**Introduction**

With all this Covid19 madness going on in the world, I felt inspired to utilize the vast amount of data we have from it. I came across a dataset by the [The NY Times](https://github.com/nytimes/covid-19-data). From reading over the GitHub repository, they obtain this time series data from state and local governments and health departments. To get a better sense of the data please go to the link above! The objective of what I am trying to accomplish is to create an animated time series map showing how Covid19 spread throughout NJ.

**Read in the data**

To read in the data I will use the readr package.

library(readr) #loads package

covid19\_data<-read\_csv("us\_counties.csv") #function that reads in csv files

Now that the data is read in.. lets check the structure of the data

str(covid19\_data)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 38197 obs. of 6 variables:

## $ date : Date, format: "2020-01-21" "2020-01-22" ...

## $ county: chr "Snohomish" "Snohomish" "Snohomish" "Cook" ...

## $ state : chr "Washington" "Washington" "Washington" "Illinois" ...

## $ fips : chr "53061" "53061" "53061" "17031" ...

## $ cases : num 1 1 1 1 1 1 1 1 1 1 ...

## $ deaths: num 0 0 0 0 0 0 0 0 0 0 ...

## - attr(\*, "spec")=

## .. cols(

## .. date = col\_date(format = ""),

## .. county = col\_character(),

## .. state = col\_character(),

## .. fips = col\_character(),

## .. cases = col\_double(),

## .. deaths = col\_double()

## .. )

Everything seems to look correct. All the columns seem to have the right structure!

**Filter data**

Being that we only want to see how covid19 spread in NJ, we will filter the data frame to only show cases in NJ. This can be done with the dplyr package. The data also has a great deal of cases that are “Unknown”. By reading over the documentation of the data, it says many state health departments chose to report cases separately when the patient’s county of residence is unknown or pending determination. I will take this data out since we wouldn’t be able to link it to a specific county.

library(dplyr) # used for data wrangling

NJ\_covid19<-covid19\_data%>%

dplyr::filter(state == "New Jersey",county != "Unknown") #filters data frame

head(NJ\_covid19) # Returns the first or last parts of the data frame

## # A tibble: 6 x 6

## date county state fips cases deaths

##

## 1 2020-03-04 Bergen New Jersey 34003 1 0

## 2 2020-03-05 Bergen New Jersey 34003 2 0

## 3 2020-03-06 Bergen New Jersey 34003 3 0

## 4 2020-03-06 Camden New Jersey 34007 1 0

## 5 2020-03-07 Bergen New Jersey 34003 3 0

## 6 2020-03-07 Camden New Jersey 34007 1 0

**Obtain county shapefile**

Since the data frame doesn’t have any “shapes”, I have to download a shapefile with NJ’s counties to join to the NJ\_covid19 data frame. Luckily, the New Jersey Office of GIS has a great website to download hundreds of spatial datasets across the state. By going to [this website](https://njogis-newjersey.opendata.arcgis.com/), I can download the county shapefile.

**Read in county shapefile**

I can read in the NJ county shapefile by using the sf package. The sf package is a great package to analyze spatial data.

library(sf)

NJ\_counties<-st\_read(getwd(),"New\_Jersey\_Counties") #function to read in shapefile

## Reading layer `New\_Jersey\_Counties' from data source `/Users/kevinzolea/Desktop/Personal\_Website/content/post/covid19\_nj' using driver `ESRI Shapefile'

## Simple feature collection with 21 features and 22 fields

## geometry type: MULTIPOLYGON

## dimension: XY

## bbox: xmin: 193684.7 ymin: 34945.75 xmax: 657059.7 ymax: 919549.4

## epsg (SRID): 3424

## proj4string: +proj=tmerc +lat\_0=38.83333333333334 +lon\_0=-74.5 +k=0.9999 +x\_0=150000 +y\_0=0 +datum=NAD83 +units=us-ft +no\_defs

As you can see above, the shapefile got read into R. Now let’s plot it to make sure everythings okay.

plot(st\_geometry(NJ\_counties))

**Join shapefile to data frame**

In order to make a map with the original data, I have to join the shapefile, NJ\_counties, with the NJ\_covid19 data frame. This can be done by using the left\_join() function from the dplyr package. The join will be based on the common column shared by both datasets, which would be the county column. First, I need to make the county column header in the NJ\_counties dataset lowercase. This is so I can match the county columns from both datasets. Next, I have to make all the counties in the county column lowercase in both datasets, so they can match. See below.

names(NJ\_counties)<-tolower(names(NJ\_counties)) # Makes county column header lowercase

NJ\_counties$county<-tolower(NJ\_counties$county)#Makes all rows in the county column lowercase

NJ\_covid19$county<-tolower(NJ\_covid19$county)#Makes all rows in the county column lowercase

Now I can join the two datasets based on the county column in both.

NJ\_covid19\_shapes<-left\_join(NJ\_covid19,NJ\_counties,by="county")%>%

dplyr::select(date,county,state,cases,deaths,geometry)#selects only the columns of interest

head(NJ\_covid19\_shapes)

## # A tibble: 6 x 6

## date county state cases deaths geometry

##

## 1 2020-03-04 bergen New Je… 1 0 (((656201 783614.4, 656141.1 7834…

## 2 2020-03-05 bergen New Je… 2 0 (((656201 783614.4, 656141.1 7834…

## 3 2020-03-06 bergen New Je… 3 0 (((656201 783614.4, 656141.1 7834…

## 4 2020-03-06 camden New Je… 1 0 (((342764 423475.8, 342804.1 4234…

## 5 2020-03-07 bergen New Je… 3 0 (((656201 783614.4, 656141.1 7834…

## 6 2020-03-07 camden New Je… 1 0 (((342764 423475.8, 342804.1 4234…

**Make map with ggplot2 and gganimate**

All the data manipulation part is done, now I can start building the actual map. I will be using the ggplot2 package in conjuction with the gganimate package to create the final animated time series map showing how covid19 spread throughout NJ’s counties.

library(ggplot2) #Used for plotting

library(gganimate) #Used for animations

library(RColorBrewer) #Used for color scale

# Used to make new data frame an sf object

# Must use st\_as\_sf in order to use geom\_sf() to plot polygons

NJ\_covid19\_shapes<-st\_as\_sf(NJ\_covid19\_shapes)

# Makes plot with ggplot2 and gganimate to animate through the days

covid\_map<-ggplot()+

geom\_sf(data = NJ\_counties,fill = "white")+

geom\_sf(data = NJ\_covid19\_shapes,aes(fill=cases))+

ggtitle("Spread of Covid19 Throughout New Jersey")+

xlab("")+

ylab("")+

labs(subtitle = "Date: {current\_frame}",

caption = "Date Source: The New York Times\nAuthor: Kevin Zolea")+

cowplot::background\_grid(major = "none", minor = "none") +

theme(axis.text.x = element\_blank(), axis.ticks.x = element\_blank(),

axis.text.y = element\_blank(), axis.ticks.y = element\_blank(),

axis.line = element\_blank(),

legend.background = element\_blank(),

legend.position=c(-0.3,0.8),

plot.background = element\_blank(),

panel.background = element\_blank(),

legend.text = element\_text(size=12),

legend.title = element\_text(colour="black", size=12, face="bold"),

plot.title=element\_text(size=20, face="bold",hjust =0.5),

plot.subtitle = element\_text(hjust = 0.5,size=12),

plot.caption = element\_text(size = 11,

hjust = .5,

color = "black",

face = "bold"))+

scale\_fill\_distiller("Number of Positive Cases",

palette ="Reds",type = "div",

direction = 1)+

transition\_manual(date)

animate(covid\_map, nframe=27,fps = 2, end\_pause = 15,height = 500, width =500)

**Things to note about above code**

Being that I joined a non-spatial data set with a spatial data set, I had to use st\_as\_sf() to make sure the geom\_sf() function was able to plot an sf object. I had to use transition\_manual() to make the animation. This is because when I tried to use transition\_time() it made the polygons on the map move all over the place. The animate function allows you to change how fast the animation is, how many frames to use, height, width, etc. To learn more about the gganimate package and the ggplot2 package,