Project 1

Name:
Partner:

2023-04-09

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.

Project Objectives

Objective 1: What was the origin country of the COVID-19 outbreak?

```
#load data
confirmed_cases <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/
covid_deaths <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/css

cases_df <- read.csv(confirmed_cases, header = TRUE, na.strings = c("", " "))
deaths_df <- read.csv(covid_deaths, header = TRUE, na.strings = c("", " "))

#segment first day of COVID data
data_cases <- dplyr::select(cases_df, Province.State, Country.Region, X1.22.20);
data_deaths <- dplyr::select(deaths_df, Province.State, Country.Region, X1.22.20)

# Filter for the first day and select relevant columns
first_day_cases <- cases_df %>%
  filter(X1.22.20 != 0) %>%
  select(Province.State, Country.Region, X1.22.20)
first_day_deaths <- deaths_df %>%
  filter(X1.22.20 != 0) %>%
  select(Province.State, Country.Region, X1.22.20)

# Identify the area with the highest confirmed cases and highest deaths
max_cases <- first_day_cases %>%
  filter(X1.22.20 == max(X1.22.20)) %>%
  pull(Province.State)
max_deaths <- first_day_deaths %>%
  filter(X1.22.20 == max(X1.22.20)) %>%
  pull(Province.State)

# Determine if the area(s) identified is the origin of the outbreak
if(max_cases == max_deaths) {
  output <- paste("The origin of the COVID-19 outbreak was likely", max_cases)
  print(output)
}
```

```
## [1] "The origin of the COVID-19 outbreak was likely Hubei"
```

Objective 2: Where is the most recent area to have a first confirmed case?

```
# iterates through each (date-containing) column
for(date_column in (5:ncol(cases_df))){

  # iterates through each row (case count) for that specific date
  for(x in (1:length(cases_df[,date_column]))){ # subsets the column for a single date
    if(cases_df[x, date_column] == 1 & cases_df[x, date_column-1] == 0){ # checks if there is a new cas
      newest_case <- cases_df[x, 2] # updates variable with the corresponding country name (column 2)
    }
  }
}
```

```
cat("The most recent area to have a first confirmed case is", newest_case)
```

```
## The most recent area to have a first confirmed case is Korea, North
```

Objective 3: How far away are the areas from objective 2 from where the first confirmed case(s) occurred?

```
# assigns correct values to origin country and recent country variables using output from ob1 and ob2
recent_region <- newest_case
origin_city <- max_cases
origin_country <- "China"

# subsets the lat and long values, using the row number that corresponds to the origin country
origin_lat = cases_df[which(cases_df$Province.State == origin_city), 3]
origin_long = cases_df[which(cases_df$Province.State == origin_city), 4]
# creates a list of (longitude, latitude) -> the correct input format for distm
origin_coordinates <- c(origin_long, origin_lat)

# subsets the lat and long values, using the row number that corresponds to the most recent country
recent_lat = cases_df[which(cases_df$Country.Region == recent_region), 3]
recent_long = cases_df[which(cases_df$Country.Region == recent_region), 4]
# creates a list of (longitude, latitude) -> the correct input format for distm
recent_coordinates <- c(recent_long, recent_lat)

# calculates distance between coordinates using distm
distance = distm(origin_coordinates, recent_coordinates, fun=distGeo)
# converts dist in meters to miles using conversion factor
miles_distance = distance/1609

# prints distance between two locations in a sentence using values from above
sprintf("%s is %f miles away from %s, %s", recent_region, miles_distance, origin_city, origin_country)
```

```
## [1] "Korea, North is 1070.926759 miles away from Hubei, China"
```

Objective 4

```
deaths_df <- deaths_df[!(is.na(deaths_df$Lat) | deaths_df$Lat == 0),]

# Extract the most recent date
most_recent_date <- tail(colnames(cases_df), 1)
print(most_recent_date)
```

```
## [1] "X3.9.23"
```

```
# Extract the columns for the most recent date
confirmed_cases_latest<- dplyr::select(cases_df, Province.State, Country.Region, ncol(cases_df));
deaths_latest  <- dplyr::select(deaths_df, Province.State, Country.Region, ncol(deaths_df))
```

```r
#merge the data sets into one
merged_df <- merge(confirmed_cases_latest, deaths_latest, by = c("Province.State", "Country.Region"))

# Calculate risk scores for the most recent date
confirmed_cases_latest <- "X3.9.23.x"
deaths_latest <- "X3.9.23.y"
risk_scores_latest <- merged_df[,deaths_latest] / merged_df[,confirmed_cases_latest] * 100

# Add risk scores to the end of the dataframe as a new column
merged_df <- mutate(merged_df, risk_score = risk_scores_latest)
merged_df[is.na(merged_df)]<- 0
head(merged_df)
```

```
##                 Province.State Country.Region X3.9.23.x X3.9.23.y risk_score
## 1                      Alberta         Canada    629269      5622 0.89341760
## 2                    Anguilla United Kingdom      3904        12 0.30737705
## 3                       Anhui          China      2275         7 0.30769231
## 4                       Aruba    Netherlands     44044       282 0.64026882
## 5 Australian Capital Territory      Australia    232974       228 0.09786500
## 6                      Beijing          China     40774        20 0.04905087
```

```r
lowest_risk_highest_cases <- merged_df %>%
  filter(risk_score == min(risk_score[!is.na(risk_score)])) %>%
  filter(ncol(cases_df) == max(ncol(cases_df)))

# Print the lowest risk table
print(lowest_risk_highest_cases)
```

```
##                                 Province.State       Country.Region X3.9.23.x
## 1                             Channel Islands       United Kingdom         0
## 2                  Falkland Islands (Malvinas)       United Kingdom      1930
## 3                                     Jiangsu                China      5075
## 4                                           0           Antarctica        11
## 5                                           0             Holy See        29
## 6                                           0 Summer Olympics 2020       865
## 7                                           0               Tuvalu      2805
## 8                                           0 Winter Olympics 2022       535
## 9                                     Ningxia                China      1276
## 10                                       Niue          New Zealand       792
## 11                           Pitcairn Islands       United Kingdom         4
## 12                                    Qinghai                China       782
## 13 Saint Helena, Ascension and Tristan da Cunha       United Kingdom      2166
## 14                                      Tibet                China      1647
##    X3.9.23.y risk_score
## 1          0          0
## 2          0          0
## 3          0          0
## 4          0          0
## 5          0          0
## 6          0          0
## 7          0          0
## 8          0          0
```

```
## 9          0          0
## 10         0          0
## 11         0          0
## 12         0          0
## 13         0          0
## 14         0          0
```

```
lowest_risk_province <- lowest_risk_highest_cases %>%
  filter(X3.9.23.x == max(X3.9.23.x)) %>%
  pull(Province.State)
print(lowest_risk_province)
```

```
## [1] "Jiangsu"
```

```
#Now for the Highest Risk region
# Find the highest risk regions
highest_risk_highest_cases <- merged_df %>%
  filter(risk_score == max(risk_score[!is.na(risk_score)])) %>%
  filter(ncol(cases_df) == max(ncol(cases_df)))

# Print the highest risk table
print(highest_risk_highest_cases)
```

```
##   Province.State Country.Region X3.9.23.x X3.9.23.y risk_score
## 1              0   Korea, North         1         6        600
```

```
highest_risk_province <- highest_risk_highest_cases %>%
  filter(X3.9.23.x == max(X3.9.23.x)) %>%
  pull(Country.Region)
print(highest_risk_province)
```

```
## [1] "Korea, North"
```

```
#Global risk

global_risk <- mean(merged_df$risk_score)
print(global_risk)
```

```
## [1] 3.209061
```

Risk Comparisons

Lowest Risk: 0%

Highest Risk: 600%

Global Risk: 3.2%

Objective 5

```r
# creates a list of all countries by subsetting the country column of the dataset
countries <- cases_df$Country.Region
# removes duplicates (due to multiple provinces)
countries <- unique(countries)

deaths = 0
cases = 0

# initialization of empty lists
country_cases <- c()
country_deaths <- c()

# iterates through each unique country
for(country in countries){
  # creates a list of the indexes of every time the country appears
  country_duplicates <- which(cases_df$Country.Region == country)
  # iterates through each of the duplicate indexes (each province of that country)
  for(dup in country_duplicates){
    # adds the cases for that province to the total count for the country
    cases <- cases + cases_df[dup, 1147]
  }
  # appends the total case count for that country to the country_cases list
  country_cases <- append(country_cases, cases)
  # resets case count to 0 before moving on to the next country in the list
  cases = 0
}

# repeats same procedure, but with deaths data set
for(country in countries){
  country_duplicates <- which(deaths_df$Country.Region == country)
  for(dup in country_duplicates){
      deaths <- deaths + deaths_df[dup, 1147]
  }
  country_deaths <- append(country_deaths, deaths)
  deaths = 0
}

# creates a data frame with countries, their respective cases, and respective deaths
overview <- data.frame(countries, country_cases, country_deaths)

# uses arrange function to create two different data frames, each sorted in descending order of case/de
casewise <- arrange(overview, -country_cases)
deathwise <- arrange(overview, -country_deaths)

# creates new data frames that subset only the top 5
top_case <- casewise[1:6,]
top_death <- deathwise[1:6,]

# uses kable to display the data frames as visual tables
kable(top_case)
```

countries	country_cases	country_deaths
US	103802702	1123836
India	44690738	530779
France	39866718	166176
Germany	38249060	168935
Brazil	37076053	699276
Japan	33320438	72997

```
kable(top_death)
```

countries	country_cases	country_deaths
US	103802702	1123836
Brazil	37076053	699276
India	44690738	530779
Russia	22075858	388478
Mexico	7483444	333188
United Kingdom	24658705	220721

GitHub Log

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

```
##
## Subject: formatting
## Author: Morgan
## Date: Sun, 9 Apr 2023 23:38:15 -0700
## Body:
##
## Subject: ob4 global risk added
## Author: Morgan
## Date: Sun, 9 Apr 2023 23:34:44 -0700
## Body:
##
## Subject: added global risk
## Author: Morgan
## Date: Sun, 9 Apr 2023 23:29:00 -0700
## Body:
##
## Subject: highest risk country
## Author: Morgan
## Date: Sun, 9 Apr 2023 23:19:33 -0700
## Body:
##
## Subject: manually resolved merge changes again????
## Author: PreenaM
## Date: Sun, 9 Apr 2023 23:18:43 -0700
## Body:
##
## Subject: fixed merge error? i don't even know
## Author: PreenaM
## Date: Sun, 9 Apr 2023 23:14:45 -0700
```

```
## Body:
##
## Subject: lowest risk province
## Author: Morgan
## Date: Sun, 9 Apr 2023 23:13:37 -0700
## Body:
##
## Subject: lowest risk highest cases table
## Author: Morgan
## Date: Sun, 9 Apr 2023 23:07:48 -0700
## Body:
##
## Subject: edits
## Author: Morgan
## Date: Sun, 9 Apr 2023 22:53:15 -0700
## Body:
##
## Subject: new merged table for risk scores
## Author: Morgan
## Date: Sun, 9 Apr 2023 22:51:17 -0700
## Body:
##
## Subject: resolving merge conflict hopefully?
## Author: PreenaM
## Date: Sun, 9 Apr 2023 22:23:19 -0700
## Body:
##
## Subject: generated new reports
## Author: PreenaM
## Date: Sun, 9 Apr 2023 22:11:33 -0700
## Body:
##
## Subject: test
## Author: Morgan
## Date: Sun, 9 Apr 2023 22:08:15 -0700
## Body:
##
## Subject: Merge branch 'main' of https://github.com/PreenaM/CSIT-Group-Project-1
## Author: Morgan
## Date: Sun, 9 Apr 2023 22:06:44 -0700
## Body:
##
## Subject: objective 4 stuck spot
## Author: Morgan
## Date: Sun, 9 Apr 2023 22:06:26 -0700
## Body:
##
## Subject: added comments for ob5
## Author: PreenaM
## Date: Sun, 9 Apr 2023 21:58:52 -0700
## Body:
##
## Subject: added comments for ob3
## Author: PreenaM
```

```
## Date: Sun, 9 Apr 2023 21:52:56 -0700
## Body:
##
## Subject: completed Objective 5 & displayed both tables successfully, but comments pending
## Author: PreenaM
## Date: Sun, 9 Apr 2023 21:44:11 -0700
## Body:
##
## Subject: updated ob1 to province instead of country
## Author: Morgan
## Date: Sun, 9 Apr 2023 20:10:43 -0700
## Body:
##
## Subject: Merge branch 'main' of https://github.com/PreenaM/CSIT-Group-Project-1
## Author: Morgan
## Date: Sun, 9 Apr 2023 19:59:18 -0700
## Body:
##
## Subject: latest work on Objective 4 unfinished
## Author: Morgan
## Date: Sun, 9 Apr 2023 19:58:52 -0700
## Body:
##
## Subject: Merge branch 'main' of https://github.com/PreenaM/CSIT-Group-Project-1
## Author: PreenaM
## Date: Sun, 9 Apr 2023 19:53:29 -0700
## Body:
##
## Subject: completed Objective 3 with hard-coded values for recent region and origin region
## Author: PreenaM
## Date: Sun, 9 Apr 2023 19:53:23 -0700
## Body:
##
## Subject: added updated objective 1
## Author: Morgan
## Date: Sun, 9 Apr 2023 18:57:00 -0700
## Body:
##
## Subject: Completed Objective 2 using confirmed cases df, added comments
## Author: PreenaM
## Date: Sun, 9 Apr 2023 17:00:51 -0700
## Body:
##
## Subject: adding *Morgan's* progress on Objective 1 from previous repo
## Author: PreenaM
## Date: Sun, 9 Apr 2023 11:44:24 -0700
## Body:
##
## Subject: wget CSV for deaths (GHU)
## Author: PreenaM
## Date: Sun, 9 Apr 2023 11:40:55 -0700
## Body:
##
## Subject: wget CSV file for confirmed cases (PDL)
```

```
## Author: PreenaM
## Date: Sun, 9 Apr 2023 11:40:37 -0700
## Body:
##
## Subject: Added template
## Author: PreenaM
## Date: Sun, 9 Apr 2023 10:55:49 -0700
## Body:
##
## Subject: Updated README with team member names
## Author: PreenaM
## Date: Sun, 9 Apr 2023 10:40:48 -0700
## Body:
##
## Subject: Initial commit
## Author: PreenaM
## Date: Sun, 9 Apr 2023 10:34:23 -0700
## Body:
```