. correlation
c1 # examine
c2 <- subset(c1 > cut.point | c1 < -cut.point) # matrix of cor>cutpoint
c2 # examine; FALSE indicates cor<cutpoint
```

# plot correlations using modified panel.cor function

```r
windows(22,12)
pairs(spp.trained.data[, c(2:13)], lower.panel = panel.smooth,
    upper.panel = panel.cor,
    main = "Precipitation: Monthly Variable Correlation")
```

```r
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
    precipitation/monthly/precip_monthly_correlation.pdf",
    type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
    precipitation/monthly/precip_monthly_correlation.jpeg",
    type = "jpeg")
```

```r
dev.off()
```

# estimate VIP values => AIC & Adj deviance

```r
tr.vip <- spp.trained.data[, c(1, 2:13)] # keep only P/A & predictors
pres <- 1 # column for presence:absence
v.start <- 2 # column start predictor variables
v.stop <- ncol(tr.vip) # last column predictor variables
v.num <- v.stop - 1 # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop) # call VIP function
dev.fit # output matrix; col=1 AIC, col=2 Adj deviance
```

# built basic barplot if desired

```r
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10 # max of y-axis
ylim.r <- range(0, d.max) # range y-axis
x.labs <- names(tr.vip[2:v.stop]) # x-axis labels
windows(22,12)
par(mfrow = c(1,1))
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r,
    main = "Precipitation: Monthly Variable Importance",
    xlab = x.labs)
```
ylab = "adj.D2", names = x.labs)  # barplot
abline(h = 0)  # add horizontal line
abline(mean(dev.fit[, 2]), 0, lty = 3)  # ref lines; dash=mean
adj.dev

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
precipitation/monthly/precip_monthly_vips.pdf", type =
"pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
precipitation/monthly/precip_monthly_vips.jpeg", type =
"jpeg")

dev.off()

############################### Correlation among precip seasonal predictors
###############################
# numeric correlations

cut.point <- 0.7 # set cutpoint for correlation
c1 <- cor(spp.trained.data[, c(14:18)], use =
"pairwise.complete.obs",
method = "spearman") # est. correlation
c1 # examine

c2 <- subset(c1 > cut.point | c1 < -cut.point) # matrix of
cor>cutpoint
c2 # examine; FALSE indicates cor<cutpoint

# plot correlations using modified panel.cor function
windows(22,12)
pairs(spp.trained.data[, c(14:18)], lower.panel = panel.smooth,
upper.panel = panel.cor,
main = "Precipitation: Seasonal Variable Correlation")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
precipitation/seasonal/precip_seasons_correlations.pdf",
type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
precipitation/seasonal/precip_seasons_correlations.jpeg",
type = "jpeg")

dev.off()

# estimate VIP values => AIC & Adj deviance
tr.vip <- spp.trained.data[, c(1, 14:18)]  # keep only P/A &
predictors
pres <- 1 # column for presence:absence
v.start <- 2 # column start predictor variables
v.stop <- ncol(tr.vip) # last column predictor variables
v.num <- v.stop - 1 # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop) #
call VIP function
dev.fit # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10 # max of y-axis
ylim.r <- range(0, d.max) # range y-axis
x.labs <- names(tr.vip[2:v.stop]) # x-axis labels
windows(22,12)
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r,
main = "Precipitation: Seasonal Variable Importance",
ylab = "adj.D2", names = x.labs) # barplot
abline(h = 0) # add horizontal line
abline(mean(dev.fit[, 2]), 0, lt = 3) # ref lines; dash=mean

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
precipitation/seasonal/precip_seasons_vips.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
precipitation/seasonal/precip_seasons_vips.jpeg", type = "jpeg")

dev.off()

################ Precipitation Variables Selection
#################
# prec_01
# prec_summer
spp_vars <- c("prec_01", "prec_summer")

########################################################################
# Solar Radiation
########################################################################

########################################################################
# Monthly
########################################################################

# srad descriptive stats

# mean by PA
aggregate(spp.trained.data[, 19:30], list(presabs = spp.trained.data$PA),
FUN = mean, na.rm = T)
# median by PA
aggregate(spp.trained.data[, 19:30], list(presabs = spp.trained.data$PA),
FUN = median, na.rm = T)
# mode by PA; requires package modeest
aggregate(spp.trained.data[, 19:30], list(presabs = spp.trained.data$PA),
    FUN = mfv, na.rm = T)

# srad graphs

windows(22,12)
par(mfrow = c(1, 12))
boxplot(spp.trained.data$srad_01 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_01")
boxplot(spp.trained.data$srad_02 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_02")
boxplot(spp.trained.data$srad_03 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_03")
boxplot(spp.trained.data$srad_04 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_04")
boxplot(spp.trained.data$srad_05 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_05")
boxplot(spp.trained.data$srad_06 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_06")
boxplot(spp.trained.data$srad_07 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_07")
boxplot(spp.trained.data$srad_08 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_08")
boxplot(spp.trained.data$srad_09 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_09")
boxplot(spp.trained.data$srad_10 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_10")
boxplot(spp.trained.data$srad_11 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_11")
boxplot(spp.trained.data$srad_12 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "srad_12")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
    solar_radiation/monthly/srad_monthly_boxplots.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
    solar_radiation/monthly/srad_monthly_boxplots.jpeg",
    type = "jpeg")
dev.off()
# Correlation among srad monthly predictors

`cut.point <- 0.7` # set cutpoint for correlation

`c1 <- cor(spp.trained.data[, c(19:30)], use = "pairwise.complete.obs", method = "spearman")` # est. correlation

`c1` # examine

`c2 <- subset(c1 > cut.point | c1 < -cut.point)` # matrix of cor>cutpoint

`c2` # examine; FALSE indicates cor<cutpoint

# plot correlations using modified panel.cor function

`windows(22,12)`

`pairs(spp.trained.data[, c(19:30)], lower.panel = panel.smooth, upper.panel = panel.cor, main = "Solar Radiation: Monthly Variable Correlation")`

`savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/solar_radiation/monthly/srad_months_correlation.pdf", type = "pdf")`

`savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/solar_radiation/monthly/srad_months_correlation.jpeg", type = "jpeg")`

`dev.off()`

# estimate VIP values => AIC & Adj deviance

`tr.vip <- spp.trained.data[, c(1, 19:30)]` # keep only P/A & predictors

`pres <- 1` # column for presence:absence

`v.start <- 2` # column start predictor variables

`v.stop <- ncol(tr.vip)` # last column predictor variables

`v.num <- v.stop - 1` # number predictor variables

`dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop)` # call VIP function

`dev.fit` # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired

`d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10` # max of y-axis

`ylim.r <- range(0, d.max)` # range y-axis

`x.labs <- names(tr.vip[2:v.stop])` # x-axis labels

`windows(22,12)`

`par(mfrow=c(1,1))`

`barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r, main = "Solar Radiation: Monthly Variable Importance", ylab = "adj.D2", names = x.labs)` # barplot

`abline(h = 0)` # add horizontal line

`abline(mean(dev.fit[, 2]), 0, lt = 3)` # ref lines; dash=mean adj.dev
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
solar_radiation/monthly/srad_months_vips.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
solar_radiation/monthly/srad_months_vips.jpeg", type = "jpeg")

device.off()

#########################################################################
##### seasonal
##########################################################
names(spp.trained.data)
# srad seasons
##########################################################
windows(22,12)

par(mfrow = c(1, 5))
boxplot(spp.trained.data$srad_winter ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "srad_winter")
boxplot(spp.trained.data$srad_spring ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "srad_spring")
boxplot(spp.trained.data$srad_summer ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "srad_summer")
boxplot(spp.trained.data$srad_fall ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "srad_fall")
boxplot(spp.trained.data$srad_year ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "srad_year")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
solar_radiation/seasonal/srad_seasons_boxplots.pdf",
type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
solar_radiation/seasonal/srad_seasons_boxplots.jpeg",
type = "jpeg")

device.off()

######################## Correlation among srad monthly predictors
########################
# numeric correlations
cut.point <- 0.7 # set cutpoint for correlation
c1 <- cor(spp.trained.data[, c(31:35)], use = "pairwise.complete.obs",
       method = "spearman") # est. correlation
c1 # examine
c2 <- subset(c1 > cut.point | c1 < -cut.point)  # matrix of cor>cutpoint
   c2  # examine; FALSE indicates cor<cutpoint

# plot correlations using modified panel.cor function
windows(22,12)
pairs(spp.trained.data[, c(31:35)], 
      lower.panel = panel.smooth, 
      upper.panel = panel.cor, 
      main = "Solar Radiation: Seasonal Variable Correlation")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
            solar_radiation/seasonal/srad_seasons_correlation.pdf",
         type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
            solar_radiation/seasonal/srad_seasons_correlation.jpeg",
         type = "jpeg")

dev.off()

# estimate VIP values => AIC & Adj deviance
tr.vip <- spp.trained.data[, c(1, 31:35)]  # keep only P/A & predictors
pres <- 1  # column for presence:absence
v.start <- 2  # column start predictor variables
v.stop <- ncol(tr.vip)  # last column predictor variables
v.num <- v.stop - 1  # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop)  # call VIP function
dev.fit  # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10  # max of y-axis
ylim.r <- range(0, d.max)  # range y-axis
x.labs <- names(tr.vip[2:v.stop])  # x-axis labels
windows(22,12)
par(mfrow = c(1,1))
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r, 
         main = "Solar Radiation: Seasonal Variable Importance",
         ylab = "adj.D2", names = x.labs)  # barplot
abline(h = 0)  # add horizontal line
abline(mean(dev.fit[, 2]), 0, lt = 3)  # ref lines; dash=mean adj.dev

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
            solar_radiation/seasonal/srad_seasons_vips.pdf",
         type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
            solar_radiation/seasonal/srad_seasons_vips.jpeg",
         type = "jpeg")
dev.off()

################ Solar Radiation Variable Selection
###########################
# srad_01
# srad_summer

spp_vars <- c(spp_vars, "srad_01", "srad_summer")

#################################################################
################ Temperature Average
#################################################################

#########################################################################################
###### Monthly
#########################################################################################

# srad descriptive stats
names(spp.trained.data)

# mean by PA
aggregate(spp.trained.data[, 36:47], list(presabs = spp.trained.data$PA),
    FUN = mean, na.rm = T)
# median by PA
aggregate(spp.trained.data[, 36:47], list(presabs = spp.trained.data$PA),
    FUN = median, na.rm = T)
# mode by PA; requires package modeest
aggregate(spp.trained.data[, 36:47], list(presabs = spp.trained.data$PA),
    FUN = mfv, na.rm = T)

# tavg graphs
windows(22,12)
par(mfrow = c(1, 12))
boxplot(spp.trained.data$tavg_01 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "tavg_01")
boxplot(spp.trained.data$tavg_02 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "tavg_02")
boxplot(spp.trained.data$tavg_03 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "tavg_03")
boxplot(spp.trained.data$tavg_04 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "tavg_04")
boxplot(spp.trained.data$tavg_05 ~ spp.trained.data$PA,
    xlab = "Presence:Absence",
    ylab = "values", main = "tavg_05")
ylab = "values", main = "tavg_05")
boxplot(spp.trained.data$tavg_06 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tavg_06")
boxplot(spp.trained.data$tavg_07 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tavg_07")
boxplot(spp.trained.data$tavg_08 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tavg_08")
boxplot(spp.trained.data$tavg_09 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tavg_09")
boxplot(spp.trained.data$tavg_10 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tavg_10")
boxplot(spp.trained.data$tavg_11 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tavg_11")
boxplot(spp.trained.data$tavg_12 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tavg_12")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
  weather/temp_avg/monthly/tavg_months_boxplots.pdf", type
  = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
  weather/temp_avg/monthly/tavg_months_boxplots.jpeg",
  type = "jpeg")
dev.off()

############# Correlation among srad monthly predictors

# numeric correlations

cut.point <- 0.7 # set cutpoint for correlation
c1 <- cor(spp.trained.data[, c(36:47)], use =
  "pairwise.complete.obs",
  method = "spearman") # est. correlation
c1 # examine
c2 <- subset(c1 > cut.point | c1 < -cut.point) # matrix of cor>cutpoint
c2 # examine; FALSE indicates cor<cutpoint

# plot correlations using modified panel.cor function
windows(22,12)
pairs(spp.trained.data[, c(36:47)], lower.panel = panel.smooth,
  upper.panel = panel.cor,
  main = "Average Temperature: Monthly Variable Correlation")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
  weather/temp_avg/monthly/tavg_months_correlation.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/temp_avg/monthly/tavg_months_correlation.jpeg", type = "jpeg")

dev.off()

# estimate VIP values => AIC & Adj deviance
tr.vip <- spp.trained.data[, c(1, 36:47)] # keep only P/A & predictors
pres <- 1 # column for presence:absence
v.start <- 2 # column start predictor variables
v.stop <- ncol(tr.vip) # last column predictor variables
v.num <- v.stop - 1 # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop) # call VIP function
dev.fit # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10 # max of y-axis
ylim.r <- range(0, d.max) # range y-axis
x.labs <- names(tr.vip[2:v.stop]) # x-axis labels
windows(22,12)
par(mfrow=c(1,1))
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r,
     main = "Average Temperature: Monthly Variable Importance",
     ylab = "adj.D2", names = x.labs) # barplot
abline(h = 0) # add horizontal line
abline(mean(dev.fit[, 2]), 0, lt = 3) # ref lines; dash=mean adj.dev

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/temp_avg/monthly/temp_avg_months_vips.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/temp_avg/monthly/tavg_avg_months_vips.jpeg", type = "jpeg")
dev.off()

#################################################################
####### seasonal
#######################################################
names(spp.trained.data)
# tavg seasons
windows(22,12)
par(mfrow = c(1, 5))
boxplot(spp.trained.data$tavg_winter ~ spp.trained.data$PA,
     xlab = "Presence:Absence",
     ylab = "values", main = "tavg_winter")
boxplot(spp.trained.data$tavg_spring ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "tavg_spring")
boxplot(spp.trained.data$tavg_summer ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "tavg_summer")
boxplot(spp.trained.data$tavg_fall ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "tavg_fall")
boxplot(spp.trained.data$tavg_year ~ spp.trained.data$PA, xlab = "Presence:Absence", ylab = "values", main = "tavg_year")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/temp_avg/seasonal/tavg_seasons_boxplots.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/temp_avg/seasonal/tavg_seasons_boxplots.jpeg", type = "jpeg")
dev.off()

### Correlation among tavg monthly predictors

# numeric correlations
cut.point <- 0.7 # set cutpoint for correlation
c1 <- cor(spp.trained.data[, c(48:52)], use = "pairwise.complete.obs", method = "spearman") # est. correlation
c1 # examine
c2 <- subset(c1 > cut.point | c1 < -cut.point) # matrix of cor>cutpoint
c2 # examine; FALSE indicates cor<cutpoint

# plot correlations using modified panel.cor function
windows(22,12)
pairs(spp.trained.data[, c(48:52)], lower.panel = panel.smooth, upper.panel = panel.cor,
      main = "Average Temperature: Seasonal Variable Correlation")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/temp_avg/seasonal/tavg_seasonal_correlations.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/temp_avg/seasonal/tavg_seasonal_correlations.jpeg", type = "jpeg")

dev.off()

# estimate VIP values => AIC & Adj deviance
tr.vip <- spp.trained.data[, , c(1, 48:52)] # keep only P/A & predictors
pres <- 1 # column for presence:absence
v.start <- 2 # column start predictor variables
v.stop <- ncol(tr.vip) # last column predictor variables
v.num <- v.stop - 1 # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop) # call VIP function
dev.fit # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10 # max of y-axis
ylim.r <- range(0, d.max) # range y-axis
x.labs <- names(tr.vip[2:v.stop]) # x-axis labels
windows(22,12)
par(mfrow = c(1,1))
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r,
main = "Average Temperature: Seasonal Variable Importance",
ylab = "adj.D2", names = x.labs) # barplot
abline(h = 0) # add horizontal line
abline(mean(dev.fit[, 2]), 0, lt = 3) # ref lines; dash=mean
adj.dev

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/temp_avg/seasonal/temp_avg_seasonal_vips.pdf",
type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/temp_avg/seasonal/temp_avg_seasonal_vips.jpeg",
type = "jpeg")
dev.off()

########## temperature average Variable Selection

"tavg_04"
"tavg_summer"

spp_vars <- c(spp_vars, "tavg_04", "tavg_summer")

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

### Temperature Maximum

Monthly
# tmax descriptive stats

```r
names(spp.trained.data)

# mean by PA
aggregate(spp.trained.data[, 53:64], list(presabs = spp.trained.data$PA),
  FUN = mean, na.rm = T)

# median by PA
aggregate(spp.trained.data[, 53:64], list(presabs = spp.trained.data$PA),
  FUN = median, na.rm = T)

# mode by PA; requires package modeest
aggregate(spp.trained.data[, 53:64], list(presabs = spp.trained.data$PA),
  FUN = mfv, na_rm = T)
```

# tmax graphs

```r
windows(22,12)
par(mfrow = c(1, 12))
boxplot(spp.trained.data$tmax_01 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_01")
boxplot(spp.trained.data$tmax_02 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_02")
boxplot(spp.trained.data$tmax_03 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_03")
boxplot(spp.trained.data$tmax_03 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_04")
boxplot(spp.trained.data$tmax_05 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_05")
boxplot(spp.trained.data$tmax_06 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_06")
boxplot(spp.trained.data$tmax_07 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_07")
boxplot(spp.trained.data$tmax_08 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_08")
boxplot(spp.trained.data$tmax_09 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_09")
boxplot(spp.trained.data$tmax_10 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_10")
boxplot(spp.trained.data$tmax_11 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "tmax_11")
boxplot(spp.trained.data$tmax_12 ~ spp.trained.data$PA,
```
```r
xlab = "Presence:Absence",
ylab = "values", main = "tmax_12"
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
weather/temp_max/monthly/tmax_months_boxplots.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
weather/temp_max/monthly/tmax_months_boxplots.jpeg",
type = "jpeg")
dev.off()

############# Correlation among tmax monthly predictors
#############
# numeric correlations
cut.point <- 0.7 # set cutpoint for correlation
c1 <- cor(spp.trained.data[, c(53:64)], use = "pairwise.complete.obs",
method = "spearman") # est. correlation
c1 # examine
c2 <- subset(c1 > cut.point | c1 < -cut.point) # matrix of cor>cutpoint
c2 # examine; FALSE indicates cor<cutpoint

# plot correlations using modified panel.cor function
windows(22,12)
pairs(spp.trained.data[, c(53:64)], lower.panel = panel.smooth,
upper.panel = panel.cor,
main = "Maximum Temperature: Monthly Variable Correlation")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
temp_max/monthly/tmax_months_correlation.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
temp_max/monthly/tmax_months_correlation.jpeg", type = "jpeg")
dev.off()

# estimate VIP values => AIC & Adj deviance
tr.vip <- spp.trained.data[, c(1, 53:64)] # keep only P/A & predictors
pres <- 1 # column for presence:absence
v.start <- 2 # column start predictor variables
v.stop <- ncol(tr.vip) # last column predictor variables
v.num <- v.stop - 1 # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop) # call VIP function
dev.fit # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10 # max of y-axis
```
ylim.r <- range(0, d.max) # range y-axis
x.labs <- names(tr.vip[2:v.stop]) # x-axis labels
windows(22,12)
par(mfrow=c(1,1))
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r,
main = "Maximum Temperature: Monthly Variable
Importance",
ylab = "adj.D2", names = x.labs) # barplot
abline(h = 0) # add horizontal line
abline(mean(dev.fit[, 2]), 0, lt = 3) # ref lines; dash=mean
adj.dev

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
  temp_max/monthly/temp_max_months_vips.pdf", type =
  "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
  temp_max/monthly/temp_max_months_vips.jpeg", type =
  "jpeg")
rstudioapi::documentSave()

dev.off()

#################################################################
####### seasonal
#######################################################
names(spp.trained.data)
# tavg seasons
#################################################################
windows(22,12)
par(mfrow = c(1, 5))
boxplot(spp.trained.data$tmax_winter ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "tmax_winter")
boxplot(spp.trained.data$tmax_spring ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "tmax_spring")
boxplot(spp.trained.data$tmax_summer ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "tmax_summer")
boxplot(spp.trained.data$tmax_fall ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "tmax_fall")
boxplot(spp.trained.data$tmax_year ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "tmax_year")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
  temp_max/seasonal/tmax_seasons_boxplots.pdf", type =
  "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
  temp_max/seasonal/tmax_seasons_boxplots.jpeg", type =
  "jpeg")
dev.off()

############### Correlation among tmax monthly predictors
###############
# numeric correlations
cut.point <- 0.7 # set cutpoint for correlation
c1 <- cor(spp.trained.data[, c(65:69)], use =
        "pairwise.complete.obs",
        method = "spearman") # est. correlation
c1 # examine
c2 <- subset(c1 > cut.point | c1 < -cut.point) # matrix of
cor>cutpoint
c2 # examine; FALSE indicates cor<cutpoint

# plot correlations using modified panel.cor function
windows(22,12)
pairs(spp.trained.data[, c(65:69)], lower.panel = panel.smooth,
      upper.panel = panel.cor,
      main = "Maximum Temperature: Seasonal Variable
      Correlation")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
temp_max/seasonal/tmax_seasonal_correlations.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
temp_max/seasonal/tmax_seasonal_correlations.jpeg", type = "jpeg")
dev.off()

# estimate VIP values => AIC & Adj deviance
tr.vip <- spp.trained.data[, c(1, 65:69)] # keep only P/A &
predictors
pres <- 1 # column for presence:absence
v.start <- 2 # column start predictor variables
v.stop <- ncol(tr.vip) # last column predictor variables
v.num <- v.stop - 1 # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop) #
call VIP function
dev.fit # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10 # max of
y-axis
ylim.r <- range(0, d.max) # range y-axis
x.labs <- names(tr.vip[2:v.stop]) # x-axis labels
windows(22,12)
par(mfrow = c(1,1))
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r,
       main = "Maximum Temperature: Seasonal Variable
       Importance",
       xlab = "Predictor Variables", ylab = "VIP")
ylab = "adj.D2", names = x.labs) # barplot
abline(h = 0) # add horizontal line
abline(mean(dev.fit[, 2]), 0, lt = 3) # ref lines; dash=mean
adj.dev

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
weather/temp_max/seasonal/temp_max_seasonal_vips.pdf",
type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
weather/temp_max/seasonal/temp_max_seasonal_vips.jpeg",
type = "jpeg")
dev.off()

# temperature average Variable Selection

# tmax_04
# tmax_fall
spp_vars <- c(spp_vars, "tmax_04", "tmax_fall")

# tmin descriptive stats
names(spp.trained.data)

# mean by PA
aggregate(spp.trained.data[, 70:81], list(presabs =
spp.trained.data$PA),
    FUN = mean, na.rm = T)
# median by PA
aggregate(spp.trained.data[, 70:81], list(presabs =
spp.trained.data$PA),
    FUN = median, na.rm = T)
# mode by PA; requires package modeest
aggregate(spp.trained.data[, 70:81], list(presabs =
spp.trained.data$PA),
    FUN = mfv, na_rm = T)

# tmin graphs
windows(22,12)
par(mfrow = c(1, 12))
boxplot(spp.trained.data$tmin_01 ~ spp.trained.data$PA,
xlab = "Presence:Absence",
...
boxplot(spp.trained.data$tmin_01 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_01")

boxplot(spp.trained.data$tmin_02 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_02")

boxplot(spp.trained.data$tmin_03 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_03")

boxplot(spp.trained.data$tmin_04 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_04")

boxplot(spp.trained.data$tmin_05 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_05")

boxplot(spp.trained.data$tmin_06 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_06")

boxplot(spp.trained.data$tmin_07 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_07")

boxplot(spp.trained.data$tmin_08 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_08")

boxplot(spp.trained.data$tmin_09 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_09")

boxplot(spp.trained.data$tmin_10 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_10")

boxplot(spp.trained.data$tmin_11 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_11")

boxplot(spp.trained.data$tmin_12 ~ spp.trained.data$PA, 
  xlab = "Presence:Absence",
  ylab = "values", main = "tmin_12")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
 temp_min/monthly/tmin_months_boxplots.pdf", type = 
"pdf")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
 temp_min/monthly/tmin_months_boxplots.jpeg", type = 
"jpeg")

dev.off()

################################ Correlation among tmin monthly predictors
################################
# numeric correlations

cut.point <- 0.7 # set cutpoint for correlation

c1 <- cor(spp.trained.data[, c(70:81)], use = 
  "pairwise.complete.obs",
  method = "spearman") # est. correlation
c1 # examine

c2 <- subset(c1 > cut.point | c1 < -cut.point) # matrix of cor>cutpoint
c2 # examine; FALSE indicates cor<cutpoint

# plot correlations using modified panel.cor function
windows(22,12)
pairs(spp.trained.data[, c(70:81)], lower.panel = panel.smooth,
    upper.panel = panel.cor,
    main = "Minimum Temperature: Monthly Variable Correlation")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
    temp_min/monthly/tmin_months_correlation.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
    temp_min/monthly/tmin_months_correlation.jpeg", type = "jpeg")
dev.off()

# estimate VIP values => AIC & Adj deviance
tr.vip <- spp.trained.data[, c(1, 70:81)] # keep only P/A &
predictors
pres <- 1 # column for presence:absence
v.start <- 2 # column start predictor variables
v.stop <- ncol(tr.vip) # last column predictor variables
v.num <- v.stop - 1 # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop) #
call VIP function
dev.fit # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10 # max of
y-axis
ylim.r <- range(0, d.max) # range y-axis
x.labs <- names(tr.vip[2:v.stop]) # x-axis labels
windows(22,12)
par(mfrow=c(1,1))
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r,
    main = "Minimum Temperature: Monthly Variable
Importance",
    ylab = "adj.D2", names = x.labs) # barplot
abline(h = 0) # add horizontal line
abline(mean(dev.fit[, 2]), 0, lty = 3) # ref lines; dash=mean
adj.dev

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
    temp_min/monthly/temp_min_months_vips.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/
    temp_min/monthly/temp_min_months_vips.jpeg", type = "jpeg")
rstudioapi::documentSave()
dev.off()

########################################################
# seasonal
########################################################
names(spp.trained.data)
# tmin seasons
########################################################
windows(22,12)
par(mfrow = c(1, 5))
boxplot(spp.trained.data$tmin_winter ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
       ylab = "values", main = "tmin_winter")
boxplot(spp.trained.data$tmin_spring ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
       ylab = "values", main = "tmin_spring")
boxplot(spp.trained.data$tmin_summer ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
       ylab = "values", main = "tmin_summer")
boxplot(spp.trained.data$tmin_fall ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
       ylab = "values", main = "tmin_fall")
boxplot(spp.trained.data$tmin_year ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
       ylab = "values", main = "tmin_year")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/ temp_min/seasonal/tmin_seasons_boxplots.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/ temp_min/seasonal/tmin_seasons_boxplots.jpeg", type = "jpeg")
dev.off()

########################################################################
# Correlation among tmax monthly predictors
########################################################################
# numeric correlations
cut.point <- 0.7 # set cutpoint for correlation
c1 <- cor(spp.trained.data[, c(82:86)], use = "pairwise.complete.obs",
          method = "spearman") # est. correlation
c1 # examine
c2 <- subset(c1 > cut.point | c1 < -cut.point) # matrix of cor>cutpoint
c2 # examine; FALSE indicates cor<cutpoint

# plot correlations using modified panel.cor function
windows(22,12)
pairs(spp.trained.data[, c(82:86)], lower.panel = panel.smooth,
      upper.panel = panel.cor,
main = "Minimum Temperature: Seasonal Variable Correlation"

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/ temp_min/seasonal/tmin_seasonal_correlations.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/ temp_min/seasonal/tmin_seasonal_correlations.jpeg", type = "jpeg")

dev.off()

# estimate VIP values => AIC & Adj deviance
tr.vip <- spp.trained.data[, c(1, 82:86)] # keep only P/A & predictors
pres <- 1 # column for presence:absence
v.start <- 2 # column start predictor variables
v.stop <- ncol(tr.vip) # last column predictor variables
v.num <- v.stop - 1 # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop) # call VIP function
dev.fit # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10 # max of y-axis
ylim.r <- range(0, d.max) # range y-axis
x.labs <- names(tr.vip[2:v.stop]) # x-axis labels
windows(22,12)
par(mfrow = c(1,1))
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r,
main = "Minimum Temperature: Seasonal Variable Importance",
ylab = "adj.D2", names = x.labs) # barplot
abline(h = 0) # add horizontal line
abline(mean(dev.fit[, 2]), 0, lt = 3) # ref lines; dash=mean adj.dev

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/ temp_min/seasonal/temp_min_seasonal_vips.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/weather/ temp_min/seasonal/temp_min_seasonal_vips.jpeg", type = "jpeg")

dev.off()

########## temperature average Variable Selection

# tmin_02
# tmin_winter
spp_vars <- c(spp_vars, "tmin_02", "tmin_winter")

# topo descriptive stats
names(spp.trained.data)

# mean by PA
aggregate(spp.trained.data[, 87:93], list(presabs = spp.trained.data$PA),
  FUN = mean, na.rm = T)
# median by PA
aggregate(spp.trained.data[, 87:93], list(presabs = spp.trained.data$PA),
  FUN = median, na.rm = T)
# mode by PA; requires package modeest
aggregate(spp.trained.data[, 87:93], list(presabs = spp.trained.data$PA),
  FUN = mfv, na_rm = T)

# topo graphs
windows(22,12)
par(mfrow = c(1, 7))
boxplot(spp.trained.data$elev ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "elevation")
boxplot(spp.trained.data$slope ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "slope")
boxplot(spp.trained.data$aspect ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "aspect")
boxplot(spp.trained.data$rough ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "roughness")
boxplot(spp.trained.data$exp1nrm ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "exp1nrm")
boxplot(spp.trained.data$exp3nrm ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "exp3nrm")
boxplot(spp.trained.data$exp5nrm ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "exp5nrm")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/\n  topo/topo_boxplots.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/topo/topo_boxplots.jpeg", type = "jpeg")

dev.off()

################ Correlation among tmin monthly predictors
# numeric correlations

cut.point <- 0.7 # set cutpoint for correlation
c1 <- cor(spp.trained.data[, c(87:93)], use = "pairwise.complete.obs", method = "spearman") # est. correlation
c1 # examine
c2 <- subset(c1 > cut.point | c1 < -cut.point) # matrix of cor>cutpoint
c2 # examine; FALSE indicates cor<cutpoint

# plot correlations using modified panel.cor function
windows(22,12)
pairs(spp.trained.data[, c(87:93)], lower.panel = panel.smooth, upper.panel = panel.cor, main = "Topographic Variable Correlation")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/topo/topo_correlation.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/topo/topo_correlation.jpeg", type = "jpeg")
dev.off()

# estimate VIP values => AIC & Adj deviance
tr.vip <- spp.trained.data[, c(1, 87:93)] # keep only P/A & predictors
pres <- 1 # column for presence:absence
v.start <- 2 # column start predictor variables
v.stop <- ncol(tr.vip) # last column predictor variables
v.num <- v.stop - 1 # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop) # call VIP function
dev.fit # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10 # max of y-axis
ylim.r <- range(0, d.max) # range y-axis
x.labs <- names(tr.vip[2:v.stop]) # x-axis labels
windows(22,12)
par(mfrow=c(1,1))
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r, main = "Topographic Variable Importance", ylab = "adj.D2", names = x.labs) # barplot
abline(h = 0) # add horizontal line
abline(mean(dev.fit[, 2]), 0, lt = 3) # ref lines; dash=mean
adj.dev

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
topo/topo_vips.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
topo/topo_vips.jpeg", type = "jpeg")

rstudioapi::documentSave()

dev.off()

# Topographic Variable Selection
# elev

spp_vars <- c(spp_vars, "elev")

# topo descriptive stats
names(spp.trained.data)

# mean by PA
aggregate(spp.trained.data[, 94:109], list(presabs = spp.trained.data$PA),
  FUN = mean, na.rm = T)

# median by PA
aggregate(spp.trained.data[, 94:109], list(presabs = spp.trained.data$PA),
  FUN = median, na.rm = T)

# mode by PA; requires package modeest
aggregate(spp.trained.data[, 94:109], list(presabs = spp.trained.data$PA),
  FUN = mfv, na_rm = T)

# topo graphs
windows(22,12)
par(mfrow = c(1, 15))
boxplot(spp.trained.data$phmin ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "phmin")
boxplot(spp.trained.data$phmax ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "phmax")
boxplot(spp.trained.data$frag3to10 ~ spp.trained.data$PA,
  xlab = "Presence:Absence",
  ylab = "values", main = "frag3to10")
boxplot(spp.trained.data$sieveno4 ~ spp.trained.data$PA,
xlab = "Presence:Absence",
ylab = "values", main = "sieveno4")
boxplot(spp.trained.data$sieveno10 ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "sieveno10")
boxplot(spp.trained.data$sieveno40 ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "sieveno40")
boxplot(spp.trained.data$sieveno200 ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "sieveno200")
boxplot(spp.trained.data$sand ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "sand")
boxplot(spp.trained.data$silt ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "silt")
boxplot(spp.trained.data$clay ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "clay")
boxplot(spp.trained.data$omr ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "omr")
boxplot(spp.trained.data$dryweight ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "dryweight")
boxplot(spp.trained.data$ksat ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "ksat")
boxplot(spp.trained.data$awc ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "awc")
boxplot(spp.trained.data$wsat ~ spp.trained.data$PA,
       xlab = "Presence:Absence",
ylab = "values", main = "wsat")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
soils/soils_boxplots.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/
soils/soils_boxplots.jpeg", type = "jpeg")

dev.off()

# Correlation among tmin monthly predictors

cut.point <- 0.7 # set cutpoint for correlation
c1 <- cor(spp.trained.data[, c(94:109)], use =
"pairwise.complete.obs",
method = "spearman") # est. correlation
c1 # examine

c2 <- subset(c1 > cut.point | c1 < -cut.point) # matrix of
cor>cutpoint
c2 # examine; FALSE indicates cor<cutpoint
# plot correlations using modified panel.cor function
windows(22,12)
pairs(spp.trained.data[, c(94:109)], lower.panel = panel.smooth, upper.panel = panel.cor, main = "Edaphic Variable Correlation")

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/soils/soils_correlation.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/soils/soils_correlation.jpeg", type = "jpeg")
dev.off()

# estimate VIP values => AIC & Adj deviance
tr.vip <- spp.trained.data[, c(1, 94:109)] # keep only P/A & predictors
pres <- 1 # column for presence:absence
v.start <- 2 # column start predictor variables
v.stop <- ncol(tr.vip) # last column predictor variables
v.num <- v.stop - 1 # number predictor variables
dev.fit <- varimp.glm(tr.vip, tr.vip, pres, v.start, v.stop) # call VIP function
dev.fit # output matrix; col=1 AIC, col=2 Adj deviance

# built basic barplot if desired
d.max <- ceiling(signif(max(dev.fit[, 2]), 2) * 10)/10 # max of y-axis
ylim.r <- range(0, d.max) # range y-axis
x.labs <- names(tr.vip[2:v.stop]) # x-axis labels
windows(22,12)
par(mfrow=c(1,1))
barplot(dev.fit[, 2], col = "darkgreen", ylim = ylim.r, main = "Edaphic Variable Importance", ylab = "adj.D2", names = x.labs) # barplot
abline(h = 0) # add horizontal line
abline(mean(dev.fit[, 2]), 0, lt = 3) # ref lines; dash=mean adj.dev

savePlot(filename = "F:/Josh/Plants/spp/Data_explore/soils/soils_vips.pdf", type = "pdf")
savePlot(filename = "F:/Josh/Plants/spp/Data_explore/soils/soils_vips.jpeg", type = "jpeg")
dev.off()

################## Topographic Variable Selection
#############################
# phave
# sand
# omr
# ksat

spp_vars <- c(spp_vars, "phave", "sand", "omr", "ksat")

# save vars
save(spp_vars, file = "F:/Josh/Plants/spp/Data_explore/spp_vars.RData")

# Select variables for model from trained data

load("F:/Josh/Plants/spp/Data_explore/spp_vars.RData") # load data
spp_vars <- c(spp_vars, "PA") # add pres abs to list of species variables

spp.mod.vars <- spp.trained.data[, spp_vars] # select columns
spp.mod.vars <- spp.mod.vars[, c(ncol(spp.mod.vars), 1:(ncol(spp.mod.vars)-1))] # reorder, names(spp.mod.vars) # check names

head(spp.mod.vars, 2) # examine data
dim(spp.mod.vars) # examine

# save in trained data
save(spp.mod.vars, file = "F:/Josh/Plants/spp/trained_data/spp.mod.vars.RData")

# Load Predictor Variables

spp.bbox.RT <- terra::rast("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RT.tif")
load("F:/Josh/Plants/spp/presence/spp_pres.RData", verbose = T)
load("F:/Josh/Plants/spp/trained_data/spp.mod.vars.RData")

# Load predictors
preds.dom <- terra::rast("F:/Josh/DATA/Predictor_layers/All_Stack/preds.dom.final.tif")
preds.dom <- terra::crop(preds.dom, spp.bbox.RT) # crop preds
terra::plot(preds.dom)

# need to select species mod specific variables from preds.dom stack
names(preds.dom)
spp.mod.vars.PAdrop <- spp.mod.vars[2:ncol(spp.mod.vars)]
names(spp.mod.vars.PAdrop)
spp.preds <- terra::subset(preds.dom, names(spp.mod.vars.PAdrop))
terra::plot(spp.preds)

# updated preds
terra::writeRaster(spp.preds, filename = "F:/Josh/Plants/
spp/spp_preds/spp.preds.tif")

#################################################################
########### Random Forest Model
########################################
#################################################################

# Load preds
spp.preds <-
terra::rast("F:/Josh/Plants/spp/spp_preds/spp.preds.tif")
load("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RDATA", verbose = T)

library(randomForest)
library(PresenceAbsence)

path.RF <- "F:/Josh/Plants/spp/Mods/Mod.RF"
setwd(path.RF)

load("F:/Josh/Plants/spp/trained_data/spp.mod.vars.RData",
verbose = T)
dat1 <- spp.mod.vars

# examine training data
dim(dat1); table(dat1$PA) # examine data
head(dat1); str(dat1) # NOT RUN: examine data
class(dat1$PA) # response data class as factor?
#dat1$PA <- as.factor(dat1$PA)
table(is.na(dat1))
dat1 <- na.omit(dat1)

# random forest structure function
mod.form <- function(dat2, r.col, p.col){
n.col <- ncol(dat2) # number of columns in DF
resp <- colnames(dat2[r.col])
resp <- paste("as.factor(" , colnames(dat2[r.col]), ",")", sep = """
 pred <- colnames(dat2[c(p.col:n.col)])
mod.formula <- as.formula(paste(resp, ",", paste(pred, collapse = "+"))}
}
# RF model 1
mod1.RF <- randomForest(mod.form(dat1, 1, 2), importance = T, keep.forest = T, data = dat1)
mod1.pred <- predict(mod1.RF, type = "prob")[, 2]

save(mod1.RF, file="mod1.RF.RData") # save
save(mod1.pred, file = "mod1.pred.RData") # save

head(mod1.pred)
str(dat1)

mod1 <- "mod1.RF"
dat3 <- cbind(mod1, dat1[, 1], mod1.pred)

head(dat3, 2)

mod.cut <- optimal.thresholds(dat3, opt.methods = c("MaxKappa"))
mod.cut

save(mod.cut, file = "mod.cut.RData")

mod1.cfmat <- table(dat3[, 2],
                    factor(as.numeric(dat3$mod1.pred >=
                               mod.cut$mod1.pred)))
mod1.cfmat

save(mod1.cfmat, file = "mod1.cfmat.RData")

mod1.acc <- presence.absence.accuracy(dat3,
                                       threshold =
                                       mod.cut$mod1.pred, st.dev = F)
tss <- mod1.acc$sensitivity + mod1.acc$specificity - 1
mod1.acc <- cbind(mod1.acc[1:7], tss)
mod1.acc[c(1, 4:5, 7:8)]

save(mod1.acc, file = "mod1.acc.RData")

windows(22, 12)
auc.roc.plot(dat3, color = T)
savePlot("auc.roc.plot.RF.jpeg")
dev.off()

mod1.RF

cbind(mod1.cfmat, rbind(mod1.cfmat[3]/sum(mod1.cfmat[c(1,3)]),
                          mod1.cfmat[2]/sum(mod1.cfmat[c(2,4)])))

mod1.RF$confusion
head(dat3, 2)

oob.acc <- presence.absence.accuracy(dat3, st.dev = F)
tss <- oob.acc$sensitivity + oob.acc$specificity - 1
oob.acc <- cbind(oob.acc[1:7], tss)
oob.acc[c(1, 4:5, 7:8)]

save(oob.acc, file = "oob.acc.RData")

windows(22,12)
varImpPlot(mod1.RF, main = "Variable Importance")
savePlot("varimplot.RF.jpeg")

#windows(22,12)
#par(mfrow = c(1, 2))
#partialPlot(mod1.RF, dat1, bio_01, which.class = "1",
#  main = "Partial Plot: Bio_01", y = "Logit(Presence)"
#) dev.off()

library(raster)
library(terra)
spp.predsR <- as(spp.preds, "Raster")

mod1.RFprobR <- predict(spp.predsR, mod1.RF,
                         filename = "mod1RFprob.tif",
                         type = "prob",
                         fun = predict, index = 2, overwrite = T)

# probability
save(mod1.RFprobR, file = "F:/Josh/Plants/spp/Mods/
     Mod.RF/mod1.RFprobR.RData") # save prob model

mod1.Rfclas <- reclassify(mod1.RFprobR, c(0, mod.cut [[2]]),
                          0, mod.cut [[2]],
                          1, 1)) # reclassify of raster
mod1.Rfclas.bin <- reclassify(mod1.RFprobR,c(0,.2,1,
                                          .2,.4,2,
                                          .4,.6,3,
                                          .6,.8,4,
                                          0.8,1,5))

mod1.Rfclas <- as(mod1.Rfclas, "SpatRaster") # reclassify as SpatRaster
mod1.Rfclas.bin <- as(mod1.Rfclas.bin, "SpatRaster") # reclassify as SpatRaster
mod1.Rfprob <- as(mod1.RFprobR, "SpatRaster")

terra::writeRaster(mod1.Rfclas, filename = "mod1.RFclas.tif",
                   overwrite = T)
terra::writeRaster(mod1.Rfclas.bin,
    filename = "mod1.RFclas.bin.tif", overwrite = T) # save
terra::writeRaster(mod1.Rfprob, filename = "mod1.RFprob.tif",
    overwrite = T)

# save rasters
mod1.Rfclas.R <- as(mod1.Rfclas, "Raster")
mod1.Rfclas.bin.R <- as(mod1.Rfclas.bin, "Raster")
mod1.Rfprob.R <- as(mod1.Rfprob, "Raster")

raster::writeRaster(mod1.Rfclas.R,
    filename = "mod1.RFclas.R.tif", overwrite = T)
raster::writeRaster(mod1.Rfclas.bin.R,
    filename = "mod1.RFclas.bin.R.tif", overwrite = T)
raster::writeRaster(mod1.Rfprob.R,
    filename = "mod1.RFprob.R.tif", overwrite = T)

load("F:/ Josh/Plants/spp/frames/bbox/spp.bbox.RDATA", verbose = T)
load("F:/Josh/Plants/spp/presence/spp_pres.RData", verbose = T)
windows(22,12)
par(mfrow=c(1,3))
terra::plot(mod1.Rfprob) # prediction
plot(spp.bbox.SF, add = T, col = "transparent")
#plot(spp.presPA$geometry, add = T, pch = 20)
terra::plot(mod1.Rfclas) # prediction reclassified
plot(spp.bbox.SF, add = T, col = "transparent")
#plot(spp.presPA$geometry, add = T, pch = 20)

terra::plot(mod1.Rfclas.bin) # prediction reclassified
plot(spp.bbox.SF, add = T, col = "transparent")
savePlot("mod1.RF_prob_class.jpeg")

par(mfrow=c(1,1))
plot(mod1.Rfprob) # prediction
plot(spp.bbox.SF, add = T, col = "transparent")
plot(spp.presPA$geometry, add = T, pch = 20)
savePlot("mod1.RF.prob.pdf")
savePlot("mod1.RF.prob.jpeg")

plot(mod1.Rfclas) # prediction reclassified
plot(spp.bbox.SF, add = T)
plot(spp.presPA$geometry, add = T, pch = 20)
savePlot("mod1.RF.clas.pdf")
savePlot("mod1.RF.clas.jpeg")
dev.off()

pres <- spp.presabs %>%
  subset(PA == 1)
abs <- spp.presabs %>%
  subset(PA == 0)
windows(22,12)
plot(mod1.Rfprob) # prediction
plot(spp.bbox.SF, add = T, col = "transparent")
plot(pres, add = T, pch = 20, col = "blue")
plot(abs, add = T, pch = 20, col = "red")

savePlot("mod1.RF.pres.abs.pdf")
savePlot("mod1.RF.pres.abs.jpeg")
# End Random Forest

#################################################################
# Generalized Linear Model
#################################################################
# mod.LR pathway
path.LR <- "F:/Josh/Plants/spp/Mods/Mod.LR"
setwd(path.LR)
load("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RDATA", verbose = T)

# Load preds
spp.preds <- terra::rast("F:/Josh/Plants/spp/spp_preds/spp.preds.tif")

# Libraries
library(PresenceAbsence)
library(DAAG)
library(raster)

# load presabs trained data
load("F:/Josh/Plants/spp/trained_data/spp.mod.vars.RData", verbose = T)
dat1 <- spp.mod.vars; dim(dat1); table(dat1$PA); head(dat1, 2) # examine data
dat1 <- na.omit(dat1)

# GLM Model Formula
mod.form <- function(dat, r.col, p.col){
  n.col <- ncol(dat) # Number of columns in the dataframe
  resp <- colnames(dat[r.col]) # assign response a column name
  pred <- colnames(dat[c(p.col:n.col)]) # assign preds column names
  mod.formula <- as.formula(paste(resp, "~", paste(pred, collapse = "+"))) # formula
names(spp.mod.vars) # names of predictors

# Basic GLM with link = Binomial, dataframe = dat1
mod1.LR <- glm(as.factor(PA) ~ prec_01 + prec_summer + srad_01 + srad_summer + tavg_04 + tavg_summer + tmax_04 + tmax_fall + tmin_02 + tmin_winter + elev + phave + sand + omr + ksat,
family = binomial, data = dat1)

# save mod1.LR
save(mod1.LR, file =
"F:/Josh/Plants/spp/Mods/Mod.LR/mod1.LR.RData")

# LR model 1 summary
summary(mod1.LR)

# model 1 fit
mod1.fit <- 100 * (1 - mod1.LR$deviance/mod1.LR$null.deviance) # model fit
mod1.fit # examine
mod1.pred <- predict(mod1.LR, type = "response") # model prediction

save(mod1.pred, file =
"F:/Josh/Plants/spp/Mods/Mod.LR/mod1.pred.LR.RData")

# model 1 prediction
head(mod1.pred) # examine prediction

# LR model 2; backwards stepwise variable reduction
mod2.LR <- step(mod1.LR, trace = F)

# save mod2.LR
save(mod2.LR, file =
"F:/Josh/Plants/spp/Mods/Mod.LR/mod2.LR.RData")

# model 2 fit
100 * (1 - mod2.LR$deviance/mod2.LR$null.deviance) # model 1 fit
100 * (1 - mod1.LR$deviance/mod1.LR$null.deviance)

# model 2 prediction
mod2.pred <- predict(mod2.LR, type = "response")

save(mod2.pred, file =
"F:/Josh/Plants/spp/Mods/Mod.LR/mod2.pred.LR.RData")

# examine model predictions
head(mod2.pred) # model 2
head(mod1.pred) # model 1  

# model 2 summary  
summary(mod2.LR)  

# add var to keep track of model  
modl <- "mod2.LR"  
dat2 <- cbind(modl, dat1[1], mod2.pred) # build prediction df with mod2 preds  

# Create confusion matrix  
mod.cut <- optimal.thresholds(dat2, opt.methods = c("MaxKappa"))  
mod2.cfmt <- table(dat2[[2]],  
    factor(as.numeric(dat2$mod2.pred >=  
    mod.cut$mod2.pred)))  
mod2.cfmt # examine confusion matrix  

save(mod.cut, file = "F:/Josh/Plants/spp/  
    Mods/Mod.LR/mod.cut.LR.RData") # save mod.cut  
save(mod2.cfmt, file = "F:/Josh/Plants/spp/  
    Mods/Mod.LR/mod2.cfmt.LR.RData")  

# Model accuracies  
mod2.acc <- presence.absence.accuracy(dat2,  
    threshold =  
    mod.cut$mod2.pred,  
    st.dev = F)  
tss <- mod2.acc$sensitivity + mod2.acc$specificity - 1  
mod2.acc <- cbind(mod2.acc[1:7], tss) # bind metrics  
mod2.acc[c(1, 4:5, 7:8)] # examine accuracies  

save(mod2.acc, file =  
"F:/Josh/Plants/spp/Mods/Mod.LR/mod2.acc.LR.RData")  

windows(22,12)  
auc.roc.plot(dat2, color = T)  
savePlot("auc.roc.plot.LR.jpeg", type = "jpeg")  
dev.off()  

# Cross validation accuracies  
mod2.cv10 <- CVbinary(mod2.LR, nfolds = 5, print.details = F) # crossval  
lss(mod2.cv10)  
mod2.cv10.1 <- mod2.cv10$cvhat  

save(mod2.cv10, file =  
"F:/Josh/Plants/spp/Mods/Mod.LR/mod2.cv10.RData")  

dat2 <- cbind(dat2, mod2.cv10.1)  
head(dat2)  

save(dat2, file =  
"F:/Josh/Plants/spp/Mods/Mod.LR/mod.predictionframe.RData")
# model CVbinary
mod2.cfmatCV <- table(dat2[[2]],
    factor(as.numeric(dat2$mod2.cv10.1 >=
        mod.cut$mod2.pred)))

# examine both mod confusion matrices
mod2.cfmatCV; mod2.cfmat
save(mod2.cfmatCV, file =
    "F:/Josh/Plants/spp/Mods/Mod.LR/mod2.cfmatCV.RData")

# calculate accuracies with std.dev = F
mod2.accB <- presence.absence.accuracy(dat2,
    threshold =
        mod.cut$mod2.pred,
        st.dev = F)
tss <- mod2.accB$sensitivity + mod2.accB$specificity - 1 # code TSS metric
mod2.accB <- cbind(mod2.accB[1:7], tss) # bind all metrics
mod2.accB[c(1, 4:5, 7:8)] # examine accuracies
save(mod2.accB, file =
    "F:/Josh/Plants/spp/Mods/Mod.LR/mod2.accB.RData")

# view both cfmats in same AUC plot
windows(22,12)
auc.roc.plot(dat2, color = T)
savePlot("auc.roc.plot.LR.BOTH.jpeg", type = "jpeg")

# spatial prediction
spp.predsR <- as(spp.preds, "Raster")
modFprob.LR.1 <- predict(spp.predsR, mod1.LR, filename =
    "mod2.LRprob.tif",
    type = "response", fun = predict,
    index = 2, overwrite = T)

# classified prediction
modFprobclas.R=reclassify(modFprob.LR.1,c(0,mod.cut[[2]],0,mod.cut[[2]],1,1))

# bins
modFprobclas.bin.R <- reclassify(modFprob.LR.1,c(0,.2,1,
    .2,.4,2,
    .4,.6,3,
    .6,.8,4,
    0.8,1,5))
raster::plot(modFprobclas.bin.R)

modFprobclas <- as(modFprobclas.R, "SpatRaster")
modFprobclas.bin <- as(modFprobclas.bin.R, "SpatRaster")
modFprob.LR <- as(modFprob.LR.1, "SpatRaster")
load("F:/Josh/Plants/spp/Mods/Mod.LR/modFprob.LR.RData")
terra::plot(modFprob.LR.1)
plot(spp.bbox.SF, add = T, col = "transparent")

terra::writeRaster(modFprob.LR, filename = "F:/Josh/Plants/
spp/Mods/Mod.LR/mod.LRprob.tif", overwrite = T)
terra::writeRaster(modFprobclas, filename = "F:/Josh/Plants/
spp/Mods/Mod.LR/mod.LRclas.tif", overwrite = T)
terra::writeRaster(modFprobclas.bin, filename = "F:/Josh/Plants/
spp/Mods/Mod.LR/mod.LRclas.bin.tif", overwrite = T)

raster::writeRaster(modFprob.LR.1, filename = "F:/Josh/Plants/
spp/Mods/Mod.LR/mod2.LRprob.R.tif", overwrite = T)
raster::writeRaster(modFprobclas.R, filename = "F:/Josh/Plants/
spp/Mods/Mod.LR/mod2.LRclas.R.tif", overwrite = T)
raster::writeRaster(modFprobclas.bin.R, filename = "F:/Josh/Plants/
spp/Mods/Mod.LR/mod2.LRclas.bin.R.tif", overwrite = T)

rstudioapi::documentSave()

#########################################################################
################## Boosted Regression Trees
#########################################################################
path.brt <- "F:/Josh/Plants/spp/Mods/Mod.BRT"
setwd(path.brt)

# Libraries
library(gbm)
library(dismo)
library(PresenceAbsence)

# load presabs training data; examine data
load("F:/Josh/Plants/spp/trained_data/spp.mod.vars.RData",
    verbose = T)
dat1 <- spp.mod.vars; dim(dat1); table(dat1$PA); head(dat1, 2) #
examine data

# BRT model formula
dat1 <- na.omit(dat1) # remove NAS
resp <- paste("as.factor(" , colnames(dat1[1]), ""), sep = "");
resp # assign resp to PA column
n.col <- ncol(dat1); n.col # number of total columns
pred <- 2:n.col; pred # column numbers of dataset predictors

# BRT Model number 1, gbm.y = PA column; learning rate 1e-04
mod1.BRT <- gbm.step(data = dat1, gbm.x = pred, gbm.y = 1, family = "bernoulli",
                      tree.complexity = 3, learning.rate = 1e-04,
                      bag.fraction = 0.75, n.folds = 10,
                      n.trees = 50, plot.main = TRUE,
                      keep.fold.fit = TRUE)
save(mod1.BRT, file = "F:/Josh/Plants/spp/Mods/Mod.BRT/mod1.BRT.RData")

# BRT Model number 2, gbm.y = PA column; learning rate .1
mod2.BRT <- gbm.step(data = dat1, gbm.x = pred, gbm.y = 1, family = "bernoulli",
                      tree.complexity = 3, learning.rate = .1,
                      bag.fraction = 0.75, n.folds = 10,
                      n.trees = 50, plot.main = TRUE,
                      keep.fold.fit = TRUE)
save(mod2.BRT, file = "F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.BRT.RData")

# BRT Model number 3, gbm.y = PA column; learning rate .01
mod3.BRT <- gbm.step(data = dat1, gbm.x = pred, gbm.y = 1, family = "bernoulli",
                      tree.complexity = 3, learning.rate = .01,
                      bag.fraction = 0.75, n.folds = 10,
                      n.trees = 50, plot.main = TRUE,
                      keep.fold.fit = TRUE)
save(mod3.BRT, file = "F:/Josh/Plants/spp/Mods/Mod.BRT/mod3.BRT.RData")

# selecting model 2

# Examine model data
ls(mod2.BRT) # examine BRT objects
head(mod2.BRT$fitted) # model fit values
head(dat1$spp65) # observed values
mod2.BRT$contributions # relative variable importance

# response:pred plots
par(mfrow = c(3, 4))
gbm.plot(mod2.BRT, n.plots = 10) # response:predictor plots
par(mfrow = c(1, 1))

# examine interactions
mod2.int <- gbm.interactions(mod2.BRT) # examine pairwise interactions
mod2.int$rank.list # matrix of 5 top interactions
# plot 3 top pairwise windows(22,12)
par(mfrow = c(1, 3))
gbm.perspec(mod2.BRT, mod2.int$rank.list[1, 1],
            mod2.int$rank.list[1, 3], theta = 30)
gbm.perspec(mod2.BRT, mod2.int$rank.list[2, 1],
            mod2.int$rank.list[2, 3], theta = 30)
gbm.perspec(mod2.BRT, mod2.int$rank.list[3, 1],
            mod2.int$rank.list[3, 3], theta = 30)
par(mfrow = c(1, 1))

savePlot("top3_pairwise.mod2.BRT.jpeg")

### BRT accuracy calculations

### BRT accuracy calculations

# build testing dataframe using model predictions
modl <- "mod2.BRT" # add var to keep track of model
dat2 <- cbind(modl, dat1[1], mod2.BRT$fitted,
              mod2.BRT$fold.fit) # build dataframe
names(dat2)[3:4] <- c("pred", "cvpred") # rename vars
head(dat2, 2) # just to see logit scale
dat2$cvpred <- exp(dat2$cvpred)/(1 + exp(dat2$cvpred)) # convert from logit
head(dat2, 2) # examine prediction dataframe

# determine best threshold via presenceabsence pkg
mod.cut <- optimal.thresholds(dat2, opt.methods = c("MaxKappa"))
mod.cut # examine

# save
save(mod.cut, file = "F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.cut.RData")

# generate the confusion matrix
mod2.cfmatR <- table(dat2[[2]],
                      factor(as.numeric(dat2$pred >=
                        mod.cut$pred)))
mod2.cfmatX <- table(dat2[[2]],
                      factor(as.numeric(dat2$cvpred >=
                        mod.cut$cvpred)))
mod2.cfmatR # examine
mod2.cfmatX # examine

# save
save(mod2.cfmatR, mod2.cfmatX, file = "F:/Josh/Plants/spp/ Mods/Mod.BRT/mod2.cfmat.RData")

# calculate model accuracies with standard deviation=F
mod2.acc <- presence.absence.accuracy(dat2, threshold =
                                      mod.cut$pred,
                                      ...)
st.dev = F)
tss <- mod2.acc$sensitivity + mod2.acc$specificity - 1  # code TSS metric
mod2.acc <- cbind(mod2.acc[1:7], tss)  # bind all metrics
mod2.acc[c(1, 4:5, 7:8)]  # examine accuracies

# save
save(mod2.acc, file = "F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.acc.RData")

# plotting AUC
windows(22,12)
auc.roc.plot(dat2, color = T)  # basic AUC plot; pkg PresenceAbsence
savePlot("auc.roc.mod2.BRT.jpeg")
dev.off()

# spatial prediction plots
# probability
mod2.BRTprob <- predict(spp.preds, mod2.BRT,
  n.trees = mod2.BRT$gbm.call$best.trees,
  type = "response")
terra::writeRaster(mod2.BRTprob, filename = "F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.BRTprob.tif")

par(mfrow=c(1,1))
# reclassified
mod2.BRTprobR <- as(mod2.BRTprob, "Raster")
mod2.BRTclas.R = reclassify(mod2.BRTprobR, c(0, mod.cut[[2]], 0,
  mod.cut[[2]], 1, 1))
plot(mod2.BRTclas.R)
mod2.BRTclas <- as(mod2.BRTclas.R, "SpatRaster")
plot(mod2.BRTclas)
terra::writeRaster(mod2.BRTclas, filename = "F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.BRTclas.tif")

# bins
mod2.BRTclas.bin.R <- reclassify(mod2.BRTprobR, c(0,.2,1,
  .2,.4,2,
  .4,.6,3,
  .6,.8,4,
  0.8,1,5))
mod2.BRTclas.bin <- as(mod2.BRTclas.bin.R, "SpatRaster")
plot(mod2.BRTclas.bin)
terra::writeRaster(mod2.BRTclas.bin, filename = "F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.BRTclas.bin.tif", overwrite = T)

windows(22,12)
par(mfrow = c(1,2))
plot(mod2.BRTprob)
plot(spp.bbox.SF, add = T)
plot(mod2.BRTclas)
plot(spp.bbox.SF, add = T)
savePlot("mod2.BRT_spatial_predictions.jpeg")
dev.off()

raster::writeRaster(mod2.BRTprobR, filename = "F:/Josh/Plants/spp/Mods/
    Mod.BRT/mod2.BRTprob.R.tif")
raster::writeRaster(mod2.BRTclas.R, filename = "F:/Josh/Plants/spp/Mods/
    Mod.BRT/mod2.BRTclas.R.tif")
raster::writeRaster(mod2.BRTclas.bin.R, filename = "F:/Josh/Plants/spp/
    Mods/Mod.BRT/mod2.BRTclas.bin.R.tif")

################################################################################
########################## Maximum Entropy
#################################################################

# path
path.MAX <- "F:/Josh/Plants/spp/Mods/Mod.MAX"
setwd(path.MAX)

load("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RDATA", verbose = T)

# libraries
library(dismo) # best for MAXENT
library(raster)
library(sp)
library(sf)

# load presabs training data
load("F:/Josh/Plants/spp/trained_data/spp.mod.vars.RData", verbose = T)
dat1 <- spp.mod.vars; dim(dat1); table(dat1$PA); head(dat1, 2) # examine data

# ALSO NEED X,Y,PA df
load("F:/Josh/Plants/spp/trained_data/spp.x,y.pa.RData", verbose = T);
dim(spp.x.y.pa)

# combine trained data and x,y's
dat1 <- cbind(spp.x.y.pa[c(1:2)], dat1); head(dat1, 2)

# load species specific preds
spp.preds <- terra::rast("F:/Josh/Plants/spp/spp_preds/spp.preds.tif")
# PA class as factor?
class(dat1$PA) # no, numeric

# assign number of columns from dataset
n.col <- ncol(dat1)

# Call for true presabs, number start of trained preds:n.col / 
dat1[PA column]
mod2.MAX <- maxent(dat1[4:n.col], dat1[3])
mod2.MAX # takes you out to the web

save(mod2.MAX, file = 
"F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.MAX.RData")

windows(22,12)
par(mfrow=c(1,1))
plot(mod2.MAX) # var importance plots
savePlot("mod2.MAX.varimp.jpeg")
response(mod2.MAX) # prediction vs. var plot
savePlot("mod2.MAX.pred-var.jpeg")
dev.off()

# MAXENT prediction and validation
mod2.pred <- predict(mod2.MAX, dat1[4:n.col]) # MAXENT prediction
save(mod2.pred, file = 
"F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.pred.RData")

mod1 <- "mod2.MAX" # var placeholder
dat2 <- cbind(mod1, dat1[3], mod2.pred) # dataframe with mod2 predictions
head(dat2)

# evaluate model (an x-fold process)
mod2.val <- evaluate(mod2.MAX, p = dat1[dat1$PA == 1, c(1:2)],
a = dat1[dat1$PA == 0, c(1,2)], x = spp.preds)
mod2.val # examine
save(mod2.val, file = 
"F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.val.RData")

# view maxent thresholds
mod.cut <- threshold(mod2.val); mod.cut
save(mod.cut, file = 
"F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.cut.RData")

# calculate model accuracies with standard deviation=F
library(PresenceAbsence) # for accuracy metrics
mod2.acc <- presence.absence.accuracy(dat2, threshold =
    mod2.acc[[2]],
        st.dev = F)
tss <- mod2.acc$sensitivity + mod2.acc$specificity - 1 # code TSS metric
mod2.acc <- cbind(mod2.acc[1:7], tss) # bind all metrics
mod2.acc[c(1, 4:5, 7:8)] # examine accuracies
save(mod2.acc, file =
  "F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.acc.RData")

# basic AUC plot
windows(22,12)
auc.roc.plot(dat2, color = T)
savePlot("auc.roc.MAX.jpeg")
dev.off()

# Spatial prediction maps
spp.predsR <- as(spp.preds, "Raster") # must be raster preds
mod2.MAXprob=predict(mod2.MAX,spp.predsR) # predict entire model domain

# basic reclassify based on threshold mod.cut per above
mod2.MAXclas=reclassify(mod2.MAXprob,c(0,mod.cut[[2]],0,mod.cut[[2]],1,1))

mod2.MAXclas.bin <- reclassify(mod2.MAXprob,c(0,.2,1,
  .2,.4,2,
  .4,.6,3,
  .6,.8,4,
  0.8,1,5))

par(mfrow=c(1,1))
plot(mod2.MAXclas.bin)
mod2.df <- as.data.frame(mod2.MAXclas)
max(mod2.df)

# plot
load("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RDATA", verbose = T)
par(mfrow=c(1,2))
plot(mod2.MAXprob)
plot(spp.bbox.SF, add = T)
plot(mod2.MAXclas)
plot(spp.bbox.SF, add = T)
#savePlot("mod2.MAX.prob.clas.jpeg")
mod2.MAXprob <- as(mod2.MAXprob, "SpatRaster")
mod2.MAXclas <- as(mod2.MAXclas, "SpatRaster")
mod2.MAXclas.bin <- as(mod2.MAXclas.bin, "SpatRaster")
terra::writeRaster(mod2.MAXprob, filename="F:/Josh/Plants/spp/Mods/
  Mod.MAX/mod2.MAXprob.tif")
terra::writeRaster(mod2.MAXclas, filename="F:/Josh/Plants/spp/Mods/
  Mod.MAX/mod2.MAXclas.tif")
terra::writeRaster(mod2.MAXclas.bin, filename="F:/Josh/Plants/spp/"
Mods/Mod.MAX/mod2.MAXclas.bin.tif")

# raster
mod2.MAXprob.R <- as(mod2.MAXprob, "Raster")
mod2.MAXclas.R <- as(mod2.MAXclas, "Raster")
mod2.MAXclas.bin.R <- as(mod2.MAXclas.bin, "Raster")
raster::writeRaster(mod2.MAXprob.R, filename="F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.MAXprob.R.tif")
raster::writeRaster(mod2.MAXclas.R, filename="F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.MAXclas.R.tif")
raster::writeRaster(mod2.MAXclas.bin.R, filename="F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.MAXclas.bin.R.tif")

#################################################################
############### Generalized Additive Models
#################################################################

# Libraries
library(gam)

# Load preds
spp.preds <-
terra::rast("F:/Josh/Plants/spp/spp_preds/spp.preds.tif")
load("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RDATA", verbose = T)

# load presabs trained data
load("F:/Josh/Plants/spp/trained_data/spp.mod.vars.RData", verbose = T)
dat1 <- spp.mod.vars; dim(dat1); table(dat1$PA); head(dat1, 2) # examine data

# examine training data
dim(dat1)
table(dat1$PA) # examine data
head(dat1) # examine data
str(dat1) # data structure

# Model formula
mod.form <- function(dat, r.col, p.col) {
  # generic glm formula construction function; inputs as:
  # resp => col 1 in dataframe; preds=> col 2 thru ncol such that p.col=2
  # NOTE: vars as factors must be coerced PRIOR to formula construction
  # example call: mod.form(dat1, 1, 2)
```r
n.col <- ncol(dat) # No. columns in dataframe
colnames(dat[r.col]) # assign resp column name
colnames(dat[c(p.col:n.col)]) # assign preds column names

mod.formula <- as.formula(paste(resp, "~", paste(pred, collapse = "+")))

# GAM mods
names(spp.preds)

# GLM model 0: all linear
# can use glm for linear model
mod0.GLM <- glm(as.factor(PA) ~ prec_11 + prec_fall + srad_12 +
  srad_winter + tavg_01 + tavg_winter + tmax_04 +
  tmax_spring + tmin_02 +
  tmin_winter + elev + rough + phave + sieveno4 +
  omr + ksat,
  family = binomial, data = dat1)
summary(mod0.GLM)

# GAM model 0: all linear
# can use glm for linear model
mod0.GAM <- gam(as.factor(PA) ~ prec_11 + prec_fall + srad_12 +
  srad_winter + tavg_01 + tavg_winter + tmax_04 +
  tmax_spring + tmin_02 +
  tmin_winter + elev + rough + phave + sieveno4 +
  omr + ksat,
  family = binomial, data = dat1)
summary(mod0.GAM)

# GAM model 1: all smoothers using defaults df (smoothers=4)
mod1.GAM <- gam(as.factor(PA) ~ s(prec_11) + s(prec_fall) +
  s(srad_12) +
  # make PA as.factor()
  s(srad_winter) + s(tavg_01) + s(tavg_winter) +
  s(tmax_04) + s(tmax_spring) + s(tmin_02) +
  s(tmin_winter) + s(elev) + s(rough) + s(phave) +
  s(sieveno4) + s(omr) +
  s(ksat),
  family = binomial, data = dat1)
summary(mod1.GAM) # model #1 summary

# GAM model 2: all smoothers using specified df (smoothers=5)
mod2.GAM <- gam(as.factor(PA) ~ s(prec_11, 5) + s(prec_fall, 5) +
  s(srad_12, 5) + # make PA as.factor()
  s(srad_winter, 5) + s(tavg_01, 5) + s(tavg_winter, 5) +
```
\[ s(t_{max \_04}, 5) + s(t_{max \_spring}, 5) + s(t_{min \_02}, 5) +
\]
\[ s(t_{min \_winter}, 5) + s(elev, 5) + s(rough, 5) +
\]
\[ s(phave, 5) + s(sieveno4, 5) + s(omr, 5) +
\]
\[ s(ksat, 5),
\]
family = binomial, data = dat1)

summary(mod2.GAM) # model #2 summary

# GAM model 3: stepwise w/diff scopes
mod3.GAM <- step.Gam(mod2.GAM,scope=list(
  "prec\_11" = ~1 + prec\_11 + s(prec\_11, 3) + s(prec\_11, 5),
  "prec\_fall" = ~1 + prec\_fall + s(prec\_fall, 3) + s(prec\_fall, 5),
  "srad\_12" = ~1 + srad\_12 + s(srad\_12, 3) + s(srad\_12, 5),
  "srad\_winter" = ~1 + srad\_winter + s(srad\_winter, 3) + s(srad\_winter, 5),
  "tavg\_01" = ~1 + tavg\_01 + s(tavg\_01, 3) + s(tavg\_01, 5),
  "tavg\_winter" = ~1 + tavg\_winter + s(tavg\_winter, 3) + s(tavg\_winter, 5),
  "tmax\_04" = ~1 + tmax\_04 + s(tmax\_04, 3) + s(tmax\_04, 5),
  "tmax\_spring" = ~1 + tmax\_spring + s(tmax\_spring, 3) + s(tmax\_spring, 5),
  "tmin\_02" = ~1 + tmin\_02 + s(tmin\_02, 3) + s(tmin\_02, 5),
  "tmin\_winter" = ~1 + tmin\_winter + s(tmin\_winter, 3) + s(tmin\_winter, 5),
  "elev" = ~1 + elev + s(elev, 3) + s(elev, 5),
  "rough" = ~1 + rough + s(rough, 3) + s(rough, 5),
  "phave" = ~1 + phave + s(phave, 3) + s(phave, 5),
  "sieveno4" = ~1 + sieveno4 + s(sieveno4, 3) + s(sieveno4, 5),
  "omr" = ~1 + omr + s(omr, 3) + s(omr, 5),
  "ksat" = ~1 + ksat + s(ksat, 3) + s(ksat, 5)), trace = F,
  gam.control(maxit = 1000))
summary(mod3.GAM) # model #3 summary
names(mod3.GAM$coefficients) # find the linear terms
summary.glm(mod3.GAM)$coefficients[c(1:2, 4:6, 10:11),] # access linear terms

save(mod0.GLM, mod0.GAM, mod1.GAM, mod2.GAM, mod3.GAM,
  file = "F:/Josh/Plants/spp/Mods/Mod.GAM/mods.GAM.RData")

# GAM mod comparisons
# estimate model fits
mod0.glm <- 100 * (1 - mod0.GLM$deviance/mod0.GLM$null.deviance)
# model fit
mod0.fit <- 100 * (1 - mod0.GAM$deviance/mod0.GAM$null.deviance)
# model fit
mod1.fit <- 100 * (1 - mod1.GAM$deviance/mod1.GAM$null.deviance)
# model fit
mod2.fit <- 100 * (1 - mod2.GAM$deviance/mod2.GAM$null.deviance)
# model fit
mod3.fit <- 100 * (1 - mod3.GAM$deviance/mod3.GAM$null.deviance)
# model fit
mods.fit <- cbind(mod0.glm, mod0.fit, mod1.fit, mod2.fit, mod3.fit)
mods.fit # examine

save(mod0.fit, mod0.GAM, mod1.GAM, mod2.GAM, mod3.GAM, file = "F:/Josh/Plants/spp/Mods/Mod.GAM/mods.GAM.RData")

library(PresenceAbsence) # load pkg for accuracy
mod.list <- c("mod0.GAM", "mod1.GAM", "mod2.GAM", "mod3.GAM")
i=1
for (i in 1:length(mod.list)) {
  pred <- predict(get(mod.list[i]), type = "response") # predict by model
  mod.num <- mod.list[i] # assign model No. to varname
  dat <- cbind(mod.num, dat1[1], pred) # build obs and prediction dataframe
  cut <- optimal.thresholds(dat, opt.methods = c("MaxKappa"))
  cfmat <- table(dat[[2]], factor(as.numeric(dat$pred >= cut$pred)))
  acc <- presence.absence.accuracy(dat, threshold = cut$pred, st.dev = F)
  tss <- acc$sensitivity + acc$specificity - 1 # code TSS metric
  acc <- cbind(acc[1:7], tss) # bind all metrics
  acc$model <- mod.num # variable substitution
  if (i == 1) mods.acc <- acc else mods.acc <- rbind(mods.acc, acc)
}
mods.acc[c(1, 4:5, 7:8)] # resubstitution accuracies by model

save(mods.acc, file = "F:/Josh/Plants/spp/Mods/Mod.GAM/mods.acc.RData")

# measures of 5-fold cross-validation accuracy
# absent set.seed() answers will differ among runs
library(DAAG) # load pkg for crossval
set.seed(1234) # set seed for replicability
mod.list <- c("mod0.GAM", "mod1.GAM", "mod2.GAM", "mod3.GAM")
for (i in 1:length(mod.list)) {
  cv.mod <- CVbinary(get(mod.list[i]), nfolds = 5, print.details = F)
  cv.pred <- cv.mod$cvhat # assign new name to jackknife estimates
  mod.num <- mod.list[i] # assign model No. to varname
  cv.dat <- cbind(mod.num, dat1[1], cv.pred) # build obs and prediction
cv.cut <- optimal.thresholds(cv.dat, opt.methods = c("PredPrev=Obs"))
  cv.cfmat <- table(cv.dat[[2]], factor(as.numeric(cv.dat$cv.pred >
cv.cut$cv.pred)))
  cv.acc <- presence.absence.accuracy(cv.dat,
threshold = cv.cut$cv.pred,

st.dev = F)
cv.tss <- cv.acc$sensitivity + cv.acc$specificity - 1  # code

TSS metric
cv.acc <- cbind(cv.acc[1:7], cv.tss)  # bind all metrics
cv.acc$model <- mod.num  # variable substitution
if (i == 1) mods.cvacc <- cv.acc else mods.cvacc <-
rbind(mods.cvacc, cv.acc)
}
mods.cvacc[c(1, 4:5, 7:8)]  # accuracies by model

save(mods.cvacc, file =
"F:/Josh/Plants/spp/Mods/Mod.GAM/mods.cvacc.RData")

#################################################################
###############
######## START FINAL GAM MODEL OUTPUT
# build GAM parameter table output; mod2.GAM chosen based on fit
/ accuracies
names(mod2.GAM$coefficients) # find the model terms

# build reduced model
mod.finl = step.Gam(mod2.GAM, scope = list(
  "prec_04" = ~1 + s(prec_04, 5),
  "prec_spring" = ~1 + s(prec_spring, 5),
  "srad_10" = ~1 + s(srad_10, 5),
  "srad_fall" = ~1 + s(srad_fall, 5),
  "tavg_04" = ~1 + s(tavg_04, 5),
  "tavg_spring" = ~1 + s(tavg_spring, 5),
  "tmax_03" = ~1 + s(tmax_03, 5),
  "tmax_spring" = ~1 + s(tmax_spring, 5),
  "tmin_04" = ~1 + s(tmin_04, 5),
  "tmin_spring" = ~1 + s(tmin_spring, 5),
  "elev" = ~1 + s(elev, 5),
  "slope" = ~1 + s(slope, 5),
  "rough" = ~1 + s(rough, 5),
  "Frag3to10" = ~1 + s(Frag3to10, 5),
  "sieveno40" = ~1 + s(sieveno40, 5),
  "clay" = ~1 + s(clay, 5)), trace = F)

summary(mod.finl)

save(mod.finl, file =
"F:/Josh/Plants/spp/Mods/Mod.GAM/mod.finl.RData")

# final model fit and accuracy
modfinl.fit <- 100 * (1 -
mod.finl$deviance/mod.finl$null.deviance)  # model fit

save(modfinl.fit, file =
"F:/Josh/Plants/spp/Mods/Mod.GAM/modfinl.fit.RData")

# final model resubstituion accuracy
modl <- "modfinl.GAM" # add var to keep track of model
modfinl.pred <- predict(mod.finl, type = "response") # predict by model
dat2 <- cbind(modl, dat1[1], modfinl.pred) # build dataframe w/modfinl
mod.cut <- optimal.thresholds(dat2, opt.methods = c("MaxKappa")) # threshold
mod.cut # examine threshold

save(mod.cut, file = "F:/Josh/Plants/spp/Mods/Mod.GAM/mod.cut.RData")

# generate resubstitution confusion matrix
modfinl.cfm <- table(dat2[[2]],
 factor(as.numeric(dat2$modfinl.pred >=
 mod.cut$modfinl.pred)))
modfinl.cfm # examine

save(modfinl.cfm, file = "F:/Josh/Plants/spp/Mods/Mod.GAM/
 modfinl.cfm.RData")

# calculate model resubstitution accuracies with standard deviation=F
modfinl.acc <- presence.absence.accuracy(dat2,
 threshold =
 mod.cut$modfinl.pred,
 st.dev = F)
tss <- modfinl.acc$sensitivity + modfinl.acc$specificity - 1 # code TSS metric
modfinl.acc <- cbind(modfinl.acc[1:7], tss) # bind all metrics
modfinl.acc[c(1, 4:5, 7:8)] # examine resubstitution accuracies

save(modfinl.acc, file = "F:/Josh/Plants/spp/Mods/Mod.GAM/modfinl.acc.RData")

# final model 5-fold cross-validation accuracy
cv.modfinl <- CVbinary(mod.finl, nfolds = 5, print.details = F) # crossval
cv.predfinl <- cv.modfinl$cvhat # assign new name to jackknife estimates
cv.datfinl <- cbind(modl, dat1[1], cv.predfinl) # build obs and prediction
cv.cutfinl <- optimal.thresholds(cv.datfinl, opt.methods =
 c("PredPrev=Obs"))

# generate 5-fold cross-validation confusion matrix
cv.cfmfinl <- table(cv.datfinl[[2]],
 factor(as.numeric(cv.datfinl$cv.predfinl >
cv.cutfinl$cv.predfinl))))

# calculate model 5-fold cross-validation accuracies with standard deviation=F
cv.accfinl <- presence.absence.accuracy(cv.datfinl,
threshold = cv.cutfinl$cv.predfinl,

accuracies

st.dev = F) # calculate

TSS metric
cv.accfinl <- cbind(cv.accfinl[1:7], tss) # bind all metrics
cv.accfinl$model <- modl # variable substitution
cv.accfinl[c(1, 4:5, 7:8)] # examine accuracies

# final model summary
names(mod.finl$coefficients) # determine which vars were smoothed
vs. linear
summary.glm(mod.finl)$coefficients[c(4:5), ] # access linear

terms

# GAM plots using interactive mode
# vars to plot here are: [2:3,5:8]
par(mfrow = c(2, 3)) # set par
plot(mod.finl, ask = T) # interactive graphs
# first select SE on, then select plots by number

# save plot if desired
#setwd(path.figs)
#savePlot(filename = "mod6fig04.pdf", type = "pdf")

# plot categorical var geonutr ; plot number=3
par(mfrow = c(1, 1)) # reset par
plot(mod.finl, ask = T) # interactive graphs
# first select SE on, then select plots by number

# save plot if desired
#setwd(path.figs)
#savePlot(filename = "mod6fig05.pdf", type = "pdf")

##### START FINAL GAM MODEL OUTPUT

##### START PREDICTION & RECLASSIFICATION OF FINAL GAM MODEL
# ASSUME pred.dom is a raster stack per Module 4.1 & Module 10
modfinl.GAMprob <- predict(spp.predsR, mod.finl,
filename = "modfinl.GAMprob.tif",
type = "response", fun = predict,
index = 2)

#basic reclassify based on threshold mod.cut per above
modfinl.GAMclas = reclassify(modfinl.GAMprob, c(0,mod.cut[[2]],
0,
mod.cut[[2]], 1))
plot(modfinl.GAMclas)
#bins
modfinl.GAMclas.bin <- reclassify(modfinl.GAMprob,c(0,.2,1,
0,.2,.4,2,
plot(modfinl.GAMclas.bin)

modfinl.GAMprob <- as(modfinl.GAMprob, "SpatRaster")
modfinl.GAMclas  <- as(modfinl.GAMclas,  "SpatRaster")
modfinl.GAMclas.bin <- as(modfinl.GAMclas.bin, "SpatRaster")

terra::writeRaster(modfinl.GAMprob, filename =
  "F:/Josh/Plants/spp/Mods/Mod.GAM/mod.GAMprob.tif")
terra::writeRaster(modfinl.GAMclas, filename =
  "F:/Josh/Plants/spp/Mods/Mod.GAM/mod.GAMclas.tif")
terra::writeRaster(modfinl.GAMclas.bin, filename =
  "F:/Josh/Plants/
  spp/Mods/Mod.GAM/mod.GAMclas.bin.tif")

modfinl.GAMprob.R  <- as(modfinl.GAMprob, "Raster")
modfinl.GAMclas.R  <- as(modfinl.GAMclas,  "Raster")
modfinl.GAMclas.bin.R  <- as(modfinl.GAMclas.bin, "Raster")

raster::writeRaster(modfinl.GAMprob.R, filename =
  "F:/Josh/Plants/spp/Mods/Mod.GAM/mod.GAMprob.R.tif")
raster::writeRaster(modfinl.GAMclas.R, filename =
  "F:/Josh/Plants/spp/Mods/Mod.GAM/mod.GAMclas.R.tif")
raster::writeRaster(modfinl.GAMclas.bin.R, filename =
  "F:/Josh/Plants/spp/Mods/Mod.GAM/mod.GAMclas.bin.R.tif")

# libraries
source("https://raw.githubusercontent.com/AMBarbosa/unpackaged/master/predict_bart_df")
devtools::install_github('cjcarlson/embarcadero')
library(embarcadero)
library(dbarts)
library(blockCV)
library(modEvA)
library(raster)
options("rgdal_show_exportToProj4_warnings"="none")
library(rgdal)
library(sf)
library(magrittr)
library(kableExtra)
library(tidyverse)
library(knitr)

# Load preds
spp.preds <-
terra::rast("F:/Josh/Plants/spp/spp_preds/spp.preds.tif")
spp.predsR <- as(spp.preds, "Raster")

# load presabs trained data
load("F:/Josh/Plants/spp/trained_data/spp.mod.vars.RData", verbose = T)
dat1 <- spp.mod.vars; dim(dat1); table(dat1$PA); head(dat1, 2) # examine data

resp <- 1  # presence/absence resp column
preds <- 2:length(names(dat1))  # predictor columns

# BART mod 1
set.seed(12344) # repeatability
date()
mod_bart1 <- bart(y.train = dat1[, resp],
                   x.train = dat1[, preds], keepTrees = TRUE)
date()

# must be run to retain model - why? if not working, clear ls()
invisible(mod_bart1$fit$state)

# save mod_bart1
save(mod_bart1, file =
     "F:/Josh/Plants/spp/Mods/Mod.BART/mod_bart1.RData")
load("F:/Josh/Plants/spp/Mods/Mod.BART/mod_bart1.RData", verbose = T)

dat1$pred_bart1 <- predict_bart_df(mod_bart1, dat1)  # can take
time for large datasets!
head(as.data.frame((cbind(spp_obs = dat1$PA,
                           spp_pred = dat1$pred_bart1))));
table(dat1$pred_bart1)

# prediction map - ~1min
date()
bart_pred <- predict2.bart(mod_bart1, x.layers = spp.predsR)
date()
par(mfrow=c(1,1))
plot(bart_pred)
plot(spp.bbox.SF, add = T, col = "transparent")

# Save
save(bart_pred, file = 
"F:/Josh/Plants/spp/Mods/Mod.BART/bart_predict.RData")
load("F:/Josh/Plants/spp/Mods/Mod.BART/bart_predict.RData",
verbose = T)

# area under the ROC Curve: modEvA package
#?AUC
windows(22,12)
par(mfrow = c(1, 1), mar = c(5, 4, 2, 1))
AUC(obs = dat1[, resp], pred = dat1[, "pred_bart1"])
# modEvA package
savePlot("bart.pred.AUC.jpeg")

# area under the precision-recall Curve:
AUC(obs = dat1[, "PA"], pred = dat1[, "PA"], curve = "PR")
savePlot("bart.prec_recall.AUC.JPEG")

dev.off()

# threshold classification dataframe
data1$clas <- ifelse(data1$pred_bart1 >= prevalence(data1[, "PA"]), 1, 0)
head(as.data.frame(cbind(data1_obs = data1$PA,
                        data1_pred = data1$pred_bart1,
                        data1_clas = data1$clas)))

data1$clasTSS <- ifelse(data1$pred_bart1 >= 0.5110887, 1, 0)

# threshold classification raster plot
table(data1$clas) # prevalence based
table(data1$clasTSS) # prevalence based

# threshold-based validation metrics: modEvA package
# sTSS & skappa standardized to the 0-1 scale for direct comparability
# set standardize = FALSE to turn off
?threshMeasures
windows(22,12)
                 "TSS", "kappa") # labels
threshMeasures(obs = data1[, "PA"], pred = data1[, "pred_bart1"],
               thresh = "preval",
               measures = plot_labels, ylim = c(0, 1),
               standardize = T)
savePlot("thresh.preval.jpeg")
threshMeasures(obs = dat1[, "PA"], pred = dat1[, "pred_bart1"],
    thresh = 0.5110887,
    measures = plot_labels, ylim = c(0, 1),
    standardize = T)
savePlot("thresh.jpeg")

sensitivity <- 0.9902754
specificity <- 0.9292438
AUC <- 0.9859425 #summary(mod_bart1)
tss <- 0.9224366
bart.acc <- cbind.data.frame(sensitivity, specificity, AUC, tss)
save(bart.acc, file = "F:/Josh/Plants/spp/Mods/Mod.BART/bart.acc.RData")

# GET PREDICTIONS ON THE DATA FRAME ####
# (including confidence intervals)

head(dat1)

source("https://raw.githubusercontent.com/AMBarbosa/
    unpackaged/master/predict_bart_df")

# creates 95%CIs about each cell prediction ie the posterior mean
bart_predCI <- predict_bart_df(mod_bart1, dat1, quantiles =
    c(0.025, 0.975))
head(bart_predCI)
bart_predCI$uncertainty <- bart_predCI[, 3] - bart_predCI[, 2]
head(bart_predCI)
save(bart_predCI, file =
    "F:/Josh/Plants/spp/Mods/Mod.BART/bart_predCI.RData")

# GET PREDICTIONS ON RASTER MAPS ####
# (if you have also raster maps of your variables)

date()
#bart_predCI <- predict2.bart(mod_bart1, x.layers = pred.dom,
#    quantiles = c(0.025, 0.975))
bart_predCI1 <- predict2.bart(mod_bart1, x.layers = spp.predsR,
    quantiles = c(0.34, 0.68))  #
date()
bart_predCI  # RasterStack with 3 layers
bart_predCI1  # RasterStack with 3 layers
names(bart_predCI)
names(bart_predCI1)

# calc prediction uncertainty
names(bart_predCI) <- c("predicted", "q0025", "q0975")
names(bart_predCI1) <- c("predicted", "q34", "q68")
bart_predUncert <- bart_predCI[["q0975"]]
bart_predUncert1 <- bart_predCI1[["q34"]]

bart_predUncert <- bart_predCI[["q0025"]]
bart_predUncert1 <- bart_predCI1[["q0025"]]

bart_predUncert1
# giggle map
par(mfrow = c(2, 2), mar = c(3, 2, 2, 2))  # set 2x2 plots per window
plot(bart_predCI["q0025"], main = 'Lower 95% CI bound', zlim = c(0, 1))
plot(bart_predCI["q0975"], main = 'Upper 95% CI bound', zlim = c(0, 1))
plot(st_geometry(spp.modldom), add = T, border = "red")
plot(bart_predCI["predicted"], main = 'Posterior mean', zlim = c(0, 1))
plot(st_geometry(spp.modldom), add = T, border = "red")
plot(bart_predUncert, main = "Credible interval width", zlim = c(0, 1))
plot(st_geometry(spp.modldom), add = T, border = "red")
par(mfrow = c(1, 1))
savePlot("bart_bounds_means_width.jpeg")

# predictions
#bart_pred <- predict2.bart(mod_bart1, x.layers = spp.predsR)
terra::plot(bart_pred)
bart_pred.R <- as(bart_pred, "Raster")
bart_clas.R <- reclassify(bart_pred.R, c(0, 0.8887663, 0, 0.8887663, 1, 1))
bart_clas.bin.R <- reclassify(bart_pred.R, c(0, 0.2, 1, 0.2, 4.2, 4.6, 3, 6.8, 4, 0.8, 1, 5))

bart_pred <- as(bart_pred.R, "SpatRaster")
bart_clas <- as(bart_clas.R, "SpatRaster")
bart_clas.bin <- as(bart_clas.bin.R, "SpatRaster")

terra::writeRaster(bart_pred, filename = "F:/Josh/Plants/spp/Mods/Mod.BART/bart_pred.tif", overwrite = T)
terra::writeRaster(bart_clas, filename = "F:/Josh/Plants/spp/Mods/Mod.BART/bart_clas.tif", overwrite = T)
terra::writeRaster(bart_clas.bin, filename = "F:/Josh/Plants/spp/Mods/Mod.BART/bart_clas.bin.tif", overwrite = T)
raster::writeRaster(bart_pred.R, filename = "F:/Josh/Plants/spp/Mods/Mod.BART/bart_pred.R.tif", overwrite = T)
raster::writeRaster(bart_clas.R, filename = "F:/Josh/Plants/spp/Mods/Mod.BART/bart_clas.R.tif", overwrite = T)
raster::writeRaster(bart_clas.bin.R, filename = "F:/Josh/Plants/spp/Mods/Mod.BART/bart_clas.bin.R.tif", overwrite = T)
### Ensemble Models

```r
load("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RDATA", verbose = T)

# Load Prob
MAXprob <- terra::rast("F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.MAXprob.tif")
RFprob <- terra::rast("F:/Josh/Plants/spp/Mods/Mod.RF/mod1.RFprob.tif")
LRprob <- terra::rast("F:/Josh/Plants/spp/Mods/Mod.LR/mod2.LRprob.tif")
BRTprob <- terra::rast("F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.BRTprob.tif")
BARTprob <- terra::rast("F:/Josh/Plants/spp/Mods/Mod.BART/bart_pred.tif")

# Prob dom
prob_dom <- c(LRprob, MAXprob, BRTprob, RFprob, BARTprob);
prob_dom

# Load Clas
MAXclas <- terra::rast("F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.MAXclas.tif")
RFclas <- terra::rast("F:/Josh/Plants/spp/Mods/Mod.RF/mod1.RFclas.tif")
LRclas <- terra::rast("F:/Josh/Plants/spp/Mods/Mod.LR/mod.LRclas.tif")
BRTclas <- terra::rast("F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.BRTclas.tif")
BARTclas <- terra::rast("F:/Josh/Plants/spp/Mods/Mod.BART/bart_clas.tif")

# clas_dom
clas_dom <- c(LRclas, MAXclas, BRTclas, RFclas, BARTclas);
clas_dom

# ensemble prob
par(mfrow=c(1,1))
prob_mean <- terra::mean(prob_dom)
plot(prob_mean, main = "Mean Probability")
plot(spp.bbox SF, add = T)

# ensemble clas
clas_sum <- sum(clas_dom)
```
plot(clas_sum, main = "Concordance")
plot(spp.bbox.SF$geometry, add = T)

mean_probR <- as(prob_mean, "Raster")

# ensemble bin
bin_sum <- raster::reclassify(mean_probR,c(0,.2,1,
              .2,.4,2,
              .4,.6,3,
              .6,.8,4,
              0.8,1,5))

bin_sum <- as(bin_sum, "SpatRaster")
plot(bin_sum, main = "Concordance Bin")
plot(spp.bbox.SF$geometry, add = T)

# save
terra::writeRaster(prob_mean, filename =
                  "F:/Josh/Plants/spp/Mods/Mean/
                  prob_mean.tif", overwrite = T)
terra::writeRaster(clas_sum, filename =
                   "F:/Josh/Plants/spp/Mods/Mean/
                   clas_mean.tif", overwrite = T)
terra::writeRaster(bin_sum, filename =
                   "F:/Josh/Plants/spp/Mods/Mean/
                   bin_mean.tif", overwrite = T)

prob_mean.R <- as(prob_mean, "Raster")
clas_sum.R <- as(clas_sum, "Raster")
bin_sum.R <- as(bin_sum, "Raster")

raster::writeRaster(prob_mean.R, filename =
                    "F:/Josh/Plants/spp/Mods/Mean/
                    prob_mean.R.tif", overwrite = T)
raster::writeRaster(clas_sum.R, filename =
                    "F:/Josh/Plants/spp/Mods/Mean/
                    clas_mean.R.tif", overwrite = T)
raster::writeRaster(bin_sum.R, filename =
                    "F:/Josh/Plants/spp/Mods/Mean/
                    bin_mean.R.tif", overwrite = T)

#################################################################
############ All Models Plot
###########################################

MAXprob <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.MAXprob.tif")
RFprob <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.RF/mod1.RFprob.tif")
LRprob <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.LR/mod2.LRprob.tif")
BRTprob <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.BRTprob.tif")
#GAMprob <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.GAM/mod.GAMprob.tif")
BARTprob <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.BART/bart_pred.tif")

MAXclas <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.MAXclas.tif")
RFclas <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.RF/mod1.RFclas.tif")
LRclas <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.LR/mod.LRclas.tif")
BRTclas <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.BRTclas.tif")
#GAMclas <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.GAM/mod.GAMclas.tif")
BARTclas <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.BART/bart_clas.tif")

MAXbin <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.MAXclas.bin.tif")
RFbin <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.RF/mod1.RFclas.bin.tif")
LRbin <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.LR/mod.LRclas.bin.tif")
BRTbin <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.BRTclas.bin.tif")
#GAMbin <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.GAM/mod.GAMclas.bin.tif")
BARTbin <-
terra::rast("F:/Josh/Plants/spp/Mods/Mod.BART/bart_clas.bin.tif")

load("F:/Josh/Plants/spp/frames/bbox/spp.bbox.RDATA", verbose = T)
windows(22,12)
par(mfrow=c(3,5))
plot(LRprob, main = "Logistic Regression")
plot(spp.bbox.SF, add = T, col = "transparent")
#plot(GAMprob, main = "Generalized Additive")
#plot(spp.bbox.SF, add = T)
plot(MAXprob, main = "MAXENT")
plot(spp.bbox.SF, add = T, col = "transparent")
plot(BRTprob, main = "Boosted Regression Trees")
plot(spp.bbox.SF, add = T, col = "transparent")
plot(RFprob, main = "Random Forest")
plot(spp.bbox.SF, add = T, col = "transparent")
plot(BARTprob, main = "Bayesian Additive Regression")
plot(spp.bbox.SF, add = T, col = "transparent")
plot(LRclas)
plot(spp.bbox.SF, add = T, col = "transparent")
#plot(GAMclas)
#plot(spp.bbox.SF, add = T)
plot(MAXclas)
plot(spp.bbox.SF, add = T, col = "transparent")
plot(BRTclas)
plot(spp.bbox.SF, add = T, col = "transparent")
plot(RFclas)
plot(spp.bbox.SF, add = T, col = "transparent")
plot(BARTclas)
plot(spp.bbox.SF, add = T, col = "transparent")
plot(LRbin)
plot(spp.bbox.SF, add = T, col = "transparent")
#plot(GAMbin)
#plot(spp.bbox.SF, add = T)
plot(MAXbin)
plot(spp.bbox.SF, add = T, col = "transparent")
plot(BRTbin)
plot(spp.bbox.SF, add = T, col = "transparent")
plot(RFbin)
plot(spp.bbox.SF, add = T, col = "transparent")
plot(BARTbin)
plot(spp.bbox.SF, add = T, col = "transparent")

savePlot("F:/Josh/Plants/spp/Mods/all_spp.jpeg")

#################################################################
########## Table of Mod info
###########################################
#################################################################

# LR
list.files("F:/Josh/Plants/spp/Mods/Mod.LR")
load("F:/Josh/Plants/spp/Mods/Mod.LR/mod2.acc.LR.RData", verbose = T)
LR_info <- mod2.acc[c(4:5, 7:8)]
model <- "Logistic Regression"
LR_info <- cbind(model, LR_info)

# GAM
#list.files("F:/Josh/Plants/spp/Mods/Mod.GAM")
#load("F:/Josh/Plants/spp/Mods/Mod.GAM/mods.acc.RData", verbose = T)
#GAM_info <- mods.acc[3, ]
#GAM_info[4:5, 7:8]

# MAXENT
list.files("F:/Josh/Plants/spp/Mods/Mod.MAX")
load("F:/Josh/Plants/spp/Mods/Mod.MAX/mod2.acc.RData", verbose = T)
MAX_info <- mod2.acc[c(4:5, 7:8)]
model <- "Maxent"
MAX_info <- cbind(model, MAX_info)
MAX_info <- MAX_info[1,]

# RF

list.files("F:/Josh/Plants/spp/Mods/Mod.RF")
load("F:/Josh/Plants/spp/Mods/Mod.RF/mod1.acc.RData", verbose = T)
RF_info <- mod1.acc[c(4:5, 7:8)]
model <- "Random Forest"
RF_info <- cbind(model, RF_info)

# BRT

list.files("F:/Josh/Plants/spp/Mods/Mod.BRT")
load("F:/Josh/Plants/spp/Mods/Mod.BRT/mod2.acc.RData", verbose = T)
BRT_info <- mod2.acc[2, ]
BRT_info <- BRT_info[c(4:5, 7:8)]
model <- "Boosted Regression Trees"
BRT_info <- cbind(model, BRT_info)

# BAYES

list.files("F:/Josh/Plants/spp/Mods/Mod.BART")
load("F:/Josh/Plants/spp/Mods/Mod.BART/bart.acc.RData", verbose = T)
BART_info <- bart.acc
model <- "Bayesian Additive Regression Trees"
BART_info <- cbind(model, BART_info)
pander::pander(BART_info)

# all_stats
all_stats <- rbind.data.frame(LR_info, MAX_info, RF_info, BRT_info, BART_info)

# save stats
write.csv(all_stats, file = "F:/Josh/Plants/spp/mod_stats/all_stats.csv")
all_stats <- read.csv("F:/Josh/Plants/spp/mod_stats/all_stats.csv")
pander::pander(all_stats)