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)
    }
  }
}
```

```r
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

```r
recent_region <- newest_case
origin_city <- max_cases
origin_country <- "China"

origin_lat = cases_df[which(cases_df$Province.State == origin_city), 3]
origin_long = cases_df[which(cases_df$Province.State == origin_city), 4]
origin_coordinates <- c(origin_long, origin_lat)


recent_lat = cases_df[which(cases_df$Country.Region == recent_region), 3]
recent_long = cases_df[which(cases_df$Country.Region == recent_region), 4]
recent_coordinates <- c(recent_long, recent_lat)

distance = distm(origin_coordinates, recent_coordinates, fun=distGeo)
miles_distance = distance/1609

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

```r
#filter out cruise ship data from the data sets
cases_df <- cases_df[!(is.na(cases_df$Lat) | cases_df$Lat == 0),]
deaths_df <- deaths_df[!(is.na(deaths_df$Lat) | deaths_df$Lat == 0),]

# Calculate risk scores
risk_scores <- deaths_df[, -c(1:4)] / cases_df[, -c(1:4)] * 100
risk_scores <- as.numeric(as.matrix(risk_scores))


# Find area with lowest risk score and most confirmations

lowest_risk <- which.min(risk_scores)
most_confirmations <- which.max(cases_df[, ncol(cases_df)])

#paste("The area of the world with the lowest risk score is", colnames(risk_scores)[lowest_risk],
#      "with a risk score of", risk_scores[most_confirmations, lowest_risk],
#      "and the most confirmations of", cases_df[most_confirmations, ncol(cases_df)])
```

Objective 4.1

Objective 4.2

Objective 5

```
countries <- cases_df$Country.Region
countries <- unique(countries)

deaths = 0
cases = 0

country_cases <- c()
country_deaths <- c()

for(country in countries){
  country_duplicates <- which(cases_df$Country.Region == country)
  for(dup in country_duplicates){
      cases <- cases + cases_df[dup, 1147]
  }
  country_cases <- append(country_cases, cases)
  cases = 0
}

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
}

overview <- data.frame(countries, country_cases, country_deaths)
casewise <- arrange(overview, -country_cases)
deathwise <- arrange(overview, -country_deaths)

top_case <- casewise[1:6,]
top_death <- deathwise[1:6,]

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: 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:
```