Project 1

Skryabin Nikolay, Daniel Brock

2022-04-10

Contents

Background

The World Health Organization has recently employed a new data science initiative, *CSIT-165*, that uses data science to characterize pandemic diseases. *CSIT-165* disseminates data driven analyses to global decision makers.

CSIT-165 is a conglomerate comprised of two fabricated entities: *Global Health Union (GHU)* and *Private Diagnostic Laboratories (PDL)*. Your and your partner's role is to play a data scientist from one of these two entities.

Data

2019 Novel Coronavirus COVID-19 (2019-nCoV) Data Repository by John Hopkins CSSE

Data for 2019 Novel Coronavirus is operated by the John Hopkins University Center for Systems Science and Engineering (JHU CSSE). Data includes daily time series CSV summary tables, including confirmations, recoveries, and deaths. Country/region are countries/regions hat conform to World Health Organization (WHO). Lat and Long refer to coordinates references for the user. Date fields are stored in MM/DD/YYYY format.

GitHub Repository of the Project

group-project-1

```
# URLs for the confirmed and death cases files
URL_CONFIRMED <- paste0("https://raw.githubusercontent.com/CSSEGISandData/",
                "COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/",
                "time_series_covid19_confirmed_global.csv")


URL_DEATHS <- paste0("https://raw.githubusercontent.com/CSSEGISandData/",
                "COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/",
                "time_series_covid19_deaths_global.csv")

# Loads a file from URL and reads data from the file into data frame
getDataFrame <- function(url)
{
   # split the URL into list using "/" as delimiter
  url_list<-strsplit(url,split = "/")

  file_name <- sapply(url_list, tail, 1) # get the file name

  download.file(url, file_name) # download the file

  # create data frame from the file
  data_frame <- read.csv(file_name, stringsAsFactors = FALSE)

  data_frame # return it
}

# Get the data frames for deaths and confirmed cases
confirmed_df <- getDataFrame(URL_CONFIRMED)
deaths_df <- getDataFrame(URL_DEATHS)
```

Project Objectives

Objective 1

```
# Get max value of deaths on the first day
max.deaths.first.day.origin <- deaths_df[deaths_df[,5] ==
                                          max(deaths_df[,5]), 1:4]

# Get max value of confirmed cases on the first day
max.confirmed.first.day.origin <- confirmed_df[confirmed_df[,5] ==
                                                 max(confirmed_df[,5]), 1:4]

# Check if the province and state for max values of the first day are the same
if(max.confirmed.first.day.origin[1,1] == max.deaths.first.day.origin[1,1] &&
   max.confirmed.first.day.origin[1,2] == max.deaths.first.day.origin[1,2])
{
  cat(paste0("The origin of Covid 19 is: ", max.deaths.first.day.origin[1,1],
             " ",max.deaths.first.day.origin[1,2] ))
```

```
} else
{
  print("Couldn't find the origin")
}
```

```
## The origin of Covid 19 is: Hubei China
```

Objective 2

```
# Omit entries with 0 or NAs for coordinates in confirmed data frame
om_confirmed_df <- na.omit(confirmed_df[confirmed_df[,3:4]!=0,])

# Create reversed data frame and exclude first 4 columns
rev.confirmed_df <- rev(om_confirmed_df[,5:ncol(om_confirmed_df)])

# create a data frame with 4 first columns to hold result
res_df2 <- data.frame(om_confirmed_df[ ,1:4],stringsAsFactors = FALSE)

#  Finds last recent and previous columns for recent confirmed cases
getRecentColumns <- function(df,df1)
{
  for(i in 1:(ncol(df)-1)) # loop columns
  {
    current <- df[,i] # current column
    next1 <- df[,(i + 1)] # next column
    country_names <- om_confirmed_df$Country.Region # Country.Region column

    # check the conditions and return the columns.
    # Exclude the Olympics and Antarctica from the search
    ifelse(next1 == 0 & current > 0 & !grepl("Olympics",country_names) &
             !grepl("Antarctica",country_names),
           return(cbind(df[(i+1)],df[i])), next)
  }
}
# get result data frame
res_df2[,5:6] <- cbind(getRecentColumns(rev.confirmed_df, res_df2))
# get the last recent confirmed cases data frame
last_recent_confirmed_df <- res_df2[res_df2[,5]==0 & res_df2[,6]>0,]

cat(paste0("The most recent area to have a confirmed case is ",
           last_recent_confirmed_df[1,2], "."))
```

```
## The most recent area to have a confirmed case is Tonga.
```

Objective 3

```
# Get the library
library(geosphere)
```

```r
# Gets distance from origin in miles
getDistanceInMiles <- function(coordinate1, coordinate2)
{
  # call the distm function to calculate a distance in meters
  distance <- distm(coordinate1,
                    coordinate2,
                    fun=distVincentyEllipsoid)
  # Coefficient to convert meters to miles
  COEFFICIENT <- 0.000621371
  # Calculate miles
  miles_dist <- distance[1,1] * COEFFICIENT
  # return miles
  miles_dist
}

# Sorts the recent confirmed countries by distance from origin
sortCountriesByDistance <- function(origin_df, place_df)
{
   # a data frame to hold distance
  miles_dist <- data.frame(place_df[,1:2], stringsAsFactors = FALSE)

  for (i in 1:nrow(place_df))
  {
    # append a distance to the data frame
    miles_dist[,3] <-
      cbind(round(getDistanceInMiles(origin_df[,c('Long', 'Lat')],
                                     place_df[i,c('Long', 'Lat')]), 2))
  }
  colnames(miles_dist)[3] <- "Distance"   # name the last column
   # return the sorted by distance data frame
  miles_dist[order(miles_dist$Distance),]
}


# Prints a distance between the origin and last confirmed cases
printDistance <- function(origin_df, place_df)
{
  for (i in 1:nrow(place_df))
  {
    cat(paste0(i, ". ", place_df[i,1]," ",place_df[i,2], " is ",
               place_df[i,3], " miles away from ",
               origin_df[1,1],", ",
               origin_df[1,2],"."))
  }
}
# get the sorted recent confirmed countries
sorted_countries <- sortCountriesByDistance(max.deaths.first.day.origin,
                                            last_recent_confirmed_df)
# Print the result
printDistance(max.deaths.first.day.origin, sorted_countries)
```

```
## 1.  Tonga is 5999.04 miles away from Hubei, China.
```

Objective 4

```
# Omit entries with 0 or NAs for coordinates in deaths data frame
om_deaths_df <- na.omit(deaths_df[deaths_df[,3:4]!=0,])

# Calculate the risk scores for each place using the last columns
# from the deaths and confirmed data frames
risk_score <- 100 * (om_deaths_df[,ncol(om_deaths_df)] /
                       om_confirmed_df[,ncol(om_confirmed_df)])

# Calculate the global risk score
global_risk_score <- 100 * (sum(om_deaths_df[,ncol(om_deaths_df)]) /
                            sum(om_confirmed_df[,ncol(om_confirmed_df)]))

# Create a data frame to hold data with the confirmed and
# deaths last day, and risk score
risk_df <- cbind(om_confirmed_df[,c(1,2,ncol(om_confirmed_df))],
                 om_deaths_df[,ncol(om_deaths_df)], risk_score)

# Rename columns
colnames(risk_df)[3:4] <- c("Confirmed", "Deaths")

# omit values with 0 in Confirmed column
om_risk_df <- risk_df[risk_df$Confirmed != 0,]

# Sort by risk score in ascending and Confirmed in descending order
sorted_by_risk_ascend <-
  om_risk_df[order(om_risk_df$risk_score, -om_risk_df$Confirmed),]

# The first place in the data frame has the lowest risk
lowest_risk_place <- sorted_by_risk_ascend[1,]

# The last place in the data frame has the highest risk score
highest_risk_place <- sorted_by_risk_ascend[nrow(sorted_by_risk_ascend),]

cat(paste0("The place with the lowest risk is: ", lowest_risk_place[1,1]," ",
          lowest_risk_place[1,2],", with a risk score of: ",
          round(lowest_risk_place[1,5],2)), sep="\n")
```

```
## The place with the lowest risk is: Cook Islands New Zealand, with a risk score of: 0
```

```
cat(paste0("The place with the highest risk score is: ", highest_risk_place[1,1],
          highest_risk_place[1,2],", with a risk score of: ",
          round(highest_risk_place[1,5],2)), sep="\n")
```

```
## The place with the highest risk score is: Yemen, with a risk score of: 18.17
```

```
cat(paste0("The global risk score is: ", round(global_risk_score,2)), sep="\n")
```

```
## The global risk score is: 1.24
```

By seeing the drastic dispersion between the areas with the lowest and highest risk scores (0 and 18.7 respectively), it's apparent that most areas have relatively low risk scores, but some outlier areas have extremely high risk scores that are pulling the 'global risk score' up.

It's incredibly important to have succinct, yet descriptive values denoted to the various areas of the world in order to allow smarter and more informed decision making. With the knowledge of the risk for each area, it will heavily impact the amount of effort and money allocated to such areas in the effort to minimize sickness, deaths, and overall monetary costs.

One limitation with the 'risk score' used above is in how the value is calculated. By creating a percentage in terms of deaths per confirmed cases, it's telling us what areas individuals are most likely to die if they contract Covid-19, but it does leave a lot to be desired for a well encompassed single number to illustrate the 'risk' of the country or province. For example, if area A has a population of 1,000,000 people, and 3 individuals catch Covid-19, if all 3 die to it, they would have a 'risk score' of 100. While on the hand, area B, with a population of 1,000,000 could have 1,000,000 cases, with 500,000 deaths, giving them a 'risk score' of 50. Which area would you rather live in? The area where 3 people out of 1,000,000 caught and died from Covid-19, or the area where every individual (all 1,000,000) caught Covid-19, but only half (a staggering 500,000) died. Based on the 'risk score' metric, area B is twice as safe as area A, showing a massive weakness and abuse point of the 'risk score' metric.

Another limitation with the 'risk score' equation is the reliance on self-reported data. Each area is responsible for taking a reliable count of the number of cases of Covid-19, as well as the number of deaths caused by the disease. False positives or false negatives in either regard of Covid-19 cases and deaths, as well as the intentional falsification of values will result in incredibly inaccurate 'risk score' values, making the worth of the metric overall worse.

With enough poor data entered into the 'risk score' metric, it's value depreciates rapidly, until it becomes almost worthless when comparing areas to each other. If one area has been diligent in their proper diagnosing and reporting of cases and deaths while another area has not, comparisons between the two areas, like in the metric 'global risk score', are not valid or reliable.

Unless every area is doing an equally good job of diagnosing and reporting their Covid-19 cases and deaths, the metric will be wildly inaccurate and unfortunately, punish the individual areas that are honest and doing their due diligence, by having it appear that they are suffering worse from Covid-19 when compared to areas that are either ignorant of their true values, or intentionally dishonest about them.

Objective 5

```
# Gets the top 5 countries with the most cases
getTop5Countries <- function(df)
{
  # Get the country name and last day of the last cases data frame
  last_day_cases <- df[,c(2,ncol(df))]

  # Get the sum of the last day cases for each country
  countries_sums <- setNames(aggregate(last_day_cases[,2] ~ last_day_cases[,1],
  last_day_cases, FUN=sum),c("Province.State", "Sum"))

  # get 5 countries with the most cases
  top_5_countries <- (tail(countries_sums[order(countries_sums$Sum),], 5))

  # reverse the rows
  top_5_countries <- apply(top_5_countries, 2, rev)

  # return it
```

```
  top_5_countries
}

# Get the top 5 countries with confirmed cases
top_5_countries_confirmed <- getTop5Countries(om_confirmed_df)

# Get the top 5 countries with death cases
top_5_countries_deaths <- getTop5Countries(om_deaths_df)

kable(top_5_countries_confirmed, "pipe",
      col.names = c("Country Name", "Sum"),
      align = "lr", caption = "The Top 5 Countries with Confirmed Cases")
```

Table 1: The Top 5 Countries with Confirmed Cases

	Country Name	Sum
187	US	80399486
80	India	43035271
25	Brazil	30146769
63	France	27029271
67	Germany	22684849

```
kable(top_5_countries_deaths, "pipe",
      col.names = c("Country Name", "Sum"),
      align = "lr", caption = "The Top 5 Countries with Death Cases")
```

Table 2: The Top 5 Countries with Death Cases

	Country Name	Sum
187	US	985482
25	Brazil	661475
80	India	521685
144	Russia	364011
115	Mexico	323691

GitHub Log

```
git log --pretty=format:"%nSubject: %s%nAuthor: %aN%nDate: %aD%nBody: %b"
```

```
##
## Subject: Add link to the repository in the pdf file for the project 1
## Author: Nikolay Skryabin
## Date: Sun, 10 Apr 2022 17:29:06 -0700
## Body:
##
## Subject: Add pdf file for the project 1
## Author: Nikolay Skryabin
## Date: Sun, 10 Apr 2022 17:18:21 -0700
```

```
## Body:
##
## Subject: Added answers to objective 4
## Author: dbrock2379
## Date: Sun, 10 Apr 2022 16:43:32 -0700
## Body:
##
## Subject: Added tables for objective 5
## Author: dbrock2379
## Date: Sun, 10 Apr 2022 15:29:31 -0700
## Body:
##
## Subject: Change in objective 4 the reverse order for confirmed column. Change solution for objective
## Author: Nikolay Skryabin
## Date: Sun, 10 Apr 2022 10:22:57 -0700
## Body:
##
## Subject: Add objective 5 to the project
## Author: Nikolay Skryabin
## Date: Sat, 9 Apr 2022 19:28:45 -0700
## Body:
##
## Subject: Add objective 4.1
## Author: Nikolay Skryabin
## Date: Sat, 9 Apr 2022 12:31:05 -0700
## Body:
##
## Subject: Fix typo in objective 3
## Author: Nikolay Skryabin
## Date: Fri, 8 Apr 2022 19:44:11 -0700
## Body:
##
## Subject: Add change to ifelse to exclude the Olympics and Antarctica in objective 2 for the project1
## Author: Nikolay Skryabin
## Date: Fri, 8 Apr 2022 12:35:30 -0700
## Body:
##
## Subject: Add change to ifelse to exclude the Olympics and Antarctica in objective 2 for the project1
## Author: Nikolay Skryabin
## Date: Fri, 8 Apr 2022 12:31:43 -0700
## Body:
##
## Subject: Change to single loop and ifelse in objective 2 for the project1.Rmd file
## Author: Nikolay Skryabin
## Date: Thu, 7 Apr 2022 10:07:39 -0700
## Body:
##
## Subject: Add objective 2,3 to the project1.Rmd file
## Author: Nikolay Skryabin
## Date: Wed, 6 Apr 2022 10:23:35 -0700
## Body:
##
## Subject: Add project Rmd file with objective 1
## Author: Nikolay Skryabin
```

```
## Date: Tue, 29 Mar 2022 15:21:56 -0700
## Body:
##
## Subject: Merge branch 'main' of https://github.com/nick404s/group-project-1
## Author: dbrock2379
## Date: Sat, 26 Mar 2022 10:49:01 -0700
## Body:
##
## Subject: add time_series_covid19_confirmed_global.csv
## Author: dbrock2379
## Date: Sat, 26 Mar 2022 10:43:02 -0700
## Body:
##
## Subject: Delete time_series_covid19_confirmed_global.csv
## Author: NIKOLAY SKRYABIN
## Date: Sat, 26 Mar 2022 10:39:10 -0700
## Body: Deleted the file to download again by my partner
##
## Subject: Add csv files for confirmed and deaths
## Author: Nikolay Skryabin
## Date: Thu, 24 Mar 2022 10:34:39 -0700
## Body:
##
## Subject: Add names
## Author: Nikolay Skryabin
## Date: Thu, 24 Mar 2022 09:40:25 -0700
## Body:
##
## Subject: Initial commit
## Author: NIKOLAY SKRYABIN
## Date: Wed, 23 Mar 2022 19:14:37 -0700
## Body:
```