# Deliverable #2

Micah A. Perez

11/16/2020

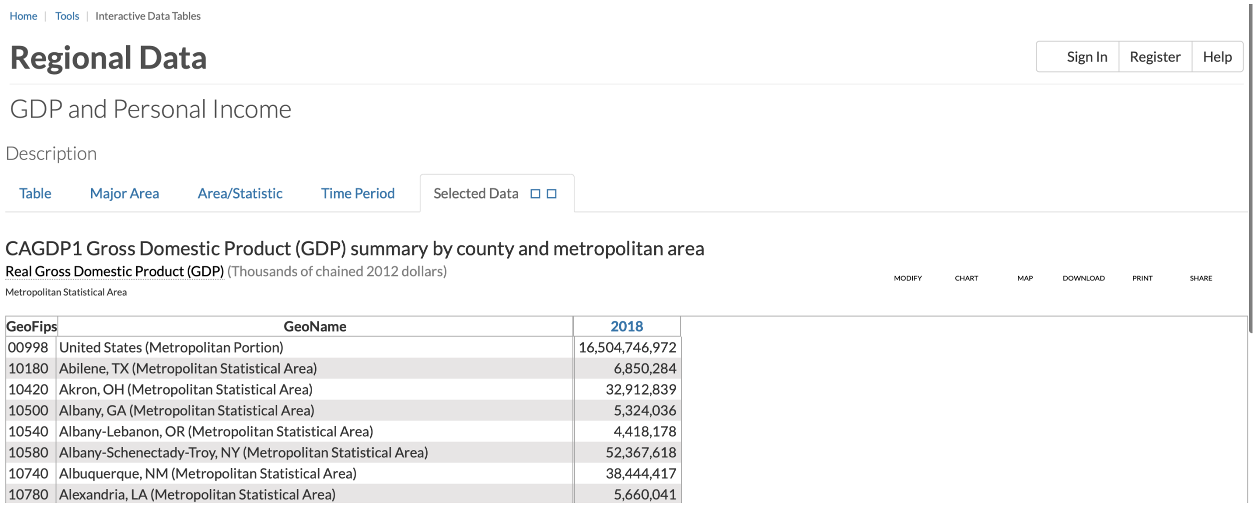
## Motivation

On my first instance of gathering, cleaning up, and visualizing data I went through quite a few steps to try to make everything usable for the rest of the project. That, being the bulk of my work, led to other pieces of the overall assignment being overall basic and not interesting. This time around, my goal is to build off of previous ideas and add to add new interesting data, visualizations, and methods of extraction. For this deliverable, I've first decided to try to include GDP data onto my already existing city data (that involved populations). I've also tried to experiment with some new ways of visualization of my data (using the us map library in R) as well as some new methods of transforming my data (using Google's API). When I started with this deliverable, I wanted to find out GDP's growth in correlation to population growth in the United States and come up with a method to rank and visualize area's by anticipated future growth. About half way through, I realized I'd have to answer that in the next deliverable, as I didn't feel that I has enough skill or time with R to answer those questions in a way that I wanted to. Ultimately, this deliverable was a learning experience that allowed me to become more proficient in R and find new and interesting libraries that I might be able to dive deeper into on the next part of this assignment. I felt that I was also lucky enough in the writing of the code to find out new and interesting data that I might implement later on as well.

## Part 1 - Data Extraction

To start, I found a government website (https://apps.bea.gov/iTable/index_regional.cfm) that included information including GDP for cities across the United States. Lucky for me, many of the city names matched up with my previous data. For the simplicity of this assignment, I downloaded the html document and web scraped the data directly from the code, giving me a nice data set that I felt was easily appended to my existing data. At first, I attempted to join the data by name directly. This however, wasn't working and I thought I was out of options. I then realized that the city names were in the same order in my newly scraped data as my old data was, which allowed me to simply create an index column for both data sets and left-join them together.

```r
library(tidyverse)
library(rvest)
library(stringr)
library(tidyverse)
library(rvest)
#Read Data in from HTML and extract City/State Names

All_time_data <- read_html("final_data/CITY_GDP_DATA.html") %>%
   html_node("body")
City_data <- All_time_data %>%
   xml_find_all("//td[contains(@class, 'NormalStyle_left Locked')]")  %>%
```



Regional Data

GDP and Personal Income

CAGDP1 Gross Domestic Product (GDP) summary by county and metropolitan area

Real Gross Domestic Product (GDP) (Thousands of chained 2012 dollars)

Metropolitan Statistical Area

| GeoFips | GeoName | 2018 |
|---|---|---|
| 00998 | United States (Metropolitan Portion) | 16,504,746,972 |
| 10180 | Abilene, TX (Metropolitan Statistical Area) | 6,850,284 |
| 10420 | Akron, OH (Metropolitan Statistical Area) | 32,912,839 |
| 10500 | Albany, GA (Metropolitan Statistical Area) | 5,324,036 |
| 10540 | Albany-Lebanon, OR (Metropolitan Statistical Area) | 4,418,178 |
| 10580 | Albany-Schenectady-Troy, NY (Metropolitan Statistical Area) | 52,367,618 |
| 10740 | Albuquerque, NM (Metropolitan Statistical Area) | 38,444,417 |
| 10780 | Alexandria, LA (Metropolitan Statistical Area) | 5,660,041 |

Figure 1: The website that I extracted my data from

```
   html_text()
#Take out half of the rows because they were doubled on extraction
City_data <- City_data[0:385]
#Extract GDP Data
GDP <- All_time_data %>%
   xml_find_all("//td[contains(@class, 'ns shade-column')]")  %>%
   html_text()
#Throw Data into one tibble
gdp_data <- data.frame(City_data, GDP)
head(gdp_data)
```

```
##                                                      City_data            GDP
## 1                        United States (Metropolitan Portion) 16,504,746,972
## 2                 Abilene, TX (Metropolitan Statistical Area)      6,850,284
## 3                   Akron, OH (Metropolitan Statistical Area)     32,912,839
## 4                  Albany, GA (Metropolitan Statistical Area)      5,324,036
## 5          Albany-Lebanon, OR (Metropolitan Statistical Area)      4,418,178
## 6 Albany-Schenectady-Troy, NY (Metropolitan Statistical Area)     52,367,618
```

```
library(tidyr)
library(dplyr)

#Since the City_data contains both the City and State information, divide the two by the Comma
gdp_data <- gdp_data %>%
  separate(City_data, c("City", "State"), ",")

#Take out the comma's of GDP so I am able to make it a double (to be used later in data analysis)
gdp_data$GDP <- as.numeric(gsub(",","",gdp_data$GDP))
gdp_data <- mutate(gdp_data, GDP=as.double(gdp_data$GDP) )

#The United States GDP strangely disappeared so I manually added it
gdp_data$GDP[which(gdp_data$City == "United States")] <- 16504746972

#Remove the State and the City Data, all we need to combine it with th main dataset is the GDP numbers
```

```r
gdp_data$State <- NULL
gdp_data$City <- NULL

head(gdp_data)
```

```
##           GDP
## 1 16504746972
## 2     6850284
## 3    32912839
## 4     5324036
## 5     4418178
## 6    52367618
```

```r
PreviousData <- read.csv(file = 'final_data/output.csv')
#finalData$Index <- NULL

#Make an index on GDP data to left_join with the Mains Index
gdp_data$Index <- seq.int(nrow(gdp_data))

mainData <- left_join(PreviousData, gdp_data, by="Index")
mainData$Index <- NULL
head(mainData)
```

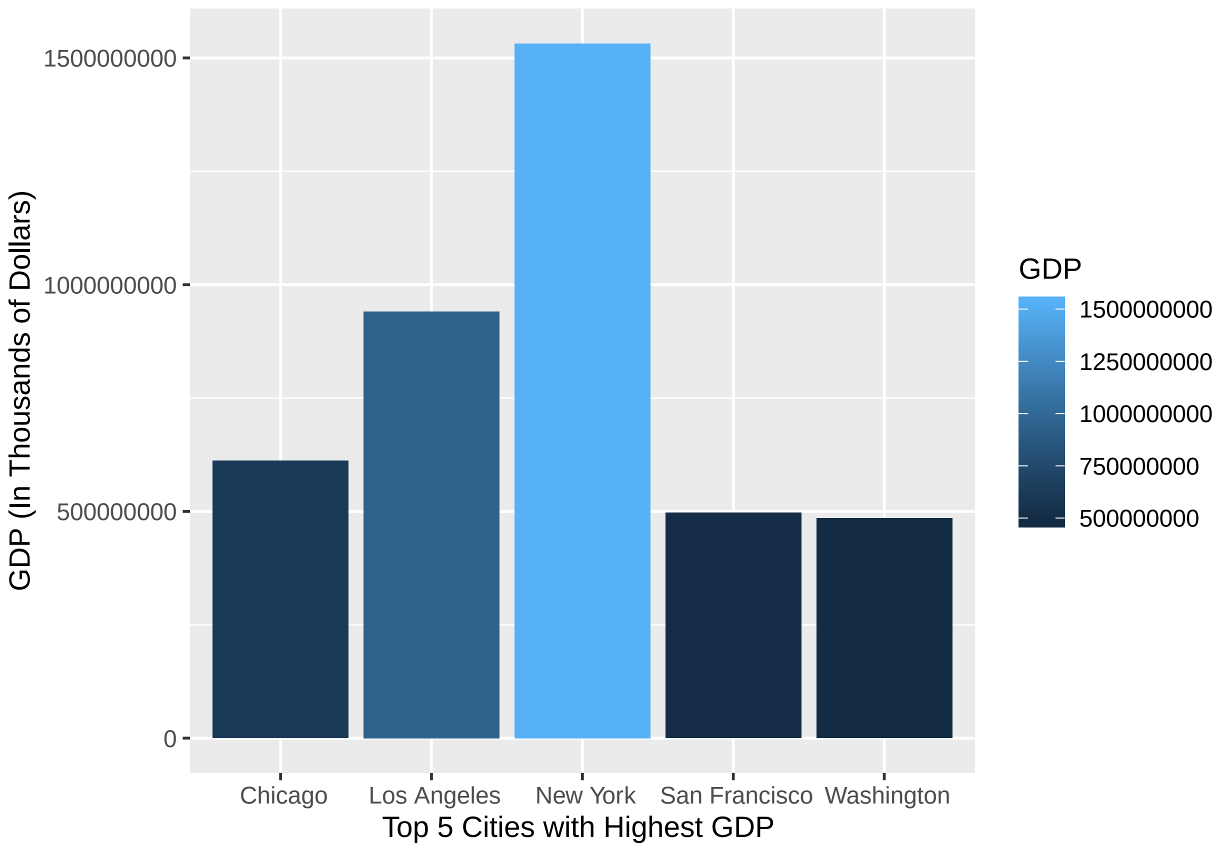
```
##                        City State      Region Political_Leaning
## 1            United States    US          US               RED
## 2                  Abilene    TX       South               RED
## 3                    Akron    OH     Midwest               RED
## 4                   Albany    GA       South               RED
## 5           Albany-Lebanon    OR        West              BLUE
## 6  Albany-Schenectady-Troy    NY   Northeast              BLUE
##   Population_Change_Decade Natural_Increase_Decade Births_Decade Deaths_Decade
## 1                 19481418                11621558      36275313      24653755
## 2                     6808                    6432         21881         15449
## 3                      283                    3307         69159         65852
## 4                    -7307                    5423         18764         13341
## 5                    13068                    2217         13642         11425
## 6                     9668                   11001         83501         72500
##   Total_Decade International_Decade Domestic_Decade Population_Change_Annual
## 1      7859860              7859860               0                 1552022
## 2          431                 3167           -2736                     910
## 3        -2597                13086          -15683                    -376
## 4       -12853                  865          -13718                   -1114
## 5        10883                  204           10679                    2298
## 6         -858                16398          -17256                   -1882
##   Natural_Increase_Annual Births_Annual Deaths_Annual Total._Annual
## 1                  956674       3791712       2835038        595348
## 2                     609          2361          1752           310
## 3                       0          7196          7196          -354
## 4                     350          1891          1541         -1465
## 5                     202          1512          1310          2087
## 6                     469          8704          8235         -2342
##   International_Annual Domestic_Annual         GDP
## 1               595348               0 16504746972
## 2                  220              90     6850284
## 3                  836           -1190    32912839
## 4                   51           -1516     5324036
## 5                   21            2066     4418178
## 6                 1033           -3375    52367618
```

## Part 2 - Simple Visualization and Mapping

After I had my final data, I decided to dos some simple visualization. I was immediately curious to find out which were the 5 biggest cities in the United States by GDP. To do this, I first cut down the names of hyphenated metropolitan areas to make the graph more readable, ordered the cities from biggest to smallest (in terms of GDP), and then spiced the Top 5 (I took 2 to 6 because the US as a whole was at 1). I wasn’t too surprised in finding out that New York City had (by far) the largest GDP out of any US City, but the other four were interesting in terms of how they matched up After this, I decided to use the us map library in R and try a different way of visualization. For this step, I extracted each states abbreviation and grouped their total states GDP (from the data I was working with). After I put this on a map view of the US. From reading the graph, one can notice that the lighter states have higher GDP’s than the darker ones.

```r
library(ggplot2)

#Put To make the numbers on the graph more readable
options(scipen=10000)

mainData$City <- as.character(mainData$City)

#Put Dataset into dataframe
City_GDP_Data <- as.data.frame(mainData)
City_GDP_Data[is.na(City_GDP_Data)] = 0
#Sliced out Sub-Cities for Major Metro areas to only have name of main city encompassing Metro
City_GDP_Data$City <- gsub("(.*)-.*", "\\1", City_GDP_Data$City)
City_GDP_Data$City <- gsub("(.*)-.*", "\\1", City_GDP_Data$City)

#Used ggplot to graph top 5 biggest Metropolitain areas
City_GDP_Data %>%
    arrange(desc(GDP)) %>%
    slice(2:6) %>%
    ggplot(., aes(x=City, y=GDP))+
              geom_bar(stat=’identity’, aes(fill = GDP)) +
              print(labs(y="GDP (In Thousands of Dollars)", x = "Top 5 Cities with Highest GDP "))
```

```
## $y
## [1] "GDP (In Thousands of Dollars)"
##
## $x
## [1] "Top 5 Cities with Highest GDP "
##
## attr(,"class")
## [1] "labels"
```



```
library(usmap) #Use USA MAP Library to Better visualize results
```

```
## Warning: package 'usmap' was built under R version 3.6.2
```

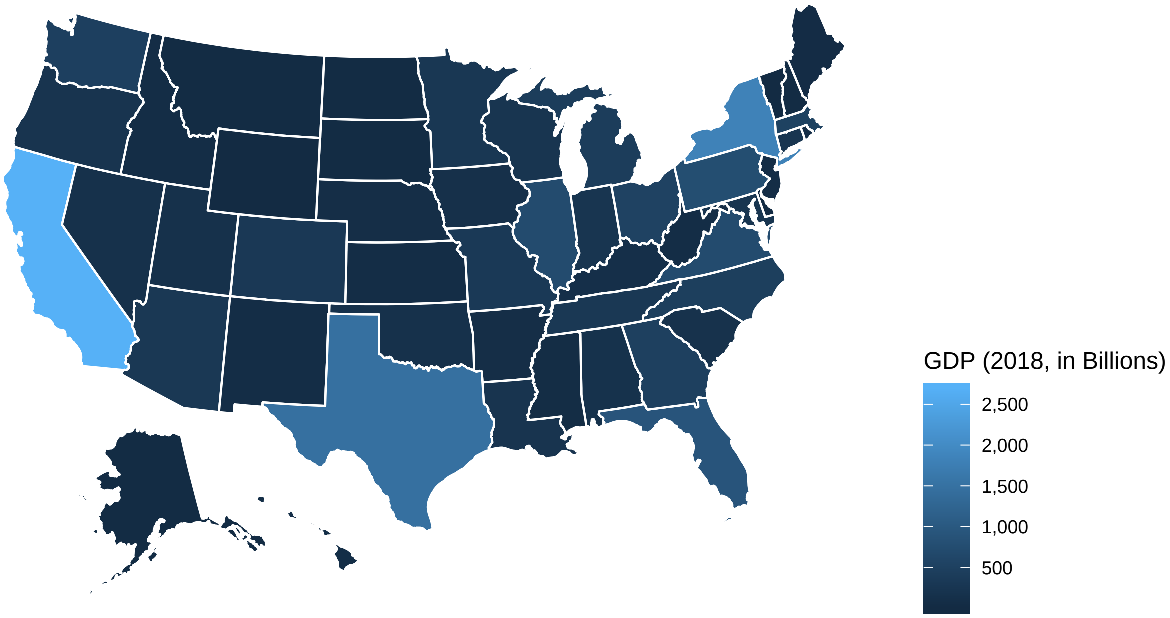
```
library(cdlTools)
```

```
## Warning: package 'cdlTools' was built under R version 3.6.2
```

```
##
## Attaching package: 'cdlTools'
```

```
## The following object is masked from 'package:usmap':
##
##     fips
```

```
#Created new data set to work with (Just includes State and GDP data)
State.population.data <- aggregate(GDP ~ State , data=mainData, FUN=sum)

# Divide GDP By 1,000,000 to Get number in Billions Rather Than Thousands (To Be Much more readable to
State.population.data$GDP <- (State.population.data$GDP/1000000)

# Made State data a character so I could turn it into a fip (readable content to usmap library)
State.population.data <- mutate(State.population.data, State=as.character(State.population.data$State)
State.population.data$State <- fips(State.population.data$State)

State.population.data <- data.frame(fips=State.population.data$State, value=State.population.data$GDP)

#Put all data into a dataframe so it was readable to function
df <- data.frame(State.population.data, na=0)

plot_usmap(data = df, values = "value", color = "white") +
  scale_fill_continuous(name = "GDP (2018, in Billions)", label = scales::comma) +
  theme(legend.position = "right")
```

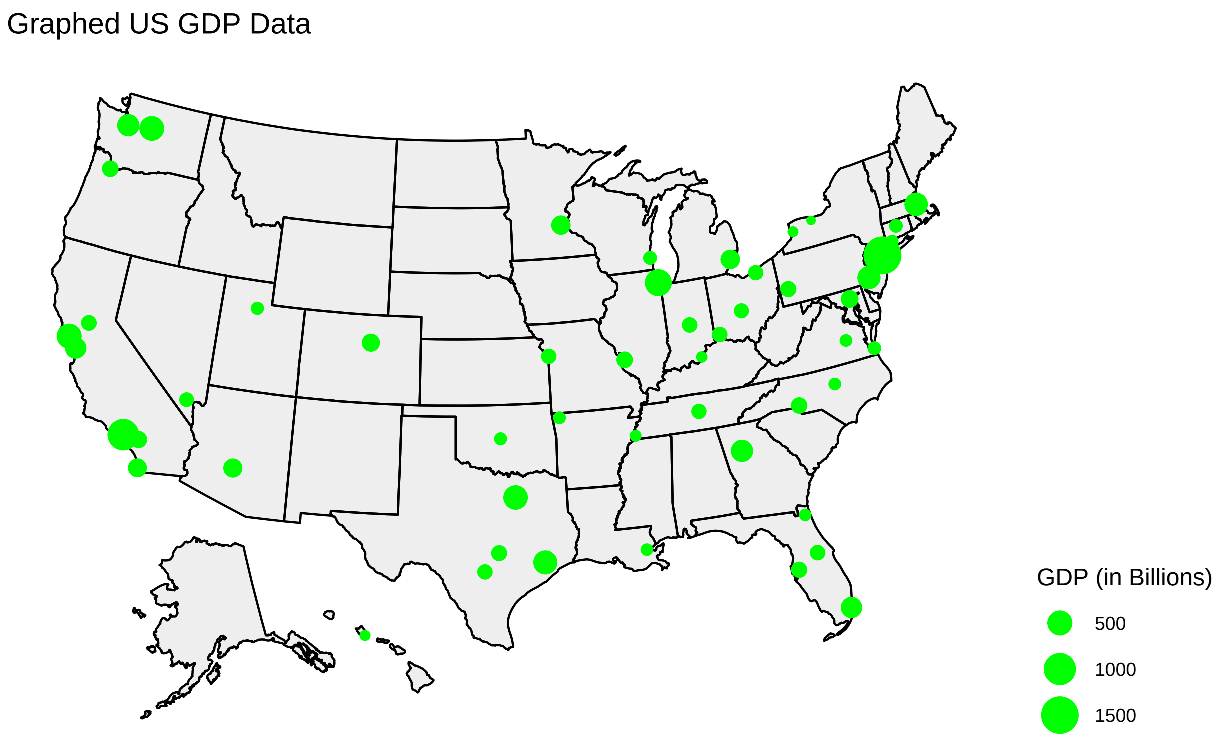


## Part 3 - Mapping Data using Google API After my "simple" visualizations I wanted to get into more complected and advanced graphing but I couldn't find any way to graph my cities by name on a map of the US. After hours of online searching, I realized that Google Maps API could be used extract the latitude and longitude points from cities names, which I could then use to graph on a plot of the us maps. In this example, I graphed the top 52 metropolitan areas by GDP in the United States.

```
require(ggmap)
require(maps)
library(mapproj)
register_google(key = "AIzaSyByXCki-hIHBM_HzbK_IE8d2xMZZYXEGLM") #Google Maps API Key to have acess to

City_GDP_Data_For_Cities <- #Ordered and sliced for top 52 US cities by GDP
    City_GDP_Data %>%
    arrange(desc(GDP)) %>%
    slice(2:53)
```

```
City_GDP_Data_For_Cities <- cbind(geocode(as.character(City_GDP_Data_For_Cities$City)), City_GDP_Data_F

City_GDP_Data_For_Cities[is.na(City_GDP_Data_For_Cities)] <- 0 #Made all NA vaules 0 to avoid errors in
```

```
City_GDP_Data_For_Cities$GDP <- (City_GDP_Data_For_Cities$GDP/1000000) #Divided GDP Data by 1 Million (


City_data_transformed <- usmap_transform(City_GDP_Data_For_Cities) #Transformed coordinate data to be r

plot_usmap(fill = "grey", alpha = 0.25) +
  geom_point(data=City_data_transformed, aes(x=lon.1, y=lat.1, size=GDP), color="green") +
  labs(title = "Graphed US GDP Data", size = "GDP (in Billions)") +
  theme(legend.position = "right")
```



## Ethics