Tarea I: Software para Data Science

Nicole Lastra Quiroz

22-08-2021

Proyecto de Análisis Covid19

I. Objetivo

A través de este proyecto de análisis se busca dar respuesta a las siguientes preguntas:

1. ¿Qué países han tenido el mayor número de pruebas realizadas?
2. ¿Qué países han tenido el mayor número de casos positivos respecto al número de pruebas realizadas?

II. Desarrollo

Procederemos siguiendo el orden de pasos indicados en las instrucciones del proyecto, para lo cual se ha dividido el trabajo en los apartados a) Carga de los datos y b) Procesamiento de los datos, tal como se detalla a continuación:

a) Carga de los datos

Luego de descargar el archivo `covid19.csv` , procederemos a cargamos el archivo y almacenar el resultado bajo la variable `covid_df`.

```
covid_df     <- read.csv('./covid19.csv')
```

Utilizamos la función `dim()` para verificar las dimensiones del data frame.

```
dim (covid_df)
```

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

Estos números se pueden interpretar como la existencia de 10.903 filas y 14 columnas en el documento.

Luego, determinaremos los nombres de las columnas del data frame `covid_df`, almacenando el resultado en la variable `covid_df_cols_names`.

```
covid_df_cols_names  <- colnames(covid_df)
covid_df_cols_names
```

```
##  [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"
```

Para poder observar las primeras filas del data frame`covid_df` utilizaremos la función `head()`.

```
head(covid_df)
```

```
##         Date Continent_Name Two_Letter_Country_Code Country_Region
## 1 2020-01-20           Asia                      KR    South Korea
## 2 2020-01-22  North America                      US  United States
## 3 2020-01-22  North America                      US  United States
## 4 2020-01-23  North America                      US  United States
## 5 2020-01-23  North America                      US  United States
## 6 2020-01-24           Asia                      KR    South Korea
##   Province_State positive hospitalized recovered death total_tested active
## 1     All States        1            0         0     0            4      0
## 2     All States        1            0         0     0            1      0
## 3     Washington        1            0         0     0            1      0
## 4     All States        1            0         0     0            1      0
## 5     Washington        1            0         0     0            1      0
## 6     All States        2            0         0     0           27      0
##   hospitalizedCurr daily_tested daily_positive
## 1                0            0              0
## 2                0            0              0
## 3                0            0              0
## 4                0            0              0
## 5                0            0              0
## 6                0            5              0
```

Para ver el resumen del data frame `covid_df` utilizaremos la función `str()`.

```
str(covid_df)
```

```
## 'data.frame':    10903 obs. of  14 variables:
##  $ Date                   : chr  "2020-01-20" "2020-01-22" "2020-01-22" "2020-01-23" ...
##  $ Continent_Name         : chr  "Asia" "North America" "North America" "North America" ...
##  $ Two_Letter_Country_Code: chr  "KR" "US" "US" "US" ...
##  $ Country_Region         : chr  "South Korea" "United States" "United States" "United States" ...
##  $ Province_State         : chr  "All States" "All States" "Washington" "All States" ...
##  $ positive               : int  1 1 1 1 1 2 1 1 4 0 ...
##  $ hospitalized           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ recovered              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ death                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ total_tested           : int  4 1 1 1 1 27 1 1 0 0 ...
##  $ active                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hospitalizedCurr       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ daily_tested           : int  0 0 0 0 0 5 0 0 0 0 ...
##  $ daily_positive         : int  0 0 0 0 0 0 0 0 0 0 ...
```

Esta función es muy útil cuando se está explorando un nuevo conjunto de datos ya que permite observar algunos detalles de los objetos en memoria. En este caso, podemos observar las categorías que hay disponibles, así como los posibles valores que adoptan éstas, y el tipo de dichos valores (en el caso de este data frame podemos observar `int` en el caso de enteros y `chr` en el caso de caracteres o cadenas).

b) Procesamiento de los datos

Comenzaremos aislando las filas que necesitamos, filtrando las filas relacionadas con "All States" de la columna `Province_States`. Para esto utilizaremos la función del paquete dplyr llamada `filter()`, para lo cual también incluimos la biblioteca necesaria al inicio del documento usando `library(datos)` y `library(tidyverse)`, y almacenaremos provisoriamente el resultado en `covid_df1`.

```
covid_df1 <- filter(covid_df,
                    Province_State == "All States",
                              )
```

Para eliminar la columna `Province_State`, ahora que ya no tiene mayor relevancia dado a que despues de nuestra selección sólo tenemos datos del tipo `"All States"`, utilizaremos la función `select()` y almacenaremos este resultado en `covid_df_all_states`.

Para mostrar el resumen de nuestro resultado utilizaremos `str()`. Observaremos que ahora el data frame contiene sólo 3781 datos de 13 variables.

```
covid_df_all_states <- select(covid_df1,
                              -Province_State
                              )
str(covid_df_all_states)
```

```
## 'data.frame':    3781 obs. of  13 variables:
##  $ Date                   : chr  "2020-01-20" "2020-01-22" "2020-01-23" "2020-01-24" ...
##  $ Continent_Name         : chr  "Asia" "North America" "North America" "Asia" ...
##  $ Two_Letter_Country_Code: chr  "KR" "US" "US" "KR" ...
##  $ Country_Region         : chr  "South Korea" "United States" "United States" "South Korea" ...
##  $ positive               : int  1 1 1 2 1 4 1 1 4 0 ...
##  $ hospitalized           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ recovered              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ death                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ total_tested           : int  4 1 1 27 1 0 31 1 0 3 ...
##  $ active                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hospitalizedCurr       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ daily_tested           : int  0 0 0 5 0 0 0 0 0 0 ...
##  $ daily_positive         : int  0 0 0 0 0 0 0 0 0 0 ...
```

Al revisar las columnas, notaremos que en el data frame hay datos acumulativos y datos diarios, por lo que procederemos a extraer únicamente estos últimos, con la finalidad de evitar un análisis sesgado (el cual se podría dar, por ejemplo, si compararamos una columna con datos acumulados versus una con datos diarios).

Para generar la extracción utilizaremos nuevamente la función `select()` y guardaremos ese resultado en `covid_df_all_states_daily`. Visualizamos el resumen de lo obtenido con `str()`.

```
covid_df_all_states_daily <- select(covid_df_all_states,
                                    Date, Country_Region, active,
                                    hospitalizedCurr,
```

```
                                    daily_tested, daily_positive
                                    )
str(covid_df_all_states_daily)
```

```
## 'data.frame':    3781 obs. of  6 variables:
##  $ Date            : chr  "2020-01-20" "2020-01-22" "2020-01-23" "2020-01-24" ...
##  $ Country_Region  : chr  "South Korea" "United States" "United States" "South Korea" ...
##  $ active          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hospitalizedCurr: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ daily_tested    : int  0 0 0 5 0 0 0 0 0 0 ...
##  $ daily_positive  : int  0 0 0 0 0 0 0 0 0 0 ...
```

En este punto, debemos resumir el data frame `covid_df_all_states_daily` calculando la suma del número de pruebas realizadas, casos positivos, casos activos y cantidad de hospitalizados, agrupando por la columna `Country_Region`. Estos resultados los almacenaremos en un nuevo data frame llamado `covid_df_all_states_daily_sum`.

Para reaizar esta tarea de forma conjunta, definiendo una vez nuestro nuevo data frame, utilizaremos el pipe `%>%` y las funciones de dplyr `group_by()`, para agrupar por `Country_Region`, `summarize()` y `sum()` para generar los resúmenes agrupados de las sumas de las columnas respectivas, siguiendo el orden y etiquetas asignadas en el trabajo:

- Asigne la suma de `daily_tested` a la variable `tested`.
- Asignar la suma de `daily_positive` a la variable `positive`.
- Asignar la suma de `active` a la variable `active`.
- Asigne la suma de `hospitalizedCurr` a la variable `hospitalized`.

Además, utilizaremos la función `arrange()` y `desc()` para ordenar la columna tested de forma decreciente, y la función `head()` para visualizar las primeras filas de nuestro nuevo data frame `covid_df_all_states_daily_sum`.

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

```
## # A tibble: 6 x 5
##   Country_Region   tested positive  active hospitalized
##   <chr>             <int>    <int>   <int>        <int>
## 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
```

Para trabajar sólo con los datos más relevantes para efectos de nuestro análisis, extraeremos sólo las 10 filas superiores del data frame recien creado y guardaremos el resultado en la variable `covid_top_10`.

```
covid_top_10 <- covid_df_all_states_daily_sum[1:10,]
covid_top_10
```

```
## # A tibble: 10 x 5
##    Country_Region   tested positive  active hospitalized
##    <chr>             <int>    <int>   <int>        <int>
##  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
```

De `covid_top_10`, de a cuerdo a lo indicado en las intrucciones, extraeremos la siguiente información:

- El vector `countries` que contiene los valores de la columna `Country_Region`.
- El vector `tested_cases` que contiene los valores de columna `tested`.
- El vector `positive_cases` que contiene los valores de columna `positive`.
- El vector `active_cases` que contiene los valores de la columna `active`.
- El vector `hospitalized_cases` que contiene los valores de la columna `hospitalized`.

```
countries           <- covid_top_10[['Country_Region']]
tested_cases        <- covid_top_10[['tested']]
positive_cases      <- covid_top_10[['positive']]
active_cases        <- covid_top_10[['active']]
hospitalized_cases  <- covid_top_10[['hospitalized']]
```

Con los vectores ya creados, escribimos el código para cambiar sus nombres al de los países corresondientes de cada uno.

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

Para identificar los tres casos positivos más altos versus la mayor cantidad de pruebas realizadas, dividiremos el vector `positive_cases` por el vector `tested_cases` usando el operador `/`. La información resultante la guardaremos provisoriamente como `positive_tested`, para luego ordenar de forma decreciente dichos resultados con la funcion `sort()`, bajo el nombre de `positive_tested_d`. Revisamos los datos para verificar que el resultado es efectivamente correcto.

```
positive_tested    <- positive_cases/tested_cases
positive_tested_d  <- sort(positive_tested, decreasing = TRUE)
positive_tested_d
```

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

Ahora, procederemos a seleccionar únicamente los tres primeros valores, ya que son parte de la infomación que buscamos obtener para responder a nuestras preguntas objetivo. Llamaremos a esta nueva información `positive_tested_top_3`.

```
positive_tested_top_3 <- positive_tested_d[1:3]
positive_tested_top_3
```

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

Para el siguiente paso de nuestro análisis, se debe construir el data frame `positive_tested_top_3_df` usando la información obtenida en `positive_tested_top_3` y el data frame `covid_top_10`, considerando las columnas `Country_Region`, `tested`, `positive`, `active`, `hospitalized` y `ratio`, donde esta última corresponde a las proporciones obtenidas en el paso anterior.

Procederemos definiendo provisoiamente el data frame `top_3`, que contiene la información de los 3 países con mayor tasa de "casos positivos verus el número de pruebas realizadas".

```
top_3 <- filter(covid_top_10,
                Country_Region == "United Kingdom" |
                Country_Region == "United States" |
                Country_Region == "Turkey"
                )
top_3
```

```
## # A tibble: 3 x 5
##   Country_Region   tested positive  active hospitalized
##   <chr>             <int>    <int>   <int>        <int>
## 1 United States  17282363  1877179       0            0
## 2 Turkey          2031192   163941 2980960            0
## 3 United Kingdom  1473672   166909       0            0
```

Luego, definiremos `positive_tested_top_3_df` como el data frame que contiene la misma información que `top_3` con la columna `ratio` como extra, para lo cual utilizaremos las funciones `mutate()`, al generar la columna `ratio`, y `arrange()` y `desc()` para ordenar en orden decreciente los datos, considerando a esta nueva columna `ratio` como referencia.

```
positive_tested_top_3_df <- top_3 %>%
  mutate(ratio = positive/tested) %>%
  arrange(desc(ratio))
positive_tested_top_3_df
```

```
## # A tibble: 3 x 6
##   Country_Region   tested positive  active hospitalized  ratio
##   <chr>             <int>    <int>   <int>        <int>  <dbl>
## 1 United Kingdom  1473672   166909       0            0 0.113
## 2 United States  17282363  1877179       0            0 0.109
## 3 Turkey          2031192   163941 2980960            0 0.0807
```

Para cerrar el procesamiento de los datos, vamos a reunir todas las respuestas y conjunto de datos en una lista.

Comenzaremos creando la variable de tipo caracter llamada `question` que contenga la pregunta "¿Qué países han tenido el mayor número de casos positivos en comparación con el número de pruebas realizadas?" y complementaremos con el vector llamado `answer` que contendrá la respuesta.

```
question <- "¿Qué países han tenido el mayor número de casos positivos en comparación con el número de p
answer    <- c("Casos testeados positivos" = positive_tested_top_3)
```

Luego, crearemos una lista que contendrá los data frames trabajados anteriormente: `covid_df`, `covid_df_all_states`, `covid_df_all_states_daily` y `covid_top_10`.

```
datasets  <- list(covid_df,
                  covid_df_all_states,
                  covid_df_all_states_daily,
                  covid_top_10
                  )
matrices  <- list(positive_tested_top_3_df)
vectors   <- list(covid_df_cols_names, countries)
```

Estas listas, las reuniremos a su vez en la lista `data_structure_list`.

```
data_structure_list <- list(datasets, matrices, vectors)
```

Para cerrar el proceso, crearemos una lista que contenga las listas question, answer y `data_structure_list`, y mostraremos el contenido del segundo elemento de la lista.

```
covid_analysis_list <- list(question, answer, data_structure_list)
covid_analysis_list [2]
```

```
## [[1]]
## Casos testeados positivos.United Kingdom
##                                0.11326062
##  Casos testeados positivos.United States
##                                0.10861819
##         Casos testeados positivos.Turkey
##                                0.08071172
```

III. Conclusiones

Podemos evidenciar, finalmente, que los países que tuvieron mayor número de pruebas realizadas fueron: Estados unidos, Russia e Italia, en orden decreciente, mientras que los países tuvieron el mayor número de casos positivos respecto al número de pruebas realizadas, fueron: Reino Unido, Estados Unidos y Turkía, también en orden decreciente. No debemos olvidar que estos resultados son acotados únicamente al procesamiento de los datos presentes en covid19.csv (versión no actualizada), y bajo el filtrado de datos por "All States", por lo que no necesariamente son representativos de una realidad (cabe señalar que otros análisis enfocados en otros de los datos entregados se podrían realizar, idealmente con la base de datos actualizada).

A través del desarrollo de este proyecto de análisis podemos notar la importancia que tiene el orden de los procesos, ya que si bien los caminos para llegar a responder las preguntas objetivo son muchos, según el coding que se emplee, es crucial el respetar un orden lógico de prioridades, así como un constante monitoreo (visualización) de los resultados que se van obteniendo en cada paso, todo esto con la finalidad de evitar cesgos a la hora de realizar comparativas, así como los posibles errores en el código, y/o en el criterio de análisis utilizado.