Final Project

Paul Markley

2022-11-29

Preamble

This script will take in any dataframe with a column 'species' that also has the columns latitude, longitude, year, month, and date. It then cleans the species identity based on the World Flora Online (WFO) taxonomic backbone. After, the script plots the counts of species and genus in the data frame, followed by spatial plotting. It lastly runs a mantel randomization test on the date and locations to see if there is any correlation between the space and time dimensions of the data.

This script is intended to be used with a dataset downloaded from GBIF.

Below is a list of packages needed.

```
if (!require("data.table", quietly = TRUE))
  install.packages("data.table")
if (!require('tidyverse', quietly = T))
  install.packages('tidyverse')
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
if (!require('ade4', quietly = T))
  install.packages('ade4')
if (!require('sf', quietly = T))
  install.packages('sf')
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
if (!require('terra', quietly = T))
  install.packages('terra')
```

```
## terra 1.6.17
##
## Attaching package: 'terra'
##
## The following object is masked from 'package:tidyr':
```

```
##
##     extract
##
## The following object is masked from 'package:data.table':
##
##     shift
```

```
if (!require('WorldFlora', quietly = T))
  install.packages('WorldFlora')
```

```
## WorldFlora 1.11: Use function WFO.match to check plant names;
##
## First you need to download and unzip the World Flora Online taxonomic backbone from
## www.worldfloraonline.org/downloadData;
##
## Use functions WFO.download and WFO.remember to download and reload the backbone data;
##
## Package RcmdrPlugin.WorldFlora provides a graphical user interface for this package.
```

Data Step

Now the data will be read in. These files probably are not in the repository since they are too big for GitHub. These will be called from the local location after setting the working directory.

```
#setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
df <- read.csv('/Users/pm/COURSES/COMPBIO/grad-project/data/heaths.csv') # On onedrive
back <- data.table::fread("/Users/pm/COURSES/COMPBIO/grad-project/data/classification.txt") # on onedri
```

```
## Warning in data.table::fread("/Users/pm/COURSES/COMPBIO/grad-project/data/
## classification.txt"): Found and resolved improper quoting out-of-sample. First
## healed line 118: <<wfo-0000000117 GCC-1AF25765-5E36-4AED-8F64-12BB4F58DEC0
## Hieracium onosmoides subsp. sphaerianthum SUBSPECIES wfo-0000034880 (Arv.-
## Touv.) Zahn Asteraceae Hieracium onosmoides sphaerianthum subsp. "Zahn, in
## Engler, Pflanzenr. 82. 1923." 1676 1923 ref-0000000117 Accepted wfo-0000118008
## More details could be found in <a href=http://www.theplantlist.org/tpl1.1/
## record/gcc-10011 >The Plant List v.1.1.</a> Originally in <a href=http://
## www.theplantlist.org/tpl/record/gcc-10011 >The Plant List v.1>>. If the fields
## are not quoted (e.g. field separator does not appear within any field), try
## quote="" to avoid this warning.
```

Taxonomic Resolution

After reading in the data I create the intersection of the species vectors. It is the list of species IDs that are OK and don't need correction. Using the function, I pick the rows from the clean.sp function that don't match any of the values in the intersected vector and call the unique function on that vector. Then the clean.sp function uses WFO.match's fuzzy matching algorithms to match species that may have been misspelled or ones with outdated names with the correct name. This vector is then matched back to the extracted rows and integrated back into the input data frame, which is returned at the end of this function.

```
# remove rows that have matched with backbone, then gets all unique species names
# takes these species names and matches with accepted taxonomy.
S <- intersect(df$species, back$scientificName)
source('/Users/pm/COURSES/COMPBIO/grad-project/scripts/cleansp.R')
df <- clean.sp(df, sp.list = S)
```

```
## Fuzzy matches for  Arctostaphylos alpinus were:  Arctostaphylos alpina, Arctostaphylos spinulosa
## Best fuzzy matches for  Arctostaphylos alpinus were:  Arctostaphylos alpina
## Fuzzy matches for  Vaccinium intermedium were:  Megaclinium intermedium, Vaccinium ×intermedium, ×Rh
## Best fuzzy matches for  Vaccinium intermedium were:  Vaccinium ×intermedium
## Fuzzy matches for Pyrola minor × rotundifolia were only found for first 2 terms
## Fuzzy matches for  Pyrola minor × rotundifolia were:  Pyrola monophyla, Pyrola minor, Pyrola major, P
## With Fuzzy.two, reduced matches to those of 2 words only
## Best fuzzy matches for  Pyrola minor × rotundifolia were:  Pyrola morrisonensis
## Fuzzy matches for Vaccinium microcarpum × oxycoccos were only found for first 2 terms
## Fuzzy matches for  Vaccinium microcarpum × oxycoccos were:  Vaccinium pterocarpum, Vaccinium macroca
## With Fuzzy.two, reduced matches to those of 2 words only
## Best fuzzy matches for  Vaccinium microcarpum × oxycoccos were:  Vaccinium microcarpum
## Fuzzy matches for  Vaccinium macrocarpum were:  Vaccinium pterocarpum, Vaccinium lasiocarpum, Vaccini
## Best fuzzy matches for  Vaccinium macrocarpum were:  Vaccinium microcarpum
##
##  Checking new accepted IDs
```

```
# Final check: if it worked the number of species matched to the backbone should
# equal to the number of species in the dataset and prints TRUE.
S <- intersect(df$species, back$scientificName); length(S) == length(unique(df$species))
```

```
## [1] TRUE
```

Date of Year Calculation

I calculate the date of year (DOY) by calling lubridate's yday after collapsing the year, month, and day columns in the data frame into one column and updating the vector's class to Date. This will be used later in the script and for plotting the temporal trends in the data.

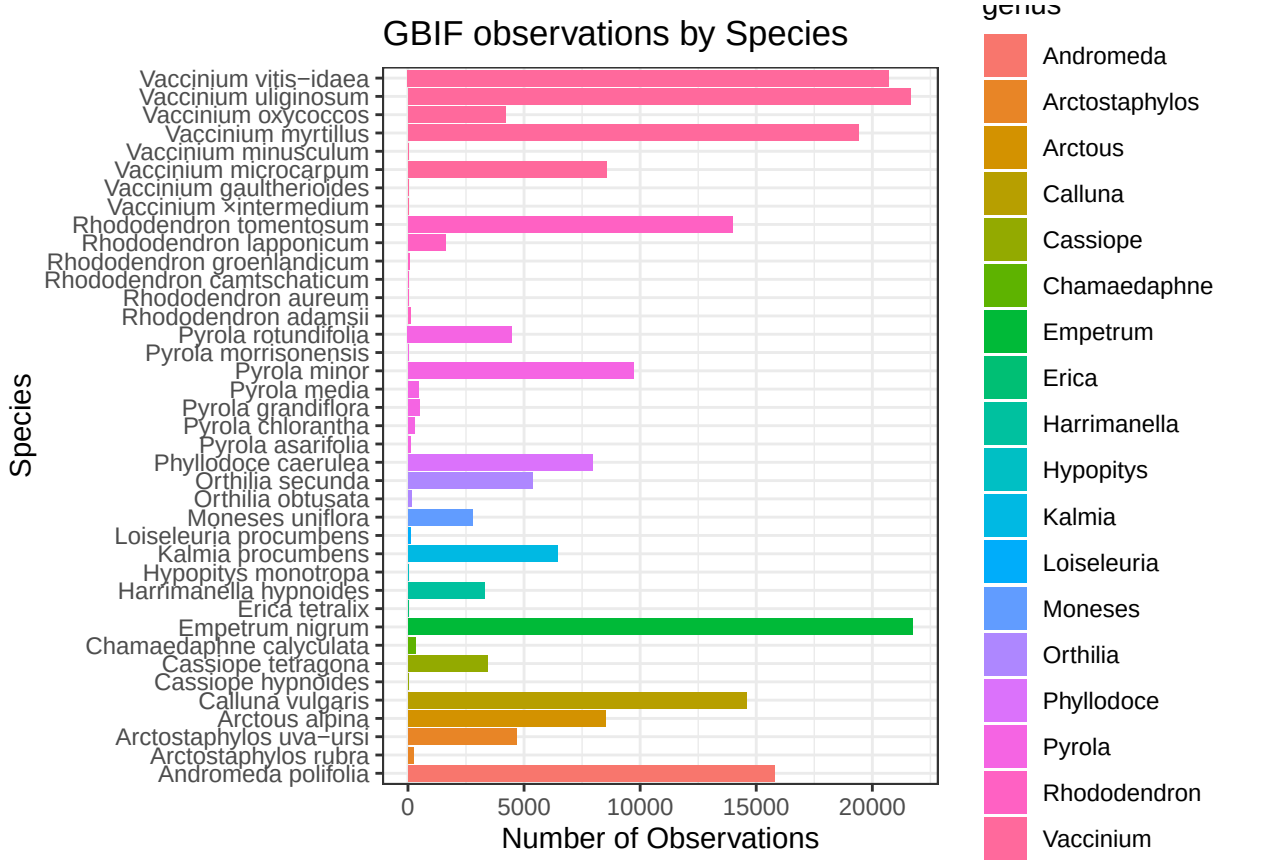
```
# Convert the YMD columns to DOY for plotting and brief analysis.
DOY = lubridate::yday(as.Date(paste0(df$year, '-', df$month, '-', df$day)))
df$DOY = DOY
```

Plotting

I want to look at the species and genus observation totals. So I call my second function in the freqsptbl.R file. It's a simple function that calls the table function on the string vector and then creates a genus column from the species binomial column. It also includes a function to filter the table by number of observations.

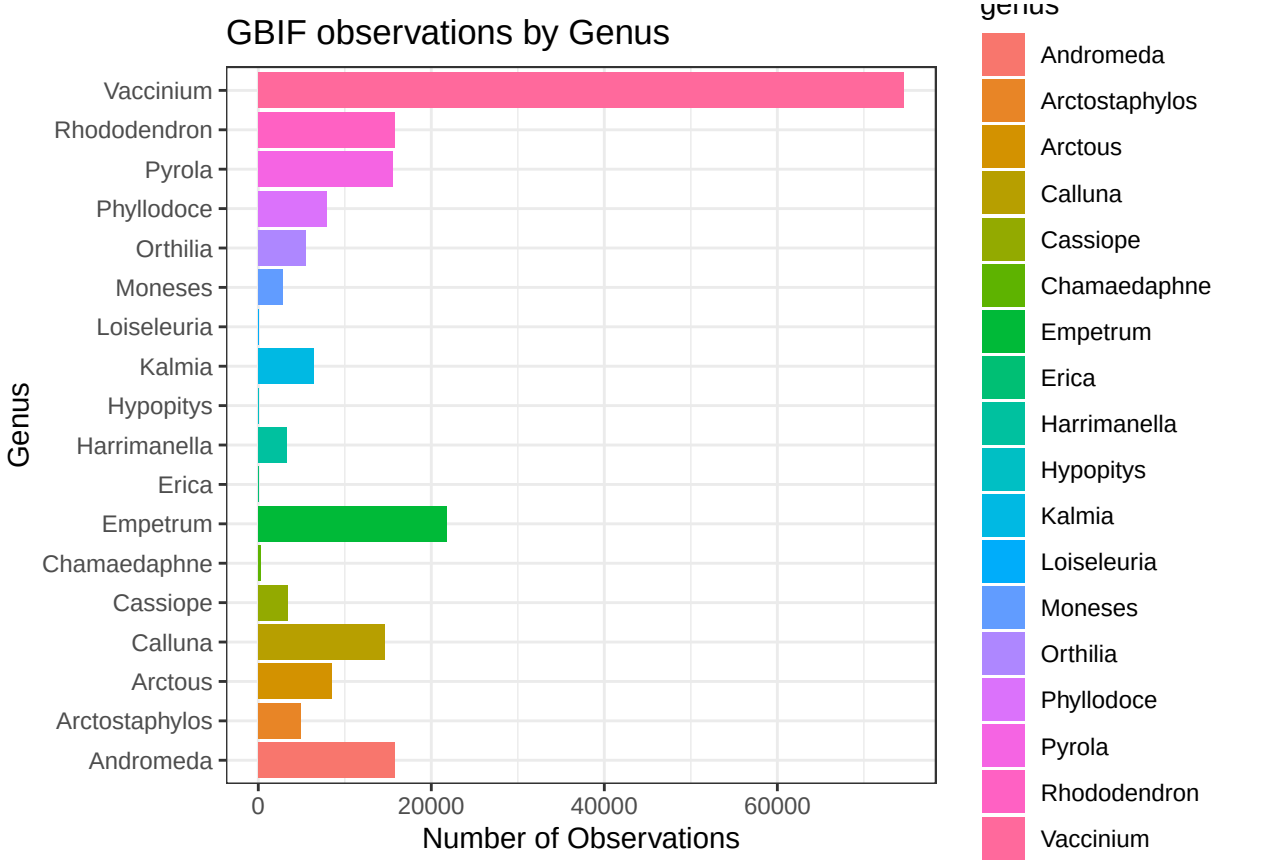
```
# Making a few plots to visualize what happened here in the data.
library(tidyverse)

# This script contains a function that summarizes the data frame into a table by species
# It then makes a genus column from the species binomial.
source('/Users/pm/COURSES/COMPBIO/grad-project/scripts/freqsptbl.R')
tbl <- freq.sp.tbl(df)
```

```
# Now the plots
# First is frequency of species colored by genus
 ggplot(tbl, aes(y = Var1, x = Freq)) +
   geom_col(aes(fill = genus)) +
   theme_bw() +
   labs(y='Species', x= 'Number of Observations', title = 'GBIF observations by Species')
```




```
#ggsave('../figures/Speciestbl.png')
```

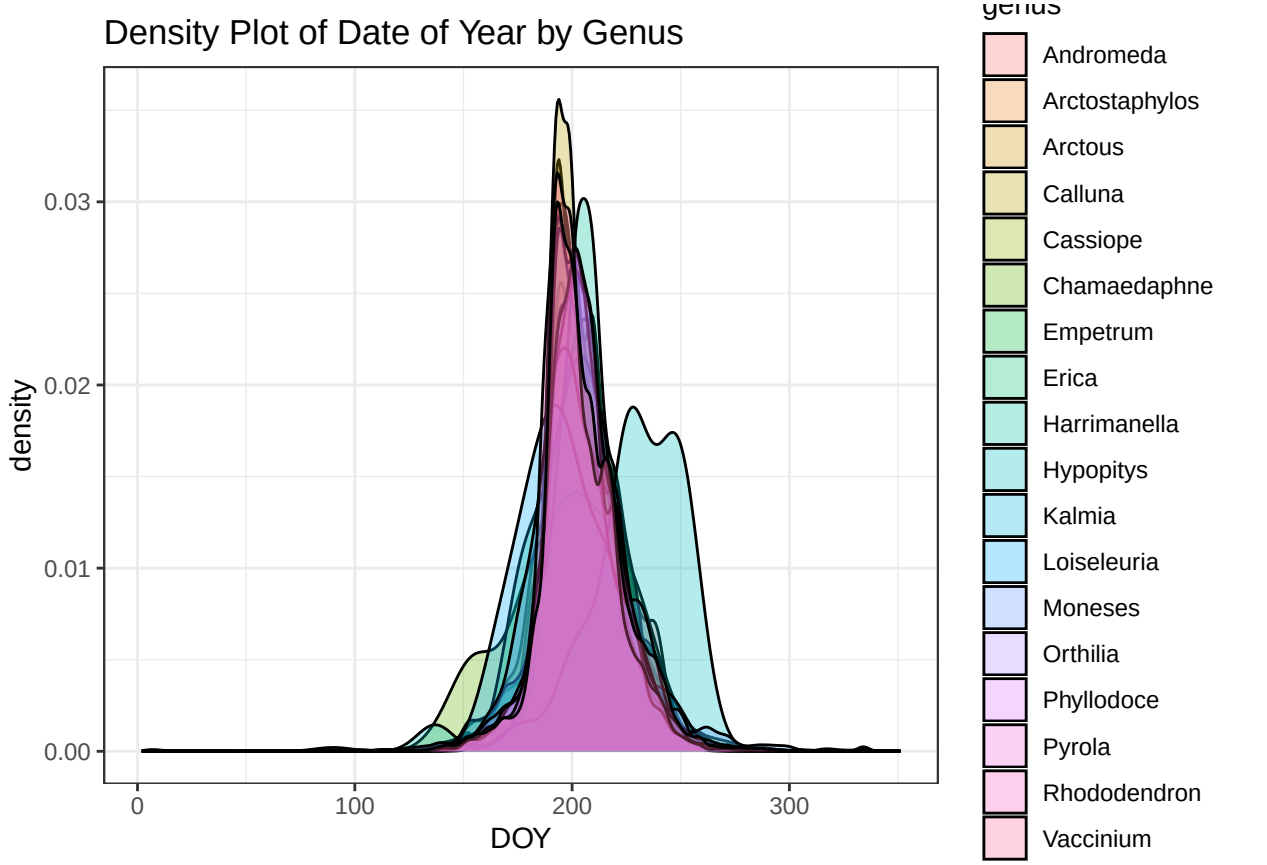
```
#Second is just genus frequencies
ggplot(tbl, aes(y = genus, x = Freq)) +
 geom_col(aes(fill = genus)) +
 theme_bw() +
 labs(y='Genus',x = 'Number of Observations', title = 'GBIF observations by Genus')
```




```
#ggsave('../figures/genustbl.png')
```

```
# Third is a density plot of DOY by genus
ggplot(data = df, aes(x = DOY, fill = genus)) +
geom_density(alpha = 0.3) +
theme_bw() +
labs(title = 'Density Plot of Date of Year by Genus')
```

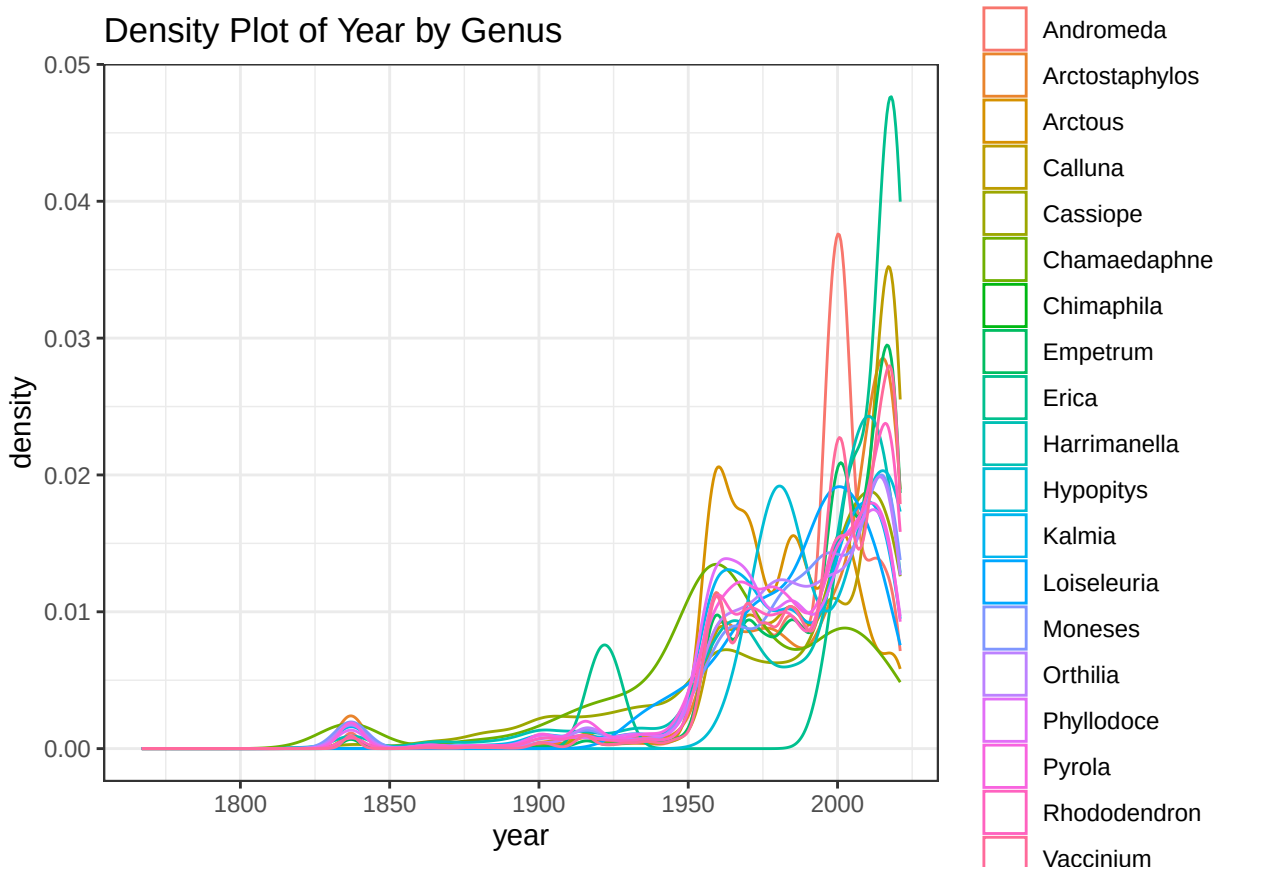
```
## Warning: Removed 16470 rows containing non-finite values (`stat_density()`).
```




```
#ggsave('../figures/densdoy.png')

#Fourth is a density plot of year by genus.
ggplot(data = df, aes(year, color = genus)) +
   geom_density() +
   labs(title = 'Density Plot of Year by Genus') +
  theme_bw()
```

```
## Warning: Groups with fewer than two data points have been dropped.
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```

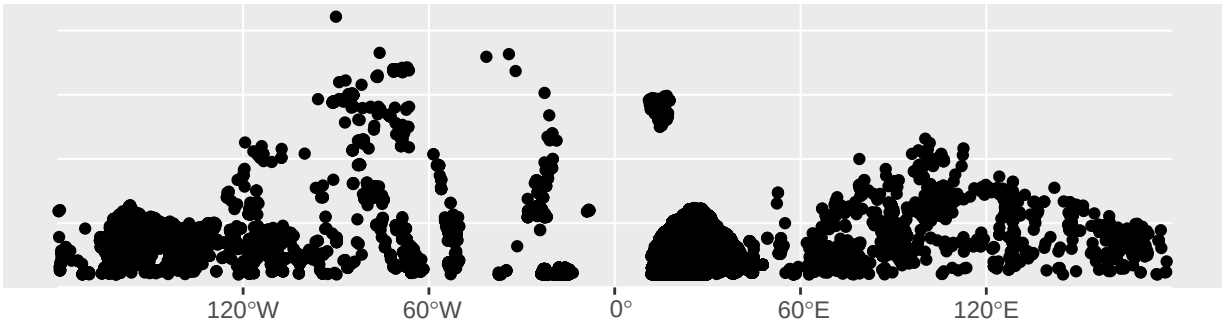



```
#ggsave('../figures/densyear.png')
```

Spatial Plots

Here I visualize the spatial patterns in the data to answer the question of: where are the most observations in the dataset?

```
# Making GIS maps. This requires the sf and terra packages.
# I make from the dataframe a sf object in WGS84
sf1 <- sf::st_as_sf(df, coords = c('lon','lat'), crs = 'EPSG:4326')
ggplot(data = sf1)+
geom_sf()
```

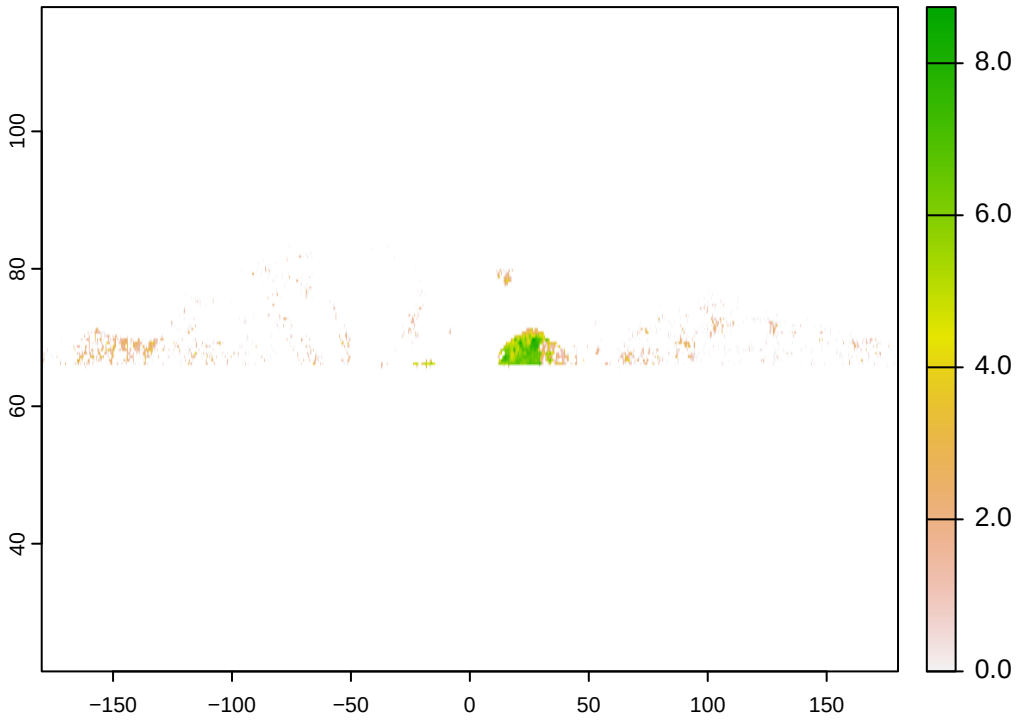



```
#ggsave('../figures/map.png')
```

The figure that I made with geom_sf() is great, but it's hard to read. I have here next a rasterization of the

data points. What is happening here is that I am binning number of points in each grid cell to produce a value to be plotted onto the map. The distribution of this new data set can be considered akin to a zero inflated Poisson, so I logged the values to make them a bit more readable and normally distributed.

```
#Next I make a blank raster to rasterize the number of observations within each cell.
#The resolution of each cell is 50km by 50km, roughly a degree by a degree.
rst <- terra::rast(nrows = 57, ncols = 720, nlyrs = 1, resolution = c(0.5,0.5),
                   xmin = -180, xmax = 180, ymin = 55.5, ymax = 84, crs = 'EPSG:4326')

# I take the log of the observations to make the plot a bit easier to read.
abd <- log(terra::rasterize(terra::vect(sf1), rst, fun=length, touches = T))
plot(abd)
```




```
#png(filename = '../figures/rasterlogobs.tif', width = 500, height = 275)
#plot(abd)
#dev.off()
```

Analysis

For the analysis I wanted to examine the relationship between location and time, or the correlation between the spatial and temporal dimensions. I did this first by making a distance matrix and then computing the Mantel test statistic from the ade4 package.

```
# My dataset is too big, so I will reduce it with a predetermined subset of observations
# Original dataset: ~20,000 observations, restricted data has ~4,000
# These were based on whether or not the points were within a polygon.
df1 <- df[df$in_s %in% 'PM',]
df1 = df1[complete.cases(df1) == T,]

# Minor Analysis of DOY vs Location and Year vs Location using the Mantel randomization test
d1 <- dist(cbind(df1$lon, df1$lat))
d2 <- dist(df1$DOY)
```

```r
d3 <- dist(df1$year)

(mtl <- ade4::mantel.rtest(d1, d2))
```

```
## Warning in is.euclid(m1): Zero distance(s)
```

```
## Warning in is.euclid(m2): Zero distance(s)
```

```
## Warning in is.euclid(distmat): Zero distance(s)
```

```
## Monte-Carlo test
## Call: ade4::mantel.rtest(m1 = d1, m2 = d2)
##
## Observation: 0.02663408
##
## Based on 99 replicates
## Simulated p-value: 0.01
## Alternative hypothesis: greater
##
##       Std.Obs   Expectation      Variance
##  5.448256e+00 -1.472081e-04  2.416284e-05
```

```r
saveRDS(mtl, '/Users/pm/COURSES/COMPBIO/grad-project/figures/mantel_out.RData')
# H0: Location and DOY are not linearly correlated.
# Ha: Location and DOY have some linear correlation.
# pvalue = 0.01 < alpha 0.05: reject null hypothesis
# There is sufficient evidence to conclude that location and DOY are significantly positively correlate

(mtl2 <- ade4::mantel.rtest(d1,d3))
```

```
## Warning in is.euclid(m1): Zero distance(s)
```

```
## Warning in is.euclid(m2): Zero distance(s)
```

```
## Warning in is.euclid(distmat): Zero distance(s)
```

```
## Monte-Carlo test
## Call: ade4::mantel.rtest(m1 = d1, m2 = d3)
##
## Observation: 0.03558159
##
## Based on 99 replicates
## Simulated p-value: 0.01
## Alternative hypothesis: greater
##
##       Std.Obs   Expectation      Variance
##  9.169707e+00 -2.970679e-04  1.530952e-05
```

```r
saveRDS(mtl2, '/Users/pm/COURSES/COMPBIO/grad-project/figures/mantel_out2.RData')
# H0: Location and year are not linearly correlated.
# Ha: Location and year have some linear correlation.
# pvalue = 0.01 < alpha 0.05: reject null hypothesis.
# There is sufficient evidence to conclude that location and year are significantly positively correlat
```