

Spatial Analysis I

There are dozens of spatial analyses you can perform, but let's start with some basic ones. You'll continue working with the cleaned bat occurrence data for the mainland U.S. One of the most common and straightforward analyses is to count the number of species and total occurrences within each state.

First, load all the necessary libraries and import the data. Since the occurrence points were saved as an RDS file, you'll use the function `readRDS()` to load them.

```
library(sf)
library(raster)
library(dplyr)
library(ggplot2)
library(tidyr)

US_border <- read_sf("../shp/us-state-boundaries/us-state-boundaries.shp")

occ <- readRDS("../GBIF/occ_final.rds")
```

Now, you'll crop the U.S. polygon using a bounding box (bbox) to retain only the mainland states. Then, you'll transform the occurrences into a spatial object.

```
US_Crop <- st_bbox(c(xmin = -125, ymin = 22,
                    xmax = -66, ymax = 49),
                  crs = 4326)

US_border <- st_crop(US_border, US_Crop)

occ_sf <- st_as_sf(occ, coords = c("X", "Y"), crs = 4326)
occ_sf <- st_transform(occ_sf, st_crs(US_border)) # same projection with the polygon
```

Now, let's perform the count. There are several ways to do this, but in this case, you'll use a pipeline that simplifies the process.

This line of code creates a summary table that shows how many bat occurrences and how many different species were recorded in each U.S. state. It starts by using `st_join()` to combine the occurrence points (`occ_sf`) with the state polygons (`US_border`), so each point inherits information about the state it falls within. Then, `st_drop_geometry()` removes the spatial (coordinate) information, since it's not needed for the summary. The data is grouped by state and name, which allows us to count separately for each state. The `summarise()` function then calculates two things: the total number of records (`n_occurrences = n()`) and the number of unique species in each state (`n_species = n_distinct(species)`). Finally, `ungroup()` is used to remove the grouping structure, so the resulting data frame can be used more flexibly afterward.

```
state_counts <- occ_sf %>%
  st_join(US_border) %>% #joint both points and polygons
  st_drop_geometry() %>% # remove geometry (coordinates)
```

```

group_by(state, name) %>% # group all the information by state and name
summarise(
  n_occurrences = n(), #count the number of records
  n_species = n_distinct(species) # count unique species names
) %>%
ungroup()

state_counts

```

```

## # A tibble: 50 x 4
##   state name                n_occurrences n_species
##   <chr> <chr>                <int>      <int>
## 1 01    Alabama                842        12
## 2 04    Arizona                5553       35
## 3 05    Arkansas                1132       14
## 4 06    California            15202       37
## 5 08    Colorado                3454       22
## 6 09    Connecticut              112         8
## 7 10    Delaware                  6         3
## 8 11    District of Columbia        1         1
## 9 12    Florida                1335       16
## 10 13   Georgia                 250       12
## # i 40 more rows

```

This lines creates a new spatial object called `US_bats`, which contains the map of U.S. states along with the bat occurrence and species count data. It starts with `US_border`, which is the polygon layer of U.S. states, and uses `left_join()` to attach the summary data from `state_counts` by matching the state and name columns. This means that each state in the map will now include information about how many occurrences and species were found there. However, not all states may have bat data, so `replace_na()` is used to fill in zeros (0) for any missing values in the columns `n_occurrences` and `n_species`, instead of leaving them as NA (not available).

```

US_bats <- US_border %>%
  left_join(state_counts, by = c("state", "name")) %>%
  replace_na(list(n_occurrences = 0, n_species = 0))

```

```

US_bats$n_occurrences
US_bats$n_species

```

Now, you have a well-structured spatial object with biodiversity information (number of species and occurrences). So, let's create a nice plot using this new data.

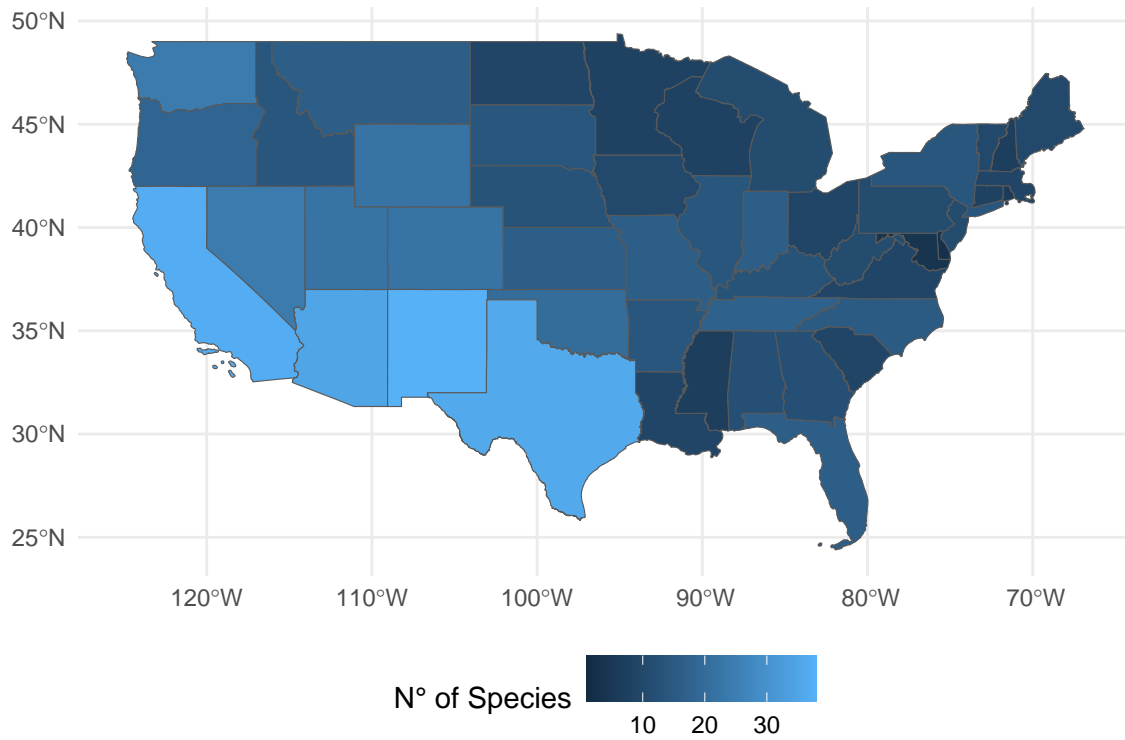
```

ggplot()+
  geom_sf(data = US_bats, aes(fill= n_species))+
  scale_fill_continuous(name="N° of Species")+
  theme_minimal()+
  theme(legend.position = "bottom")+
  labs(title = "Bat species by state",
        subtitle = "Data from GBIF")

```

Bat species by state

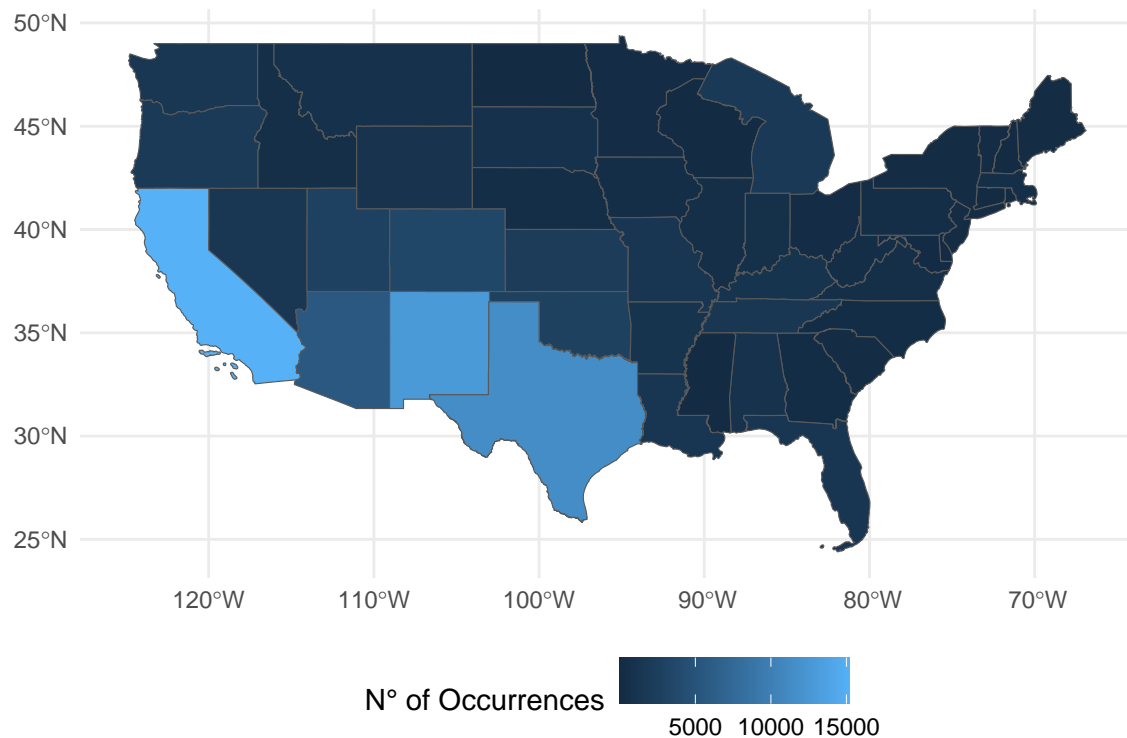
Data from GBIF



```
ggplot()+  
  geom_sf(data = US_bats, aes(fill= n_occurrences))+  
  scale_fill_continuous(name="N° of Occurrences")+  
  theme_minimal()+  
  theme(legend.position = "bottom")+  
  labs(title = "Bat occurrences by state",  
        subtitle = "Data from GBIF")
```

Bat occurrences by state

Data from GBIF



Finally, you can save this new spatial object to use it in other software like QGIS. You can also convert all the information into a regular table and save it as a CSV file.

```
st_write(US_bats, "../shp/US_bats.gpkg")

US_bata_data <- data.frame(State= US_bats$name,
                           Species = US_bats$n_species,
                           Occurrences = US_bats$n_occurrences)

head(US_bata_data)

write.csv(US_bata_data, "../GBIF/US_bats_data.csv", quote = F, row.names = F)
```