Project 1

Name: Wesley Newcomb Partner: Manuel Ricardo Vargas and Diva Medina Camp

2023-04-06

## 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](https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series) 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.

For this project, we will use global data sets for COVID-19 associated confirmations and deaths.

In order to download these files without cloning the whole repository, enter one of two commands in your terminal depending on your machine:

Windows wget <https://address.to.data/goes/here.csv> Mac curl <https://address.to.data/goes/here.csv> -O

### Instructions

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. Discuss with your partner to decide who will be part of GHU and PDL.

### Getting Started

One project member per group must create a new repository on GitHub. Initialize this repository with a readme.md file that lists each member of the group. If your group decides to collaborate using a centralized workflow (recommended), then the project member that created the repository must declare their partners as collaborators in GitHub. Each project member will clone this repository onto their machine using RStudio. In RStudio, create a project from version control with GitHub using the HTTP address of the repository created by project member.

All project members must first contribute to analyses by uploading data sets respective to the entity they belong to in the CSIT-165 initiative. GHU project members provide time series data counting COVID-19 related recoveries and and deaths. PDL project members provide time series data counting COVID-19 related confirmations.

## Project Objectives

This project will encompass many of the lessons we have learned throughout the course. RMarkdown files must be written such that each time you render the document it will download the necessary data sets for analysis. Please render the RMarkdown file the day it is due to reflect the most recent data sets. With this added functionality, your code must be able to analyze the datasets regardless of the date you render your document.

### Objective 1

CSIT-165’s first objective is to determine where COVID-19 originated from. Predict where the origin started based on the area with the greatest number of confirmations and deaths on the first recorded day in the data set. Show this is the origin using an if statement.

confirmed\_df<-read.csv("data/time\_series\_covid19\_confirmed\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
deaths\_df<-read.csv("data/time\_series\_covid19\_deaths\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
recovered\_df<-read.csv("data/time\_series\_covid19\_recovered\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
  
## Confirmed Dataset  
confirmed\_ordered<-select(arrange(confirmed\_df, -X1.22.20),   
 Province.State, Country.Region, X1.22.20)  
cat("Confirmed Dataset ")

## Confirmed Dataset

head(confirmed\_ordered)

## Province.State Country.Region X1.22.20  
## 1 Hubei China 444  
## 2 Guangdong China 26  
## 3 Beijing China 14  
## 4 Zhejiang China 10  
## 5 Shanghai China 9  
## 6 Chongqing China 6

## Deaths Dataset  
death\_ordered<-select(arrange(deaths\_df, -X1.22.20),   
 Province.State, Country.Region, X1.22.20)  
cat("Deaths Dataset ")

## Deaths Dataset

head(death\_ordered)

## Province.State Country.Region X1.22.20  
## 1 Hubei China 17  
## 2 Afghanistan 0  
## 3 Albania 0  
## 4 Algeria 0  
## 5 Andorra 0  
## 6 Angola 0

## Recovered Dataset  
recovered\_ordered<-select(arrange(recovered\_df, -X1.22.20),   
 Province.State, Country.Region, X1.22.20)  
cat("Recovered Dataset \n")

## Recovered Dataset

head(recovered\_ordered)

## Province.State Country.Region X1.22.20  
## 1 Hubei China 28  
## 2 Thailand 2  
## 3 Afghanistan 0  
## 4 Albania 0  
## 5 Algeria 0  
## 6 Andorra 0

cat(confirmed\_ordered[1,1], ", ", confirmed\_ordered[1,2],   
 " has the most confirmed cases on the first day. \n",   
 death\_ordered[1,1], ", ", death\_ordered[1,2],   
 " has the most deaths from the virus on the first day. \n",  
 recovered\_ordered[1,1], ", ", recovered\_ordered[1,2],   
 " has the most recovered cases from the virus on the first day. \n",   
 sep="")

## Hubei, China has the most confirmed cases on the first day.   
## Hubei, China has the most deaths from the virus on the first day.   
## Hubei, China has the most recovered cases from the virus on the first day.

if (confirmed\_ordered[1,1] == death\_ordered[1,1] &&   
 confirmed\_ordered[1,1] == recovered\_ordered[1,1] &&   
 death\_ordered[1,1] == recovered\_ordered[1,1])  
{  
 cat(confirmed\_ordered[1,1], ", ", confirmed\_ordered[1,2],   
 " is the most likely origin of the virus. \n", sep="")  
}

## Hubei, China is the most likely origin of the virus.

Based on the data above: Hubei, China is the most likely origin of the virus. This is based on Hubei, China having by far the most confirmed cases and is the only region with any recoveries or deaths on the first day of data avaliable. Also, all other regions that have confirmed cases on the first day are regions that are near Hubei. The conditional statement also proves that Hubei is the most likely origin of the virus because it shows that the place with the most deaths, recovered, and confirmed cases on the first day is Hubei.

### Objective 2

Where is the most recent area to have a first confirmed case? To do this, you will need to use a for loop, if statement, and subsets.

confirmed\_df<-read.csv("data/time\_series\_covid19\_confirmed\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
recent\_df<-arrange(confirmed\_df[confirmed\_df[,ncol(confirmed\_df)-1] == 0   
 & confirmed\_df[,ncol(confirmed\_df)] > 0,])  
  
i<-0  
# If there are no new cases today loop back to find most recent region  
# to have new cases  
if (nrow(recent\_df) == 0) {  
 while (nrow(recent\_df) == 0) {  
 i<-i+1  
 recent\_df<-arrange(confirmed\_df[confirmed\_df[,ncol(confirmed\_df)-1-i] == 0   
 & confirmed\_df[,ncol(confirmed\_df)-i] > 0,])  
 }  
}  
  
head(select(recent\_df, Province.State, Country.Region, ncol(confirmed\_df)-1-i,   
 ncol(confirmed\_df)-i))

## Province.State Country.Region X7.19.22 X7.20.22  
## 1 Pitcairn Islands United Kingdom 0 4

# Vector is small enough that loop is reasonable  
for(i in 1:nrow(recent\_df))  
{  
 if (recent\_df[i,1] == "") {  
 cat(recent\_df[i,2], "has recently had their first confirmed case. \n")  
 } else {  
 if (recent\_df[i,2] == "") {  
 cat(recent\_df[i,1], "has recently had their first confirmed case. \n")  
 } else {  
 cat(recent\_df[i,1], ", ", recent\_df[i,2],   
 " has recently had their first confirmed case. \n", sep="")  
 }  
 }  
}

## Pitcairn Islands, United Kingdom has recently had their first confirmed case.

Most recent territories were found by going through the data and selecting the countries that did not have cases before yesterday and had their first cases today. If there are no regions meeting this check, then each previous day is looked at until there are regions found with new cases.

### Objective 3

How far away are the areas from objective 2 from where the first confirmed case(s) occurred? Please provide answer(s) in terms of miles. Use the function distm from the R package geosphere to calculate the distance between two coordinates in meters (geosphere::distm). You will need to convert the value returned by distm from meters to miles (this conversion is simple and can be found online). Please use a table or printed statement to describe what Province/State and Country/Region first confirmed cases occurred as well as the distance (in miles) away from the origin. Please print the following: {recent region} is {distance in miles} away from {origin city, origin country}.

origin\_df<-arrange(confirmed\_df, -X1.22.20)[1,]  
confirmed\_df<-read.csv("data/time\_series\_covid19\_confirmed\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
recent\_df<-arrange(confirmed\_df[confirmed\_df[,ncol(confirmed\_df)-1] == 0   
 & confirmed\_df[,ncol(confirmed\_df)] > 0,])  
i<-0  
  
# If there are no new cases today loop back to find most recent region  
# to have new cases  
if (nrow(recent\_df) == 0) {  
 while (nrow(recent\_df) == 0) {  
 i<-i+1  
 recent\_df<-arrange(confirmed\_df[confirmed\_df[,ncol(confirmed\_df)-1-i] == 0   
 & confirmed\_df[,ncol(confirmed\_df)-i] > 0,])  
 }  
}  
  
# Compute distances from origin  
distances<-distm(select(recent\_df, Long, Lat), select(origin\_df, Long, Lat))  
  
# Convert from m to miles  
distances<-distances \* 0.00062137  
  
# Add distance from origin to dataframe and sort by distance  
recent\_df$distance<-distances[,1]  
recent\_df<-arrange(recent\_df, distance)  
  
head(select(recent\_df, Province.State, Country.Region, Lat, Long, distance))

## Province.State Country.Region Lat Long distance  
## 1 Pitcairn Islands United Kingdom -24.3768 -128.3242 8746.96

# Vector is small enough that loop is reasonable  
for (i in 1:nrow(recent\_df)) {  
 city<-recent\_df[i, "Province.State"]  
   
 # If there is no city use country  
 if (city == "") {  
 city<-recent\_df[i, "Country.Region"]  
 }  
   
 cat(city, "is", recent\_df[i, "distance"],   
 "miles away from the virus origin in",   
 paste0(origin\_df[1, "Province.State"], ","),   
 paste0(origin\_df[1, "Country.Region"], "."), "\n")  
}

## Pitcairn Islands is 8746.96 miles away from the virus origin in Hubei, China.

### Objective 4

CSIT-165 characterizes diseases using risk scores. Risk scores are calculated as the ratio of deaths to confirmations, that is Riskscore=100×deathsconfirmations . Risk scores equal to 100 indicate the highest risk while risk scores equal to 0 indicate the lowest risk. Areas are characterized as being especially vulnerable to loss if they have higher risk scores. For this assignment, exclude cruise ships (hint: they have lat and long coordinates of 0 or NA in this data set, filter this out before calculating risk scores).

Which area of the world currently has the lowest risk score (if more than one, display the one with the most confirmations)? Which area of the world currently has the highest risk score (if more than one, display the one with the most confirmations)? How do risk scores in these areas compare to global risk score? Why might it be helpful to calculate metrics like risk scores for different areas of the world and what would their limitations be (what assumptions does risk score make and what important variables might be left out)?

#### Objective 4.1

# Datasets respresent a cumlative sum by date, so last column represents   
# summation for region  
confirmed\_df<-read.csv("data/time\_series\_covid19\_confirmed\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
confirmed\_df<-select(confirmed\_df, Province.State,   
 Country.Region, ncol(confirmed\_df))  
names(confirmed\_df)[3] <- "confirmed"  
deaths\_df<-read.csv("data/time\_series\_covid19\_deaths\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
deaths\_df<-select(deaths\_df, Province.State,   
 Country.Region, ncol(deaths\_df))  
names(deaths\_df)[3] <- "deaths"  
recovered\_df<-read.csv("data/time\_series\_covid19\_recovered\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
recovered\_df<-select(recovered\_df, Province.State,   
 Country.Region, ncol(recovered\_df))  
names(recovered\_df)[3] <- "recovered"  
  
# Combine the datasets into one and fill NA with 0  
combined\_df<-full\_join(confirmed\_df, recovered\_df,   
 by=c("Province.State", "Country.Region"))  
combined\_df<-full\_join(combined\_df, deaths\_df,   
 by=c("Province.State", "Country.Region"))  
combined\_df[is.na(combined\_df)] <- 0  
  
# Assignment is unclear if we are to consider state and region or   
# just region. Based on how data is formatted, I think it is cleaner   
# and makes more sence to use region only. For instance, in confirmed   
# dataset, Canada is broken up by region, but in recovered dataset it   
# uses Canada as a whole. There are numerous examples of this in  
# the data  
grouped\_df<-as.data.frame(summarise\_each(group\_by(  
 select(combined\_df, -Province.State), Country.Region), sum))

## Warning: `summarise\_each\_()` was deprecated in dplyr 0.7.0.  
## ℹ Please use `across()` instead.  
## ℹ The deprecated feature was likely used in the dplyr package.  
## Please report the issue at <]8;;https://github.com/tidyverse/dplyr/issueshttps://github.com/tidyverse/dplyr/issues]8;;>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

# Compute risk and burden by region  
grouped\_df$risk<-grouped\_df$deaths / grouped\_df$recovered  
grouped\_df$burden<-grouped\_df$confirmed \* grouped\_df$risk  
  
cat("Highest risk scores ")

## Highest risk scores

head(arrange(grouped\_df, -risk, -confirmed))

## Country.Region confirmed recovered deaths risk burden  
## 1 US 103802702 0 1123836 Inf Inf  
## 2 India 44690738 0 530779 Inf Inf  
## 3 France 39866718 0 166176 Inf Inf  
## 4 Germany 38249060 0 168935 Inf Inf  
## 5 Brazil 37076053 0 699276 Inf Inf  
## 6 Japan 33320438 0 72997 Inf Inf

cat("Highest risk scores, that are not infinite ")

## Highest risk scores, that are not infinite

head(arrange(grouped\_df[grouped\_df$risk != Inf,], -risk, -confirmed))

## Country.Region confirmed recovered deaths risk burden  
## NA <NA> NA NA NA NA NA  
## NA.1 <NA> NA NA NA NA NA  
## NA.2 <NA> NA NA NA NA NA  
## NA.3 <NA> NA NA NA NA NA  
## NA.4 <NA> NA NA NA NA NA

cat("Lowest Risk Scores ")

## Lowest Risk Scores

head(arrange(grouped\_df, risk, confirmed))

## Country.Region confirmed recovered deaths risk burden  
## 1 Korea, North 1 0 6 Inf Inf  
## 2 MS Zaandam 9 0 2 Inf Inf  
## 3 Diamond Princess 712 0 13 Inf Inf  
## 4 Kiribati 5014 0 18 Inf Inf  
## 5 Nauru 5247 0 1 Inf Inf  
## 6 Palau 5991 0 9 Inf Inf

cat("Lowest risk scores, that are not zero ")

## Lowest risk scores, that are not zero

head(arrange(grouped\_df[grouped\_df$risk != 0,], risk, confirmed))

## Country.Region confirmed recovered deaths risk burden  
## 94 Korea, North 1 0 6 Inf Inf  
## 108 MS Zaandam 9 0 2 Inf Inf  
## 50 Diamond Princess 712 0 13 Inf Inf  
## 93 Kiribati 5014 0 18 Inf Inf  
## 127 Nauru 5247 0 1 Inf Inf  
## 138 Palau 5991 0 9 Inf Inf

global\_confirmed<-sum(grouped\_df$confirmed)  
global\_deaths<-sum(grouped\_df$deaths)  
global\_recovered<-sum(grouped\_df$recovered)  
global\_risk<-global\_deaths / global\_recovered  
global\_burden<-global\_confirmed \* global\_risk  
  
cat("Global Data\n",   
 "Confirmed:", global\_confirmed, "\n",   
 "Deaths: ", global\_deaths, "\n",  
 "Recovered:", global\_recovered, "\n",  
 "Risk: ", global\_risk, "\n",  
 "Burden: ", global\_burden, "\n")

## Global Data  
## Confirmed: 676570149   
## Deaths: 6881802   
## Recovered: 0   
## Risk: Inf   
## Burden: Inf

Based on how the equation is written, any region which has had at least one person recover and no deaths will have a risk score of zero. Examples of this can be seen in the “Lowest Risk Scores” table above. When filtering out risk scores of 0, the regions of lowest risk can be seen in “Lowest risk scores, that are not zero” table above. Any region that has no recoveries and yet at least one death will have infinite risk. Examples of this can be seen in the “Highest risk scores” table above. If filtering out regions that have infinite risk, we see that the regions in the “Highest risk scores, that are not infinite” table above. When looking at the global score, it seems like the risk is high when considering it represents the people that have recovered versus those who have died. This value seems especially high when looking at the regions in the “Lowest risk scores, that are not zero” table, but when compared to the “Highest risk scores, that are not infinite” table where the risk scores are extremely high, the global risk seems less significant. This wide range in risk numbers indicates that while the risk in some regions is extremely high, for the most part, the risk is rather low globally.

Risk assessments like this are important because they are good indicators of where danger is located or help is needed that can be used across many industries. For example, the travel industry may wish to impose bans on travelling to and from locations of high risk. The medical field can use these values to determine locations that are in the most need for medical support. Research fields may also use this data to help identify trends. For example, if a region has a high amount of recoveries and almost no deaths, i.e. low risk score, it may be worth looking into what kind of treatment they are using in that region and if it could be used in other locations throughout the world. The thing to be careful though is that risk scores may be a little misleading. For instance, several regions have almost no cases, but one death and no recoveries causing a massive risk score. Even though these regions have pretty much no cases, they are still seen as extremely risky. This is why it could be beneficial to filter out the extremes before considering the data as valid.

#### Objective 4.2

You are asked to make two tables with the top 10 countries that have the most COVID-19 related confirmations and and deaths. Make sure to include all of the counts for the country, not just the counts for one area in the country. To do this we will need to sum all of the values for each country, create new data frames from these values, and use the package kable to convert those data frames into tables.

Hint: Sum each country’s counts by subsetting the data frame using a list of countries available in the data set. Use a for loop to iterate through the data frame using the list of countries. For each country, calculate the count sum and assign this value to a list.

# Datasets respresent a cumlative sum by date, so last column represents   
# sumation for region  
confirmed\_df<-read.csv("data/time\_series\_covid19\_confirmed\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
confirmed\_df<-select(confirmed\_df, Province.State,   
 Country.Region, ncol(confirmed\_df))  
names(confirmed\_df)[3] <- "confirmed"  
deaths\_df<-read.csv("data/time\_series\_covid19\_deaths\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
deaths\_df<-select(deaths\_df, Province.State,   
 Country.Region, ncol(deaths\_df))  
names(deaths\_df)[3] <- "deaths"  
recovered\_df<-read.csv("data/time\_series\_covid19\_recovered\_global.csv",   
 header=TRUE, stringsAsFactors=FALSE)  
recovered\_df<-select(recovered\_df, Province.State,   
 Country.Region, ncol(recovered\_df))  
names(recovered\_df)[3] <- "recovered"  
  
# Combine the datasets into one and fill NA with 0  
combined\_df<-full\_join(confirmed\_df, recovered\_df,   
 by=c("Province.State", "Country.Region"))  
combined\_df<-full\_join(combined\_df, deaths\_df,   
 by=c("Province.State", "Country.Region"))  
combined\_df[is.na(combined\_df)] <- 0  
  
# Group and combine data by region  
grouped\_df<-as.data.frame(summarise\_each(group\_by(  
 select(combined\_df, -Province.State), Country.Region), sum))  
  
# Compute risk and burden by region  
grouped\_df$risk<-grouped\_df$deaths / grouped\_df$recovered  
grouped\_df$burden<-grouped\_df$confirmed \* grouped\_df$risk  
  
confirmed\_tb = kable(arrange(grouped\_df, -confirmed)[1:5,])  
deaths\_tb = kable(arrange(grouped\_df, -deaths)[1:5,])  
recovered\_tb = kable(arrange(grouped\_df, -recovered)[1:5,])  
  
cat("Top 5 confirmed regions ")

## Top 5 confirmed regions

confirmed\_tb

Country.Region

confirmed

recovered

deaths

risk

burden

US

103802702

0

1123836

Inf

Inf

India

44690738

0

530779

Inf

Inf

France

39866718

0

166176

Inf

Inf

Germany

38249060

0

168935

Inf

Inf

Brazil

37076053

0

699276

Inf

Inf

cat("Top 5 deaths regions ")

## Top 5 deaths regions

deaths\_tb

Country.Region

confirmed

recovered

deaths

risk

burden

US

103802702

0

1123836

Inf

Inf

Brazil

37076053

0

699276

Inf

Inf

India

44690738

0

530779

Inf

Inf

Russia

22075858

0

388478

Inf

Inf

Mexico

7483444

0

333188

Inf

Inf

cat("Top 5 recovered regions")

## Top 5 recovered regions

recovered\_tb

Country.Region

confirmed

recovered

deaths

risk

burden

Afghanistan

209451

0

7896

Inf

Inf

Albania

334457

0

3598

Inf

Inf

Algeria

271496

0

6881

Inf

Inf

Andorra

47890

0

165

Inf

Inf

Angola

105288

0

1933

Inf

Inf

### GitHub Log

#{bash gitlog} #git log --pretty=format:"%nSubject: %s%nAuthor: %aN%nDate: %aD%nBody: %b" #

Could not get the above script to work correctly on my system. This is the output from running the above script directly from the command line:

Subject: Update README.md Author: wesne Date: Thu, 6 Apr 2023 10:20:17 -0700 Body:

Subject: Added first rev of readme.md Author: Chato1969 Date: Wed, 5 Apr 2023 23:28:47 -0700 Body:

Subject: Added first rev of index.md Author: Chato1969 Date: Wed, 5 Apr 2023 23:25:33 -0700 Body:

Subject: Updated responses to be more generic and automatic such that they are created based on data pulled automatically from the table. Fixed typos and minor bugs. Author: Chato1969 Date: Wed, 5 Apr 2023 18:06:15 -0700 Body:

Subject: Merge branch ‘new-chart’ Author: Wesley Newcomb