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)
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
library(spatstat)
```

```
## Loading required package: spatstat.data

## Loading required package: spatstat.geom

## spatstat.geom 3.1-0

## Loading required package: spatstat.random

## spatstat.random 3.1-4

## Loading required package: spatstat.explore

## Loading required package: nlme

## spatstat.explore 3.1-0

## Loading required package: spatstat.model

## Loading required package: rpart

## spatstat.model 3.2-1
## Loading required package: spatstat.linnet
```

```
## spatstat.linnet 3.0-6
```

```
##
## spatstat 3.0-3
## For an introduction to spatstat, type 'beginner'
```

```
library(maptools)
```

```
## Checking rgeos availability: FALSE
## Please note that 'maptools' will be retired during 2023,
## plan transition at your earliest convenience;
## some functionality will be moved to 'sp'.
##      Note: when rgeos is not available, polygon geometry     computations in maptools depend on gpcli
##      which has a restricted licence. It is disabled by default;
##      to enable gpclib, type gpclibPermit()
```

```
#library(raster)
#library(mapcan)
#library(bcmaps)
#library(tidyverse)
#library(rgdal)
```

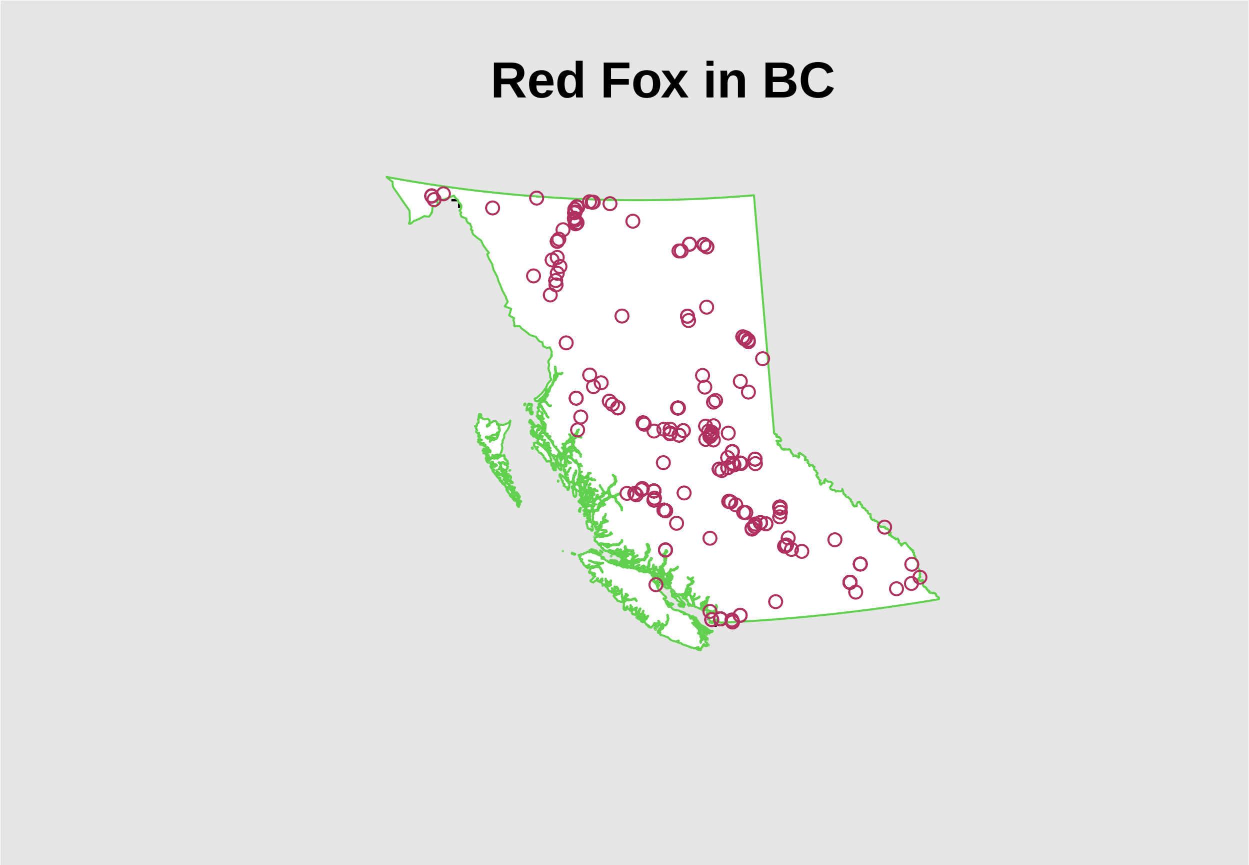
```
#occ_count() # occurance count for all the species in GBIF (Global Biodiversity Information Facility) -

redFox <- name_backbone(name="Vulpes vulpes")
redFoxList <- occ_data(taxonKey = redFox$speciesKey, hasCoordinate=TRUE, stateProvince='British Columbi
mydata <- redFoxList$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