Assignment 2

Dipesh Poudel

9/11/2021

Assignment 2

Creating Dataframe and using it to plot the data

1. Create data frame with these two column vectors in R Studio x = 1:30 y = x^3

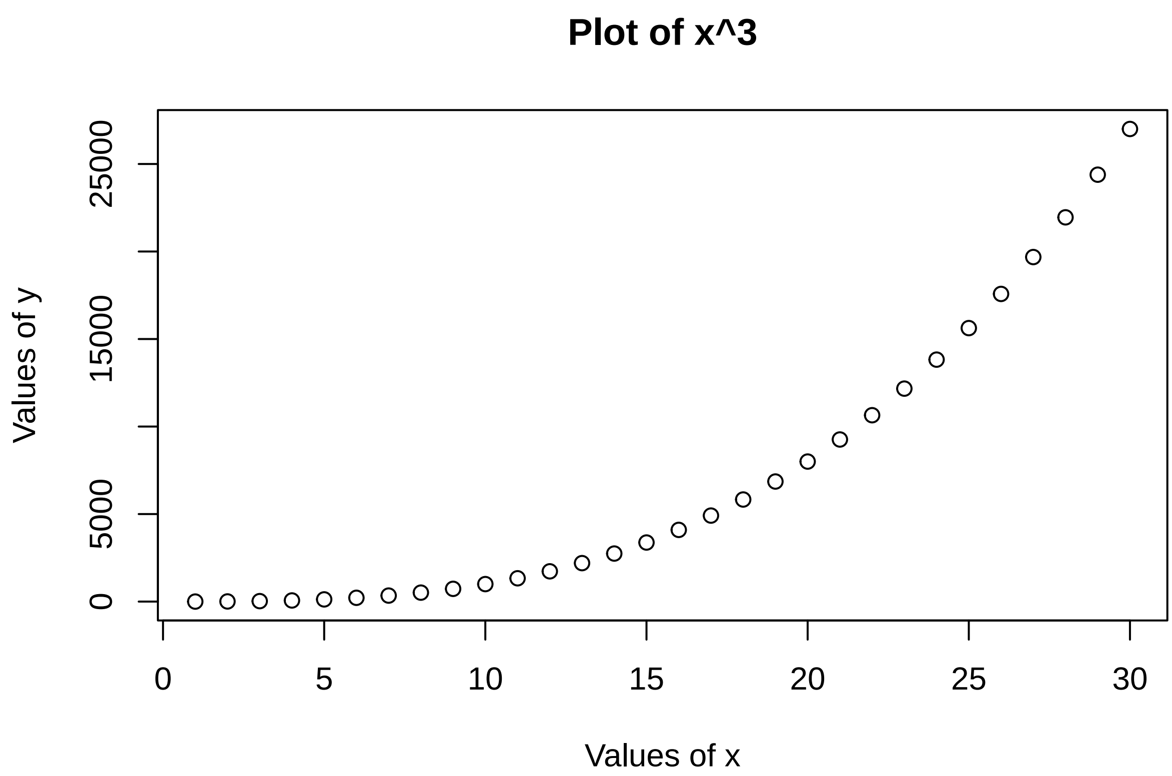
```
# Creating a dataframe
df<-data.frame(x<-c(1:30), y<-x^3)
# Giving names to the columns
colnames(df)<-c('x','y')
print(df)
```

```
##      x     y
## 1    1     1
## 2    2     8
## 3    3    27
## 4    4    64
## 5    5   125
## 6    6   216
## 7    7   343
## 8    8   512
## 9    9   729
## 10  10  1000
## 11  11  1331
## 12  12  1728
## 13  13  2197
## 14  14  2744
## 15  15  3375
## 16  16  4096
## 17  17  4913
## 18  18  5832
## 19  19  6859
## 20  20  8000
## 21  21  9261
## 22  22 10648
## 23  23 12167
## 24  24 13824
## 25  25 15625
## 26  26 17576
## 27  27 19683
```

```
## 28 28 21952
## 29 29 24389
## 30 30 27000
```

2. Create plot of x and y variables in R Studio and interpret it carefully

```
# Plotting the values of x and y
plot(df$x,df$y, main = "Plot of x^3", xlab = 'Values of x', ylab = 'Values of y')
```



In the plot above we can see that the value of y increases exponentially. From the graph, we can see that, as value of x grows a small change in value of x increases the value of y drastically.

3. Get appropriate correlation coefficient of this data in R Studio and interpret it carefully

Since the relationship between the variables is not linear we should not be using pearson's correlation coefficient rather we use spearman's correlation.

```
cor_val<-cor(df$x,df$y,method = "spearman")
print(cor_val)
```

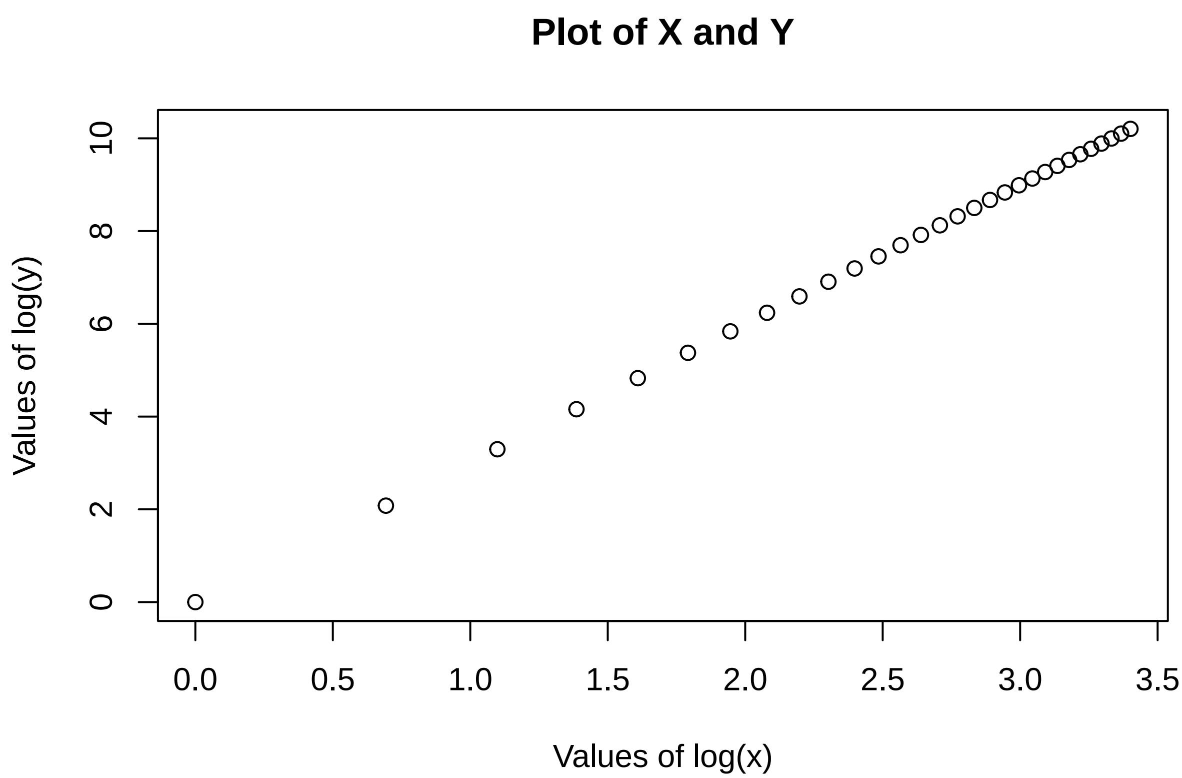
```
## [1] 1
```

This shows that there is perfect correlation between the variables.

Converting Non-Linear to Linear using Log

4. Transform the plot to linear using appropriate mathematical function in R Studio

```
# Using log function to transform the plot to linear
df$a<-log(x)
df$b<-log(y)
plot(df$a,df$b,main = "Plot of X and Y", xlab = 'Values of log(x)', ylab = 'Values of log(y)')
```



5. Get appropriate correlation coefficient now in R Studio and interpret it carefully too

Since we have converted the values into linear we can use the pearson correlation coefficient.

```
cor_val_lin <-cor(df$a,df$b,method = "pearson")
print(cor_val_lin)
```

```
## [1] 1
```

This shows that there is a perfect correlation between the two variables.

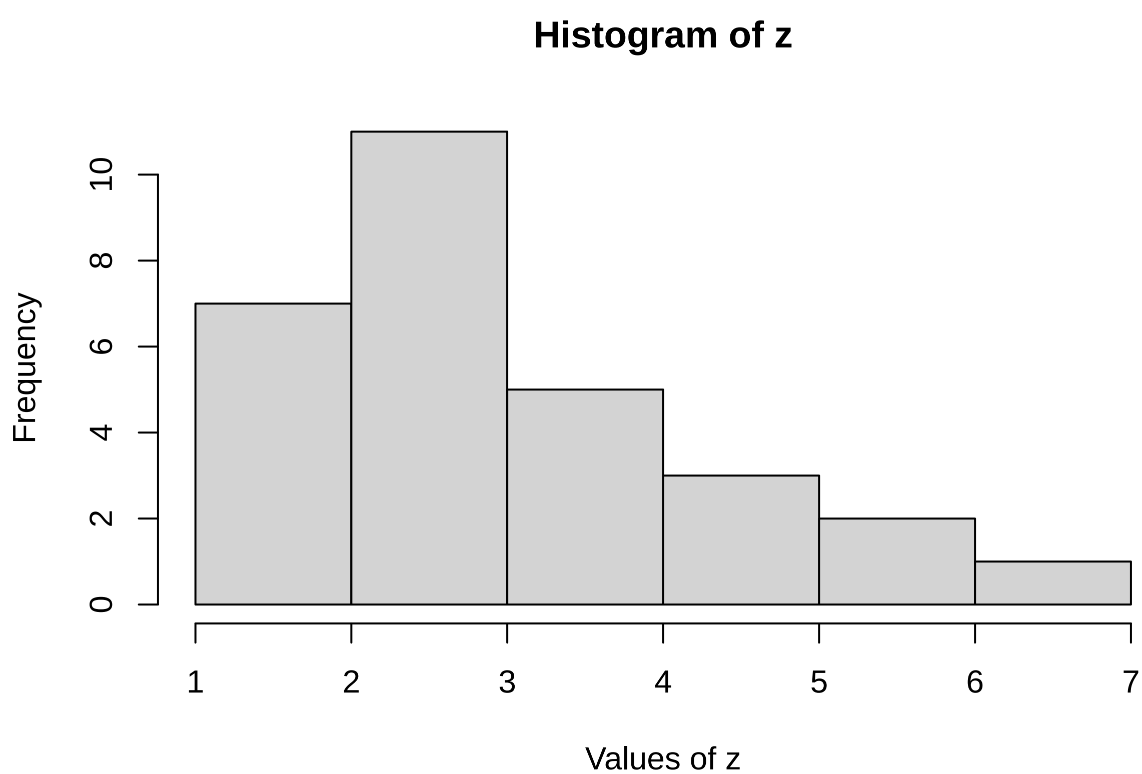
6. Create a new column vector z defined in the slide 18 of session two slide deck in R Studio

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

```
##  [1] 1 1 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 5 5 5 6 6 7
```

7. Create a histogram of z variable in R Studio and interpret it carefully

```r
hist(z,main="Histogram of z",xlab = "Values of z")
```



The histogram shows that the value 3 has highest frequency. It also shows a right skewed distribution. In case of skewed data median is the appropriate measure of central tendency.

8. Get summary statistics of z variable in R Studio and interpret it carefully

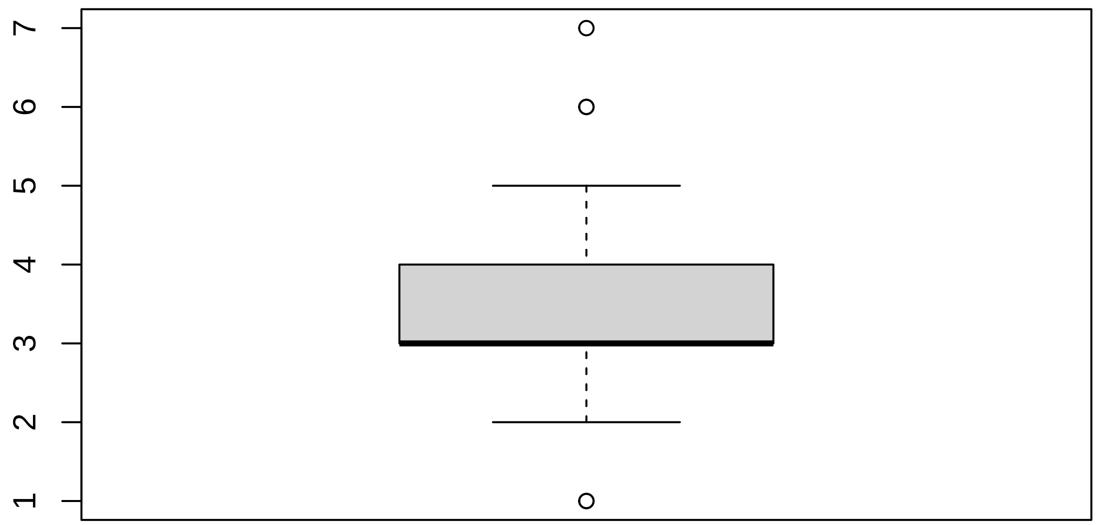
```r
summary(z)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.000   3.000   3.000   3.414   4.000   7.000
```

The summary provides a quick glimpse at the data. It gives us mean, median, minimum and maximum values alongside q1 and q3.

9. Get box-plot of z variable in R Studio and interpret the result carefully.

```r
boxplot(z)
```



The box plot above shows that the median of the data is 3. There are 3 outlier points If a data point is greater than Q3+(1.5 * IQR) or less than Q1-(1.5 * IQR) then they care considered as outlier. In our case, IQR = Q3-Q1 = 4-3=1 So, For data point less than 1.5 and greater than 5.5 are shown as ouliers indicated by 'o' symbol.

10. Import "covnep_252days.csv" data in R Studio and describe the variables in it

```r
file_path = "covnep_252days.csv"
data_csv = read.csv(file = file_path)
print(head(data_csv))
```

```
##        date totalCases newCases totalRecoveries newRecoveries totalDeaths
## 1 1/23/2020          1        1               0             0           0
## 2 1/24/2020          0        0               0             0           0
## 3 1/25/2020          0        0               0             0           0
## 4 1/26/2020          0        0               0             0           0
## 5 1/27/2020          0        0               0             0           0
## 6 1/28/2020          0        0               0             0           0
##   newDeaths
## 1         0
## 2         0
## 3         0
## 4         0
```

```
## 5          0
## 6          0
```

```
names(data_csv)
```

```
## [1] "date"             "totalCases"      "newCases"        "totalRecoveries"
## [5] "newRecoveries"   "totalDeaths"     "newDeaths"
```

```
summary(data_csv)
```

```
##      date             totalCases       newCases      totalRecoveries
##  Length:252         Min.   :    0   Min.   :   0.0   Min.   :    0
##  Class :character   1st Qu.:    2   1st Qu.:   0.0   1st Qu.:    2
##  Mode  :character   Median :  963   Median :  82.5   Median :  182
##                     Mean   :13376   Mean   : 308.8   Mean   : 8380
##                     3rd Qu.:19340   3rd Qu.: 463.2   3rd Qu.:13932
##                     Max.   :77816   Max.   :2020.0   Max.   :56282
##  newRecoveries     totalDeaths       newDeaths
##  Min.   :   0.0   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:   0.0   1st Qu.:  0.00   1st Qu.: 0.000
##  Median :   3.5   Median :  6.00   Median : 0.000
##  Mean   : 223.3   Mean   : 66.67   Mean   : 1.976
##  3rd Qu.: 197.2   3rd Qu.: 53.75   3rd Qu.: 2.000
##  Max.   :2287.0   Max.   :498.00   Max.   :16.000
```

There are seven variables in the csv file and summary of each of them is shown above.

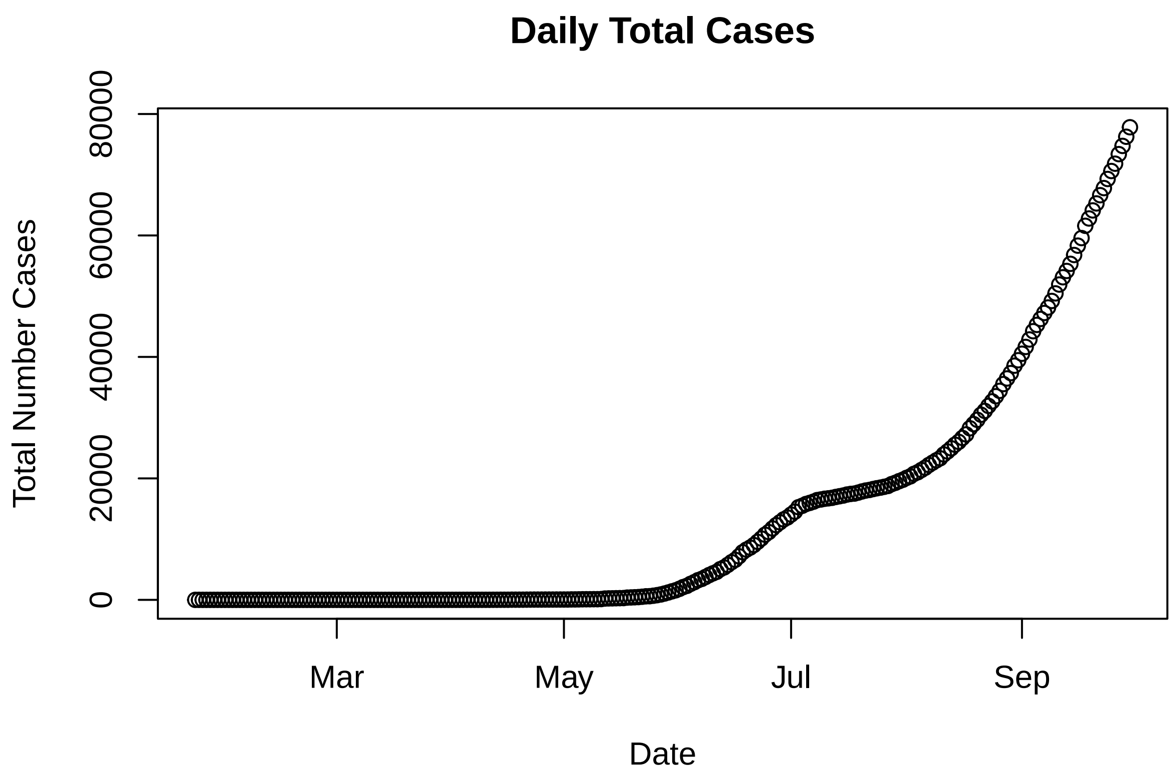
11. Create a chart with "totalCases" variable in y-axis and "date" variable in the x-axis in R Studio, describe the process leading to the creation of this chart

```
# Setting the data type of the date variable as date
data_csv$date<-as.Date(data_csv$date,format="%m/%d/%y")
```

```
head(data_csv)
```

```
##         date totalCases newCases totalRecoveries newRecoveries totalDeaths
## 1 2020-01-23          1        1               0             0           0
## 2 2020-01-24          0        0               0             0           0
## 3 2020-01-25          0        0               0             0           0
## 4 2020-01-26          0        0               0             0           0
## 5 2020-01-27          0        0               0             0           0
## 6 2020-01-28          0        0               0             0           0
##   newDeaths
## 1         0
## 2         0
## 3         0
## 4         0
## 5         0
## 6         0
```

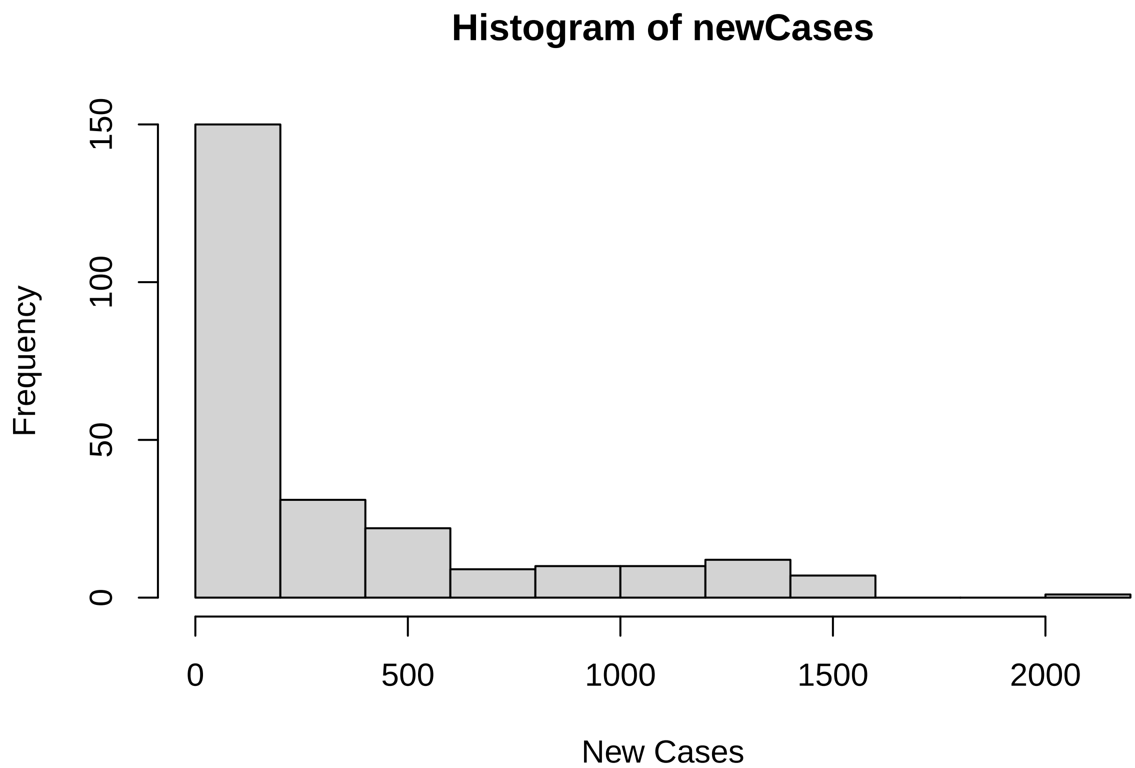
```r
# The totalCases column is a cumulative value column. In the first row, the value is 1 and second row t
data_csv['totalCases'][data_csv['totalCases']==0]<-1
plot(data_csv$date,data_csv$totalCases,main='Daily Total Cases',xlab='Date',ylab ='Total Number Cases')
```



The steps leading upto the plot above is described below. 1. Reading the data into a dataframe using `read_csv` function. 2. Converted the `date` variable into appropriate date data type. 3. Replaced 0 with 1 in `totalCases` column since in the first row, the value is 1 and second row the value is 0 which is not mistake

12. Create histogram of "newCases" variable in R Studio and interpret it carefully

```r
hist(data_csv$newCases, main = 'Histogram of newCases',xlab = 'New Cases')
```



The histogram above shows highly skewed data. The ditribution of data is right skewed.

13. Get summary statistics of "newCases" variable in R Studio and interpret it carefully

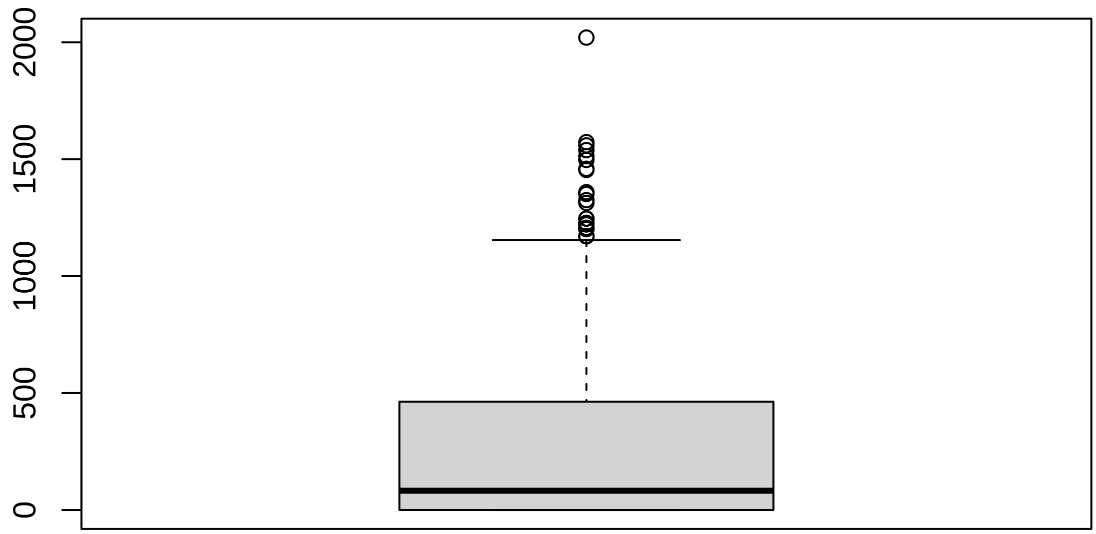
```
summary(data_csv$newCases)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0    82.5   308.8   463.2  2020.0
```

The summary shows that the median is 82.5 which means that for half of the dates we took the new cases per day was less than 82.5 and the new cases per day picked at 2020.

14. Get "box and whisker" plot of "newCases" variable in R Studio and interpret it carefully

```r
boxplot(data_csv$newCases)
```



The plot shows that the very high new cases per day was exception and not the norm. The median of the data is very low compared to the max value which means that the for up-to mid point the number of cases till the mid point of the data was less and it later on increased exponentially.

15. Import "SAQ8.sav" data in R Studio and get frequency distribution (number and percentage of the attributes) of q01, q03, q06 and q08 variables on R Studio and interpret them carefully

```r
# Reading the SPSS file
library(haven)
file_path1="SAQ8.sav"
savdf<-read_sav(file = file_path1)
head(savdf)
```

```
## # A tibble: 6 x 8
##                   q01      q02      q03      q04      q05     q06     q07     q08
##           <dbl+lbl> <dbl+lb> <dbl+lb> <dbl+lb> <dbl+lb> <dbl+l> <dbl+l> <dbl+l>
## 1 2 [Agree]          1 [Stro~ 4 [Disa~ 2 [Agre~ 2 [Agre~ 2 [Agr~ 3 [Nei~ 1 [Str~
## 2 1 [Strongly agree] 1 [Stro~ 4 [Disa~ 3 [Neit~ 2 [Agre~ 2 [Agr~ 2 [Agr~ 2 [Agr~
## 3 2 [Agree]          3 [Neit~ 2 [Agre~ 2 [Agre~ 4 [Disa~ 1 [Str~ 2 [Agr~ 2 [Agr~
## 4 3 [Neither]        1 [Stro~ 1 [Stro~ 4 [Disa~ 3 [Neit~ 3 [Nei~ 4 [Dis~ 2 [Agr~
## 5 2 [Agree]          1 [Stro~ 3 [Neit~ 2 [Agre~ 2 [Agre~ 3 [Nei~ 3 [Nei~ 2 [Agr~
## 6 2 [Agree]          1 [Stro~ 3 [Neit~ 2 [Agre~ 4 [Disa~ 4 [Dis~ 4 [Dis~ 2 [Agr~
```

```r
summary(savdf)
```

```
##       q01             q02             q03             q04
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :2.000   Median :1.000   Median :3.000   Median :3.000
##  Mean   :2.374   Mean   :1.623   Mean   :2.585   Mean   :2.786
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##       q05             q06             q07             q08
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :3.000   Median :2.000   Median :3.000   Median :2.000
##  Mean   :2.722   Mean   :2.227   Mean   :2.924   Mean   :2.237
##  3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
```

```r
# install.packages('plyr')
library(plyr)
col_list<-c('q01','q03','q06','q08')
for (i in 1:length(col_list)){
  cat("Frequency and Pecentage For",col_list[i],"\n")
  df_count<-count(savdf[col_list[i]])
  df_count$Percentage <- round(100*df_count$freq/sum(df_count$freq),3)
  print(df_count)
  }
```

```
## Frequency and Pecentage For q01
##   q01 freq Percentage
## 1   1  270     10.502
## 2   2 1338     52.042
## 3   3  735     28.588
## 4   4  187      7.273
## 5   5   41      1.595
## Frequency and Pecentage For q03
##   q03 freq Percentage
## 1   1  497     19.331
## 2   2  672     26.138
## 3   3  878     34.150
## 4   4  448     17.425
## 5   5   76      2.956
## Frequency and Pecentage For q06
##   q06 freq Percentage
## 1   1  702     27.305
## 2   2 1127     43.835
## 3   3  344     13.380
## 4   4  252      9.802
## 5   5  146      5.679
## Frequency and Pecentage For q08
##   q08 freq Percentage
## 1   1  383     14.897
## 2   2 1487     57.837
## 3   3  482     18.748
```

```
## 4   4  147      5.718
## 5   5   72      2.800
```

For the given columns we calculated the Frequency and Percentage of each factor

16. Import "MR_drugs.xls" data in R Studio and replicate multiple response frequency distribution as shown in the slide 35 of the session 2 slide deck

```r
library(readxl)
file_path_xl = "MR_Drugs.xls"
drug_df<-readxl::read_xls(file_path_xl)
head(drug_df)
```

```
## # A tibble: 6 x 27
##      id   sex  city inco1 inco2 inco3 inco4 inco5 inco6 inco7 pinco1 pinco2
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
## 1  1001     2     1     0     0     0     0     0     1     0      6     -1
## 2  1002     2     1     0     1     0     0     0     0     0      2     -1
## 3  1003     2     1     0     0     0     0     0     1     0      6     -1
## 4  1004     2     1     0     1     0     0     0     0     0      2     -1
## 5  1005     2     1     0     0     0     0     0     0     1      7     -1
## 6  1006     2     1     1     1     0     0     0     0     0      2      1
## # ... with 15 more variables: pinco3 <dbl>, pinco4 <dbl>, pinco5 <dbl>,
## #   pinco6 <dbl>, sinco1 <chr>, sinco2 <chr>, sinco3 <chr>, sinco4 <chr>,
## #   sinco5 <chr>, sinco6 <chr>, crime1 <dbl>, crime2 <dbl>, crime3 <dbl>,
## #   crime4 <dbl>, crime5 <dbl>
```

```r
summary(drug_df)
```

```
##        id            sex             city           inco1
##  Min.   :1001   Min.   :1.000   Min.   :1.000   Min.   :0.0000
##  1st Qu.:1254   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000
##  Median :3148   Median :2.000   Median :2.000   Median :0.0000
##  Mean   :2803   Mean   :1.736   Mean   :1.988   Mean   :0.2325
##  3rd Qu.:4098   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:0.0000
##  Max.   :4365   Max.   :2.000   Max.   :3.000   Max.   :1.0000
##                 NA's   :1
##      inco2            inco3            inco4             inco5
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :1.0000   Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   :0.6245   Mean   :0.3014   Mean   :0.05144   Mean   :0.08436
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##
##      inco6            inco7            pinco1           pinco2
##  Min.   :0.0000   Min.   :0.0000   Min.   :-1.000   Min.   :-1.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 2.000   1st Qu.:-1.000
##  Median :0.0000   Median :0.0000   Median : 3.000   Median : 1.000
##  Mean   :0.1553   Mean   :0.3621   Mean   : 3.628   Mean   : 1.297
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.: 6.000   3rd Qu.: 3.000
##  Max.   :1.0000   Max.   :1.0000   Max.   : 7.000   Max.   : 7.000
```

```
##
##      pinco3            pinco4           pinco5           pinco6
##  Min.   :-1.00000   Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.0000
##  1st Qu.:-1.00000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000
##  Median :-1.00000   Median :-1.0000   Median :-1.0000   Median :-1.0000
##  Mean   :-0.01646   Mean   :-0.7274   Mean   :-0.9095   Mean   :-0.9794
##  3rd Qu.:-1.00000   3rd Qu.:-1.0000   3rd Qu.:-1.0000   3rd Qu.:-1.0000
##  Max.   : 7.00000   Max.   : 7.0000   Max.   : 7.0000   Max.   : 6.0000
##
##     sinco1             sinco2             sinco3             sinco4
##  Length:972         Length:972         Length:972         Length:972
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     sinco5             sinco6              crime1           crime2
##  Length:972         Length:972         Min.   :0.0000   Min.   :0.00000
##  Class :character   Class :character   1st Qu.:0.0000   1st Qu.:0.00000
##  Mode  :character   Mode  :character   Median :0.0000   Median :0.00000
##                                        Mean   :0.3881   Mean   :0.08159
##                                        3rd Qu.:0.0000   3rd Qu.:0.00000
##                                        Max.   :3.0000   Max.   :3.00000
##                                        NA's   :65       NA's   :65
##      crime3            crime4           crime5
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.00000   Median :0.0000   Median :0.00000
##  Mean   :0.06946   Mean   :0.2745   Mean   :0.07056
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :2.00000   Max.   :3.0000   Max.   :3.00000
##  NA's   :65        NA's   :65       NA's   :65
```

```r
drug_data_inc<-data.frame(N=colSums(drug_df[4:10]),
                          Percent=round((colSums(drug_df[4:10])/sum(drug_df[4:10]))*100,3),
                          PercentOfCases=round((colSums(drug_df[4:10])/nrow(drug_df[4:10]))*100,3)
                          )
drug_data_inc
```

```
##         N Percent PercentOfCases
## inco1 226  12.834         23.251
## inco2 607  34.469         62.449
## inco3 293  16.638         30.144
## inco4  50   2.839          5.144
## inco5  82   4.656          8.436
## inco6 151   8.575         15.535
## inco7 352  19.989         36.214
```