Covid Project

James Gabriel

2022-10-19

Covid Data Analysis

The dataset of our study contains daily & cumulative number of COVID-19 tests conducted, number of positive, hospitalized, recovered & death cases reported by country. In details here are the columns in the dataset:

1. Date: Date
2. Continent_Name: Continent names
3. Two_Letter_Country_Code: Country codes
4. Country_Region: Country names
5. Province_State: States/province names; value is All States when state/provincial level data is not available
6. positive: Cumulative number of positive cases reported.
7. active: Number of active cases on that day.
8. hospitalized: Cumulative number of hospitalized cases reported.
9. hospitalizedCurr: Number of actively hospitalized cases on that day.
10. recovered: Cumulative number of recovered cases reported.
11. death: Cumulative number of deaths reported.
12. total_tested: Cumulative number of tests conducted.
13. daily_tested: Number of tests conducted on the day; if daily data is unavailable, daily tested is averaged across number of days in between.
14. daily_positive: Number of positive cases reported on the day; if daily data is unavailable, daily positive is averaged across number of days in.

This analysis tries to provide an answer to this question: Which countries have had the highest number of positive cases against the number of tests?

Firstly, we'd load in the `tidyverse` package as we would be using this for our analysis.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
covid_df <- read_csv('covid.csv')
```

```
## Rows: 10903 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): Continent_Name, Two_Letter_Country_Code, Country_Region, Province_...
## dbl  (9): positive, hospitalized, recovered, death, total_tested, active, ho...
## date (1): Date
##
## 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.
```

To have a brief idea of what the dataset looks like.

```
dim(covid_df)
```

```
## [1] 10903    14
```

Displaying the column names.

```
vector_cols <- colnames(covid_df)
```

```
vector_cols
```

```
##  [1] "Date"                    "Continent_Name"
##  [3] "Two_Letter_Country_Code" "Country_Region"
##  [5] "Province_State"          "positive"
##  [7] "hospitalized"            "recovered"
##  [9] "death"                   "total_tested"
## [11] "active"                  "hospitalizedCurr"
## [13] "daily_tested"            "daily_positive"
```

```
class(covid_df$Country_Region)
```

```
## [1] "character"
```

```
class(vector_cols)
```

```
## [1] "character"
```

Let's diaplay a few rows of the dataset to explore it visually.

```
head(covid_df)
```

```
## # A tibble: 6 x 14
##   Date       Continent_N~1 Two_L~2 Count~3 Provi~4 posit~5 hospi~6 recov~7 death
##   <date>     <chr>         <chr>   <chr>   <chr>     <dbl>   <dbl>   <dbl> <dbl>
## 1 2020-01-20 Asia          KR      South ~ All St~       1       0       0     0
## 2 2020-01-22 North America US      United~ All St~       1       0       0     0
```

```
## 3 2020-01-22 North America US      United~ Washin~       1       0       0     0
## 4 2020-01-23 North America US      United~ All St~       1       0       0     0
## 5 2020-01-23 North America US      United~ Washin~       1       0       0     0
## 6 2020-01-24 Asia          KR      South ~ All St~       2       0       0     0
## # ... with 5 more variables: total_tested <dbl>, active <dbl>,
## #   hospitalizedCurr <dbl>, daily_tested <dbl>, daily_positive <dbl>, and
## #   abbreviated variable names 1: Continent_Name, 2: Two_Letter_Country_Code,
## #   3: Country_Region, 4: Province_State, 5: positive, 6: hospitalized,
## #   7: recovered
## # i Use `colnames()` to see all variable names
```

The `glimpse` function is used to have an idea of the different data types of each column with a few samples.

```
glimpse(covid_df)
```

```
## Rows: 10,903
## Columns: 14
## $ Date                    <date> 2020-01-20, 2020-01-22, 2020-01-22, 2020-01-2~
## $ Continent_Name          <chr> "Asia", "North America", "North America", "Nor~
## $ Two_Letter_Country_Code <chr> "KR", "US", "US", "US", "US", "KR", "US", "US"~
## $ Country_Region          <chr> "South Korea", "United States", "United States~
## $ Province_State          <chr> "All States", "All States", "Washington", "All~
## $ positive                <dbl> 1, 1, 1, 1, 1, 2, 1, 1, 4, 0, 3, 0, 0, 0, 0, 1~
## $ hospitalized            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ recovered               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ death                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ total_tested            <dbl> 4, 1, 1, 1, 1, 27, 1, 1, 0, 0, 0, 0, 0, 0, 0, ~
## $ active                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ hospitalizedCurr        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ daily_tested            <dbl> 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ daily_positive          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Considering most of the data is either referring to a particular province or the entire country, it is necessary to have consistency to ensure proper analysis. For this reason, I'm filtering out the data to contain only rows where the `Province_State` says `All States`. I would also be dropping the `Province_State` column.

```
covid_df_all_states <- covid_df %>% filter(Province_State=='All States') %>% select(-Province_State)
```

```
covid_df_all_states
```

```
## # A tibble: 3,781 x 13
##    Date       Continent_~1 Two_L~2 Count~3 posit~4 hospi~5 recov~6 death total~7
##    <date>     <chr>        <chr>   <chr>     <dbl>   <dbl>   <dbl> <dbl>   <dbl>
##  1 2020-01-20 Asia         KR      South ~       1       0       0     0       4
##  2 2020-01-22 North Ameri~ US      United~       1       0       0     0       1
##  3 2020-01-23 North Ameri~ US      United~       1       0       0     0       1
##  4 2020-01-24 Asia         KR      South ~       2       0       0     0      27
##  5 2020-01-24 North Ameri~ US      United~       1       0       0     0       1
##  6 2020-01-25 Oceania      AU      Austra~       4       0       0     0       0
##  7 2020-01-25 Europe       GB      United~       1       0       0     0      31
##  8 2020-01-25 North Ameri~ US      United~       1       0       0     0       1
##  9 2020-01-26 Oceania      AU      Austra~       4       0       0     0       0
```

```
## 10 2020-01-26 Asia         IL      Israel        0       0       0     0       3
## # ... with 3,771 more rows, 4 more variables: active <dbl>,
## #   hospitalizedCurr <dbl>, daily_tested <dbl>, daily_positive <dbl>, and
## #   abbreviated variable names 1: Continent_Name, 2: Two_Letter_Country_Code,
## #   3: Country_Region, 4: positive, 5: hospitalized, 6: recovered,
## #   7: total_tested
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

The data also contains daily and cummulative aggregation. To ensure consistency, I would be filtering for only columns that record daily changes.

```
covid_df_all_states_daily <- covid_df_all_states %>% select(Date, Country_Region, active, hospitalizedCu
```

```
covid_df_all_states_daily
```

```
## # A tibble: 3,781 x 6
##    Date       Country_Region active hospitalizedCurr daily_tested daily_positive
##    <date>     <chr>           <dbl>            <dbl>        <dbl>          <dbl>
##  1 2020-01-20 South Korea         0                0            0              0
##  2 2020-01-22 United States       0                0            0              0
##  3 2020-01-23 United States       0                0            0              0
##  4 2020-01-24 South Korea         0                0            5              0
##  5 2020-01-24 United States       0                0            0              0
##  6 2020-01-25 Australia           0                0            0              0
##  7 2020-01-25 United Kingdom      0                0            0              0
##  8 2020-01-25 United States       0                0            0              0
##  9 2020-01-26 Australia           0                0            0              0
## 10 2020-01-26 Israel              0                0            0              0
## # ... with 3,771 more rows
## # i Use `print(n = ...)` to see more rows
```

I then checked to see the total number of `tested`, `positive`, `active`, and `hospitalized` cases for each country present in the dataset. After this, I sorted the result in descending order and filtered to just the top 10 countries with the highest `tested` cases.

```
covid_df_all_states_daily_sum <- covid_df_all_states_daily %>%
                                 group_by(Country_Region) %>%
                                 summarise(
                                   tested = sum(daily_tested),
                                   positive = sum(daily_positive),
                                   active = sum(active),
                                   hospitalized = sum(hospitalizedCurr)) %>%
                                 arrange(-tested)
```

```
covid_df_all_states_daily_sum
```

```
## # A tibble: 108 x 5
##    Country_Region   tested positive  active hospitalized
##    <chr>             <dbl>    <dbl>   <dbl>        <dbl>
##  1 United States  17282363  1877179       0            0
##  2 Russia         10542266   406368 6924890            0
##  3 Italy           4091291   251710 6202214      1699003
```

```
##  4 India           3692851    60959       0            0
##  5 Turkey          2031192   163941 2980960            0
##  6 Canada          1654779    90873   56454            0
##  7 United Kingdom  1473672   166909       0            0
##  8 Australia       1252900     7200  134586         6655
##  9 Peru             976790    59497       0            0
## 10 Poland           928256    23987  538203            0
## # ... with 98 more rows
## # i Use 'print(n = ...)' to see more rows
```

```
covid_top_10 <- head(covid_df_all_states_daily_sum, 10)
```

```
covid_top_10
```

```
## # A tibble: 10 x 5
##    Country_Region   tested positive  active hospitalized
##    <chr>             <dbl>    <dbl>   <dbl>        <dbl>
##  1 United States  17282363  1877179       0            0
##  2 Russia         10542266   406368 6924890            0
##  3 Italy           4091291   251710 6202214      1699003
##  4 India           3692851    60959       0            0
##  5 Turkey          2031192   163941 2980960            0
##  6 Canada          1654779    90873   56454            0
##  7 United Kingdom  1473672   166909       0            0
##  8 Australia       1252900     7200  134586         6655
##  9 Peru             976790    59497       0            0
## 10 Poland           928256    23987  538203            0
```

Assigning each of the columns in the top 10 result to various vectors to use for further analysis.

```
countries <- covid_top_10 %>% pull(Country_Region)

tested_cases <- covid_top_10 %>% pull(tested)

positive_cases <- covid_top_10 %>% pull(positive)

active_cases <- covid_top_10 %>% pull(active)

hospitalized_cases <- covid_top_10 %>% pull(hospitalized)
```

```
tested_cases
```

```
##  [1] 17282363 10542266  4091291  3692851  2031192  1654779  1473672  1252900
##  [9]   976790   928256
```

```
names(tested_cases)
```

```
## NULL
```

Giving names to the values in the vector.

```
names(positive_cases) <- countries
names(active_cases) <- countries
names(hospitalized_cases) <- countries
```

```
names(active_cases)
```

```
##  [1] "United States"  "Russia"         "Italy"          "India"
##  [5] "Turkey"         "Canada"         "United Kingdom" "Australia"
##  [9] "Peru"           "Poland"
```

```
active_cases
```

```
##  United States         Russia          Italy          India         Turkey
##              0        6924890        6202214              0        2980960
##         Canada United Kingdom      Australia           Peru         Poland
##          56454              0         134586              0         538203
```

To answer the question asked earlier, feature engineering had to be put in place to generate a new result that calculates the total number of positive cases divided by the tested cases to find the countries with the highest ratio.

```
positive_cases/tested_cases
```

```
##  United States         Russia          Italy          India         Turkey
##    0.108618191    0.038546552    0.061523368    0.016507300    0.080711720
##         Canada United Kingdom      Australia           Peru         Poland
##    0.054915490    0.113260617    0.005746668    0.060910738    0.025840932
```

`United Kingdom`, `United States`, and `Turkey` took the lead with the highest ratios.

```
positive_tested_top_3 <- positive_cases/tested_cases
```

```
positive_tested_top_3 <- positive_tested_top_3[c(7,1,5)]
```

```
positive_tested_top_3
```

```
## United Kingdom  United States         Turkey
##     0.11326062     0.10861819     0.08071172
```

A matrix was created to store this information.

```
united_kingdom <- c(0.11, 1473672, 166909, 0, 0)
united_states <- c(0.10, 17282363, 1877179, 0, 0)
turkey <- c(0.08, 2031192, 163941, 2980960, 0)
```

```
covid_mat <- rbind(united_kingdom, united_states, turkey)
```

```r
colnames(covid_mat) <- c("Ratio", "tested", "positive", "active", "hospitalized")
```

```r
covid_mat
```

```
##                Ratio   tested positive  active hospitalized
## united_kingdom  0.11  1473672   166909       0            0
## united_states   0.10 17282363  1877179       0            0
## turkey          0.08  2031192   163941 2980960            0
```

```r
question <- "Which countries have had the highest number of positive cases against the number of tests?
```

```r
answer <- c("Positive tested cases" = positive_tested_top_3)
```

```r
data_structure_list <- list( c(covid_df, covid_df_all_states, covid_df_all_states_daily, covid_top_10),
                             covid_mat, c(vector_cols, countries))
```

```r
covid_analysis_list <- list(question, answer, data_structure_list)
```

```r
covid_analysis_list[[2]]
```

```
## Positive tested cases.United Kingdom  Positive tested cases.United States
##                           0.11326062                           0.10861819
##         Positive tested cases.Turkey
##                           0.08071172
```

Conclusion

From the analysis, it was observed that the country with the highest number of positive cases against the number of tested cases was the `United Kingdom`, followed closely by the `United States` and then `Turkey`.