Project 1

Name: Benjamin Kelly
Partner: Chanel Fraikin

2020-03-31

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

```
confirmed_ds<-read.csv("time_series_covid19_confirmed_global.csv",
                       header=TRUE, stringsAsFactors=FALSE)
deaths_ds<-read.csv("time_series_covid19_deaths_global.csv",
                    header=TRUE, stringsAsFactors=FALSE)
recovered_ds<-read.csv("time_series_covid19_recovered_global.csv",
                       header=TRUE, stringsAsFactors=FALSE)

cat("Confirmed Dataset")
```

```
## Confirmed Dataset
```

```
head(select(arrange(confirmed_ds, -X1.22.20), Province.State, Country.Region, X1.22.20))
```

```
##   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
```

```
cat("Deaths Dataset")
```

```
## Deaths Dataset
```

```
head(select(arrange(deaths_ds, -X1.22.20), Province.State, Country.Region, X1.22.20))
```

```
##   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
```

```
cat("Recovered Dataset")
```

```
## Recovered Dataset
```

```
head(select(arrange(recovered_ds, -X1.22.20), Province.State, Country.Region, X1.22.20))
```

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

```
#Proving origin using conditional statement
confirmed_ordered<-head(select(arrange(confirmed_ds, -X1.22.20), Province.State, Country.Region, X1.22.2
death_ordered<-head(select(arrange(deaths_ds, -X1.22.20), Province.State, Country.Region, X1.22.20))
recovered_ordered<-head(select(arrange(recovered_ds, -X1.22.20), Province.State, Country.Region, X1.22.2
if(confirmed_ordered[1,1]==death_ordered[1,1] && confirmed_ordered[1,1]==recovered_ordered[1,1] && deat
{
  print(confirmed_ordered[1,1])
}
```

```
## [1] "Hubei"
```

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

```
confirmed_ds<-read.csv("time_series_covid19_confirmed_global.csv",
                       header=TRUE, stringsAsFactors=FALSE)
#recent_ds<-arrange(confirmed_ds[confirmed_ds$X3.25.20 == 0
#& confirmed_ds$X3.26.20 > 0, ], -X3.26.20)
num_columns<-ncol(confirmed_ds)
recent_ds<-arrange(confirmed_ds[confirmed_ds[,num_columns-1] == 0
                                & confirmed_ds[,num_columns] > 0,], -X3.26.20)
#head(select(recent_ds, Province.State, Country.Region, X3.25.20, X3.26.20))
head(select(recent_ds, Province.State, Country.Region, ncol(confirmed_ds)-1, ncol(confirmed_ds)))
```

```
##          Province.State Country.Region X3.25.20 X3.26.20
## 1                              Kosovo        0       71
## 2                 Yukon         Canada        0        3
## 3 Northwest Territories         Canada        0        1
```

As of 3/26, Kosovo, Yukon, Canada and Northwest Territories, Canada are the most recent areas to have their first confirmed cases. We found this by going through the data and selecting the countries that did not have cases before March 26th and had their first cases on March 26th.Our code can be used for an updated version of the confirmed cases file, but our report was written with March 26th as the last date of data.

Objective 3

```
confirmed_ds<-read.csv("time_series_covid19_confirmed_global.csv",
                       header=TRUE, stringsAsFactors=FALSE)
recent_ds<-arrange(confirmed_ds[confirmed_ds$X3.25.20 == 0 &
                                  confirmed_ds$X3.26.20 > 0, ], -X3.26.20)
origin_ds<-arrange(confirmed_ds, -X1.22.20)[1,]

# Compute distances from origin
distances<-distm(select(recent_ds, Long, Lat), select(origin_ds, Long, Lat))
# Convert from m to miles
distances<-distances * 0.00062137

# Add distance from origin to dataframe and sort by distance
recent_ds$distance<-distances[,1]
recent_ds<-arrange(recent_ds, distance)

# Vector is small enough that loop is reasonable
for (i in 1:nrow(recent_ds)) {
  city<-recent_ds[i, "Province.State"]

  # If there is no city use country
  if (city == "") {
    city<-recent_ds[i, "Country.Region"]
  }

  cat(city, "is", recent_ds[i, "distance"], "miles away from",
      paste0(origin_ds[1, "Province.State"], ","),
      paste0(origin_ds[1, "Country.Region"], "."), "\n")
}
```

```
## Kosovo is 4883.677 miles away from Hubei, China.
## Yukon is 4939.766 miles away from Hubei, China.
```

```
## Northwest Territories is 5156.52 miles away from Hubei, China.
```

Objective 4

Objective 4.1

```
# Datasets respresent a cumlative sum by date, so last column represents
# sumation for region
confirmed_ds<-select(read.csv("time_series_covid19_confirmed_global.csv",
                      header=TRUE, stringsAsFactors=FALSE), Province.State,
                    Country.Region, X3.26.20)
names(confirmed_ds)[3] <- "confirmed"
deaths_ds<-select(read.csv("time_series_covid19_deaths_global.csv",
                    header=TRUE, stringsAsFactors=FALSE), Province.State,
                  Country.Region, X3.26.20)
names(deaths_ds)[3] <- "deaths"
recovered_ds<-select(read.csv("time_series_covid19_recovered_global.csv",
                      header=TRUE, stringsAsFactors=FALSE), Province.State,
                    Country.Region, X3.26.20)
names(recovered_ds)[3] <- "recovered"

# Combine the datasets into one and fill NA with 0
combined_ds<-full_join(confirmed_ds, recovered_ds,
                      by=c("Province.State", "Country.Region"))
combined_ds<-full_join(combined_ds, deaths_ds,
                      by=c("Province.State", "Country.Region"))
combined_ds[is.na(combined_ds)] <- 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_ds<-as.data.frame(summarise_each(group_by(
  select(combined_ds, -Province.State), Country.Region), sum))

# compute risk and burden by region
grouped_ds$risk<-grouped_ds$deaths / grouped_ds$recovered
grouped_ds$burden<-grouped_ds$confirmed * grouped_ds$risk

cat("Highest risk scores")
```

```
## Highest risk scores
```

```
head(arrange(grouped_ds, -risk, -confirmed))
```

```
##        Country.Region confirmed recovered deaths risk burden
## 1              Serbia       384         0      1  Inf    Inf
## 2           Mauritius        81         0      2  Inf    Inf
## 3              Kosovo        71         0      1  Inf    Inf
## 4          Montenegro        69         0      1  Inf    Inf
## 5 Trinidad and Tobago        65         0      1  Inf    Inf
## 6            Honduras        52         0      1  Inf    Inf
```

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

```
## Highest risk scores, that are not infinite
```

```
head(arrange(grouped_ds[grouped_ds$risk != Inf,], -risk, -confirmed))
```

```
##   Country.Region confirmed recovered deaths     risk    burden
## 1    Netherlands      7468         6    435 72.50000 541430.00
## 2         Brazil      2985         6     77 12.83333  38307.50
## 3        Ecuador      1403         3     34 11.33333  15900.67
## 4     San Marino       208         4     21  5.25000   1092.00
## 5        Ukraine       196         1      5  5.00000    980.00
## 6         Sweden      2840        16     77  4.81250  13667.50
```

```
cat("Lowest Risk Scores")
```

```
## Lowest Risk Scores
```

```
head(arrange(grouped_ds, risk, confirmed))
```

```
##   Country.Region confirmed recovered deaths risk burden
## 1          Nepal         3         1      0    0      0
## 2        Namibia         8         2      0    0      0
## 3        Bahamas         9         1      0    0      0
## 4       Maldives        13         8      0    0      0
## 5           Togo        23         1      0    0      0
## 6         Monaco        33         1      0    0      0
```

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

```
## Lowest risk scores, that are not zero
```

```
head(arrange(grouped_ds[grouped_ds$risk != 0,], risk, confirmed))
```

```
##         Country.Region confirmed recovered deaths       risk     burden
## 1            Singapore       683       172      2 0.01162791   7.941860
## 2     Diamond Princess       712       597     10 0.01675042  11.926298
## 3              Bahrain       458       204      4 0.01960784   8.980392
## 4              Iceland       802        82      2 0.02439024  19.560976
## 5         Korea, South      9241      4144    131 0.03161197 292.126207
## 6 United Arab Emirates       333        52      2 0.03846154  12.807692
```

```
global_confirmed<-sum(grouped_ds$confirmed)
global_deaths<-sum(grouped_ds$deaths)
global_recovered<-sum(grouped_ds$recovered)
global_risk<-global_deaths / global_recovered
global_burden<-global_confirmed * global_risk

cat("Global Data\n",
    "Confirmed:", global_confirmed,
    "Deaths:", global_deaths,
    "Recovered:", global_recovered, "\n",
    "Risk:", global_risk, "Burden:", global_burden, "\n")
```

```
## Global Data
##  Confirmed: 529591 Deaths: 23970 Recovered: 122150
##  Risk: 0.1962341 Burden: 103923.8
```

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. Some examples of this are Nepal, Namibia and the Bahamas. When filtering out risk scores of 0, the regions of lowest risk include Singapore, Diamond Princess and Bahrain. Any region that has no recoveries and yet at least one death will have infinite risk. Examples of this are Serbia, Mauritius and Kosovo. If filtering out regions that have infinite risk, we see that the Netherlands, Brazil and Ecuador have the highest risk. When looking at the global score of risk of 0.196, it seems like the risk is high in that almost 20% of the people that have recovered have died. This value seems especially high when looking at regions like Singapore which have a risk score of only 0.012, but when compared to the Netherlands where the risk is 72.5, the global risk seems less significant. This wide range in risk numbers indicate that while the risk in some regions is extremely high, yet globally, for the most part, the risk is rather low.

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 benficial to filter out the extremes before considering the data as valid.

Objective 4.2

```
# Datasets respresent a cumlative sum by date, so last column represents
# sumation for region
confirmed_ds<-select(read.csv("time_series_covid19_confirmed_global.csv",
                      header=TRUE, stringsAsFactors=FALSE), Province.State,
                    Country.Region, X3.26.20)
names(confirmed_ds)[3] <- "confirmed"
deaths_ds<-select(read.csv("time_series_covid19_deaths_global.csv",
                    header=TRUE, stringsAsFactors=FALSE), Province.State,
                  Country.Region, X3.26.20)
names(deaths_ds)[3] <- "deaths"
recovered_ds<-select(read.csv("time_series_covid19_recovered_global.csv",
                      header=TRUE, stringsAsFactors=FALSE), Province.State,
                    Country.Region, X3.26.20)
names(recovered_ds)[3] <- "recovered"

# Combine the datasets into one and fill NA with 0
combined_ds<-full_join(confirmed_ds, recovered_ds,
                       by=c("Province.State", "Country.Region"))
combined_ds<-full_join(combined_ds, deaths_ds,
                       by=c("Province.State", "Country.Region"))
combined_ds[is.na(combined_ds)] <- 0

# Group and combine data by region
grouped_ds<-as.data.frame(summarise_each(group_by(
  select(combined_ds, -Province.State), Country.Region), sum))

# compute risk and burden by region
grouped_ds$risk<-grouped_ds$deaths / grouped_ds$recovered
grouped_ds$burden<-grouped_ds$confirmed * grouped_ds$risk

confirmed_tb = kable(arrange(grouped_ds, -confirmed)[1:5,])
deaths_tb = kable(arrange(grouped_ds, -deaths)[1:5,])
```

```
recovered_tb = kable(arrange(grouped_ds, -recovered)[1:5,])

cat("Top 5 confirmed regions")
```

```
## Top 5 confirmed regions
```

```
confirmed_tb
```

Country.Region	confirmed	recovered	deaths	risk	burden
US	83836	681	1209	1.7753304	148836.599
China	81782	74181	3291	0.0443645	3628.214
Italy	80589	10361	8215	0.7928771	63897.175
Spain	57786	7015	4365	0.6222381	35956.649
Germany	43938	5673	267	0.0470650	2067.944

```
cat("Top 5 deaths regions")
```

```
## Top 5 deaths regions
```

```
deaths_tb
```

Country.Region	confirmed	recovered	deaths	risk	burden
Italy	80589	10361	8215	0.7928771	63897.175
Spain	57786	7015	4365	0.6222381	35956.649
China	81782	74181	3291	0.0443645	3628.214
Iran	29406	10457	2234	0.2136368	6282.204
France	29551	4955	1698	0.3426842	10126.660

```
cat("Top 5 recovered regions")
```

```
## Top 5 recovered regions
```

```
recovered_tb
```

Country.Region	confirmed	recovered	deaths	risk	burden
China	81782	74181	3291	0.0443645	3628.214
Iran	29406	10457	2234	0.2136368	6282.204
Italy	80589	10361	8215	0.7928771	63897.175
Spain	57786	7015	4365	0.6222381	35956.649
Germany	43938	5673	267	0.0470650	2067.944

GitHub Log

```
#{bash gitlog}  # Can't get this to work :'( # Probably need to install something #git
log --pretty=format:"%nSubject: %s%nAuthor: %aN%nDate: %aD%nBody: %b" #
```