Analysis

Huy Le Quang

7/15/2020

1.Introduction

Coronavirus (Covid19) brings about a more severe crisis than almost any before. To protect people from the coronavirus, governments worldwide have had to implement safety measures that have a huge impact on personal and economic life. This report, however, does not aim at analyzing the consequences of Covid19, but to give an overview of the changes in the total of cases and infection rates over time (from January 2020 until July 2020). This report will zoom in the situation in some selected countries where coronavirus is more prevalent.

2. Dataset

The dataset is taken from John Hopkins University (the JHU CSSE COVID-19 Dataset) which monitors the coronavirus epidemie situation all over the world and collects data on daily basis. The main data for analysis is the "Time_series_covid19_confirmed_global" showing the total of confirmed infected cases per country from 22 Jan 2020 until now. This dataset is then merged with the "UID_ISO_FIPS_LookUp_Table" to add some additional country characteristics variables (e.g. population, longitude, latitude, country code, states...)

3. Descriptive analysis

The general approach for analysis in this assignment is to first download the two datasets from GitHub, clean and merge these two datasets on Spark server, and then carry out some descriptive analysis (graphs) and regression analysis.

First of all, we need to load necessary packages in R for data cleaning and analysis. There are three packages that we will need in this exercise: tidyverse, sparklyr and lubridate.

```r
# Load necessary packages

library(tidyverse)
```

```
## -- Attaching packages -------------------------------------------------------------------------------
```

```
## v ggplot2 3.3.1     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------ tid
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(sparklyr)
```

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

After loading the necessary packages, we download the two datasets for Coronavirus from Github and store them in two objects: (1) lookup.table and (2) time.series.

```
# Download data

lookup.table.url <- "https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/UID_ISO_

lookup.table <- read.csv(url(lookup.table.url))


time.series.url <- "https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/csse_covi
time.series <- read.csv(url(time.series.url))
```

We proceed with cleaning the time series data by first deleting unnecessary columns, and reshaping the time series data in long format. We also calculate the number of days from the begin of data collection (22 January 2020) until the last day of the dataset.

```
# Clean the time series dataset

time.series <- time.series[-c(3, 4)] # delete unnecessary columns

names(time.series)[names(time.series) == 'Country.Region'] <- 'country'
names(time.series)[names(time.series) == 'Province.State'] <- 'state'

time.series.long <- pivot_longer(data = time.series,
                          -c(state, country),
                          names_to = "date") # reshape to long format

# Change to date format

time.series.long$date <- as.Date(gsub("X", "", time.series.long$date), "%m.%d.%y")

begin.date <- as.Date("2020-01-22")

# Calculate the number of days since the start of data collection

time.series.long <- time.series.long %>%
    mutate(day_difference = date - begin.date)

# Change variable names in lookup dataset

names(lookup.table)[names(lookup.table) == 'Country_Region'] <- 'country'
names(lookup.table)[names(lookup.table) == 'Province_State'] <- 'state'
```

Establish Spark server

To take advantage of the ability for large-scale data processing, we set up Spark server and do further analysis on this server.

First, we create a local Spark server, and then copy both datasets that we cleaned above to this server. Further cleaning tasks are: joining two datasets, and make a smaller version of the dataset which includes only 8 countries, namely: Germany, China, Japan, United Kingdom, US, Brazil, and Mexico. Finally, we create two new variables: (1) log number of confirmed cases, and infection rate per 100,000 people.

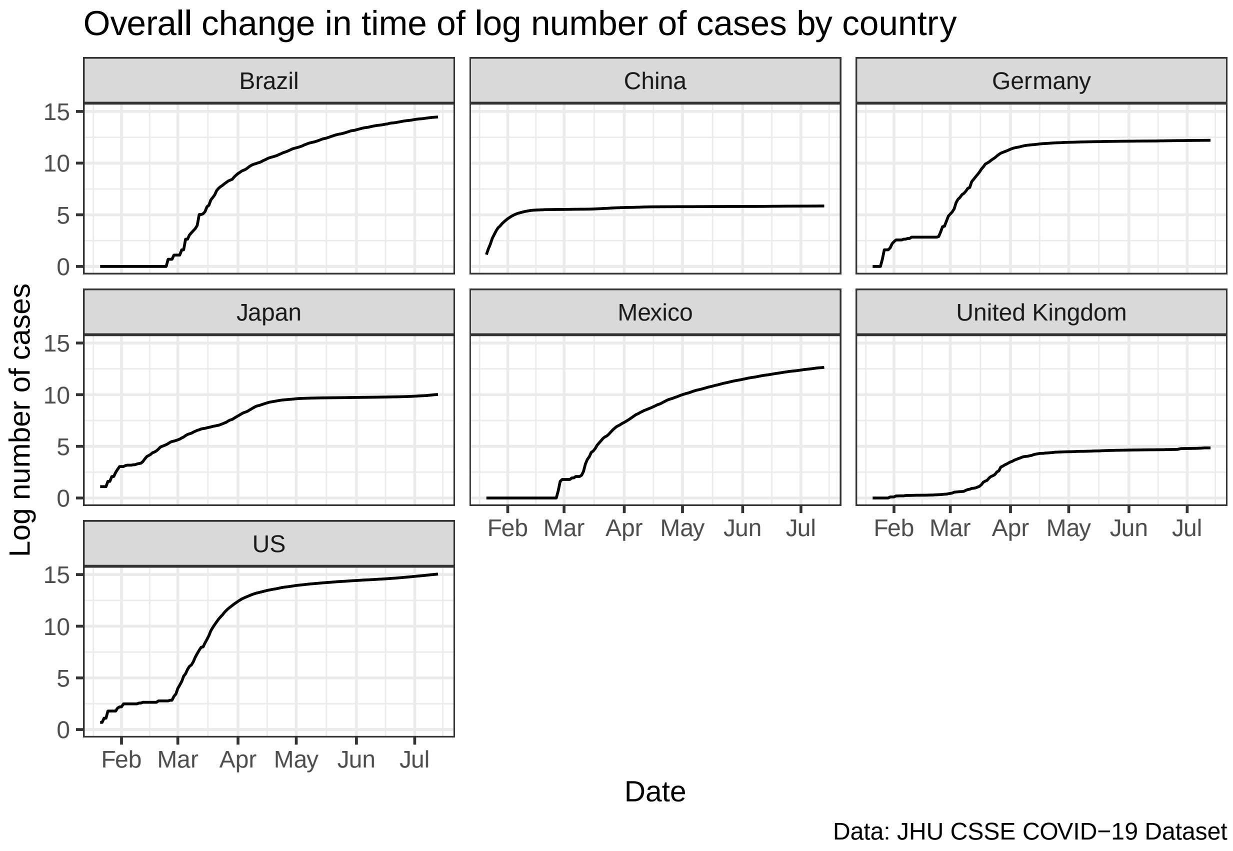
```r
# Connect to local Spark server

sc <- spark_connect(master = "local",
                    version = "2.3")

# Copy two datasets to Spark server

time.series.long <- copy_to(sc, time.series.long, overwrite = T)
lookup.table <- copy_to(sc, lookup.table, overwrite = T)

# Merge two datasets

data_long <- full_join(time.series.long, lookup.table, by = c("state","country"))

# Rename variables

data_long <- data_long %>%
    rename(latitude = Lat,
           longitude = Long_,
           population = Population,
           country_code = iso3,
           confirmed_cases = value)

# Make a smaller dataset and generate two new variables

data_final <- data_long %>%
    filter(country == "Germany"|country == "China"|country == "Japan"|
            country == "United Kingdom"|country == "US"|
            country =="Brazil"|country == "Mexico")%>%
    mutate(log.confirmed_cases = log((confirmed_cases+1)),
           infection_rate = confirmed_cases/population*100000)
```

Draw graphs

After having a cleaned dataset, we start the analysis by drawing two graphs. The first graph shows the overall change in time of log number of confirmed cases of infection in eight selected countries. While the number of cases is leveled off in China, Germany and Japan, it slightly increases in the United Kingdom, and remarkably increases in Brazil, the US and Mexico. These three latter countries will expect an increasing number of infected people in the upcoming days due to this increasing trend.

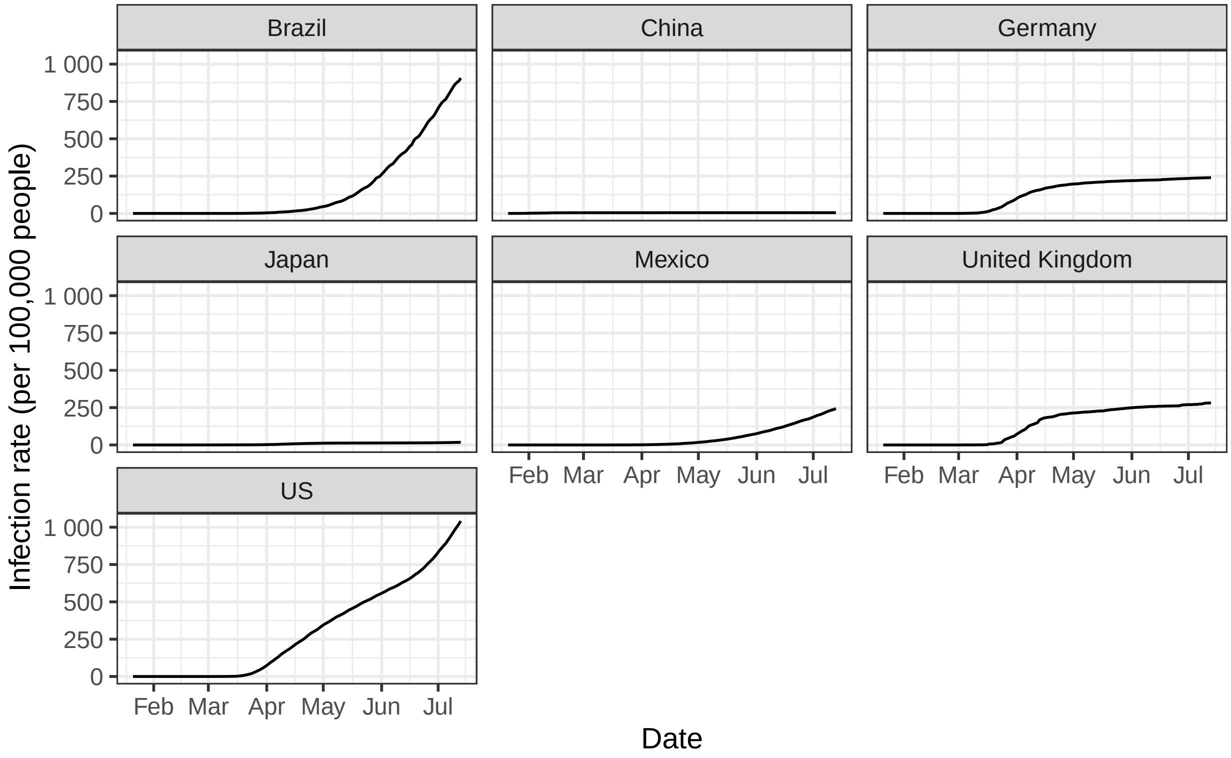
The second graph shows the infection rate for each of eight selected countries. The infection rate is calculated by dividing the total of confirmed cases by the total population. To be easy to read, the infection rate is presented per 100,000 people. The country with highest infection rate is the US with more than 1,000 patients per 100,000 people, followed by Brazil with nearly 900 cases per 100,000 inhabitants. Noticeably, this rate is increasing sharply in both countries. Two countries with lowest infection rate are China and Japan. Germany and the United Kingdom have around 250 cases per 100,000 people, however, the infection rate seems to be leveled off. Mexico has the infection rate similar to Germany and United Kingdom, but this rate is on the increasing trend.

```
## Overall change in time of log number of cases

ggplot(aes(x = date, y = log.confirmed_cases),
       data = data_final) +
    stat_summary(fun.y = mean,
                 geom = "line") +
    theme_bw() +
    facet_wrap(~country) +
    labs(x = "Date", y = "Log number of cases",
         title = "Overall change in time of log number of cases by country",
         caption = "Data: JHU CSSE COVID-19 Dataset")
```



```
# Infection rate

ggplot(aes(x = date, y = infection_rate),
       data = data_final) +
    stat_summary(fun.y = mean,
                 geom = "line") +
    facet_wrap(~country) +
    scale_y_continuous(labels = scales::number_format(accuracy = 1)) +
    theme_bw() +
    labs(x = "Date", y = "Infection rate (per 100,000 people)",
         title = "Change in time of infection rate by country",
         caption = "Data: JHU CSSE COVID-19 Dataset. Infection rate per 100,000 people")
```

Change in time of infection rate by country



Data: JHU CSSE COVID−19 Dataset. Infection rate per 100,000 people

Regression model

Finally, we run the regression model to explain the number of confirmed cases by three independent variables, namely, country fixed effects, population and the number of days since the begin of data collection.

The base country for comparison is Brazil. Compared to Brazil, only US has lower confirmed cases, *ceteris paribus*. Both population and the number of days since the begin of data collection are positively correlated with the number of infected people, other things being equal. For example, each day since 22 Jan 2020, there are 0.02155 confirmed cases increased, compared to the previous day, *ceteris paribus*.

```
data_final %>% lm(formula = log.confirmed_cases ~ country + population + day_difference) %>%
summary()
```

```
##
## Call:
## lm(formula = log.confirmed_cases ~ country + population + day_difference,
##     data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9384 -1.2633 -0.0869  1.2320  6.1247
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -2.860e+00  2.474e-01  -11.56   <2e-16 ***
## countryChina              4.577e+00  2.162e-01   21.17   <2e-16 ***
## countryGermany            6.549e+00  2.455e-01   26.68   <2e-16 ***
## countryJapan           3.088e+00  2.300e-01   13.43   <2e-16 ***
## countryMexico          2.540e+00  2.296e-01   11.06   <2e-16 ***
## countryUnited Kingdom  3.714e+00  2.445e-01   15.19   <2e-16 ***
## countryUS             -3.040e+00  2.407e-01  -12.63   <2e-16 ***
## population             4.404e-08  8.958e-10   49.16   <2e-16 ***
## day_difference         2.155e-02  4.424e-04   48.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.027 on 8216 degrees of freedom
##   (3817 observations deleted due to missingness)
## Multiple R-squared:  0.5482, Adjusted R-squared:  0.5478
## F-statistic:  1246 on 8 and 8216 DF,  p-value: < 2.2e-16
```

4. Conclusion

Overall, both the number of cases as well as the infection rate have increased sharply in North and Latin America since March 2020. Some countries have reacted quickly with strong measures such as Germany, Japan and China have seen a level off in the number of cases.

***Notes*:**

- I already set up the AWS and RStudio Server, but I could not complete the assignment on RStudio Server because it failed to install the sparklyr package, and it constantly asked me to give Username and Password each time I want to commit to GitHub.
- I am not able to run the ml_linear_regression because I always received error message: org.apache.spark.sql.Analysis (*I did not have this problem if I use mtcar database as an example*)