ECOL 620 Lab 1

Natalie Schmer

1/21/2021

ECOL 620 Lab 1

save.image(file = "ecol620lab1.RData")  
load(file = "ecol620lab1.RData")

# Load in packages

{  
 library(tidyverse)  
 library(rgdal)  
 library(viridis)  
 library(tidylog)  
 library(ggthemes)  
 library(ggrepel)  
 library(here)  
 library(sp)  
}

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.2 ✓ purrr 0.3.4  
## ✓ tibble 3.0.4 ✓ dplyr 1.0.2  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

## Loading required package: sp

## rgdal: version: 1.5-18, (SVN revision 1082)  
## Geospatial Data Abstraction Library extensions to R successfully loaded  
## Loaded GDAL runtime: GDAL 3.1.1, released 2020/06/22  
## Path to GDAL shared files: /Users/natalieschmer/Library/R/4.0/library/rgdal/gdal  
## GDAL binary built with GEOS: TRUE   
## Loaded PROJ runtime: Rel. 6.3.1, February 10th, 2020, [PJ\_VERSION: 631]  
## Path to PROJ shared files: /Users/natalieschmer/Library/R/4.0/library/rgdal/proj  
## Linking to sp version:1.4-4  
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,  
## use options("rgdal\_show\_exportToProj4\_warnings"="none") before loading rgdal.

## Loading required package: viridisLite

##   
## Attaching package: 'tidylog'

## The following objects are masked from 'package:dplyr':  
##   
## add\_count, add\_tally, anti\_join, count, distinct, distinct\_all,  
## distinct\_at, distinct\_if, filter, filter\_all, filter\_at, filter\_if,  
## full\_join, group\_by, group\_by\_all, group\_by\_at, group\_by\_if,  
## inner\_join, left\_join, mutate, mutate\_all, mutate\_at, mutate\_if,  
## relocate, rename, rename\_all, rename\_at, rename\_if, rename\_with,  
## right\_join, sample\_frac, sample\_n, select, select\_all, select\_at,  
## select\_if, semi\_join, slice, slice\_head, slice\_max, slice\_min,  
## slice\_sample, slice\_tail, summarise, summarise\_all, summarise\_at,  
## summarise\_if, summarize, summarize\_all, summarize\_at, summarize\_if,  
## tally, top\_frac, top\_n, transmute, transmute\_all, transmute\_at,  
## transmute\_if, ungroup

## The following objects are masked from 'package:tidyr':  
##   
## drop\_na, fill, gather, pivot\_longer, pivot\_wider, replace\_na,  
## spread, uncount

## The following object is masked from 'package:stats':  
##   
## filter

## here() starts at /Users/natalieschmer/Desktop/GitHub/ECOL\_620

#need to download old usmap   
install.packages("/Users/natalieschmer/Desktop/GitHub/ECOL\_620/data/usmap\_0.5.1.tar", repos = NULL, type="source")

## Installing package into '/Users/natalieschmer/Library/R/4.0/library'  
## (as 'lib' is unspecified)

# 1. Briefly describe (1-2 sentences) what each of the following functions achieve. All of these functions are seen throughout the coding exercise.

* c() is for concatenate, and creates a list or vector.
* geom\_smooth() is in {ggplot2} and draws a line through data to help show the pattern of the data. Need to specify the formula and fit arguments.
* rep() replicates values as specified in the arguments.
* filter\_by() in {dplyr} subsets rows using values in columns
* %>%() is a pipe (from {magrittr})! Moves whatever object is on the left to the operation on the right.
* png() writes an object (like a plot) to a .png file.
* coord\_map() is for projection in {ggplot}.
* plot\_grid() from {cowplot} allows for arranging multiple plots in a grid.

# US Maps

## 2. Ploting the contiguous U.S. states colored by area

# Load the data we used earlier  
states <- as.data.frame(state.x77)  
states$region <- tolower(rownames(states))  
states\_map <- map\_data("state")  
fact\_join <- left\_join(states\_map, states, by = "region")

## left\_join: added 8 columns (Population, Income, Illiteracy, Life Exp, Murder, …)

## > rows only in x 10

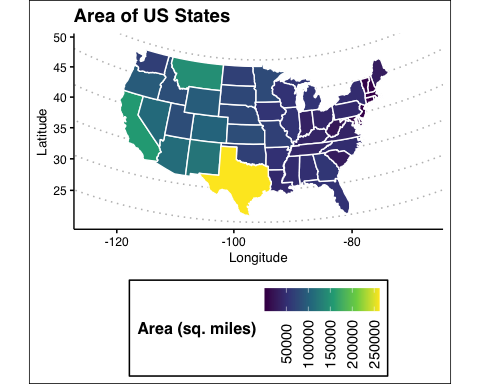
## > rows only in y ( 2)

## > matched rows 15,527

## > ========

## > rows total 15,537

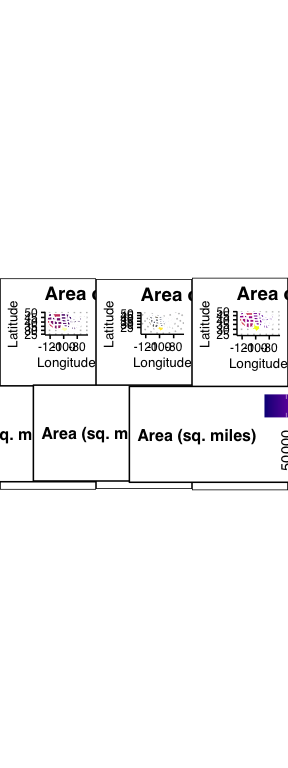
(state\_areas <- ggplot(fact\_join, aes(long, lat, group = group))+  
 geom\_polygon(aes(fill = Area), colour = "white")+  
 scale\_fill\_viridis\_c(option = "D")+  
 ggthemes::theme\_clean() +  
 coord\_map("bonne", lat0 = 40)+  
 labs(y = "Latitude",   
 x = "Longitude",   
 fill = "Area (sq. miles)",  
 title = "Area of US States")+  
 theme(legend.position = "bottom",  
 legend.text = element\_text(angle = 90, vjust = 0.5))  
)



## 3 Three additional plots of the same data, each with different color scale and geographic projection, put together with plot\_grid function to make a three-panel horizontal plot.

### 3-panel of the different color scales and projections

cowplot::plot\_grid(state\_areas\_1, state\_areas\_2, state\_areas\_3, nrow = 1)



# US University Data

## 4. Plot Colorado colleges and universities within the top

96th quantile of total student enrollment, add in geom\_text\_repel(data=NAME\_OF\_YOUR\_FILTERED\_DATA, aes(label=NAME, x=LON, y = LAT), force=20, size=2)

# Read in the csv of university data   
colo\_unis <- read.csv(here::here("data/data\_for\_lab1/Colleges\_and\_Universities/colorado\_universities.csv"))   
  
# Filter to only CO   
colo\_unis <- colo\_unis %>%   
 filter(LSTATE == "CO")

## filter: removed 7,014 rows (98%), 112 rows remaining

# check   
unique(colo\_unis$LSTATE)

## [1] "CO"

# Calculate and filter the 96th quantile universities  
quan\_96 <- quantile(colo\_unis$TOT\_ENROLL, .96, na.rm = T)  
  
co\_96 <- colo\_unis %>%   
 filter(TOT\_ENROLL >= quan\_96)

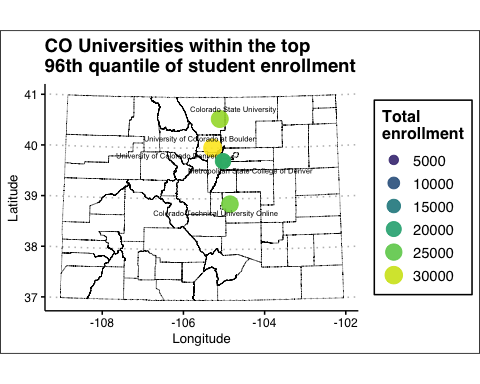
## filter: removed 107 rows (96%), 5 rows remaining

# Load in the CO shapefile   
co\_counties= readOGR(here::here("data/data\_for\_lab1/counties/Colorado\_County\_Boundaries.shp"))

## OGR data source with driver: ESRI Shapefile   
## Source: "/Users/natalieschmer/Desktop/GitHub/ECOL\_620/data/data\_for\_lab1/counties/Colorado\_County\_Boundaries.shp", layer: "Colorado\_County\_Boundaries"  
## with 64 features  
## It has 9 fields  
## Integer64 fields read as strings: OBJECTID NUM\_FIPS

# Set the min and max enroll  
min\_enroll=min(colo\_unis$TOT\_ENROLL)  
max\_enroll=max(colo\_unis$TOT\_ENROLL)  
  
# Plot   
(CO\_MAP\_UNI <- ggplot() +  
 geom\_polygon(data = co\_counties, aes(x=long, y = lat, group = group), fill = NA, color ="black", lwd=.1) +  
 geom\_point(data = co\_96, aes(x=LON, y = LAT, size=TOT\_ENROLL, colour=TOT\_ENROLL), alpha=.9) +  
 coord\_map("bonne", lat0 = 40)+  
 theme\_bw()+  
 theme(panel.grid.minor=element\_blank(),panel.grid.major=element\_blank())+  
 theme( panel.border=element\_blank())+  
 scale\_color\_viridis(limits=c(min\_enroll, max\_enroll), breaks=seq(5000, 30000, by=5000), name = "Total\nenrollment")+  
 guides(color= guide\_legend(), size=guide\_legend())+  
 scale\_size\_continuous(limits=c(min\_enroll, max\_enroll), breaks=seq(5000, 30000, by=5000),name = "Total\nenrollment") +  
 geom\_text\_repel(data = co\_96, aes(label=NAME, x=LON, y = LAT), force=20, size=2)+  
 ggthemes::theme\_clean()+  
 labs(y = "Latitude",   
 x = "Longitude",  
 title = "CO Universities within the top\n96th quantile of student enrollment")  
)

## Regions defined for each Polygons



## 5. Other universities: On, Wisconsin!

Wisconsin county boundary shapefile downloaded from <https://data-wi-dnr.opendata.arcgis.com/datasets/county-boundaries-24k>

# load in shapefiles for WI counties and universities  
wi\_counties <- readOGR(here::here("data/County\_Boundaries\_24K-shp/County\_Boundaries\_24K.shp"))

## OGR data source with driver: ESRI Shapefile   
## Source: "/Users/natalieschmer/Desktop/GitHub/ECOL\_620/data/County\_Boundaries\_24K-shp/County\_Boundaries\_24K.shp", layer: "County\_Boundaries\_24K"  
## with 72 features  
## It has 7 fields

us\_uni <- readOGR(here::here("data/data\_for\_lab1/Colleges\_and\_Universities/CollegesUniversities.shp"))

## OGR data source with driver: ESRI Shapefile   
## Source: "/Users/natalieschmer/Desktop/GitHub/ECOL\_620/data/data\_for\_lab1/Colleges\_and\_Universities/CollegesUniversities.shp", layer: "CollegesUniversities"  
## with 7126 features  
## It has 38 fields

# Filter out WI schools  
wi\_uni <- subset(us\_uni, LSTATE=="WI")  
  
# check crs   
proj4string(wi\_counties)

## [1] "+proj=tmerc +lat\_0=0 +lon\_0=-90 +k=0.9996 +x\_0=520000 +y\_0=-4480000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no\_defs"

proj4string(wi\_uni)

## [1] "+proj=longlat +datum=WGS84 +no\_defs"

# reproject counties with the uni projection   
wi\_counties\_reprj <- spTransform(wi\_counties, "+proj=longlat +datum=WGS84 +no\_defs")  
  
# make wi unis a df   
wi\_unis\_df <- as.data.frame(wi\_uni)

ggplot() +  
 geom\_polygon(data = wi\_counties\_reprj, aes(x=long, y = lat, group = group), fill = NA, color ="black", lwd=.1) +  
 geom\_point(data = wi\_unis\_df, aes(x=LON, y = LAT, size=INST\_SIZE), alpha=.9) +  
 geom\_point(data = wi\_unis\_df %>% filter(NAME == "University of Wisconsin-Madison"), aes(x=LON, y = LAT, size=INST\_SIZE), alpha=.9, color = "blue", show.legend = F)+  
 coord\_map("bonne", lat0 = 40)+  
 theme\_clean()+  
 theme(panel.grid.minor=element\_blank(),panel.grid.major=element\_blank())+  
 theme( panel.border=element\_blank())+  
 # scale\_color\_viridis(limits=c(min\_enroll, max\_enroll), breaks=seq(5000, 30000, by=5000), name = "Total\nenrollment")+  
 # guides(color= guide\_legend(), size=guide\_legend())+  
 scale\_size\_continuous(breaks = seq(-2, 5, by = 1), name = "Insitution\nsize")+  
 labs(y = "Latitude",   
 x = "Longitude",  
 title = "University of Wisconsin-Madison")

## Regions defined for each Polygons

## filter: removed 105 rows (99%), one row remaining

