Project 1

Name: Kira Degelsmith Partner: Felix Lopez

2025-04-03

Contents

GitHub Repo Link

Felix and Kira Project 1 GitHub Repo

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)*.

For this project, *Kira Degelsmith* will act on behalf of *GHU* and *Felix Lopez* will act on behalf of *PDL*.

Data

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

```r
# Create data frames for deaths and confirmations from csv files
deaths_df <- readr::read_csv("time_series_covid19_deaths_global.csv")
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```r
confirmations_df <- readr::read_csv("time_series_covid19_confirmed_global.csv")
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

Project Objectives

Objective 1

```r
# Create variable to hold first date
first_date <- colnames(confirmations_df)[5]  # "1/22/20"

# Find the region with the most confirmed cases on the first date
max_confirmations <- max(confirmations_df[[first_date]], na.rm = TRUE)
origin_confirm <- confirmations_df %>%
  filter(!!sym(first_date) == max_confirmations) %>%
  select(`Province/State`, `Country/Region`)

# Find the region with the most deaths on the first date
max_deaths <- max(deaths_df[[first_date]], na.rm = TRUE)
origin_death <- deaths_df %>%
  filter(!!sym(first_date) == max_deaths) %>%
  select(`Province/State`, `Country/Region`)

# Convert to character strings for better comparison
origin_confirm_str <- paste(origin_confirm$`Province/State`, origin_confirm$`Country/Region`, sep=", ")
origin_death_str <- paste(origin_death$`Province/State`, origin_death$`Country/Region`, sep=", ")

# Check if both values match using an if statement
if (all(origin_confirm_str %in% origin_death_str)) {
  cat("The origin of COVID-19 is:", origin_confirm_str, sep=" ")
} else {
  cat("The origin is uncertain as the highest confirmations and deaths do not match exactly.")
}
```

```
## The origin of COVID-19 is: Hubei, China
```

Objective 2

```r
# Extract only the date columns (starting from the 5th column)
date_columns <- colnames(confirmations_df)[5:ncol(confirmations_df)]

# Initialize variables to track the most recent first case
latest_first_case_date <- as.Date("2020-01-01", format="%Y-%m-%d")
latest_region <- ""

# Loop through each row to find the first confirmed case
for (i in 1:nrow(confirmations_df)) {
  region_name <- paste(confirmations_df$`Province/State`[i],
                       confirmations_df$`Country/Region`[i], sep=", ")

  # Find the first date where cases > 0
  first_case_index <- which(confirmations_df[i, date_columns] > 0)[1]

  if (!is.na(first_case_index)) {
    first_case_date <- as.Date(date_columns[first_case_index], format="%m/%d/%y")

    # Check if this is the most recent first case
    if (first_case_date > latest_first_case_date) {
      latest_first_case_date <- first_case_date
      latest_region <- region_name
    }
  }
}

# Split latest_region into Province/State and Country/Region
latest_parts <- unlist(strsplit(latest_region, ", "))
province <- latest_parts[1]
country <- latest_parts[2]

# Create recent_confirm for use in Objective 3
recent_confirm <- confirmations_df %>%
  filter(`Province/State` == province, `Country/Region` == country)

# Print the result
cat("The most recent area to have its first confirmed case is: ",
    latest_region, " on ", as.character(latest_first_case_date)
    , ".", sep="", fill=TRUE)
```

```
## The most recent area to have its first confirmed case is:
## Pitcairn Islands, United Kingdom on 2022-07-20.
```

Objective 3

```r
library(geosphere) # Load geosphere package

# Extract coordinates for origin
origin_location <- confirmations_df %>%
  filter(`Province/State` == origin_confirm$`Province/State`,
         `Country/Region` == origin_confirm$`Country/Region`)

origin_lat <- origin_location$Lat
origin_long <- origin_location$Long

# Extract coordinates for the most recent confirmed case
recent_lat <- recent_confirm$Lat
recent_long <- recent_confirm$Long

# Compute distance in meters
distance_meters <- distm(c(origin_long, origin_lat), c(recent_long, recent_lat))
```

```
# Convert to miles
distance_miles <- distance_meters * 0.000621371

# Print formatted output
cat(recent_confirm$`Province/State`, recent_confirm$`Country/Region`, "is",
    round(distance_miles, 2), "miles away from", origin_confirm$`Province/State`,
    origin_confirm$`Country/Region`, sep=" ", fill=TRUE)
```

```
## Pitcairn Islands United Kingdom is 8746.97 miles away from Hubei China
```

Objective 4

```
# Create function to calculate risk scores
# Convert NaN values to 0, since they mean there were not any confirmations
risk_score <- function(deaths, confirmations)
{
  risk <- 100 * (deaths/confirmations)
  risk[is.nan(risk)] <- 0
  return(risk)
}

# Filter out cruise ships from the two data frames
deaths_no_cruise <- deaths_df[deaths_df$Lat != 0 & deaths_df$Long != 0
                              & !is.na(deaths_df$Lat)
                              & !is.na(deaths_df$Long),]
confirmations_no_cruise <- confirmations_df[confirmations_df$Lat != 0
                                            & confirmations_df$Long != 0
                                            & !is.na(confirmations_df$Lat)
                                            & !is.na(confirmations_df$Long),]

# Create a vector containing the most recent risk score for each area
current_deaths <- deaths_no_cruise[, ncol(deaths_no_cruise), drop=TRUE]
current_confirmations <- confirmations_no_cruise[, ncol(confirmations_no_cruise),
                                                 drop=TRUE]
current_risk_scores <- risk_score(current_deaths, current_confirmations)

# Find the index (which will correspond to the row) of the lowest risk score
# If more than one are the same, show the one with more confirmations
low_index <- which(current_risk_scores == min(current_risk_scores))
if(length(low_index) > 1)
{
  low_index <- which(current_confirmations == max(current_confirmations[low_index]))
}

# Find and display the area with the lowest risk score
cat("The area that currently has the lowest risk score is\n",
    confirmations_no_cruise[low_index,1,drop=TRUE], ", ",
    confirmations_no_cruise[low_index,2,drop=TRUE], ", with a score of ",
    round(current_risk_scores[low_index],2), "%.", sep = "", fill = TRUE)
```

```
## The area that currently has the lowest risk score is
## Jiangsu, China
## , with a score of 0%.
```

```r
# Find the index (which will refer to the row) of the highest risk score
# If more than one are the same, show the one with more confirmations
high_index <- which(current_risk_scores == max(current_risk_scores))
if(length(high_index) > 1)
{
  high_index <- which(current_confirmations == max(current_confirmations[high_index]))
}

# Find and display the area with the highest risk score
cat("The area that currently has the highest risk score is\n",
    confirmations_no_cruise[high_index,2,drop=TRUE], ", with a score of ",
    round(current_risk_scores[high_index],2), "%.", sep = "")
```

```
## The area that currently has the highest risk score is
## Korea, North, with a score of 600%.
```

```r
# Since the highest risk score is over 100%...
# Find the highest risk score which is less than or equal to 100%
risk_scores_adjusted <- ifelse(current_risk_scores > 100, NA, current_risk_scores)
high_index_adjusted <- which(risk_scores_adjusted == max(risk_scores_adjusted, na.rm=TRUE))
if(length(high_index_adjusted) > 1)
{
  high_index_adjusted <- which(current_confirmations == max(
    current_confirmations[high_index_adjusted], na.rm=TRUE))
}

# Find and display the area with the highest risk score <=100
cat("The area that currently has the highest risk score is\n",
    confirmations_no_cruise[high_index_adjusted,2,drop=TRUE],
    ", with a score of ", round(current_risk_scores[high_index_adjusted],2),
    "%.", sep = "")
```

```
## The area that currently has the highest risk score is
## Yemen, with a score of 18.07%.
```

```r
# Find global risk score
global_risk <- risk_score(sum(current_deaths), sum(current_confirmations))

# Compare global risk score to the highest and lowest risk areas
low_to_global <- global_risk - current_risk_scores[low_index]
high_to_global <- global_risk - current_risk_scores[high_index]
high_adj_to_global <- global_risk - current_risk_scores[high_index_adjusted]

# Display comparisons of high, low, and global risk scores
cat("The global risk score is currently ", round(global_risk,2),
    "%. The area that currently has",
    "the lowest risk score has a score that is ", round(low_to_global,2),
    "% lower", "than the global risk score. The area that currently has",
    "the highest risk score (<= 100%) has a score that is\n",
    abs(round(high_adj_to_global,2)), "% higher than the global ",
    "risk score.", sep = "", fill = TRUE)
```

```
## The global risk score is currently 1.01%. The area that currently has
## the lowest risk score has a score that is 1.01% lower
## than the global risk score. The area that currently has
## the highest risk score (<= 100%) has a score that is
## 17.07
## % higher than the global risk score.
```

Objective 4 Responses

- Calculating metrics like risk scores for different areas of the world can be helpful for many reasons. Knowing the COVID-19 risk score for a certain area of the world can help researchers determine where risk is highest and lowest, and where resources may be most effective and beneficial if they are deciding on distribution locations. Additionally, tracking risk scores over time for different areas of the world can provide information about trends in risk scores over the years and also during different times of the year and how that varies from location to location.
- One limitation from calculating risk scores is that the risk score assumes that valid and accurate data has been reported. An example of this limitation is evident in the code chunk above. The highest risk score without any adjustment came out to be 600% in North Korea, since the most recent North Korean data shows 6 deaths but only 1 confirmed case. Since the risk score is over 100%, it is possible to decide that the data for North Korea is erroneous in some way, so to find the highest risk rate, you go to the highest rate that is less than or equal to 100%. This example with North Korea made it obvious to see that there was something going wrong with the data, but if the outcome didn't surpass 100%, it would be harder to see an error similar to this. Incorrect, incomplete, or otherwise erroneous data can cause problems when calculating metrics like risk scores, and it is important to take that into account when running analyses on data sets.

Objective 5

```r
# Create a function to create a new data frame with top 5 countries and sums for
# deaths and confirmations
top_5_country_sums <- function(df)
{
  # Find all unique countries from the data set
  countries <- unique(df[, "Country/Region", drop=TRUE])

  # Create empty list to hold sum for each country
  sum_list <- list()

  # For loop to add death sum for each country to the list
  for(country in countries)
  {
    sum_list[[country]] <- sum(df[df[,2,drop=TRUE]==country,
                                  grep("[0-9]{1,2}/[0-9]{1,2}/[0-9]{1,2}",
                                       colnames(df))])
  }

  # Create new data frame with countries and sums
  sums_df <- data.frame("Country"=countries, "Total"=unlist(sum_list))

  # Return df with only the top 5 countries
  return(sums_df[order(sums_df$Total, decreasing=TRUE),][1:5,])
}

# Create tables for the top 5 countries for deaths and confirmations
top_5_death <- top_5_country_sums(deaths_df)
knitr::kable(top_5_death, caption = "Total Deaths by Country: Top 5 Countries",
             row.names = FALSE)
```

Table 1: Total Deaths by Country: Top 5 Countries

Country	Total
US	713877215
Brazil	488181000
India	364921237
Mexico	241085189
Russia	220983590

```
top_5_confirmations <- top_5_country_sums(confirmations_df)
knitr::kable(top_5_confirmations,
             caption = "Total Confirmations by Country: Top 5 Countries",
             row.names = FALSE)
```

Table 2: Total Confirmations by Country: Top 5 Countries

Country	Total
US	53813184406
India	29131119694
Brazil	21182690594
France	16105911886
Germany	13686043720

GitHub Log

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

```
##
## Subject: Delete deaths_analysis.Rmd (code put into .R files instead)
## Author: kadegel
## Date: Thu, 3 Apr 2025 17:08:58 -0700
## Body:
##
## Subject: Delete deaths_analysis.html
## Author: kadegel
## Date: Thu, 3 Apr 2025 17:08:39 -0700
## Body:
##
## Subject: Merge branch 'main' of https://github.com/kadegel/Felix-Kira-Project-1
## Author: FEL1HIL
## Date: Thu, 3 Apr 2025 16:59:58 -0700
## Body:
##
## Subject:  add Objective 2 and 3 and add them to write up
## Author: FEL1HIL
## Date: Thu, 3 Apr 2025 16:53:37 -0700
## Body:
##
## Subject: edit some formatting for writeup.Rmd
## Author: kadegel
## Date: Thu, 3 Apr 2025 16:43:21 -0700
## Body:
##
## Subject: add objective 1 file and add objective 1 in write up file.
## Author: FEL1HIL
## Date: Thu, 3 Apr 2025 16:36:24 -0700
## Body:
##
## Subject: edit formatting for objective 4 in writeup
## Author: kadegel
## Date: Wed, 2 Apr 2025 14:23:30 -0700
## Body:
##
## Subject: create R script file for objective 5 and add obj 5 to writeup.Rmd
## Author: kadegel
## Date: Wed, 2 Apr 2025 12:53:39 -0700
## Body:
##
## Subject: edited death_conf_df.R to contain the variables after accidental removal at previous commit
## Author: kadegel
## Date: Wed, 2 Apr 2025 11:41:25 -0700
## Body:
##
## Subject: Updated death_conf_df.R and added deaths_analysis files
## Author: FEL1HIL
## Date: Tue, 1 Apr 2025 19:06:25 -0700
## Body:
##
## Subject: create objective 4 R script and add objective 4 code and responses to the writeup
## Author: kadegel
## Date: Fri, 28 Mar 2025 18:09:11 -0700
## Body:
##
## Subject: create r script file to load the scv data into data frames and add that code to the writeup
## Author: kadegel
## Date: Fri, 28 Mar 2025 16:47:20 -0700
## Body:
##
## Subject: add global confiormations data
## Author: FEL1HIL
## Date: Thu, 27 Mar 2025 19:32:00 -0700
## Body:
##
## Subject: add deaths csv file
## Author: kadegel
## Date: Thu, 27 Mar 2025 19:05:07 -0700
## Body:
##
## Subject: add recovered csv file
## Author: kadegel
## Date: Thu, 27 Mar 2025 18:45:36 -0700
## Body:
##
## Subject: update readme file to include chosen team member assignments
## Author: kadegel
## Date: Thu, 27 Mar 2025 18:37:16 -0700
## Body:
##
## Subject: add writeup.Rmd file, a drafted template version
## Author: kadegel
## Date: Thu, 27 Mar 2025 17:36:22 -0700
## Body:
##
## Subject: add team member names to readme file
## Author: kadegel
## Date: Thu, 27 Mar 2025 17:28:02 -0700
## Body:
##
## Subject: Initial commit
## Author: kadegel
## Date: Thu, 27 Mar 2025 17:22:56 -0700
## Body:
```