Lab: Spatial data 2: Working with OBA data (Student Version)

Conservation/ecology Topics

- Species distributions

Computational Topics
- Convert a data frame to a spatial object.
- Plot multiple spatial layers.

```{r load-libraries-lab, echo=FALSE, results="hide", message=FALSE, warning=FALSE}
library(terra)
library(ggplot2)
library(dplyr)
library(sf)
library(ggspatial)
library(stringr)
```

Lab part 1: Oregon bee atlas data exploration

- 1a. Import the OBA data. (remember it's in the main data folder of the book and not in the labs folder, so you have to move back a folder)
```{r}
oba <- read.csv("../data/OBA_2018-2024.csv")
```

- 1b. Find the columns related to genus and species and paste them together (with a space between) using the function paste(). Name the new column GenusSpecies.
```{r}
colnames(oba)
oba <- oba %>%
  mutate(GenusSpecies = paste(genus, specificEpithet, " "))
```

- 1c. Use `sort()` and `unique()` to print the unique values of GenusSpecies in alphabetical order. How many species are there?
ANSWER: 587

```{r}
#sort(unique(oba$GenusSpecies))
length(sort(unique(oba$GenusSpecies)))
```

Some specimens are not identified to species, only genus. How is this reflected in the data?
ANSWER: In the data, the species name simply shows up as missing space (e.g. "Xylocopa   " vs. "Xylocopa virginica")

- 1d. So many bees, so little time. Count up the occurrences of each bee species, and subset the data to bees that have been seen at least two times.
You can use the tidyverse or any other functions in R that you like. How many "species" are there?

```{r}
# Will count those only with species in their genus
oba_counts <- oba %>%
  filter(specificEpithet != "") %>% # selects for bees with both genus and species name
  filter(GenusSpecies != "") %>% # selects for non-empty cells within column GenusSpecies
  mutate(GenusSpecies = str_trim(GenusSpecies)) %>% #trims white space in column GenusSpecies
  group_by(GenusSpecies) %>%
  summarise(count = n())

oba_counts <- oba_counts %>%
  filter(count >= 2)
```

```
#min(oba_counts$count)

oba_bees2 <- oba %>%
  mutate(GenusSpecies = str_trim(GenusSpecies)) %>%
  filter(GenusSpecies %in% oba_counts$GenusSpecies)
```

- 1e. Google a few bee names (that have been seen > 2 times) and find one with an a look that resonates with you.

What is the name of your bee? Osmia coloradensis

Import the photos into Rmarkdown below (hint: googling bee name "discover life" or "inat" can often get you a photo. Many bees will no have any photos :(use the format {alt='My spirt bee: ****'}

Lab part 2: Plotting the distribution of your spirit bee.

How that have chosen your spirit bee, we would like to plot it's distribution. What is the crs of the data? Annoyingly it is not described anywhere in the spreadsheet (always list your crs in your data) but it is the same as what inat uses because all bees have a georeferenced plant host. If the data is in lat long, it is "unprojected" so only a datum will be listed.
DATUM: WGS84, unprojected lat long. EPSG code: 4326.

```{r}
crs("EPSG:4326")
```

- 2a. Extract the X and Y locations for your species only from the data and create a spatial object. Don't forget to set the CRS!
Hint 1: consider what other data you would like to keep as attributes, for example what flower they were foraging on. Hint 2: Remember the lat is y and long is x.
Hint 3: You may want to rename the column names you can use, `colnames()` or `rename()` from dplyr and reassign the names. This is only if you don't love the names they already have.

```{r}
# Note to self: longitude is x, latitude is y

# Removing empty coordinates
oba_bees2 <- oba_bees2 %>%
  filter(GenusSpecies == "Osmia coloradensis") %>% # Extracting only the species we want
  filter(decimalLongitude != "") %>% # Removing empty coordinates
  filter(decimalLatitude != "")

# Creating spatial object
bee_spatial <- st_as_sf(oba_bees2,
                        coords = c("decimalLongitude", "decimalLatitude"),
                        crs = 4326)
```

- 2b. Plot your exciting bee data!

```{r plot-data-points-lab}
ggplot() +
  geom_sf(data = bee_spatial)
```

Not so exciting without some kind of background...

Luckily we can download basemaps into R using the map_data function in ggplot (among many others). There is an example for retrieving the Oregon county polygons.

```{r plot-or}
```

```
or <- map_data("county", "oregon") %>%
  select(lon = long, lat, group, id = subregion)
```

- 2c. Add your species's points to your choice or an Oregon basemap.

```{r plot-data-points-basemap-lab}
ggplot() +
  geom_polygon(data = or, mapping = aes(x = lon, y = lat, group = group),
               fill = "darkgreen",
               color = "white",
               alpha = 0.4,
               linewidth = 0.2) +
  geom_sf(data = bee_spatial, color = "black", size = 2) +
  coord_sf()
```

Lab part 3: Cartography

- 3a. Here is your moment to explore your cartographic skills.

1. Add the ecoregion shape file (in data/OR-ecoregions) and tweek the Oregon map in anyway that is useful/visually appealing. You may need to crop that layer to the extent of your species's distribution if it is only in a few ecoregions.
2. Color your points according to some data attribute and add a legend (month collected, county, collector, associated plant, whatever you think is interesting). You may need to circle back to 2.1 to save additional attributes when you converted the dataframe to a spatial object.
3. Fine-tune your map: add a title, make sure the legend label makes sense, add a scale bar (google "add scale bar map ggplot" and choose your favorite package). All maps must always have a scale bar. You can add a N arrow as well, though some cartographers argue that is only necessary if N isn't at the top of the map.
4. Write a figure caption for your map explaining any interesting trends you see.
5. Export your cropped layer to a .shp so you can use it again for your final project.

```{r plot-creative-lab}
# Reading in ecoregion shapefile
or_ecoregion <- st_read("../labs/data/OR-ecoregions/Ecoregions_OregonConservationStrategy.shp")

# Reproject
or_ecoregion <- st_transform(or_ecoregion, 4326)
crs(or_ecoregion)
```

```{r plot-creative-lab}
bee_spatial$sex <- as.factor(bee_spatial$sex)
```

```{r plot-creative-lab}
ggplot() +
  geom_polygon(data = or, mapping = aes(x = lon, y = lat, group = group),
               fill = "darkgreen",
               color = "white",
               alpha = 0.4,
               linewidth = 0.2) +
  geom_sf(data = or_ecoregion,
          fill = "darkgreen",
          alpha = 0.4,
          color = NA) +
  geom_sf(data = bee_spatial, aes(fill = sex), shape = 21) +
  scale_fill_manual(values = c("female" = "gold", "male" = "navy")) +
  coord_sf() +
  annotation_scale(location = "br")
  theme_bw() +
```

```
  labs(fill = "Sex of bees",
       title = "Osmia coloradensis observations in Oregon")
```
FIGURE CAPTION: Osmia coloradensis can be observed all over Oregon though they seem to be most observed closer to in the upper and lower parts of Oregon. Also, there seems to be more female Osmia coloradensis observed relative to male ones.

```{r}
st_write(bee_spatial,
         "data/Osmia_bee_spatial.shp",
         driver = "ESRI Shapefile", append=FALSE)
```

We are looking forward to seeing the maps you create!

Lab part 4: Spatial summary statistics
 For your final projects, you will likely need to come up with summary statistics that describes the areas around where bees are captured.

- 4a. Using the distribution of your chosen bee and the a raster spatial layer from [Oregon Geohub](https://geohub.oregon.gov/), extract a meaningful summary statistics from your spatial layer within a buffer of 500, 750 1000 m.

- 4b. Create a plot that illustrates this summary data (box plot, barplot, scatter plot, historgram).

- 4c. Create a map of your cropped spatial data.

```{r}
# Read in raster data (Oregon Average Annual Minimum Temperature, 1991-2020 (30 arc-second))
or_mintemp <- rast("data/OR_PRISM_tmin_30yr_normal_800m_annual/or_tmin_800m/w001001.adf")

# Reproject data to match
or_mintemp <- project(or_mintemp, "EPSG:4326")
crs(or_mintemp)
crs(or_ecoregion)

plot(or_mintemp)

hist(or_mintemp,
     main = "Oregon Average Annual Minimum Temperature (1991–2020)",
     xlab = "Minimum Temperature (°C)")

summary(values(or_mintemp))

# Making this new layer compatible with previous layers
crs(or_mintemp)

or_mintemp_df <- as.data.frame(or_mintemp, xy = TRUE) # converting to df to use with ggplot
names(or_mintemp_df) <- c("lon", "lat", "min_temp") # renaming layers for clarity
```

```{r}
# Adding into plot
ggplot() +
  geom_raster(data = or_mintemp_df,
              aes(x = lon,
                  y = lat,
                  fill = min_temp)) +
  scale_fill_viridis_c(option = "plasma", name = "Minimum Temperature (°C)") +
  geom_polygon(data = or,
               aes(x = lon, y = lat, group = group),
               fill = NA,
               color = "black",
               linewidth = 0.4) +
```

```
  geom_sf(data = or_ecoregion,
          fill = NA,
          linewidth = 0.4,
          color = "darkgreen") +
  geom_sf(data = bee_spatial,
          aes(color = sex),
          shape = 16,
          size = 2.5) +
  scale_color_viridis_d(option = "D", name = "Sex") +
  coord_sf() +
  annotation_scale(location = "tl") +
  labs(fill = "Sex of bees",
       title = expression(italic("Osmia coloradensis") ~ "observations in Oregon")) +
  theme_minimal()
```