Group Project: Covid Origins

Name: Sal Figueroa
Partner: Kylie Stearns

2025-04-03

Contents

Background 1
Data 1
Project Objectives 1
Objective 1 1
Objective 2 2
Objective 3 4
Objective 4 4
GitHub Log 9

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
```

```
#For loop starting it column 5 to final column using nrow().
for(i in 1:nrow(deaths_global[,])) #loops 289 times through every row on 1st day
{
num <- deaths_global[i,5] + confirmed_global[i,5]#Holds value to be tested against-
                                                 #the current max value.

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

#holds col row info. for ob3
ob1Loc_PS <- deaths_global[RowMax,1]
ob1Loc_CR <- deaths_global[RowMax,2]
ob1Loc_Lat <- deaths_global[RowMax,3]
ob1Loc_Long  <- deaths_global[RowMax,4]

#Prints out location of most deatch and confirmations
deathNconfimred <- deaths_global[RowMax,5] + confirmed_global[RowMax,5]
cat("Province/State:",deaths_global[RowMax,1],"Country/Region:",deaths_global[RowMax,2],"\n")
```

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

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

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

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

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

```
cat("Predicted area of Origin had the most initial deaths and confirmed cases on (01.22.2020):"
    ,deathNconfimred,"\n")
```

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

```
RowMax <- nrow(confirmed_global[,]) #limit Qty of Rows 289
ColMax <- ncol(confirmed_global[,]) #limit QTy of Columns 1147

ColName <- names(confirmed_global) #Loads the the column names into a vector to print date
NoConfirmRow <- c() # creates an empty vector to hold excluded rows to avoid

for(i in 1:RowMax)#loops 289x
  {
  if(confirmed_global[i,ColMax] == 0) #loops 1147, loop finds rows that never had a-
                                      #confirmed case and excludes them later.
    {NoConfirmRow <- c(NoConfirmRow,i)} #Loads row numbers to be used as filter later.
  }

#initiates the loop counter and used for pulling data from data frame.
```

```
i <- 1 #COl counter 1147
j <- ColMax #row 289, Start the loop at last column (1147)


#The while loop stops running when the num variable loads a value > 0
#set initial condition to enter while loop. This allows the loop to stop once it-
#encounter the first (0) in the data frame.
num <- 1
while(j < ColMax+1 && num > 0) #Iterates through Columns back wards 1147
  {
    while(i < RowMax && num > 0) #Iterates through rows (289) top-to-bottom.
      {
        #This if conditional serves to filter out rows that never had a confirmation
        if(i==NoConfirmRow[1] || i==NoConfirmRow[2])
          {
          #prevents the num variable loading one of these rows.
          }
        else if(confirmed_global[i,j] == 0) #This if conditional serves to exit the entire-
          {                                 #while loop when it encounters it's first zero
          num <- confirmed_global[i,j]#holds the variable to be tested
          iPT <- i #holds row position of first encountered zero location
          jPT <- j+1 #holds Column position of that zero, increments +1
          num <- 0 #CRITICAL! First time the loop-
          }        #ends num sets to (0) this stops the all the loops
          i <- i + 1 #Row, Serves as the while loops increment line
      }
    i <- 1 #The nature of the nested while loops requires the inner loop to reset.
    j <- j - 1 #Col, Serves as the while loops de-increment line
  }

cat("Row",iPT,"Col",jPT,"Cell Found\n") #VERIFY! Row:276, Col: 915 --Pitcairn IslandS
```

```
## Row 276 Col 915 Cell Found
```

```
#holds info. for ob3
ob2Loc_PS <- confirmed_global[iPT,1]
ob2Loc_CR <- confirmed_global[iPT,2]
ob2Loc_Lat <- confirmed_global[iPT,3]
ob2Loc_Long  <- confirmed_global[iPT,4]

#print location from loop
cat("Province/State: ",confirmed_global[iPT,1],"Country/Region: ",confirmed_global[iPT,2],"\n")
```

```
## Province/State:  Pitcairn Islands Country/Region:  United Kingdom
```

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

```
## Latitude:  -24.3768  ,Longitude: -128.3242
```

```
cat("Here is the most recent area to have a first confirmed case.", ColName[iPT] ,"-\n")
```

```
## Here is the most recent area to have a first confirmed case. X10.19.20 -
```

```
cat("at:",confirmed_global[iPT,jPT],"\n")
```

```
## at: 4
```

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}.

```
#Calculate the distance using distm() and distHaversine.
distance_meters <- distm(c(ob1Loc_Long, ob1Loc_Lat), c(ob2Loc_Long, ob2Loc_Lat), fun = distHaversine)

#print information
cat("From (ob1):\n")
```

```
## From (ob1):
```

```
 #prints Province/State and Country/Region location
cat("Province/State:",ob1Loc_PS,"Country/Region:",ob1Loc_CR,"\n")
```

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

```
#Prints lat and long 1st (ob1) location
cat("Latitude:",ob1Loc_Lat,",Longitude:",ob1Loc_Long,"\n\n")
```

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

```
cat("To (ob2):\n")
```

```
## To (ob2):
```

```
#prints Province/State and Country/Region location
cat("Province/State:",ob2Loc_PS,"Country/Region:",ob2Loc_CR,"\n")
```

```
## Province/State: Pitcairn Islands Country/Region: United Kingdom
```

```
#Prints lat and long 2nd location
cat("Latitude:",ob2Loc_Lat,",Longitude:",ob2Loc_Long,"\n\n")
```

```
## Latitude: -24.3768 ,Longitude: -128.3242
```

```
#Print the distance in meters
cat("Distance between locations:",round(distance_meters), "- meters\n")
```

```
## Distance between locations: 14090120 - meters
```

```
#Print the distance in Kilometers
cat("Distance between locations:",round(distance_meters/1000),"- Kilometers\n" )
```

```
## Distance between locations: 14090 - Kilometers
```

```
#1 mile = 1609.34 meters
cat("Distance between locations:",round(distance_meters/1609.34),"- miles\n" )
```

```
## Distance between locations: 8755 - miles
```

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)?

```
#Row and Column variables
iCntLength <- ncol(deaths_global)#1147 number of columns
jCntLength <- nrow(deaths_global)#289 number of rows

#Variables for holding highest and lowest Risk score
low_value <- 10
high_value <- 0

#creates vectors to his and lows
#Hold the deaths_global value of each separate area of the world
VSUMdeaths_global <- 1:jCntLength
#Hold the confirmed_global value of each separate area of the world
VSUMconfirmed_global <- 1:jCntLength
#Hold the risk score value of each separate area of the world
VSUMRiskScore <- 1:jCntLength

# removes NA values
deaths_global[is.na(deaths_global)] <- 0
confirmed_global[is.na(confirmed_global)] <- 0

#Running sum defaults
SUMdeaths_global <- 0
SUMconfirmed_global <- 0
SUMRiskScore  <- 0
emptyClr <- 0 #serves to remove rows(regions) without lat/long coordinates.

#For loop iterates through the entire data frame
for(j in 1:jCntLength) #Iterate through rows 1-289
  {
      #Running sum defaults are reset every 1142 loops as every 1142 loops accounts for a region.
      SUMdeaths_global <- 0
      SUMconfirmed_global <- 0
      SUMRiskScore  <- 0

  for(i in 5:iCntLength) #Iterate through columns 1-1142 as we start at column 5
    {
      #This variable filters out locations with zero coordinates
      #This also filters out where confirmed_global values are (0) as to avoid division by zero
      emptyClr <- (deaths_global[j,3] + deaths_global[j,4])

      if(emptyClr > 0 && confirmed_global[j,i] > 0)
        {
          #holds the sum of every death per row or city, country.
          SUMdeaths_global <- SUMdeaths_global + deaths_global[j,i]
```

```
          #holds the sum of every confirmation in data frame.
          SUMconfirmed_global <- SUMconfirmed_global + confirmed_global[j,i]
          #Risk score eqn.=(deaths/confirmations)x100#Risk score eqn.=(deaths/confirmations)x100
          SUMRiskScore <- (SUMdeaths_global/SUMconfirmed_global)*100
        }
       #also filters out locations with zero coordinates
       else if(emptyClr > 0 && confirmed_global[j,i] == 0)
         {SUMRiskScore <-0} #As confirmed_global is a denominator in the risk-
    }                       #score equation it can never equal zero.
     #This loads the each vector with rolling derived risk score equation values.
     VSUMdeaths_global[j] <- SUMdeaths_global
     VSUMconfirmed_global[j] <- SUMconfirmed_global
     #Vector (length 1:289) Holds the risk score (all time) for each location.
     VSUMRiskScore[j] <- SUMRiskScore
}
```

Objective 4.1

```
#holds the low/Hi risk score values for the IF condition statements.
low_value <- min(VSUMRiskScore)
high_value <- max(VSUMRiskScore)

#Holds the rotating low/hi values of the for loop
HiMaxConfirmed <- 0
LoMaxConfirmed <- 1

for(j in 1:jCntLength) #Iterate through rows 1-289
{
      chkRiskScore <- VSUMRiskScore[j]#Regions risk score to be checked

      #Nested IF statements serve to find which common high risk score value has the highest confirmed_
      if(chkRiskScore == high_value)
      {
        if(VSUMconfirmed_global[j] > HiMaxConfirmed)
        {
        HiMaxConfirmed <- VSUMconfirmed_global[j]
        HiColNum <- j #saves the location by recording what row
        }
      }
      #Nested IF statements serve to find which common low risk score value has the highest confirmed_g
      else if(chkRiskScore == low_value)
     {
     if(VSUMconfirmed_global[j] < LoMaxConfirmed)
        {
        LoMaxConfirmed <- VSUMconfirmed_global[j]
        LoColNum <- j #saves the location by recording what row
        }
     }
}
#prints results
cat("Global Risk Score:",mean(VSUMRiskScore),",Max Risk Score:",max(VSUMRiskScore),",Min Risk Score:",m
```

Objective 4.2

```
## Global Risk Score: 2.895648 ,Max Risk Score: 600 ,Min Risk Score: 0
```

```
cat("Highest Risk Score:",VSUMRiskScore[HiColNum]," - Region(Row) w/ most confrimations:",HiColNum,"\n")
```

```
## Highest Risk Score: 600  - Region(Row) w/ most confrimations: 162
```

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

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

```
cat("Lat:", deaths_global[HiColNum, 3],",Long:",deaths_global[HiColNum, 4],"\n\n")
```

```
## Lat: 40.3399 ,Long: 127.5101
```

```
cat("Lowest Risk Score:",VSUMRiskScore[LoColNum]," - Region(Row) w/ most confrimations:", LoColNum,"\n")
```

```
## Lowest Risk Score: 0  - Region(Row) w/ most confrimations: 6
```

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

```
## Province/State:  ,Country/Region: Antarctica
```

```
cat("Lat:", deaths_global[LoColNum, 3],",Long:",deaths_global[LoColNum, 4],"\n\n")
```

```
## Lat: -71.9499 ,Long: 23.347
```

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.

```
#j_Row <- c(vector, add)

#create empty vectors
RecentDeath <- c() #Will Hold all 289 entries of deaths_global from (3/9/2023)
RecentConfirm <- c() #Will Hold all 289 entries of confirmed_global from (3/9/2023)
V_comfirmedPdeaths <- c() #Will Hold all 289 entries of the sum of deaths_global+confirmed_global
V_Bloc <- character() #Will Hold all 289 entries of the column 3, Country/Region
sumCnD <- 0
iCntLength <- ncol(deaths_global)#1147 number of columns
i <- iCntLength # postion of final column (3/9/2023)
jCntLength <- nrow(deaths_global)#289 number of rows


#For loop arrangement goes through an entire column and while loading data into vectors.
for(j in 1:jCntLength) #Iterate through rows 1-289
  {
    Drecent <- deaths_global[j,i] #deaths_global tmp var
    CRecent <- confirmed_global[j,i] #confirmed_global tmp var
    DCtot <- deaths_global[j,i] + confirmed_global[j,i] #deaths_global + confirmed_global tmp var
    Bloc <- deaths_global[j,2] #country/region tmp var

    #Populate vectors
    V_comfirmedPdeaths <- c(V_comfirmedPdeaths, DCtot) #adds deaths_global + confirmed_global to new ve
    V_Bloc <- c(V_Bloc, Bloc) #adds Country/Region to new vector
```

```
  }

#These nmew vectors hold the shorten data frame after consolidating common country data.
Final_comfirmedPdeaths <- c() #Will Hold all 289 entries of the sum of deaths_global+confirmed_global
Final_Bloc <- character() #Will Hold all 289 entries of the column 3, Country/Region

#for loop consolidates all multiple country data rows
for(j in 1:jCntLength) #Iterate through rows 1-289
{
  #The following if else ladder sorts and filter the duplicate countries and combine the data.
  if(V_Bloc[j] == V_Bloc[1]) #This if conditional is need as the the first row will not fall through th
  {
    Final_comfirmedPdeaths <- c(Final_comfirmedPdeaths, V_comfirmedPdeaths[j]) #adds deaths_global + co
    Final_Bloc <- c(Final_Bloc, deaths_global[j,2]) #adds Country/Region to new vector
  }
  else if(V_Bloc[j] %in% V_Bloc[j+1]) #sums the commmon country data
    {
    sumCnD <- sumCnD + V_comfirmedPdeaths[j]
    }
  else if((V_Bloc[j] %in% V_Bloc[j-1]) && (V_Bloc[j] != V_Bloc[j+1])) #loads the combine data under one
    {
    sumCnD <- sumCnD + V_comfirmedPdeaths[j]
    Final_comfirmedPdeaths <- c(Final_comfirmedPdeaths, sumCnD) #adds deaths_global + confirmed_global
    Final_Bloc <- c(Final_Bloc, deaths_global[j,2]) #adds Country/Region to new vector
    sumCnD <- 0
    }
  else
   {
    Final_comfirmedPdeaths <- c(Final_comfirmedPdeaths, V_comfirmedPdeaths[j]) #adds deaths_global + co
    Final_Bloc <- c(Final_Bloc, deaths_global[j,2]) #adds Country/Region to new vector
   }
  }

#creates a data frame with country/region and (deaths_global + confirmed_global) values.
df_comfirmedPdeaths <- data.frame(Final_Bloc, Final_comfirmedPdeaths)
RowCut <- nrow(df_comfirmedPdeaths) #this creates a variable need to cut/delete unneeded rows.

df_decreasing <- df_comfirmedPdeaths[order(df_comfirmedPdeaths$Final_comfirmedPdeaths, decreasing = TRU
top5 <- df_decreasing[-(6:RowCut),] #removes the rows starting at row 6.
#creates table for "TOP 5: Recent Comfirmations & Deaths"
knitr::kable(top5, "pipe", col.name=c('Country/Region','comfirmed & deaths'),caption = "TOP 5: Recent C
```

Table 1: TOP 5: Recent Comfirmations & Deaths

	Country/Region	comfirmed & deaths
187	US	104926538
81	India	45221517
64	France	40032894
68	Germany	38417995
25	Brazil	37775329

```
df_increasing <- df_comfirmedPdeaths[order(df_comfirmedPdeaths$Final_comfirmedPdeaths, decreasing = FALS
bot5 <- df_increasing[-(6:RowCut),] #removes the rows starting at row 6.
#creates table for "TOP 5: Recent Comfirmations & Deaths"
knitr::kable(bot5, "pipe", col.name=c('Country/Region','comfirmed & deaths'),caption = "BOTTOM 5: Recen
```

Table 2: BOTTOM 5: Recent Comfirmations & Deaths

	Country/Region	comfirmed & deaths
94	Korea, North	7
6	Antarctica	11
108	MS Zaandam	11
77	Holy See	29
198	Winter Olympics 2022	535

GitHub Log

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

```
##
## Subject: Go over comments one last time. Tables FINALLY knitted.
## Author: Sal - Figgs0bit
## Date: Thu, 3 Apr 2025 20:43:02 -0700
## Body:
##
## Subject: OB5 finished, need to double check code
## Author: Sal - Figgs0bit
## Date: Thu, 3 Apr 2025 16:06:49 -0700
## Body:
##
## Subject: Still working on ob5
## Author: Sal - Figgs0bit
## Date: Thu, 3 Apr 2025 01:00:31 -0700
## Body:
##
## Subject: Correcting errors in OB2
## Author: Sal - Figgs0bit
## Date: Tue, 1 Apr 2025 21:07:45 -0700
## Body:
##
## Subject: Finished code, now adding comments and double checking
## Author: Sal - Figgs0bit
## Date: Tue, 1 Apr 2025 11:34:12 -0700
## Body:
##
## Subject: Working on ob5
## Author: Sal - Figgs0bit
## Date: Mon, 31 Mar 2025 21:23:53 -0700
## Body:
##
## Subject: Finished ob4.1, ob4.2
## Author: Sal - Figgs0bit
## Date: Mon, 31 Mar 2025 13:56:47 -0700
## Body:
```

```
##
## Subject: Finished ob3 coordinates, all comments up to date. Ob4.1 & ob4.2 need corrections.
## Author: Sal - Figgs0bit
## Date: Sun, 30 Mar 2025 18:06:18 -0700
## Body:
##
## Subject: code and Comments for ob2 updated
## Author: Sal - Figgs0bit
## Date: Sun, 30 Mar 2025 08:09:05 -0700
## Body:
##
## Subject: Updateding notes, ob3 and ob 5 needed.
## Author: Sal - Figgs0bit
## Date: Sun, 30 Mar 2025 05:49:15 -0700
## Body:
##
## 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:
```