Project 1

Name: Benjamin Kelly
Partner: Chanel Fraikin

2020-04-03

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)

confirmed_ordered<-select(arrange(confirmed_ds, -X1.22.20),
```

```
                          Province.State, Country.Region, X1.22.20)
death_ordered<-select(arrange(deaths_ds, -X1.22.20),
                      Province.State, Country.Region, X1.22.20)
recovered_ordered<-select(arrange(recovered_ds, -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
```

```
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
```

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

```
## Recovered Dataset
```

```
head(recovered_ordered)
```

```
##   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
```

```
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

```
confirmed_ds<-read.csv("time_series_covid19_confirmed_global.csv",
                       header=TRUE, stringsAsFactors=FALSE)
recent_ds<-arrange(confirmed_ds[confirmed_ds[,ncol(confirmed_ds)-1] == 0
                                & confirmed_ds[,ncol(confirmed_ds)] > 0,])

i<-0
# If there are no new cases today loop back to find most recent region
# to have new cases
if (nrow(recent_ds) == 0) {
  while (nrow(recent_ds) == 0) {
    i<-i+1
    recent_ds<-arrange(confirmed_ds[confirmed_ds[,ncol(confirmed_ds)-1-i] == 0
                                & confirmed_ds[,ncol(confirmed_ds)-i] > 0,])
  }
}

head(select(recent_ds, Province.State, Country.Region, ncol(confirmed_ds)-1-i,
            ncol(confirmed_ds)-i))
```

```
##                     Province.State Country.Region X4.1.20 X4.2.20
## 1 Bonaire, Sint Eustatius and Saba    Netherlands       0       2
## 2                                         Malawi       0       3
```

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

```
## Bonaire, Sint Eustatius and Saba, Netherlands has recently had their first confirmed case.
## Malawi 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

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

i<-0
# If there are no new cases today loop back to find most recent region
# to have new cases
if (nrow(recent_ds) == 0) {
  while (nrow(recent_ds) == 0) {
    i<-i+1
    recent_ds<-arrange(confirmed_ds[confirmed_ds[,ncol(confirmed_ds)-1-i] == 0
                                & confirmed_ds[,ncol(confirmed_ds)-i] > 0,])
  }
}

# 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)

head(select(recent_ds, Province.State, Country.Region, Lat, Long, distance))
```

```
##                     Province.State Country.Region       Lat      Long distance
## 1                                          Malawi -13.25431  34.30152 5995.123
## 2 Bonaire, Sint Eustatius and Saba    Netherlands  12.17840 -68.23850 9462.555
```

```
# 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 the virus origin in",
      paste0(origin_ds[1, "Province.State"], ","),
      paste0(origin_ds[1, "Country.Region"], "."), "\n")
}
```

```
## Malawi is 5995.123 miles away from the virus origin in Hubei, China.
## Bonaire, Sint Eustatius and Saba is 9462.555 miles away from the virus origin in Hubei, China.
```

Objective 4

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

Objective 4.1

```
## Highest risk scores
```

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

```
##   Country.Region confirmed recovered deaths risk burden
## 1         Serbia      1476         0     39  Inf    Inf
## 2      Mauritius       186         0      7  Inf    Inf
## 3          Niger       120         0      5  Inf    Inf
```

```
## 4    El Salvador        46         0      2  Inf    Inf
## 5           Mali        39         0      3  Inf    Inf
## 6         Guyana        23         0      4  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             Ireland      4273         5    120 24.000000 102552.0
## 2      United Kingdom     38689       208   3611 17.360577 671663.4
## 3             Bolivia       132         1      9  9.000000   1188.0
## 4 Trinidad and Tobago        98         1      6  6.000000    588.0
## 5         Netherlands     15821       260   1490  5.730769  90666.5
## 6            Honduras       222         3     15  5.000000   1110.0
```

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

```
## Lowest Risk Scores
```

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

```
##                     Country.Region confirmed recovered deaths risk burden
## 1 Saint Vincent and the Grenadines         3         1      0    0      0
## 2                           Bhutan         5         2      0    0      0
## 3                            Nepal         6         1      0    0      0
## 4                          Somalia         7         1      0    0      0
## 5                      Saint Lucia        13         1      0    0      0
## 6                         Mongolia        14         2      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    New Zealand       868       103      1 0.009708738  8.427184
## 2        Bahrain       672       382      4 0.010471204  7.036649
## 3        Iceland      1364       309      4 0.012944984 17.656958
## 4        Senegal       207        66      1 0.015151515  3.136364
## 5         Brunei       134        65      1 0.015384615  2.061538
## 6           Oman       252        57      1 0.017543860  4.421053
```

```
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, "\n",
    "Deaths:   ", global_deaths, "\n",
    "Recovered:", global_recovered, "\n",
    "Risk:     ", global_risk, "\n",
    "Burden:   ", global_burden, "\n")
```

```
## Global Data
##  Confirmed: 1095917
##  Deaths:    58787
##  Recovered: 225796
##  Risk:      0.2603545
##  Burden:    285326.9
```

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.

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

```r
# 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")
```

Objective 4.2

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

```r
confirmed_tb
```

Country.Region	confirmed	recovered	deaths	risk	burden
US	275586	9707	7087	0.7300917	201203.047
Italy	119827	19758	14681	0.7430408	89036.349
Spain	119199	30513	11198	0.3669911	43744.974
Germany	91159	24575	1275	0.0518820	4729.511
China	82511	76760	3326	0.0433299	3575.190

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

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

```r
deaths_tb
```

Country.Region	confirmed	recovered	deaths	risk	burden
Italy	119827	19758	14681	0.7430408	89036.35
Spain	119199	30513	11198	0.3669911	43744.97
US	275586	9707	7087	0.7300917	201203.05
France	65202	14135	6520	0.4612664	30075.49
United Kingdom	38689	208	3611	17.3605769	671663.36

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

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

```r
recovered_tb
```

Country.Region	confirmed	recovered	deaths	risk	burden
China	82511	76760	3326	0.0433299	3575.190
Spain	119199	30513	11198	0.3669911	43744.974
Germany	91159	24575	1275	0.0518820	4729.511
Italy	119827	19758	14681	0.7430408	89036.349
Iran	53183	17935	3294	0.1836632	9767.761

GitHub Log

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