

589Project

Nowshaba Durrani

2023-04-09

Libraries

```
#install.packages("mapcan")
#install.packages("bcmaps")
#install.packages("rgbif")
```

```
library(rgbif) #allows searching and retrieving data from GBIF
library(ggplot2) #use ggplot2 to add layer for visualization
library(sp) #Standardized Support for Spatial Vector Data
library(sf)
library(spatstat)
library(maptools)
#library(raster)
#library(mapcan)
#library(bcmaps)
#library(tidyverse)
#library(rgdal)
```

```
#occ_count() # occurrence count for all the species in GBIF (Global Biodiversity Information Facility) -
redFox <- name_backbone(name="Vulpes vulpes")
redFoxList <- occ_data(taxonKey = redFox$speciesKey, hasCoordinate=TRUE, stateProvince='British Columbia')
mydata <- redFoxList$data
```

```
load("BC_Covariates.Rda")

# Create a spatial points data frame from the longitude and latitude columns
coordinates <- mydata[,c("decimalLongitude", "decimalLatitude")]
dat.sp <- SpatialPointsDataFrame(c(mydata[,c('decimalLongitude','decimalLatitude')]), data = mydata)

# Set the current CRS
proj4string(dat.sp)<- CRS("+proj=longlat +datum=WGS84")

# Define the new CRS you want to transform to
new_crs <- CRS("+proj=aea +lat_0=45 +lon_0=-126 +lat_1=50 +lat_2=58.5 +x_0=1000000 +y_0=0 +datum=NAD83")

# Transform the data to the new CRS
data.sp_trans <- spTransform(dat.sp, new_crs)
```

```

#data_transformed
#data.sp_trans

#plot(data.sp_trans, main = "Locations in BC", cex = 0.8, col ="blue")

library(sf)
lapply(DATA, FUN = class)

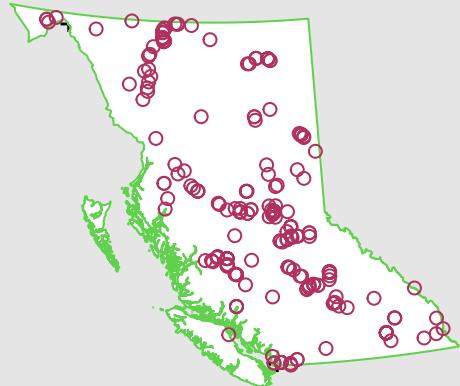
## $Window
## [1] "SpatialPolygons"
## attr(,"package")
## [1] "sp"
##
## $Elevation
## [1] "im"
##
## $Forest
## [1] "im"
##
## $HFI
## [1] "im"
##
## $Dist_Water
## [1] "im"

parks_ppp <- ppp(x = data.sp_trans@coords[,1], # X coordinates
                    y = data.sp_trans@coords[,2], # Y coordinates
                    window = as.owin( DATA[["Window"]]),# Observation window
                    )

col_pal <- c("maroon")
plot(parks_ppp,
      main = "Red Fox in BC",
      cex = 0.9,
      col ="white",
      border = 3,
      cols = col_pal,
      par(bg = "grey90",cex.main = 1.6))

```

Red Fox in BC



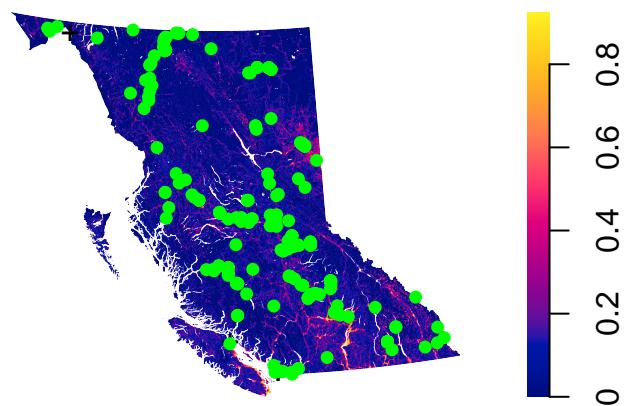
Covariates Analysis

Our data includes 4 covariates we can explore: the elevation, the forest cover, the human footprint inventory (HFI), and the distance to water. Given our research questions, we will start with investigating the HFI and the forest cover.

HFI

```
plot(DATA$HFI, box = F, par(cex.main = 2), main = "HFI")
plot(parks_ppp, pch = 16, cex = 0.9, col = "green", use.marks = F, add = T)
```

HFI

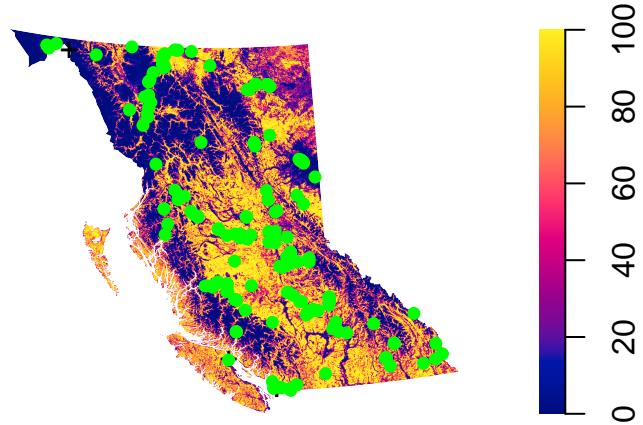


From this plot, it is hard to tell if there is a possible relationship between a red fox occurrence and the HFI.

Forest Cover

```
plot(DATA$Forest, box = F, par(cex.main = 2), main = "Forest")
plot(parks_ppp, pch = 16, cex = 0.9, col = "green", use.marks = F, add = T)
```

Forest

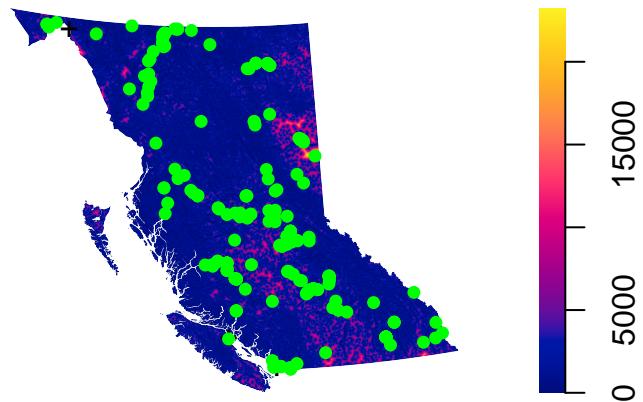


It is also very hard to see if there is a relationship, this time because there are a lot of high values for forest cover all over the province.

Distance to water

```
plot(DATA$Dist_Water, box = F, par(cex.main = 2), main = "Distance to water")
plot(parks_ppp, pch = 16, cex = 0.9, col = "green", use.marks = F, add = T)
```

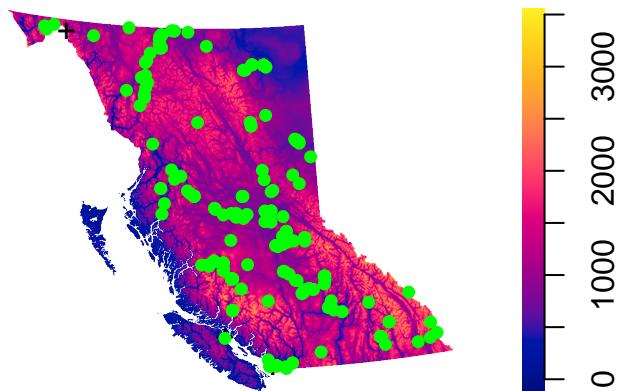
Distance to water



Elevation

```
plot(DATA$Elevation, box = F, par(cex.main = 2), main = "Elevation")
plot(parks_ppp, pch = 16, cex = 0.9, col = "green", use.marks = F, add = T)
```

Elevation



Conclusion: maybe need to segment the continuous values of covariates so that it is easier to see trend, like histogram in lab 1.

Morisita's Index plot

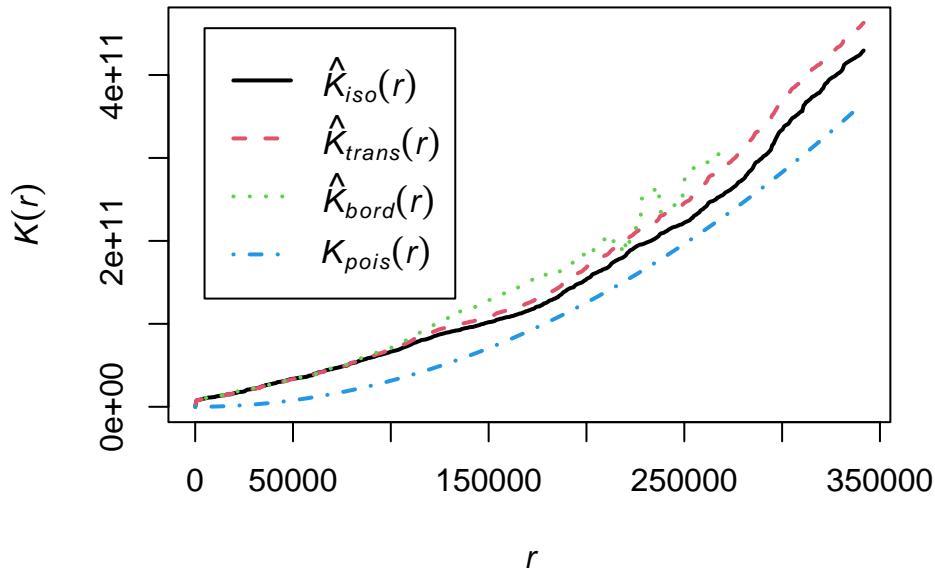
```
miplot(parks_ppp,  
       main = "",  
       pch = 16,  
       col = "maroon")
```

Ripley's K function

Ripley's K-function provides information on whether there are significant deviations from independence between points.

```
k_fox <- Kest(parks_ppp)  
plot(k_fox, lwd=2)
```

k_fox



The blue line is the theoretical line. (I don't know what the other lines mean)

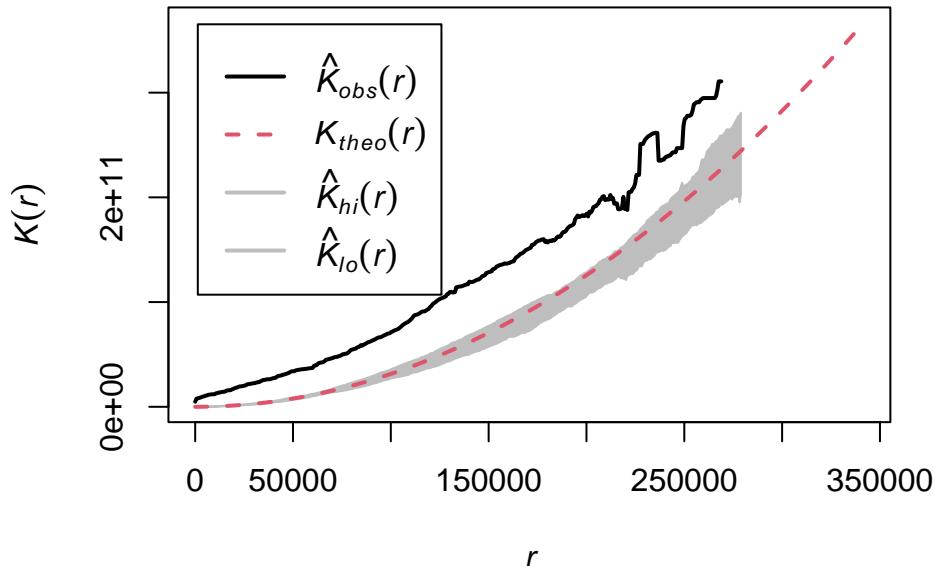
Adding confidence interval at a significance level of 0.05:

```
E_fox_homo <- envelope(parks_ppp,
    Kest,
    correction="border",
    rank = 1,
    nsim = 19, # aka alpha of 0.05
    fix.n = T)

## Generating 19 simulations of CSR with fixed number of points ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19.
##
## Done.

plot(E_fox_homo, lwd = 2)
```

E_fox_homo



(I don't know why it only plotted these lines and not the other ones) We see that the effect appears significant. It is suspicious that the confidence bands increase a lot, (what's the explanation for that?)

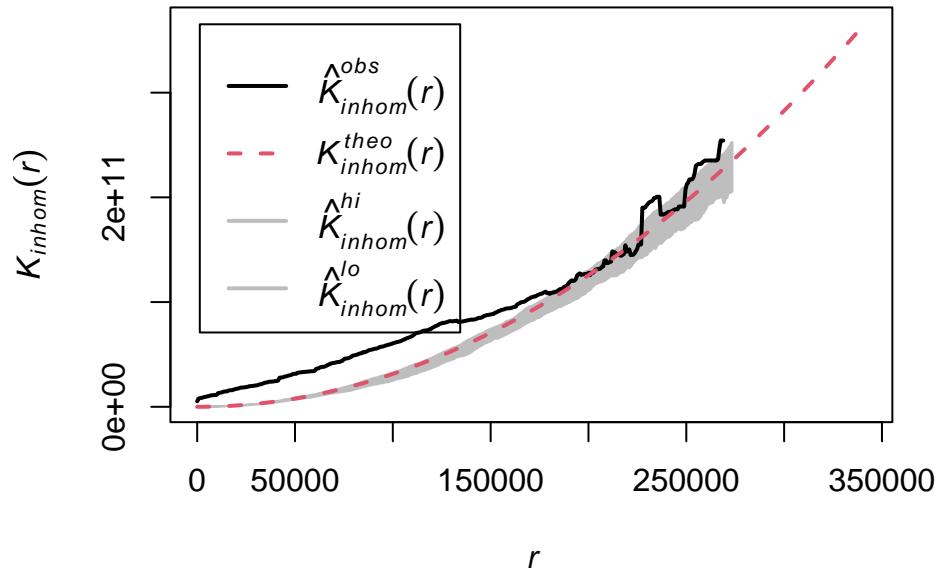
However, we know from first moment analysis that the intensity does not seem homogenous (right? Might need to double check). Trying inhomogenous:

```
E_fox <- envelope(parks_ppp,
                    Kinhom,
                    correction="border",
                    rank = 1,
                    nsim = 19, # aka alpha of 0.05
                    fix.n = T)

## Generating 19 simulations of CSR with fixed number of points ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19.
##
## Done.

plot(E_fox, lwd = 2)
```

E_fox



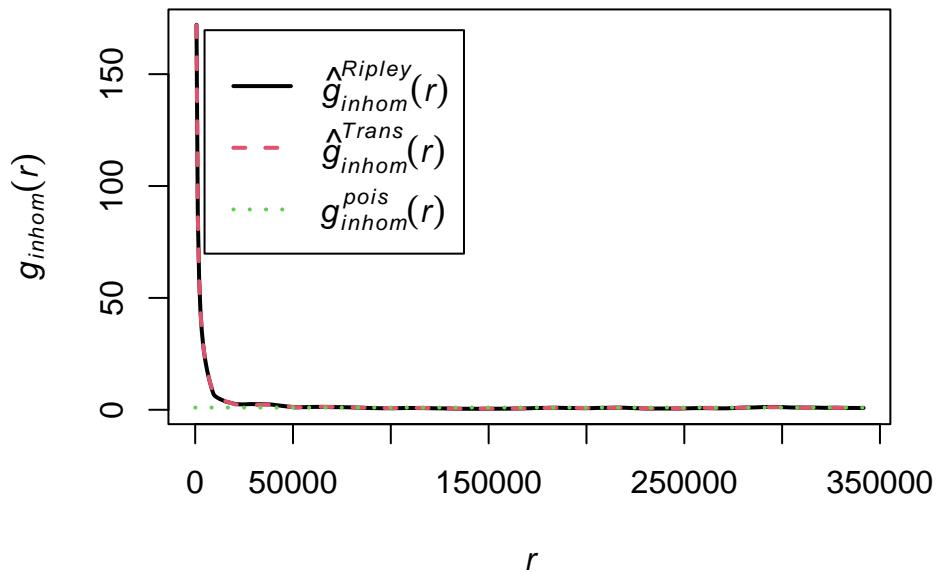
Using the Kinhom function ensures that we are not assuming the intensity is homogenous. It seems like from the smaller numbers, ie smaller distances between points, there is ‘evidence’ of clustering, whereas there are funny things going on as distances increase. The deviations are still meaningful in the ‘smaller’ distances, suggesting that the relationship between points may be due to effects between points rather than relationship with covariates.

Pair correlation function

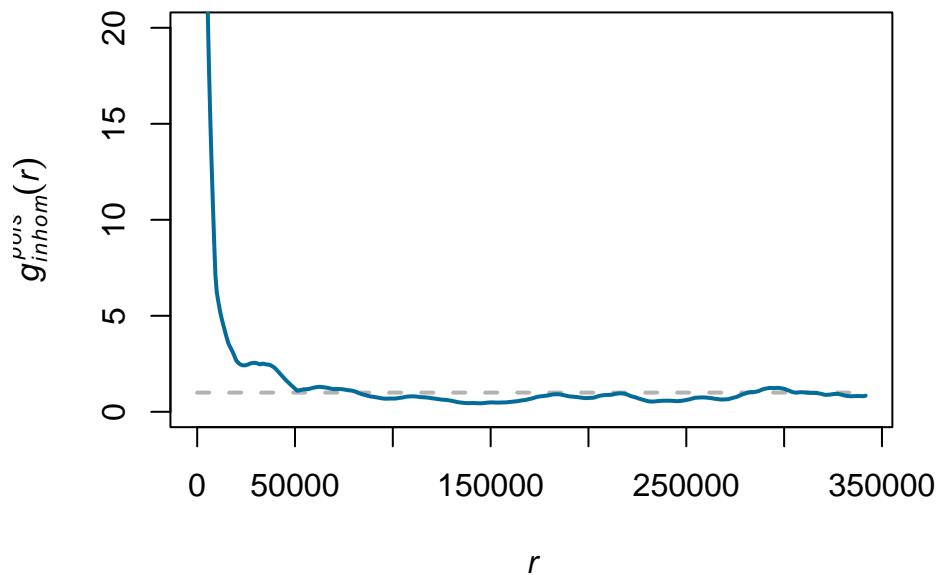
```
# Estimate the g function
pcf_fox <- pcfinhom(parks_ppp) # assumes inhomogeneity

# Default plot method
plot(pcf_fox, lwd = 2)
```

pcf_fox



```
# visualise the results
plot(pcf_fox,
      theo ~ r,
      ylim = c(0,20),
      main = "",
      col = "grey70",
      lwd = 2,
      lty = "dashed")
plot(pcf_fox,
      iso ~ r,
      col = c("#046C9A"),
      lwd = 2,
      add = T)
```



We observe that there seem to be evidence for clustering at smaller than 50 000, but after that it rides the $y = 1$ line slightly under.