COVID-19 Vaccination Analysis Project

Moses Merugu

Project Overview

This project aims to analyze COVID-19 vaccination data to understand vaccination trends and coverage. The objective is to provide insights into the progress of the global vaccination campaign.

Library Descriptions

```
# Set working directory and load necessary libraries
setwd("C:/Users/mmeru/Downloads/vaccination-analysis-project-main")
library(ggplot2) # Used for creating visualizations
library(formatR) # Assists in formatting R code
library(tidyverse) # Collection of R packages for data manipulation
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Data Loading and Preprocessing

```
# Loading COVID-19 vaccination data
vax <- read_csv("https://raw.githubusercontent.com/govex/COVID-19/master/data_tables/vaccine_data/global
```

```
## Rows: 216 Columns: 813
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (5): iso2, iso3, Province_State, Country_Region, Combined_Key
## dbl (806): UID, code3, Lat, Long_, Population, 2020-12-29, 2020-12-30, 2020-...
## lgl   (2): FIPS, Admin2
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Loading COVID-19 hospital bed data
beds <- read_csv("hospitalbed.csv")
```

```
## Rows: 1770 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Country
## dbl (2): Year, Hospital beds (per 10 000 population)
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Loading COVID-19 demographics data
demo <- read_csv("demographics.csv")
```

```
## Rows: 3885 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Country Name, Country Code, Series Name, Series Code
## dbl (1): YR2015
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Clean vaccine data

```r
# VACCINE DATA Filter out provinces and countries with no
# population
vax <- vax %>%
    filter(is.na(Province_State), !is.na(Population)) %>%
    view()
# Tidy number of vaccinations on given date
vax <- vax %>%
    pivot_longer(-c(1:12), names_to = "Date", values_to = "Vaccinations",
        values_drop_na = TRUE) %>%
    view()
# Delete irrelevant columns
vax <- vax[, -c(1, 2, 3, 4, 5, 6, 7, 9, 10, 11)] %>%
    view()
# Filter out rows containing dates with 0 vaccinations
vax <- vax %>%
    filter(!Vaccinations == 0) %>%
    view()
# Calculate vaccination rate and add respective column
vax <- vax %>%
    select(Country_Region, Population, Vaccinations) %>%
    group_by(Country_Region) %>%
    mutate(Vaccination_Rate = Vaccinations/Population) %>%
    view()
# Add column that tracks days since first vaccination
```

```r
vax <- vax %>%
    group_by(Country_Region) %>%
    mutate(Days_Since_First_Vaccination = 1:n()) %>%
    view()
```

Clean hospital bed data

```r
# BEDS DATA Most recent year appears first, keep the first
# bed value per country using summarize() Year column is
# not needed
beds <- beds %>%
    group_by(Country) %>%
    summarize(Beds = first(`Hospital beds (per 10 000 population)`)) %>%
    view()
```

Clean demographics data

```r
# DEMOGRAPHICS DATA Tidy data
demo <- demo %>%
    pivot_wider(id_cols = -"Series Name", names_from = "Series Code",
        values_from = YR2015) %>%
    view()
# Add male and female data together
demo <- demo %>%
    mutate(SP.POP.0014.IN = SP.POP.0014.MA.IN + SP.POP.0014.FE.IN) %>%
    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.DYN.AMRT = SP.DYN.AMRT.MA + SP.DYN.AMRT.FE) %>%
    mutate(SP.POP.TOTL.IN = SP.POP.TOTL.FE.IN + SP.POP.TOTL.MA.IN) %>%
    mutate(SP.POP.65UP.IN = SP.POP.65UP.FE.IN + SP.POP.65UP.MA.IN) %>%
    view()
# Drop country code and gender specific columns, filter NAs
demo <- demo[, -c(2, 6:17)] %>%
    filter(!is.na(SP.DYN.LE00.IN), !is.na(SP.URB.TOTL), !is.na(SP.POP.0014.IN),
        !is.na(SP.POP.80UP), !is.na(SP.POP.1564.IN), !is.na(SP.DYN.AMRT),
        !is.na(SP.POP.TOTL.IN), !is.na(SP.POP.65UP.IN)) %>%
    view()
```

Format data

```r
# UNIFORMING COUNTRY NAMES TO MATCH VACCINE DATA
beds <- beds %>%
    mutate(Country = replace(Country, Country == "Iran (Islamic Republic of)",
        "Iran"))
beds <- beds %>%
```

```
    mutate(Country = replace(Country, Country == "Republic of Korea",
        "South Korea"))
beds <- beds %>%
    mutate(Country = replace(Country, Country == "United Kingdom of Great Britain and Northern Ireland"
        "United Kingdom"))
beds <- beds %>%
    mutate(Country = replace(Country, Country == "Bolivia (Plurinational State of)",
        "Bolivia"))
beds <- beds %>%
    mutate(Country = replace(Country, Country == "Lao People's Democratic Republic",
        "Laos"))
beds <- beds %>%
    mutate(Country = replace(Country, Country == "Venezuela (Bolivarian Republic of)",
        "Venezuela"))
beds <- beds %>%
    mutate(Country = replace(Country, Country == "Republic of Moldova",
        "Moldova"))
beds <- beds %>%
    mutate(Country = replace(Country, Country == "United States of America",
        "US"))
beds <- beds %>%
    mutate(Country = replace(Country, Country == "Viet Nam",
        "Vietnam"))

demo <- demo %>%
    mutate(`Country Name` = replace(`Country Name`, `Country Name` ==
        "Korea, Rep.", "South Korea"))
demo <- demo %>%
    mutate(`Country Name` = replace(`Country Name`, `Country Name` ==
        "Iran, Islamic Rep.", "Iran"))
demo <- demo %>%
    mutate(`Country Name` = replace(`Country Name`, `Country Name` ==
        "Venezuela, RB", "Venezuela"))
demo <- demo %>%
    mutate(`Country Name` = replace(`Country Name`, `Country Name` ==
        "St. Vincent and the Grenadines", "Saint Vincent and the Grenadines"))
demo <- demo %>%
    mutate(`Country Name` = replace(`Country Name`, `Country Name` ==
        "St. Lucia", "Saint Lucia"))
demo <- demo %>%
    mutate(`Country Name` = replace(`Country Name`, `Country Name` ==
        "Slovak Republic", "Slovakia"))
demo <- demo %>%
    mutate(`Country Name` = replace(`Country Name`, `Country Name` ==
        "Czech Republic", "Czechia"))
demo <- demo %>%
    mutate(`Country Name` = replace(`Country Name`, `Country Name` ==
        "Bahamas, The", "Bahamas"))
demo <- demo %>%
    mutate(`Country Name` = replace(`Country Name`, `Country Name` ==
        "United States", "US"))
```

Merging tables

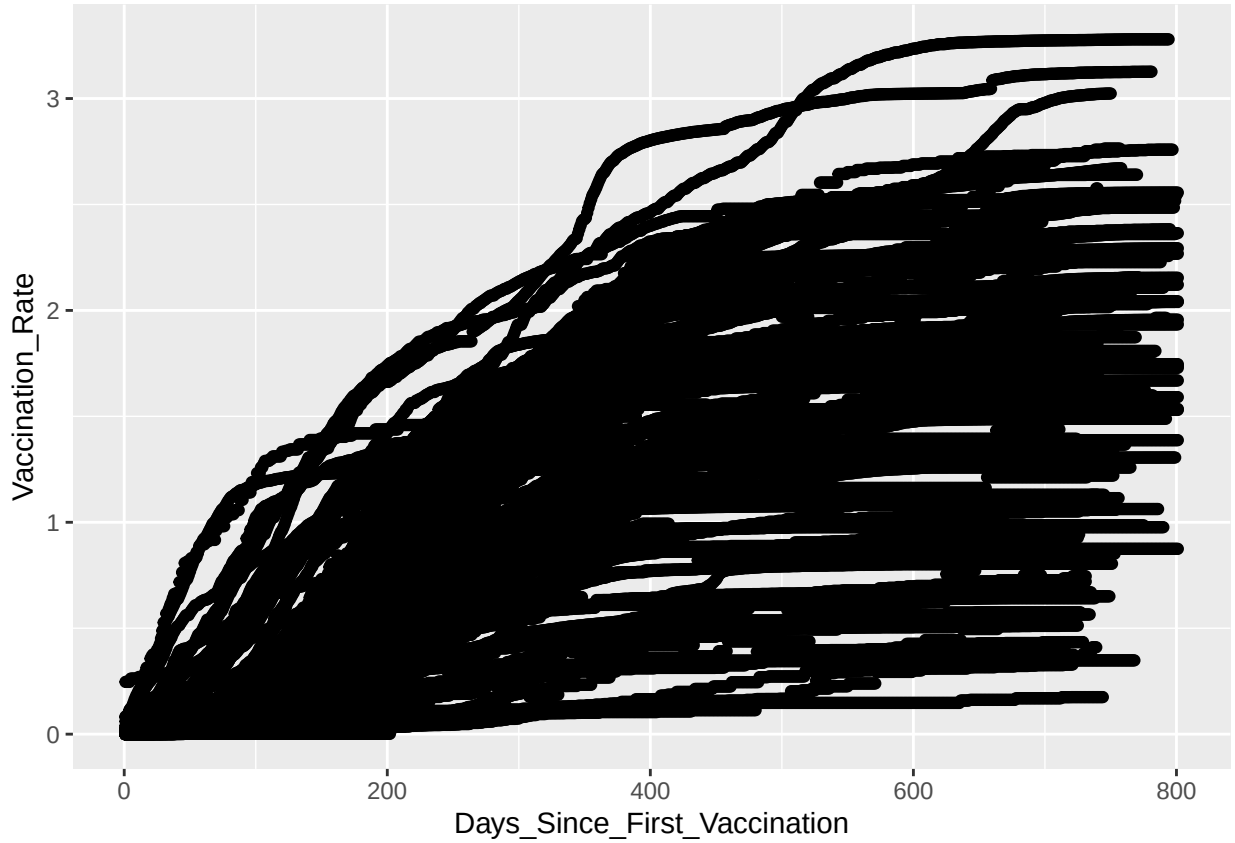
```
# Perform inner joins to merge tables
join <- beds %>%
    inner_join(vax, by = c(Country = "Country_Region")) %>%
    inner_join(demo, by = c(Country = "Country Name")) %>%
    view()

# Rearrange column order to match example
final <- join[, c(1, 5, 4, 3, 6, 2, 7, 8)]

view(final)
```

Plots and Linear Models

```
# PLOTS AND LINEAR MODELS Scatterplot of only the most
# recent vaccination rate for every country and the number
# of days since first vaccination
forplot <- final %>%
    relocate(Vaccinations, .after = Country) %>%
    relocate(Vaccination_Rate, .after = Country)
scatter <- ggplot(data = forplot) + geom_point(mapping = aes(x = Days_Since_First_Vaccination,
    y = Vaccination_Rate))
scatter
```




```
m1 <- lm(data = final, Vaccination_Rate ~ Days_Since_First_Vaccination)
summary(m1)  # R-squared: 0.6125
```

```
##
## Call:
## lm(formula = Vaccination_Rate ~ Days_Since_First_Vaccination,
##     data = final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.78313 -0.33088 -0.06805  0.42618  1.68120
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.561e-01  3.975e-03   39.29   <2e-16 ***
## Days_Since_First_Vaccination 2.422e-03  8.979e-06  269.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5769 on 84069 degrees of freedom
## Multiple R-squared:  0.4639, Adjusted R-squared:  0.4639
## F-statistic: 7.275e+04 on 1 and 84069 DF,  p-value: < 2.2e-16
```

```
m2 <- lm(data = final, Vaccination_Rate ~ Days_Since_First_Vaccination +
    Beds)
```

```r
summary(m2)  # R-squared: 0.6341
```

```
##
## Call:
## lm(formula = Vaccination_Rate ~ Days_Since_First_Vaccination +
##     Beds, data = final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63113 -0.35257 -0.07258  0.40599  1.70683
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -3.536e-03  4.587e-03  -0.771    0.441
## Days_Since_First_Vaccination  2.410e-03  8.763e-06 275.010   <2e-16 ***
## Beds                          5.550e-03  8.519e-05  65.151   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5628 on 84068 degrees of freedom
## Multiple R-squared:  0.4897, Adjusted R-squared:  0.4897
## F-statistic: 4.033e+04 on 2 and 84068 DF,  p-value: < 2.2e-16
```

```r
m3 <- lm(data = final, Vaccination_Rate ~ Days_Since_First_Vaccination +
    SP.DYN.LE00.IN)
summary(m3)  # R-squared: 0.7473
```

```
##
## Call:
## lm(formula = Vaccination_Rate ~ Days_Since_First_Vaccination +
##     SP.DYN.LE00.IN, data = final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.75335 -0.31013 -0.01133  0.30898  1.37159
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -3.634e+00  1.776e-02  -204.6   <2e-16 ***
## Days_Since_First_Vaccination  2.352e-03  7.197e-06   326.9   <2e-16 ***
## SP.DYN.LE00.IN                5.163e-02  2.381e-04   216.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4619 on 84068 degrees of freedom
## Multiple R-squared:  0.6563, Adjusted R-squared:  0.6562
## F-statistic: 8.025e+04 on 2 and 84068 DF,  p-value: < 2.2e-16
```

```r
m4 <- lm(data = final, Vaccination_Rate ~ Days_Since_First_Vaccination +
    SP.URB.TOTL)
summary(m4)  # R-squared: 0.6131
```

```
##
## Call:
## lm(formula = Vaccination_Rate ~ Days_Since_First_Vaccination +
##     SP.URB.TOTL, data = final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.76576 -0.33691 -0.06999  0.41830  1.70170
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.366e-01  4.003e-03   34.13   <2e-16 ***
## Days_Since_First_Vaccination 2.419e-03  8.929e-06  270.87   <2e-16 ***
## SP.URB.TOTL                  6.805e-10  2.205e-11   30.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5736 on 84068 degrees of freedom
## Multiple R-squared:  0.4699, Adjusted R-squared:  0.4699
## F-statistic: 3.726e+04 on 2 and 84068 DF,  p-value: < 2.2e-16
```

```
m5 <- lm(data = final, Vaccination_Rate ~ Days_Since_First_Vaccination +
    SP.URB.TOTL + SP.DYN.LE00.IN)
summary(m5)  # R-squared: 0.7478
```

```
##
## Call:
## lm(formula = Vaccination_Rate ~ Days_Since_First_Vaccination +
##     SP.URB.TOTL + SP.DYN.LE00.IN, data = final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.75053 -0.30623 -0.01273  0.30456  1.38502
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -3.644e+00  1.763e-02 -206.73   <2e-16 ***
## Days_Since_First_Vaccination  2.350e-03  7.143e-06  328.94   <2e-16 ***
## SP.URB.TOTL                   6.340e-10  1.762e-11   35.98   <2e-16 ***
## SP.DYN.LE00.IN                5.153e-02  2.363e-04  218.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4584 on 84067 degrees of freedom
## Multiple R-squared:  0.6615, Adjusted R-squared:  0.6615
## F-statistic: 5.475e+04 on 3 and 84067 DF,  p-value: < 2.2e-16
```

```
# Organize 5 models and corresponding R2 values into data
# frame
df <- data.frame(Model = c("M1", "M2", "M3", "M4", "M5"), R2 = c(summary(m1)$r.squared,
    summary(m2)$adj.r.squared, summary(m3)$adj.r.squared, summary(m4)$adj.r.squared,
    summary(m5)$adj.r.squared))
```

```r
# Create bar plot comparing models and their R2 values
bar <- ggplot(data = df, aes(x = Model, y = R2)) + geom_bar(stat = "identity")
bar
```