

CPSC 375 FINAL PROJECT REPORT

Members:

Hiep Do, hgdo2803@csu.fullerton.edu

Jon Sundin, jsundin@csu.fullerton.edu

Daisy Catalan, dcatalan@csu.fullerton.edu

Project Requirements

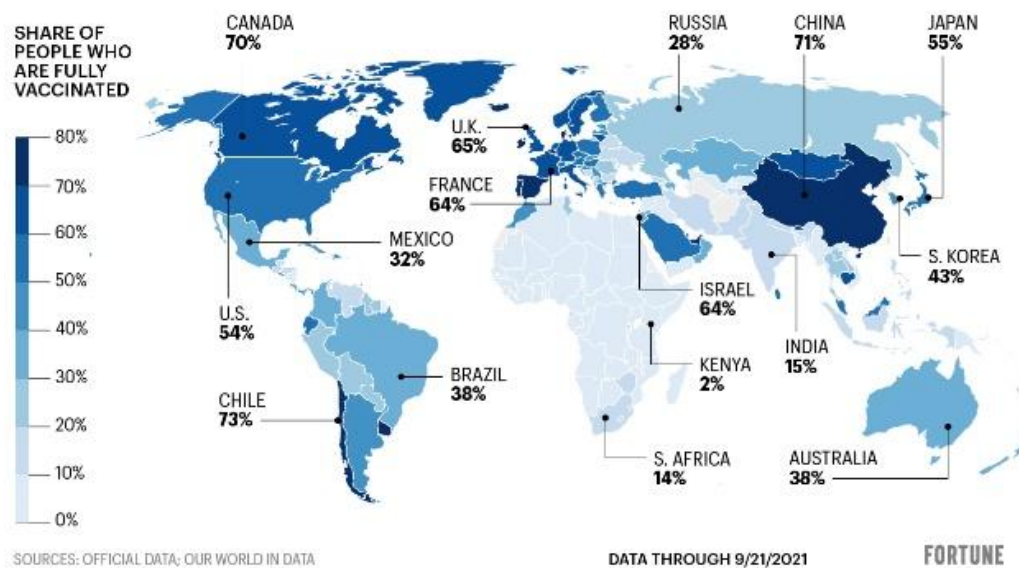


Image from: <https://fortune.com/2021/09/22/covid-vaccine-rate-world-us-latest-update-coronavirus-vaccines/>

COVID vaccination rates vary a great deal between countries. There are several reasons, including economics (richer countries tend to have higher rates) and demographics (countries prioritize recipients by age). The goal of this project is to use linear regression to model the vaccination rates in different countries in terms of their GDP (a measure of economic output) and demographics.

Datasets

1. [time_series_covid19_vaccine_doses_admin_global.csv](#)¹

¹ More information: [International vaccine data](#)

- This file contains the number of vaccine doses given every day in different countries. Note that you can read the “raw” CSV file from a URL directly, like so:
`read_csv("https://raw.githubusercontent.com/govex/COVID-19/master/data_tables/vaccine_data/global_data/time_series_covid19_vaccine_doses_admin_global.csv")`
- 2. GDP data in `API_NY.GDP.MKTP.CD_DS2_en_csv_v2_3011433.csv`
 - This file (available in the Datasets module on Canvas) contains the GDP of different countries in different years². You will use only the GDP from *the most recent year for a country*.
- 3. `demographics.csv`³
 - This gives the proportion of a country’s population in different age groups and some other demographic data such as mortality rates and expected lifetime.

There are two major steps in this project:

- 1) **Data preparation/wrangling** to get all the data into **one table** that can be used for linear modeling
 - a) reading the data files using `read_csv()`

```
library(tidyverse)
library(modelr)
library(ggplot2)

# setwd("~/R/cpsc_375_project")

vaccine_data_global <- read_csv(
  paste("https://raw.githubusercontent.com/",
        "govex/COVID-19/master/data_tables/",
        "vaccine_data/global_data/",
        "time_series_covid19_vaccine_doses_admin_global.csv", sep=""))

gdp_data_global <- read_csv("API_NY.GDP.MKTP.CD_DS2_en_csv_v2_3011433.csv")
```

² From <https://data.worldbank.org/indicator/NY.GDP.MKTP.CD>

³ Original dataset:

<https://databank.worldbank.org/source/population-estimates-and-projections/Type/TABLE/preview/on#>

```
demographics_data_global <- read_csv("demographics.csv")
```

- b) Removing unneeded rows (e.g., countries like Brazil and India report Province_State-level data that is not needed as we are studying only country-level rates) and columns.

```
# only including relevant columns with a positive select function
vaccine_data_global_reduced <- vaccine_data_global %>%
  select(iso3, Country_Region, Population, 13:ncol())

# omitting rows with NA values
vaccine_data_global_reduced <- na.omit(vaccine_data_global_reduced)

# renaming "Country Code" column to iso3, and "Country Name" to Country_Region
# selecting all columns that do not start with "Indicator"
gdp_data_global_reduced <- gdp_data_global %>%
  rename(iso3 = `Country Code`, Country_Region = `Country Name`) %>%
  select(!starts_with("Indicator"))

# renaming "Country Code" column to iso3, and "Country Name" to Country_Region
# Removing irrelevant `Series Name` Column
demographics_data_global_reduced <- demographics_data_global %>%
  rename(iso3 = `Country Code`, Country_Region = `Country Name`) %>%
  select(!`Series Name`)
```

- c) Tidying tables, as needed. For example, the vaccination data is not tidy.

```
# Picking the column names with year starting with 2 and putting them into a
# new "Date" column. Taking the values from the specific day columns and
# putting the values into a new column named "shots". Dropping any NA values.
vaccine_data_global_reduced_tidy <- vaccine_data_global_reduced %>%
  pivot_longer(cols = starts_with("2"), names_to = "Date",
```

```

      values_to = "shots", values_drop_na = TRUE)

# Picking column names of 3 to the end and making them the values in a new
# column called "Year". Adding the cell values from these columns and making
# a new column called "GDP". Dropping NA values.
gdp_data_global_reduced_tidy <- gdp_data_global_reduced %>%
  pivot_longer(cols = 3:ncol(.), names_to = "Year",
               values_to = "GDP", values_drop_na = TRUE)

# Arranging gdp_data_global_reduced_tidy by Country_Region and Year
# and grouping Country_Region values together and summarize_all applies to all
# variables in Year and the last Year is returned.
gdp_data_global_reduced_tidy <- gdp_data_global_reduced_tidy %>%
  arrange(Country_Region, Year) %>%
  group_by(Country_Region) %>% summarize_all(last)

# Making each Country as an observation in a single row by making unique
# "Series Code", which represent the demographic value into subsequent column
# names, and the population value corresponding to that demographic as a value
# with column name "YR2015"
demographics_data_global_reduced_tidy <- demographics_data_global_reduced %>%
  pivot_wider(names_from = `Series Code`, values_from = YR2015)

```

d) Calculate the vaccination rate: vaccinations/population

```
vaccine_data_global_reduced_tidy %>% mutate(vacRate = shots/Population)
```

e) Since the most important factor affecting vaccination rate is the number of days since vaccination began (vaccination rate always increases), calculate a variable that is: number of days since first non-zero vaccination number. This variable will be important for modeling.

```
# Days since vaccination began (vaccination rate always increases), calculate a
```

```
# variable that is: number of days since first non-zero vaccination number.
# This variable will be important for modeling.

vaccine_data_global_reduced_tidy_mutate <- vaccine_data_global_reduced_tidy %>%
  mutate(vacRate = shots/Population) %>% filter(shots > 0, na.rm = TRUE) %>%
  group_by(Country_Region) %>%
  mutate(daysSinceStart = row_number())
```

- f) Discard data that is not needed. For example, only the GDP of the most recent year is necessary.

```
# Selecting every column except Date and reordering them.
vaccine_data_global_reduced_tidy_mutate_clean <-
  vaccine_data_global_reduced_tidy_mutate %>%
  select(iso3, Country_Region, vacRate, shots, Population, daysSinceStart)

# We do not need the Country and Year column for GDP because we only need iso3
# to join the tables later on.
gdp_data_global_reduced_tidy_clean <- gdp_data_global_reduced_tidy %>%
  select(-Country_Region, -Year)

# Selecting all columns except Country_Region
demographics_data_global_reduced_tidy_clean <-
  demographics_data_global_reduced_tidy %>%
  select(-Country_Region)
```

- g) You can ignore sex-related differences in demographics in this project, so add the male/female population numbers together

```
# Creating categories/columns with both male and female combined
demographics_data_global_reduced_tidy_clean_unisex <-
  demographics_data_global_reduced_tidy_clean %>%
  mutate(SP.POP.80UP = SP.POP.80UP.FE + SP.POP.80UP.MA) %>%
```

```
mutate(SP.POP.1564.IN = SP.POP.1564.MA.IN + SP.POP.1564.FE.IN) %>%
mutate(SP.POP.0014.IN = SP.POP.0014.MA.IN + SP.POP.0014.FE.IN) %>%
mutate(SP.DYN.AMRT = (SP.DYN.AMRT.FE + SP.DYN.AMRT.MA) / 2 ) %>%
mutate(SP.POP.65UP.IN = SP.POP.65UP.FE.IN + SP.POP.65UP.MA.IN) %>%
select(c("iso3", "SP.DYN.LE00.IN", "SP.URB.TOTL", "SP.POP.TOTL",
        "SP.POP.80UP":ncol(.)))
```

h) Merge all tables (Hint: Join using the 3-letter ISO code for a country)

```
#renaming final datasets for clarity
vaccine_data_global_table <- vaccine_data_global_reduced_tidy_mutate_clean


gdp_data_global_table <- gdp_data_global_reduced_tidy_clean

demographics_data_global_table <-
demographics_data_global_reduced_tidy_clean_unisex

# Merging the 3 tables into one with inner join
merged_table <- vaccine_data_global_table %>%
inner_join(gdp_data_global_table) %>%
inner_join(demographics_data_global_table) %>% view()

# creating a csv file of the final table to be used for linear modeling
path <- "~/R/cpsc_375_project"
merged_table %>% write.csv(file.path(path, "merged_table.csv"), row.names =
FALSE)
```

With the steps above executed, the table below is made that shows vacRate (dependent variable) with respect to Country Region as well as the predictor variables (independent variables) from Population to the end of the data.frame table.



merged_table														
iso3	Country_Region	vacRate	shots	Population	daysSinceStart	GDP	SPDYN.LE00.IN	SPURB.TOTL	SPPORTOTL	SPPOP80UP	SPPOP1564.IN	SPPOP0014.IN	SPDYN.AMRT	SPPOP65UP.IN
DZA	Algeria	6.84134240547026E-07	30	43851043	1	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	2	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	3	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	4	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	5	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	6	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	7	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	8	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	9	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	10	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	11	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	12	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	13	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506
DZA	Algeria	6.84134240547026E-07	30	43851043	14	145163902228.168	76.09	28146511	39728025	453741	25993589	11404930	95.8155	2329506

2) Linear modeling the Covid vaccination rate

The below code produces a data.frame of the predictor variables from the merged table of data. With these predictor variables, five linear models will be created based on a combination of these variables.

```
# Make a list of all predictor variables that are available.
predictor_df <- merged_table %>% ungroup() %>% select(5:15)
predictor_df_names <- colnames(predictor_df)
view(predictor_df_names)
```

```
# PREDICTOR VARIABLES
# "Population"
# "daysSinceStart"
# "GDP"
# "SP.DYN.LE00.IN" is Life expectancy at birth, total (years)
# "SP.URB.TOTL" is Urban Population
# "SP.POP.TOTL" is Population Total
# "SP.POP.0014.IN" is population ages 0-14
# "SP.POP.1564.IN" is Population ages 15-64
# "SP.POP.65UP.IN" is Population ages 65 and above
# "SP.POP.80UP" is Population ages 80 and above
# "SP.DYN.AMRT" is mortality rate
```

Prediction variables were chosen for linear modeling based on their correlation matrix score. We created a correlation matrix with `cor()`. We found that `DaysSinceStart` had the greatest influence in vaccines and so we included that in every model. To pick other variables we just looked at the next most influential variables in our correlation matrix and created combinations.

```
#First linear regression model of vaccination rate vs. number of days since
COVID-19 began.
m1 <- lm(formula = vacRate~daysSinceStart, data = merged_table)

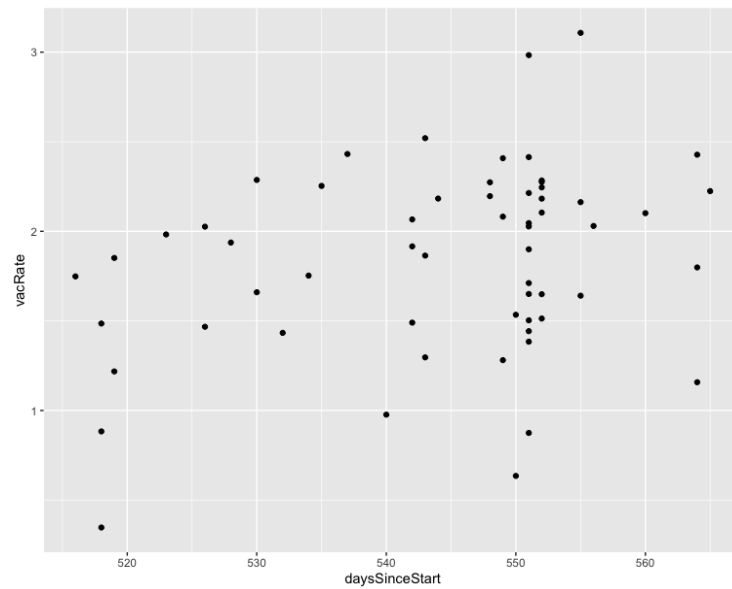
#Second linear regression model of vaccination rate vs. number of days since
COVID-19 began and birth rate.
m2 <- lm(formula = vacRate~daysSinceStart+SP.DYN.LE00.IN, data = merged_table)

#Third linear regression model of vaccination rate vs. number of days since
COVID-19 began and birth rate and GDP.
m3 <- lm(formula = vacRate~daysSinceStart+SP.DYN.LE00.IN+GDP, data =
merged_table)

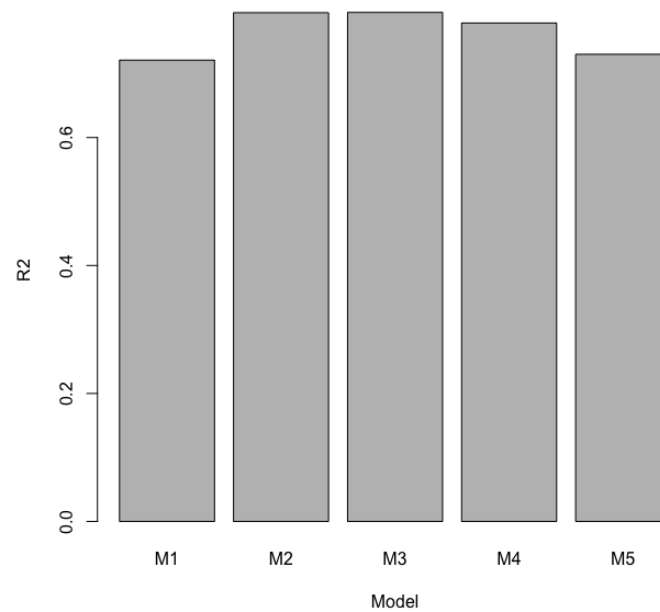
#Fourth linear regression model of vaccination rate vs. number of days since
COVID-19 began and mortality rate.
m4 <- lm(formula = vacRate~daysSinceStart+SP.DYN.AMRT, data = merged_table)

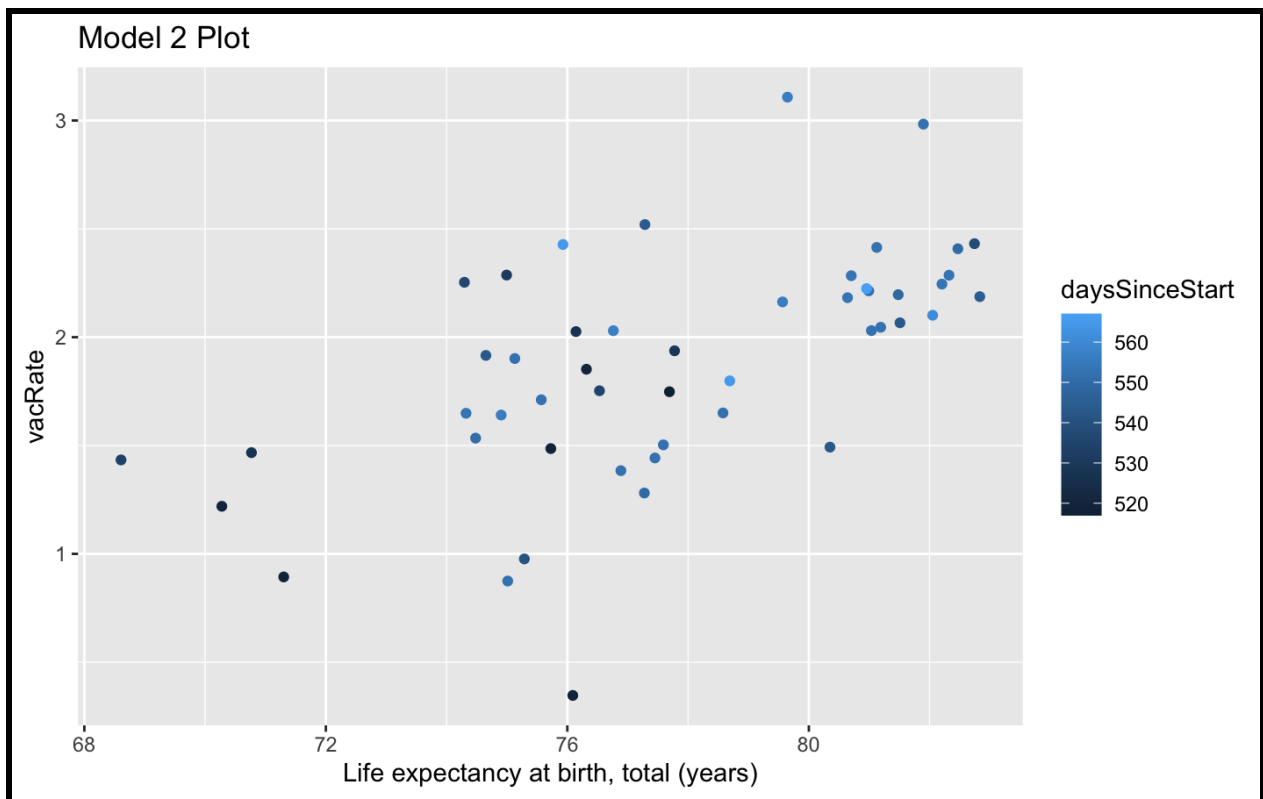
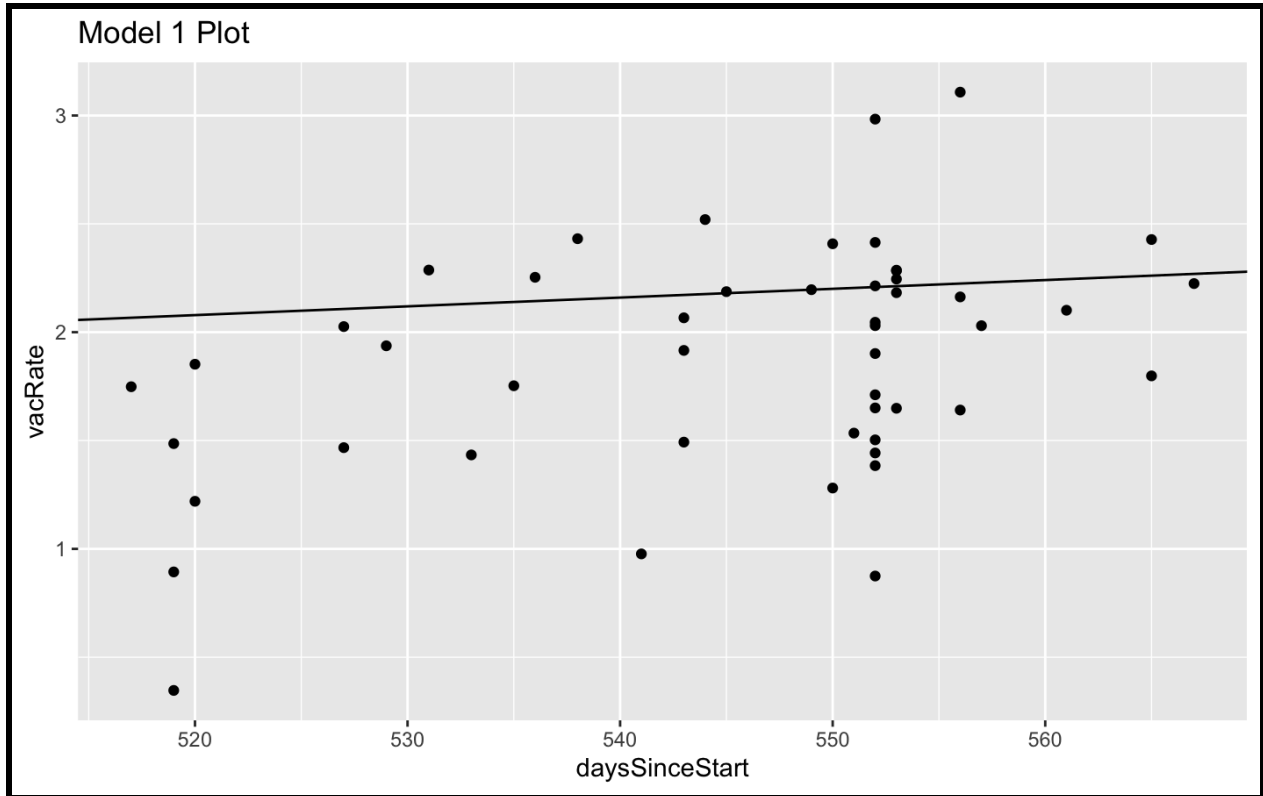
#Fifth linear regression model of vaccination rate vs. number of days since
COVID-19 began and urban population.
m5 <- lm(formula = vacRate~daysSinceStart+SP.URB.TOTL, data = merged_table)
```

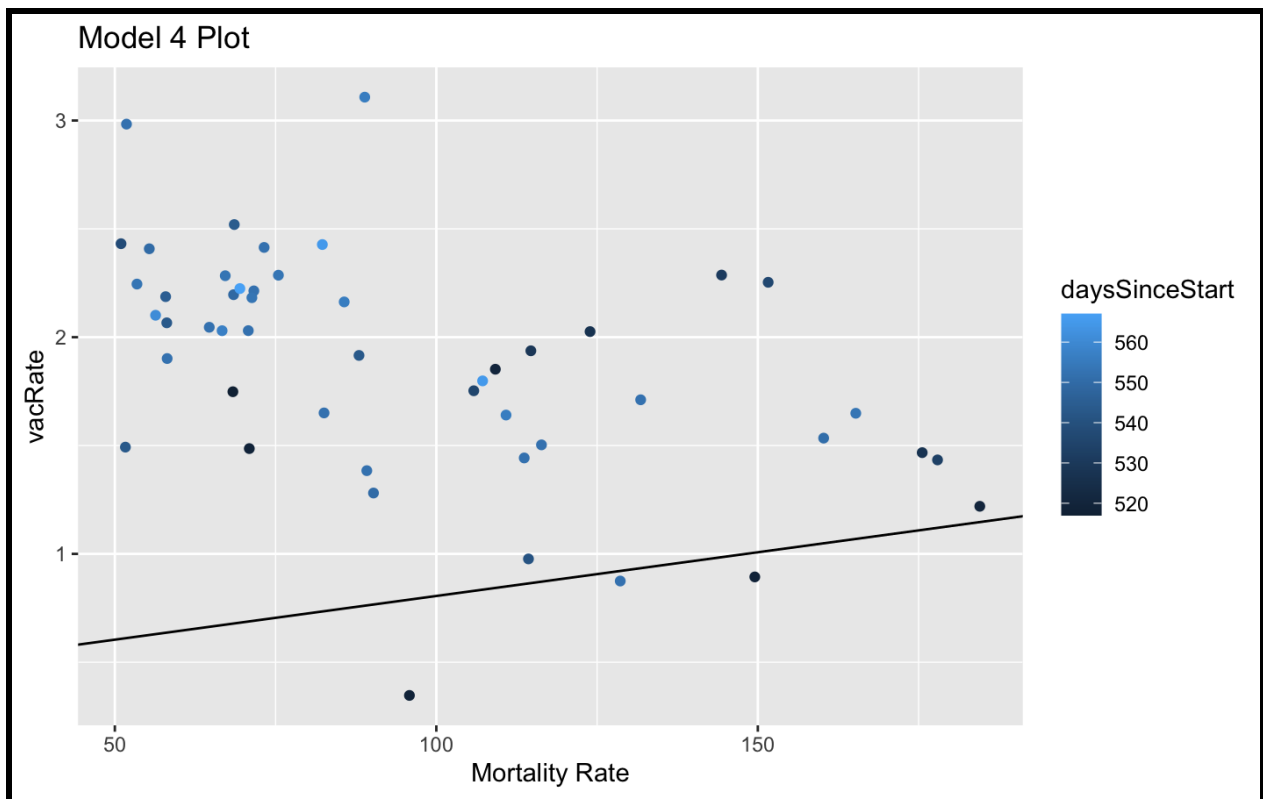
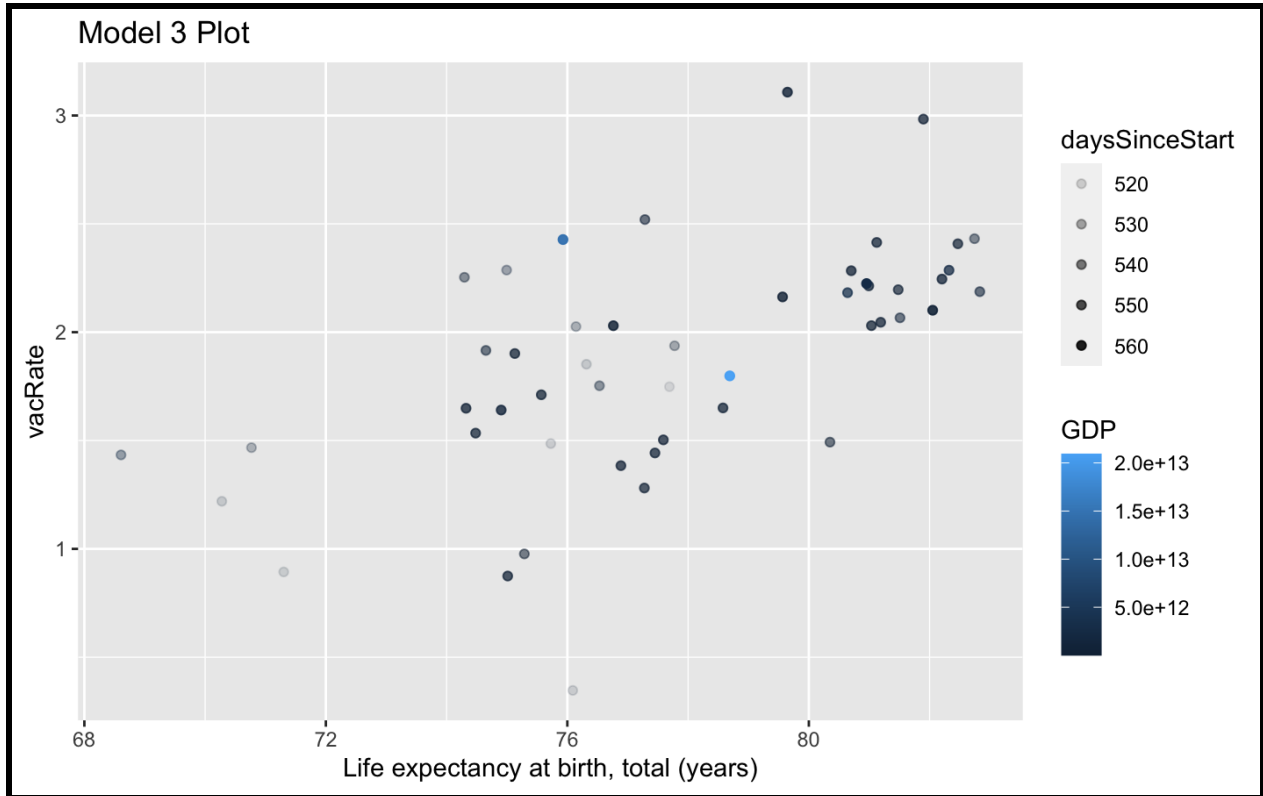

Below is a scatterplot of only the most recent vaccination rate for every country and the number of days since first vaccination:

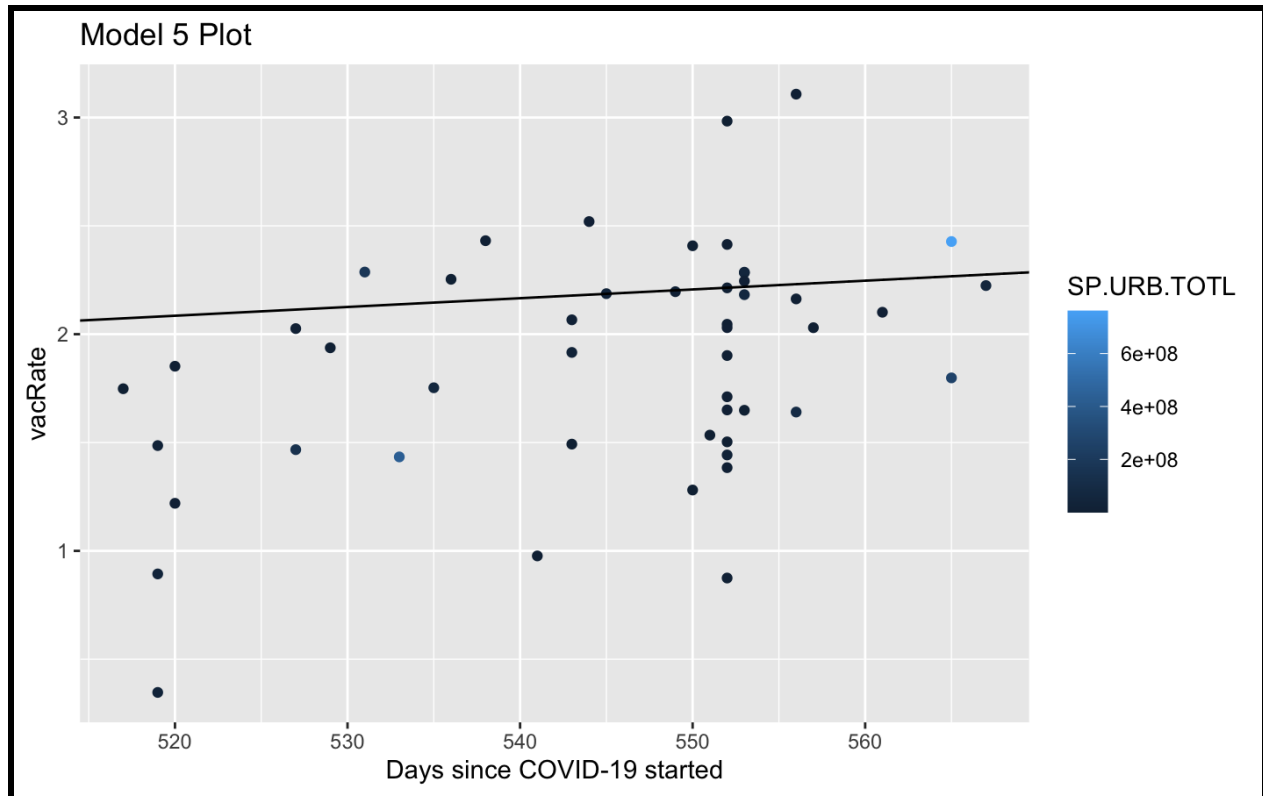


Below is a barplot of the five linear models created in this report and their corresponding R-squared value.









Linear Regression Models Graphs:

```
# MODEL PLOTS
```

```
#M1 Plot
```

```
ggplot(merged_table_max_daysSinceStart) +  
  geom_point(mapping = aes(x = daysSinceStart, y = vacRate )) +  
  geom_abline(slope = m1_coef[2], intercept = m1_coef[1] ) +  
  labs(title = "Model 1 Plot")
```

```
# M2 Plot
```

```
ggplot(merged_table_max_daysSinceStart) +  
  geom_point(mapping = aes(x = SP.DYN.LE00.IN, y = vacRate, color =  
daysSinceStart )) +  
  geom_abline(slope = m2_coef[2], intercept = m2_coef[1] ) +  
  labs(title = "Model 2 Plot", x = "Life expectancy at birth, total  
(years)")
```

```
# M3 Plot
```

```
ggplot(merged_table_max_daysSinceStart) +
```

```

    geom_point(mapping = aes(x = SP.DYN.LE00.IN, y = vacRate, color = GDP,
alpha = daysSinceStart)) +
    geom_abline(slope = m3_coef[2], intercept = m3_coef[1] ) +
    labs(title = "Model 3 Plot", x = "Life expectancy at birth, total
(years)")

# M4 Plot
ggplot(merged_table_max_daysSinceStart) +
    geom_point(mapping = aes(x = SP.DYN.AMRT, y = vacRate, color =
daysSinceStart )) +
    geom_abline(slope = m4_coef[2], intercept = m4_coef[1] ) +
    labs(title = "Model 4 Plot", x = "Mortality Rate")

# M5 Plot
ggplot(merged_table_max_daysSinceStart) +
    geom_point(mapping = aes(x = daysSinceStart, y = vacRate, color =
SP.URB.TOTL )) +
    geom_abline(slope = m5_coef[2], intercept = m5_coef[1] ) +
    labs(title = "Model 5 Plot", x = "Days since COVID-19 started")

```

Conclusion

Based on the barplot, it can be observed that the model with highest R-squared value is model M3 at 0.7956, which is the linear regression model of vaccination rate vs. number of days since COVID-19 began and birth rate and GDP. This indicates to us that these 3 independent variables are among the most significant factors that help to predict vaccination rate across different countries. More specifically, the number of days since COVID-19 began stands out as the most influential variable across all of our models. We also found birth rate and GDP to be more significant across our models when compared with mortality rate and urban population.