

# 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, methods of extraction, and most of all modeling. For this deliverable, I've first decided to try to include GDP data onto my already existing city data (that involved populations). Then I try to answer the question of how we can predict GDP by some population variable with a simple model. Lastly (mostly for fun), I attempt to map the 50 largest US cities by population's growth rate from the last decade using a Google Maps API.

## Part 1 - Data Extraction

To start, I found a government website ([https://apps.bea.gov/iTable/index\\_regional.cfm](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. After this didn't seem to work properly, I thought I was out of options. Fortunately, 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.

[Home](#) | [Tools](#) | [Interactive Data Tables](#)

## Regional Data

[Sign In](#) [Register](#) [Help](#)

### GDP and Personal Income

Description

[Table](#) [Major Area](#) [Area/Statistic](#) [Time Period](#) [Selected Data](#) ☐ ☐

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

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

Metropolitan Statistical Area

[MODIFY](#) [CHART](#) [MAP](#) [DOWNLOAD](#) [PRINT](#) [SHARE](#)

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

```

library(tidyverse)
library(rvest)
library(stringr)
library(tidyverse)
library(rvest)
library(modelr)

#Read Data in from HTML and extract City/State Names
#2018 Data Extraction
All_time_data_2018 <- read_html("final_data/CITY_GDP_DATA.html") %>%
  html_node("body")
City_data <- All_time_data_2018 %>%
  xml_find_all("//td[contains(@class, 'NormalStyle_left Locked')]") %>%
  html_text()
#Take out half of the rows because they were doubled on extraction
City_data <- City_data[0:385]
#Extract GDP_2018 Data
GDP_2018 <- All_time_data_2018 %>%
  xml_find_all("//td[contains(@class, 'ns shade-column')]") %>%
  html_text()

#2010 Data from different HTML page
All_time_data_2010 <- read_html("final_data/CITY_GDP_DATA_2010.html") %>%
  html_node("body")
City_data_2010 <- All_time_data_2010 %>%
  xml_find_all("//td[contains(@class, 'NormalStyle_left Locked')]") %>%
  html_text()
#Take out half of the rows because they were doubled on extraction
City_data_2010 <- City_data_2010[0:385]
#Extract GDP_2018 Data
GDP_2010 <- All_time_data_2010 %>%
  xml_find_all("//td[contains(@class, 'ns shade-column')]") %>%
  html_text()

#Throw Data into one tibble
gdp_data <- data.frame(City_data, GDP_2010, GDP_2018)
head(gdp_data)

```

```

##                               City_data      GDP_2010
## 1      United States (Metropolitan Portion) 13,845,441,884
## 2      Abilene, TX (Metropolitan Statistical Area)      6,103,245
## 3      Akron, OH (Metropolitan Statistical Area)      30,124,290
## 4      Albany, GA (Metropolitan Statistical Area)      5,341,783
## 5      Albany-Lebanon, OR (Metropolitan Statistical Area) 3,572,329
## 6 Albany-Schenectady-Troy, NY (Metropolitan Statistical Area) 48,892,808
##      GDP_2018
## 1 16,504,746,972
## 2   6,850,284
## 3  32,912,839
## 4   5,324,036
## 5   4,418,178
## 6  52,367,618

```

## Note 1 - Addition of Population Data

A while through project 1 I realized that working with population data that was purely differences in growth and decline and no actual total population would be limiting to my potential ideas, visualizations, and models. When I started on the Simple Model, I realized that using population percentage change would be incredibly useful so I decided to download a CSV and add it on to my current data as I did below.

```
library(tidyr)
library(dplyr)

#Read a CSV file to add actual population data to my dataset
Population_data <- read.csv(file = 'final_data/Population_Data_2010-2018.csv')

#Take out commas in Population Data to be able to Properly add it as a double column to my existing data
Population_data$X2010 <- as.numeric(gsub(",", "", Population_data$X2010))
Population_data$X2018 <- as.numeric(gsub(",", "", Population_data$X2018))

#Put Population data into a dataframe
Population_data <- as.data.frame(Population_data)

#Added Population data as Columns to existing data
gdp_data$Population_2010 <- Population_data$X2010#mutate(Population_data, X2010=as.double(Population_data$X2010))
gdp_data$Population_2018 <- Population_data$X2018#mutate(Population_data, X2018=as.double(Population_data$X2018))

#Take out the comma's of GDP_2018 so I am able to make it a double (to be used later in data analysis)
gdp_data$GDP_2018 <- as.numeric(gsub(",", "", gdp_data$GDP_2018))
gdp_data <- mutate(gdp_data, GDP_2018=as.double(gdp_data$GDP_2018) )

#Did the Same for the 2010's Data
gdp_data$GDP_2010 <- as.numeric(gsub(",", "", gdp_data$GDP_2010))
gdp_data <- mutate(gdp_data, GDP_2010=as.double(gdp_data$GDP_2010) )

#Remove the State and the City Data, all we need to combine it with the main dataset is the GDP numbers
gdp_data$State <- NULL
gdp_data$City <- NULL

#Create a new column of the change between 2010 to 2018 and label it as change over the decade (2018 was the last year)
gdp_data$Decade_GDP_Change <- (gdp_data$GDP_2018 - gdp_data$GDP_2010)

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

#Make an index on GDP_2018 data to left_join with the Mains Index
gdp_data$Index <- seq.int(nrow(gdp_data))

#Adding the data through a Left Join
mainData <- left_join(PreviousData, gdp_data, by="Index")
mainData$Index <- NULL
```

## Part 2 - Simple Visualization and Mapping

After I had my final data, I decided to do 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.

```
library(ggplot2)
library(scales)

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

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

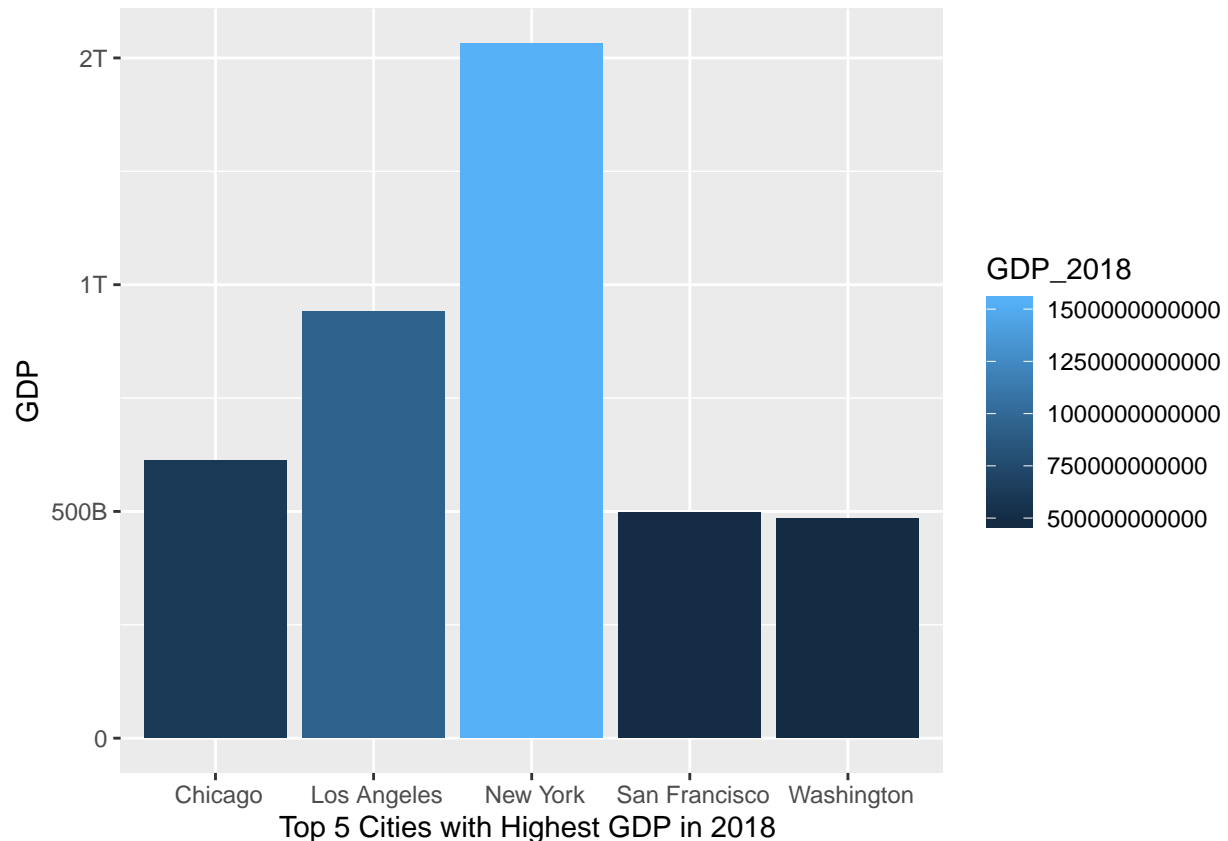
#Put Dataset into dataframe
City_GDP_Data <- as.data.frame(mainData)
City_GDP_Data[is.na(City_GDP_Data)] = 0
#Sliced out Sub-Cities for Major Metro areas to only have name of main city encompassing Metro
City_GDP_Data$City <- gsub("(.)-.*", "\\1", City_GDP_Data$City)
City_GDP_Data$City <- gsub("(.)-.*",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

#Used ggplot to graph top 5 biggest Metropolitan areas

##Here I am multiplying each of my calculated values by 1000. I Do this because when I extracted the dat
City_GDP_Data$GDP_2018 <- (City_GDP_Data$GDP_2018*1000)
City_GDP_Data$GDP_2010 <- (City_GDP_Data$GDP_2010*1000)
City_GDP_Data$Decade_GDP_Change <- (City_GDP_Data$Decade_GDP_Change*1000)

City_GDP_Data %>%
  arrange(desc(GDP_2018)) %>%
  slice(2:6) %>%
  ggplot(., aes(x=City, y=GDP_2018))+
    geom_bar(stat='identity', aes(fill = GDP_2018)) +
    print(labs(y="GDP", x = "Top 5 Cities with Highest GDP in 2018 ")) +
    scale_y_continuous(labels = scales::label_number_si())

## $y
## [1] "GDP"
##
## $x
## [1] "Top 5 Cities with Highest GDP in 2018 "
##
## attr(,"class")
## [1] "labels"
```



### Note 2 - Here (mostly for fun and to learn about new functions and libraries) I decided to map GDP data across the US on an actual map, where one can visualize what states have the most economic activity going on.

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

#Created new data set to work with (Just includes State and GDP_2018 data)
State.population.data <- aggregate(GDP_2018 ~ State , data=mainData, FUN=sum)

# Divide GDP_2018 By 1,000,000 to Get number in Billions Rather Than Thousands (To Be Much more readable)
State.population.data$GDP_2018 <- (State.population.data$GDP_2018/1000000)

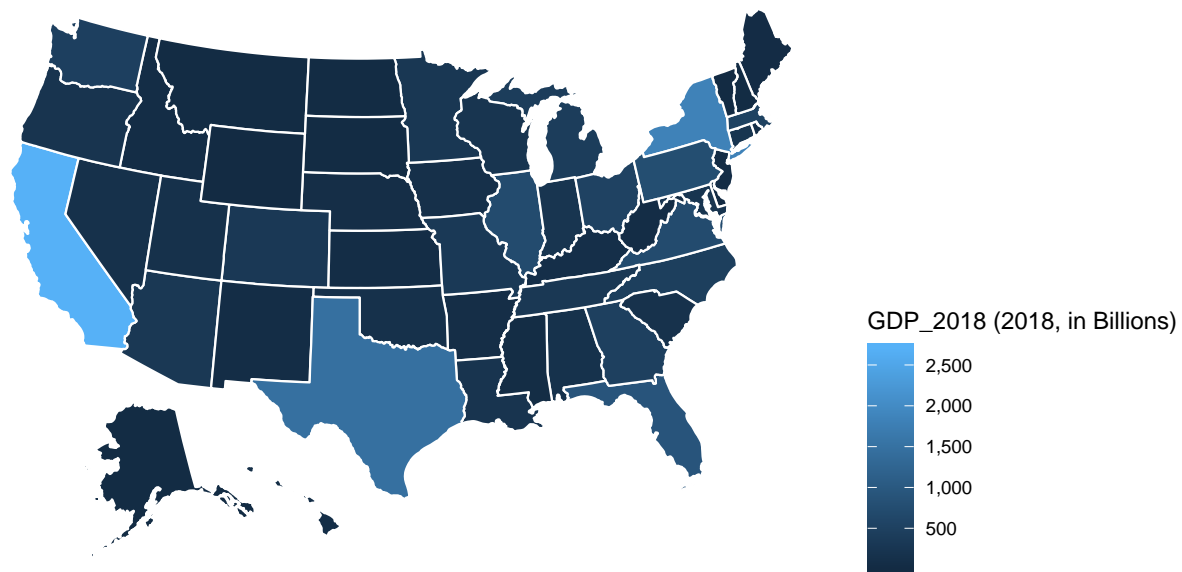
# Made State data a character so I could turn it into a fip (readable content to usmap library)
State.population.data <- mutate(State.population.data, State=as.character(State.population.data$State))

State.population.data$State <- fips(State.population.data$State)

State.population.data <- data.frame(fips=State.population.data$State, value=State.population.data$GDP_2018)

#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 (2018, in Billions)", label = scales::comma) +
  theme(legend.position = "right")
```



## Part 3 - Simple Model In this section of the project I try to answer a simple question - can you predict a cities GDP percentage growth from another factor? I found, as a matter of fact, that you nearly can with decent accuracy using that cities Population percentage growth. Since the model involves a variation of code, I wanted to describe my step by step thinking with it.

```
#Loaded in all Libraries needed for functions
require(ggmap)
require(maps)
library(mapproj)
library(ggrepel)
library(lme4)
library(caret)
library(modelr)

#Create new columns for percent change of GDP and Population using my data from their numbers in 2010 as
City_GDP_Data$Population_Percent_Change <- (City_GDP_Data$Population_2018/City_GDP_Data$Population_2010)
City_GDP_Data$GDP_Percent_Change <- (City_GDP_Data$GDP_2018/City_GDP_Data$GDP_2010)
```

**Note 3 -** In these first few steps I extract the data needed for the modeling and plot the initial data to get a look of what the model will attempt to answer. The plot itself was an after though (post model and Exploratory data analysis) as was the population percent change above.

```
#Slice data frame only taking data from cities whose GDP data I have (leaving out the United States as
df <- data.frame(City_GDP_Data) %>%
```

```

slice(2:385)

ggplot(df, aes(x = df$GDP_Percent_Change, y = df$Population_Percent_Change, color = df$Region)) +
  geom_point() +
  print(labs(x="Population Percentage Change (2010 - 2018)", y = "GDP Percentage Change (2010 - 2018)"))

## $x
## [1] "Population Percentage Change (2010 - 2018)"
##
## $y
## [1] "GDP Percentage Change (2010 - 2018) "
##
## $colour
## [1] "State"
##
## attr(,"class")
## [1] "labels"

```



Note 4 - To start my modeling process, I wanted to see what a model without training of taking out a validation of training set would look like. I also wanted to see how residuals would compare. This is my first model with nothing more than a few lines of code and no further testing. Using Population Percent change is another part of the afterthought post data analysis. Throughout my creation of the model I was changing that Y variable around in the lm function and the plots.

```
first_model <- lm(GDP_Percent_Change ~ Population_Percent_Change, data = df)
summary(first_model)
```

```
##
## Call:
## lm(formula = GDP_Percent_Change ~ Population_Percent_Change,
##     data = df)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-0.35162	-0.05986	-0.00685	0.03903	1.41490

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.3503	0.1094	-3.201	0.00149 **
Population_Percent_Change	1.4209	0.1039	13.674	< 0.0000000000000002 ***

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1342 on 382 degrees of freedom
## Multiple R-squared:  0.3286, Adjusted R-squared:  0.3269
## F-statistic: 187 on 1 and 382 DF, p-value: < 0.00000000000000022
```

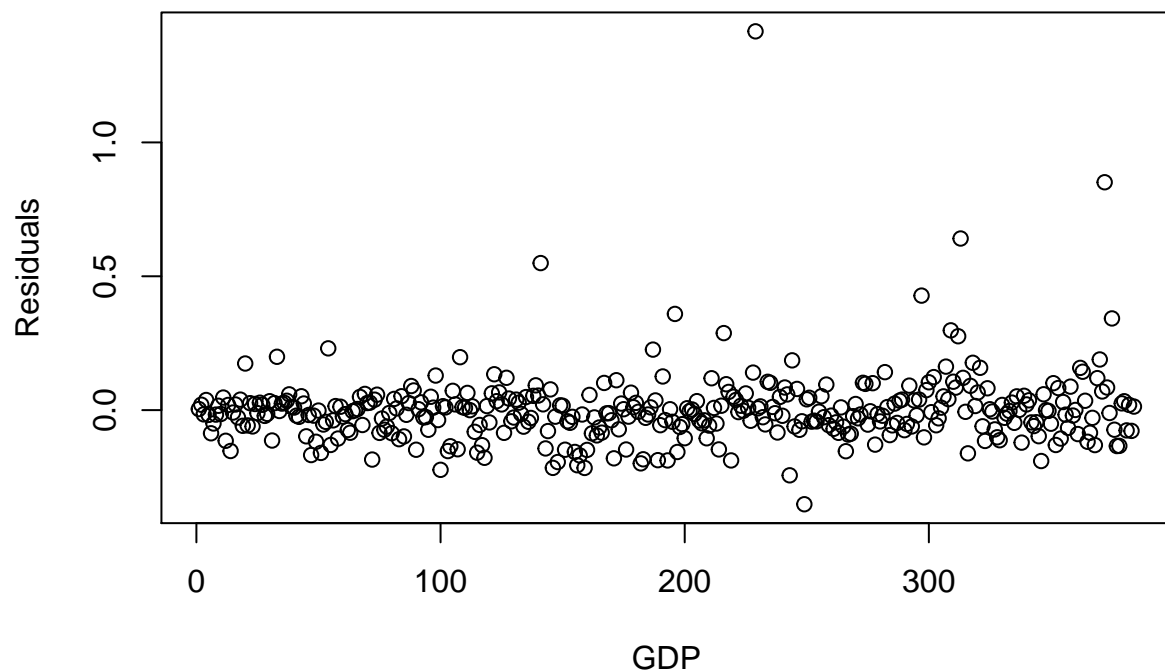
```
first_model$coef
```

```
##              (Intercept) Population_Percent_Change
##              -0.3502554              1.4209169
```

```
GDP_res <- resid(first_model)
```

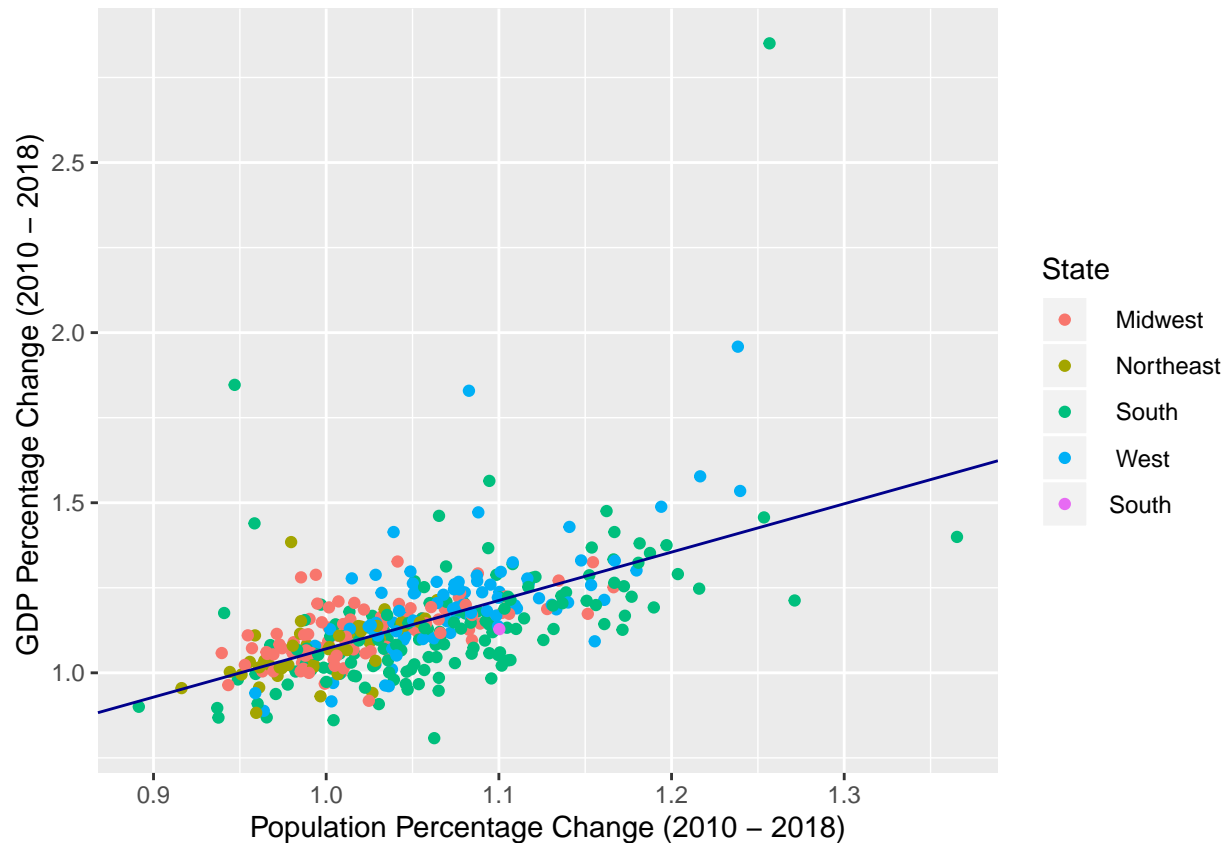
```
plot(GDP_res, ylab="Residuals", xlab="GDP")
```





```
ggplot(data=df, mapping=aes(x=df$Population_Percent_Change, y=df$GDP_Percent_Change, color = df$Region))
  geom_point() +
  geom_abline(intercept = first_model$coefficients[1], slope = first_model$coefficients[2], color = "darkred")
print(labs(x="Population Percentage Change (2010 - 2018)", y = "GDP Percentage Change (2010 - 2018) ")
```

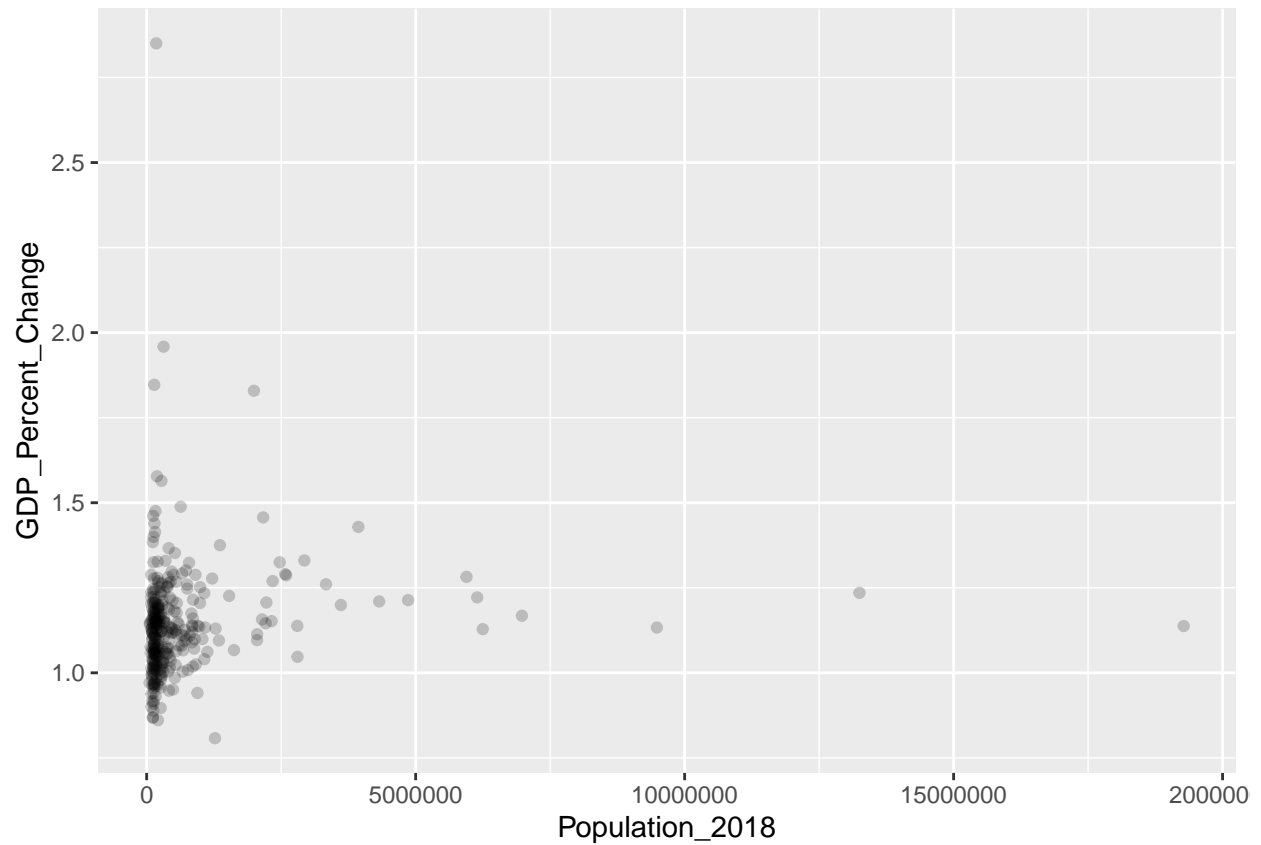
```
## $x
## [1] "Population Percentage Change (2010 - 2018)"
##
## $y
## [1] "GDP Percentage Change (2010 - 2018) "
##
## $colour
## [1] "State"
##
## attr(,"class")
## [1] "labels"
```



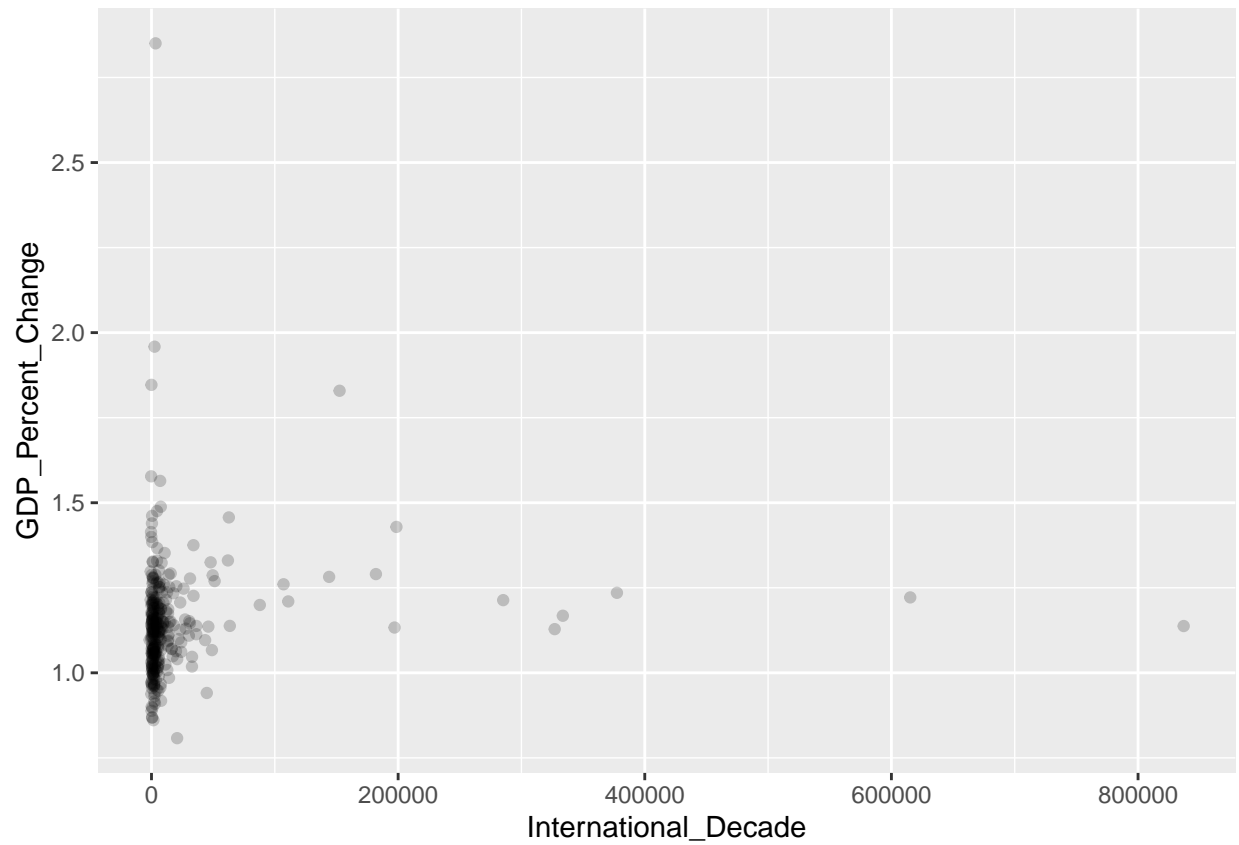
Note 5 - Here is where I attempt to do modeling with the help of validation, training, and testing sets splitting them into a 60/20/20 fashion. I also do a little analysis to find out what kind of independent variable would work the best at predicting GDP percentage change. After testing the total population of cities (for 2018) and the amount of Immigrant growth for the last decade, I realized that these numbers didn't work well. This led me to creating a new variable, Population Percent Change, which allows me to make a stronger and more accurate model.

```
#Splitting My test and rest data
rest_rows <- as.vector(createDataPartition(df$GDP_Percent_Change, p = 0.8, list = FALSE))
test <- df[-rest_rows, ]
rest <- df[rest_rows, ]

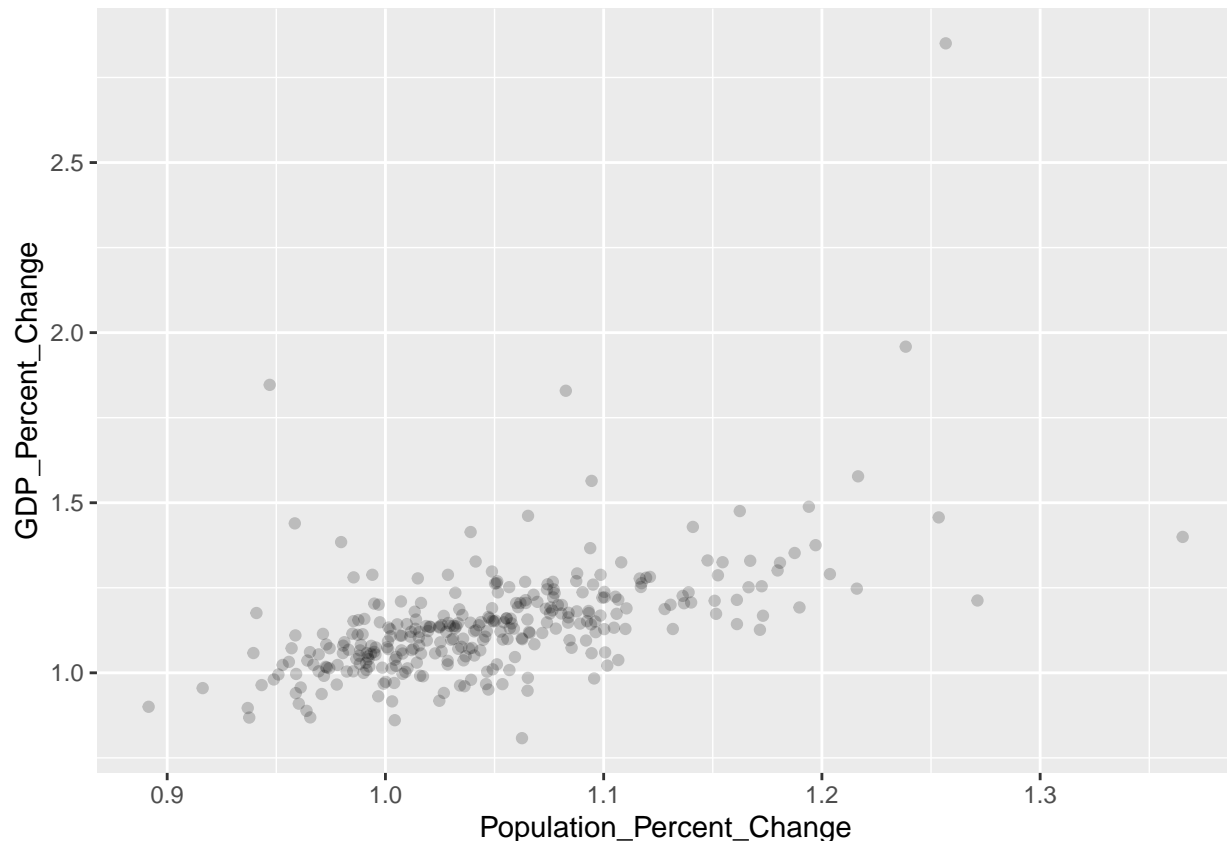
#Exploratory Data Analysis
#Total Population (2018)
ggplot(data = rest) +
  geom_point(mapping = aes(x = Population_2018, y = GDP_Percent_Change), alpha = 0.2)
```



```
#Immigrant growth for the last decade (2010-2019)  
ggplot(data = rest) +  
  geom_point(mapping = aes(x = International_Decade, y = GDP_Percent_Change), alpha = 0.2)
```



```
#Population percentage change (2010-2018)  
ggplot(data = rest) +  
  geom_point(mapping = aes(x =Population_Percent_Change, y = GDP_Percent_Change), alpha = 0.2)
```



Note 6 - In this section I broke apart the rest of my data to validation and training data which I added predictions to and plotted in respect to a line with a slope of 1. I realized that this actually worked decently well and used functions to test the accuracy of my data. When my goodness of fit measures data matched up quite well, I decided to keep going and find residuals. The residuals themselves weren't too far off either (with the exception of some outliers)

```
#
#second_model <- lm(GDP_Percent_Change ~ Population_Percent_Change, data = df)

#grid <- data_grid(df, Population_Percent_Change) %>%
#  add_predictions(second_model)
#data <- add_residuals(df, second_model)

#set.seed(1234)

#Split rest of data into validation and training data
row_num <- as.vector(createDataPartition(rest$GDP_Percent_Change, p=0.75, list=FALSE))
validate = rest[row_num,]
train = rest[-row_num,]

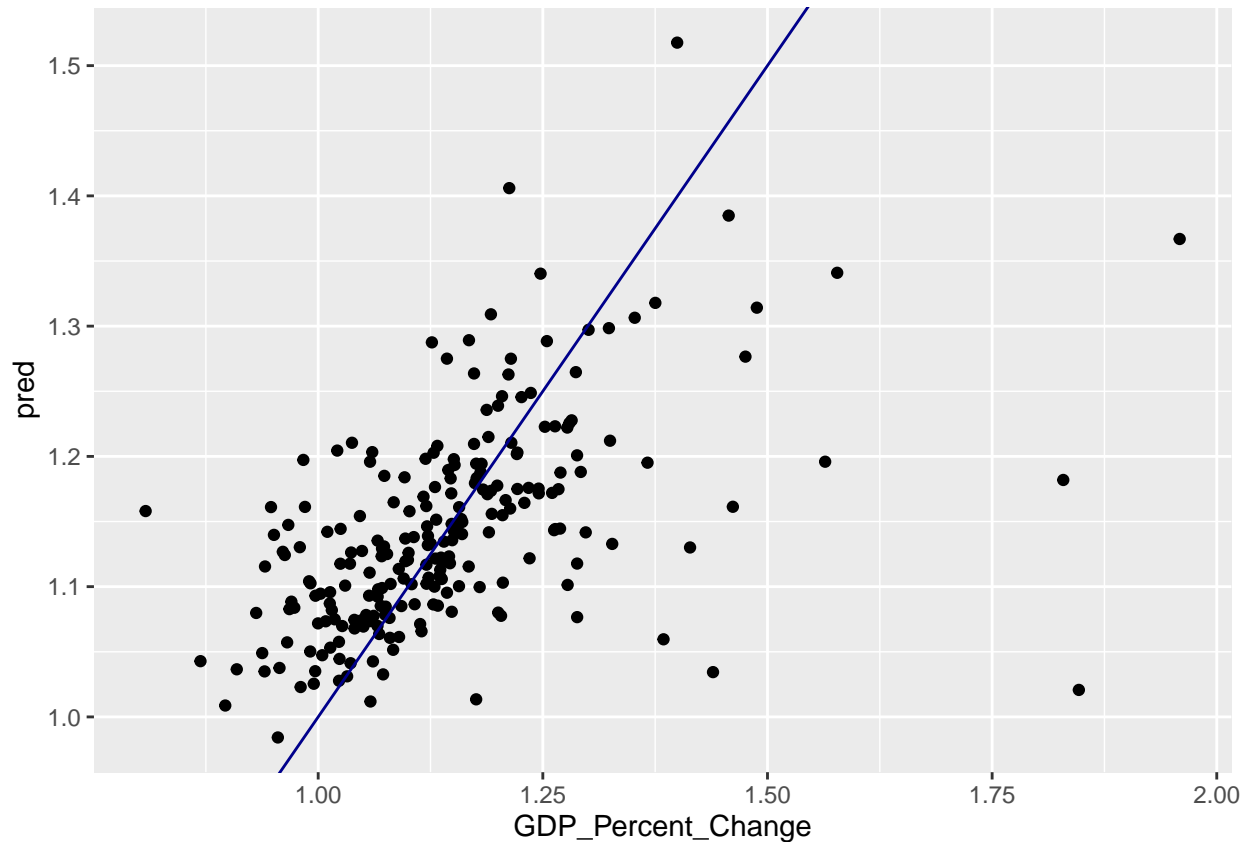
#Create a validation model out of validation data
validate_model <- lm(GDP_Percent_Change ~ Population_Percent_Change, data = validate)
```

```

predictions <- add_predictions(validate, validate_model)

ggplot(data=predictions, mapping = aes(x=GDP_Percent_Change, y = pred)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "dark blue")

```



```

#Goodness of fit measures
R2(predictions$pred, predictions$GDP_Percent_Change)

```

```
## [1] 0.2802485
```

```
MAE(predictions$pred, predictions$GDP_Percent_Change)
```

```
## [1] 0.07920761
```

```
RMSE(predictions$pred, predictions$GDP_Percent_Change)
```

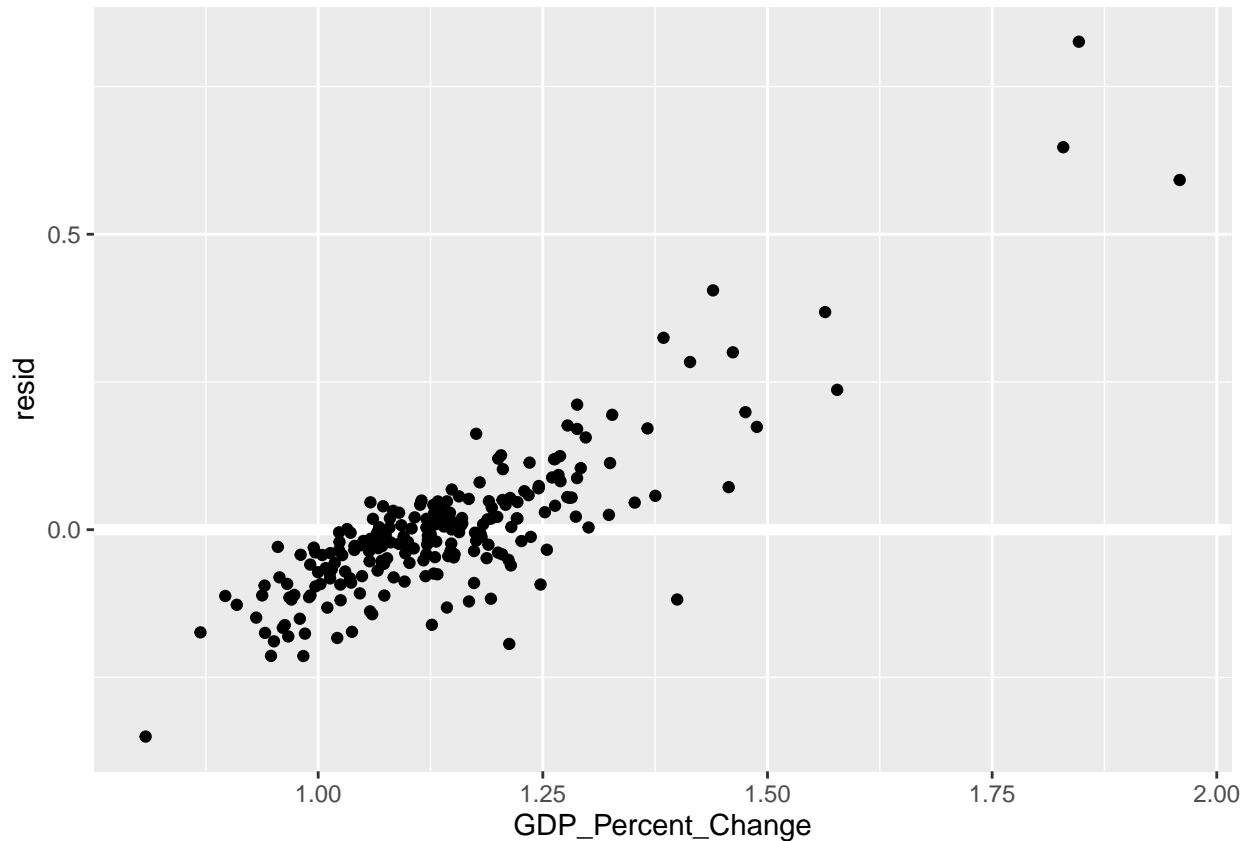
```
## [1] 0.1266896
```

```

#Residual calculation and plot graph
resids <- add_residuals(validate, validate_model)

```

```
ggplot(data=resids, mapping = aes(x=GDP_Percent_Change, y = resid)) +
  geom_ref_line(h = 0) +
  geom_point()
```

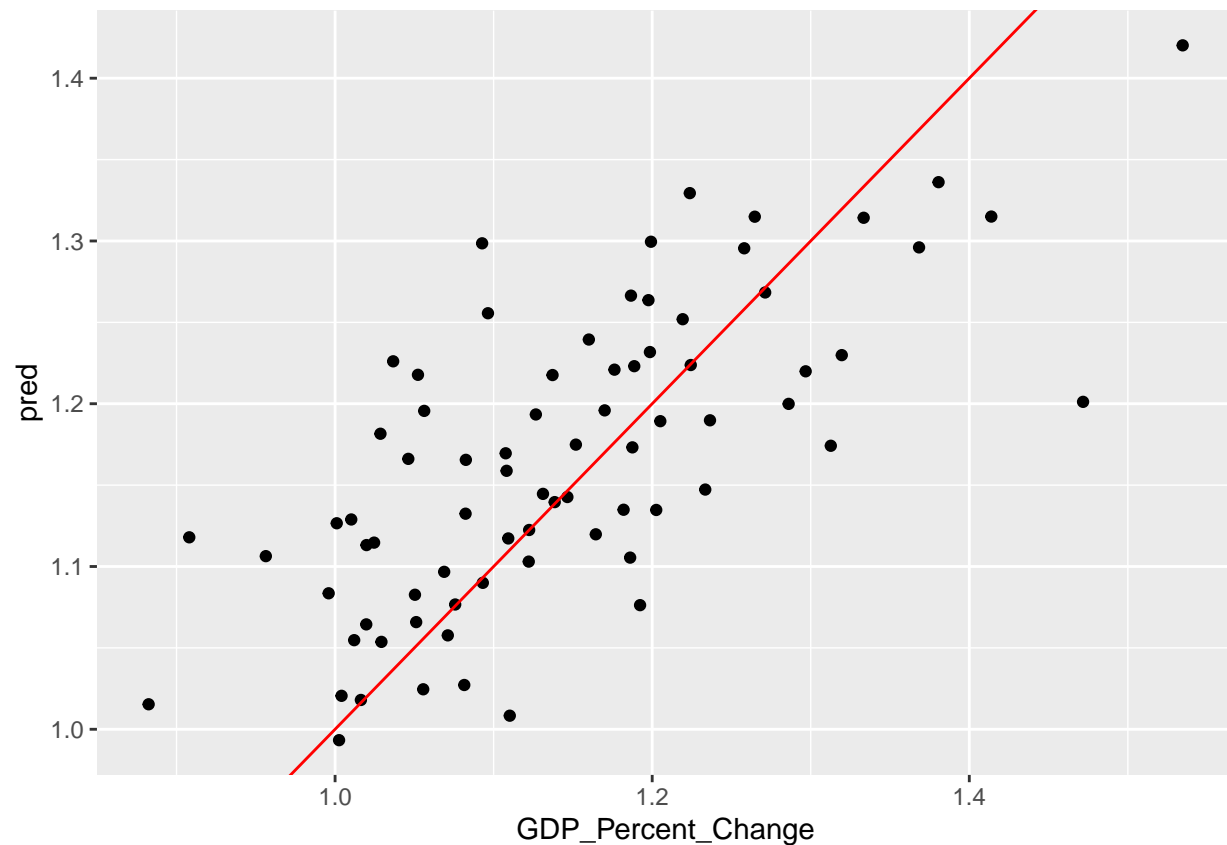


Note 7 - Here I trained my data with the rest and used a cross validation approach to cut it up more. Then I used the same strategies as I did with my previous model on this one as well as graphing a residual. I was happy with my data and this was the last step I decided to take.

```
#K-fold cross-validation approach.
train.control <- trainControl(method = "cv", number = 5)
final_model <- train(GDP_Percent_Change ~ Population_Percent_Change, data = rest, method = "lm",
  trControl = train.control)

predictions <- add_predictions(test, final_model)

ggplot(data = predictions, mapping = aes(x = GDP_Percent_Change, y = pred)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red")
```



```
#Goodness of fit on new predictions
R2(predictions$pred, predictions$GDP_Percent_Change)
```

```
## [1] 0.5078697
```

```
MAE(predictions$pred, predictions$GDP_Percent_Change)
```

```
## [1] 0.06900216
```

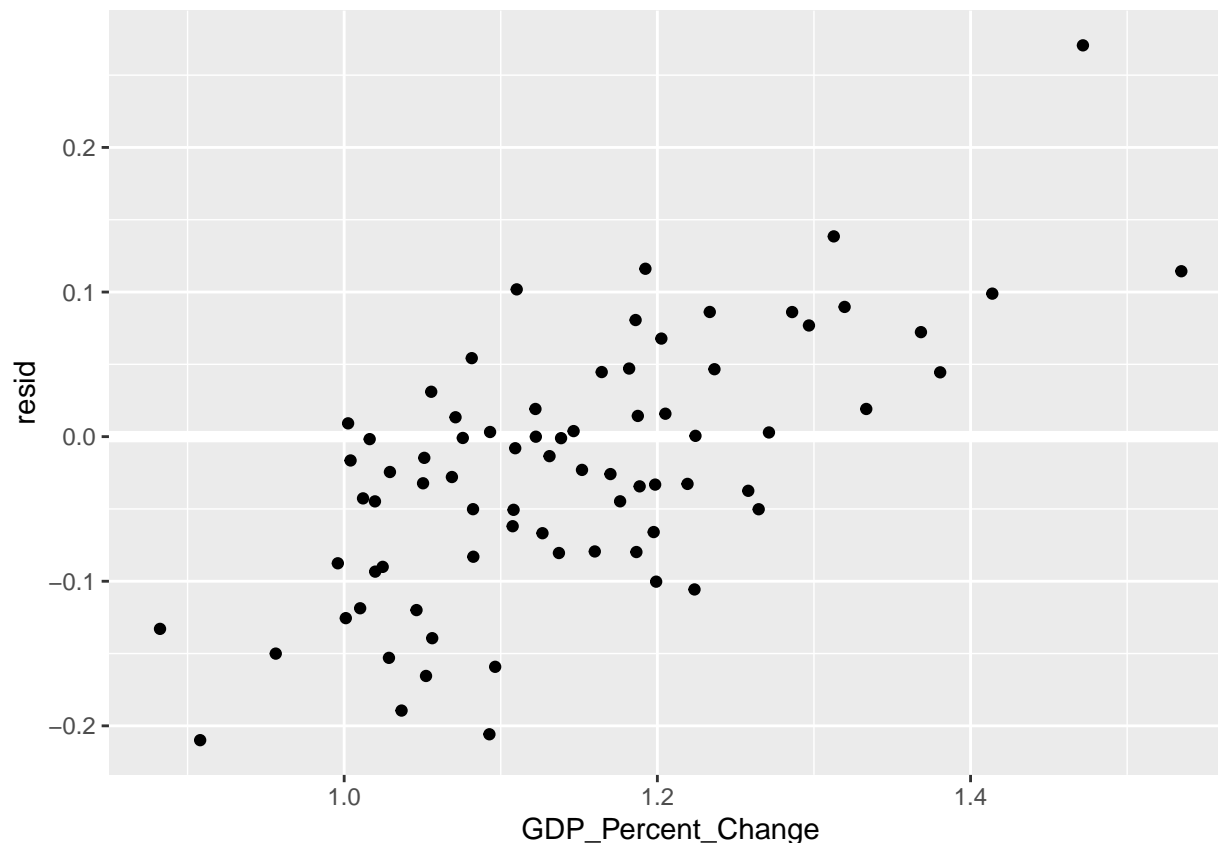
```
RMSE(predictions$pred, predictions$GDP_Percent_Change)
```

```
## [1] 0.08938721
```

```
resids <- add_residuals(test, final_model)
```

```
ggplot(data = resids, mapping = aes(x=GDP_Percent_Change, y = resid)) +
  geom_ref_line(h = 0) +
  geom_point()
```





Note 8 - I used the predict function to predict how a cities GDP (essentially economy) growth would be affected by higher populations. From this, I found that metro area's whose population grows not only can expect an equal amount of GDP growth per capita, but as a matter of a fact more. This goes to prove that area's that are more heavily populated make not only the city or area as a whole richer but rather each individual as well. I suppose this is good evidence that living in big cities is a better idea if you want to gain capital and have a higher standard of living than in a smaller area.

```
#Predictions to test how I can work with model
predict(final_model, data.frame(Population_Percent_Change = c(1.00, 1.25, 1.50)))
```

```
##          1          2          3
## 1.073902 1.435006 1.796109
```

## Part 4 - Using Google Maps API and graphing Metro Areas by Population Change

Just for fun, I wanted to try to see if I could get an idea of what cities were growing fastest in the last decade (which might give a good idea of what cities will be economic powerhouses). I connected to the Google maps API and took their location data which I then graphed below.

```

City_GDP_2018_Data_For_Cities <- data.frame(City_GDP_Data) %>%
  slice(2:52)

#Google Maps API Key to have access to data
register_google(key = "AIzaSyByXCki-hIHBM_HzbK_IE8d2xMZZYXEGLM")

#Used City names and Google Maps API to get coordinates and binded that to my data frame
City_GDP_2018_Data_For_Cities <- cbind(geocode(as.character(City_GDP_2018_Data_For_Cities$City)), City_GDP_2018_Data_For_Cities)

#Made all NA vaules 0 to avoid errors in code
City_GDP_2018_Data_For_Cities[is.na(City_GDP_2018_Data_For_Cities)] <- 0

#Transformed coordinate data to be readable by usmap plot
City_data_transformed <- usmap_transform(City_GDP_2018_Data_For_Cities)

plot_usmap(fill = "grey", alpha = 0.25) +
  geom_point(data=City_data_transformed, aes(x=lon.1, y=lat.1, size=Population_Change_Decade), color="green") +
  labs(title = "Graphed US Population Change Data", size = "Population Change Over the Decade") +
  theme(legend.position = "right")

```

## Graphed US Population Change Data



## Conclusion

Overall I would call this modeling process a success. My residuals in the assignment never seemed too far off and I felt like I got lucky with choosing the Population percentage change as my independent variable.

One thing that I may want to add next time is just more data overall. In this assignment more rural areas were completely left out which could be omitting a big chunk of the story, something I hope to include in the next phase of the project. I also could potentially come up with a better model, maybe even a non-linear one that can predict better and more accurately across the board. On this assignment, I've learned that no model is perfect. In the real world there are too many variables to keep up with and without a time machine predicting anything with 100% accuracy isn't possible. I have learned though that models can be useful and even interesting. I was really excited to use and expand my data in this assignment and was surprised with many of the findings. Hopefully in the last phase of the project I can dive even deeper.

## **Ethics**

On a national scale, wealth and economy have an intense impact on many factors of citizens lives from life expectancy rate to overall quality. With population and growth as a input to predict GDP percentage change in this project, I feel that I got a clear idea of how areas with a larger population can grow to make all its residents lives better. My predictive model in this assignment alongside my data can paint a decent picture of how many metropolitan area's can look in the future. It also can make it easier to notice both state and region growth which may help influence where one may decide to move, invest in, start a company, or do anything else in that may be helped by knowing its future economic state. Next time, I hope to add more data (hopefully involving industry and businesses) to what already exist see if it is possible to more accurately predict what area's will grow faster economically in the future. I also hope to create new and more interesting visualizations that can make the results of my models even more obvious to average person with no background knowledge on statistics.