Group Project: Covid Origins

Name: Sal Figueroa
Partner: Kylie Stearns

2025-03-30

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 Corona virus 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

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.

```
#Most deaths and confirmed cases on x1.22.20 (date)
#find max value in column 5 (10.22.20) and check to see how many common max values there is.
max_value <- 0 #holds the updated max_value
RowMax <- 0 #this holds row (i) position in Data frame
```

```r
#For loop starting it column 5 to final column using nrow().
for(i in 1:nrow(deaths_global[,]))
{
num <- deaths_global[i,5] + confirmed_global[i,5]#Holds value to be tested against the current max valu

  if (num > max_value) #confitional if loop that updates the max value as iterates through column 5.
    {
    max_value <- num
    RowMax <- i #Holds the row position where the max value is located
    }
}

deathNconfimred <- deaths_global[RowMax,5] + confirmed_global[RowMax,5]
cat("Province/State:",deaths_global[RowMax,1],"Province/State:",deaths_global[RowMax,2],"\n")
```

```
## Province/State: Hubei Province/State: China
```

```r
cat("Latitude:",deaths_global[RowMax,3],",Longitude:",deaths_global[RowMax,4],"\n")
```

```
## Latitude: 30.9756 ,Longitude: 112.2707
```

```r
cat("Deaths:",deaths_global[RowMax,5],",confirmed cases:",confirmed_global[RowMax,5],"\n")
```

```
## Deaths: 17 ,confirmed cases: 444
```

```r
cat("Predicted area of Origin had the most initial deaths and confirmed cases (01.22.2020) at:",deathNc
```

```
## Predicted area of Origin had the most initial deaths and confirmed cases (01.22.2020) at: 461
```

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.

```r
#find max value in column 5 (10.22.20)
#check to see how many max location there is
#run conditional loop
max_value <- 0
RowMax <- 0
num <- 0
i <- 5
j <- 1

while(j < 331484 && num == 0) #while(num == 0 && j < 290)
 {
  num <- confirmed_global[j,i]
#  cat("INSIDE WHILE: j:",j,"i:",i,"\n")

  j <- j + 1  #Iterate through rows 1-289

     if( j %% 289 == 0)
      {
      i <- i + 1 #Iterate through columns 1-1147
#      cat("INSIDE IF - Col: ",j,"Row",i,"\n")
      }
 }
```

```
j <- j - 1 #

cat("Province/State: ",confirmed_global[j,1],"Province/State: ",confirmed_global[j,2],"\n")
```

```
## Province/State:  Anhui Province/State:  China
```

```
cat("Latitude: ",confirmed_global[j,3]," ,Longitude:",confirmed_global[j,4],"\n")
```

```
## Latitude:  31.8257  ,Longitude: 117.2264
```

```
cat("Predicted area of Origin had the most initial deaths (01.22.2020) at:",confirmed_global[j,i],"\n")
```

```
## Predicted area of Origin had the most initial deaths (01.22.2020) at: 1
```

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}.

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)?

```
# removes NA values
deaths_global[is.na(deaths_global)] <- 0
confirmed_global[is.na(confirmed_global)] <- 0
#1147 number of columns
iCntLength <- ncol(deaths_global)
#289 number of rows
jCntLength <- nrow(deaths_global)
#loop counter default
i <- 5
j <- 1

SUMdeaths_global <- 0
SUMconfirmed_global <- 0
sumRiskScore  <- 0

#Find higest and lowest Risk score
low_value <- 1 #holds the lowest risk score value.
```

```
iLow <- 1 #Holds data frame coordinates for low
jLow <- 1
high_value <- 0 #holds the highest risk score value.
iHi <- 1 #Holds data frame coordinates for hi
jHi <- 1


for(j in 1:jCntLength) #Iterate through rows 1-289
  {
  for(i in 5:iCntLength) #Iterate through columns 1-1147
    {
      emptyClr <- (deaths_global[j,3] + deaths_global[j,4]) #deaths_global[row, col]

      if(emptyClr > 0 && confirmed_global[j,i] != 0) #Filters out the first 4 Columns of empty cells in
        {
#Global riskscore
        SUMdeaths_global <- SUMdeaths_global + deaths_global[j,i]
        SUMconfirmed_global <- SUMconfirmed_global + confirmed_global[j,i]

          riskScore <- (deaths_global[j,1147]/confirmed_global[j, 1147])*100
          #cat(riskScore, "\n")
#Country with lowest riskscore (if more than one, display the one with the most confirmations)
          if (riskScore > high_value) #if (riskScore > high_value && i == iCntLength)
            {
            high_value <- riskScore
            jHi <- j
             cat("High - Risk score:",riskScore, ",Row/Col:", jHi,",1147\n")
             cat("Deaths:",deaths_global[j,1147]," ,Confirmed:", confirmed_global[j, 1147],"\n")
            }
#Country with lowest riskscore (if more than one, display the one with the most confirmations)
          if (riskScore < low_value)
            {
            low_value <- riskScore
            jLow <- j
            cat("Low - Risk score:",riskScore, ",Row/Col:",jLow,",1147\n")
            cat("Deaths:",deaths_global[j,1147]," ,Confirmed:", confirmed_global[j, 1147],"\n")
            }
        }
    }
  }
```

Objective 4.1

```
## High - Risk score: 3.769855 ,Row/Col: 1 ,1147
## Deaths: 7896  ,Confirmed: 209451
## Low - Risk score: 0.3445396 ,Row/Col: 4 ,1147
## Deaths: 165  ,Confirmed: 47890
## Low - Risk score: 0.097865 ,Row/Col: 10 ,1147
## Deaths: 228  ,Confirmed: 232974
## Low - Risk score: 0.08657514 ,Row/Col: 12 ,1147
## Deaths: 91  ,Confirmed: 105111
## Low - Risk score: 0.07360098 ,Row/Col: 17 ,1147
## Deaths: 952  ,Confirmed: 1293461
## Low - Risk score: 0.03353186 ,Row/Col: 28 ,1147
```

```
## Deaths: 21  ,Confirmed: 62627
## High - Risk score: 4.052483 ,Row/Col: 30 ,1147
## Deaths: 16280  ,Confirmed: 401729
## Low - Risk score: 0.01168088 ,Row/Col: 63 ,1147
## Deaths: 2  ,Confirmed: 17122
## Low - Risk score: 0.009685418 ,Row/Col: 65 ,1147
## Deaths: 10  ,Confirmed: 103248
## High - Risk score: 6.259445 ,Row/Col: 73 ,1147
## Deaths: 4515  ,Confirmed: 72131
## Low - Risk score: 0 ,Row/Col: 76 ,1147
## Deaths: 0  ,Confirmed: 5075
## High - Risk score: 600 ,Row/Col: 162 ,1147
## Deaths: 6  ,Confirmed: 1
```

```
print("----------CHECK DATA ABOVE---------")
```

```
## [1] "----------CHECK DATA ABOVE---------"
```

```
sumRiskScore <- (SUMdeaths_global/SUMconfirmed_global)*100
cat("\nAverage Risk Score:",sumRiskScore, "\n\n")
```

```
##
## Average Risk Score: 1.110684
```

```
cat("Highest current Risk Score:", high_value,"\n")
```

```
## Highest current Risk Score: 600
```

```
cat("Row positions: ",jHi,",Column Position:", 1147,"\n")
```

```
## Row positions:  162 ,Column Position: 1147
```

```
cat("Deaths",deaths_global[jHi,1147],"/confirmed",confirmed_global[jHi,1147],"\n")
```

```
## Deaths 6 /confirmed 1
```

```
cat("Province/State:", deaths_global[jHi, 1],",Country/Region:", deaths_global[jHi, 2],"\n")
```

```
## Province/State:  ,Country/Region: Korea, North
```

```
cat("Lat.", deaths_global[jHi, 3],",Long.", deaths_global[jHi, 4],"\n\n")
```

```
## Lat. 40.3399 ,Long. 127.5101
```

```
cat("Lowest current Risk Score:", low_value,"\n")
```

```
## Lowest current Risk Score: 0
```

```
cat("Row positions: ",jLow,",Column Position:", 1147,"\n")
```

```
## Row positions:  76 ,Column Position: 1147
```

```
cat("Deaths",deaths_global[jLow,1147],"/confirmed",confirmed_global[jLow,1147],"\n")
```

```
## Deaths 0 /confirmed 5075
```

```
cat("Province/State:", deaths_global[jLow, 1],",Country/Region:", deaths_global[jLow, 2],"\n")
```

```
## Province/State: Jiangsu ,Country/Region: China
```

```
cat("Lat.", deaths_global[jLow, 3],",Long.", deaths_global[jLow, 4],"\n\n")
```

```
## Lat. 32.9711 ,Long. 119.455
```

```
#Country with highest riskscore (if more than one, display the one with the most confirmations)
n <- 1
HiRMaxConfirm <-0
HiRownum <- 0
LowRMaxConfirm <-0
LowRownum <-0
HiMaxConfirmed <- 1
LoMaxConfirmed <- 1

for(n in 1:jCntLength) #Iterate through rows 1-289
  {
  #cat(n,"- risk score: ",chkRiskScore, ",Deaths:",deaths_global[n,1147],"confirmed:",confirmed_global[

  if(confirmed_global[n, 1147] == 0)
    {
      chkRiskScore <- 0
    }
  else
    {
      chkRiskScore <- (deaths_global[n,1147]/confirmed_global[n, 1147])*100
    }

  #cat(n,"- risk score: ",chkRiskScore,",Deaths:",deaths_global[n,1147],",confirmed:",confirmed_global[

  cat("HI Check:", chkRiskScore,",high_value:",high_value,"\n")
  if(chkRiskScore == high_value)
    {
      if(confirmed_global[n, 1147] >= HiMaxConfirmed)
        {
        HiMaxConfirmed <- confirmed_global[n, 1147]
        HiColNum <- n
        cat("----------high_value:",chkRiskScore,",confirmed_global:",confirmed_global[n, 1147],"\n")
        }
    }


  cat("Low Check:", chkRiskScore,",low_value:",low_value,"\n")
  if(chkRiskScore == low_value)
    {
      if(confirmed_global[n, 1147] < LoMaxConfirmed)
        {
        LoMaxConfirmed <- confirmed_global[n, 1147]
        LoColNum <- n
        cat("---------low_value:",chkRiskScore,",confirmed_global:",confirmed_global[n, 1147],"\n")
        }
    }



  }
```

Objective 4.2

```
## HI Check: 3.769855 ,high_value: 600
## Low Check: 3.769855 ,low_value: 0
## HI Check: 1.075774 ,high_value: 600
## Low Check: 1.075774 ,low_value: 0
## HI Check: 2.534476 ,high_value: 600
## Low Check: 2.534476 ,low_value: 0
## HI Check: 0.3445396 ,high_value: 600
## Low Check: 0.3445396 ,low_value: 0
## HI Check: 1.835917 ,high_value: 600
## Low Check: 1.835917 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 1.603338 ,high_value: 600
## Low Check: 1.603338 ,low_value: 0
## HI Check: 1.298881 ,high_value: 600
## Low Check: 1.298881 ,low_value: 0
## HI Check: 1.951005 ,high_value: 600
## Low Check: 1.951005 ,low_value: 0
## HI Check: 0.097865 ,high_value: 600
## Low Check: 0.097865 ,low_value: 0
## HI Check: 0.1667266 ,high_value: 600
## Low Check: 0.1667266 ,low_value: 0
## HI Check: 0.08657514 ,high_value: 600
## Low Check: 0.08657514 ,low_value: 0
## HI Check: 0.1545908 ,high_value: 600
## Low Check: 0.1545908 ,low_value: 0
## HI Check: 0.1544782 ,high_value: 600
## Low Check: 0.1544782 ,low_value: 0
## HI Check: 0.08904131 ,high_value: 600
## Low Check: 0.08904131 ,low_value: 0
## HI Check: 0.2558531 ,high_value: 600
## Low Check: 0.2558531 ,low_value: 0
## HI Check: 0.07360098 ,high_value: 600
## Low Check: 0.07360098 ,low_value: 0
## HI Check: 0.3685535 ,high_value: 600
## Low Check: 0.3685535 ,low_value: 0
## HI Check: 1.223177 ,high_value: 600
## Low Check: 1.223177 ,low_value: 0
## HI Check: 2.221867 ,high_value: 600
## Low Check: 2.221867 ,low_value: 0
## HI Check: 0.2185191 ,high_value: 600
## Low Check: 0.2185191 ,low_value: 0
## HI Check: 1.44489 ,high_value: 600
## Low Check: 1.44489 ,low_value: 0
## HI Check: 0.542145 ,high_value: 600
## Low Check: 0.542145 ,low_value: 0
## HI Check: 0.7160699 ,high_value: 600
## Low Check: 0.7160699 ,low_value: 0
## HI Check: 0.7134711 ,high_value: 600
## Low Check: 0.7134711 ,low_value: 0
## HI Check: 0.972342 ,high_value: 600
## Low Check: 0.972342 ,low_value: 0
## HI Check: 0.5821636 ,high_value: 600
## Low Check: 0.5821636 ,low_value: 0
```

```
## HI Check: 0.03353186 ,high_value: 600
## Low Check: 0.03353186 ,low_value: 0
## HI Check: 1.872681 ,high_value: 600
## Low Check: 1.872681 ,low_value: 0
## HI Check: 4.052483 ,high_value: 600
## Low Check: 4.052483 ,low_value: 0
## HI Check: 0.8494108 ,high_value: 600
## Low Check: 0.8494108 ,low_value: 0
## HI Check: 1.886058 ,high_value: 600
## Low Check: 1.886058 ,low_value: 0
## HI Check: 0.08045455 ,high_value: 600
## Low Check: 0.08045455 ,low_value: 0
## HI Check: 2.946229 ,high_value: 600
## Low Check: 2.946229 ,low_value: 0
## HI Check: 1.79543 ,high_value: 600
## Low Check: 1.79543 ,low_value: 0
## HI Check: 3.074375 ,high_value: 600
## Low Check: 3.074375 ,low_value: 0
## HI Check: 0.07085454 ,high_value: 600
## Low Check: 0.07085454 ,low_value: 0
## HI Check: 0.6530264 ,high_value: 600
## Low Check: 0.6530264 ,low_value: 0
## HI Check: 2.203015 ,high_value: 600
## Low Check: 2.203015 ,low_value: 0
## HI Check: 1.579684 ,high_value: 600
## Low Check: 1.579684 ,low_value: 0
## HI Check: 0.8934176 ,high_value: 600
## Low Check: 0.8934176 ,low_value: 0
## HI Check: 1.322776 ,high_value: 600
## Low Check: 1.322776 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## ---------low_value: 0 ,confirmed_global: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 1.592637 ,high_value: 600
## Low Check: 1.592637 ,low_value: 0
## HI Check: 0.9384917 ,high_value: 600
## Low Check: 0.9384917 ,low_value: 0
## HI Check: 0.5807477 ,high_value: 600
## Low Check: 0.5807477 ,low_value: 0
## HI Check: 0.1911215 ,high_value: 600
## Low Check: 0.1911215 ,low_value: 0
## HI Check: 0.5639485 ,high_value: 600
## Low Check: 0.5639485 ,low_value: 0
## HI Check: 0.1982441 ,high_value: 600
## Low Check: 0.1982441 ,low_value: 0
## HI Check: 1.013785 ,high_value: 600
## Low Check: 1.013785 ,low_value: 0
## HI Check: 0.1645348 ,high_value: 600
## Low Check: 0.1645348 ,low_value: 0
## HI Check: 1.375419 ,high_value: 600
## Low Check: 1.375419 ,low_value: 0
## HI Check: 0 ,high_value: 600
```

```
## Low Check: 0 ,low_value: 0
## HI Check: 1.23006 ,high_value: 600
## Low Check: 1.23006 ,low_value: 0
## HI Check: 0.6414111 ,high_value: 600
## Low Check: 0.6414111 ,low_value: 0
## HI Check: 0.7352941 ,high_value: 600
## Low Check: 0.7352941 ,low_value: 0
## HI Check: 2.526371 ,high_value: 600
## Low Check: 2.526371 ,low_value: 0
## HI Check: 1.237856 ,high_value: 600
## Low Check: 1.237856 ,low_value: 0
## HI Check: 0.3076923 ,high_value: 600
## Low Check: 0.3076923 ,low_value: 0
## HI Check: 0.04905087 ,high_value: 600
## Low Check: 0.04905087 ,low_value: 0
## HI Check: 0.07475365 ,high_value: 600
## Low Check: 0.07475365 ,low_value: 0
## HI Check: 0.01168088 ,high_value: 600
## Low Check: 0.01168088 ,low_value: 0
## HI Check: 0.1148106 ,high_value: 600
## Low Check: 0.1148106 ,low_value: 0
## HI Check: 0.009685418 ,high_value: 600
## Low Check: 0.009685418 ,low_value: 0
## HI Check: 0.01495774 ,high_value: 600
## Low Check: 0.01495774 ,low_value: 0
## HI Check: 0.0789266 ,high_value: 600
## Low Check: 0.0789266 ,low_value: 0
## HI Check: 0.05723552 ,high_value: 600
## Low Check: 0.05723552 ,low_value: 0
## HI Check: 0.2126367 ,high_value: 600
## Low Check: 0.2126367 ,low_value: 0
## HI Check: 0.2726034 ,high_value: 600
## Low Check: 0.2726034 ,low_value: 0
## HI Check: 0.2312023 ,high_value: 600
## Low Check: 0.2312023 ,low_value: 0
## HI Check: 0.4682373 ,high_value: 600
## Low Check: 0.4682373 ,low_value: 0
## HI Check: 6.259445 ,high_value: 600
## Low Check: 6.259445 ,low_value: 0
## HI Check: 0.05378513 ,high_value: 600
## Low Check: 0.05378513 ,low_value: 0
## HI Check: 0.01130327 ,high_value: 600
## Low Check: 0.01130327 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 0.05842828 ,high_value: 600
## Low Check: 0.05842828 ,low_value: 0
## HI Check: 0.01226572 ,high_value: 600
## Low Check: 0.01226572 ,low_value: 0
## HI Check: 0.05638568 ,high_value: 600
## Low Check: 0.05638568 ,low_value: 0
## HI Check: 3.443369 ,high_value: 600
## Low Check: 3.443369 ,low_value: 0
## HI Check: 0 ,high_value: 600
```

```
## Low Check: 0 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 0.06825007 ,high_value: 600
## Low Check: 0.06825007 ,low_value: 0
## HI Check: 0.170068 ,high_value: 600
## Low Check: 0.170068 ,low_value: 0
## HI Check: 0.8875298 ,high_value: 600
## Low Check: 0.8875298 ,low_value: 0
## HI Check: 0.01395284 ,high_value: 600
## Low Check: 0.01395284 ,low_value: 0
## HI Check: 0.08237798 ,high_value: 600
## Low Check: 0.08237798 ,low_value: 0
## HI Check: 0.06830601 ,high_value: 600
## Low Check: 0.06830601 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 5.401113 ,high_value: 600
## Low Check: 5.401113 ,low_value: 0
## HI Check: 0.09711881 ,high_value: 600
## Low Check: 0.09711881 ,low_value: 0
## HI Check: 0.04105512 ,high_value: 600
## Low Check: 0.04105512 ,low_value: 0
## HI Check: 0.008440243 ,high_value: 600
## Low Check: 0.008440243 ,low_value: 0
## HI Check: 2.238354 ,high_value: 600
## Low Check: 2.238354 ,low_value: 0
## HI Check: 1.7873 ,high_value: 600
## Low Check: 1.7873 ,low_value: 0
## HI Check: 1.546618 ,high_value: 600
## Low Check: 1.546618 ,low_value: 0
## HI Check: 1.528998 ,high_value: 600
## Low Check: 1.528998 ,low_value: 0
## HI Check: 0.7642233 ,high_value: 600
## Low Check: 0.7642233 ,low_value: 0
## HI Check: 0.9449033 ,high_value: 600
## Low Check: 0.9449033 ,low_value: 0
## HI Check: 1.417051 ,high_value: 600
## Low Check: 1.417051 ,low_value: 0
## HI Check: 0.766643 ,high_value: 600
## Low Check: 0.766643 ,low_value: 0
## HI Check: 0.2044 ,high_value: 600
## Low Check: 0.2044 ,low_value: 0
## HI Check: 0.9200659 ,high_value: 600
## Low Check: 0.9200659 ,low_value: 0
## HI Check: 0.08078943 ,high_value: 600
## Low Check: 0.08078943 ,low_value: 0
## HI Check: 0.1754239 ,high_value: 600
## Low Check: 0.1754239 ,low_value: 0
## HI Check: 0.2436841 ,high_value: 600
## Low Check: 0.2436841 ,low_value: 0
## HI Check: 1.825843 ,high_value: 600
## Low Check: 1.825843 ,low_value: 0
## HI Check: 1.204589 ,high_value: 600
```

```
## Low Check: 1.204589 ,low_value: 0
## HI Check: 0.4695431 ,high_value: 600
## Low Check: 0.4695431 ,low_value: 0
## HI Check: 0.6634483 ,high_value: 600
## Low Check: 0.6634483 ,low_value: 0
## HI Check: 3.4068 ,high_value: 600
## Low Check: 3.4068 ,low_value: 0
## HI Check: 4.810774 ,high_value: 600
## Low Check: 4.810774 ,low_value: 0
## HI Check: 2.096291 ,high_value: 600
## Low Check: 2.096291 ,low_value: 0
## HI Check: 1.062163 ,high_value: 600
## Low Check: 1.062163 ,low_value: 0
## HI Check: 1.010894 ,high_value: 600
## Low Check: 1.010894 ,low_value: 0
## HI Check: 0.4786874 ,high_value: 600
## Low Check: 0.4786874 ,low_value: 0
## HI Check: 1.918753 ,high_value: 600
## Low Check: 1.918753 ,low_value: 0
## HI Check: 1.514049 ,high_value: 600
## Low Check: 1.514049 ,low_value: 0
## HI Check: 1.281605 ,high_value: 600
## Low Check: 1.281605 ,low_value: 0
## HI Check: 0.612649 ,high_value: 600
## Low Check: 0.612649 ,low_value: 0
## HI Check: 0.4212523 ,high_value: 600
## Low Check: 0.4212523 ,low_value: 0
## HI Check: 0.831465 ,high_value: 600
## Low Check: 0.831465 ,low_value: 0
## HI Check: 0.5002823 ,high_value: 600
## Low Check: 0.5002823 ,low_value: 0
## HI Check: 0.4768143 ,high_value: 600
## Low Check: 0.4768143 ,low_value: 0
## HI Check: 0.4451957 ,high_value: 600
## Low Check: 0.4451957 ,low_value: 0
## HI Check: 0.3924166 ,high_value: 600
## Low Check: 0.3924166 ,low_value: 0
## HI Check: 0.186213 ,high_value: 600
## Low Check: 0.186213 ,low_value: 0
## HI Check: 0.1102738 ,high_value: 600
## Low Check: 0.1102738 ,low_value: 0
## HI Check: 0.05793743 ,high_value: 600
## Low Check: 0.05793743 ,low_value: 0
## HI Check: 0.5134056 ,high_value: 600
## Low Check: 0.5134056 ,low_value: 0
## HI Check: 0.2042603 ,high_value: 600
## Low Check: 0.2042603 ,low_value: 0
## HI Check: 0.4182243 ,high_value: 600
## Low Check: 0.4182243 ,low_value: 0
## HI Check: 0.624732 ,high_value: 600
## Low Check: 0.624732 ,low_value: 0
## HI Check: 2.95285 ,high_value: 600
## Low Check: 2.95285 ,low_value: 0
## HI Check: 0.9286269 ,high_value: 600
```

```
## Low Check: 0.9286269 ,low_value: 0
## HI Check: 0.441671 ,high_value: 600
## Low Check: 0.441671 ,low_value: 0
## HI Check: 0.8538273 ,high_value: 600
## Low Check: 0.8538273 ,low_value: 0
## HI Check: 0.6268195 ,high_value: 600
## Low Check: 0.6268195 ,low_value: 0
## HI Check: 1.20935 ,high_value: 600
## Low Check: 1.20935 ,low_value: 0
## HI Check: 1.629885 ,high_value: 600
## Low Check: 1.629885 ,low_value: 0
## HI Check: 1.220373 ,high_value: 600
## Low Check: 1.220373 ,low_value: 0
## HI Check: 1.964286 ,high_value: 600
## Low Check: 1.964286 ,low_value: 0
## HI Check: 1.776257 ,high_value: 600
## Low Check: 1.776257 ,low_value: 0
## HI Check: 2.514473 ,high_value: 600
## Low Check: 2.514473 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 2.352779 ,high_value: 600
## Low Check: 2.352779 ,low_value: 0
## HI Check: 2.219679 ,high_value: 600
## Low Check: 2.219679 ,low_value: 0
## HI Check: 0.1257549 ,high_value: 600
## Low Check: 0.1257549 ,low_value: 0
## HI Check: 1.187671 ,high_value: 600
## Low Check: 1.187671 ,low_value: 0
## HI Check: 2.388478 ,high_value: 600
## Low Check: 2.388478 ,low_value: 0
## HI Check: 1.913986 ,high_value: 600
## Low Check: 1.913986 ,low_value: 0
## HI Check: 1.029184 ,high_value: 600
## Low Check: 1.029184 ,low_value: 0
## HI Check: 0.5108824 ,high_value: 600
## Low Check: 0.5108824 ,low_value: 0
## HI Check: 0.2566497 ,high_value: 600
## Low Check: 0.2566497 ,low_value: 0
## HI Check: 0.735532 ,high_value: 600
## Low Check: 0.735532 ,low_value: 0
## HI Check: 2.275671 ,high_value: 600
## Low Check: 2.275671 ,low_value: 0
## HI Check: 0.2190758 ,high_value: 600
## Low Check: 0.2190758 ,low_value: 0
## HI Check: 0.8083586 ,high_value: 600
## Low Check: 0.8083586 ,low_value: 0
## HI Check: 1.27253 ,high_value: 600
## Low Check: 1.27253 ,low_value: 0
## HI Check: 1.658614 ,high_value: 600
## Low Check: 1.658614 ,low_value: 0
## HI Check: 0.3589948 ,high_value: 600
## Low Check: 0.3589948 ,low_value: 0
## HI Check: 600 ,high_value: 600
```

```
## ----------high_value: 600 ,confirmed_global: 1
## Low Check: 600 ,low_value: 0
## HI Check: 0.1113585 ,high_value: 600
## Low Check: 0.1113585 ,low_value: 0
## HI Check: 1.174848 ,high_value: 600
## Low Check: 1.174848 ,low_value: 0
## HI Check: 0.3871298 ,high_value: 600
## Low Check: 0.3871298 ,low_value: 0
## HI Check: 1.446969 ,high_value: 600
## Low Check: 1.446969 ,low_value: 0
## HI Check: 0.3476697 ,high_value: 600
## Low Check: 0.3476697 ,low_value: 0
## HI Check: 0.6421478 ,high_value: 600
## Low Check: 0.6421478 ,low_value: 0
## HI Check: 0.8793603 ,high_value: 600
## Low Check: 0.8793603 ,low_value: 0
## HI Check: 2.078183 ,high_value: 600
## Low Check: 2.078183 ,low_value: 0
## HI Check: 3.646477 ,high_value: 600
## Low Check: 3.646477 ,low_value: 0
## HI Check: 1.269157 ,high_value: 600
## Low Check: 1.269157 ,low_value: 0
## HI Check: 0.4152669 ,high_value: 600
## Low Check: 0.4152669 ,low_value: 0
## HI Check: 0.7339489 ,high_value: 600
## Low Check: 0.7339489 ,low_value: 0
## HI Check: 0.384413 ,high_value: 600
## Low Check: 0.384413 ,low_value: 0
## HI Check: 22.22222 ,high_value: 600
## Low Check: 22.22222 ,low_value: 0
## HI Check: 2.096069 ,high_value: 600
## Low Check: 2.096069 ,low_value: 0
## HI Check: 3.027946 ,high_value: 600
## Low Check: 3.027946 ,low_value: 0
## HI Check: 0.7327863 ,high_value: 600
## Low Check: 0.7327863 ,low_value: 0
## HI Check: 0.1674402 ,high_value: 600
## Low Check: 0.1674402 ,low_value: 0
## HI Check: 2.247293 ,high_value: 600
## Low Check: 2.247293 ,low_value: 0
## HI Check: 0.7040218 ,high_value: 600
## Low Check: 0.7040218 ,low_value: 0
## HI Check: 0.1086331 ,high_value: 600
## Low Check: 0.1086331 ,low_value: 0
## HI Check: 1.565936 ,high_value: 600
## Low Check: 1.565936 ,low_value: 0
## HI Check: 0.3526527 ,high_value: 600
## Low Check: 0.3526527 ,low_value: 0
## HI Check: 4.452335 ,high_value: 600
## Low Check: 4.452335 ,low_value: 0
## HI Check: 0.2547186 ,high_value: 600
## Low Check: 0.2547186 ,low_value: 0
## HI Check: 1.964034 ,high_value: 600
## Low Check: 1.964034 ,low_value: 0
```

```
## HI Check: 0.415607 ,high_value: 600
## Low Check: 0.415607 ,low_value: 0
## HI Check: 0.2119258 ,high_value: 600
## Low Check: 0.2119258 ,low_value: 0
## HI Check: 0.9722722 ,high_value: 600
## Low Check: 0.9722722 ,low_value: 0
## HI Check: 1.280639 ,high_value: 600
## Low Check: 1.280639 ,low_value: 0
## HI Check: 0.9613488 ,high_value: 600
## Low Check: 0.9613488 ,low_value: 0
## HI Check: 2.389633 ,high_value: 600
## Low Check: 2.389633 ,low_value: 0
## HI Check: 0.01905851 ,high_value: 600
## Low Check: 0.01905851 ,low_value: 0
## HI Check: 1.200614 ,high_value: 600
## Low Check: 1.200614 ,low_value: 0
## HI Check: 0.6402688 ,high_value: 600
## Low Check: 0.6402688 ,low_value: 0
## HI Check: 0.3473399 ,high_value: 600
## Low Check: 0.3473399 ,low_value: 0
## HI Check: 0.654547 ,high_value: 600
## Low Check: 0.654547 ,low_value: 0
## HI Check: 0.8257713 ,high_value: 600
## Low Check: 0.8257713 ,low_value: 0
## HI Check: 0.2673262 ,high_value: 600
## Low Check: 0.2673262 ,low_value: 0
## HI Check: 0.02844546 ,high_value: 600
## Low Check: 0.02844546 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 0.1143477 ,high_value: 600
## Low Check: 0.1143477 ,low_value: 0
## HI Check: 1.564995 ,high_value: 600
## Low Check: 1.564995 ,low_value: 0
## HI Check: 3.313 ,high_value: 600
## Low Check: 3.313 ,low_value: 0
## HI Check: 1.18343 ,high_value: 600
## Low Check: 1.18343 ,low_value: 0
## HI Check: 2.785626 ,high_value: 600
## Low Check: 2.785626 ,low_value: 0
## HI Check: 0.3523473 ,high_value: 600
## Low Check: 0.3523473 ,low_value: 0
## HI Check: 1.158596 ,high_value: 600
## Low Check: 1.158596 ,low_value: 0
## HI Check: 1.942677 ,high_value: 600
## Low Check: 1.942677 ,low_value: 0
## HI Check: 0.1502253 ,high_value: 600
## Low Check: 0.1502253 ,low_value: 0
## HI Check: 0.8344229 ,high_value: 600
## Low Check: 0.8344229 ,low_value: 0
## HI Check: 1.43086 ,high_value: 600
## Low Check: 1.43086 ,low_value: 0
## HI Check: 2.458928 ,high_value: 600
## Low Check: 2.458928 ,low_value: 0
```

```
## HI Check: 4.892176 ,high_value: 600
## Low Check: 4.892176 ,low_value: 0
## HI Check: 1.623269 ,high_value: 600
## Low Check: 1.623269 ,low_value: 0
## HI Check: 1.846559 ,high_value: 600
## Low Check: 1.846559 ,low_value: 0
## HI Check: 0.4715219 ,high_value: 600
## Low Check: 0.4715219 ,low_value: 0
## HI Check: 0.1389646 ,high_value: 600
## Low Check: 0.1389646 ,low_value: 0
## HI Check: 2.024359 ,high_value: 600
## Low Check: 2.024359 ,low_value: 0
## HI Check: 1.759741 ,high_value: 600
## Low Check: 1.759741 ,low_value: 0
## HI Check: 1.102152 ,high_value: 600
## Low Check: 1.102152 ,low_value: 0
## HI Check: 0.7124451 ,high_value: 600
## Low Check: 0.7124451 ,low_value: 0
## HI Check: 1.363152 ,high_value: 600
## Low Check: 1.363152 ,low_value: 0
## HI Check: 1.28272 ,high_value: 600
## Low Check: 1.28272 ,low_value: 0
## HI Check: 0.1746252 ,high_value: 600
## Low Check: 0.1746252 ,low_value: 0
## HI Check: 0.5165989 ,high_value: 600
## Low Check: 0.5165989 ,low_value: 0
## HI Check: 1.225919 ,high_value: 600
## Low Check: 1.225919 ,low_value: 0
## HI Check: 1.158618 ,high_value: 600
## Low Check: 1.158618 ,low_value: 0
## HI Check: 2.21645 ,high_value: 600
## Low Check: 2.21645 ,low_value: 0
## HI Check: 0.7151994 ,high_value: 600
## Low Check: 0.7151994 ,low_value: 0
## HI Check: 0.3394849 ,high_value: 600
## Low Check: 0.3394849 ,low_value: 0
## HI Check: 1.623711 ,high_value: 600
## Low Check: 1.623711 ,low_value: 0
## HI Check: 0.07703685 ,high_value: 600
## Low Check: 0.07703685 ,low_value: 0
## HI Check: 0.788551 ,high_value: 600
## Low Check: 0.788551 ,low_value: 0
## HI Check: 0.5314983 ,high_value: 600
## Low Check: 0.5314983 ,low_value: 0
## HI Check: 0.6225839 ,high_value: 600
## Low Check: 0.6225839 ,low_value: 0
## HI Check: 4.980969 ,high_value: 600
## Low Check: 4.980969 ,low_value: 0
## HI Check: 2.52258 ,high_value: 600
## Low Check: 2.52258 ,low_value: 0
## HI Check: 0.7513066 ,high_value: 600
## Low Check: 0.7513066 ,low_value: 0
## HI Check: 0.8676491 ,high_value: 600
## Low Check: 0.8676491 ,low_value: 0
```

```
## HI Check: 2.504319 ,high_value: 600
## Low Check: 2.504319 ,low_value: 0
## HI Check: 7.860064 ,high_value: 600
## Low Check: 7.860064 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 1.702499 ,high_value: 600
## Low Check: 1.702499 ,low_value: 0
## HI Check: 0.8808453 ,high_value: 600
## Low Check: 0.8808453 ,low_value: 0
## HI Check: 0.3219367 ,high_value: 600
## Low Check: 0.3219367 ,low_value: 0
## HI Check: 5.505769 ,high_value: 600
## Low Check: 5.505769 ,low_value: 0
## HI Check: 0.1772351 ,high_value: 600
## Low Check: 0.1772351 ,low_value: 0
## HI Check: 0.7028 ,high_value: 600
## Low Check: 0.7028 ,low_value: 0
## HI Check: 1.971752 ,high_value: 600
## Low Check: 1.971752 ,low_value: 0
## HI Check: 0.7173582 ,high_value: 600
## Low Check: 0.7173582 ,low_value: 0
## HI Check: 0.5892651 ,high_value: 600
## Low Check: 0.5892651 ,low_value: 0
## HI Check: 0.7361153 ,high_value: 600
## Low Check: 0.7361153 ,low_value: 0
## HI Check: 0.07733492 ,high_value: 600
## Low Check: 0.07733492 ,low_value: 0
## HI Check: 2.293095 ,high_value: 600
## Low Check: 2.293095 ,low_value: 0
## HI Check: 2.548896 ,high_value: 600
## Low Check: 2.548896 ,low_value: 0
## HI Check: 0.5955152 ,high_value: 600
## Low Check: 0.5955152 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 1.082665 ,high_value: 600
## Low Check: 1.082665 ,low_value: 0
## HI Check: 2.128483 ,high_value: 600
## Low Check: 2.128483 ,low_value: 0
## HI Check: 2.088314 ,high_value: 600
## Low Check: 2.088314 ,low_value: 0
## HI Check: 0.2230318 ,high_value: 600
## Low Check: 0.2230318 ,low_value: 0
## HI Check: 0.307377 ,high_value: 600
## Low Check: 0.307377 ,low_value: 0
## HI Check: 0.8497982 ,high_value: 600
## Low Check: 0.8497982 ,low_value: 0
## HI Check: 0.8761123 ,high_value: 600
## Low Check: 0.8761123 ,low_value: 0
## HI Check: 0.1175648 ,high_value: 600
## Low Check: 0.1175648 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
```

```
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 0.5432389 ,high_value: 600
## Low Check: 0.5432389 ,low_value: 0
## HI Check: 0.1886199 ,high_value: 600
## Low Check: 0.1886199 ,low_value: 0
## HI Check: 0.3051989 ,high_value: 600
## Low Check: 0.3051989 ,low_value: 0
## HI Check: 0.2425027 ,high_value: 600
## Low Check: 0.2425027 ,low_value: 0
## HI Check: 0.5702067 ,high_value: 600
## Low Check: 0.5702067 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 0.57918 ,high_value: 600
## Low Check: 0.57918 ,low_value: 0
## HI Check: 0.9004922 ,high_value: 600
## Low Check: 0.9004922 ,low_value: 0
## HI Check: 0.736438 ,high_value: 600
## Low Check: 0.736438 ,low_value: 0
## HI Check: 0.6515501 ,high_value: 600
## Low Check: 0.6515501 ,low_value: 0
## HI Check: 0.1165307 ,high_value: 600
## Low Check: 0.1165307 ,low_value: 0
## HI Check: 1.060196 ,high_value: 600
## Low Check: 1.060196 ,low_value: 0
## HI Check: 0.374651 ,high_value: 600
## Low Check: 0.374651 ,low_value: 0
## HI Check: 0.8116855 ,high_value: 600
## Low Check: 0.8116855 ,low_value: 0
## HI Check: 0 ,high_value: 600
## Low Check: 0 ,low_value: 0
## HI Check: 18.07451 ,high_value: 600
## Low Check: 18.07451 ,low_value: 0
## HI Check: 1.182333 ,high_value: 600
## Low Check: 1.182333 ,low_value: 0
## HI Check: 2.145863 ,high_value: 600
## Low Check: 2.145863 ,low_value: 0
```

```
#cat("HiMaxConfirmed:", HiMaxConfirmed,",HiColNum:",HiColNum,",LoMaxConfirmed:",LoMaxConfirmed,", LoCol

#cat("A:",((deaths_global[HiColNum,1147]/confirmed_global[HiColNum, 1147])*100),"\n")

#cat("confirmed_global:",(confirmed_global[HiColNum, 1147])*100,"\n")

#cat("B:",confirmed_global[LoColNum,1147],"error:",((deaths_global[LoColNum,1147]/confirmed_global[LoCo
```

Objective 5 *You are asked to make two tables with the top 5 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.

GitHub Log

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

```
##
## Subject: Finished Code for Ob2, need code comments
## Author: Sal - Figgs0bit
## Date: Fri, 28 Mar 2025 12:23:15 -0700
## Body:
##
## Subject: Ob1 code done, need to update code comments
## Author: Sal - Figgs0bit
## Date: Fri, 28 Mar 2025 03:30:43 -0700
## Body:
##
## Subject: R ob1 chunk
## Author: Sal - Figgs0bit
## Date: Fri, 28 Mar 2025 02:12:28 -0700
## Body:
##
## Subject: Objective 1, Deaths finished. Need Confirmation for loops.
## Author: Sal - Figgs0bit
## Date: Mon, 24 Mar 2025 20:39:21 -0700
## Body:
##
## Subject: Merge branch 'main' of github.com:Figgs0bit/CSIT165-CovidGroupProj
## Author: Sal - Figgs0bit
## Date: Mon, 24 Mar 2025 17:34:18 -0700
## Body:
##
## Subject: added {r Data setup} chunk. Downloads csv from repository to dataframe
## Author: Sal - Figgs0bit
## Date: Mon, 24 Mar 2025 17:34:05 -0700
## Body:
##
## Subject: Uploaded data
## Author: Figgs0bit
## Date: Mon, 24 Mar 2025 17:14:29 -0700
## Body: time_series_covid19 data (global deaths, global recovered)
##
## Subject: Merge branch 'main' of github.com:Figgs0bit/CSIT165-CovidGroupProj
## Author: Sal - Figgs0bit
## Date: Mon, 24 Mar 2025 17:12:01 -0700
## Body:
##
## Subject: Template knited to PDF w/o (\usepackage{tabu}, library(kableExtra))
## Author: Sal - Figgs0bit
## Date: Mon, 24 Mar 2025 17:11:54 -0700
## Body:
##
## Subject: Update README.md
```

```
## Author: Figgs0bit
## Date: Mon, 24 Mar 2025 17:10:19 -0700
## Body: Contains copy of Group Contract
##
## Subject: First Commit: setting up Github repository need to add template
## Author: Sal - Figgs0bit
## Date: Mon, 24 Mar 2025 17:05:25 -0700
## Body:
##
## Subject: Initial commit
## Author: Figgs0bit
## Date: Mon, 24 Mar 2025 17:00:21 -0700
## Body:
```