# NEON data formatting

Tad Dallas and Eric Sokol

## Contents

`data.frame` with the following columns and date ranges:

- year (2014-2019 for training data, 2020 for test data)
- season (3 month spans 1-3, 4-6, 7-9, 10-12 month)
- latitude and longitude
- species richness (per site)
- abundance (mean plot estimate per site, standardized by trapping effort)

maybe? + plant diversity at the site per season/year + annual-level variation that can capture spatial differences (e.g., mean annual temperature and mean annual precipitation, temperature seasonality, etc. for each year if available). +

Set working directory to neonCodeFest root dir so relative filepaths work

```
my_wd <- "~/GitHub/neonCodeFest" #edit this based on what your path to the cloned repo
setwd(my_wd)
```

Load packages. Install `ecocomDP` from CRAN if necessary

```
# Load packages
# install.packages('ecocomdp')
library(ecocomDP)
```

```
## Warning: package 'ecocomDP' was built under R version 4.0.5
```

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------- tidy

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 4.0.5

## -- Conflicts ---------------------------------------------------------------------------- tidyverse_
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Use ecocomDP functions to find the mammal data set

```
# how to find the mammal data in ecocomDP
data_list <- ecocomDP::search_data("mammal")
View(data_list)
```

After identifying the data package id for hte NEON small mammal data set, read it in to your R environment, and save an archive of the data as downloaded.

```
# the id for the neon small mammal data mapped to ecocomDP
my_id <- "neon.ecocomdp.10072.001.001"

# read neon data --
#   for the two sites specified below, download size is ~52 MB.
#   for all NEON sites, download size is ~ 595 MB
my_ecocomDP_data <- ecocomDP::read_data(
  id = my_id,
  site = c("CPER","OSBS"), #comment out this line to download all sites
  # token = Sys.getenv("NEON_TOKEN"), #uncomment to use NEON_TOKEN if you have it set up
  check.size = FALSE)
```

```
## Finding available files
##   |                                                                                |
```

```
# make sure wd is correct
setwd(my_wd)

# save data initial download locally (RDS file is only 39.7KB)
saveRDS(my_ecocomDP_data, "data/my_ecocomDP_data.RDS")
```

Read the locally save data, flatten, and calculate species richness and abundance standardized to 100 trapnights by season by year by site.

```
# make sure wd is correct
setwd(my_wd)

# read in data from local file
my_ecocomDP_data <- readRDS("data/my_ecocomDP_data.RDS")


# flatten data
data_flat <- my_ecocomDP_data[[1]]$tables %>%
  ecocomDP::flatten_data() %>%
  mutate(
    season = case_when(
      month %in% c(1:3) ~ "winter",
      month %in% c(4:6) ~ "spring",
      month %in% c(7:9) ~ "summer",
      month %in% c(10:12) ~ "fall",
      TRUE ~ NA_character_)) %>%
  dplyr::select(
    observation_id, event_id,
    year, season,
    siteID, domainID, nlcdClass,
    taxon_rank,
    taxon_id,
    variable_name, value, unit) %>%
  dplyr::filter(value > 0)
```

```
## Joining, by = "location_id"
```

```
# average counts for each taxon_id within a season
data_by_site_year_season_taxon <- data_flat %>%
  group_by(siteID, year, season, taxon_id) %>%
  summarize(
    count_per_100_trapnight = mean(value)) %>%
  ungroup()
```

```
## `summarise()` regrouping output by 'siteID', 'year', 'season' (override with `.groups` argument)
```

```
# sum average counts for each taxon in each season to
# get estimate of total small mammal count per trapping effort
data_by_site_year_season <- data_by_site_year_season_taxon %>%
  dplyr::filter(count_per_100_trapnight > 0) %>%
  group_by(siteID, year, season) %>%
  summarize(
    total_count_per_100_trapnight = sum(count_per_100_trapnight),
    richness = length(unique(taxon_id)))
```

```
## `summarise()` regrouping output by 'siteID', 'year' (override with `.groups` argument)
```

Read in NEON site information from the NEON website and merge with the small mammal table, and save in the `data/` subdirectory in this repo.

```
# get site info data
site_info <- read_csv(file = "https://www.neonscience.org/sites/default/files/NEON_Field_Site_Metadata_
```

```
## Parsed with column specification:
```

```
## cols(
##   .default = col_character(),
##   field_latitude = col_double(),
##   field_longitude = col_double(),
##   field_utm_northing = col_double(),
##   field_utm_easting = col_double(),
##   field_mean_elevation_m = col_double(),
##   field_minimum_elevation_m = col_double(),
##   field_maximum_elevation_m = col_double(),
##   field_mean_annual_temperature_C = col_double(),
##   field_mean_annual_precipitation_mm = col_double(),
##   field_watershed_size_km2 = col_double(),
##   field_lake_depth_mean_m = col_double(),
##   field_lake_depth_max_m = col_double(),
##   field_tower_height_m = col_double(),
##   field_avg_number_of_green_days = col_double(),
##   field_number_tower_levels = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
# rename columns
names(site_info) <- names(site_info) %>% gsub("field_","",.)

site_info <- site_info %>%
  dplyr::select(
    domain_id, site_id, site_name,
    site_type,
    latitude, longitude,
    mean_elevation_m, minimum_elevation_m, maximum_elevation_m,
    mean_annual_temperature_C, mean_annual_precipitation_mm,
    dominant_wind_direction,
    mean_canopy_height_m, dominant_nlcd_classes, domint_plant_species,
    usgs_geology_unit, megapit_soil_family, soil_subgroup,
    avg_number_of_green_days, avg_green_max_doy)

# combine site info data with small mammal data
data_merged <- data_by_site_year_season %>%
  left_join(
    site_info,by = c("siteID" = "site_id"))

# make sure wd is correct
setwd(my_wd)

# write to the data subdir
write_csv(data_merged, "data/data.csv")
```