

Appendix S3 - Group Sites by County and Flyway

Nicholas J Clark, David J Harris, Ceridwen Fraser

This appendix describes how to extract flyway and county grouping variables for each BBS survey site. These variables can be accessed in the `BBS.occurrences` library using `data('Site.descriptors')`.

Load the cleaned BBS survey dataset that was prepared in **Appendix S1**

```
load("./BBS.Data.cleaned/BBS.2003-2009.filtered.Rdata")
```

Load the Landcover summary data for survey points that was prepared in **Appendix S2**

```
uniq.points.landcover <- read.csv("./LandCover/MODIS_Landcover_summary.csv")
```

Put landcover covariates and site descriptors into a single dataframe

```
library(dplyr)
Site.descriptors = sites.df %>% dplyr::left_join(uniq.points.landcover,
  by = c(Latitude = "lat", Longitude = "long", year = "year")) %>%
  dplyr::select(-modis.band, -richness, -simpsons.d,
    -simpsons.evenness, -no.data.fill)
```

Here we write a function to group sites into counties. This is accomplished by loading the US Government Census Cartographic Boundaries shapefile. Note, this file can be manually downloaded here: https://www.census.gov/geo/maps-data/data/cbf/cbf_counties.html if the `tempfile` strategy below fails. We then use a simple function to ensure that the coordinate system of our latitudes and longitudes match that of the shapefile and then intersect each unique point with the counties

```
# install.packages('sf')
library(sf)
library(dplyr)
temp <- tempfile()
download.file("http://www2.census.gov/geo/tiger/GENZ2017/shp/cb_2017_us_county_5m.zip",
  temp)
county.shp <- sf::st_read(unzip(temp, "cb_2017_us_county_5m.shp"))
unlink(temp)

geo_inside <- function(lon, lat, map, variable) {
  # Escape the variable name
  variable <- dplyr::enquo(variable)

  # Convert each unique point into an sf object
  pt <- tibble::data_frame(x = lon, y = lat) %>%
    sf::st_as_sf(coords = c("x", "y"), crs = 4269) %>%
    sf::st_transform(crs = sf::st_crs(county.shp))

  # Intersect the point with the map object
  pt %>% sf::st_join(map) %>% dplyr::pull(!variable)
}
```

Next we write a similar function to group sites into regions based on recognised North American migratory flyways. This is accomplished by loading the US Fish and Wildlife Service's **WaterfowlFlyways** shapefile. Note, this file can be manually downloaded here: <https://ecos.fws.gov/ServCat/Reference/Profile/42276> if the `tempfile` strategy below fails. We then use a simple function to ensure that the coordinate system of our latitudes and longitudes match that of the shapefile and then intersect each unique point with the flyway layer

```
# install.packages('sf')
library(sf)
library(dplyr)
temp <- tempfile()
download.file("https://ecos.fws.gov/ServCat/DownloadFile/113670",
  temp)
fly.shp <- sf::st_read(unzip(temp, "WaterfowlFlyways.shp"))
unlink(temp)

geo_inside <- function(lon, lat, map, variable) {
  # Escape the variable name
  variable <- dplyr::enquo(variable)

  # Convert each unique point into an sf object
  pt <- tibble::data_frame(x = lon, y = lat) %>%
    sf::st_as_sf(coords = c("x", "y"), crs = 4269) %>%
    sf::st_transform(crs = sf::st_crs(fly.shp))

  # Intersect the point with the map object
  pt %>% sf::st_join(map) %>% dplyr::pull(!variable)
}
```

Now that all of our site-level covariates are stored together, we can intersect each unique point with the shapefile above to group the sites into counties.

```
point_counties = Site.descriptors %>% dplyr::select(Latitude,
  Longitude) %>% dplyr::distinct() %>% dplyr::mutate(County = geo_inside(lon = lon,
  lat = lat, map = county.shp, variable = NAME))

Site.descriptors = Site.descriptors %>% left_join(point_counties,
  by = c(Latitude = "lat", Longitude = "lon"))
```

Repeat for flyways and then join the grouping information back to the descriptors data

```
point_site.groups = Site.descriptors %>% dplyr::select(lat,
  lon) %>% dplyr::distinct() %>% dplyr::mutate(site.group = geo_inside(lon = lon,
  lat = lat, map = fly.shp, variable = NAME))

Site.descriptors = Site.descriptors %>% left_join(point_site.groups,
  by = c(Latitude = "lat", Longitude = "lon"))
colnames(Site.descriptors) <- stringr::str_to_title(colnames(Site.descriptors))
```

Save the grouped descriptor data for downstream analysis. Note, the flyway variables can be directly accessed in the `BBS.occurrences` library using `data('Site.descriptors')`.

```
dir.create("./Analysis_data")
save(Site.descriptors, file = "./Analysis_data/Site.descriptors.Rdata")
```