COVID-19 Analysis Using Spark

Sungjoo Cho

2024-04-12

1. Libraries

```
# library
library(tidyr)
library(lubridate)
library(tidyverse)
library(broom)
library(texreg)
library(knitr)
library(ggplot2)
library(dplyr)
library(haven)
```

2. Set up a local Spark server

A local Spark server can be set up by importing the `sparklyr` library.The code below will check the installed version and available Spark versions.

```
#spark_install(version = "3.5.1")
library(sparklyr)
```

```
##
## Attaching package: ’sparklyr’

## The following object is masked from ’package:purrr’:
##
##     invoke

## The following object is masked from ’package:stats’:
##
##     filter
```

```
# check Java version
system("java -version")

# check sparklyr version
packageVersion("sparklyr")
```

```
## [1] '1.8.5'
```

```
# check available Spark versions
spark_installed_versions()
```

```
##   spark hadoop                                                dir
## 1 2.3.4    2.7 /Users/sungjoocho/spark/spark-2.3.4-bin-hadoop2.7
## 2 3.5.1      3   /Users/sungjoocho/spark/spark-3.5.1-bin-hadoop3
```

3. Adding two datasets about COVID-19 and Data cleaning

Two datasets about COVID-19 were obtained from the GitHub repository.

```
# get two data sets from github
count_city_github_url <-
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/UID_ISO_FIPS_Loo
count_city <- read.csv(count_city_github_url)

timeseries_github_url <-
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_ti
timeseries <- read.csv(timeseries_github_url)
```

The variables 'Province.State', 'Country.Region', 'Lat', 'Long' in the `count_city` dataframe were dropped as they were duplicates in the `timeseries` dataframe. Then, the data was transformed into long format with the number of Covid cases. Also, a new variable named `days` was added, representing the number of days since the start of the data collection.

```
# drop columns that are in timeseries
count_city <- select(count_city, -Province_State, -Country_Region, -Lat, -Long_)

# change timeseries data to longer format
timeseries_long <- timeseries %>%
  pivot_longer(
    cols = !c(Province.State, Country.Region, Lat, Long),
    names_to = "time",
    values_to = "case"
  )

# change time to month-day-year format
timeseries_long$date <- gsub("^X", "", timeseries_long$time)
timeseries_long$date <- mdy(timeseries_long$date)

# create another variable (number of days since the start of the data collection)
start_date <- min(timeseries_long$date)
timeseries_long <- timeseries_long %>%
  mutate(days = as.numeric(date - start_date))

# create combined key
timeseries_long$Combined_Key <-
  ifelse(is.na(timeseries_long$Province.State) | timeseries_long$Province.State == "",
         timeseries_long$Country.Region,
         paste(timeseries_long$Province.State, timeseries_long$Country.Region, sep = ", "))
```

These are the first few rows of the two datasets before they are merged in Spark.

```
head(count_city)
```

```
##   UID iso2 iso3 code3 FIPS Admin2 Combined_Key Population
## 1   4   AF  AFG     4   NA          Afghanistan   38928341
## 2   8   AL  ALB     8   NA              Albania    2877800
## 3  10   AQ  ATA    10   NA           Antarctica         NA
## 4  12   DZ  DZA    12   NA              Algeria   43851043
## 5  20   AD  AND    20   NA              Andorra      77265
## 6  24   AO  AGO    24   NA               Angola   32866268
```

```
head(timeseries_long)
```

```
## # A tibble: 6 x 9
##   Province.State Country.Region   Lat  Long time      case date        days
##   <chr>          <chr>          <dbl> <dbl> <chr>    <int> <date>     <dbl>
## 1 ""             Afghanistan     33.9  67.7 X1.22.20     0 2020-01-22     0
## 2 ""             Afghanistan     33.9  67.7 X1.23.20     0 2020-01-23     1
## 3 ""             Afghanistan     33.9  67.7 X1.24.20     0 2020-01-24     2
## 4 ""             Afghanistan     33.9  67.7 X1.25.20     0 2020-01-25     3
## 5 ""             Afghanistan     33.9  67.7 X1.26.20     0 2020-01-26     4
## 6 ""             Afghanistan     33.9  67.7 X1.27.20     0 2020-01-27     5
## # i 1 more variable: Combined_Key <chr>
```

4. Merging two datasets in Spark

In Spark, two datasets were merged with a subset that includes only: Germany, China, Japan, United Kingdom, US, Brazil, and Mexico. To connect to the local cluster, `spark_connect()` was used.

```
# set up a local Spark connection
sc <- spark_connect(master = "local")

# copying datasets into Spark
city <- copy_to(sc, count_city, overwrite = TRUE)
time <- copy_to(sc, timeseries_long, overwrite = TRUE)

# merging data
covid_full <- time %>%
  left_join(city, by = "Combined_Key")

# selected countries
sel_countries <- c("Germany", "China", "Japan", "United Kingdom", "US", "Brazil", "Mexico")
covid <- covid_full %>%
  filter(Country_Region %in% sel_countries)
```

These are the first few rows of the merged dataset in Spark, containing only seven countries.

```
head(covid)
```

```
## # Source:   SQL [6 x 16]
## # Database: spark_connection
##   Province_State Country_Region   Lat  Long time      case date        days
##   <chr>          <chr>          <dbl> <dbl> <chr>    <int> <date>     <dbl>
## 1 ""             Brazil         -14.2 -51.9 X1.22.20     0 2020-01-22     0
## 2 ""             Brazil         -14.2 -51.9 X1.23.20     0 2020-01-23     1
## 3 ""             Brazil         -14.2 -51.9 X1.24.20     0 2020-01-24     2
## 4 ""             Brazil         -14.2 -51.9 X1.25.20     0 2020-01-25     3
## 5 ""             Brazil         -14.2 -51.9 X1.26.20     0 2020-01-26     4
## 6 ""             Brazil         -14.2 -51.9 X1.27.20     0 2020-01-27     5
## # i 8 more variables: Combined_Key <chr>, UID <int>, iso2 <chr>, iso3 <chr>,
## #   code3 <int>, FIPS <int>, Admin2 <chr>, Population <int>
```

```r
# save original dataset locally
save(covid, file = "data/covid.csv")
```

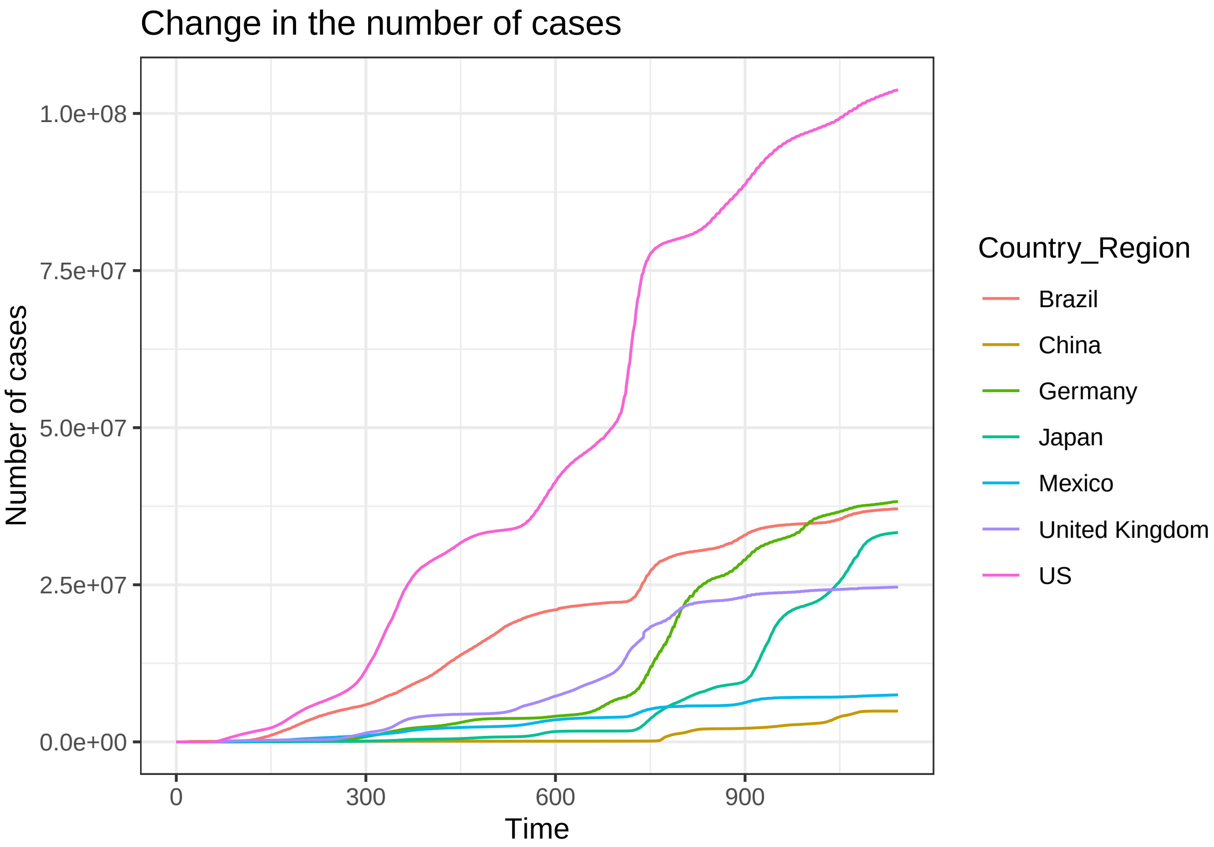
5. Calculating the number of cases and rate of cases (cases/population) by country and day and Creating two graphs and interpreting them: change in the number of cases and change in rate by country.

The summary table and graphs below show the change in the number of cases by country over time. All seven countries exhibit increasing trends in the number of COVID cases over time. Notably, US has experienced particularly rapid increases in the number of cases, while other countries show a more steady trend.

```r
# calculate the number of cases by country and day
tab_change_case <- covid %>%
  group_by(Country_Region, days) %>%
  summarise(sum_case = sum(case, na.rm = TRUE),
            .groups = "drop")
head(tab_change_case)
```

```
## # Source:   SQL [6 x 3]
## # Database: spark_connection
##   Country_Region  days sum_case
##   <chr>          <dbl>    <dbl>
## 1 China              1      643
## 2 United Kingdom     1        0
## 3 Japan              1        2
## 4 Germany            1        0
## 5 China             18    39829
## 6 China             21    44759
```

```r
plot_change_case <- ggplot(data=tab_change_case, aes(days, sum_case, color = Country_Region)) +
  geom_line() +
  theme_bw() +
  labs(x = "Time",
       y = "Number of cases",
       title = "Change in the number of cases")
plot_change_case
```



```
# save
ggsave("figs/plot_change_case.png", plot = plot_change_case)
```

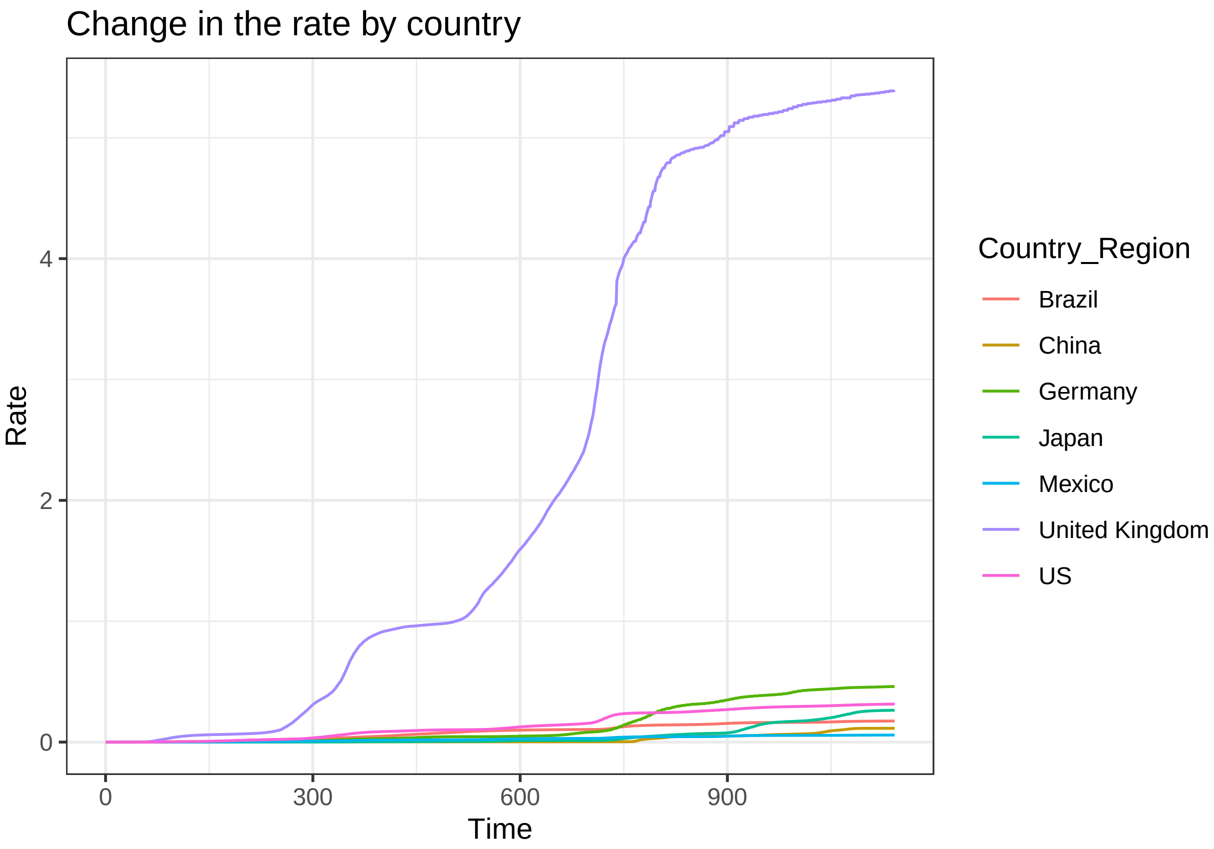
```
## Saving 6.5 x 4.5 in image
```

Furthermore, the summary table and graphs below illustrate the change in the rate of cases by country over time. The rate of cases was calculated as (cases/population). It shows that the rate of cases is significantly and rapidly increasing over time in the United Kingdom, whereas other countries exhibit a more steady trend.

```
# calculate rate of cases (cases/population) by country and day
tab_change_rate <- covid %>%
  group_by(Country_Region, days) %>%
  summarise(sum_case = sum(case, na.rm = TRUE),
            population = mean(Population, na.rm = T),
            .groups = "drop") %>%
  mutate(rate = (sum_case / population))
head(tab_change_rate)
```

```
## # Source:   SQL [6 x 5]
## # Database: spark_connection
##   Country_Region  days sum_case population         rate
##   <chr>          <dbl>    <dbl>      <dbl>        <dbl>
## 1 China              1      643  42967426. 0.0000150
## 2 United Kingdom     1        0   4571579. 0
## 3 Japan              1        2 126476458  0.0000000158
## 4 Germany            1        0  83155031  0
## 5 China             18    39829  42967426. 0.000927
## 6 China             21    44759  42967426. 0.00104
```

```
plot_change_rate <- ggplot(data=tab_change_rate, aes(days, rate, color = Country_Region)) +
  geom_line() +
  theme_bw() +
  labs(x = "Time",
       y = "Rate",
       title = "Change in the rate by country")
plot_change_rate
```



```
# save
ggsave("figs/plot_change_rate.png", plot = plot_change_rate)
```

```
## Saving 6.5 x 4.5 in image
```

6. Fitting a ml_linear_regression explaining the log of number of cases using: country, population size and day since the start of the pandemic. Interpret the results.

Next, a linear model was fitted to approximate the relationship between the log number of cases and three predictors: country, population size, and day since of the pandemic. The `ml_linear_regression()` function was used for this analysis. The table presented below displays the output from the regression model. The United States was used as a reference category in this model.

It indicates that all predictors (country, population, and days) significantly influence the log number of COVID19 cases (p<0.05). Holding all other predictors constant, the log number of cases is higher in all other countries than that of the US. Additionally, the one-unit increase in the number of days results in a increase of 0.0043287 in the log number of cases in the US, when the other predictor is hold constant.

```r
# log case and remove NA in Population variable
covid <- covid %>%
  mutate(log_case = log(case+1)) %>%
  filter(!is.na(Population))

# log number of cases
model <- ml_linear_regression(covid, log_case ~ Country_Region + Population + days)
```

```r
# coefficients
coeff <- tidy(model)
kable(coeff, caption = "Coefficients of regression model")
```

Table 1: Coefficients of regression model

term	estimate	std.error	statistic	p.value
(Intercept)	2.0735418	0.1473887	14.068528	0.0e+00
Country_Region_China	0.6463879	0.1328440	4.865765	1.1e-06
Country_Region_United Kingdom	1.3025076	0.1460547	8.917946	0.0e+00
Country_Region_Brazil	3.1833170	0.1121814	28.376520	0.0e+00
Country_Region_Germany	7.0732186	0.1399199	50.551903	0.0e+00
Country_Region_Japan	4.2865586	0.1291576	33.188583	0.0e+00
Country_Region_Mexico	4.6086747	0.1288506	35.767590	0.0e+00
Population	0.0000000	0.0000000	93.774942	0.0e+00
days	0.0043287	0.0000302	143.157712	0.0e+00

```r
# regression model table
texreg(model, caption = "Output from regression model")
```

	Model 1
(Intercept)	2.07^{***}
	(0.15)
Country_Region_China	0.65^{***}
	(0.13)
Country_Region_United Kingdom	1.30^{***}
	(0.15)
Country_Region_Brazil	3.18^{***}
	(0.11)
Country_Region_Germany	7.07^{***}
	(0.14)
Country_Region_Japan	4.29^{***}
	(0.13)
Country_Region_Mexico	4.61^{***}
	(0.13)
Population	0.00^{***}
	(0.00)
days	0.00^{***}
	(0.00)
explained.variance	8.91
mean.absolute.error	1.75
mean.squared.error	6.03
R^2	0.60
root.mean.squared.error	2.46

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 2: Output from regression model