Names: Tilak Ghorashainee, Edgar Lara, Allen Rivas

Project 1: COVID19 Data Wrangling

Part 1:

Steps for Data Wrangling:

1. **We loaded the three different CSV files by using tidyverse's read.csv. Stored global_data/time_series_covid19_vaccine_doses_admin_global.csv into covid19. Then API_NY.GDP.MKTP.CD_DS2_en_csv_v2_3011433.csv into gdp. Finally, we stored demographics.csv into demographics.**

```
> covid19  <-
read.csv("https://raw.githubusercontent.com/govex/COVID-19/master/data_tables/vaccine_data/global_data/time_series_covid19_vaccine_doses_admin_global.csv")

> gdp <-
read_csv("C:/Users/user/Downloads/API_NY.GDP.MKTP.CD_DS2_en_csv_v2_3011433.csv")

> demographics <- read_csv("C:/Users/user/Downloads/demographics.csv")
```

2. **We tidied the covid19 CSV file. We did this by pivot_longer to decrease the number of columns and create one column that had all the dates. Next, we selected the columns we needed from covid19. We then filtered out the rows which had shots of 0. Finally, we group by the iso of each country and mutated it to add two new variables being the vacRate and daysSinceStart. Then saved it into a different variable called newcovid19.**

```
> newcovid19 <- covid19 %>% pivot_longer(cols = starts_with("X"), names_to
= "Date", values_to = "shots",  values_drop_na = TRUE) %>%
select(c("iso3","Country_Region","Population","shots" )) %>%
filter(!if_any(starts_with("shots"), ~ . == 0)) %>% group_by(iso3) %>%
mutate(vacRate =(shots/Population), daysSinceStart=row_number())
```

Output of newcovid19:

```
# A tibble: 97,173 x 6
# Groups:   iso3 [147]
   iso3  Country_Region Population shots  vacRate daysSinceStart
```

```
   <fct> <fct>                <int> <dbl>    <dbl>          <int>
 1 AFG   Afghanistan       38928341  8200 0.000211              1
 2 AFG   Afghanistan       38928341  8200 0.000211              2
 3 AFG   Afghanistan       38928341  8200 0.000211              3
 4 AFG   Afghanistan       38928341  8200 0.000211              4
 5 AFG   Afghanistan       38928341  8200 0.000211              5
 6 AFG   Afghanistan       38928341  8200 0.000211              6
 7 AFG   Afghanistan       38928341  8200 0.000211              7
 8 AFG   Afghanistan       38928341  8200 0.000211              8
 9 AFG   Afghanistan       38928341  8200 0.000211              9
10 AFG   Afghanistan       38928341  8200 0.000211             10
# ... with 97,163 more rows
```

3. **We tidied the gdp CSV file. We did this by selecting the code for each country which is important for later when joining all the tables. Since we want the recent GDP we selected the year 2020. Then renaming 2020 to GDP was for data readability. Finally, saving it into a different variable called newgdp.**

```
> newgdp <- gdp %>% select(c("Country Code","2020")) %>% rename(GDP =
"2020")
```

Output of newgdp:

```
# A tibble: 266 x 2
   `Country Code`      GDP
   <chr>             <dbl>
 1 ABW              NA
 2 AFE               8.98e11
 3 AFG               1.98e10
 4 AFW               7.87e11
 5 AGO               6.23e10
 6 ALB               1.48e10
 7 AND              NA
 8 ARB               2.53e12
 9 ARE              NA
10 ARG               3.83e11
# ... with 256 more rows
```

4. **We tidied the demographics CSV file. We did this by using a pivot wider so we could make each variable under the Series code into a column. Then we mutated all the columns which were under the Series Code and combined all the variables which had male and female counterparts into one singular variable. Finally saving it into a different variable called newdemographics.**

```
> newdemographics <- demographics %>% pivot_wider(names_from = 'Series
Code', values_from = 'YR2015', -c('Series Name')) %>%
```

```
mutate(SP.POP.80UP=SP.POP.80UP.FE+SP.POP.80UP.MA,
SP.POP.1564.IN=SP.POP.1564.MA.IN+SP.POP.1564.FE.IN,
SP.POP.0014.IN=SP.POP.0014.MA.IN+SP.POP.0014.FE.IN,
SP.DYN.AMRT=SP.DYN.AMRT.FE+SP.DYN.AMRT.MA,
SP.POP.TOTL.IN=SP.POP.TOTL.FE.IN+SP.POP.TOTL.MA.IN,
SP.POP.65UP.IN=SP.POP.65UP.FE.IN+SP.POP.65UP.MA.IN) %>%
pivot_wider(-c(SP.POP.80UP.FE, SP.POP.80UP.MA, SP.POP.1564.MA.IN,
SP.POP.1564.FE.IN, SP.POP.0014.MA.IN, SP.POP.0014.FE.IN, SP.DYN.AMRT.FE,
SP.DYN.AMRT.MA, SP.POP.TOTL.FE.IN, SP.POP.TOTL.MA.IN, SP.POP.65UP.FE.IN,
SP.POP.65UP.MA.IN)) %>% select(c("Country Code", "SP.DYN.LE00.IN",
"SP.URB.TOTL", "SP.POP.TOTL", "SP.POP.80UP", "SP.POP.1564.IN",
"SP.POP.0014.IN", "SP.DYN.AMRT", "SP.POP.TOTL.IN", "SP.POP.65UP.IN"))
```

Output of newdemographics:

```
# A tibble: 259 x 10
   `Country Code` SP.DYN.LE00.IN SP.URB.TOTL SP.POP.TOTL SP.POP.80UP SP.POP.1564.IN SP.POP.0014.IN SP.DYN.AMRT SP.POP.TOTL.IN SP.POP.65UP.IN
   <chr>                   <dbl>       <dbl>       <dbl>       <dbl>          <dbl>          <dbl>       <dbl>          <dbl>          <dbl>
 1 AFG                      63.4     8535606    34413603       85552       18116800       15443807         455.       34413603         852996
 2 ALB                      78.0     1654503     2880703       66965        1979175         537788         150.        2880703         363740
 3 DZA                      76.1    28146511    39728025      453741       25993589       11404930         192.       39728025        2329506
 4 ASM                      NA        48689       55812          NA             NA             NA          NA            NA             NA
 5 AND                      NA        68919       78011          NA             NA             NA          NA            NA             NA
 6 AGO                      59.4    17691524    27884381       69363       14113726       13136043         486.       27884381         634612
 7 ATG                      76.5       23392       93566        1571          64812          21121         260.          93566           7634
 8 ARB                      71.2   229821020   396028278     2689793      248365376      130629537         277.      396028278       17033367
 9 ARG                      76.1    39467043    43131966     1095211       27630345       10874072         234.       43131966        4627549
10 ARM                      74.5     1845585     2925553       77292        2019878         587451         251.        2925553         318224
# ... with 249 more rows
```

5. **We first used a full_join to join newgdp and newdemographics based on their Country Code. Which was saved into a new variable called GDPdemographics.**

```
> GDPdemographics <- newgdp %>% full_join(newdemographics, by=c("Country
Code" = "Country Code"))
```

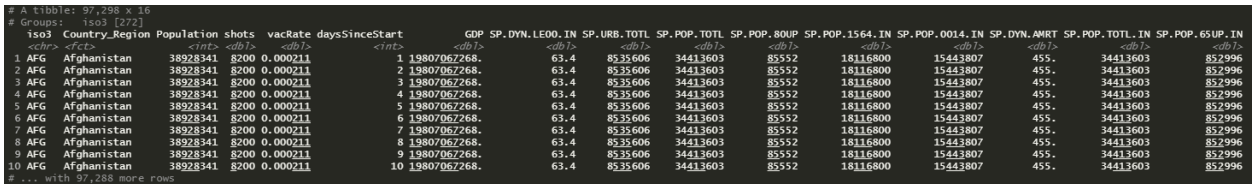
Output of GDPdemographics:

```
# A tibble: 266 x 11
   `Country Code`      GDP SP.DYN.LE00.IN SP.URB.TOTL SP.POP.TOTL SP.POP.80UP SP.POP.1564.IN SP.POP.0014.IN SP.DYN.AMRT SP.POP.TOTL.IN SP.POP.65UP.IN
   <chr>             <dbl>          <dbl>       <dbl>       <dbl>       <dbl>          <dbl>          <dbl>       <dbl>          <dbl>          <dbl>
 1 ABW                  NA             75.7       44979      104341        2103          72164          19515         187.         104341          12662
 2 AFE            8.98e+11             NA          NA          NA          NA             NA             NA          NA             NA             NA
 3 AFG            1.98e+10             63.4     8535606    34413603       85552       18116800       15443807         455.       34413603         852996
 4 AFW            7.87e+11             NA          NA          NA          NA             NA             NA          NA             NA             NA
 5 AGO            6.23e+10             59.4    17691524    27884381       69363       14113726       13136043         486.       27884381         634612
 6 ALB            1.48e+10             78.0     1654503     2880703       66965        1979175         537788         150.        2880703         363740
 7 AND                  NA             NA          68919       78011          NA             NA             NA          NA             NA             NA
 8 ARB            2.53e+12             71.2   229821020   396028278     2689793      248365376      130629537         277.      396028278       17033367
 9 ARE                  NA             77.3     7935897     9262900       10385        7864454        1311989         137.        9262900          86457
10 ARG            3.83e+11             76.1    39467043    43131966     1095211       27630345       10874072         234.       43131966        4627549
# ... with 256 more rows
```

6. **Then we used a full_join to join newcovid19 and GDPdemographics based on the iso3 for newcovid19 and Country Code for GDPdemographics. It was then stored into the variable called tidycovid19.**

```
> tidycovid19 <- newcovid19 %>% full_join(GDPdemographics, by=c("iso3" =
"Country Code"))
```

Output of tidycovid19:



```
# A tibble: 97,298 x 16
# Groups:   iso3 [272]
   iso3  Country_Region Population shots  vacRate daysSinceStart         GDP SP.DYN.LE00.IN SP.URB.TOTL SP.POP.TOTL SP.POP.80UP SP.POP.1564.IN SP.POP.0014.IN SP.DYN.AMRT SP.POP.TOTL.IN SP.POP.65UP.IN
   <chr> <fct>              <int> <dbl>   <dbl>          <int>       <dbl>         <dbl>       <dbl>       <dbl>        <dbl>          <dbl>          <dbl>          <dbl>          <dbl>          <dbl>
 1 AFG   Afghanistan     38928341  8200 0.000211              1 19807067268.          63.4     8535606    34413603       85552       18116800       15443807          455.       34413603         852996
 2 AFG   Afghanistan     38928341  8200 0.000211              2 19807067268.          63.4     8535606    34413603       85552       18116800       15443807          455.       34413603         852996
 3 AFG   Afghanistan     38928341  8200 0.000211              3 19807067268.          63.4     8535606    34413603       85552       18116800       15443807          455.       34413603         852996
 4 AFG   Afghanistan     38928341  8200 0.000211              4 19807067268.          63.4     8535606    34413603       85552       18116800       15443807          455.       34413603         852996
 5 AFG   Afghanistan     38928341  8200 0.000211              5 19807067268.          63.4     8535606    34413603       85552       18116800       15443807          455.       34413603         852996
 6 AFG   Afghanistan     38928341  8200 0.000211              6 19807067268.          63.4     8535606    34413603       85552       18116800       15443807          455.       34413603         852996
 7 AFG   Afghanistan     38928341  8200 0.000211              7 19807067268.          63.4     8535606    34413603       85552       18116800       15443807          455.       34413603         852996
 8 AFG   Afghanistan     38928341  8200 0.000211              8 19807067268.          63.4     8535606    34413603       85552       18116800       15443807          455.       34413603         852996
 9 AFG   Afghanistan     38928341  8200 0.000211              9 19807067268.          63.4     8535606    34413603       85552       18116800       15443807          455.       34413603         852996
10 AFG   Afghanistan     38928341  8200 0.000211             10 19807067268.          63.4     8535606    34413603       85552       18116800       15443807          455.       34413603         852996
# ... with 97,288 more rows
```

tidycovid19 is our final dataset where all three datasets are combined into one.

Part 2:

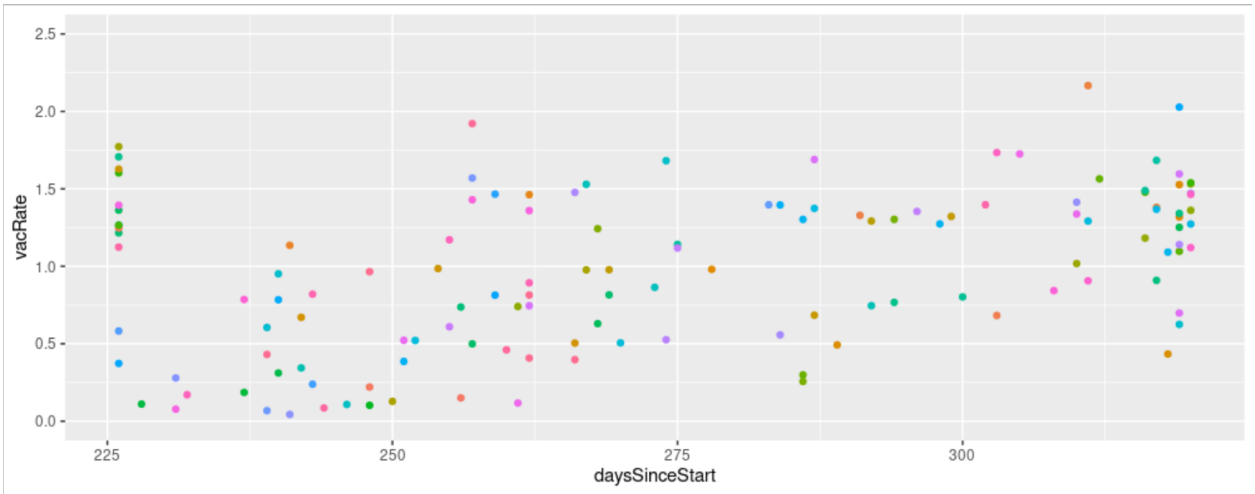
Since we needed a scatter plot of only the most recent vaccination rate we decided to store that into a variable called tidycovid19.2. Where we use top_n to filter the data by largest daysSinceStart in order to obtain the value of each country since their start of vaccinations. According to the project description this is a major factor in contributing to the vaccination rate.

```
> tidycovid19.2 <- tidycovid19 %>% top_n(1,daysSinceStart)
```

We then plotted this by using ggplot2 and since we needed a scatterplot we used geom_point. Where we plot daysSinceStart as our x-axis and vacRate as our y-axis.

From the project requirements, it said to model vaccination rate therefore our group decided that meant to choose it as the dependent variable.

```
> ggplot(data=tidycovid19.2) +
geom_point(mapping=aes(x=daysSinceStart,y=vacRate,color=iso3), show.legend
= F) + scale_x_continuous(limits=c(226,320)) + scale_y_continuous(limits =
c(0.0,2.5))
Warning message:
Removed 491 rows containing missing values (geom_point).
```



As for the predictor variables, we decided to just go by column order. As we figure, the more predictor variables included per model the more data the model has as a result has a potential at increasing the strength R-Squared value. Since the closer R-Squared is to 1 indicates a very accurate model.

Summary of linear model for daysSinceStart:

```
> lmDaysSinceStart <- summary(lm(data = tidycovid19,
vacRate~daysSinceStart))
> lmDaysSinceStart

Call:
lm(formula = vacRate ~ daysSinceStart, data = tidycovid19)

Residuals:
      Min    1Q  Median        3Q     Max
-1.1374 -0.4243 -0.1578  0.3468  2.0692

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.648e-01  2.547e-03  182.49   <2e-16 ***
daysSinceStart 1.886e-04  2.538e-06   74.29   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4878 on 47911 degrees of freedom
  (49385 observations deleted due to missingness)
Multiple R-squared:  0.1033,    Adjusted R-squared:  0.1033
F-statistic:  5519 on 1 and 47911 DF,  p-value: < 2.2e-16
```

Summary of linear model for daysSinceStart + Population:

```
> lmPopulation <- summary(lm(data = tidycovid19, vacRate~daysSinceStart +
Population))
> lmPopulation

Call:
lm(formula = vacRate ~ daysSinceStart + Population, data = tidycovid19)

Residuals:
      Min    1Q  Median        3Q     Max
-1.1321 -0.4233 -0.1557  0.3465  2.0616

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.730e-01  2.637e-03  179.35   <2e-16 ***
daysSinceStart  1.858e-04  2.546e-06   72.96   <2e-16 ***
Population  -1.643e-10  1.397e-11  -11.76   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4871 on 47910 degrees of freedom
  (49385 observations deleted due to missingness)
Multiple R-squared:  0.1059,    Adjusted R-squared:  0.1058
F-statistic:  2837 on 2 and 47910 DF,  p-value: < 2.2e-16
```

Summary of linear model for daysSinceStart + Population + GDP:

```
> lmGDP <- summary(lm(data = tidycovid19, vacRate~daysSinceStart +
Population + GDP ))
> lmGDP

Call:
lm(formula = vacRate ~ daysSinceStart + Population + GDP, data =
tidycovid19)

Residuals:
      Min    1Q  Median        3Q     Max
-1.1475 -0.3664 -0.1475  0.3024  1.6800

Coefficients:
```

```
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       4.044e-01  2.624e-03  154.11   <2e-16 ***
daysSinceStart  1.826e-04  2.449e-06   74.55   <2e-16 ***
Population  -3.982e-10  1.564e-11  -25.45   <2e-16 ***
GDP               4.173e-14  1.174e-15   35.55   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4472 on 42590 degrees of freedom
  (54704 observations deleted due to missingness)
Multiple R-squared:  0.1746,    Adjusted R-squared:  0.1746
F-statistic:  3004 on 3 and 42590 DF,  p-value: < 2.2e-16
```

Summary of linear model for daysSinceStart + Population + GDP + SP.DYN.LE00.IN:

```
> lmSPDYNLEOOIN <- summary(lm(data = tidycovid19, vacRate~daysSinceStart +
Population + GDP + SP.DYN.LE00.IN))
> lmSPDYNLEOOIN
Call:
lm(formula = vacRate ~ daysSinceStart + Population + GDP + SP.DYN.LE00.IN,
      data = tidycovid19)

Residuals:
      Min    1Q  Median        3Q    Max
-1.0223 -0.3243 -0.0541  0.2701  1.6999

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.694e+00  2.516e-02 -67.323  < 2e-16 ***
daysSinceStart  1.184e-04  2.405e-06  49.246  < 2e-16 ***
Population  -1.221e-10  1.491e-11  -8.188 2.74e-16 ***
GDP               1.784e-14  1.128e-15  15.815  < 2e-16 ***
SP.DYN.LE00.IN  2.818e-02  3.366e-04  83.733  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4156 on 42120 degrees of freedom
  (55173 observations deleted due to missingness)
Multiple R-squared:  0.2931,    Adjusted R-squared:  0.293
F-statistic:  4366 on 4 and 42120 DF,  p-value: < 2.2e-16
```

Summary of linear model for daysSinceStart + Population + GDP + SP.DYN.LE00.IN + SP.POP.TOTL:

```
> lmPopTotl <- summary(lm(data = tidycovid19, vacRate~daysSinceStart +
Population + GDP + SP.DYN.LE00.IN + SP.POP.TOTL))
> lmPopTotl

Call:
lm(formula = vacRate ~ daysSinceStart + Population + GDP + SP.DYN.LE00.IN +
      SP.POP.TOTL, data = tidycovid19)

Residuals:
      Min         1Q   Median       3Q    Max
-0.99623 -0.32608 -0.05547  0.27249  1.70429

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.656e+00  2.589e-02 -63.944  < 2e-16 ***
daysSinceStart  1.006e-04  3.763e-06  26.719  < 2e-16 ***
Population  -1.169e-09  1.701e-10  -6.871 6.46e-12 ***
GDP               1.620e-14  1.159e-15  13.984  < 2e-16 ***
SP.DYN.LE00.IN  2.767e-02  3.465e-04  79.842  < 2e-16 ***
SP.POP.TOTL       1.104e-09  1.788e-10   6.177 6.58e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4154 on 42119 degrees of freedom
  (55173 observations deleted due to missingness)
Multiple R-squared:  0.2937,    Adjusted R-squared:  0.2936
F-statistic:  3503 on 5 and 42119 DF,  p-value: < 2.2e-16
```

We added a transformation variable called Population.Proportion where it calculates using population divide by SP.POP.TOTL. In order to help in increasing the strength of the model.

```
tidycovid19 <- tidycovid19 %>% mutate(Population.Proportion =
Population/SP.POP.TOTL)
```

Summary of linear model for Population.Proportion:

```
> lmPopulation.Prop <- summary(lm(data=tidycovid19,vacRate~daysSinceStart +
```

```
Population + GDP+ SP.DYN.LE00.IN + SP.POP.TOTL+Population.Proportion))
> lmPopulation.Prop

Call:
lm(formula = vacRate ~ daysSinceStart + Population + GDP + SP.DYN.LE00.IN +
      SP.POP.TOTL + Population.Proportion, data = tidycovid19)

Residuals:
      Min         1Q   Median       3Q    Max
-0.97792 -0.32658 -0.05541  0.27263  1.70750

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)             -1.532e+00  3.334e-02 -45.939  < 2e-16 ***
daysSinceStart           9.794e-05  3.788e-06  25.858  < 2e-16 ***
Population               5.129e-10  3.320e-10   1.545   0.1224
GDP                      1.745e-14  1.177e-15  14.823  < 2e-16 ***
SP.DYN.LE00.IN           2.735e-02  3.507e-04  77.974  < 2e-16 ***
SP.POP.TOTL             -6.509e-10  3.472e-10  -1.875   0.0608 .
Population.Proportion -9.743e-02  1.652e-02  -5.898 3.71e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4153 on 42118 degrees of freedom
  (55173 observations deleted due to missingness)
Multiple R-squared:  0.2943,    Adjusted R-squared:  0.2942
F-statistic:  2928 on 6 and 42118 DF,  p-value: < 2.2e-16
```

Summary of linear model for all predictor variables:

```
> lm.all.predictor <-
summary(lm(data=tidycovid19,vacRate~iso3+daysSinceStart + Population + GDP+
SP.DYN.LE00.IN + SP.URB.TOTL + SP.POP.TOTL + SP.POP.80UP+ SP.POP.1564.IN+
SP.POP.0014.IN + SP.DYN.AMRT + SP.POP.65UP.IN + Population.Proportion))
> lm.all.predictor

Call:
lm(formula = vacRate ~ iso3 + daysSinceStart + Population + GDP +
      SP.DYN.LE00.IN + SP.URB.TOTL + SP.POP.TOTL + SP.POP.80UP +
      SP.POP.1564.IN + SP.POP.0014.IN + SP.DYN.AMRT + SP.POP.65UP.IN +
      Population.Proportion, data = tidycovid19)
```

```
Residuals:
      Min         1Q   Median       3Q    Max
-1.20298 -0.17171 -0.01737  0.15720  1.30177

Coefficients: (7 not defined because of singularities)
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          6.582e+05  1.883e+08   0.003  0.997
iso3AGO             -1.100e+07  2.286e+08  -0.048 0.962
iso3ALB              2.253e+04  3.803e+07   0.001  1.000
iso3ARG             -2.267e+07  1.696e+09  -0.013 0.989
iso3ATG             -8.932e+05  1.891e+08  -0.005 0.996
iso3AUT             -7.439e+07  6.411e+08  -0.116 0.908
iso3AZE             -3.724e+06  1.819e+07  -0.205 0.838
iso3BEL             -8.311e+07  4.725e+08  -0.176 0.860
iso3BGD              8.374e+06  2.806e+09   0.003  0.998
iso3BHS             -3.051e+06  2.218e+08  -0.014 0.989
iso3BLR              5.854e+06  6.133e+08   0.010  0.992
iso3BLZ             -8.578e+05  1.846e+08  -0.005 0.996
iso3BOL              4.964e+05  1.572e+08   0.003  0.997
iso3BRA             -1.438e+08  3.907e+09  -0.037 0.971
iso3BRB             -9.386e+05  1.684e+08  -0.006 0.996
iso3CHE             -1.508e+08  2.035e+09  -0.074 0.941
iso3CHL             -3.187e+07  2.465e+08  -0.129 0.897
iso3CHN             -2.153e+09  1.058e+10  -0.203 0.839
iso3CIV             -1.144e+07  2.582e+08  -0.044 0.965
iso3COL             -1.724e+07  1.097e+09  -0.016 0.987
iso3CRI             -9.611e+06  1.650e+08  -0.058 0.954
iso3CYP             -4.212e+06  1.828e+08  -0.023 0.982
iso3CZE             -3.158e+07  1.977e+08  -0.160 0.873
iso3DEU             -6.059e+08  2.670e+09  -0.227 0.821
iso3DNK             -6.921e+07  9.245e+08  -0.075 0.940
iso3DOM             -1.028e+07  7.174e+07  -0.143 0.886
iso3DZA             -6.596e+06  6.472e+08  -0.010 0.992
iso3ECU             -9.474e+06  1.435e+08  -0.066 0.947
iso3EGY             -4.265e+07  5.532e+08  -0.077 0.939
iso3ESP             -1.274e+08  3.455e+09  -0.037 0.971
iso3EST             -3.837e+06  1.076e+08  -0.036 0.972
iso3FIN             -4.710e+07  4.593e+08  -0.103 0.918
iso3FRA             -3.669e+08  2.203e+09  -0.167 0.868
iso3GBR             -4.393e+08  2.152e+09  -0.204 0.838
iso3GHA             -1.256e+07  2.304e+08  -0.054 0.957
iso3GIN             -2.234e+06  1.456e+08  -0.015 0.988
iso3GMB             -8.025e+05  1.807e+08  -0.004 0.996
```

```
iso3GNQ               -2.813e+06  2.193e+08  -0.013 0.990
iso3GRC               -1.339e+06  1.259e+09  -0.001 0.999
iso3GRD               -7.654e+05  1.852e+08  -0.004 0.997
iso3GTM               -9.006e+06  3.662e+07  -0.246 0.806
iso3GUY               -1.484e+06  1.874e+08  -0.008 0.994
iso3HKG               -6.211e+07  6.402e+08  -0.097 0.923
iso3HND               -8.227e+05  1.073e+07  -0.077 0.939
iso3HRV               -6.602e+05  2.494e+08  -0.003 0.998
iso3HUN               -1.253e+07  4.261e+08  -0.029 0.977
iso3IDN               -1.312e+08  1.573e+09  -0.083 0.933
iso3IND               5.511e+07  2.316e+10   0.002  0.998
iso3IRN               6.464e+06  1.672e+09   0.004  0.997
iso3ISL               -5.003e+06  2.393e+08  -0.021 0.983
iso3ISR               -7.943e+07  1.081e+09  -0.073 0.941
iso3JAM               -2.813e+05  6.115e+07  -0.005 0.996
iso3JOR               -7.907e+06  2.159e+08  -0.037 0.971
iso3KAZ               -2.856e+07  2.958e+08  -0.097 0.923
iso3KEN               -1.686e+07  2.449e+08  -0.069 0.945
iso3KHM               -1.738e+06  4.565e+07  -0.038 0.970
iso3LAO               -2.977e+06  1.564e+08  -0.019 0.985
iso3LBN               -3.382e+06  6.645e+07  -0.051 0.959
iso3LCA               -8.202e+05  1.832e+08  -0.004 0.996
iso3LKA               4.325e+05  4.983e+08   0.001  0.999
iso3LTU               -4.606e+06  5.242e+07  -0.088 0.930
iso3LVA               -2.195e+06  9.743e+06  -0.225 0.822
iso3MAR               -4.902e+06  4.802e+08  -0.010 0.992
iso3MDA               1.331e+06  7.818e+06   0.170  0.865
iso3MDV               -1.411e+06  1.949e+08  -0.007 0.994
iso3MEX               -1.405e+08  1.283e+09  -0.110 0.913
iso3MKD               -6.785e+05  9.338e+07  -0.007 0.994
iso3MLT               -2.975e+06  1.914e+08  -0.016 0.988
iso3MMR               4.784e+06  6.835e+08   0.007  0.994
iso3MNE               -6.156e+05  1.488e+08  -0.004 0.997
iso3MNG               -2.684e+06  1.884e+08  -0.014 0.989
iso3MOZ               1.500e+06  3.128e+07   0.048  0.962
iso3MUS               -1.830e+06  1.624e+08  -0.011 0.991
iso3MWI               -5.666e+05  9.009e+07  -0.006 0.995
iso3MYS               -6.318e+07  7.643e+08  -0.083 0.934
iso3NAM               -2.247e+06  1.852e+08  -0.012 0.990
iso3NGA               -8.057e+07  9.130e+08  -0.088 0.930
iso3NLD               -1.691e+08  1.725e+09  -0.098 0.922
iso3NPL               3.056e+06  2.634e+08   0.012  0.991
iso3PAK               1.348e+07  2.591e+09   0.005  0.996
```

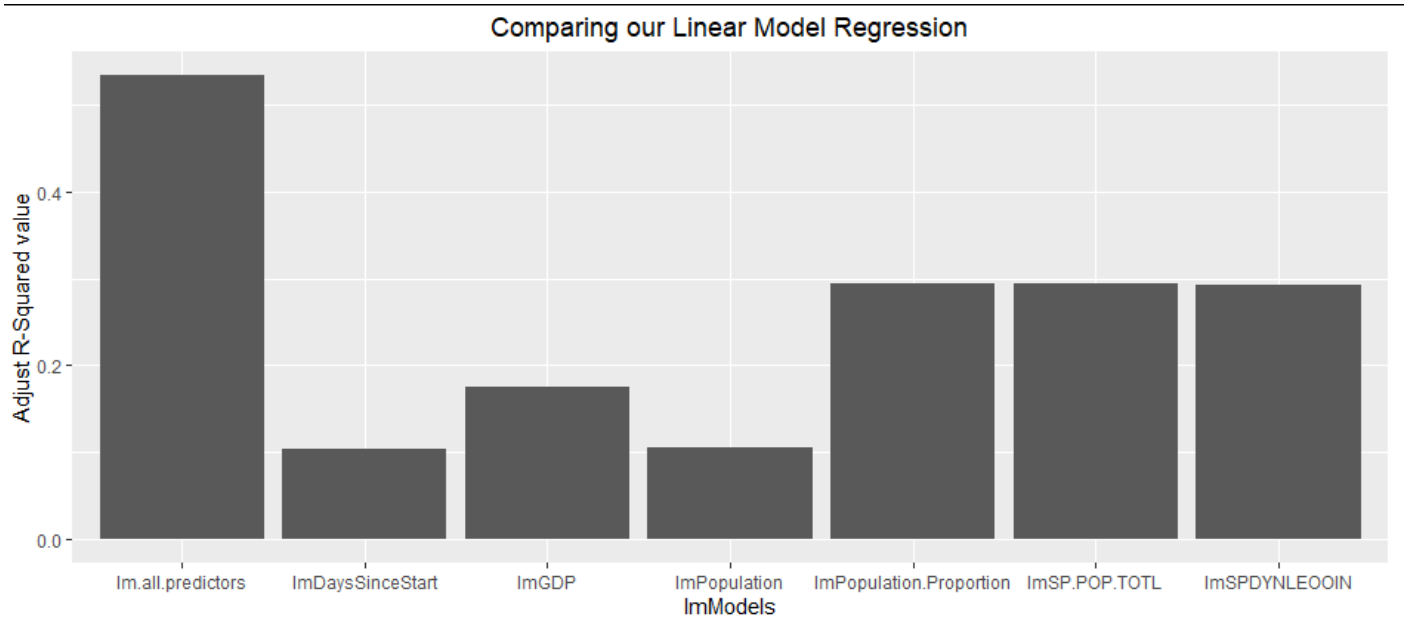
```
iso3PAN                -8.792e+06  1.894e+08  -0.046 0.963
iso3PER                -1.996e+07  4.223e+08  -0.047 0.962
iso3PHL                -4.335e+07  5.090e+08  -0.085 0.932
iso3POL                -4.464e+07  2.240e+09  -0.020 0.984
iso3PRT                -1.749e+07  7.716e+08  -0.023 0.982
iso3PRY                -4.421e+06  1.042e+08  -0.042 0.966
iso3QAT                -3.488e+07  7.888e+08  -0.044 0.965
iso3ROU                -8.085e+06  1.382e+09  -0.006 0.995
iso3RWA                -2.648e+05  8.752e+07  -0.003 0.998
iso3SAU                -1.561e+08  2.660e+09  -0.059 0.953
iso3SEN                -3.344e+06  1.301e+08  -0.026 0.979
iso3SGP                -7.430e+07  1.293e+09  -0.057 0.954
iso3SLE                -1.098e+05  1.303e+08  -0.001 0.999
iso3SLV                -9.220e+04  3.313e+07  -0.003 0.998
iso3SRB                3.200e+06  4.216e+08   0.008  0.994
iso3SUR                -1.140e+06  1.829e+08  -0.006 0.995
iso3SWE                -9.588e+07  8.928e+08  -0.107 0.914
iso3SYC                -8.244e+05  1.879e+08  -0.004 0.996
iso3TGO                -1.344e+06  1.638e+08  -0.008 0.993
iso3THA                -2.626e+07  2.443e+09  -0.011 0.991
iso3TTO                -4.323e+06  2.062e+08  -0.021 0.983
iso3TUN                1.194e+06  2.125e+08   0.006  0.996
iso3TUR                -9.639e+07  7.131e+08  -0.135 0.892
iso3UGA                -4.512e+06  9.287e+07  -0.049 0.961
iso3URY                -4.124e+06  5.821e+07  -0.071 0.944
iso3USA                -4.178e+09  5.084e+10  -0.082 0.934
iso3VCT                -6.930e+05  1.837e+08  -0.004 0.997
iso3VNM                4.634e+07  4.317e+09   0.011  0.991
iso3ZAF                -4.916e+07  3.452e+08  -0.142 0.887
iso3ZWE                -1.151e+06  8.218e+07  -0.014 0.989
daysSinceStart         1.248e-03  1.281e-05  97.482   <2e-16 ***
Population             -3.153e-08  1.079e-09 -29.209   <2e-16 ***
GDP                    2.351e-04  4.189e-03   0.056  0.955
SP.DYN.LE00.IN               NA         NA    NA    NA
SP.URB.TOTL                  NA         NA    NA    NA
SP.POP.TOTL                  NA         NA    NA    NA
SP.POP.80UP            -6.211e+01  3.155e+03  -0.020 0.984
SP.POP.1564.IN               NA         NA    NA    NA
SP.POP.0014.IN               NA         NA    NA    NA
SP.DYN.AMRT                  NA         NA    NA    NA
SP.POP.65UP.IN               NA         NA    NA    NA
Population.Proportion  2.599e+00  6.094e-02  42.645   <2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3122 on 31402 degrees of freedom
  (65781 observations deleted due to missingness)
Multiple R-squared:  0.5355,    Adjusted R-squared:  0.5338
F-statistic: 317.5 on 114 and 31402 DF,  p-value: < 2.2e-16
```

Comparing all the adjusted R2 values of our linear models:

Since all our models use different numbers of predictor variables we went ahead and adjusted r squared values. The reason being it accounts for the number of predictors used and gives a better representation of how much better the model is for the added predictors. The data in our project helps us find either the good or bad model.



Conclusion:

In the bar graph above we notice our best model is lm.all.predictors which is the one with the most predictor variables that includes all valid predictor variables. This model leads to a 0.5503509 value. This value was the nearest to 1. The closer the value is to 1, the better the model is at explaining the dependent variable from the set of predictor variables. We can interpret the 0.5503509 as our model is about 55 percent in explaining the vacRate seen for the entire data given to the model. When looking at the summary of the linear model we notice a lot of NA values. We tried to omit any NA values before acquiring the model but it still resulted in the same NA values. We are not entirely sure

what caused the NA but we think it led to a relatively low model prediction strength of about 55 percent but these are the results we reached. A reason we thought was that maybe Rstudio does not consider these to be a contributing factor to explaining the vaccination rate seen in the data. Finding the optimal model would require more variable transformation which has an infinite amount of possibilities. Since this code could be run daily to obtain new data the values shown will slightly vary as we obtain more data.