Coastal Solutions Workshop
MODULE 2: Spatial data and eBird
February 20, 2023 2:30-5:30 PM

The goal of this module is to introduce the data products available from the eBird Status and Trends program at the Cornell Lab of Ornithology and demonstrate how to access and work with the data. By the end of this lesson, you should be able to:

- Access Status and Trends data using the eBird website or `eBirdst` package.
- Crop and mask an eBird raster and create a plot.
- Conduct basic analysis of eBird data products, including
    - Custom maximum relative abundance
    - Custom percent of population
    - Chronology plots

This module will include live coding demonstrations along with practice sessions for you to test these methods on your own. All code from the live demonstrations is included in this document for you to copy + paste into your R session. We will use the spatial R packages `sf` and `raster`.

Recommended tutorials
For the R package `sf`: https://r-spatial.github.io/sf/articles/sf1.html
For the R package `eBirdst`: https://ebird.github.io/ebirdst/

Setting up your code

```
####----------------------------------------------------------- ####
#### Spatial R Tutorial: Coastal Solutions Fellows Meeting
##   1. Load and explore data
##   2. Practice plotting the data
##   3. Practice with common eBird calculations
####----------------------------------------------------------- ####

#### Load packages
library(raster)
library(sf)
library(rnaturalearth)
library(ebirdst)
set_ebirdst_access_key("8n4j77a2bv76")  # Required first time only


##### Set working directory
folder <- getwd()
setwd(paste0(folder,"/Data"))
```

Vector data with `sf`

We will access pre-packaged country and state boundaries using the `rnaturalearth` package. If you want to load a shapefile from your computer, use the function `st_read()`.

```
na.map <- ne_states(country = c("United States of America",
      "Mexico","Canada"), returnclass = "sf")
```

There are many ways to plot the data. For example:

```
plot(st_geometry(na.map))  # Just plots the outline
plot(na.map["iso_a2"])     # Plots values from a column
```

We can subset the data in the same way that we would subset a data frame.

```
## Create a subset just for the USA
usa.states <- na.map[which(na.map$iso_a2=="US"),]
plot(st_geometry(usa.states))

usa.states <- usa.states[-which(usa.states$postal %in%
      c("AK","HI")),]
plot(st_geometry(usa.states))

## Highlight a single state
ny <- usa.states[which(usa.states$name == "New York"),]
plot(st_geometry(ny), col="red", add=T)
```

Raster data with `raster`

```
abund.max <- raster("baleag_abundance_seasonal_full-year_max_2021.tif")
plot(abund.max)

## Check the projection and resolution
projection(abund.max)
abund.max@crs
res(abund.max)
```

------------------------------------ **Practice Break #4!** ------------------------------------

Use the code provided above to practice these methods using a North American bird species of your choice.

1. Produce a vector map of the United States with California highlighted in red.

2. Use the eBird website to download a maximum abundance high resolution raster for a species of your choice. Load the raster into R and check projection and resolution.

--

Crop and mask a raster

```
#### Crop and mask the raster to usa.states
st_crs(abund.max)==st_crs(usa.states)
usa.states <- st_transform(x = usa.states, crs = st_crs(abund.max))
plot(abund.max)
plot(st_geometry(usa.states), add = T)

abund.max <- mask(crop(abund.max, usa.states), usa.states)
plot(abund.max)
plot(st_geometry(usa.states), add = T)
```

Create a nice plot of the eBird raster

Note: There is a lot going on here. If you are new to R, try to copy + paste this code into your R session and tweak it so that it runs with your eBird raster.

First we need to set some things up to make the plot.

```
#### Reproject spatial data to a nice-looking CRS
abund.max.p <- projectRaster(from = abund.max, crs = 5070,
      method = "ngb")
usa.states.p <- st_transform(x = usa.states,
      crs = st_crs(abund.max.p))
na.map.p <- st_transform(x = na.map, crs = st_crs(abund.max.p))


#### Assign color bins to quantiles using ebirdst palette
v <- values(abund.max.p)
v <- v[!is.na(v) & v > 0]
bins <- quantile(v, seq(0, 1, by = 0.1))
bins <- c(0, bins)                             # add a bin for 0
pal <- ebirdst::abundance_palette(length(bins) - 1, "weekly")
pal <- c("#dddddd", pal)                       # add a color for 0
```

Now it’s time to plot! The `png()` function exports the plot as a .png file.

```
### Plotting
#png("BAEA_max.png", width=5, height=4, units="in", res=600)

par(mar=c(1,1,3,5))

## Set the spatial extent for the plot
bb <- st_as_sfc(st_bbox(usa.states.p))
plot(bb, col = "white", border = "white")

## Add background reference data
plot(st_geometry(na.map.p), col = "#eeeeee", border = NA, add =
      TRUE)
plot(st_geometry(usa.states.p), col=NA, border="white", lwd=1,
      add=T)
```

```
## Plot prediction region and explicit zeros
plot(abund.max.p, col = "#dddddd", maxpixels = ncell(abund.max.p),
     legend = FALSE, add = TRUE)

## Plot abundances
plot(abund.max.p, col = pal, breaks = bins,
      maxpixels = ncell(abund.max.p),axes = FALSE, legend = F,
      add = TRUE)
plot(st_geometry(usa.states.p), add = TRUE, col = NA,
      border = "black", lwd = 0.6)


## Plot the legend
labels <- quantile(bins, c(0, 0.25, 0.5, 0.75, 0.95))
labels <- round(labels, 2)

plot(abund.max.p, zlim=c(0,1), legend.only = TRUE, col = pal,
     breaks = seq(0, 1, length.out = length(bins)),
     legend.shrink = 0.4, legend.width = 0.6,
     axis.args = list(at = seq(0, 0.95, length.out =
      length(labels)), labels = labels, fg=NA, cex.axis = 0.6,
      lwd.ticks = 0, hadj=0.8),legend.args=list(text='Max relative
      abundance', side=4, font=1, line=-1.7, cex=0.7))
title(main = "Bald Eagle")

#dev.off()
```

------------------------------------ Practice Break #5! ------------------------------------

Use the code from above to make a plot of your eBird raster!

--

Downloading eBird Status and Trends data using `eBirdst()`

```
#### Getting started with eBirdst
ebirdst_download("baleag", dry_run = TRUE) # Shows available data
ebirdst_download("baleag", pattern = "abundance_median_hr",
      dry_run = TRUE)
ebirdst_download("baleag", pattern = "abundance_median_hr")
path <- get_species_path(species = "Bald Eagle")

abund.weekly <- load_raster(path = path, product = "abundance",
      period = "weekly")

#### Explore this data product
plot(abund.weekly$w2021.01.04)
plot(abund.weekly$w2021.07.20)
```

For our next task, we need to mask and crop the weekly eBird relative abundance layers

```
#### Mask and crop to NY
ny <- usa.states[which(usa.states$name == "New York"),]
ny <- st_transform(x = ny, crs = st_crs(abund.weekly))
st_crs(abund.weekly)==st_crs(ny)

abund.ny <- mask(crop(abund.weekly, ny), ny)  # takes ~20 seconds
plot(abund.ny$w2021.01.04)
```

Custom maximum relative abundance

```
#### Task 1: Calculate the maximum relative abundance over
##   weeks of interest
names(abund.ny)

abund.ny.max <- calc(abund.ny[[c(18:22)]], fun = max, na.rm = TRUE)
abund.ny.max[abund.ny.max == -Inf] <- NA

## Reproject for nicer plotting
abund.ny.max <- projectRaster(from = abund.ny.max, crs = 5070,
      method = "ngb")
plot(abund.ny.max)
```

Custom % of population

```
#### Task 2: Calculate percent of population over a custom area
ebirdst_download("baleag", dry_run = TRUE)
ebirdst_download("baleag", pattern = "percent-
population_seasonal_mean_hr")
perc.pop <- load_raster(path = path, product = "percent-population",
period = "seasonal", metric = "mean")

## Crop and mask to NY
perc.pop.ny <- mask(crop(perc.pop$nonbreeding, ny), ny)
plot(perc.pop.ny)
cellStats(x = perc.pop.ny, stat = "sum", na.rm = TRUE) * 100


#### Task 3: Percent of regional population
## Start with the same NY raster as above
total.sum <- cellStats(x = perc.pop.ny, stat = "sum", na.rm = TRUE)
perc.pop.ny2 <- perc.pop.ny / total.sum
plot(perc.pop.ny2)

## This layer should now sum to 1
cellStats(x = perc.pop.ny2, stat = "sum", na.rm = TRUE)
```

Chronology plot

```
#### Task 4: Produce a chronology plot of relative abundance
ny.sum <- data.frame(week = names(abund.ny),
                     week.number = 1:52,
                     sum = NA)
ny.sum$sum <- cellStats(x = abund.ny, stat = "sum", na.rm = TRUE)
ny.sum$prop.total <- ny.sum$sum / max(ny.sum$sum)

dev.off()
par(las = 1)
plot(ny.sum$week.number, ny.sum$prop.total, ylim = c(0,1), type =
"b", pch = 16, lwd = 1.5,
     ylab = "Relative abundance (proportion of max)",
     xlab = "Week of the year")
```

------------------------------------ **Practice Break #6!** ------------------------------------

Use the code from above to practice these common uses of eBird data products.

--

BONUS! Extract raster values from spatial points
The dataset "Turbine_sites.csv" describes the proposed sites of wind turbines in the finger lakes region of New York State.

```
#### OPTIONAL Task 5: Extract raster values from points
sites <- read.csv("Turbine_sites.csv")
head(sites)

sites.sf <- st_as_sf(sites, coords=c("Lon","Lat"), crs=4326)  # EPSG
4326 is for Google Maps decimal degrees
sites.sf <- st_transform(sites.sf, crs = st_crs(abund.ny.max))
st_crs(sites.sf) == st_crs(abund.ny.max)
plot(abund.ny.max)
plot(st_geometry(sites.sf), pch=16, add=T)

## Produce a plot that zooms in on the project sites
ext <- zoom()  # the zoom() function is interactive
ext <- c(1470000, 1612000, 2278000, 2371000)
plot(crop(abund.ny.max, ext))
plot(st_geometry(sites.sf), pch=16, add=T)

## Extract raster values at points
extract(x = abund.ny.max, y = sites.sf)
```