2019 - 2020 Covid-19 outbreak

Linlin Sun

3/10/2020

Background

The 2019-2020 Covid-19 outbreak is an ongoing global outbreak of Covid-19 disease 2019 that has been declared a Public Health Emergency of International Concern. It is caused by the SARS-CoV-2 Covid-19, first identified in Wuhan, Hubei, China. Over 100 countries and territories have been affected at the beginning of March 2020 with major outbreaks in central China, South Korea, Italy, Iran, France, and Germany.

Background of the author

As a newbie in the data science world, I would like to keep up with how covid-19 is spreading everyday.

I am hoping we will get through this soon and wish the best for everyone!

Data files

I have been referring to Johns Hopkins CSSE Covid-19

https://gisanddata.maps.arcgis.com/apps/opsdashboard/index.html#/bda7594740fd40299423467b48e9ecf for current cases around the world.

Here is the link to get the data. github link

I got the world population information from https://www.worldometers.info/world-population/population-by-country/

Exploratory Data Analysis

Including the library.

```
library(tidyverse)
library(gridExtra)
library(lubridate)
library(matrixStats)
library(kableExtra)
options(digits=2)
```

Loading the data.

```r
path <- getwd()
covid_Confirmed_ts <- "data/time_series_covid_19_confirmed_global.csv"
covid_Deaths_ts <- "data/time_series_covid_19_deaths_global.csv"

c_ts <- read.csv(paste(path, covid_Confirmed_ts, sep = "/"), header = TRUE)
c_ts_col_count <- length(colnames(c_ts))
d_ts <- read.csv(paste(path, covid_Deaths_ts, sep = "/"), header = TRUE)
d_ts_col_count <- length(colnames(d_ts))
```

Set some constant variables.

```r
# This decide how many countries will be displayed.
# For the plot, the default colors do not all work well. I handpicked 15 colors
# which I think works better.
# If you want to run this using more than 15 countries, the code will go back to
# using the default coloring by R.
select_top <- 15

day1_count <- 100

manual_colors <- c("aquamarine2", "brown", "blue", "orange", "chartreuse",
            "darkgoldenrod1", "cyan", "darkgreen", "grey39", "darkseagreen",
            "yellow", "deepskyblue4", "darkorchid", "pink2", "red1")
```

First, I get the confirmed and deaths data ready and combine them.

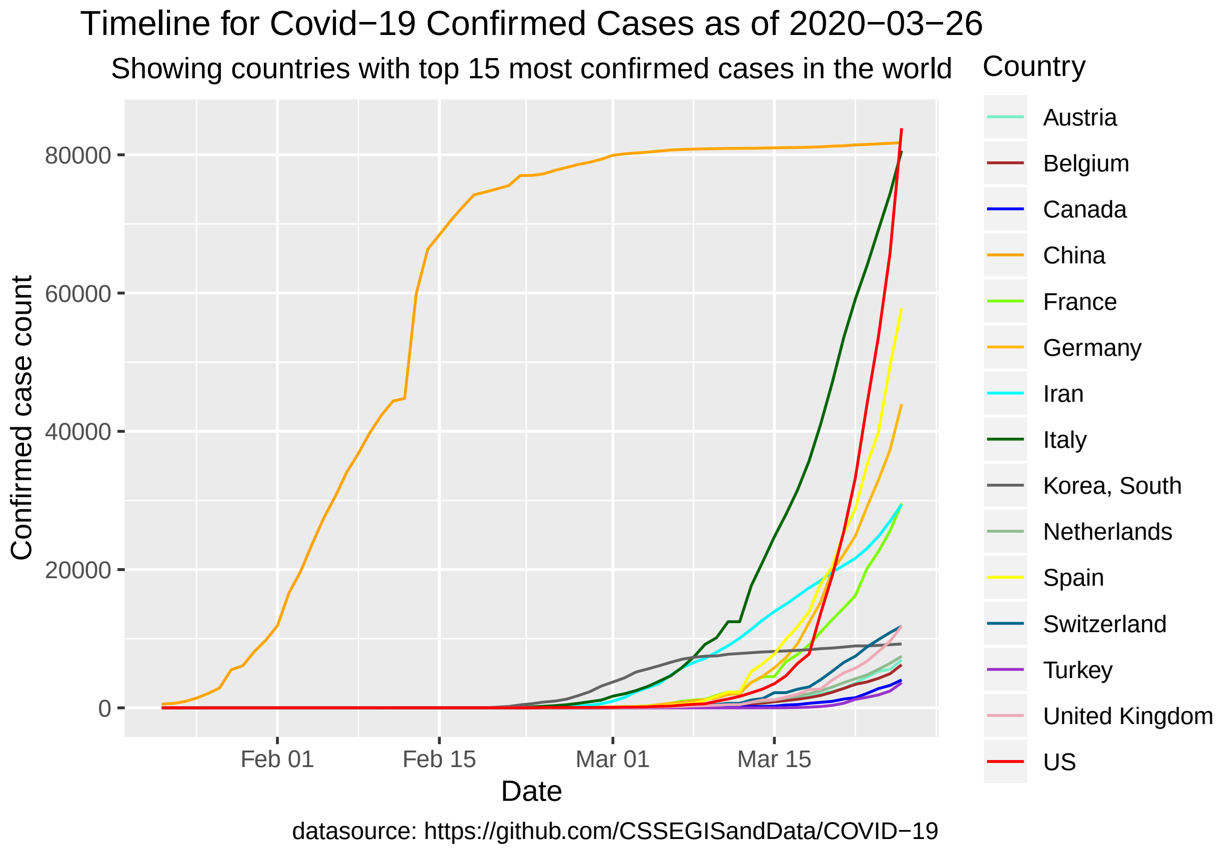
```r
confirmed <- c_ts %>% select(-Province.State, -Lat, -Long) %>%
    gather(Date_temp, value, -Country.Region) %>%
    group_by(Country.Region, Date_temp) %>%
    summarize(Confirmed = sum(value)) %>%
    ungroup() %>%
    mutate(Date_temp = str_replace(Date_temp, "X", "")) %>%
    mutate(Date = as.POSIXct(strptime(Date_temp, "%m.%d.%y")))

deaths <- d_ts %>% select(-Province.State, -Lat, -Long) %>%
    gather(Date_temp, value, -Country.Region) %>%
    group_by(Country.Region, Date_temp) %>%
    summarize(Deaths = sum(value)) %>%
    ungroup() %>%
    mutate(Date_temp = str_replace(Date_temp, "X", "")) %>%
    mutate(Date = as.POSIXct(strptime(Date_temp, "%m.%d.%y")))

all <- cbind(confirmed[, c(1, 4)], confirmed[, 3], deaths[3])

str(all)
```

```
## 'data.frame':    11375 obs. of  4 variables:
##  $ Country.Region: Factor w/ 175 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date          : POSIXct, format: "2020-01-22" "2020-01-23" ...
##  $ Confirmed     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Deaths        : int  0 0 0 0 0 0 0 0 0 0 ...
```

```r
today <- max(confirmed$Date)
```

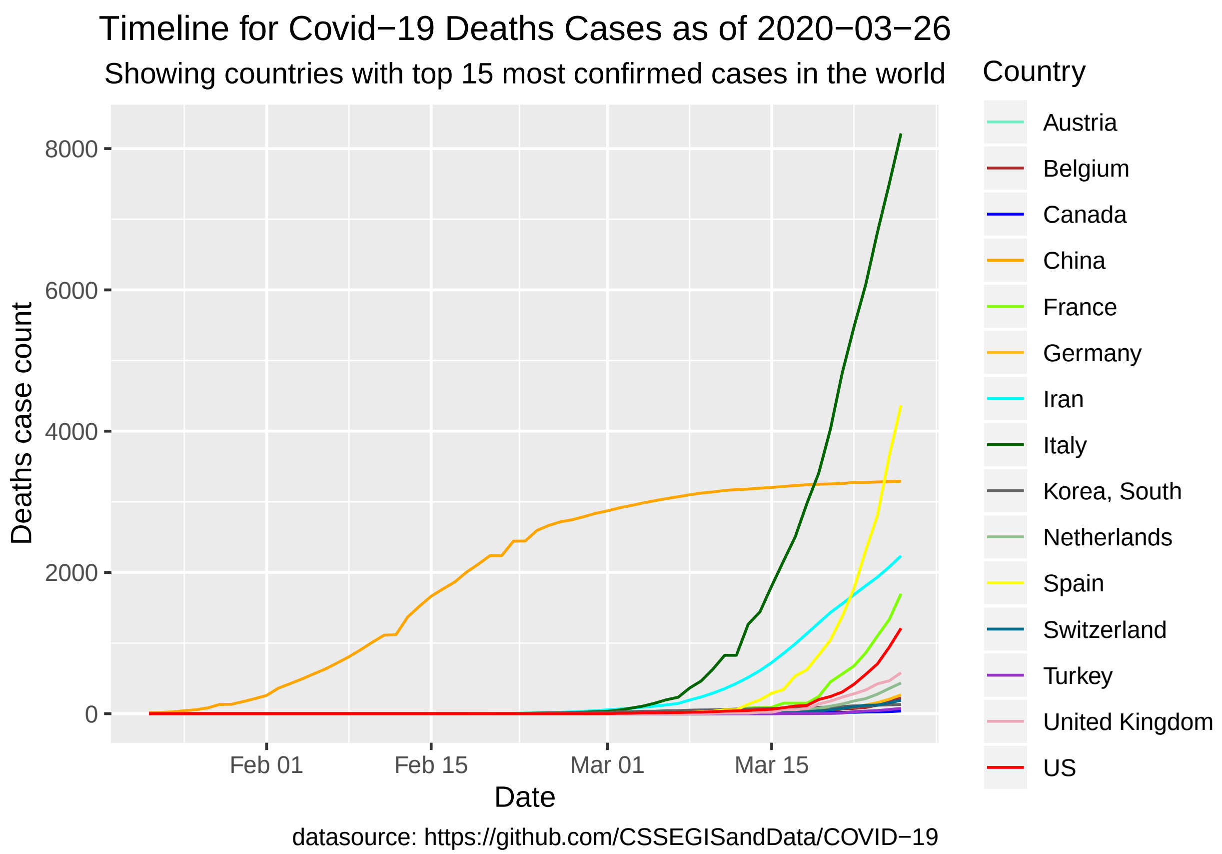
I would like to plot the time series data for a few countries with the most confirmed cases.

```r
top_confirmed_countries <- confirmed %>% filter(Date == today) %>%
    arrange(desc(Confirmed)) %>%
    top_n(select_top, Confirmed) %>%
    mutate(Country.Region = as.character(Country.Region)) %>%
    .$Country.Region

# confirmed %>% filter(Country.Region == "United Kingdom")

p_c <- confirmed %>% filter(Country.Region %in% top_confirmed_countries) %>%
    ggplot(aes(Date, Confirmed, color = Country.Region)) +
    geom_line() +
    theme(plot.title = element_text(hjust = 0.5),
          plot.subtitle = element_text(hjust = 0.5)) +
    labs(title = paste("Timeline for Covid-19 Confirmed Cases as of", today, sep = " "),
         subtitle = paste("Showing countries with top", select_top,
                          "most confirmed cases in the world", sep = " "),
         x = "Date",
         y = "Confirmed case count",
         caption = "datasource: https://github.com/CSSEGISandData/COVID-19",
         color = "Country")
if (select_top <= length(manual_colors)) {
  p_c <- p_c + scale_color_manual(values = manual_colors)
}

p_c
```



The above plot shows how the covid-19 confirmed case changed in the past few months for each country displayed.

```
p_d <- deaths %>% filter(Country.Region %in% top_confirmed_countries) %>%
    ggplot(aes(Date, Deaths, color = Country.Region)) +
    geom_line() +
    theme(plot.title = element_text(hjust = 0.5),
          plot.subtitle = element_text(hjust = 0.5)) +
    labs(title = paste("Timeline for Covid-19 Deaths Cases as of", today, sep = " "),
         subtitle = paste("Showing countries with top", select_top,
                          "most confirmed cases in the world", sep = " "),
         x = "Date",
         y = "Deaths case count",
         caption = "datasource: https://github.com/CSSEGISandData/COVID-19",
         color = "Country")
if (select_top <= length(manual_colors)) {
  p_d <- p_d + scale_color_manual(values = manual_colors)
}

p_d
```



The above plot shows how the covid-19 deaths case changed in the past few months for each country displayed.

I would like to combine country information such as population, density and median_age into my analysis.

I am getting the world population data.

```
source("WorldPopulation.R")
wp <- getWorldPopulation()
```

I am joining the world population data with the covid-19 data by country name.

```
wppd <- wp %>%
    mutate(Density = str_replace_all(Density, ",", "")) %>%
    mutate(Density = as.numeric(Density)) %>%
    mutate(Population = str_replace_all(Population, ",", "")) %>%
    mutate(Population = as.numeric(Population)) %>%
    select(Country.Region, Density, Population, Median_Age)

all <- all %>% mutate(Country.Region = as.character(Country.Region))

ALL <- left_join(all, wppd, by = "Country.Region")

# sapply(ALL, function(col) {sum(is.na(col))})
unique(ALL[is.na(ALL$Density),]$Country.Region)
```

```
## [1] "Congo (Brazzaville)"   "Congo (Kinshasa)"      "Diamond Princess"
## [4] "Kosovo"                "Saint Kitts and Nevis" "West Bank and Gaza"
```

Table 1: World Covid-19 Summary 2020-03-26 C/Population, D/Population, R/Population are per 10,000 people

Country.Region	Date	Confirmed	Deaths	Density	Population	Median_Age	D/C %	C/Population	D/Population	D/C % by Density
US	2020-03-26	83836	1209	36	3.3e+08	38	1.44	2.53	0.04	0.04
China	2020-03-26	81782	3291	153	1.4e+09	38	4.02	0.57	0.02	0.03
Italy	2020-03-26	80589	8215	206	6.0e+07	47	10.19	13.33	1.36	0.05
Spain	2020-03-26	57786	4365	94	4.7e+07	45	7.55	12.36	0.93	0.08
Germany	2020-03-26	43938	267	240	8.4e+07	46	0.61	5.24	0.03	0.00
France	2020-03-26	29551	1698	119	6.5e+07	42	5.75	4.53	0.26	0.05
Iran	2020-03-26	29406	2234	52	8.4e+07	32	7.60	3.50	0.27	0.15
United Kingdom	2020-03-26	11812	580	281	6.8e+07	40	4.91	1.74	0.09	0.02
Switzerland	2020-03-26	11811	191	219	8.7e+06	43	1.62	13.65	0.22	0.01
Korea, South	2020-03-26	9241	131	527	5.1e+07	44	1.42	1.80	0.03	0.00
Netherlands	2020-03-26	7468	435	508	1.7e+07	43	5.82	4.36	0.25	0.01
Austria	2020-03-26	6909	49	109	9.0e+06	43	0.71	7.67	0.05	0.01
Belgium	2020-03-26	6235	220	383	1.2e+07	42	3.53	5.38	0.19	0.01
Canada	2020-03-26	4042	38	4	3.8e+07	41	0.94	1.07	0.01	0.24
Turkey	2020-03-26	3629	75	110	8.4e+07	32	2.07	0.43	0.01	0.02

```
# Three countries do not get a matching density

# There are two Congo entries in covid-19 data, one congo entry from the world population data
ALL$Density[str_detect(ALL$Country.Region, "Congo")] <-
    wppd$Density[wppd$Country.Region == "Congo"]

ALL$Population[str_detect(ALL$Country.Region, "Congo")] <-
    wppd$Population[wppd$Country.Region == "Congo"]/2

# Cruise ship is not a real country.
ALL$Density[str_detect(ALL$Country.Region, "Diamond Princess")] <- 10
ALL$Population[str_detect(ALL$Country.Region, "Diamond Princess")] <-2000
```

I am calculating below columns.

C/Population and D/Population shows number of cases per 10,000 people.

"D/C by Density" is calculated by Deaths/Confirmed divided by country density. (Density here is number of people per square km). So this rate removes the density factor. If a country has higher density, that makes the virus to be transmitted more easily.

```
today_ALL <- ALL %>% filter(Date == today) %>%
    mutate("D/C %" = (Deaths/Confirmed)*100) %>%
    mutate("C/Population" = (Confirmed/Population)*10000,
           "D/Population" = (Deaths/Population)*10000) %>%
    mutate("D/C % by Density" = `D/C %`/Density) %>%
    arrange(desc(Confirmed))

knitr::kable(slice(today_ALL, 1:select_top),
  caption = paste("World Covid-19 Summary", today, "C/Population, D/Population, R/Population are per 10
  format="latex", booktabs=TRUE) %>%
  kable_styling(latex_options="scale_down")
```

Below are my personal opinion from table 1. I could be wrong. Please let me know what you think. I will really appreciate it.

1. If not considering the population and density factors, China has the highest confirmed case. Italy has the highest Deaths/Confirmed rate (D/C %).

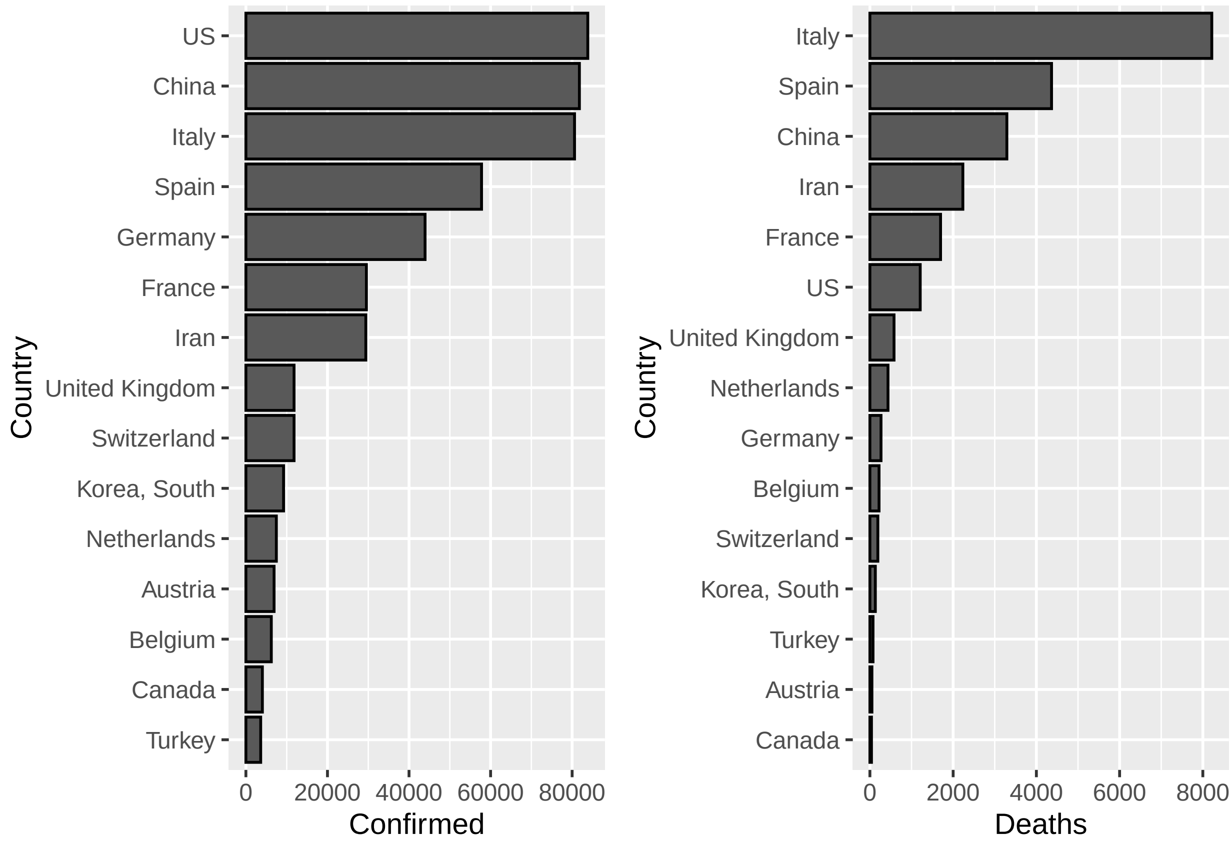
2. Adding consideration of the countries' population, Italy has the highest Confirmed and Death cases per 10,000 people.

3. Adding the consideration of population density, Iran has relative high Deaths/Confirmed by density. Italy is not on the top of this list.

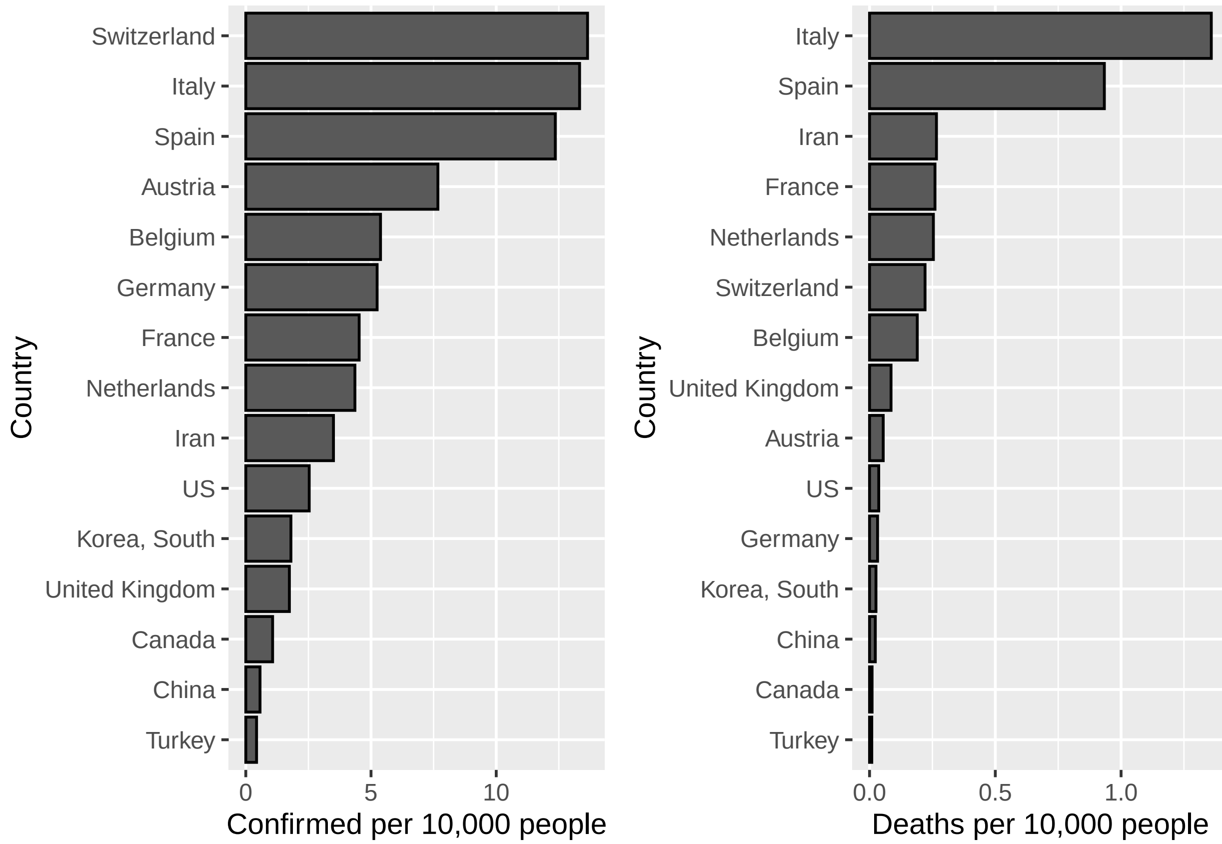
4. For most of the countries which have higher confirmed cases, most of them have D/C % by Density within 0.05. This means to me that the virus transmission seems to be similar across ALL countries.

5. There can be other factors that contribute to D/C % by Density. If a country has more people older than a certain age, it will be more affected since covid-19 has much worse impact on older people. Italy is a great exmaple. Better medical facilities will have positive effect on this.

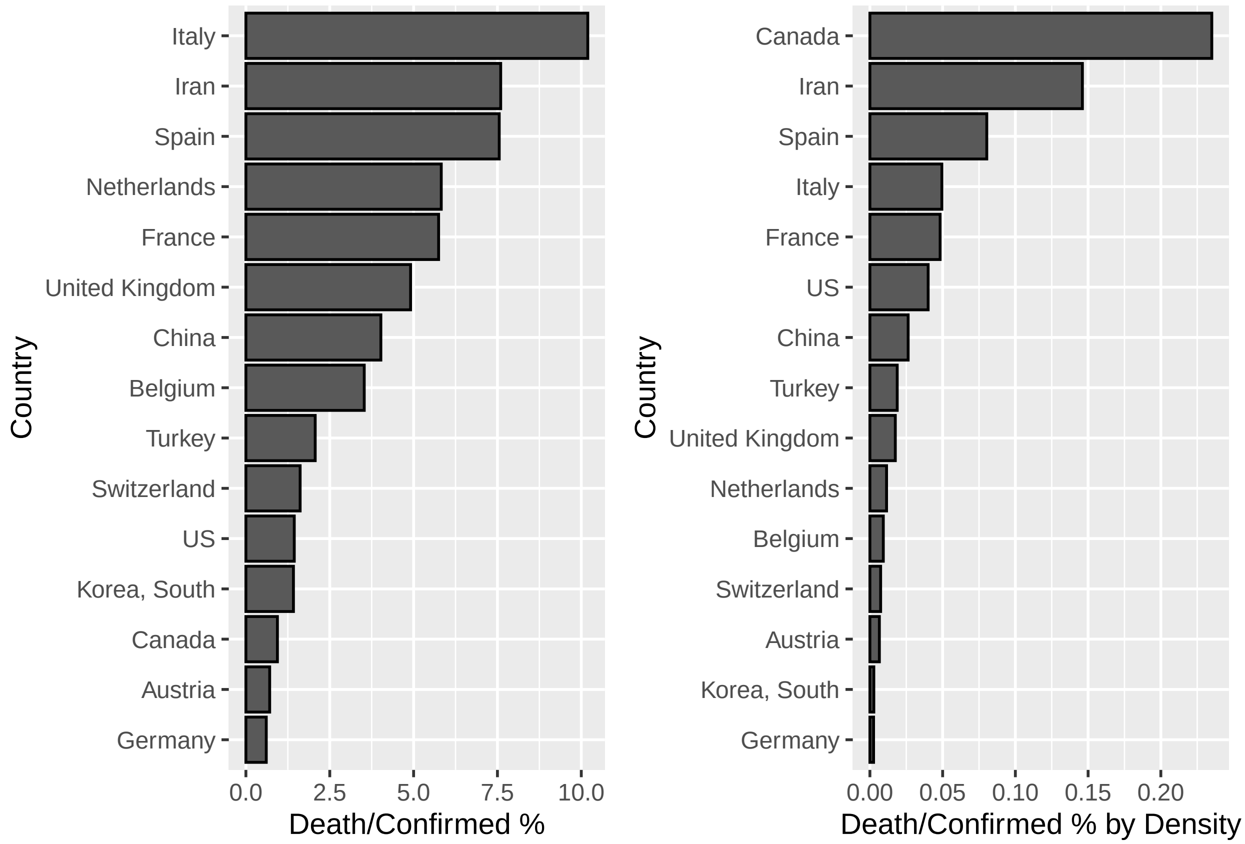
```
today_ALL <- today_ALL %>%
  mutate(Country.Region = as.factor(Country.Region)) %>%
  slice(1:select_top)

p_confirmed <- today_ALL %>%
  mutate(Country.Region = reorder(Country.Region, Confirmed, FUN = mean)) %>%
  ggplot(aes(Country.Region, Confirmed)) +
  geom_bar(stat = "identity", color = "black") +
  labs(y = "Confirmed",
       x = "Country") +
  coord_flip()

p_deaths <- today_ALL %>%
  mutate(Country.Region = reorder(Country.Region, Deaths, FUN = mean)) %>%
  ggplot(aes(Country.Region, Deaths)) +
  geom_bar(stat = "identity", color = "black") +
  labs(y = "Deaths",
       x = "Country") +
  coord_flip()

grid.arrange(p_confirmed, p_deaths, ncol = 2)
```



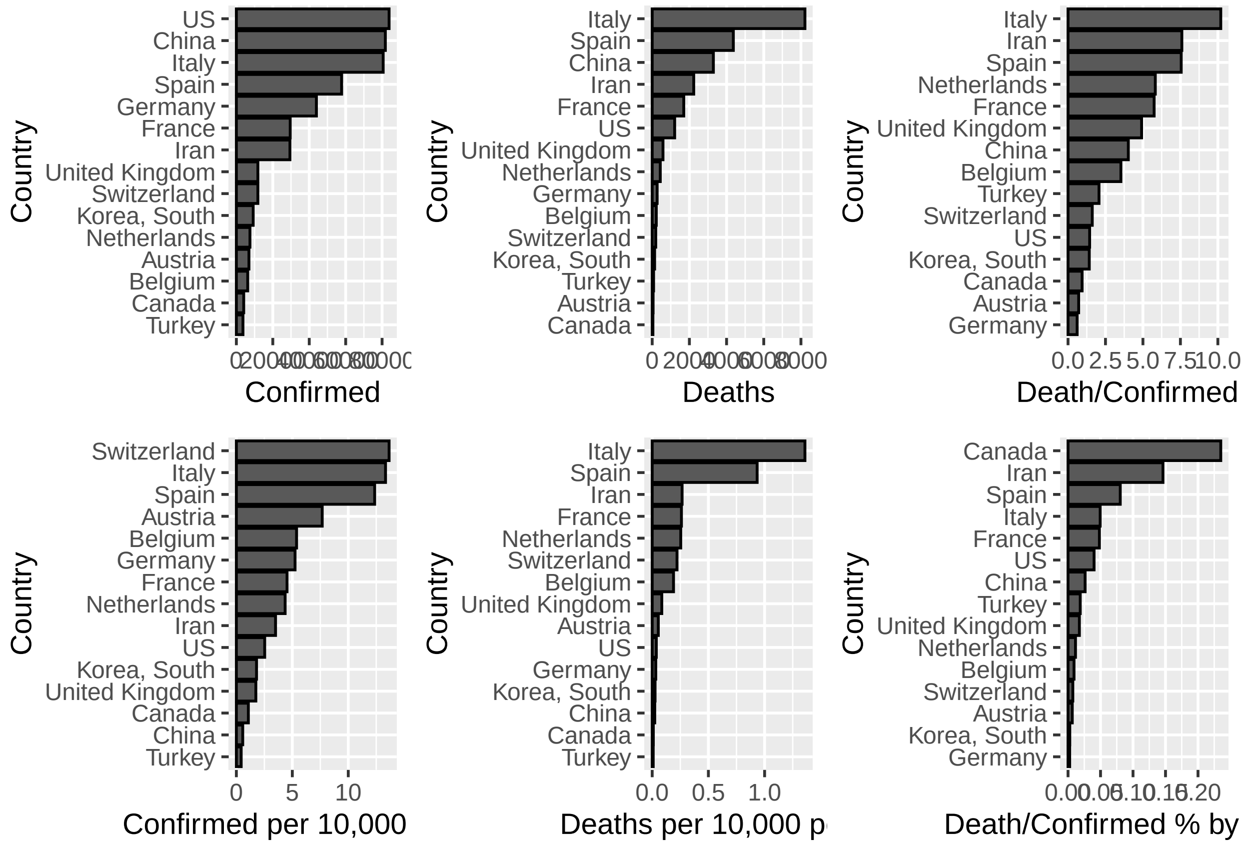

```r
p_c_population <- today_ALL %>%
  mutate(Country.Region = reorder(Country.Region, `C/Population`, FUN = mean)) %>%
  ggplot(aes(Country.Region, `C/Population`)) +
  geom_bar(stat = "identity", color = "black") +
  labs(y = "Confirmed per 10,000 people",
       x = "Country") +
  coord_flip()

p_d_population <- today_ALL %>%
  mutate(Country.Region = reorder(Country.Region, `D/Population`, FUN = mean)) %>%
  ggplot(aes(Country.Region, `D/Population`)) +
  geom_bar(stat = "identity", color = "black") +
  labs(y = "Deaths per 10,000 people",
       x = "Country") +
  coord_flip()

grid.arrange(p_c_population, p_d_population, ncol = 2)
```




```
p_deaths_confirmed <- today_ALL %>%
  mutate(Country.Region = reorder(Country.Region, `D/C %`, FUN = mean)) %>%
  ggplot(aes(Country.Region, `D/C %`)) +
  geom_bar(stat = "identity", color = "black") +
  labs(y = "Death/Confirmed %",
       x = "Country") +
  coord_flip()

p_deaths_confirmed_by_density <- today_ALL %>%
  mutate(Country.Region = reorder(Country.Region, `D/C % by Density`, FUN = mean)) %>%
  ggplot(aes(Country.Region, `D/C % by Density`)) +
  geom_bar(stat = "identity", color = "black") +
  labs(y = "Death/Confirmed % by Density",
       x = "Country") +
  coord_flip()

grid.arrange(p_deaths_confirmed, p_deaths_confirmed_by_density, ncol = 2)
```




```r
grid.arrange(p_confirmed, p_deaths, p_deaths_confirmed,
             p_c_population, p_d_population, p_deaths_confirmed_by_density,
             ncol = 3, nrow = 2)
```

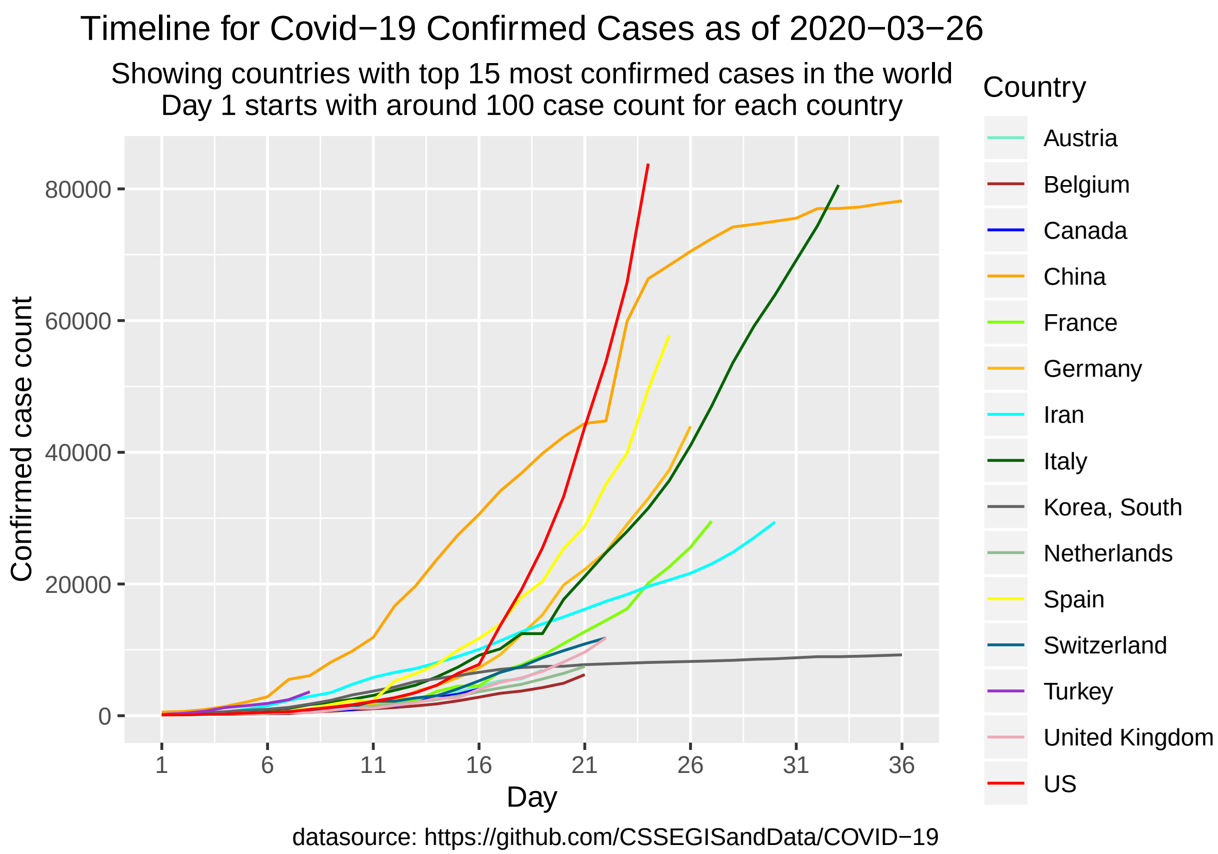


I am going to find out the day that the country has around 100 confirmed cases, then plot each country at the same starting point. This way, it is easier to do visual comparison among the countries.

```
ALL_day1 <- ALL %>% filter(Confirmed >= day1_count) %>% group_by(Country.Region) %>%
    arrange(Confirmed) %>% mutate(Day = row_number()) %>% ungroup()

x_lim_max <- ALL_day1 %>% filter(Country.Region %in% top_confirmed_countries) %>%
  filter(Country.Region != "China") %>%
  arrange(desc(Day)) %>%
  select(Day) %>%
  summarize(max_day = max(Day)) %>%
  pull(max_day)


p_day1_base <- ALL_day1 %>% filter(Country.Region %in% top_confirmed_countries) %>%
    ggplot(aes(Day, Confirmed, color = Country.Region)) +
    geom_line() +
    theme(plot.title = element_text(hjust = 0.5),
          plot.subtitle = element_text(hjust = 0.5)) +
    scale_x_continuous(breaks = seq(1, x_lim_max, 5), lim = c(1, x_lim_max))

if(select_top <= length(manual_colors)) {
  p_day1_base <- p_day1_base + scale_color_manual(values = manual_colors)
}

p_day1 <- p_day1_base +
```

```
    labs(title = paste("Timeline for Covid-19 Confirmed Cases as of", today, sep = " "),
         subtitle = paste("Showing countries with top", select_top,
    "most confirmed cases in the world\nDay 1 starts with around 100 case count for each country",
                  sep = " "),
         x = "Day",
         y = "Confirmed case count",
         caption = "datasource: https://github.com/CSSEGISandData/COVID-19",
         color = "Country")
p_day1
```

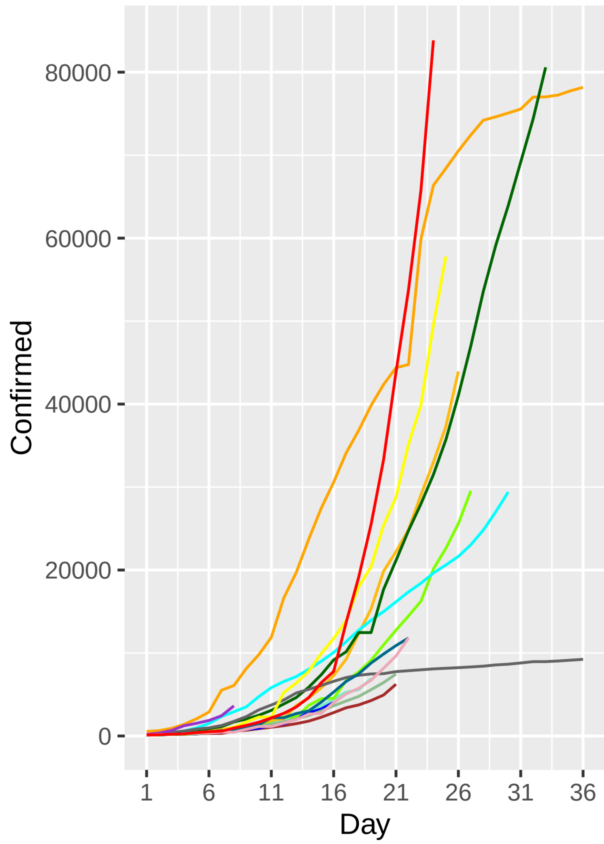
```
## Warning: Removed 29 rows containing missing values (geom_path).
```

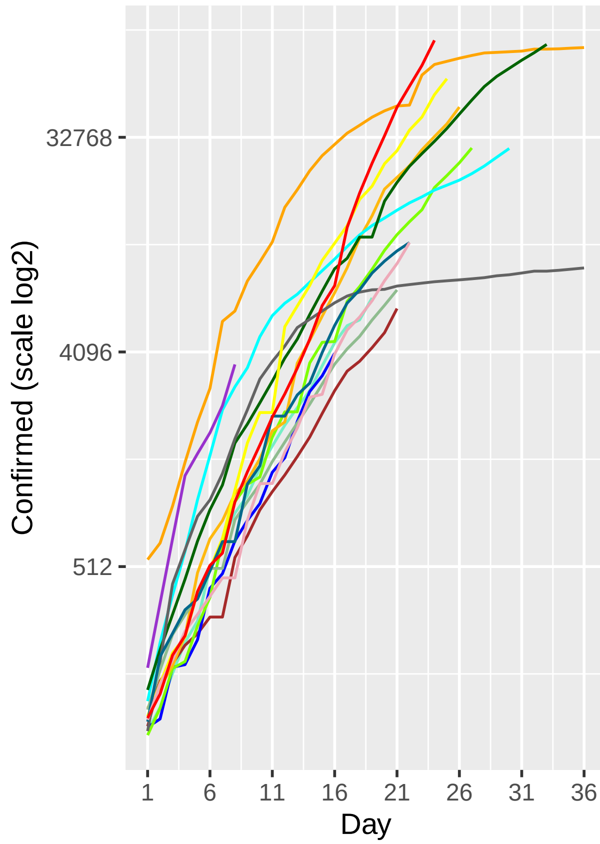


Show the same plot side by side with the confirmed case being transformed on log2 scale

```
p_day1_log2 <- p_day1_base +
  scale_y_continuous(trans = "log2") +
  theme(legend.position = "none") +
  labs(y = "Confirmed (scale log2)")
grid.arrange(p_day1_base + theme(legend.position = "none"), p_day1_log2, ncol = 2)
```

```
## Warning: Removed 29 rows containing missing values (geom_path).

## Warning: Removed 29 rows containing missing values (geom_path).
```





Below is what I see from the above plots.

1. China had the fastest growth at the beginning, most likely due to the dense population and no prepareness being the first being hit.
2. European countries were the next that got most impacted. The countries are Spain, Germany and Italy.
3. US started after European countries got impacted. The confirmed case growth rate of US exceed the other three European countries.
4. From the log2 scaled plot, all countries confirmed growth rate seems to be similar until the situation got controlled.

Modeling and prediction for US data

```
str(ALL_day1)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1195 obs. of  8 variables:
##  $ Country.Region: chr  "France" "Israel" "Taiwan*" "Iraq" ...
##  $ Date          : POSIXct, format: "2020-02-29" "2020-03-12" ...
##  $ Confirmed     : int  100 100 100 101 101 101 102 102 102 102 ...
##  $ Deaths        : int  2 0 1 9 5 0 0 2 0 0 ...
##  $ Density       : num  119 400 673 93 566 25 46 464 18 18 ...
```

```
## $ Population    : num  65273511 8655535 23816775 40222493 33931 ...
## $ Median_Age    : chr  "42" "30" "42" "21" ...
## $ Day           : int  1 1 1 1 1 1 1 1 1 2 ...
```

```
US_all_day1 <- ALL_day1 %>% filter(Country.Region == "US")

US_all_day1
```

```
## # A tibble: 24 x 8
##    Country.Region Date                Confirmed Deaths Density Population
##    <chr>          <dttm>                  <int>  <int>   <dbl>      <dbl>
##  1 US             2020-03-03 00:00:00       118      7      36  331002651
##  2 US             2020-03-04 00:00:00       149     11      36  331002651
##  3 US             2020-03-05 00:00:00       217     12      36  331002651
##  4 US             2020-03-06 00:00:00       262     14      36  331002651
##  5 US             2020-03-07 00:00:00       402     17      36  331002651
##  6 US             2020-03-08 00:00:00       518     21      36  331002651
##  7 US             2020-03-09 00:00:00       583     22      36  331002651
##  8 US             2020-03-10 00:00:00       959     28      36  331002651
##  9 US             2020-03-11 00:00:00      1281     36      36  331002651
## 10 US             2020-03-12 00:00:00      1663     40      36  331002651
## # ... with 14 more rows, and 2 more variables: Median_Age <chr>, Day <int>
```

```
#
us_fit_1 <- US_all_day1 %>% lm(Confirmed ~ Day, data = .)
summary(us_fit_1)
```

```
##
## Call:
## lm(formula = Confirmed ~ Day, data = .)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -17271 -11412  -2481   8617  36956
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -18597       5890   -3.16   0.0046 **
## Day             2728        412    6.62  1.2e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14000 on 22 degrees of freedom
## Multiple R-squared:  0.666,  Adjusted R-squared:  0.65
## F-statistic: 43.8 on 1 and 22 DF,  p-value: 1.18e-06
```

```
us_fit_2 <- US_all_day1 %>% mutate(Day2 = Day^2) %>% lm(Confirmed ~ Day + Day2, data = .)
summary(us_fit_2)
```

```
##
## Call:
## lm(formula = Confirmed ~ Day + Day2, data = .)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8359  -4803    243   4057  12610
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12676.9     3694.7    3.43   0.0025 **
## Day          -4488.9      681.0   -6.59  1.6e-06 ***
## Day2           288.7       26.4   10.92  4.1e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5540 on 21 degrees of freedom
## Multiple R-squared:  0.95,   Adjusted R-squared:  0.945
## F-statistic:  199 on 2 and 21 DF,  p-value: 2.22e-14
```

```r
tail(US_all_day1, 6)
```

```
## # A tibble: 6 x 8
##   Country.Region Date                Confirmed Deaths Density Population
##   <chr>          <dttm>                  <int>  <int>   <dbl>      <dbl>
## 1 US             2020-03-21 00:00:00     25489    307      36  331002651
## 2 US             2020-03-22 00:00:00     33276    417      36  331002651
## 3 US             2020-03-23 00:00:00     43847    557      36  331002651
## 4 US             2020-03-24 00:00:00     53740    706      36  331002651
## 5 US             2020-03-25 00:00:00     65778    942      36  331002651
## 6 US             2020-03-26 00:00:00     83836   1209      36  331002651
## # ... with 2 more variables: Median_Age <chr>, Day <int>
```

```r
test_Day <- seq(1:100)
test_Day2 <- test_Day^2
test_data_2 <- data.frame(Day = test_Day, Day2 = test_Day2)
y_hat_2_20200327 <- predict(us_fit_2, newdata = test_data_2)
y_hat_2_20200327
```

```
##       1       2       3       4       5       6       7       8       9      10
##    8477    4854    1808    -660   -2550   -3864   -4600   -4758   -4340   -3343
##      11      12      13      14      15      16      17      18      19      20
##   -1770     381    3109    6415   10298   14758   19796   25411   31603   38373
##      21      22      23      24      25      26      27      28      29      30
##   45720   53645   62147   71226   80883   91117  101928  113317  125283  137826
##      31      32      33      34      35      36      37      38      39      40
##  150947  164645  178921  193774  209204  225212  241797  258960  276699  295016
##      41      42      43      44      45      46      47      48      49      50
##  313911  333383  353432  374059  395263  417044  439403  462339  485853  509943
##      51      52      53      54      55      56      57      58      59      60
##  534612  559857  585680  612080  639058  666613  694746  723455  752743  782607
##      61      62      63      64      65      66      67      68      69      70
##  813049  844068  875665  907839  940590  973919 1007825 1042309 1077369 1113008
##      71      72      73      74      75      76      77      78      79      80
## 1149223 1186016 1223387 1261334 1299859 1338962 1378641 1418899 1459733 1501145
##      81      82      83      84      85      86      87      88      89      90
```

```
## 1543134 1585701 1628845 1672566 1716865 1761741 1807195 1853225 1899834 1947019
##      91      92      93      94      95      96      97      98      99     100
## 1994782 2043123 2092040 2141535 2191608 2242257 2293485 2345289 2397671 2450630
```

```
us_fit_3 <- US_all_day1 %>% mutate(Day2 = Day^2, Day3 = Day^3) %>% lm(Confirmed ~ Day + Day2 + Day3, da
summary(us_fit_3)
```

```
##
## Call:
## lm(formula = Confirmed ~ Day + Day2 + Day3, data = .)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1917   -814   -191    784   2205
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4507.597   1197.270   -3.76   0.0012 **
## Day          3005.737    406.312    7.40  3.8e-07 ***
## Day2         -445.698     37.361  -11.93  1.5e-10 ***
## Day3           19.584      0.984   19.91  1.2e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1240 on 20 degrees of freedom
## Multiple R-squared:  0.998,  Adjusted R-squared:  0.997
## F-statistic: 2.76e+03 on 3 and 20 DF,  p-value: <2e-16
```

```
test_Day3 <- test_Day^3
test_data_3 <- data.frame(Day = test_Day, Day2 = test_Day2, Day3 = test_Day3)
y_hat_3_20200327 <- predict(us_fit_3, newdata = test_data_3)
y_hat_3_20200327
```

```
##       1       2       3       4       5       6       7       8
##   -1928    -122    1027    1638    1827    1712    1411    1040
##       9      10      11      12      13      14      15      16
##     719     564     692    1221    2269    3953    6391    9700
##      17      18      19      20      21      22      23      24
##   13997   19401   26028   33996   43423   54426   67123   81631
##      25      26      27      28      29      30      31      32
##   98067  116550  137196  160123  185449  213291  243767  276994
##      33      34      35      36      37      38      39      40
##  313090  352171  394357  439763  488508  540710  596484  655950
##      41      42      43      44      45      46      47      48
##  719225  786426  857671  933076 1012760 1096841 1185435 1278660
##      49      50      51      52      53      54      55      56
## 1376634 1479474 1587298 1700223 1818367 1941846 2070780 2205284
##      57      58      59      60      61      62      63      64
## 2345478 2491477 2643400 2801364 2965487 3135886 3312679 3495983
##      65      66      67      68      69      70      71      72
## 3685915 3882594 4086137 4296660 4514283 4739121 4971293 5210917
##      73      74      75      76      77      78      79      80
## 5458109 5712987 5975669 6246272 6524914 6811712 7106783 7410246
```

```
##       81       82       83       84       85       86       87       88
##  7722218  8042815  8372157  8710359  9057541  9413818  9779310 10154132
##       89       90       91       92       93       94       95       96
## 10538403 10932240 11335761 11749084 12172325 12605602 13049033 13502735
##       97       98       99      100
## 13966825 14441422 14926643 15422605
```

```
us_fit_4 <- US_all_day1 %>% mutate(Day2 = Day^2, Day3 = Day^3, Day4 = Day^4) %>% lm(Confirmed ~ Day + Da
summary(us_fit_4)
```

```
##
## Call:
## lm(formula = Confirmed ~ Day + Day2 + Day3 + Day4, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2146.1  -227.3    74.1   417.0  1515.4
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -440.534   1057.968   -0.42    0.682
## Day          253.807    560.618    0.45    0.656
## Day2          26.475     88.628    0.30    0.768
## Day3          -9.384      5.276   -1.78    0.091 .
## Day4           0.579      0.105    5.53  2.5e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 790 on 19 degrees of freedom
## Multiple R-squared:  0.999,  Adjusted R-squared:  0.999
## F-statistic: 5.15e+03 on 4 and 19 DF,  p-value: <2e-16
```

```
test_Day4 <- test_Day^4
test_data_4 <- data.frame(Day = test_Day, Day2 = test_Day2, Day3 = test_Day3, Day4 = test_Day4)
y_hat_4_20200327 <- predict(us_fit_4, newdata = test_data_4)
y_hat_4_20200327[1:26]
```

```
##      1      2      3      4      5      6      7      8      9     10     11
##   -169    107    353    546    679    759    806    853    948   1154   1547
##     12     13     14     15     16     17     18     19     20     21     22
##   2215   3263   4808   6982   9929  13809  18796  25075  32849  42332  53752
##     23     24     25     26
##  67352  83390 102134 123870
```

```
US_all_day1$Confirmed[1:24]
```

```
##  [1]   118   149   217   262   402   518   583   959  1281  1663  2179  2727
## [13]  3499  4632  6421  7783 13677 19100 25489 33276 43847 53740 65778 83836
```