Analytical Notebook - Change in Covid Cases by Countries

Anthony Benyarko

2024-04-13

1. Introduction

This report presents an analysis of COVID-19 data using Spark and R. We have performed various data manipulation and modeling task to gain insights into two data sets that give us information about COVID-19 and information about various countries.

2. Setup

In order to perform the various data manipulation and modeling task in Spark and R, we deployed various packages and libraries to complete each task.

```
#Install packages and load libraries
# Set the CRAN mirror
options(repos = c(CRAN = "https://cloud.r-project.org"))

install.packages("xfun")
```

```
## Installing package into 'C:/Users/abeny/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'xfun' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'xfun'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\abeny\AppData\Local\R\win-library\4.3\00LOCK\xfun\libs\x64\xfun.dll to
## C:\Users\abeny\AppData\Local\R\win-library\4.3\xfun\libs\x64\xfun.dll:
## Permission denied
```

```
## Warning: restored 'xfun'
```

```
##
## The downloaded binary packages are in
##  C:\Users\abeny\AppData\Local\Temp\RtmpK2uKNi\downloaded_packages
```

```r
install.packages("packrat")
```

```
## Installing package into 'C:/Users/abeny/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'packrat' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\abeny\AppData\Local\Temp\RtmpK2uKNi\downloaded_packages
```

```r
install.packages("sparklyr")
```

```
## Installing package into 'C:/Users/abeny/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'sparklyr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\abeny\AppData\Local\Temp\RtmpK2uKNi\downloaded_packages
```

```r
install.packages("sparklyr.nested")
```

```
## Installing package into 'C:/Users/abeny/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'sparklyr.nested' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\abeny\AppData\Local\Temp\RtmpK2uKNi\downloaded_packages
```

```r
install.packages("tidyverse")
```

```
## Installing package into 'C:/Users/abeny/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\abeny\AppData\Local\Temp\RtmpK2uKNi\downloaded_packages
```

```r
install.packages("rmarkdown")
```

```
## Installing package into 'C:/Users/abeny/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'rmarkdown' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\abeny\AppData\Local\Temp\RtmpK2uKNi\downloaded_packages
```

```r
install.packages("stargazer")
```

```
## Installing package into 'C:/Users/abeny/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'stargazer' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\abeny\AppData\Local\Temp\RtmpK2uKNi\downloaded_packages
```

```r
install.packages("corrr")
```

```
## Installing package into 'C:/Users/abeny/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'corrr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\abeny\AppData\Local\Temp\RtmpK2uKNi\downloaded_packages
```

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3

## Warning: package 'ggplot2' was built under R version 4.3.3

## Warning: package 'tidyr' was built under R version 4.3.2

## Warning: package 'readr' was built under R version 4.3.2

## Warning: package 'dplyr' was built under R version 4.3.2

## Warning: package 'stringr' was built under R version 4.3.2

## Warning: package 'forcats' was built under R version 4.3.3

## -- 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 ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2

## -- 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
```

```r
library(dplyr)
library(ggplot2)
library(sparklyr)
```

```
## Warning: package 'sparklyr' was built under R version 4.3.3
```

```
##
## Attaching package: 'sparklyr'
##
## The following object is masked from 'package:purrr':
##
##     invoke
##
## The following object is masked from 'package:stats':
##
##     filter
```

```r
library(DBI)
```

```
## Warning: package 'DBI' was built under R version 4.3.3
```

```r
library(tidyr)
library(forcats)
library(ggmosaic)
```

```
## Warning: package 'ggmosaic' was built under R version 4.3.3
```

```r
library(broom)
library(texreg)
```

```
## Warning: package 'texreg' was built under R version 4.3.3
```

```
## Version:  1.39.3
## Date:     2023-11-09
## Author:   Philip Leifeld (University of Essex)
##
## Consider submitting praise using the praise or praise_interactive functions.
## Please cite the JSS article in your publications -- see citation("texreg").
##
## Attaching package: 'texreg'
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

3. Data Preparation

We were tasked with downloading two data sets about COVID-19 from a GitHub Repository.

3.1. *Load Data*

We loaded the data into R using the read.csv function.

```
#Load CSV File into a Data set in R
covid19_confirmed_globa_data <- read.csv("Data/Raw Data/time_series_covid19_confirmed_global.csv")

#Load CSV File into a Data set in R
UID_ISO_FIPS_LookUp_Table_data <- read.csv("Data/Raw Data/UID_ISO_FIPS_LookUp_Table.csv")
```

3.2. *Data Cleaning*

Preparing our data for analysis

We created a long version of our COVID-19 data set.

```
#Creating a long version of the data set;Define the time variable as a date
long_data_covid19_confirmed_data <- covid19_confirmed_globa_data %>%
  pivot_longer(cols = starts_with("X"),
               names_to = "Date",
               values_to = "Confirmed_Cases") %>%
  mutate(Date = as.Date(gsub("X", "", Date), format = "%m.%d.%Y"))

#New Variable that shows Days since start of Data Collection
long_data_covid19_confirmed_data <- long_data_covid19_confirmed_data %>%
  mutate(Days_Since_Data_Collection_Started = as.numeric(Date - min(Date)))

#Save long data set as .CSV in the Clean Data folder
write.csv(long_data_covid19_confirmed_data, file = "Data/Clean Data/Long_data_Covid19.csv")

#Save edited version of UID_ISO_FIPS_LookUp_Table_data
write.csv(UID_ISO_FIPS_LookUp_Table_data, file = "Data/Clean Data/Country_Lookup_Table.csv" )
```

After this step, we continued our data manipulation in Spark There is code you will need to adjust to run in Spark environment

We merged our two data sets.

```
#Load CSV (use spark_read_csv() to run in Spark)
Confirmed_Cases <- read.csv("Data/Clean Data/Long_data_Covid19.csv")

Country_Table <- read.csv("Data/Clean Data/Country_Lookup_Table.csv")

#Combining both data frames
Country_Confirmed_Cases <- Confirmed_Cases %>%
  inner_join(Country_Table, by = c("Country.Region" = "Country_Region"))


#Making smaller Data frame that contains the information Germany,
#China, United Kingdom, US, Brazil and Mexico

#The countries I want to look at in the new data frame
selected_countries <- c("Germany", "China", "Japan", "United Kingdom", "US", "Brazil", "Mexico")
```

```r
Specific_Country_Confirmed_Cases <- Country_Confirmed_Cases %>%
  filter(Country.Region %in% selected_countries) %>%
  mutate(
    Population = as.numeric(Population),
    rate_of_cases = Confirmed_Cases / Population)

#Remove Columns that are not needed
Clean_Specific_Country_Confirmed_Cases <- Specific_Country_Confirmed_Cases %>%
  select(-iso2, -iso3, -FIPS, -Admin2, -Long_, -code3, -UID, -Combined_Key)

# Calculate the number of cases by country and day
cases_by_country_day <- Clean_Specific_Country_Confirmed_Cases %>%
  group_by(Country.Region, Date) %>%
  mutate(total_cases = max(Confirmed_Cases)) %>%
  ungroup() %>%
  distinct(Country.Region, Date, total_cases)

# Convert the Spark DataFrame to a local R DataFrame
local_cases_by_country_day <- collect(cases_by_country_day)

# Convert Country_Region and Date to factors
local_cases_by_country_day$Country.Region <- as.factor(local_cases_by_country_day$Country.Region)
local_cases_by_country_day$Date <- as.Date(local_cases_by_country_day$Date)

#Calculate rate of cases by country and date

# Group by Country_Region and Date, and calculate the rate of cases
rate_by_country_day <- Clean_Specific_Country_Confirmed_Cases %>%
  group_by(Country.Region, Date) %>%
  mutate(rate_of_cases = sum(Confirmed_Cases) / max(Population, na.rm = TRUE)) %>%
  ungroup() %>%
  distinct(Country.Region, Date, rate_of_cases)

# Convert the Spark DataFrame to a local R DataFrame
local_rate_by_country_day <- collect(rate_by_country_day)

# Convert Country.Region and Date to factors
local_rate_by_country_day$Country.Region <- as.factor(local_rate_by_country_day$Country.Region)
local_rate_by_country_day$Date <- as.Date(local_rate_by_country_day$Date)
```

We took that merge data set and prepared the data for visualization.

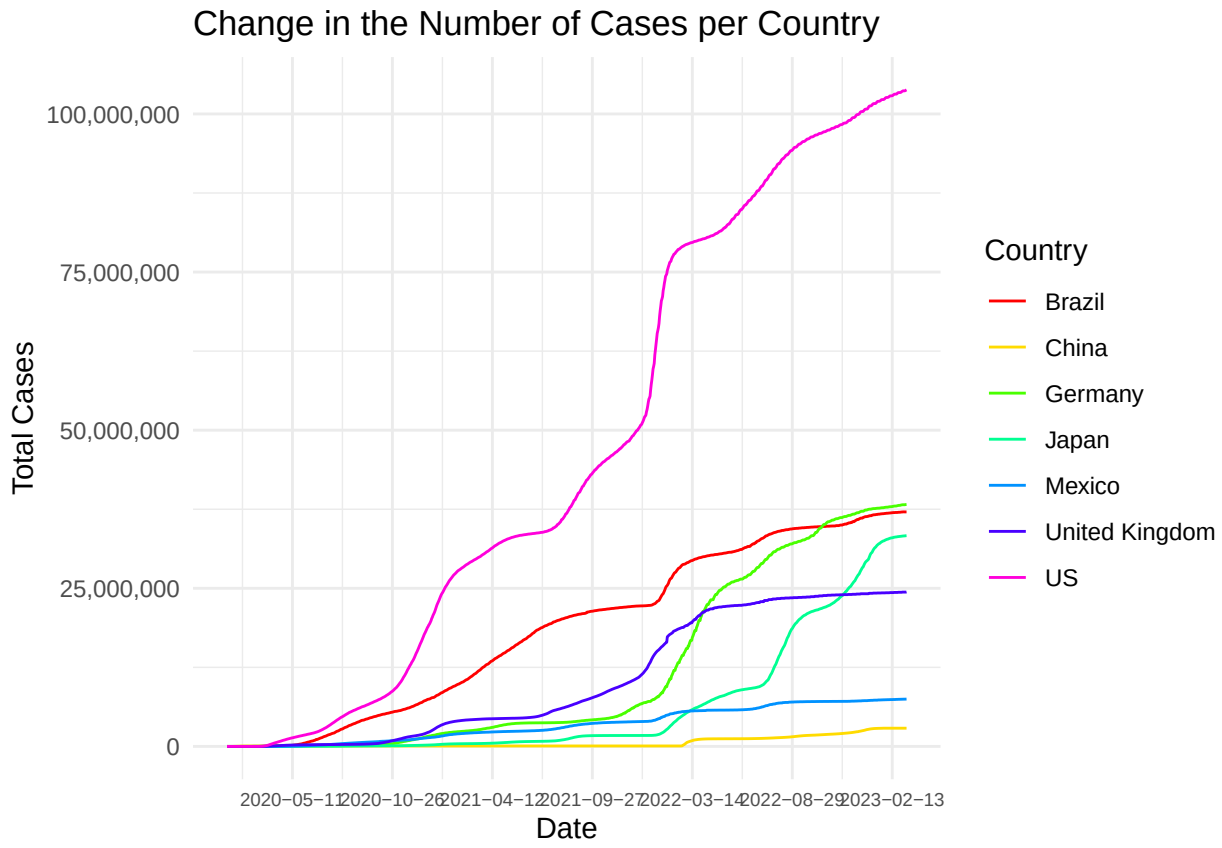
4. Exploratory Data Analysis

4.1 *Visualization*

4.1.1 *Change in the number of COVID-19 cases per country*

```r
# Create a custom color palette for the countries
country_colors <- rainbow(length(unique(local_cases_by_country_day$Country.Region)))
```

```r
ggplot(local_cases_by_country_day, aes(x = Date, y = total_cases, color = Country.Region)) +
  geom_line() +
  scale_x_date(date_labels = "%Y-%m-%d", date_breaks = "24 week") +
  labs(title = "Change in the Number of Cases per Country",
       x = "Date",
       y = "Total Cases",
       color = "Country") +
  scale_color_manual(values = country_colors) +  # Assign custom colors
  theme_minimal() +
  scale_y_continuous(labels = scales::comma_format()) +
  theme(axis.text.x = element_text(size = 7))
```

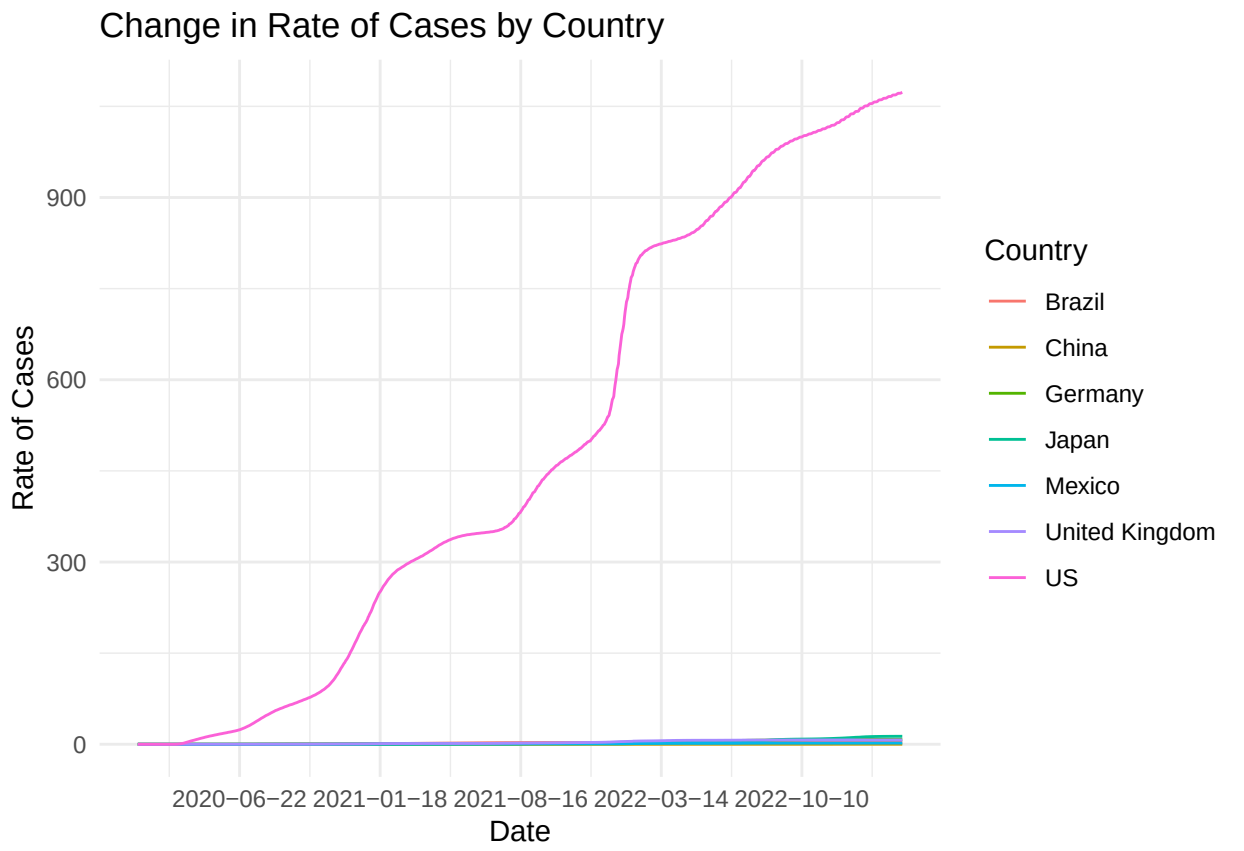


This visualization depicts the evolving number of cases across the selected countries. Notably, the United States exhibited the most significant surge in cases during the specified time frame, while China experienced the most modest increase over the same period.

4.1.2 *Change in rate of COVID-19 cases by country*

```r
ggplot(local_rate_by_country_day, aes(x = Date, y = rate_of_cases, color = Country.Region)) +
  geom_line() +
  labs(title = "Change in Rate of Cases by Country",
       x = "Date",
       y = "Rate of Cases",
```

```
      color = "Country") +
  theme_minimal() +
  scale_x_date(date_labels = "%Y-%m-%d", date_breaks = "30 week") +
  scale_color_discrete(name = "Country")
```



This graph illustrates the fluctuation in the rate of cases across different countries. Notably, the United States demonstrated the most substantial escalation in its rate of cases during this time frame.

5. Modeling

5.1 *Linear Regression (R Code)*

Run a linear regression explaining the log number of cases using country, population, and day since the start of the pandemic

```
#Run a linear regression explaining the log number of cases using country,

#population, and day since the start of the pandemic

# Log transform the Confirmed_Cases column
log_cases_by_country_day <- Clean_Specific_Country_Confirmed_Cases %>%
  group_by(Country.Region, Date) %>%
  mutate(log_Confirmed_Cases = log(Confirmed_Cases + 1)) %>%
  ungroup() %>%
```

```r
  mutate(Confirmed_Cases = ifelse(is.na(Confirmed_Cases), 0, Confirmed_Cases),
         log_Confirmed_Cases = ifelse(log_Confirmed_Cases == 0, NA, log_Confirmed_Cases)) %>%
  distinct(Country.Region, log_Confirmed_Cases, Population, Days_Since_Data_Collection_Started)

# Remove rows with NA values in the log_Confirmed_Cases column
log_cases_by_country_day <- log_cases_by_country_day %>%
  filter(!is.na(log_Confirmed_Cases))

# Remove all rows with NA values in any column
log_cases_by_country_day <- na.omit(log_cases_by_country_day)

#Model function in R
#Model using lm() function in R
model_forlogcasesR <- lm(log_Confirmed_Cases ~ Country.Region + Population + Days_Since_Data_Collection_

print(model_forlogcasesR)
```

```
##
## Call:
## lm(formula = log_Confirmed_Cases ~ Country.Region + Population +
##     Days_Since_Data_Collection_Started, data = log_cases_by_country_day)
##
## Coefficients:
##                           (Intercept)                  Country.RegionChina
##                             1.243e+01                            -8.959e+00
##                Country.RegionGermany                  Country.RegionJapan
##                            -1.125e+00                            -2.394e+00
##                 Country.RegionMexico         Country.RegionUnited Kingdom
##                            -1.673e+00                            -8.729e+00
##                     Country.RegionUS                           Population
##                             6.616e-01                             1.201e-21
## Days_Since_Data_Collection_Started
##                             5.906e-03
```

5.1.1 ***Linear Regression (Spark / R Code)*** Run a linear regression explaining the log number of cases using country, population, and day since the start of the pandemic.

```r
#Run a linear regression explaining the log number of cases using country, population, and day since th

# Log transform the Confirmed_Cases column
#log_cases_by_country_day <- Clean_Specific_Country_Confirmed_Cases %>%
  #group_by(Country.Region, Date) %>%
  #mutate(log_Confirmed_Cases = log(Confirmed_Cases + 1)) %>%
  #ungroup() %>%
  #mutate(Confirmed_Cases = ifelse(is.na(Confirmed_Cases), 0, Confirmed_Cases),
         #log_Confirmed_Cases = ifelse(log_Confirmed_Cases == 0, NA, log_Confirmed_Cases)) %>%
  #distinct(Country.Region, log_Confirmed_Cases, Population, Days_Since_Data_Collection_Started)

# Remove rows with NA values in the log_Confirmed_Cases column
#log_cases_by_country_day <- log_cases_by_country_day %>%
  #filter(!is.na(log_Confirmed_Cases))
```

```
# Remove all rows with NA values in any column
#log_cases_by_country_day <- na.omit(log_cases_by_country_day)

#Summary Data
#summary_data <- log_cases_by_country_day %>%
  #group_by(Country.Region) %>%
  #summarize(
    #Total_Log_Confirmed_Cases = sum(log_Confirmed_Cases, na.rm = TRUE),
    #Total_Population = sum(Population, na.rm = TRUE),
    #Days_Since_Data_Collection_Started = max(Days_Since_Data_Collection_Started, na.rm = TRUE)
 # )



#local_summary_data <- collect(summary_data)

#local_summary_data$Country.Region <- as.factor(local_summary_data$Country.Region)

# Convert Total_Population to numeric
#local_summary_data$Total_Population <- as.numeric(local_summary_data$Total_Population)

# Convert Days_Since_Data_Collection_Started to numeric
#local_summary_data$Days_Since_Data_Collection_Started <- as.numeric(local_summary_data$Days_Since_Data_

#Create .CSV for data frame for Spark Regression Table
#write.csv(local_summary_data, file = "Data/Clean Data/local_sum_data.csv")

# Run the linear regression model
#Regression_data <- spark_read_csv(sc, "Data/Clean Data/local_sum_data.csv")

#selected_Regression_data <- Regression_data %>%
  #select(Country.Region, Total_Log_Confirmed_Cases, Total_Population, Days_Since_Data_Collection_Start_

# Perform linear regression
#model_forlogcases <- ml_linear_regression(
  #selected_Regression_data,
  #formula = Total_Log_Confirmed_Cases ~ Country_Region + Total_Population + Days_Since_Data_Collection_
#)
```

5.2 *Extract Model Statistics*

Extract tidy summary of coefficients, and extract model performance metrics

```
# Extract tidy summary of coefficients
tidy_summary <- tidy(model_forlogcasesR)

# Extract model performance metrics
glance_summary <- glance(model_forlogcasesR)

# View the results
print(tidy_summary)
```

```
## # A tibble: 9 x 5
```

```
##   term                                estimate std.error statistic p.value
##   <chr>                                  <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)                         1.24e+ 1  1.36e- 2  9.12e+ 2    0
## 2 Country.RegionChina                -8.96e+ 0  1.37e- 2 -6.55e+ 2    0
## 3 Country.RegionGermany              -1.12e+ 0  2.18e- 2 -5.17e+ 1    0
## 4 Country.RegionJapan                -2.39e+ 0  1.69e- 2 -1.42e+ 2    0
## 5 Country.RegionMexico               -1.67e+ 0  1.84e- 2 -9.11e+ 1    0
## 6 Country.RegionUnited Kingdom       -8.73e+ 0  1.43e- 2 -6.12e+ 2    0
## 7 Country.RegionUS                    6.62e- 1  1.36e- 2  4.88e+ 1    0
## 8 Population                          1.20e-21  8.90e-12  1.35e-10    1.00
## 9 Days_Since_Data_Collection_Started  5.91e- 3  3.11e- 6  1.90e+ 3    0
```

```
print(glance_summary)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df     logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>      <dbl>  <dbl>  <dbl>
## 1     0.798         0.798  2.38  2665372.       0     8 -12352926. 2.47e7 2.47e7
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

5.3 *Model Summary*

```
# Create a table using screenreg
screenreg(model_forlogcasesR)
```

```
##
## ================================================
##                                      Model 1
## ------------------------------------------------
## (Intercept)                            12.43 ***
##                                        (0.01)
## Country.RegionChina                    -8.96 ***
##                                        (0.01)
## Country.RegionGermany                  -1.12 ***
##                                        (0.02)
## Country.RegionJapan                    -2.39 ***
##                                        (0.02)
## Country.RegionMexico                   -1.67 ***
##                                        (0.02)
## Country.RegionUnited Kingdom           -8.73 ***
##                                        (0.01)
## Country.RegionUS                        0.66 ***
##                                        (0.01)
## Population                              0.00
##                                        (0.00)
## Days_Since_Data_Collection_Started      0.01 ***
##                                        (0.00)
## ------------------------------------------------
## R^2                                     0.80
## Adj. R^2                                0.80
## Num. obs.                         5406274
```

```
## ================================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

China, Germany, Japan, Mexico, the United Kingdom, and the United States, they all show big differences in the number of confirmed cases compared to other countries.

China exhibits a considerably lower log confirmed cases compared to the reference, with a coefficient of approximately -8.96, indicating a logarithmic decrease. Conversely, the United States displays a positive coefficient of around 0.66, signifying a logarithmic increase in confirmed cases compared to the reference.

When we consider population size, it doesn't seem to make much of a difference in predicting the number of confirmed cases.

As time goes by and we collect more data, it seems like the number of confirmed cases tends to go up.

6. Conclusion / Session Info

6.1.1 *Conclusion*

The United States exhibited the most significant surge in cases during the specified time frame, while China experienced the most modest increase over the same period. The United States demonstrated the most substantial escalation in its rate of cases during this time frame.

China exhibits a considerably lower log confirmed cases compared to the reference, with a coefficient of approximately -8.96, indicating a logarithmic decrease. Conversely, the United States displays a positive coefficient of around 0.66, signifying a logarithmic increase in confirmed cases compared to the reference. Population size is not a good predictor of the number of confirmed cases of COVID-19. As time goes by and we collect more data, it seems like the number of confirmed cases tends to go up.

6.1.2 *Session Info from Local R

```
# R version 4.3.1 (2023-06-16 ucrt)
#Platform: x86_64-w64-mingw32/x64 (64-bit) Running under: Windows 11 x64 (build 22631)
#Matrix products: default
#locale:
#[1] LC_COLLATE=English_United States.utf8  [2] LC_CTYPE=English_United States.utf8
#[3] LC_MONETARY=English_United States.utf8 [4] LC_NUMERIC=C                           [5] LC_TIME=Engl
#time zone: America/New_York tzcode
#source: internal  attached base
#packages: [1] stats     graphics  grDevices utils     datasets  methods   [7] base
#other attached packages:  [1] texreg_1.39.3   broom_1.0.5     ggmosaic_0.3.3  DBI_1.2.2        [5] spa
#loaded via a namespace (and not attached):
#[1] plotly_4.10.4     utf8_1.2.4        generics_0.1.3     [4] stringi_1.8.3     hms_1.1.3         dig
#[13] ggrepel_0.9.5     backports_1.4.1   httr_1.4.7         [16] fansi_1.0.6       viridisLite_0.4.2 sc
#[25] withr_3.0.0       tools_4.3.1       tzdb_0.4.0         [28] colorspace_2.1-0  vctrs_0.6.5       R6
#[37] data.table_1.15.0 glue_1.7.0        Rcpp_1.0.12        [40] xfun_0.43         tidyselect_1.2.1  rs
```

6.1.3 *Session Info from Instance on AWS*

```
#R version 4.3.2 (2023-10-31)
#Platform: x86_64-pc-linux-gnu (64-bit)
#Running under: Ubuntu 22.04.4 LTS

#Matrix products: default
#BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.20.so;  LAPACK version 3.10.0

#locale:
 #[1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8        LC_COLLATE=en_US
 #[5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
#LC_PAPER=en_US.UTF-8       LC_NAME=C
 #[9] LC_ADDRESS=C               LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATIO

#time zone: Etc/UTC
#tzcode source: system (glibc)

#attached base packages:
#[1] stats     graphics  grDevices utils     datasets  methods   base

#other attached packages:
 #[1] shiny_1.8.0     sparklyr_1.8.4  lubridate_1.9.3 forcats_1.0.0
#stringr_1.5.1   dplyr_1.1.4     purrr_1.0.2
 #[8] readr_2.1.5     tidyr_1.3.1     tibble_3.2.1    ggplot2_3.5.0   tidyverse_2.0.0

#loaded via a namespace (and not attached):
 #[1] utf8_1.2.4        generics_0.1.3    stringi_1.8.3     hms_1.1.3         digest_0.6.34     magritt
 #[7] evaluate_0.23     grid_4.3.2        timechange_0.3.0  fastmap_1.1.1     jsonlite_1.8.8    DBI_1.2
#[13] promises_1.2.1    httr_1.4.7        fansi_1.0.6       scales_1.3.0      cli_3.6.2         rlang_1
#[19] dbplyr_2.4.0      ellipsis_0.3.2    munsell_0.5.0     yaml_2.3.8        withr_3.0.0       tools_4
#[25] tzdb_0.4.0        colorspace_2.1-0  httpuv_1.6.14     vctrs_0.6.5       R6_2.5.1          mime_0.
#[31] lifecycle_1.0.4   pkgconfig_2.0.3   pillar_1.9.0      later_1.3.2       gtable_0.3.4      glue_1.
#[37] Rcpp_1.0.12       xfun_0.42         tidyselect_1.2.0  knitr_1.45        rstudioapi_0.15.0 xtable_
#[43] htmltools_0.5.7   rmarkdown_2.25    compiler_4.3.2    askpass_1.2.0     openssl_2.1.1
```