Predicting Range Expansion of Yellowstone Grizzly Bears

Jacob Dym

Department of Mathematical Sciences
Montana State University

May 3, 2018

A writing project submitted in partial fulfillment
of the requirements for the degree

Master of Science in Statistics

APPROVAL

of a writing project submitted by

Jacob Dym

This writing project has been read by the writing project advisor and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Statistics Faculty.

Date	First Last Andrew Hoegh
Date	Mark C. Greenwood Writing Projects Coordinator

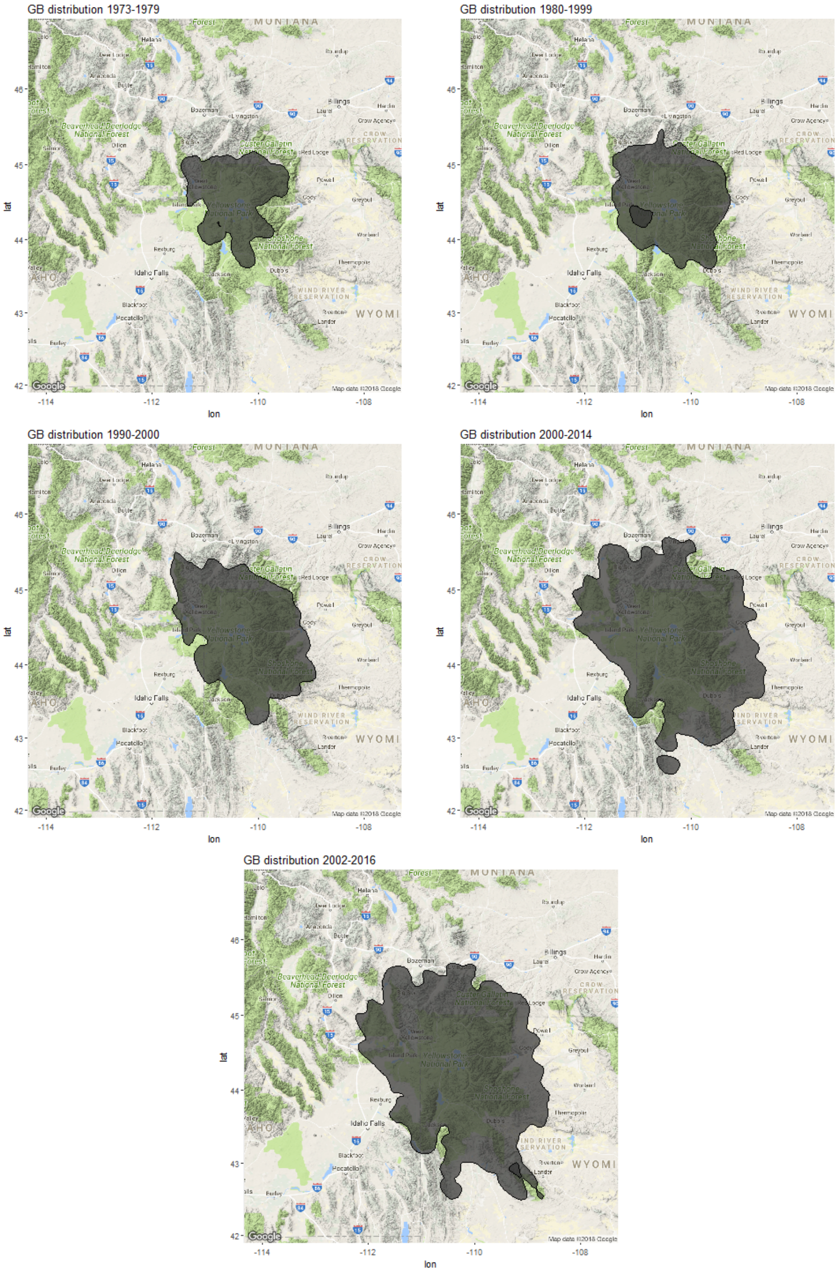
Contents

Abstract

The grizzly bear population in the Greater Yellowstone Ecosystem (GYE) has increased in size and range extent over the past several decades. As the grizzly bear population continues to expand the range will also spread further outside the GYE. The goal was to predict areas that bears will diffuse into as the population increases and the grizzly population range grows. To make predictions of plausible range expansion covariate information was extracted from map files of the GYE. Range expansion predictions were made utilizing the step selection function (SSF) to construct correlated random walks for simulated grizzly bears. The correlated random walks are based on how attractive areas might be to a grizzly bear as they move around, as well as the movement behavior of bears from the GYE.

1 Introduction

The grizzly bear population in the Greater Yellowstone Ecosystem (GYE) has increased over the past several decades along with the range extent of the bears [Peck et al., 2017]. The distribution of the grizzly bear population in the GYE and the populations range expansion is shown in Figure 1 for 1973-2016 from the Inter Agency Grizzly Bear Study Team 2018. Understanding the future expansion of the bear population in the GYE is important for addressing habitat needs for bears. According to Schwartz 2006, “Understanding the current distribution of bears within the GYE is a required part of the delisting process, but is also useful to the US Forest Service and their efforts to address habitat needs for the grizzly bear under their new Forest Management Plans”. Knowing the spatial range of the bear population is beneficial for managing the bear population as noted by the National Park Service (N.P.S.) 2017, “while grizzly bears are no longer on the threatened species list, bear management is an important part of park management in Yellowstone and helps reduce the number of bear-caused human injuries”.



GB distribution 1973-1979

GB distribution 1980-1999

GB distribution 1990-2000

GB distribution 2000-2014

GB distribution 2002-2016

Figure 1: Estimated grizzly bear distribution in the GYE from 1973-2016. Units are in latitude and longitude, where Easting corresponds to Longitude and Northing corresponds to Latitude

Studying animal movements from radio tagging has been common among wildlife biologists for decades, though telemetry data collection has improved greatly over the past 40 years with the introduction of GPS based movement data as noted by Millspaugh and Marzluff [2009]. GPS telemetry data for a single animal records the animals location at certain time intervals, and can either be stored on the device or sent to a receiver. The telemetry data for this study comes from the internally stored memory of GPS devices. For this study, devices were attached to the bears using procedures approved by Animal Care and Use Committees (ACUC), usually by trapping the bears and attaching collars [Peck et al., 2017]. The on-board data must be downloaded from the device by either re-capturing the bears or retrieving the data after the GPS has been dislodged from a bear.

There are a variety of models for animal movements based on telemetry data, one commonly used modeling approach are resource selection functions (RSF) where inferences are focused on choices individuals make given the type of environment that is available to them. Other models for analyzing telemetry data include spatio-temporal point process models Hooten et al. [2017], where intensity is a function of location and time. Random walks utilize inferences from resource selection function but focus on the physical process of movement; such models are often referred to as "state-space models" where the data are modeled conditioned on a latent process [Hooten et al., 2017].

The goal of this paper is to predict regions bears will explore as the grizzly bear population of the GYE grows and the distribution of bears expands beyond the existing borders of the GYE habitat. Telemetry data of bears in the GYE and environmental factors biologically relevant for grizzly bears such as human presence, roads and highways, water availability, as well as other measurable information are used to simulate grizzly bear range expansion.

The remaining sections of this paper continue with Section 2, an overview of the data used in this study, section 3 details the methods used for modeling animal movements, section 4 summarizes a simulation study which explores the effectiveness of resource selection functions, section 5 is an outline of the simulation study performed, and section 6 is a summary of the study conclusions.

2 Data

The covariates used for predicting bear movements and the underlying data on bear locations are described in Peck et al. [2017]. Grizzly bear movements came from GPS telemetry data from 116 male bear in the GYE from May 2000 to October 20015 in UTM coordinates. The telemetry data for movements was restricted to data from the on-board memory of the GPS transmitters as it was more accurate than some of the remotely downloaded GPS data, the sample time intervals between steps were between 3.5 to 4 hours. Modeling habitat selection based on active movements was the goal of the analysis so the data was constrained to steps that were 100m apart or more similarly readings that were consecutively less than 100m apart were also excluded.

Several raster files of the GYE were included to establish the covariate values of grids in the GYE. The raster files were constructed as grids such that each location was a 300m x 300m grid cell. The raster files were measure on UTM coordinates and ranged from 50179.43 to 814579.4 Easting, and 4602855 to 5439855 Northing from the GYE to the NCDE creating an area of grid cells by 2790 x 2548 for 7,108,920 grid cells.

Covariates recorded in the raster files included the Euclidean distance to the nearest forest edge for a grid cell. The distance to roads as well as the density of a road were also included as grizzly bears generally avoid roads. The distance to rivers and perennial streams was included as a single covariate. The log transformed density of housing per square block was included as to account for human presence. Elevation was included as a linear and quadratic term as animal movement behavior changes vary depending on the aspect and slope of areas. A vegetation covariate based on the relative greenness of areas based on Google earth images was recorded as a normalized vegetation density index (NVDI), and a measure of natural land cover was included.

The information from the raster files was combined into a covariate matrix, observed locations from the telemetry data were associated with grid cells using the extract function from the "raster" package in R, locations were determined to be in a grid cell if the UTM coordinates fall within the boundaries of that cell. The SSF was used to generate a weight matrix for the GYE grids, where grid cells with higher weights had a higher relative probability of being visited by a bear. A raster map of the estimated weights can be seen in figure 2.

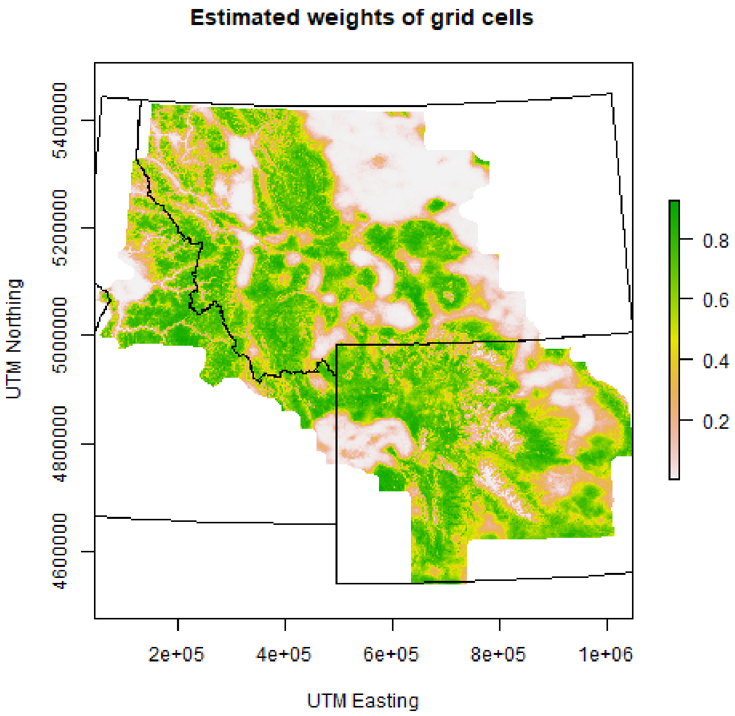


Figure 2: Plot of grid cell weights from the GYE to the NCDE in UTM coordinates, regions with larger weights are more likely to be chosen by a bear when making movements

From the estimated weight matrix a conductance matrix was constructed as a possible alternative method for predicting grizzly bear range expansion. The weight matrix as well as a conductance matrix covered the geographic region from the GYE to the Northern Continental Divide Ecosystem (NCDE). The conductance matrix was constructed from the estimated transition weights, and were transformed into a transition object using the "gdistance" package. The conductance matrix contains information from the GYE and NCDE as well as the conductance values for moving from one cell to its potentially 8 nearest neighbors defined using the kings connectivity. The conductance layer was initially structured as a transition layer and only contained non symmetric transition probabilities from one grid cell to one of its neighbors for 31,397,556 transition values.

3 Methods

To make predictions of grizzly bear range expansion correlated random walks were used similar to the methods described by Clark et al. [2015]. The framework for utilizing correlated random walks requires weights for every grid cell within the study region, as well as the general characteristics of how bears make movements. Bear movement behavior for the SSF consists of the step length a bear might take as well as the trajectory a bear will likely have based on previous movements. The length of a step bears make and the associated trajectory were found to be uncorrelated based on the linear-circular correlation coefficient [Peck et al., 2017], so bear movements consist of a distance and trajectory and are evaluated on what grid cells are proposed by the algorithm (illustration of step proposal below in Figure 3).

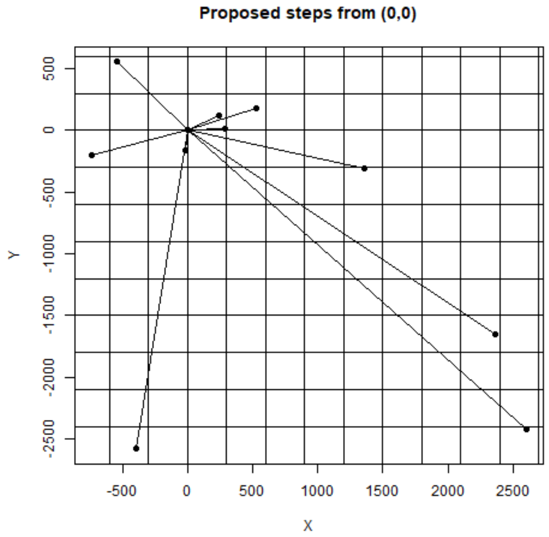


Figure 3: Proposed steps a bear could make from the point of origin (0,0) based on the parameters of the SSF (trajectories bears are likely to have, how far bears are likely to step)

To establish the framework for random walks, parameter estimates of bear movements are estimated using a SSF on the telemetry data and associated environmental covariates [Fortin et al., 2005]. Coefficients for generating the weight matrix are then obtained using conditional logistic regression [Clark et al., 2015], with the weight matrix and estimates of bear movement behavior, which will be detailed in section 3.2.

3.1 The Step Selection Function

The SSF used as defined by Fortin et al. [2005] analyzes animal telemetry data by decomposing movements into step lengths and turning angles of steps. Angles and step lengths can be modeled by two separate distributions so long as turn angles and step lengths are independent of one another. Step lengths and turn angles can be extracted from telemetry data by looking at where an animal was and where they appeared next in the the data set. After the step length and turning angles are extracted from the data, the parameters of the step length and turning angle distribution can be estimated in a variety of different ways. The linear-circular correlation coefficient between step length and turning angle determines if angles and lengths are reasonably independent, this determines if available steps can be simulated by sampling step length independently from turning angle.

For each bear the distribution of step lengths was modeled by a scaled beta distribution, where the step lengths were standardized to an interval of [0,1] from the observed range of step lengths (100 - 15,000m). Once all of the step lengths were extracted for each bear the beta distribution parameters were estimated using maximum-likelihhod estimation (MLE) [Peck et al., 2017]. The beta distribution can be written as: $Beta(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1}$, where $\mu \in [0,1]$ and $\alpha\,, \beta > 0$

For each bear the distribution of turning angles was approximated by a von Mises distribution, where turning angles were calculated relative to the previous step taken by an animal as shown below in Figure 4. The parameters for the von Mises distribution were estimated using MLE as well [Peck et al., 2017]. The von Mises distribution can be written as, $p(\theta|\theta_0, m) = \frac{1}{2\pi I_0(m)} e^{m\, cos(\theta-\theta_0)}$, where $I_0 = \frac{1}{2\pi}\int_0^{2\pi} e^{m\, cos(\theta)} d\theta$, θ_0 is the mean direction, and m is the measure of concentration. Turn angles and step lengths can be sampled jointly, in Peck et al. [2017] the linear-circulation coefficient between the two was found to be sufficiently small such that the turn angles and step lengths could be sampled independently of one another. Bayesian methods were explored to obtain parameter estimates, however in the simulation study parameter estimates were nearly identical under both methods so the less computationally intense MLEs were used.

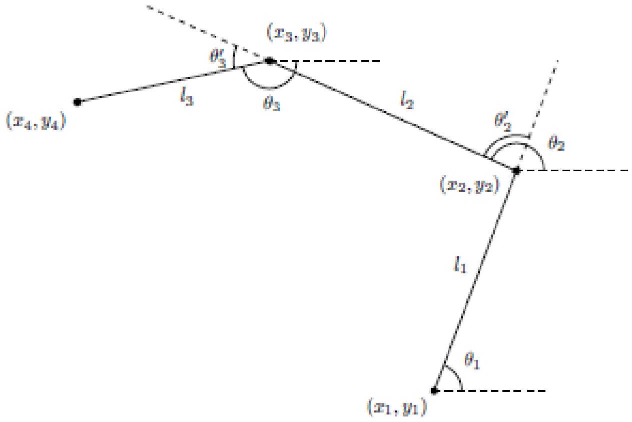


Figure 4: Diagram of step turn angle calculations

3.2 Conditional logistic regression

As described by Gillies et al. [2006] weights for grid cells are defined as, $w(x) = \exp(\beta_1 x_1 + \cdots + \beta_n x_n)$. Estimating coefficients is done using logistic regression conditioned on an animals current location. For each pair of consecutive points in the telemetry data set $m - 1$, m, a set of k random points are generated from the SSF starting at $m - 1$ for a total of $m(1 + k)$ observations. Each step is given an identifier for the observed animal movement and the associated k generated points. Logistic regression is then performed on the $m(1 + k)$ observations with a random intercept included for each set of $m + 1$ locations, the model for a bear at location $i \in 1, \ldots, m$ with n covariates is, $\alpha_i + \beta_1 x_1 + \cdots + \beta_n$ [Gillies et al., 2006].

Each step a bear made in the telemetry data is paired with k steps generated from the same point of origin the bear was previously at similar to the methods used in Fortin et al. [2005]. The covariates from the randomly generated steps would be associated with a location a bear could have moved to but did not. This procedure is repeated for each grizzly bear such that every bear will have its own estimated coefficients associated with habitat selection. To obtain population coefficients the coefficients from all bears are averaged $\hat{\beta}_1 = \frac{1}{m} \sum_{i=1}^{m} \hat{\beta}_{i1}$, where $i = 1, \ldots, m$ represents the observed set of bears.

Once the coefficient estimates were obtained the weight estimates were calculated as, $\hat{w}(x) = \exp(\hat{\beta}_1 x_1 + \cdots + \hat{\beta}_n x_n)$. To obtain a probabilistic interpretation of the weights, the weights are transformed using the inverse logit function. The weights must be shifted before using the inverse logit function as all the weights will be positive and the inverse logit function maps real values from $(-\infty, \infty)$ to $[0, 1]$. To transform the weights onto a reasonable probabilistic scale the median weight is subtracted from all estimated weights and then transformed using inverse logit function [Peck et al., 2017].

3.3 Correlated random walks

Bear movements were simulated using correlated random walks similar to those used by Clark et al. [2015]. Origin points of bears are randomly generated from a Uniform distribution in of the center region of the GYE where bears are known to live. Proposal steps are generated for a bears potential movement based on the parameters found in the SSF, turning angles and step lengths are generated independently of one another as the two were found to be uncorrelated. Each proposal step a bear can make is evaluated by observing the weight of the cell the proposal step is in, of the n proposals steps one is sampled randomly from all available proposals weighted by their respective grid cell probabilities. The accuracy of these methods was explored in a simulation study.

4 Simulation Study

A simulation study was conducted to evaluate the effectiveness of the SSF for estimating the true probability matrix for animal movements. To start terrain was generated with 3 features that would drive animal movement; elevation, forest density, and the presence/absence of water. A grid was constructed being 10,000 x 10,000 units, where grid cells were 100 x 100 units. Terrain for the grid was constructed such that the left region would be much more desirable for occupancy than the right region, terrain features are visually shown in Figure 5 below. Regions with higher forest density were coded to be more desirable as were grids with available water, higher elevations were coded to be less desirable. The region boundaries are constructed to be impassable, that is movements can only take place inside the boundaries of the generated terrain and have probability 0 ending outside the region.

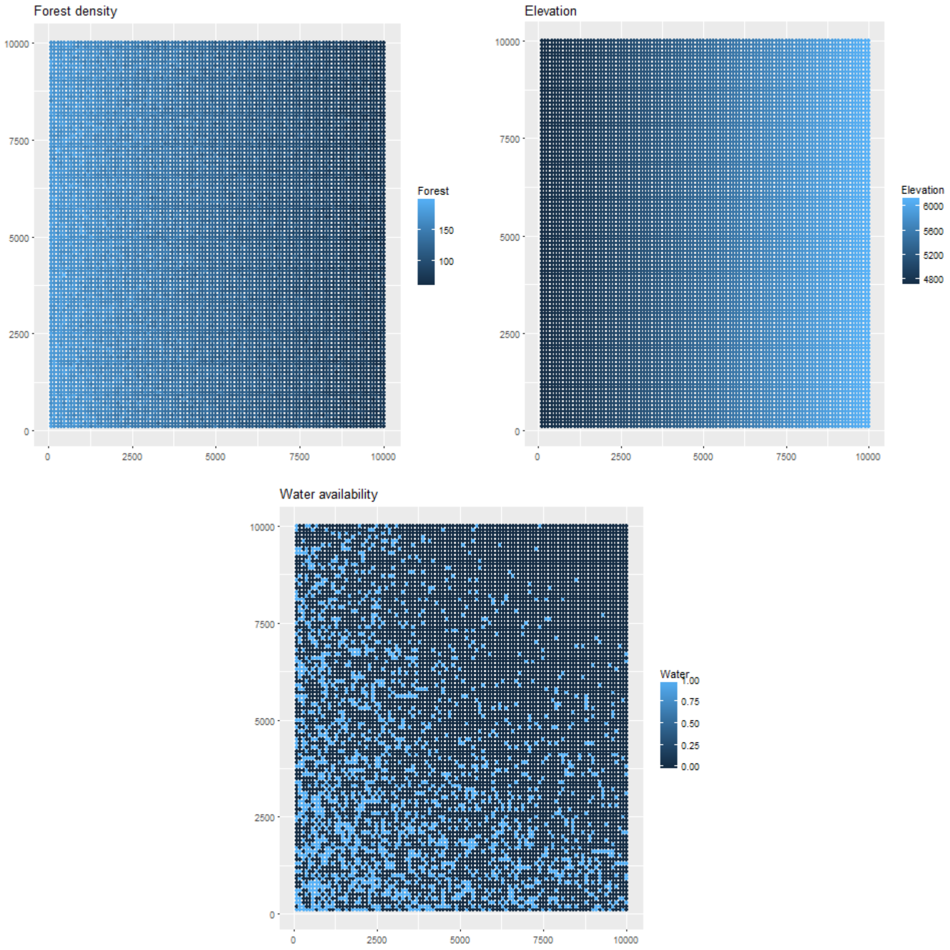


Figure 5: Simulated terrain features of forest density, elevation, and the presence/absence of water for grid cells

Bear movements analogous to the original telemetry data were simulated using a correlated random walks with movement decisions driven by the "true" probabilities generated from the terrain. 25 bears in total were simulated with each bear taking 3,000 steps. Bear step lengths were drawn from a scaled beta distribution were the maximum length of any step could be 400 units. Turn angles of bears were sampled independently from step lengths, where the true distribution of turn angles were drawn from a von Mises distribution. For a given given step 10 proposed steps were created using the population turn angle distribution and step length distribution, a step was then selected based on the probability of the grid cell the step ended in. The first step for each bear is randomly sampled from a uniform distribution from the center region of the grid within the range of (4500, 5500), each initial step is given a initial trajectory sampled from a uniform distribution on [0,2π]. Proposed steps that end outside the simulated study region were assigned a probability of 0, and the algorithm ensured at least 2 steps ended within the study range. The 3,000 steps of two randomly selected simulated bears took are displayed below in Figure 6.

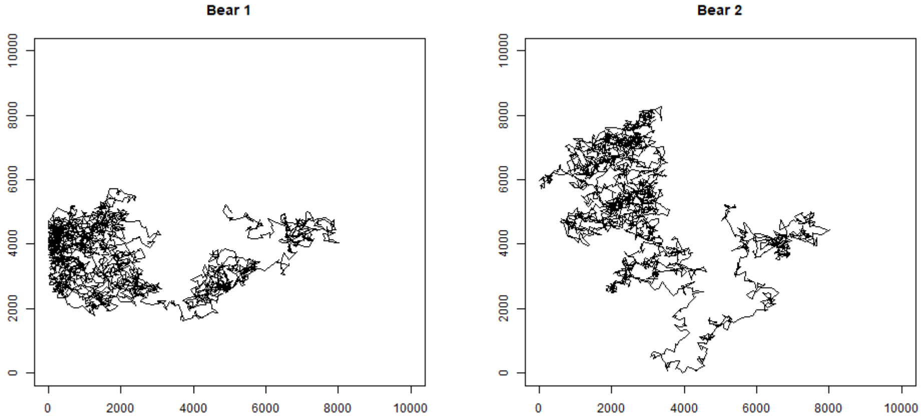


Figure 6: 3,000 simulated steps of Bear 1 and Bear2

Once the movement data was generated step lengths and turning angles were extracted, and parameter estimates for step length and turn angle were estimated using MLE with a scaled beta distribution for step length and a von Mises distribution for turn angles of the observed steps. The average estimates of the von Mises distribution parameters were $\bar{\hat{\mu}} = 3.06177833$ and $\bar{\hat{k}} = 0.03026097$, compared to the parameters of the underlying distribution $\mu = 0$ and $k = 0.3$ (as k approaches 0 the von Mises distribution becomes uniform on [0,2π] so the mean discrepancy is reasonable). The average estimates of the beta distribution parameters were $\bar{\hat{\alpha}} = 1.144596$ and $\bar{\hat{\beta}} = 2.690521$, compared to the parameters of the underlying distribution $\alpha = 1.2$ and $\beta = 3$.

Once the step and turning angle parameters were estimated from the simulated data the weights of grid cells were calculated using conditional logistic regression with the clogit function from the survival package; where 10 randomly generated steps based on the estimated parameters were generated and grouped with the actual movement a simulated bear made. The slope coefficients were averaged across all bears in order to get an estimation for the population as outlined in Peck et al. [2017]. The SSF and methods for estimating grid cell weights worked well as shown in the estimated probability plot and actual probability plot shown below in Figure 7, the sum of absolute errors between the estimated probability for a grid cell and the true probabilities was 605.8502 (for 10,000 grid cells estimates were off by 0.060585 on average).

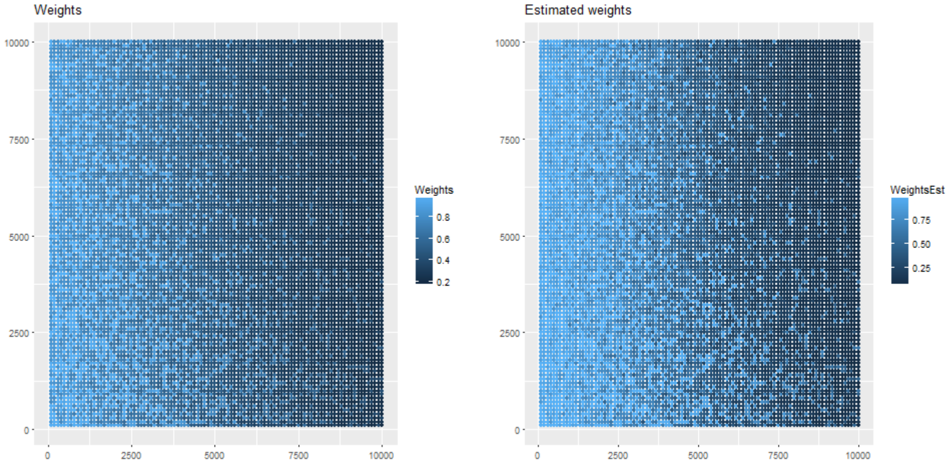


Figure 7: Original probability plot (left), estimated probability plot (right) of bear movements

5 Data Analysis

Correlated random walks were simulated for 5,000 bears each taking 15,000 steps or roughly 5 active seasons. Each bear is started within the estimated grizzly bear distribution from 1973-1979. Simulating the correlated random walks was somewhat computationally expensive running approximately 2 hours for 30,000,000 steps on 7 threads of an Intel i7-7700HQ 2.80Ghz processor. Walks were run in parallel using the foreach function via the doParallel package. 3,000 steps correspond roughly to one year of movements or active season for grizzly bears as suggested by conversations with the inter-agency grizzly bear study team (IGBST), so 15,000 steps for 5,000 bears roughly corresponds to to 5,000 bears actively exploring for the study region for 5 years. A sample of points from the first 3,000 steps and from all 15,000 steps are displayed in Figure 10 .

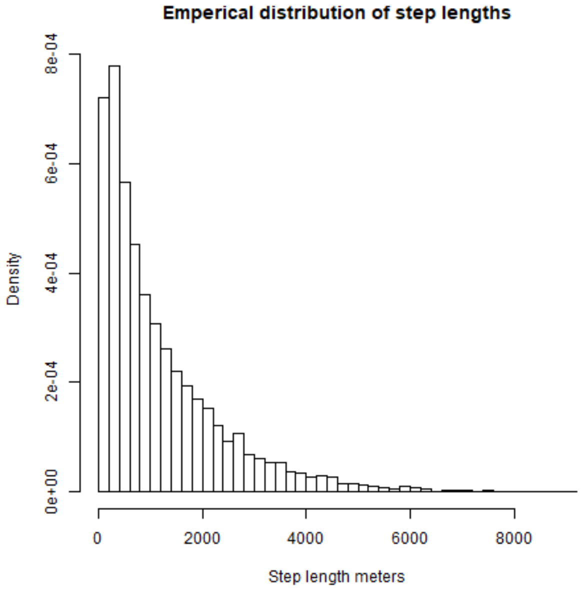


Figure 8: Empirical distribution of step lengths used in correlated random walks

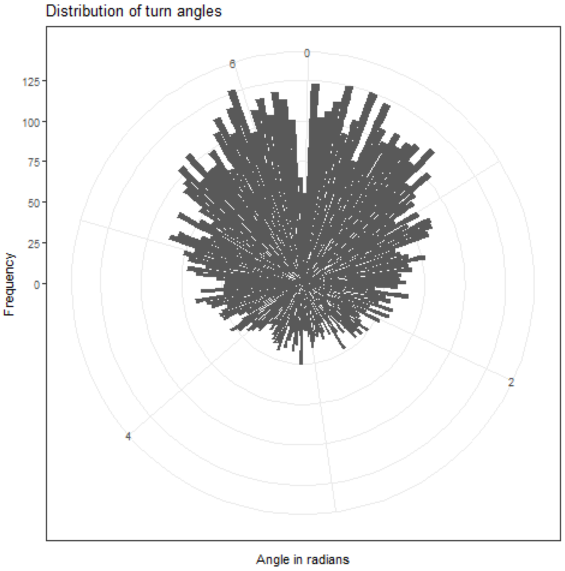


Figure 9: Empirical distribution of turning angles used in correlated random walks

The correlated random walks were run with varying number of bears, different starting locations, and varying number of walk lengths. Generally as the number of bears increased and the length of walks increased the study region exploration densities became similar to the estimated weights, with higher densities in the starting regions of the walks. As walk lengths increased past 9,000 steps the boundaries of the study region became more apparent as there was an imposed boundary the bears could not cross. Within 5 years the 5,000 simulated bears moved from the 1970's grizzly bear distribution estimate to occupy roughly the same area if not more as the most recent distribution estimates. The correlated random walks are based on the SSF of more active bears, so this is result is not unreasonable. To analyze the relative frequency areas were explored with, the number of steps per grid cell were calculated using the rasterize function from the "raster" package.

6 Discussion

There is an assumption that bears will constantly explore each season with behavior and preferences based on the SSF and grid cells estimated by Peck et al. [2017]. There is an assumption that no other factors other than the ones included in the model impact movement decisions in the correlated random walks, bears are also assumed to explore terrain independently and identically to one another.

The modeling assumptions are fairly strong and are likely not reasonable for a typical grizzly bears movements, as the model assumes that bears can explore terrain without impediments. While 3,000 steps in the correlated random walks will likely cover more terrain than a bear would in a given year, it does still provide an insight to the ways that grizzly bear distributions could expand in the future.

Most of the correlated random walks took place within the GYE and grizzly bear distributions estimated by the USGS [2018]. Boundary conditions became an issue as walk lengths were allowed to run longer than 6,000 steps (for the Western and Eastern borders), however walks did not reach the region boundaries for the Northern map borders.

The correlated random walks were run with varying number of bears, different starting locations, and varying number of walk lengths. Generally as the number of bears increased and the length of walks increased the study region exploration densities became similar to the estimated weights, with higher densities in the starting regions of the walks. As walk lengths increased past 9,000 steps the boundaries of the study region became more apparent as there was an imposed boundary the bears could not cross.

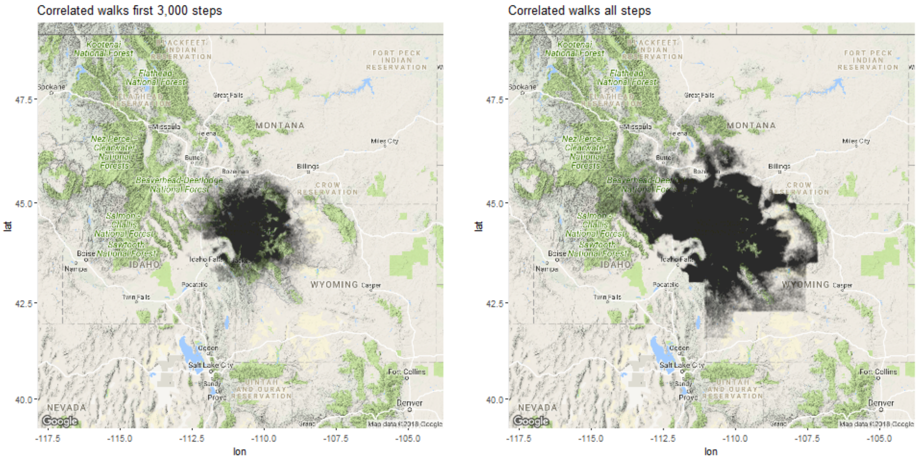


Figure 10: Locations simulated bears were in for first 3,000 steps (left), locations simulated bears were in for 15,000 steps

Correlated random walks do provide an approximation of how the grizzly bear population in the GYE will expand in the near future, though there other methods that could be used to predict future bear range expansion. More complex behaviors could be incorporated into a data driven model such as; active and inactive behaviors in bear movements via a Markov switching model, accounting for the proximity to other bears in a different modeling framework or even in the correlated random walks, or trying to model seasonal effect such as hibernation. Modeling seasonal behavior would also require step calibrations for each season, as bears would be expected to move much less in the winter than in the summer. Whatever method may be used a larger spatial range of covariates should be implemented as well in order to minimize any potential edge effects in the modeling phase.

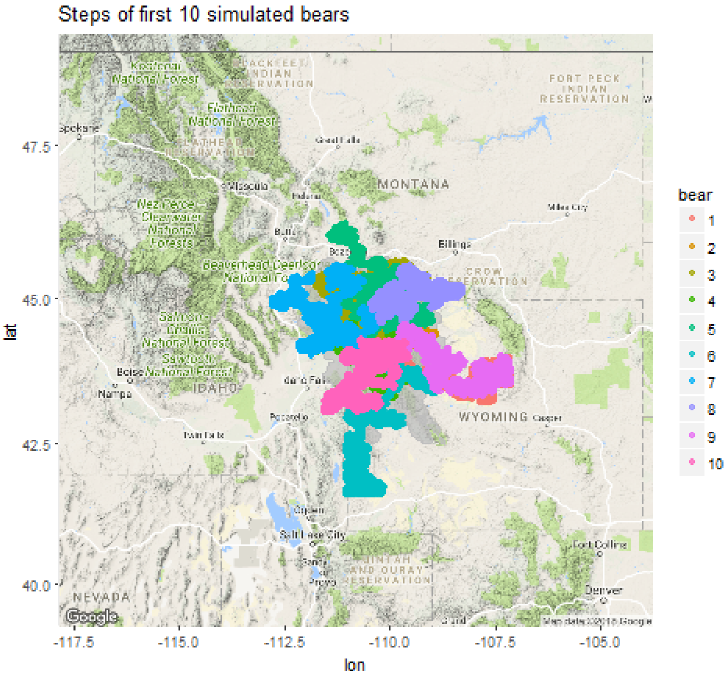


Figure 11: Correlated random walks of first 10 simulated bears

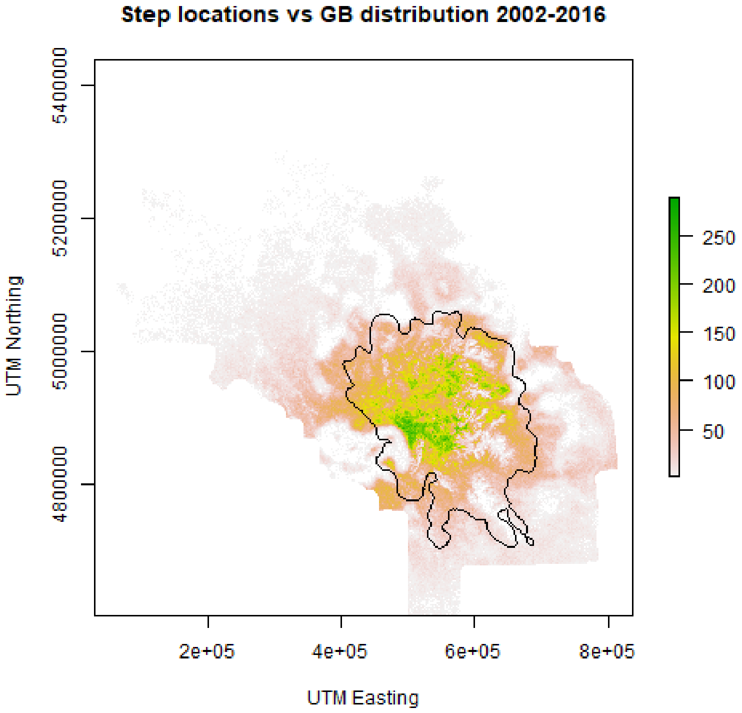


Figure 12: Step locations of correlated random walks compared to the estimated grizzly bear distribution of 2002-2016

References

Joseph D. Clark, Jared S. Laufenberg, Maria Davidson, and Jennifer L. Murrow. Connectivity among subpopulations of Louisiana black bears as estimated by a step selection function. *The Journal of Wildlife Management*, 79(8):1347–1360, 2015. ISSN 1937-2817. doi: 10.1002/jwmg.955. URL `http://dx.doi.org/10.1002/jwmg.955`.

Daniel Fortin, Hawthorne L. Beyer, Mark S. Boyce, Douglas W. Smith, Thierry Duchesne, and Julie S. Mao. Wolves influence elk movements: Behavior shapes a trophic cascade in Yellowstone National Park. *Ecology*, 86(5):1320–1330, 2005. ISSN 1939-9170. doi: 10.1890/04-0953. URL `http://dx.doi.org/10.1890/04-0953`.

Cameron S. Gillies, Mark Hebblewhite, Scott E. Nielsen, Meg A. Krawchuk, Cameron L. Aldridge, Jacqueline L. Frair, D. Joanne Saher, Cameron E. Stevens, and Christopher L. Jerde. Application of random effects to the study of resource selection by animals. *Journal of Animal Ecology*, 75(4): 887–898, 2006. doi: 10.1111/j.1365-2656.2006.01106.x.

Mevin B. Hooten, Devin S. Johnson, Brett T. McClintock, and Juan M. Morales. *Animal Movement Statistical Models for Telemetry Data*. CRC Press, 2017.

Joshua J. Millspaugh and John M. Marzluff. *Radio tracking and animal populations*. Academic Press, 2009.

Christopher P. Peck, Frank T. van Manen, Cecily M. Costello, Mark A. Haroldson, Lisa A. Landenburger, Lori L. Roberts, Daniel D. Bjornlie, and Richard D. Mace. Potential paths for male-mediated gene flow to and from an isolated grizzly bear population. *Ecosphere*, 8(10): e01969–n/a, 2017. ISSN 2150-8925. doi: 10.1002/ecs2.1969. URL `http://dx.doi.org/10.1002/ecs2.1969`. e01969.

USGS. Grizzly bear boundary layers. `https://www.sciencebase.gov/catalog/folder/52fe7f75e4b0354fef6de4f0`, Mar 2018.

7 Appendix-1: R Code

```
# Load packages
library(sf)
library(raster)
library(gdistance)
library(fields)
library(circular)
library(ff)
library(doParallel)
library(boot)
library(rgdal)
library(MASS)
library(parallel)
library(snowfall)

# Construct weight matrix
setwd("C:/Users/jacob/Documents/Statistics/Writing project/Grizzly data/
tif and coeffs")
d2ForestEdge300_m <- raster(paste0(getwd(),"/d2ForestEdge300_m.tif"))
d2Hwy300_m <- raster(paste0(getwd(),"/d2Hwy300_m.tif"))
RoadHwyDen300_mw1500 <- raster(paste0(getwd(),"/RoadHwyDen300_mw1500.tif"))
d2River300_m <- raster(paste0(getwd(),"/d2River300_m.tif"))
d2Stream300_m <- raster(paste0(getwd(),"/d2Stream300_m.tif"))
HomeDen300 <- raster(paste0(getwd(),"/HomeDen300.tif"))
Elev300_m <- raster(paste0(getwd(),"/Elev300_m.tif"))
Nat300_contag10k <- raster(paste0(getwd(),"/Nat300_contag10k.tif"))
VRM300m_9pixelmw <- raster(paste0(getwd(),"/VRM300m_9pixelmw.tif"))
ndvi300 <- raster(paste0(getwd(),"/ndvi300.tif"))
load("gye_avg_coef.Rdata")

crw_stack_gye <- stack(x = c(d2ForestEdge300_m,
d2Hwy300_m, RoadHwyDen300_mw1500,
d2River300_m, d2Stream300_m, HomeDen300, Elev300_m,
Nat300_contag10k, VRM300m_9pixelmw, ndvi300))

mat_gye <- ff(vmode = "double", dim = c(ncell(crw_stack_gye),
nlayers(crw_stack_gye)),
filename = paste0(getwd(),"/mat_gye.ffdata"))

for(i in 1:nlayers(crw_stack_gye)){
mat_gye[,i] <- crw_stack_gye[[i]][]
}

mat_gye[][,6] <- log10(mat_gye[][,6] + 0.000001) ## create log_HomeDen column
mat_gye <- cbind(mat_gye[], (mat_gye[][,7])^2)
id_raster <- raster(crw_stack_gye[[1]])
id_raster[] <- 1:ncell(crw_stack_gye[[1]])
weights <- inv.logit((mat_gye %*% gye_avg_coef) - 13.32842)

# Correlated walks function
bearmoves <- function(){
steps.proposed <- 10
num.steps <- 15000
step.max <- 15000
location <- matrix(ncol = 2, nrow = num.steps)
# Used boundaries of 1973-1979 GB distribution
location[1, 1] <- runif(1, 464241.9, 625253.3)
location[1, 2] <- runif(1, 4829879, 5003267)
turn.angle.prev <- runif(1, 0, 2 * pi)
for(i in 2:num.steps){
# Generate step lengths and turning angles
rep_num <- 1
repeat{
proposed <- matrix(ncol = 2, nrow = steps.proposed)
step.length <- (rbeta(steps.proposed, 0.7077255, 9.0272418) *
(step.max - 100)) + 100 # MLE values from Chris paper
turn.angle <- rvonmises(steps.proposed, (circular(0.02) - turn.angle.prev),
0.607)
# Generate proposed movements
proposed[, 1] <- step.length * cos(turn.angle) + location[i-1, 1]
proposed[, 2] <- step.length * sin(turn.angle) + location[i-1, 2]
# Find probabilities of proposed movements
mw <- c()
proposed_id <- extract(id_raster, proposed)
mw <- weights[proposed_id, ]
proposed <- proposed[!is.na(mw), ]
correct_steps <- sum(!is.na(mw))
if(correct_steps > 1){
mw <- mw[!is.na(mw)]
mw <- mw / sum(mw)
index <- sample(1:correct_steps, 1, prob = mw)
location[i, ] <- proposed[index, ]
turn.angle.prev <- turn.angle[index]
break
}
if(rep_num > 10){
location[i, 1] <- runif(1, 464241.9, 625253.3)
location[i, 2] <- runif(1, 4829879, 5003267)
turn.angle.prev <- runif(1, 0, 2 * pi)
break
}
rep_num <- rep_num + 1
}
}
return(location)
}

# Run random walks
start_time <- Sys.time()
num_cores <- parallel::detectCores() - 1
cl <- parallel::makeCluster(num_cores)
doParallel::registerDoParallel(cl)
ogbears <- foreach(i = 1:5000 , .combine = rbind,
.export = c("weights", "id_raster"),
.packages = c("circular", "gdistance", "raster"))
%dopar% bearmoves()
parallel::stopCluster(cl)
end_time <- Sys.time()
start_time - end_time
```

Appendix-2: Simulation Study

Jacob Dym

May 3, 2018

Library packages

```
library(ggplot2)
library(mnormt)
library(survival)
library(circular)
```

```
##
## Attaching package: 'circular'
```

```
## The following objects are masked from 'package:stats':
##
##     sd, var
```

```
library(MASS)
library(fitdistrplus)
library(glmm)
```

```
## Loading required package: trust
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: Matrix
```

```
## Loading required package: digest
```

```
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:survival':
##
##     aml
```

```
library(parallel)
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js

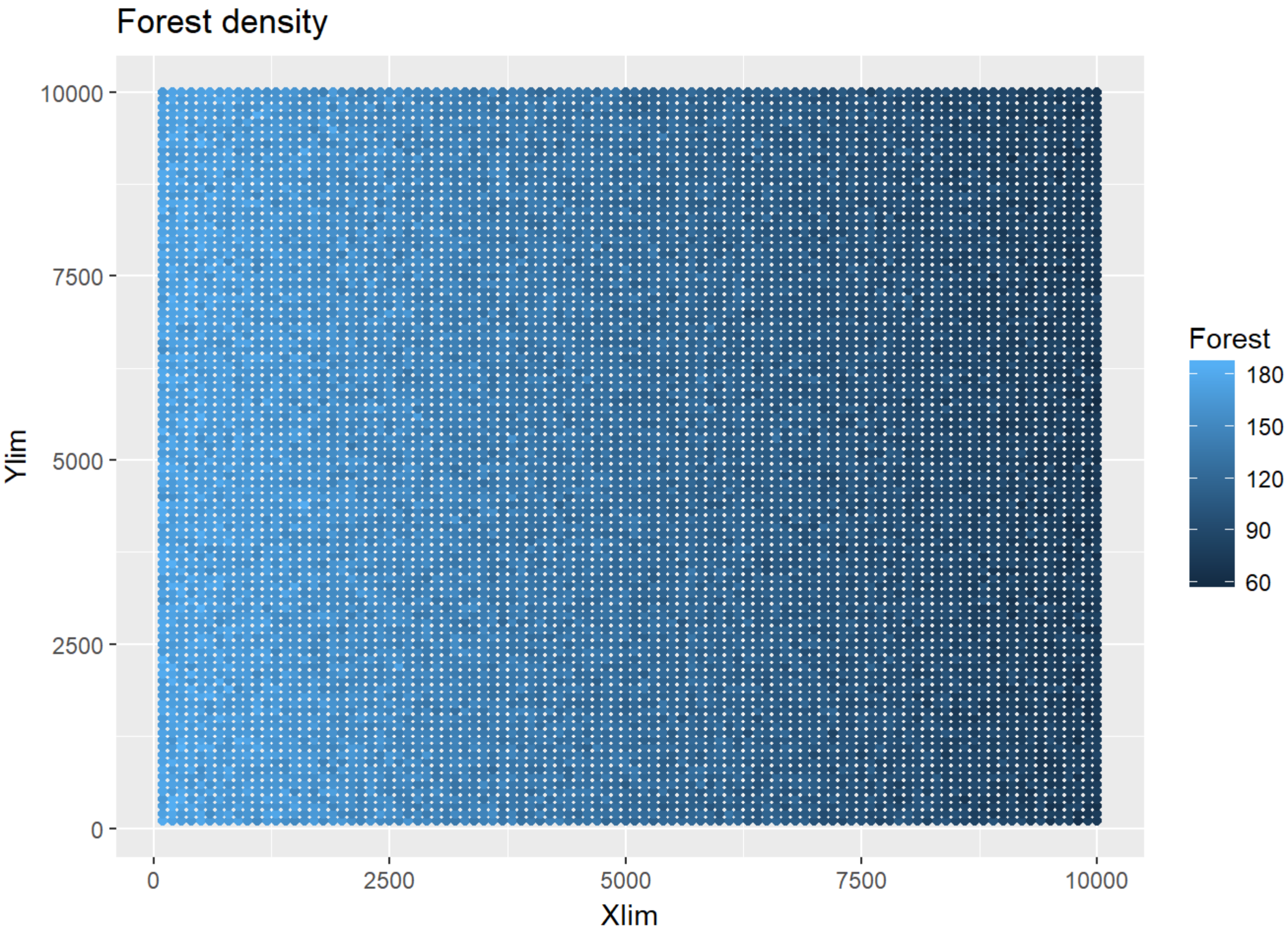
```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

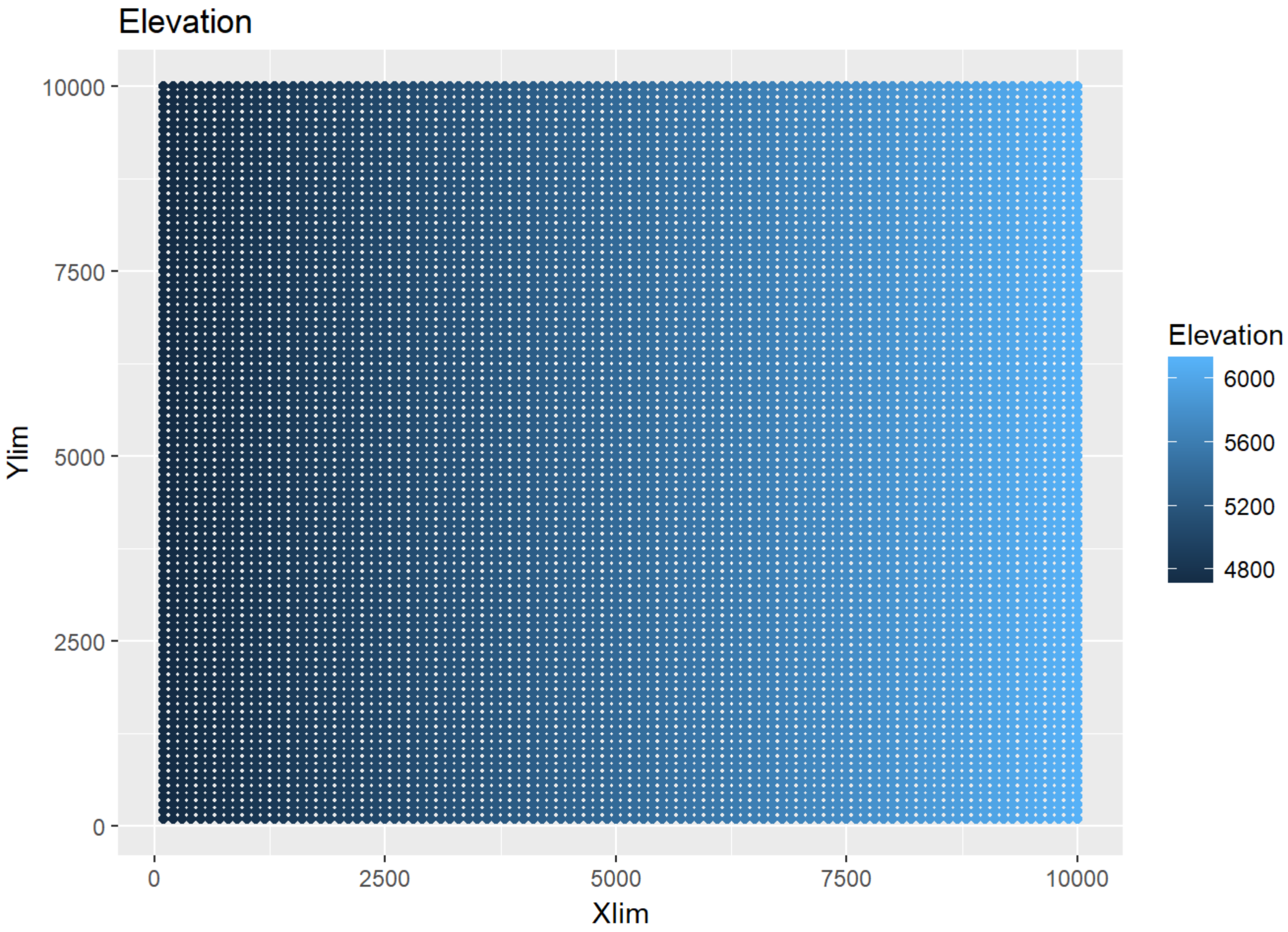
Simulating Range

Three covariates were included to simulate the landscape; the elevation of a grid cell, whether there is a water source in the grid cell, and what the forest density of the grid cell is.

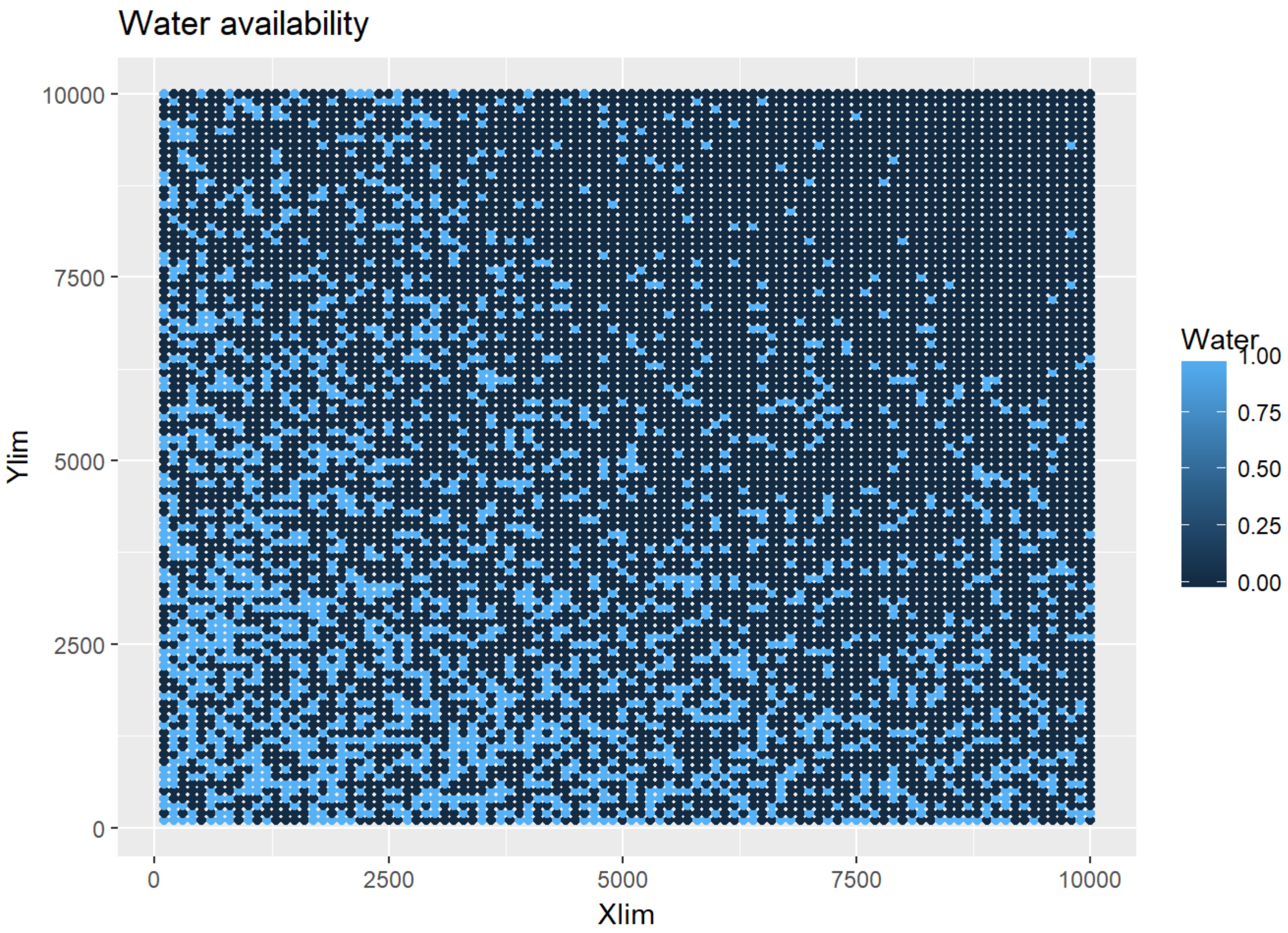
```
x <- seq(0, 10000, 100)
y <- seq(0, 10000, 100)

# Create grid
Grid_xlim <- c()
Grid_ylim <- c()
index <- 0
for (i in 1:101) {
    for (j in 1:101) {
        Grid_xlim[index + j] <- x[i]
        Grid_ylim[index + j] <- y[j]
    }
    index <- index + 101
}
Grids <- data.frame(Xlim = Grid_xlim, Ylim = Grid_ylim)
Grids <- Grids[Grids[, 1] != 0, ]
Grids <- Grids[Grids[, 2] != 0, ]
Grids$XLlim <- Grids$Xlim - 100
Grids$YLlim <- Grids$Ylim - 100
l <- dim(Grids)[1]

# Randomly generate forest densities
forest <- rep(0, l)
for (i in 1:l) {
    forest[i] <- round(rgamma(1, 150, 2))
}
forestry <- function(x) {
    50 * dbeta(x/10001, 1, 2)
}

Grids$Forest <- forest + forestry(Grids$Xlim)
ggplot(Grids, aes(Xlim, Ylim, color = Forest)) + geom_point() + labs(title = "Forest density")
  # Forest is random throughout the grids
```



```
# Generate elevations
elev <- function(x, y) {
    4700 + 700 * dbeta(x/10001, 2, 1) + 500 * dnorm(y, 2000, 300)
}
Grids$Elevation <- elev(Grids$Xlim, Grids$Ylim)
ggplot(Grids, aes(Xlim, Ylim, color = Elevation)) + geom_point() + labs(title = "Elevation")
# Elevation is set up to be highest at the centroid
```

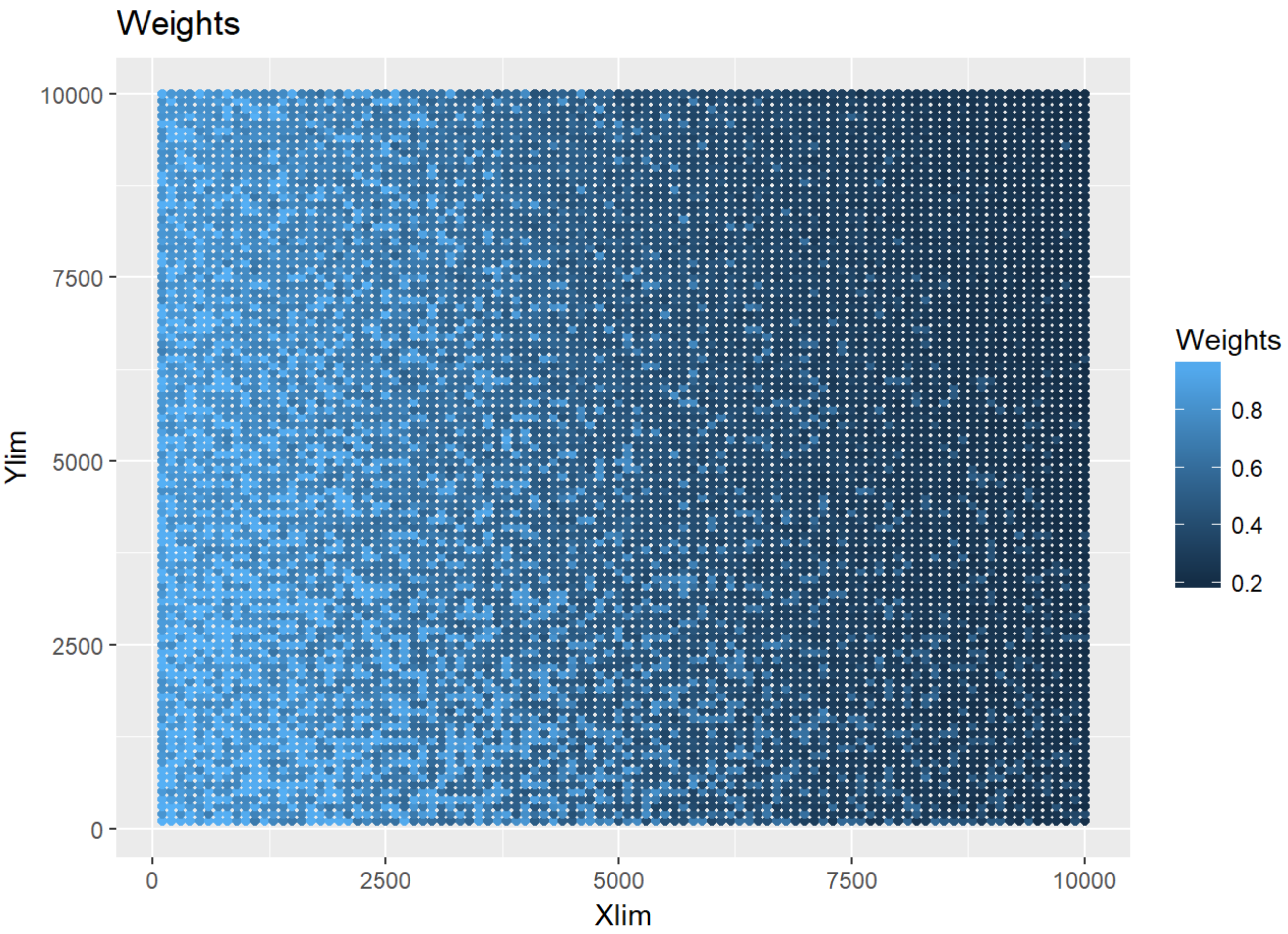


```
# Generate water locations, make lower left region more likely to have water
# and upper right least likely
water_prob <- function(x, y) {
    dbeta(y/10001, 1, 3) + dbeta(x/10001, 1, 3)
}
p <- water_prob(Grids$Xlim, Grids$Ylim)
p <- p/(max(p) + 4)
water <- rbinom(10000, 1, p)
Grids$Water <- water
ggplot(Grids, aes(Xlim, Ylim, color = Water)) + geom_point() + labs(title = "Water availabilit
y")  # Lake area towards left boundary
```



Assign true movement probabilities

```
# Set up grid for true probability attraction
w <- exp(cbind(Grids$Forest, Grids$Elevation, Grids$Water) %*% c(0.01, -5e-05,
    0.4))
# Higher elevation less desireable, water more desireable, and higher
# density more desireable
w <- inv.logit(w - median(w))
Grids$Weights <- w
ggplot(Grids, aes(Xlim, Ylim, color = Weights)) + geom_point() + labs(title = "Weights")
```



Bear movement data generation function

```
# bearmoves is a function of the number of steps a bear could make each
# movement, the number of steps a bear will eventually take in the
# simulation, and the maximum step length a bear can make.
bearmoves <- function(steps.proposed, num.steps, step.max) {

    # Initialize location matrix
    location <- matrix(ncol = 2, nrow = num.steps)
    # Pick starting location
    location[1, ] <- runif(2, 4500, 5500)
    # Generate proposed movement matrix
    proposed <- matrix(ncol = 2, nrow = steps.proposed)
    # Set first turn angle
    turn.angle.prev <- runif(1, 0, 2 * pi)  # Start bears headed in a random direction

    # Generate remaining number of steps
    for (i in 2:num.steps) {
        # Set weight vector of proposed movements
        mw <- c()
        # Ensure there are at least 2 proposed steps in bounds for each set of steps
        repeat {
            # Generate step lengths and turning angles
            step.length <- step.max * rbeta(steps.proposed, 1.2, 3)  # arbitrary beta distribu
tion parameters
```

```
            turn.angle <- rvonmises(steps.proposed, (circular(0) - turn.angle.prev),
                0.3)  # bears prefer to move in the same direction slightly
            # Generate proposed movements
            proposed[, 1] <- step.length * cos(turn.angle) + location[i - 1,
                1]
            proposed[, 2] <- step.length * sin(turn.angle) + location[i - 1,
                2]
            # Set boundary conditions
            bound <- (proposed[, 1] < 0) | (proposed[, 2] < 0) | (proposed[,
                1] > 10000) | (proposed[, 2] > 10000)
            proposed_inbounds <- proposed[!bound, ]
            proposed_outbounds <- proposed[bound, ]
            turn_in <- turn.angle[!bound]
            turn_out <- turn.angle[bound]
            if (sum(!bound) > 1) {
                break
            }
        }
        # Find probabilities of proposed movements
        for (j in 1:dim(proposed_inbounds)[1]) {
            idl <- (proposed_inbounds[j, 1] < Grids$Xlim) & (proposed_inbounds[j,
                1] > Grids$XLlim) & (proposed_inbounds[j, 2] < Grids$Ylim) &
                (proposed_inbounds[j, 2] > Grids$YLlim)
            mw[j] <- Grids$Weights[idl]
        }
        mw_out <- rep(0, sum(bound))
        mw <- c(mw, mw_out)
        if (length(mw) != steps.proposed) {
            stop(print(proposed))
        }
        proposed <- rbind(proposed_inbounds, proposed_outbounds)
        turn.angle <- c(turn_in, turn_out)
        # Save current location
        index <- sample(steps.proposed, 1, prob = mw/sum(mw))
        location[i, ] <- proposed[index, ]
        # Set previous turn angle
        turn.angle.prev <- turn.angle[index]
    }
    return(location)
}
```

Simulating bear movements

25 bears were run for this simulation to demonstrate the model utility. Each bear takes 3,000 steps with a maximum step length of 400 units. Simulation takes about 1.3 minutes to run on 4 cores/threads for 25 bears with 3,000 steps each.

```
start_time <- Sys.time()
num_cores <- parallel::detectCores() - 4  # Detects number of cores/threads on PC and uses 4 l
ess to run simulations in parallel
# Note if there are 4 or less cores on the PC this method will need to be
# adjusted
cl <- parallel::makeCluster(num_cores)
```

```
doParallel::registerDoParallel(cl)
bear_sim <- foreach(i = 1:25, .combine = rbind, .export = c("Grids"), .packages = c("circular"
)) %dopar%
    bearmoves(10, 3000, 400)
```

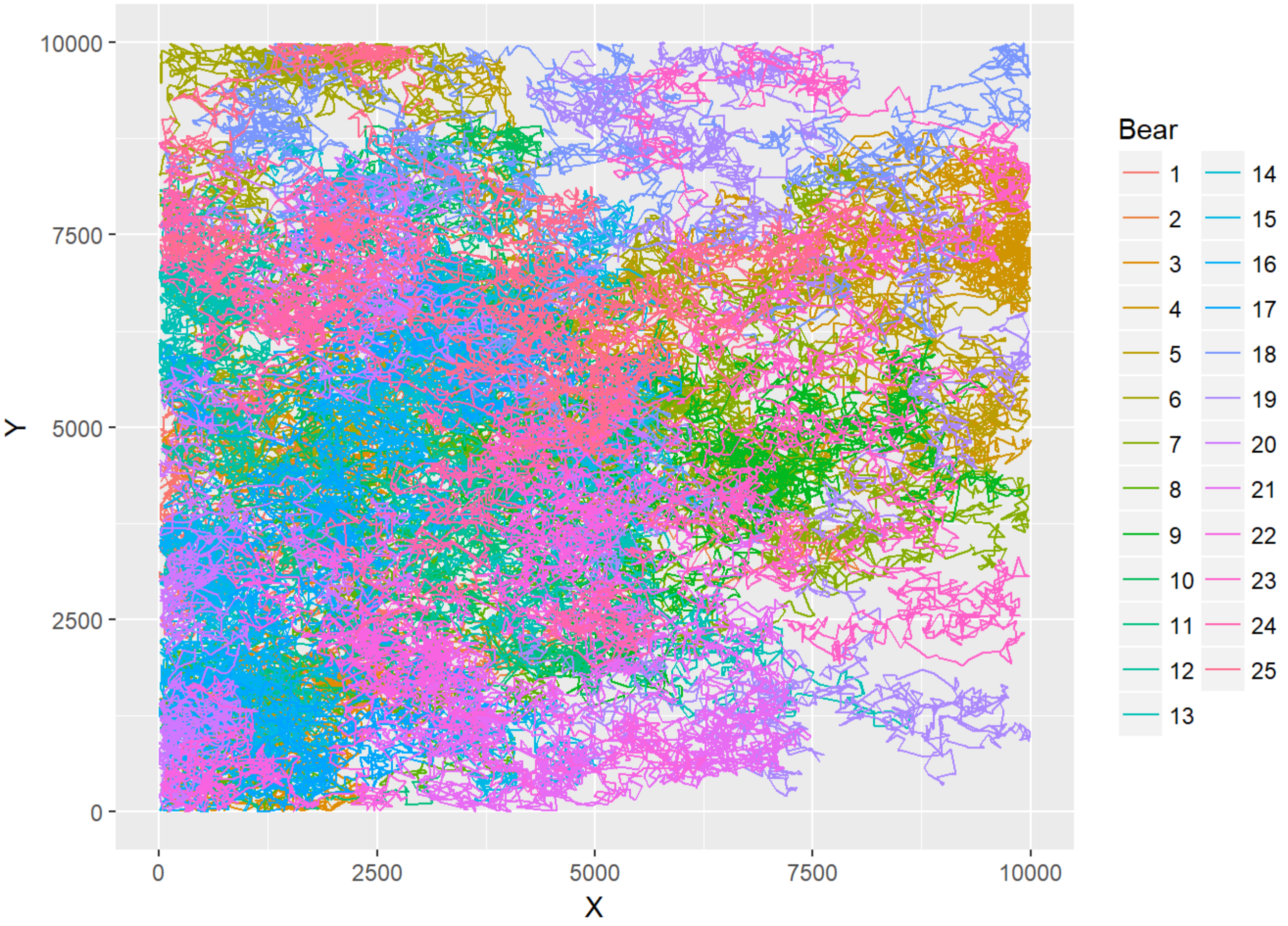
```
## Warning in e$fun(obj, substitute(ex), parent.frame(), e$data): already
## exporting variable(s): Grids
```

```
parallel::stopCluster(cl)
end_time <- Sys.time()
start_time - end_time
```

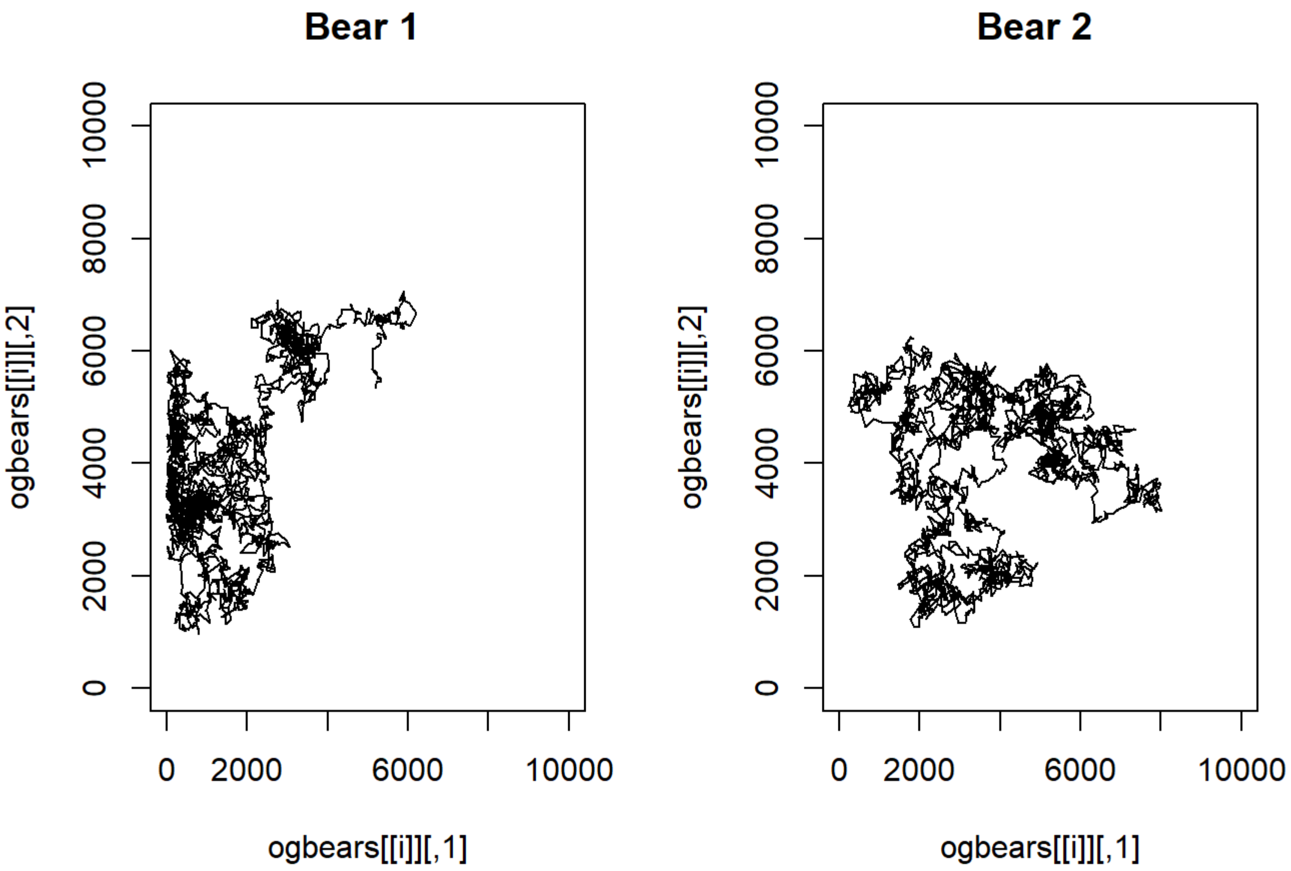
```
## Time difference of -1.272688 mins
```

Plot bear movements

```
ogbears <- list()
for(i in 1:25){
  s <- 3000 * (i-1) + 1
  e <- 3000 * i
  ogbears[[i]] <- bear_sim[c(s:e), ]
}
thebears <- c()
for(i in 1:25){thebears <- c(thebears, rep(i, 3000))}
bear.dat <- cbind(bear_sim, thebears)
colnames(bear.dat) <- c("X", "Y", "Bear")
bear.dat <- as.data.frame(bear.dat)
bear.dat$Bear <- as.factor(bear.dat$Bear)
ggplot(bear.dat, aes(X, Y, color = Bear)) + geom_path()
```



```r
par(mfrow = c(1, 2))
for(i in 1:2){plot(ogbears[[i]], xlim = c(0, 10000), ylim = c(0, 10000), main = paste('Bear',
i, ''), type = 'l')}
```



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

Von Mises distribution

Bayesian methods were explored for estimating the parameters of the turning angles and step lengths, however the MLE approach was taken as there was little difference between the estimates from the two methods. The choice of using MLE was consistent with the original literature so to best mimic it this approach was taken.

Metropolis Hastings algorithms

The Metropolis Hastings algorithm was used for estimating parameters of step length distribution (Beta) and turn angle distribution (Von Mises). The von Mises distribution converges to a uniform distribution as the variance parameter approaches 0, and converges to a normal distribution as the variance parameter approaches ∞.

Extract step lengths and turning angles functions

Extract step lengths and turning angles (both conditional on trajectory and not) from data. Data must be of the form n x 2 matrix for [X, Y] coordinates.

```
Steps_Angles <- function(dat) {
    n <- dim(dat)[1]
    theta <- rep(0, n - 1)  # cardinal turning angles
    theta.c <- rep(0, n - 2)  # conditional turning angles
    steplength <- rep(0, n - 1)  # set up vector to store step lengths
    for (i in 2:n) {
        # Compute distance of steps
        x.dist <- dat[i, 1] - dat[i - 1, 1]
        y.dist <- dat[i, 2] - dat[i - 1, 2]
        dist <- sqrt(x.dist^2 + y.dist^2)
        # Classify what trig function to use based on step
        if ((y.dist > 0) & (x.dist > 0)) {
            theta[i - 1] <- acos(x.dist/dist)
        } else if ((y.dist > 0) & (x.dist < 0)) {
            theta[i - 1] <- acos(x.dist/dist)
        } else if ((y.dist < 0) & (x.dist < 0)) {
            theta[i - 1] <- pi/2 + acos(x.dist/dist)
        } else theta[i - 1] <- 3 * pi/2 + acos(x.dist/dist)
        steplength[i - 1] <- dist
    }
    # If there are any NAs, turn angle is set to 0
    theta[is.na(theta)] <- 0
    n2 <- n - 1
    # Calcualte trajectories
    for (i in 2:n2) {
        theta.c[i - 1] <- theta[i] - theta[i - 1]
        if (theta.c[i - 1] < 0) {
            theta.c[i - 1] <- theta.c[i - 1] + 2 * pi
        }
    }
    results <- list(Lengths = steplength, Theta = theta, RTheta = theta.c)
}

# Function to rotate matrices, redundant in current build
rotate <- function(dat, r) {
    R <- matrix(c(cos(r), sin(r), -sin(r), cos(r)), ncol = 2, nrow = 2)
    dat %*% R
}
```

Extract step information

```
# Extract step information
dat.subset <- list()
angles_lengths <- list()
for (i in 1:25) {
    dat.subset[[i]] <- subset(bear.dat, bear.dat$Bear == i)
}
for (i in 1:25) {
    angles_lengths[[i]] <- Steps_Angles(dat.subset[[i]])
}
```

Get parameter estimates

Obtain parameter estimates of underlying Beta distribution and Von Mises distribution for step lengths and turning angles.

MLE is used for parameter estimates in this build.

```
params <- list()
for (i in 1:25) {
    max_step <- max(angles_lengths[[i]]$Lengths + 0.02)
    beta <- fitdist((angles_lengths[[i]]$Lengths + 0.01)/max_step, distr = "beta",
        method = "mle")
    vonm <- mle.vonmises(angles_lengths[[i]]$RTheta)
    params[[i]] <- list(MaxStep = max_step, Beta = beta, VonMises = vonm)
}
```

Simulating data from learned parameters and estimating grid weights

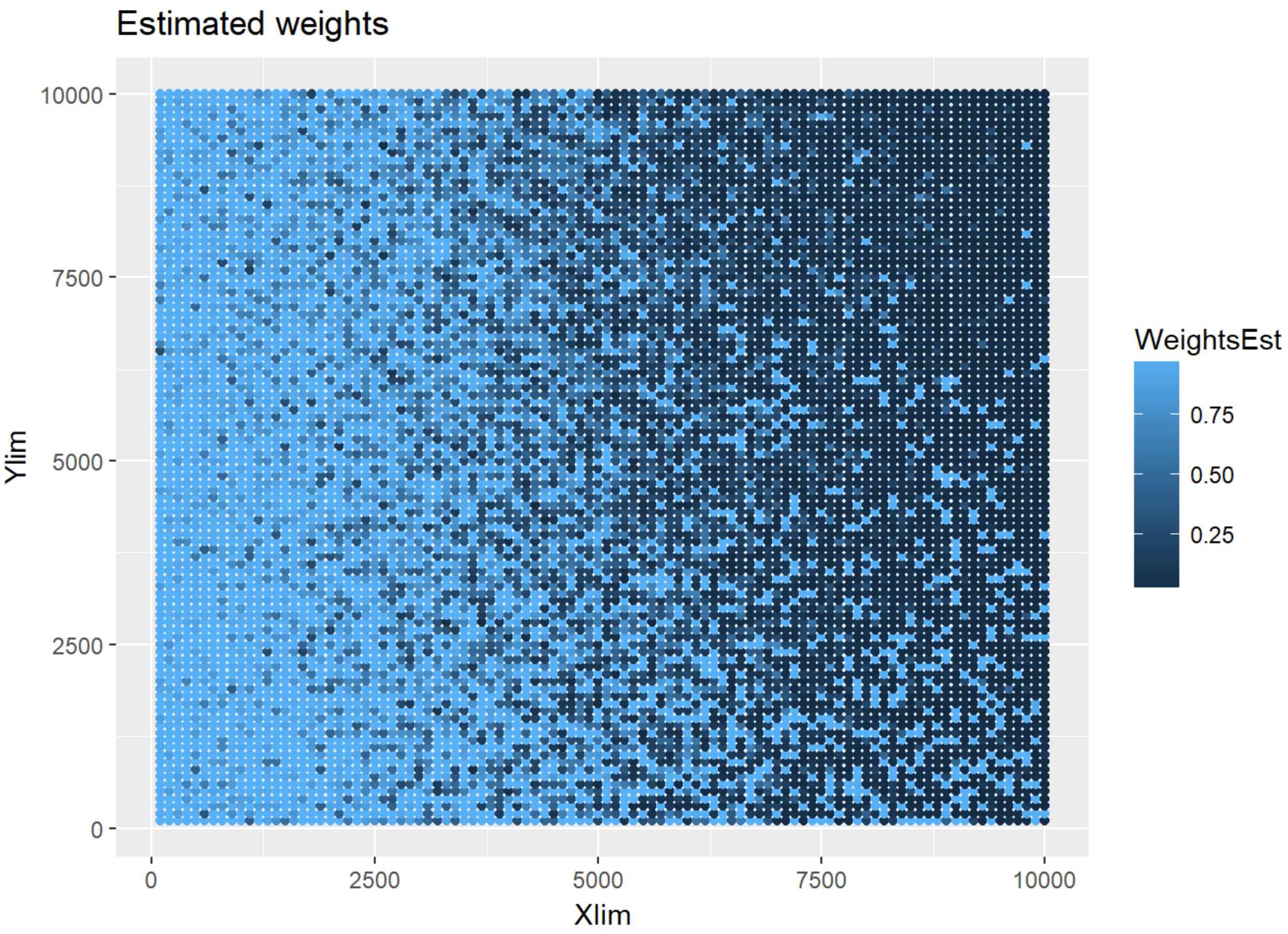
```
CLRdat <- function(dat, steps.proposed) {
    l <- dim(dat)[1]
    dat_matrix <- c()
    sm.est <- max(angles_lengths[[m]]$Lengths)
    for (i in 2:l) {
        # Ensure at least 8 proposals to compare movement with
        repeat {
            # Proposed movements based on individual parameter estimates
            proposed <- matrix(ncol = 2, nrow = steps.proposed)
            location <- Grids[(dat[i - 1, 1] < Grids$Xlim) & (dat[i - 1, 1] >
                Grids$XLlim) & (dat[i - 1, 2] < Grids$Ylim) & (dat[i - 1, 2] >
                Grids$YLlim), ]
            step_lengths <- rbeta(steps.proposed, params[[m]]$Beta$estimate[1],
                params[[m]]$Beta$estimate[2]) * params[[m]]$MaxStep
            turn_angles <- rvonmises(steps.proposed, mu = params[[m]]$VonMises$mu,
                kappa = params[[m]]$VonMises$kappa)
            turn_angles <- as.numeric(turn_angles)

            # Generate proposed movements
            proposed[, 1] <- step_lengths * cos(turn_angles) + dat[i - 1, 1]
            proposed[, 2] <- step_lengths * sin(turn_angles) + dat[i - 1, 2]

            # Clean proposed movements
            clean <- (proposed[, 1] < 0) | (proposed[, 2] < 0) | (proposed[,
                1] > 10000) | (proposed[, 2] > 10000)
            proposed <- proposed[!clean, ]
            s <- sum(!clean)
            if (s > 7) {
                break
            }
        }
        # Set up response values
        response <- c(1, rep(0, dim(proposed)[1]))

        # Get covariates of proposed locations
        ids <- c()
        for (j in 1:dim(proposed)[1]) {
            ids[j] <- which((proposed[j, 1] < Grids$Xlim) & (proposed[j, 1] >
                Grids$XLlim) & (proposed[j, 2] < Grids$Ylim) & (proposed[j,
                2] > Grids$YLlim))
        }
        covariates <- Grids[ids, 5:7]

        # Consturct the conditional movements with step as Strata
        group <- rep(i - 1, dim(proposed)[1] + 1)
        cov_matrix <- rbind(location[5:7], covariates)
        group_matrix <- cbind(response, cov_matrix, group)

        # Append current step set to full matrix
        dat_matrix <- rbind(dat_matrix, group_matrix)
    }
    return(dat_matrix)
}
```

Conditional logistic regression to estimate grid weights

Probability estimates were more drastic in the estimated weights than in the original, however the overall trend is well approximated

```
# Conditional logistic regression on all bears
clbear.est <- list()
cbinom <- list()

# Run conditional logistic regression for each bear
for (m in 1:25) {
    dat <- ogbears[[m]]
    # Estimate slope coefficients
    cbinom[[m]] <- CLRdat(dat, 10)
    c0 <- cbinom[[m]][, 1]
    c1 <- cbinom[[m]][, 2]
    c2 <- cbinom[[m]][, 3]
    c3 <- cbinom[[m]][, 4]
    c4 <- cbinom[[m]][, 5]
    clbear.est[[m]] <- clogit(c0 ~ c1 + c2 + c3 + strata(c4))
}

# Estimate weights
clcoeffs <- matrix(ncol = 3, nrow = 25)
for (i in 1:25) {
    clcoeffs[i, ] <- clbear.est[[i]]$coefficients
}
clcoeffs <- apply(clcoeffs, 2, mean)
expb <- exp(cbind(Grids$Forest, Grids$Elevation, Grids$Water) %*% clcoeffs)
expb <- expb - median(expb)
# Subtract out median as per methods in Chris's paper
nweights <- inv.logit(expb)
Grids$WeightsEst <- nweights

# Plot estimated weight map
```

```
ggplot(Grids, aes(Xlim, Ylim, color = WeightsEst)) + geom_point() + labs(title = "Estimated we
ights")
```


Estimated weights
Ylim
Xlim
WeightsEst