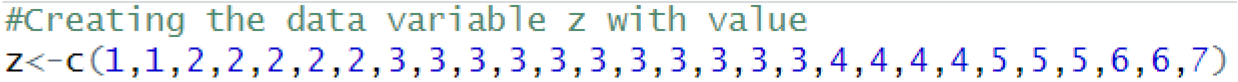
Complete these works showing your codes and outputs from R studio:

Work 1: See slide 25 of session 2 slide deck and provide answers here.

Show the histogram of the z variable and interpret it carefully.

#Creating the data variable z with value

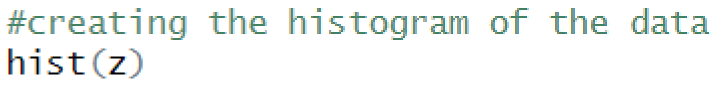
z<-c(1,1,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,5,5,5,6,6,7)



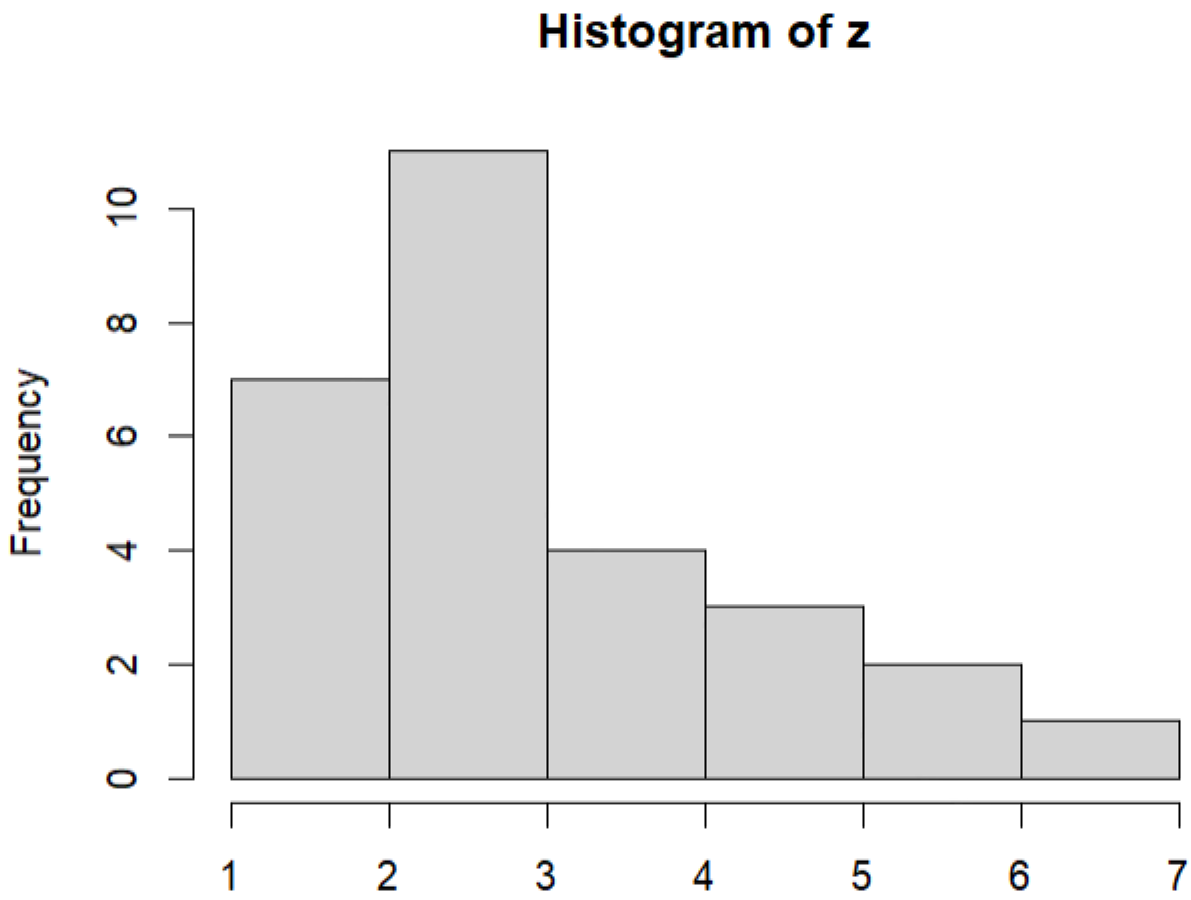
```
#Creating the data variable z with value
z<-c(1,1,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,5,5,5,6,6,7)
```

We can create the histogram of the individual variable using the hist() function. For the variable z the histogram is created by:

hist(z)

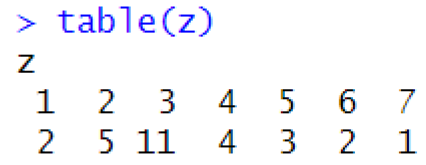


```
#creating the histogram of the data
hist(z)
```



This is the histogram graph of the data distribution of the z variable. From the histogram we can interpret that the histogram is the **Right Skewed Histogram** as the pick of the graph lies in the left side.

We can also create the frequency table of the data by using table() Function. This function will give the occurrence of the specific elements in the data.

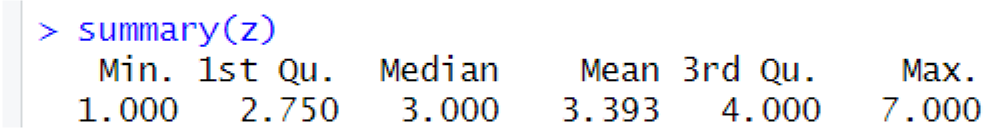


```
> table(z)
z
 1  2  3  4  5  6  7 
 2  5 11  4  3  2  1 
```

From this we can interpret that the occurrence of element 1 is 2 times whereas the occurrence of element 7 is 1 times and the mostly occurring element is 3

Get a summary of this variable and decide which measure of central tendency and measure of dispersion must be used for this variable?

We can get the summary of this variable by using the summary() function. The Summary function will give us the value of Minimum, Q1, median , mean , Q3 and Maximum Value from the data.

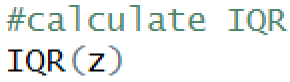


```
> summary(z)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   2.750   3.000   3.393   4.000   7.000 
```

From the above we can see the summary of the data variable z. Here the median is less than mean. Mean is used for normally distributed data which generate exactly bell-like curves but as from our histogram we have **Right Skewed Histogram** we have to use median. So we will be using Median as the measures of Central Tendency and Range as the measures of dispersion for this data.

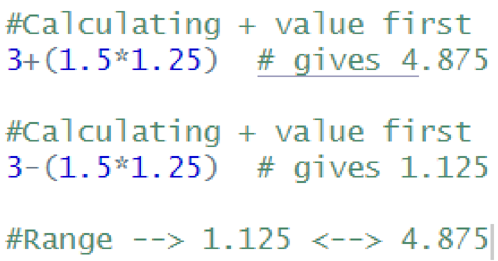
We have for the formula when we have to choose median

median +-1.5*IQR this will give 95% of data, so we need to calculate the range but before that we need to calculate IQR. To calculate IQR(Inter quartile range we have IQR() Function.



```
#calculate IQR
IQR(z)
```

This Gives IQR = 1.25 Now lets calculate Outlier Range



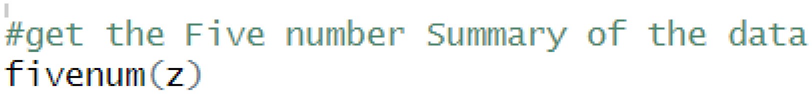
```
#Calculating + value first
3+(1.5*1.25)  # gives 4.875

#Calculating + value first
3-(1.5*1.25)  # gives 1.125

#Range --> 1.125 <--> 4.875
```

From this calculation we can figure out that we have outliers in our data on the right side because we have the element above the range as well.

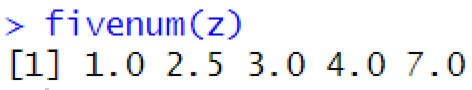
Get the five number summary of this variable and interpret it carefully

A five number summary is especially useful in descriptive analysis of the data. It is used to investigate the large data set in the early stage to understand the data. The summary gives the five values (the maximum and minimum values), the lower and upper quartiles, and the median.

We can get the five number summary by fivenum() function.



```
#get the Five number Summary of the data
fivenum(z)
```

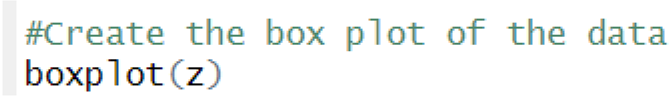


```
> fivenum(z)
[1] 1.0 2.5 3.0 4.0 7.0
```

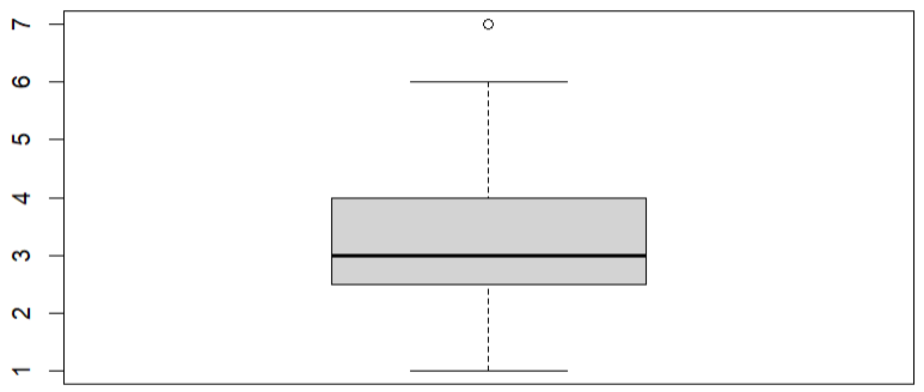
Min Q1 M Q3 Max

Create the box plot of this variable and interpret it carefully

We can create the boxplot graph by using the boxplot() function.



```
#Create the box plot of the data
boxplot(z)
```



Do you get an outlier for this variable in box plot? Why ?

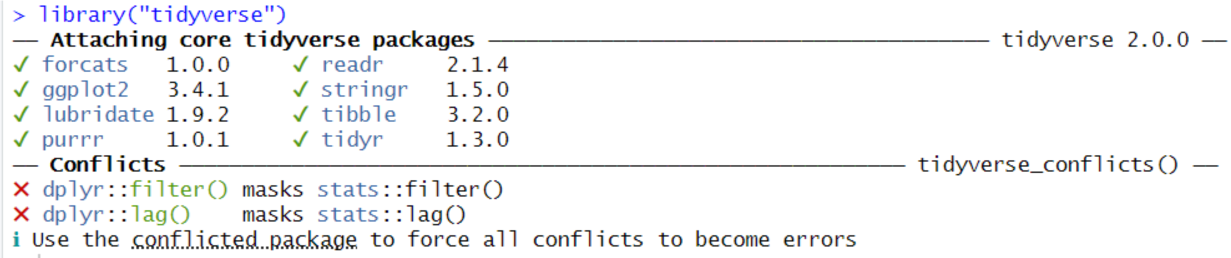
Yes we get the outlier for the variable Z. From the graph we can see the outlier at the end of data 7. It is represented by “o”. Here we have one outlier which is away from Q3. Here Element 7 is the outlier.. From the graph we have an outlier in only one end. If we have an outlier in both ends then “o” is presented in both ends of the graph.

Work 2: See slide 26-30 of session 2 slide deck and provide answers here. Data is attached.

First of all we need to import the data in R Studio. So in order to import the data in R Studio we have two options: one by using the IDE option and another by using the readr package. For that purpose we Install the package tidyverse. It includes mostly used other libraries like ggplot2, readr. readr is used for data import.

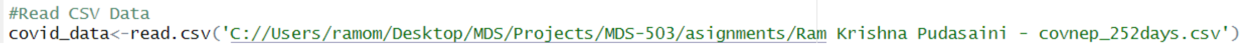
Install tidyverse

install.packages("tidyverse")
library("tidyverse")



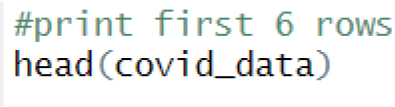
```
> library("tidyverse")
── Attaching core tidyverse packages ──────────────────────────── tidyverse 2.0.0 ──
✓ forcats   1.0.0     ✓ readr     2.1.4
✓ ggplot2   3.4.1     ✓ stringr   1.5.0
✓ lubridate 1.9.2     ✓ tibble    3.2.0
✓ purrr     1.0.1     ✓ tidyr     1.3.0
── Conflicts ─────────────────────────────────────────── tidyverse_conflicts() ──
✗ dplyr::filter() masks stats::filter()
✗ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package to force all conflicts to become errors
```

#importing the csv data in R studio using readr package function
We will use the read_csv() function to read the csv file and we need to give the path to the file.



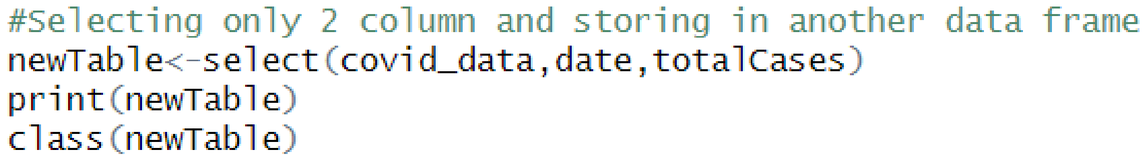
```
#Read CSV Data
covid_data<-read.csv('C://Users/ramom/Desktop/MDS/Projects/MDS-503/asignments/Ram Krishna Pudasaini - covnep_252days.csv')
```

Read the first 6 row of the data
#print first 6 rows
head(covid_data)



```
#print first 6 rows
head(covid_data)
```

In order to work with total cases we only extra two column from the data frame covid_data
And named it as newTable. For that we have used select ().
#Selecting only 2 column and storing in another data frame
newTable<-select(covid_data,date,totalCases)
print(newTable)

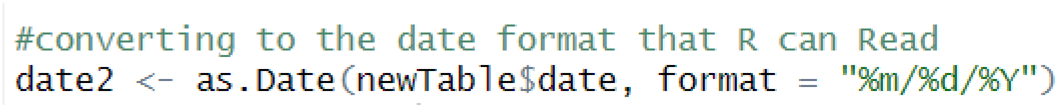


```
#Selecting only 2 column and storing in another data frame
newTable<-select(covid_data,date,totalCases)
print(newTable)
class(newTable)
```

In order to plot the data related to total cases based on the date we will be using the following command
plot(covid_data$date,covid_data$totalCases) #this gives error

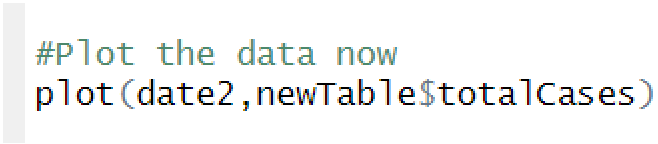
When executing this command we get an error that the x axis is unable to get the plot, (unable to plot on x axis, date as function data type)so we format the date such that the R studio will understand the date format we used and the class should be date type.

For that we use another variable date2 to store date value

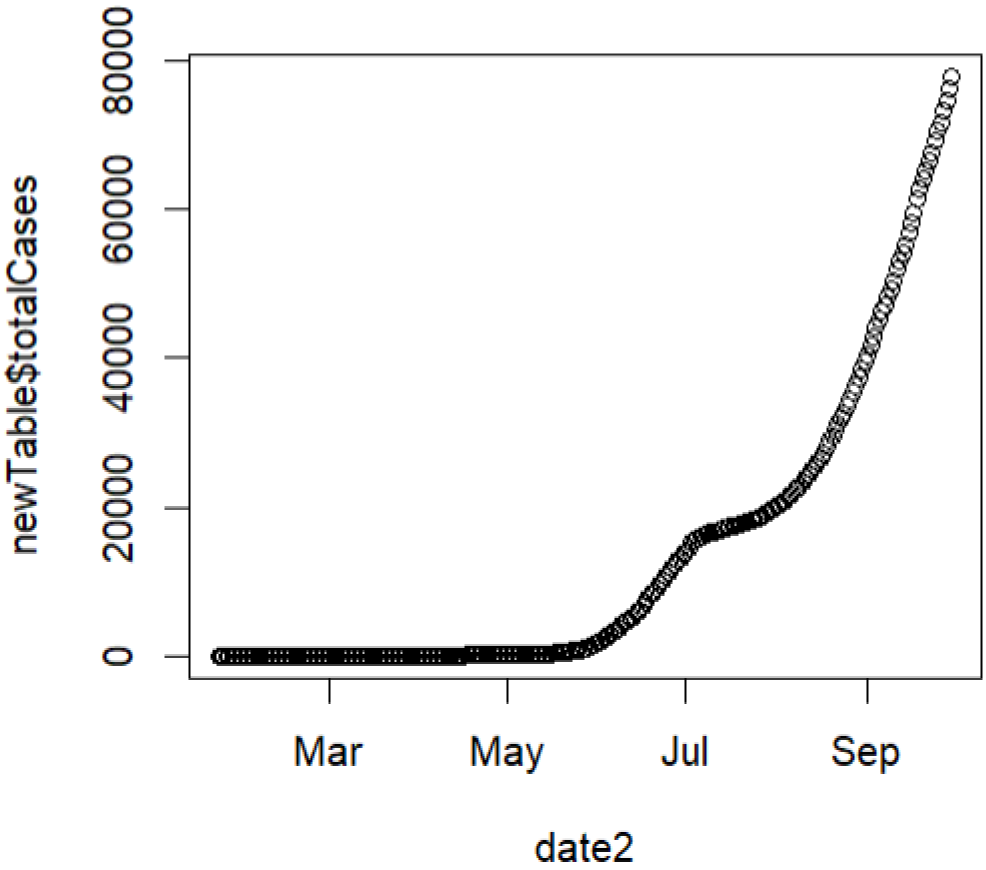


```
#converting to the date format that R can Read
date2 <- as.Date(newTable$date, format = "%m/%d/%Y")
```

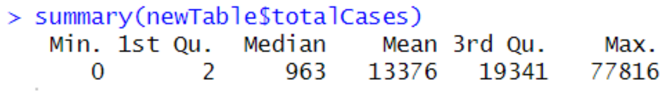
Now plotting the data



```
#Plot the data now
plot(date2,newTable$totalCases)
```

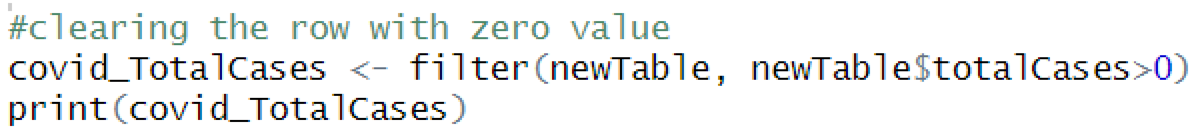


Now creating the Summary of the data we get



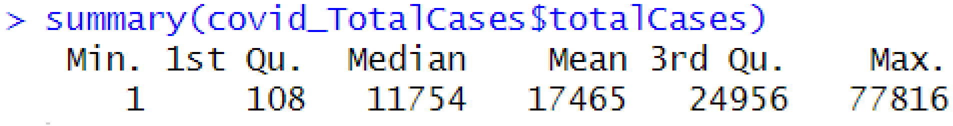
```
> summary(newTable$totalCases)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      0       2     963   13376   19341   77816 
```

From the summary we can see that the min value is zero and the max value is 77816. The data is distributed in a skewed manner as the histogram gives the Right Skewed distribution of the data. As the tail is big in the left hand side we need to change the data a little bit by removing the null values for the Total cases. In order to manipulate the data we use dplyr library so we load this library and use the filter function to filter out the null value row from our data.



```
#clearing the row with zero value
covid_TotalCases <- filter(newTable, newTable$totalCases>0)
print(covid_TotalCases)
```

Now getting the summary again for the data



```
> summary(covid_TotalCases$totalCases)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      1     108   11754   17465   24956   77816 
```

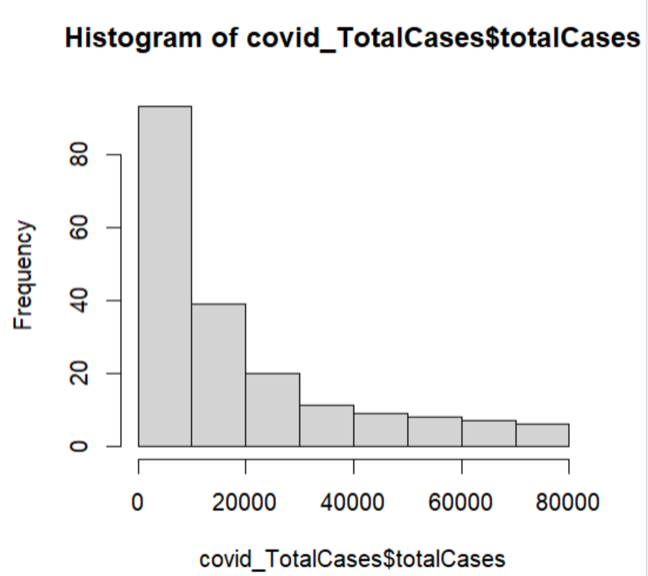
Here we will be using the median as the measurement of central tendency and range to find the outliers. We have median +- 1.5(IQR)

11754+(1.5*IQR(covid_TotalCases$totalCases)) → we get positive range 49026
11754+(1.5*IQR(covid_TotalCases$totalCases)) →we get negative range -22518

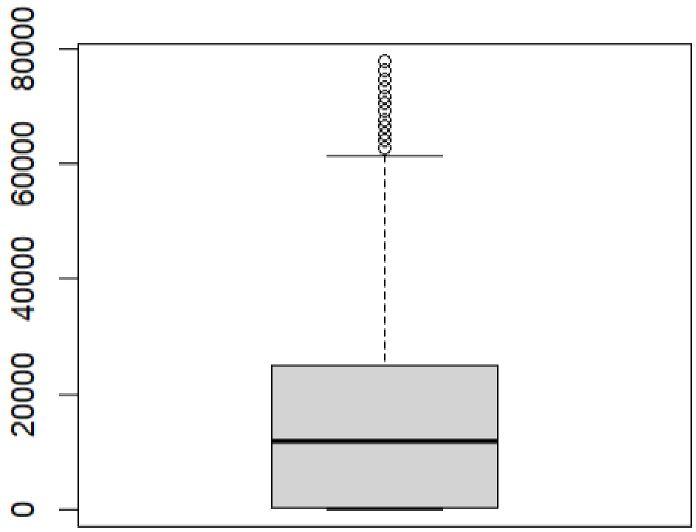
Range -22518 <—> 49026

We have outlier above 50 thousand

Getting the Histogram graph for total cases



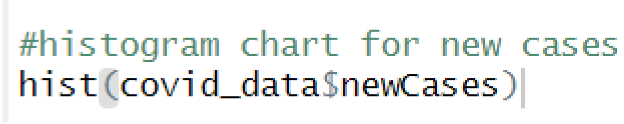
Creating the box plot
#boxplot
boxplot(covid_TotalCases$totalCases)



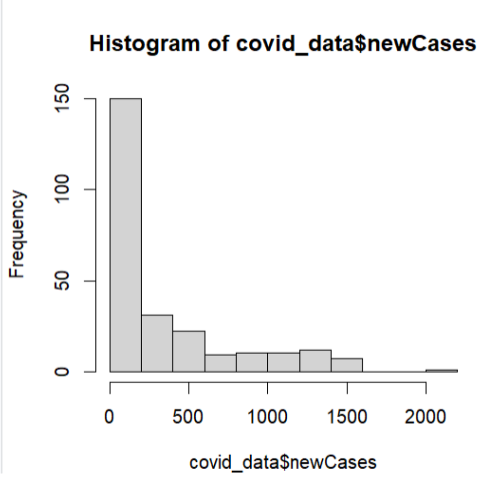
From the box plot we can see the outliers far away from the third quartile.

****

Histogram data for new cases



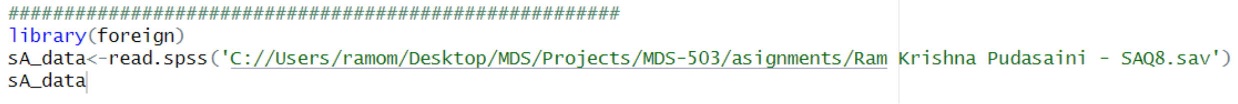
```
#histogram chart for new cases
hist(covid_data$newCases)
```



The histogram shows that the new cases 0 was 150 times and 2000 just for one times.

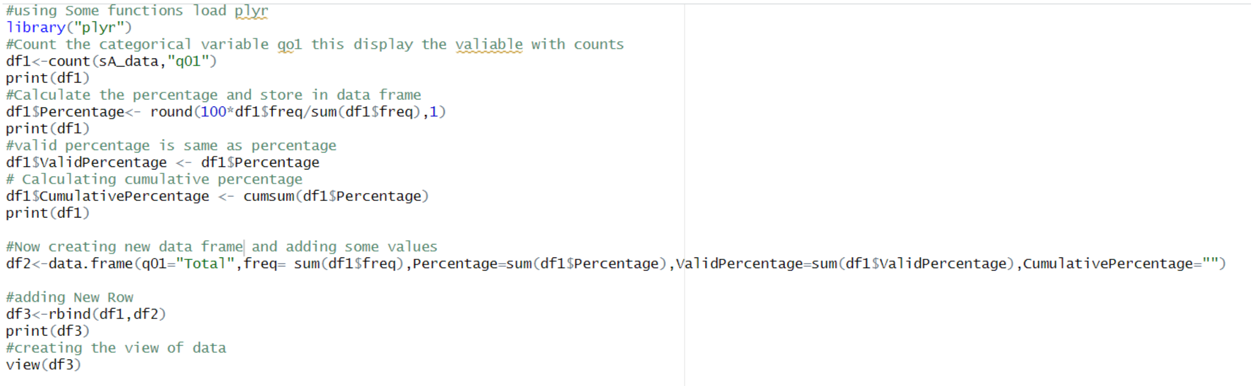
Work 3: See slide 31 of session 2 slide deck and provide answers here. Data is attached.
We know that the attached file is the .sav file. Inorder to read the .sav file (SPSS file) we need the library “foreign” so install the package foreign with **install.packages("foreign")** after that we need to load the package by **library(“foreign”)**

Once the necessary library is load we have to use read.spss() to read the .sav file



```
########################################################
library(foreign)
sA_data<-read.spss('C://Users/ramom/Desktop/MDS/Projects/MDS-503/asignments/Ram Krishna Pudasaini - SAQ8.sav')
sA_data
```

For Statistics makes me cry



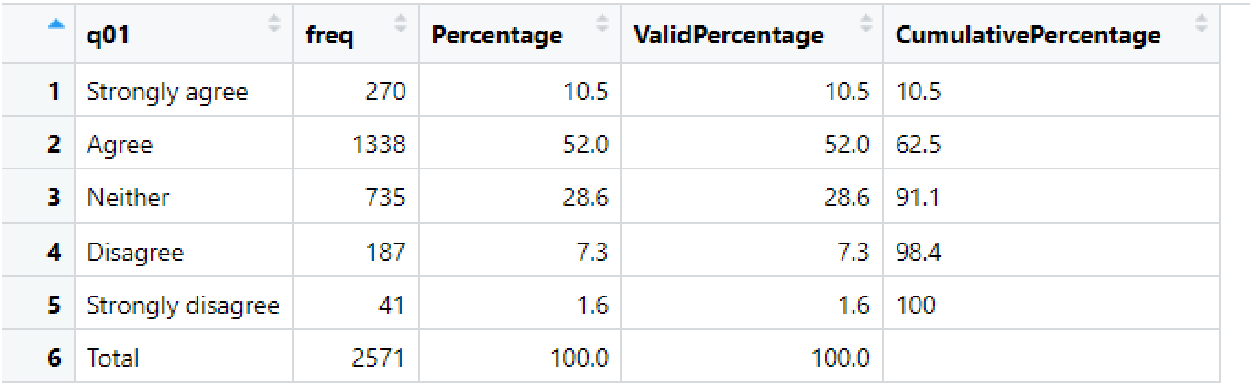
```
#using Some functions load plyr
library("plyr")
#Count the categorical variable qo1 this display the valiable with counts
df1<-count(sA_data,"q01")
print(df1)
#Calculate the percentage and store in data frame
df1$Percentage<- round(100*df1$freq/sum(df1$freq),1)
print(df1)
#valid percentage is same as percentage
df1$ValidPercentage <- df1$Percentage
# Calculating cumulative percentage
df1$CumulativePercentage <- cumsum(df1$Percentage)
print(df1)

#Now creating new data frame and adding some values
df2<-data.frame(q01="Total",freq= sum(df1$freq),Percentage=sum(df1$Percentage),ValidPercentage=sum(df1$ValidPercentage),CumulativePercentage="")

#adding New Row
df3<-rbind(df1,df2)
print(df3)
#creating the view of data
view(df3)
```

Code:

```
#For importing file
library(foreign)
sA_data<-read.spss('C://Users/ramom/Desktop/MDS/Projects/MDS-503/asignments/Ram Krishna Pudasaini - SAQ8.sav',to.data.frame = TRUE)
#View the data
view(sA_data)
#using Some functions load plyr
library("plyr")
#Count the categorical variable qo1 this display the variable with counts
df1<-count(sA_data,"q01")
print(df1)
#Calculate the percentage and store in data frame
df1$Percentage<- round(100*df1$freq/sum(df1$freq),1)
print(df1)
#valid percentage is same as percentage
df1$ValidPercentage <- df1$Percentage
# Calculating cumulative percentage
df1$CumulativePercentage <- cumsum(df1$Percentage)
print(df1)

#Now creating new data frame and adding some values
```

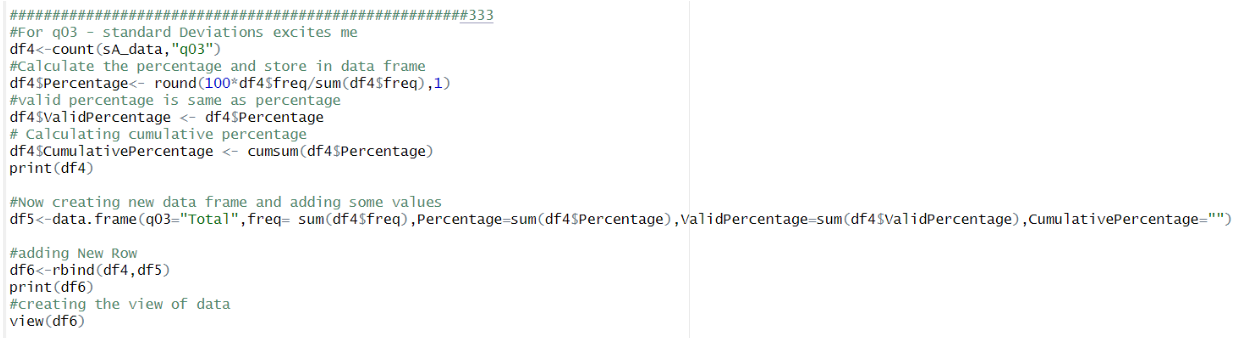
df2<-data.frame(q01="Total",freq= sum(df1$freq),Percentage=sum(df1$Percentage),ValidPercentage=sum(df1$ValidPercentage), CumulativePercentage="")

#adding New Row
df3<-rbind(df1,df2)
print(df3)
#creating the view of data
view(df3)



	q01	freq	Percentage	ValidPercentage	CumulativePercentage
1	Strongly agree	270	10.5	10.5	10.5
2	Agree	1338	52.0	52.0	62.5
3	Neither	735	28.6	28.6	91.1
4	Disagree	187	7.3	7.3	98.4
5	Strongly disagree	41	1.6	1.6	100
6	Total	2571	100.0	100.0	

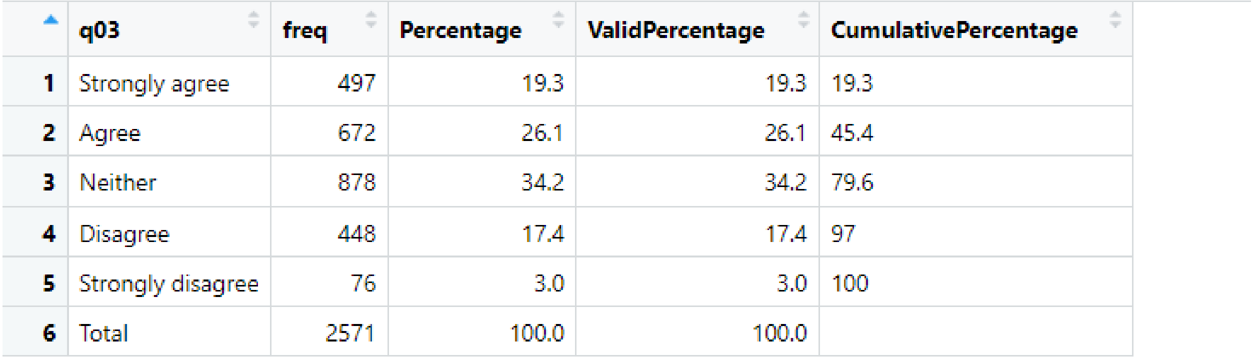
For q03 - standard Deviations excites me



```
#####################################################333
#For q03 - standard Deviations excites me
df4<-count(sA_data,"q03")
#Calculate the percentage and store in data frame
df4$Percentage<- round(100*df4$freq/sum(df4$freq),1)
#valid percentage is same as percentage
df4$ValidPercentage <- df4$Percentage
# Calculating cumulative percentage
df4$CumulativePercentage <- cumsum(df4$Percentage)
print(df4)

#Now creating new data frame and adding some values
df5<-data.frame(q03="Total",freq= sum(df4$freq),Percentage=sum(df4$Percentage),ValidPercentage=sum(df4$ValidPercentage),CumulativePercentage="")

#adding New Row
df6<-rbind(df4,df5)
print(df6)
#creating the view of data
view(df6)
```

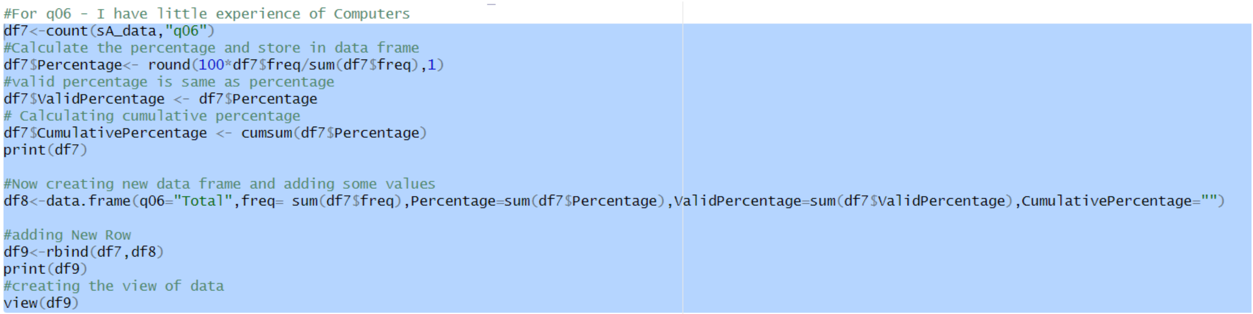
Code:
##
#For q03 - standard Deviations excites me
df4<-count(sA_data,"q03")
#Calculate the percentage and store in data frame
df4$Percentage<- round(100*df4$freq/sum(df4$freq),1)
#valid percentage is same as percentage
df4$ValidPercentage <- df4$Percentage
Calculating cumulative percentage
df4$CumulativePercentage <- cumsum(df4$Percentage)

```
print(df4)

#Now creating new data frame and adding some values
df5<-data.frame(q03="Total",freq=
sum(df4$freq),Percentage=sum(df4$Percentage),ValidPercentage=sum(df4$ValidPercentage),
CumulativePercentage="")

#adding New Row
df6<-rbind(df4,df5)
print(df6)
#creating the view of data
view(df6)
```



	q03	freq	Percentage	ValidPercentage	CumulativePercentage
1	Strongly agree	497	19.3	19.3	19.3
2	Agree	672	26.1	26.1	45.4
3	Neither	878	34.2	34.2	79.6
4	Disagree	448	17.4	17.4	97
5	Strongly disagree	76	3.0	3.0	100
6	Total	2571	100.0	100.0	

qo6 For I have little experience of Computers

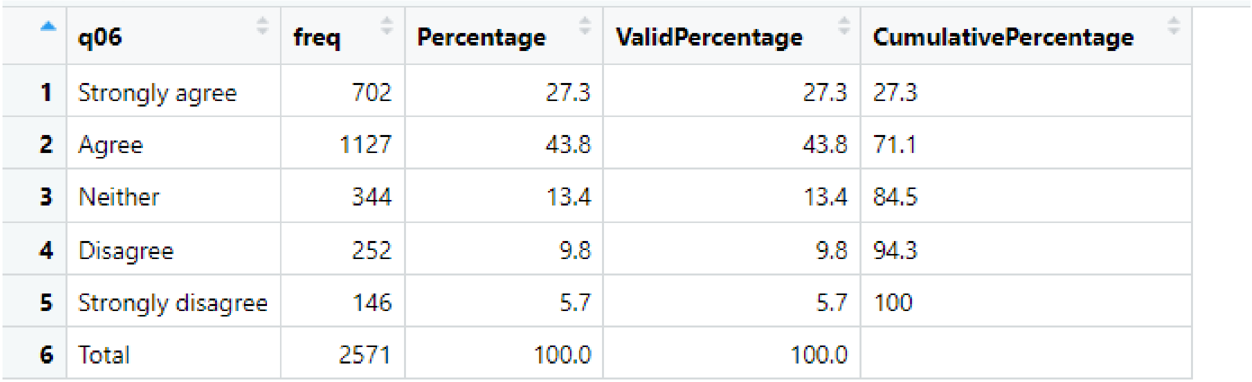


```
#For q06 - I have little experience of Computers
df7<-count(sA_data,"q06")
#Calculate the percentage and store in data frame
df7$Percentage<- round(100*df7$freq/sum(df7$freq),1)
#valid percentage is same as percentage
df7$ValidPercentage <- df7$Percentage
# Calculating cumulative percentage
df7$CumulativePercentage <- cumsum(df7$Percentage)
print(df7)

#Now creating new data frame and adding some values
df8<-data.frame(q06="Total",freq= sum(df7$freq),Percentage=sum(df7$Percentage),ValidPercentage=sum(df7$ValidPercentage),CumulativePercentage="")

#adding New Row
df9<-rbind(df7,df8)
print(df9)
#creating the view of data
view(df9)
```

Code:

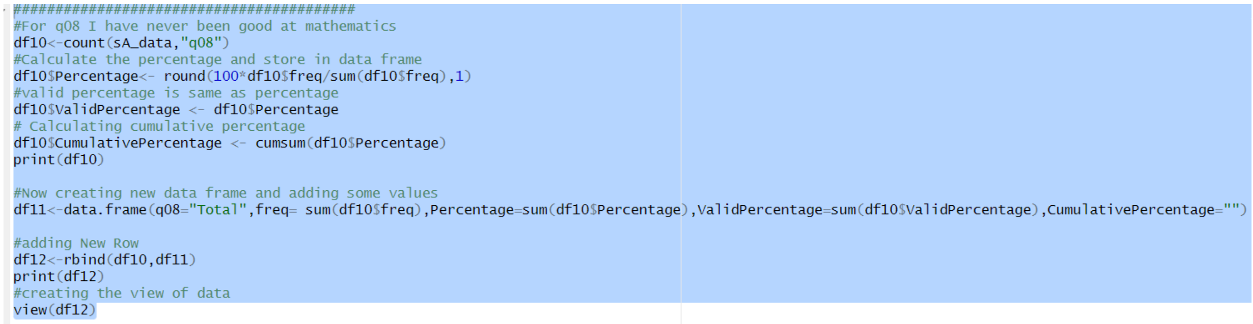
```
#For q06 - I have little experience of Computers
df7<-count(sA_data,"q06")
#Calculate the percentage and store in data frame
df7$Percentage<- round(100*df7$freq/sum(df7$freq),1)
#valid percentage is same as percentage
df7$ValidPercentage <- df7$Percentage
# Calculating cumulative percentage
df7$CumulativePercentage <- cumsum(df7$Percentage)
print(df7)
```

```
#Now creating new data frame and adding some values
df8<-data.frame(q06="Total",freq=
sum(df7$freq),Percentage=sum(df7$Percentage),ValidPercentage=sum(df7$ValidPercentage),
CumulativePercentage="")

#adding New Row
df9<-rbind(df7,df8)
print(df9)
#creating the view of data
view(df9)
```



	q06	freq	Percentage	ValidPercentage	CumulativePercentage
1	Strongly agree	702	27.3	27.3	27.3
2	Agree	1127	43.8	43.8	71.1
3	Neither	344	13.4	13.4	84.5
4	Disagree	252	9.8	9.8	94.3
5	Strongly disagree	146	5.7	5.7	100
6	Total	2571	100.0	100.0	

For q08 I have never been good at mathematics

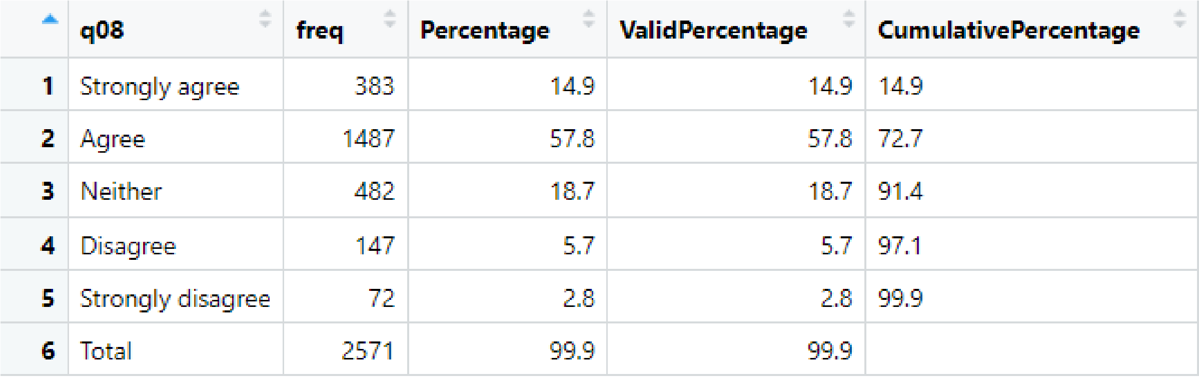


```
##########################################
#For q08 I have never been good at mathematics
df10<-count(sA_data,"q08")
#Calculate the percentage and store in data frame
df10$Percentage<- round(100*df10$freq/sum(df10$freq),1)
#valid percentage is same as percentage
df10$ValidPercentage <- df10$Percentage
# Calculating cumulative percentage
df10$CumulativePercentage <- cumsum(df10$Percentage)
print(df10)

#Now creating new data frame and adding some values
df11<-data.frame(q08="Total",freq= sum(df10$freq),Percentage=sum(df10$Percentage),ValidPercentage=sum(df10$ValidPercentage),CumulativePercentage="")

#adding New Row
df12<-rbind(df10,df11)
print(df12)
#creating the view of data
view(df12)
```

Code:

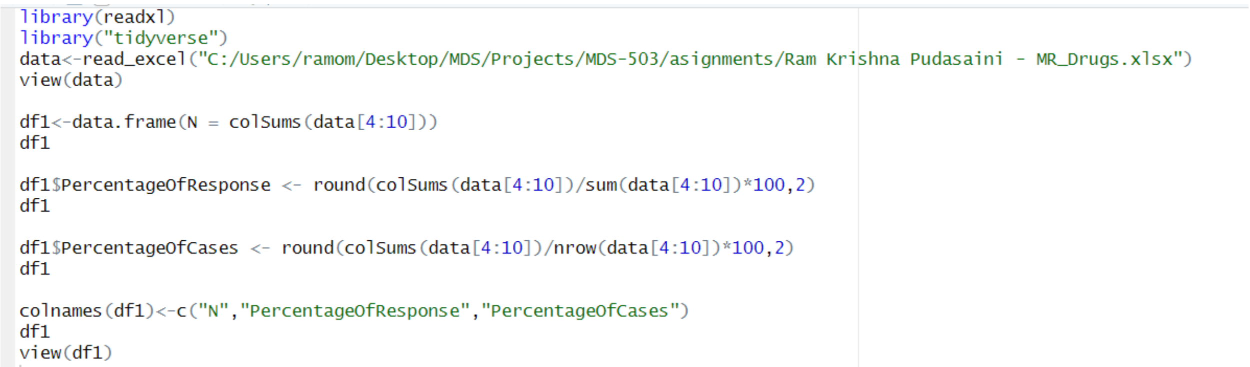
```
#For q08 I have never been good at mathematics
df10<-count(sA_data,"q08")
#Calculate the percentage and store in data frame
df10$Percentage<- round(100*df10$freq/sum(df10$freq),1)
#valid percentage is same as percentage
df10$ValidPercentage <- df10$Percentage
# Calculating cumulative percentage
df10$CumulativePercentage <- cumsum(df10$Percentage)
print(df10)

#Now creating new data frame and adding some values
```

```
df11<-data.frame(q08="Total",freq=
sum(df10$freq),Percentage=sum(df10$Percentage),ValidPercentage=sum(df10$ValidPercentage),CumulativePercentage="")

#adding New Row
df12<-rbind(df10,df11)
print(df12)
#creating the view of data
view(df12)
```

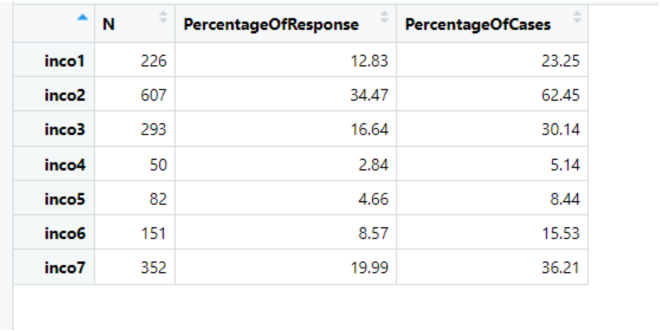


	q08	freq	Percentage	ValidPercentage	CumulativePercentage
1	Strongly agree	383	14.9	14.9	14.9
2	Agree	1487	57.8	57.8	72.7
3	Neither	482	18.7	18.7	91.4
4	Disagree	147	5.7	5.7	97.1
5	Strongly disagree	72	2.8	2.8	99.9
6	Total	2571	99.9	99.9	

Work 4: See slide 32 of session 2 slide deck and provide answers here. Data is attached.



```
library(readxl)
library("tidyverse")
data<-read_excel("C:/Users/ramom/Desktop/MDS/Projects/MDS-503/asignments/Ram Krishna Pudasaini - MR_Drugs.xlsx")
view(data)

df1<-data.frame(N = colSums(data[4:10]))
df1

df1$PercentageOfResponse <- round(colSums(data[4:10])/sum(data[4:10])*100,2)
df1

df1$PercentageOfCases <- round(colSums(data[4:10])/nrow(data[4:10])*100,2)
df1

colnames(df1)<-c("N","PercentageOfResponse","PercentageOfCases")
df1
view(df1)
```



	N	PercentageOfResponse	PercentageOfCases
inco1	226	12.83	23.25
inco2	607	34.47	62.45
inco3	293	16.64	30.14
inco4	50	2.84	5.14
inco5	82	4.66	8.44
inco6	151	8.57	15.53
inco7	352	19.99	36.21