

Project

Tianyi Xia, Zetian Zhao, Litao Zheng

17 April, 2025 12:40:08

Species => Great Blue Heron.

Spatial Analysis

```
# Load libraries
library(spatstat)

## Warning: package 'spatstat' was built under R version 4.4.3

## Loading required package: spatstat.data

## Warning: package 'spatstat.data' was built under R version 4.4.3

## Loading required package: spatstat.univar

## Warning: package 'spatstat.univar' was built under R version 4.4.3

## spatstat.univar 3.1-2

## Loading required package: spatstat.geom

## Warning: package 'spatstat.geom' was built under R version 4.4.3

## spatstat.geom 3.3-6

## Loading required package: spatstat.random

## Warning: package 'spatstat.random' was built under R version 4.4.3

## spatstat.random 3.3-3

## Loading required package: spatstat.explore

## Warning: package 'spatstat.explore' was built under R version 4.4.3

## Loading required package: nlme
```

```

## spatstat.explore 3.4-2

## Loading required package: spatstat.model

## Warning: package 'spatstat.model' was built under R version 4.4.3

## Loading required package: rpart

## spatstat.model 3.3-5

## Loading required package: spatstat.linnet

## Warning: package 'spatstat.linnet' was built under R version 4.4.3

## spatstat.linnet 3.2-5

##
## spatstat 3.3-2
## For an introduction to spatstat, type 'beginner'

library(sf)

## Linking to GEOS 3.12.1, GDAL 3.8.4, PROJ 9.3.1; sf_use_s2() is TRUE

library(raster)

## Warning: package 'raster' was built under R version 4.4.3

## Loading required package: sp

## Warning: package 'sp' was built under R version 4.4.3

##
## Attaching package: 'raster'

## The following object is masked from 'package:nlme':
##      getData

```

Load the Covariates

```

load("BC_Covariates.Rda")
ls()

## [1] "DATA"

```

Summary

```
summary(DATA)
```

```
##          Length Class      Mode
## Window       1   SpatialPolygons S4
## Elevation    10    im        list
## Forest       10    im        list
## HFI          10    im        list
## Dist_Water   10    im        list
```

Pre-process

```
# pre-process the csv file
species_data <- read.delim("species.csv", sep = "\t", header = TRUE)
species_data <- species_data[, colSums(is.na(species_data)) < nrow(species_data)]  
  
# Select the key columns for spatial analysis
species_subset <- species_data[, c("scientificName", "decimalLongitude", "decimalLatitude", "coordinateOrder")]
  
# Filter out records with missing coordinates
species_clean <- species_subset[!is.na(species_subset$decimalLongitude) & !is.na(species_subset$decimalLatitude)]
```

Begin Analysis

```
window <- DATA$Window
class(DATA$Window)
```

```
## [1] "SpatialPolygons"
## attr(,"package")
## [1] "sp"  
  
window_sf <- st_as_sf(window)
window_owin <- as.owin(window_sf)
```

PPP-Object

```
# Convert your species coordinates to match the BC window projection
library(sp)  
  
# Create a SpatialPoints object with your species data
species_sp <- SpatialPoints(
  coords = data.frame(x = species_clean$decimalLongitude, y = species_clean$decimalLatitude),
  proj4string = CRS("+proj=longlat +datum=WGS84") # GBIF typically uses WGS84
)  
  
# Transform to match the projection of your BC window
```

```

species_transformed <- spTransform(species_sp, CRS(proj4string(window)))

# Extract the transformed coordinates
transformed_coords <- coordinates(species_transformed)

# Create the ppp object with transformed coordinates
x_coordinates <- transformed_coords[,1]
y_coordinates <- transformed_coords[,2]

# make a df for the species
species_df <- data.frame(x = x_coordinates, y = y_coordinates)

# Check which points are inside the owin window
inside <- inside.owin(x_coordinates, y_coordinates, window_owin)

# filter the points out that match the window
species_df <- species_df[inside, ]

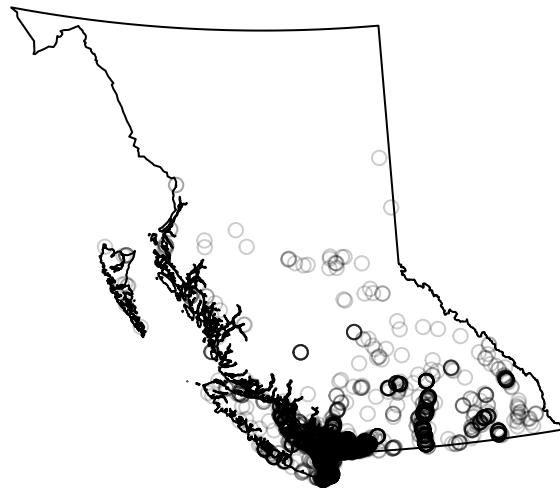
# create the ppp-object
species_ppp <- ppp(
  x = species_df$x,
  y = species_df$y,
  window = window_owin
)

## Warning: data contain duplicated points

# plot the window with points
plot(species_ppp, main = "Species Distribution in BC")

```

Species Distribution in BC



Conclusion:

First impression => the distribution is not homo.

3D Map with Elevation

```
# get the elevation from DATA
elevation <- DATA$Elevation
figure_3d <- persp(elevation,
    # horizontal and vertical rotation.
    theta = 45,
    phi = 20,
    # z-axis expansion (need to be bigger for the BC)
    expand = 25,
    # remove border
    border = NA,
    # add a base
    apron = TRUE,
    # shade intensity
    shade = 0.4,
    # axes off.
    box = FALSE,
    main = "elevation of the great blue heron",
    # allow the overlaying for the parks location.
    visible = TRUE,
```

```

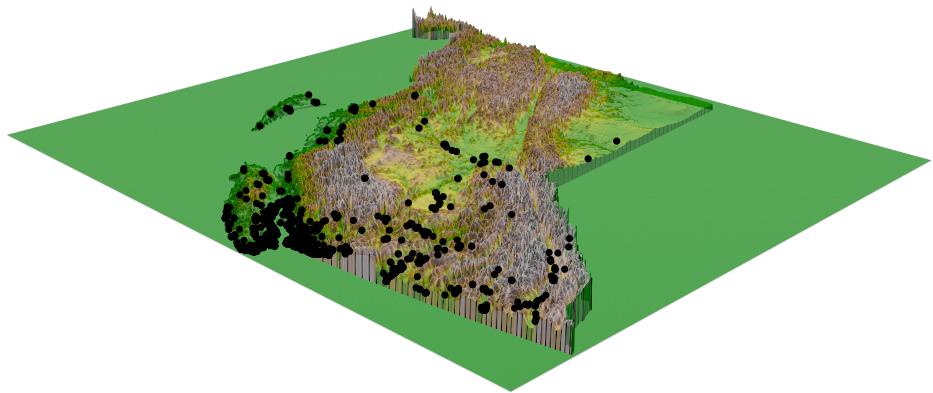
# use the terrain color palette which is built-in for R.
colmap = terrain.colors(289)

# add the points to the persp plot.
perspPoints(species_df, Z = elevation, M = figure_3d, pch = 16, cex = 0.5)

## Warning: data contain duplicated points

```

elevation of the great blue heron



Elevation class for the species

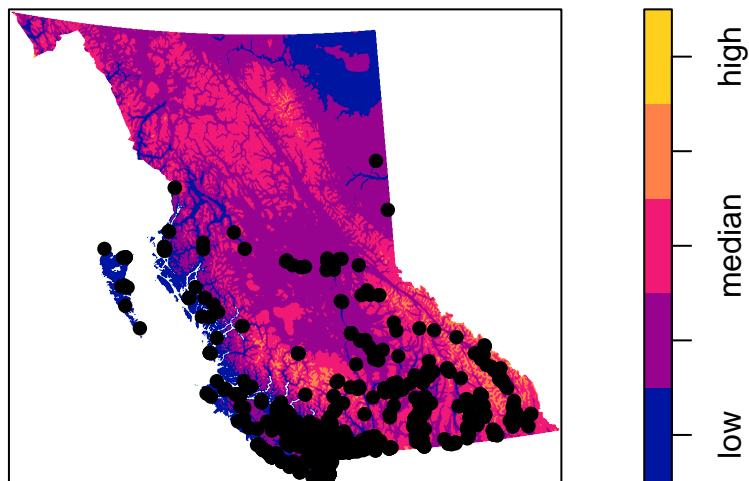
```

plot(cut(elevation, 5, labels = c("low", "median-low", "median", "median-high", "high")), main = "elevation")

# add points
points(species_df$x, species_df$y, pch = 16, col = "black")

```

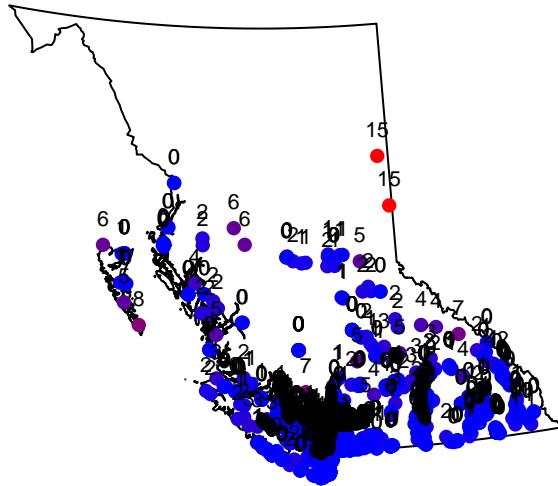
elevation classes for the bc blue heron



Group by distance

```
distances <- nnndist(species_ppp)
species_ppp_with_marks <- species_ppp
marks(species_ppp_with_marks) <- data.frame(distance = distances)
col_pal <- colorRampPalette(c("blue", "red"))(100)
dist_scaled <- cut(distances, breaks = 100, labels = FALSE)
point_cols <- col_pal[dist_scaled]
plot(species_ppp, main = "", use.marks = FALSE, cex = 0.2)
title("Group by distance")
points(species_ppp$x, species_ppp$y, pch = 16, col = point_cols)
text(species_ppp$x, species_ppp$y, labels = round(distances/10000, 0), pos = 3, offset = 0.5, cex = 0.7)
```

Group by distance



Observation:

Most points are dark blue to purple, indicating the nearest neighbors are very close. Therefore, the species tend to cluster in the south of the BC, and most of them are very close. However, the red points mean that they are far away from their neighbors with 15, indicating 150 km away from the neighbor.

Elevation Analysis

```
library(raster)
# 1. Convert spatstat::im (elevation) to raster
elev_raster <- raster(DATA$Elevation)
# get the elevation for species
elevation_sp <- extract(elev_raster, cbind(species_df$x, species_df$y))
# get the median elevation from the park locations.
median_ele_sp <- median(elevation_sp, na.rm = TRUE)
cat("Median elevation of parks in BC is: ", median_ele_sp)

## Median elevation of parks in BC is: 38.8075
```

Conclusion:

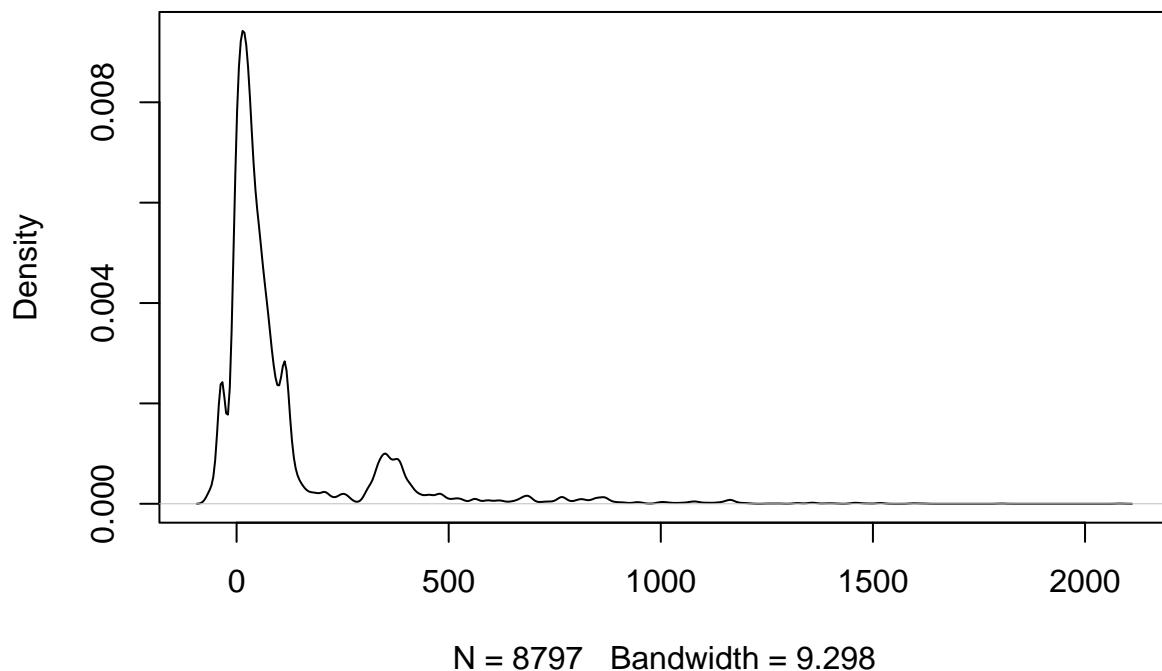
We can see the species reasonably live in the relatively low elevation area like low-lying, flat areas.

```

kde_sp <- density(elevation_sp, na.rm = TRUE)
# plot the elevation density for the specie for better visualization
plot(kde_sp, main = "KDE for blue heron elevations")

```

KDE for blue heron elevations



```

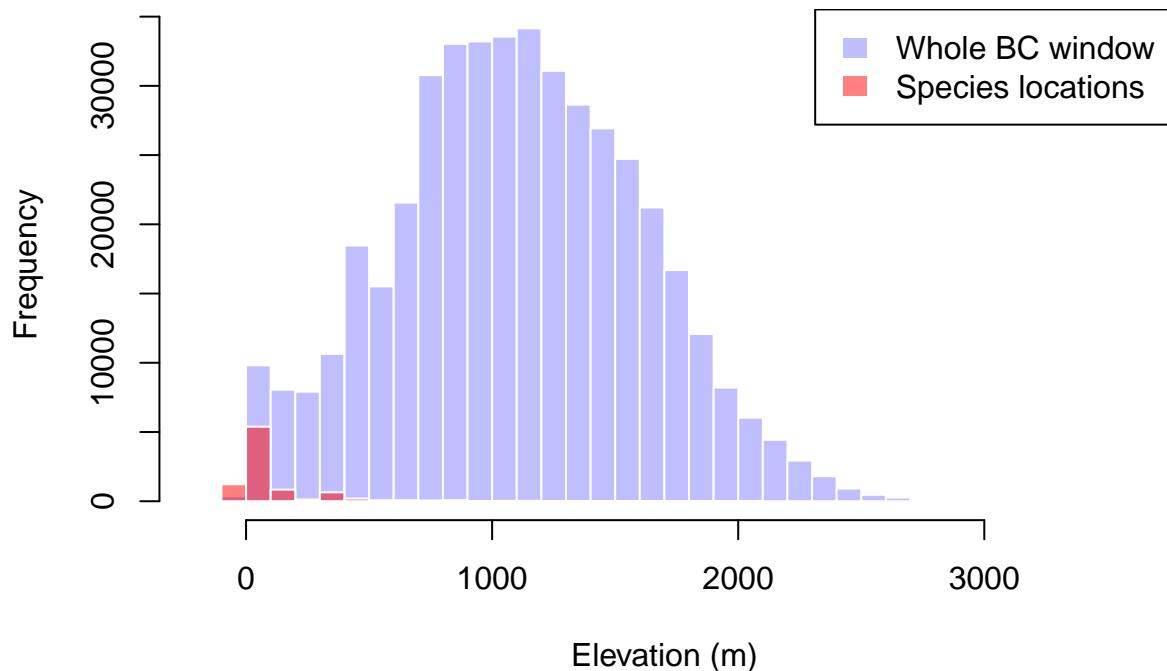
# Extract elevation values from the raster for the whole window (entire raster)
elev_all <- values(elev_raster)

# Extract elevation only at point locations
library(sp)
species_sp <- SpatialPoints(
  coords = data.frame(x = species_ppp$x, y = species_ppp$y),
  proj4string = CRS(projection(elev_raster))
)
elev_points <- extract(elev_raster, species_sp)

# Plot overlaid histograms
hist(elev_all, col = rgb(0, 0, 1, 0.25), main = "Elevation Histogram",
      xlab = "Elevation (m)", border = "white", breaks = 30)
hist(elev_points, col = rgb(1, 0, 0, 0.5), add = TRUE, border = "white", breaks = 30)
legend("topright", legend = c("Whole BC window", "Species locations"),
      fill = c(rgb(0,0,1,0.25), rgb(1,0,0,0.5)), border = NA)

```

Elevation Histogram



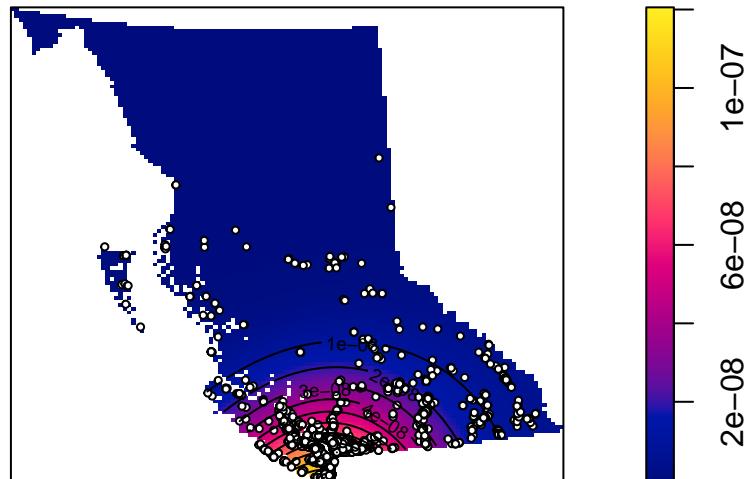
Conclusion:

Enhance the idea that **most of the species** in the low or median elevation area.

Kernel-Density Distribution

```
density_map <- density(species_ppp)
plot(density_map, main = "Kernel Density of Species Observations")
points(species_ppp, pch = 21, cex = 0.5, col = "black", bg= "white")
contour(density_map, add = TRUE)
```

Kernel Density of Species Observations



Conclusion:

Majority in the south corner and only few in the north.

rho plot

```
#Extract elevation information
elev <- DATA$Elevation
#define quartiles
b <- quantile(elev, probs = (0:4)/4, type = 2)
#Split image into 4 equal-area quadrats based on elevation values
Zcut <- cut(elev, breaks = b)
V <- tess(image = Zcut)
quadratcount(species_ppp, tess = V)

## tile
##          (-130,761]      (761,1.1e+03]  (1.1e+03,1.46e+03] (1.46e+03,3.55e+03]
##          8848                  157           48                   12
```

Apparently, most of them (8848) are located in low elevation

Quadrat Test

```
# make 10*10 quadrat
Q <- quadratcount(species_ppp, nx = 10, ny = 10)
# quadrat test
quadrat.test(Q)

## Warning: Some expected counts are small; chi^2 approximation may be inaccurate

##
## Chi-squared test of CSR using quadrat counts
##
## data:
## X2 = 317593, df = 63, p-value < 2.2e-16
## alternative hypothesis: two.sided
##
## Quadrats: 64 tiles (irregular windows)
```

Conclusion:

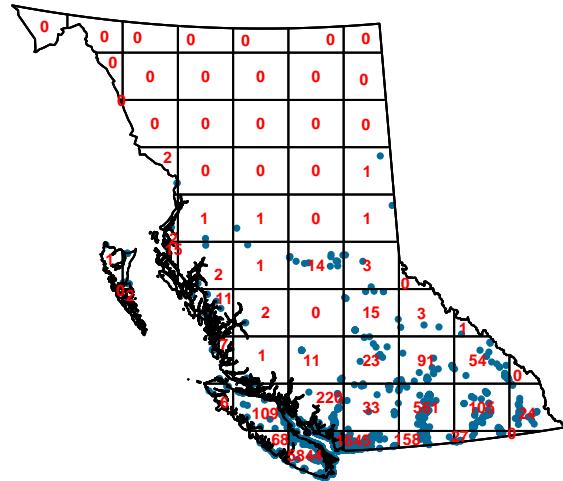
p-value small enough, so reject the null. The assumption about homo is not met.

```
Q <- quadratcount(species_ppp,
                     nx = 10,
                     ny = 10)

plot(species_ppp,
      pch = 16,
      cex = 0.5,
      cols = "#046C9A",
      main = "Beilschmiedia pendula locations")

plot(Q, cex = 0.5, col = "red", add = T, font = 2)
```

Beilschmiedia pendula locations



K-function

```
K <- Kest(species_ppp, correction = "border")
plot(K, main = "Ripley's K-Function")
```

Ripley's K–Function

