Project Report

COVID-19 IMPACT ON STATES IN USA

Submitted By:

Sherin Joseph
21 / May / 2021

INTRODUCTION

This project gives us an insight about the impact of COVID-19 on 6 different states in USA for a period of 3 months. The data collected is analyzed and interpretations are made based on the following:

- Understanding the business requirements.
- Analysing the data obtained in raw format.
- Cleaning the data with logical values.
- Summarizations of the columns in dataset.
- Visualization of the variables.
- Univariate analysis.
- Bivariate Analysis for checking relationships.

DATA SET

This following are the details of the data set:

- Number of Columns: 6
- Column Names: Date, State, Cases, Deaths, Recovered and Vulnerability
- Number of Rows: 581
- Name of states: Alabama, Arizona, California, Florida, Georgia, and Texas
- Time period: 3 months (October, November, and December 2020)
- Data Type: 1 Date, 2 Categorical and 3 numerical columns
- Target Column: Vulnerability column

The vulnerability column has 3 categories “Critical”, “Low” and “Moderate” and our project focus on finding the relationships between target column and other predictor columns.

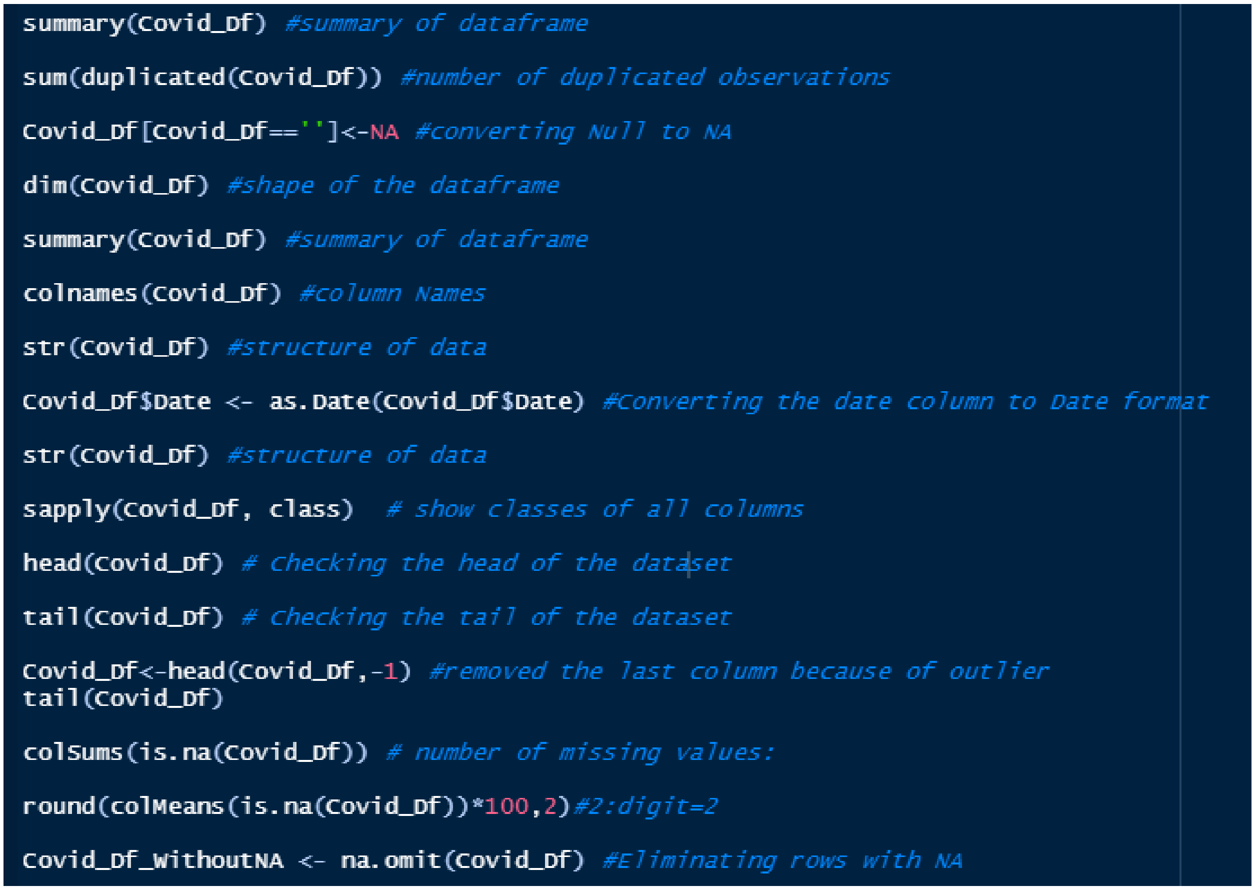
CASE STUDY

The following are the case studies:

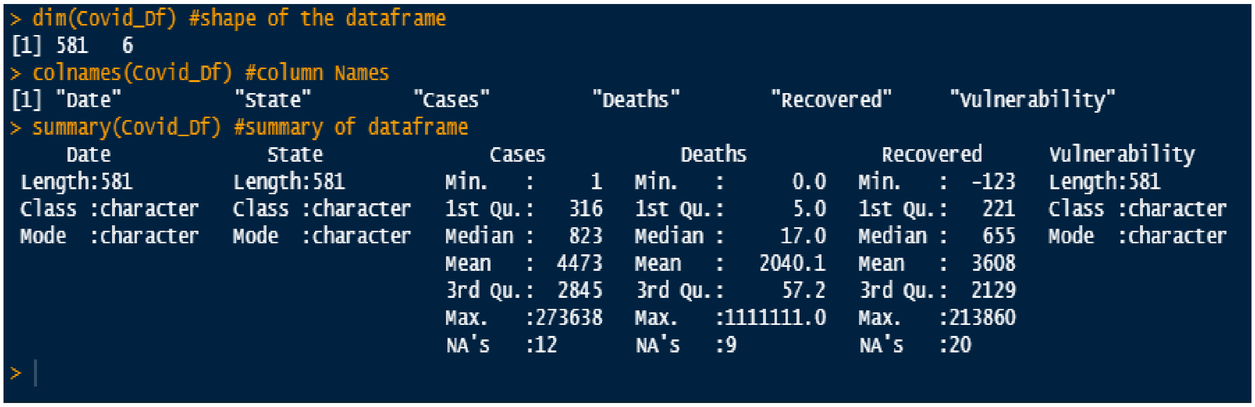
1. what is the distribution of target column (Vulnerability)?
2. what is the distribution of States?
3. what is the distribution of cases reported in all states?
4. what is the pattern of recovery reported in all states?
5. what is the total death in all states?
6. what is the relation between states and Vulnerability Level?
7. Is there relation between the month and Vulnerability Level?
8. Is there relation between the Cases and Recovery?
9. Is there relation between the Deaths and Recovery?
10. Is there relation between the Cases and Vulnerability?

FAMILIARIZING DATA SET

There are several functions in R which are used to familiarize the data such as summary, dim, colnames, str, head, tail, colSums and so on. A code of these functions is displayed below:



```
summary(Covid_Df) #summary of dataframe

sum(duplicated(Covid_Df)) #number of duplicated observations

Covid_Df[Covid_Df=='']<-NA #converting Null to NA

dim(Covid_Df) #shape of the dataframe

summary(Covid_Df) #summary of dataframe

colnames(Covid_Df) #column Names

str(Covid_Df) #structure of data

Covid_Df$Date <- as.Date(Covid_Df$Date) #Converting the date column to Date format

str(Covid_Df) #structure of data

sapply(Covid_Df, class)  # show classes of all columns

head(Covid_Df) # Checking the head of the dataset

tail(Covid_Df) # Checking the tail of the dataset

Covid_Df<-head(Covid_Df,-1) #removed the last column because of outlier
tail(Covid_Df)

colSums(is.na(Covid_Df)) # number of missing values:

round(colMeans(is.na(Covid_Df))*100,2)#2:digit=2

Covid_Df_WithoutNA <- na.omit(Covid_Df) #Eliminating rows with NA
```

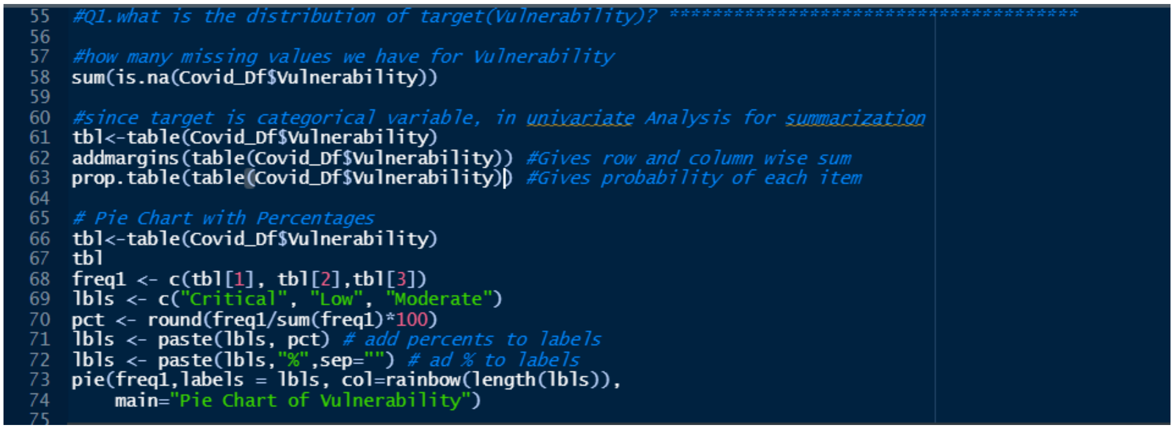


```
> dim(Covid_Df) #shape of the dataframe
[1] 581   6
> colnames(Covid_Df) #column Names
[1] "Date"          "State"         "Cases"         "Deaths"        "Recovered"     "Vulnerability"
> summary(Covid_Df) #summary of dataframe
     Date              State               Cases              Deaths            Recovered       Vulnerability     
 Length:581         Length:581         Min.   :     1   Min.   :      0.0   Min.   :  -123   Length:581        
 Class :character   Class :character   1st Qu.:   316   1st Qu.:      5.0   1st Qu.:   221   Class :character  
 Mode  :character   Mode  :character   Median :   823   Median :     17.0   Median :   655   Mode  :character  
                                       Mean   :  4473   Mean   :   2040.1   Mean   :  3608                     
                                       3rd Qu.:  2845   3rd Qu.:     57.2   3rd Qu.:  2129                     
                                       Max.   :273638   Max.   :1111111.0   Max.   :213860                     
                                       NA's   :12       NA's   :9           NA's   :20                         
> 
```

UNIVARIATE ANALYSIS

1. what is the distribution of target column (Vulnerability)?

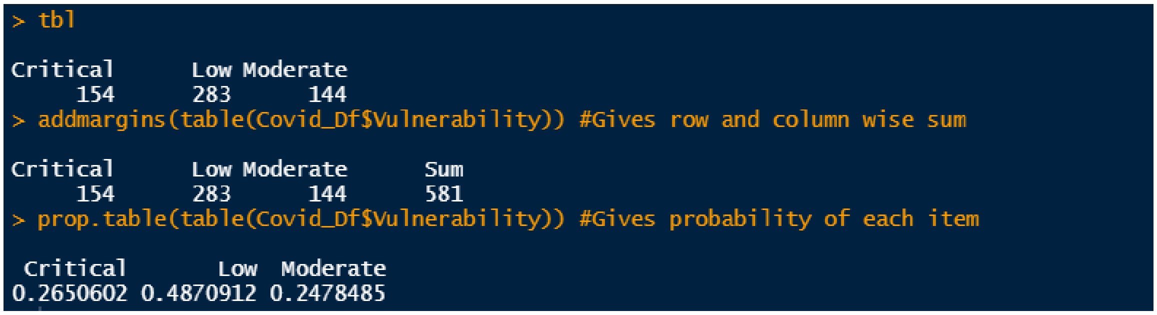
The target column Vulnerability do have any NA values and we need to find the distribution of the variable over the period in all the states.



```
#Q1.what is the distribution of target(Vulnerability)? *****************************************

#how many missing values we have for Vulnerability
sum(is.na(Covid_Df$Vulnerability))

#since target is categorical variable, in univariate Analysis for summarization
tbl<-table(Covid_Df$Vulnerability)
addmargins(table(Covid_Df$Vulnerability)) #Gives row and column wise sum
prop.table(table(Covid_Df$Vulnerability)) #Gives probability of each item

# Pie Chart with Percentages
tbl<-table(Covid_Df$Vulnerability)
tbl
freq1 <- c(tbl[1], tbl[2],tbl[3])
lbls <- c("Critical", "Low", "Moderate")
pct <- round(freq1/sum(freq1)*100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(freq1,labels = lbls, col=rainbow(length(lbls)),
    main="Pie Chart of Vulnerability")
```

#Summarization

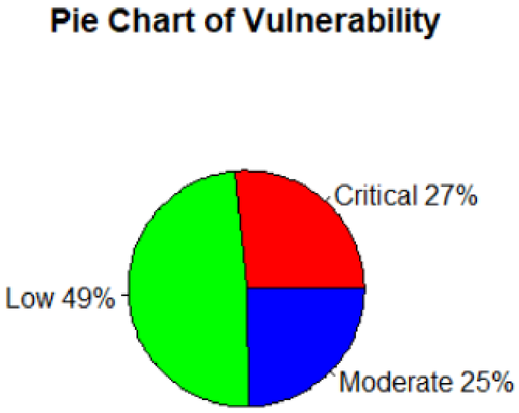
For summarization we use the table function, and the distribution of the categorical variable is as shown below.



```
> tbl

Critical      Low Moderate 
     154      283      144 
> addmargins(table(Covid_Df$Vulnerability)) #Gives row and column wise sum

Critical      Low Moderate      Sum 
     154      283      144      581 
> prop.table(table(Covid_Df$Vulnerability)) #Gives probability of each item

 Critical       Low  Moderate 
0.2650602 0.4870912 0.2478485 
```

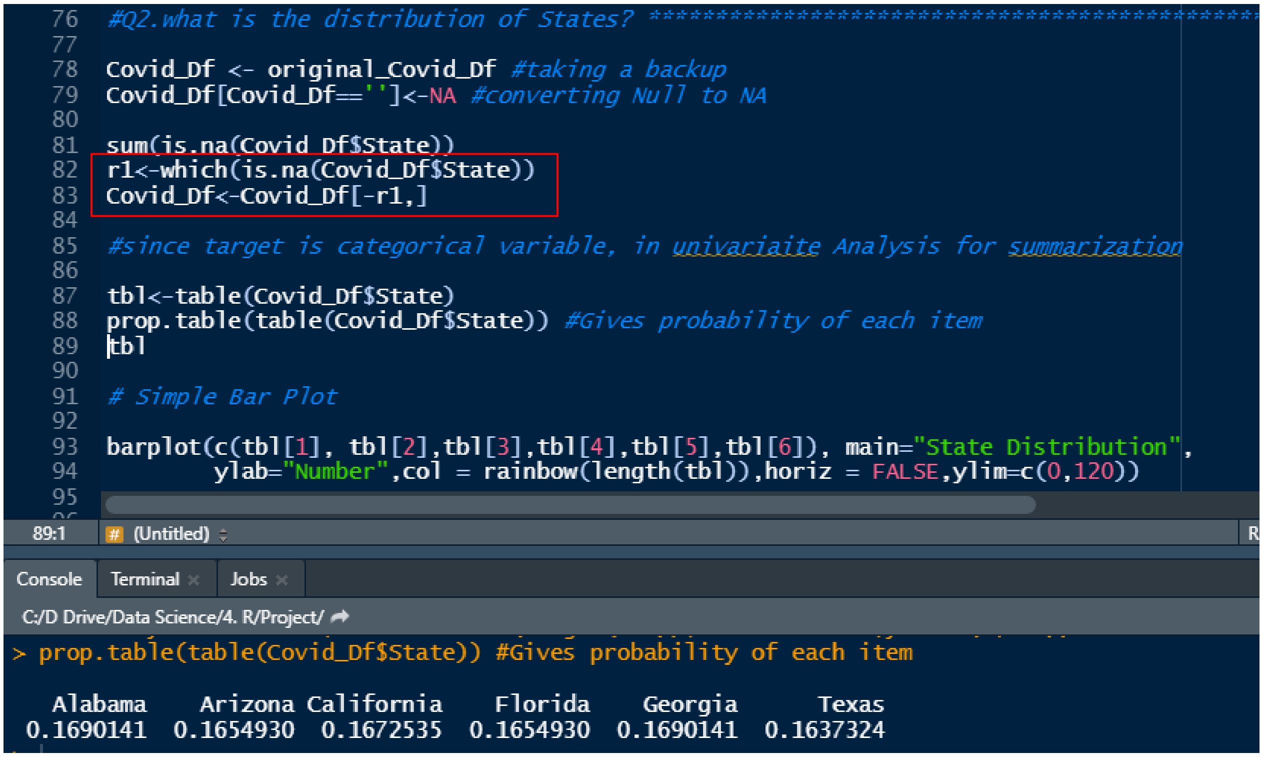
#Visualization

We use a pie chart for viewing the distribution of each level of the categorical column and the result obtained from our data set is as shown below.



2. what is the distribution of States?

We need to find the distribution of 6 different states in the data set.



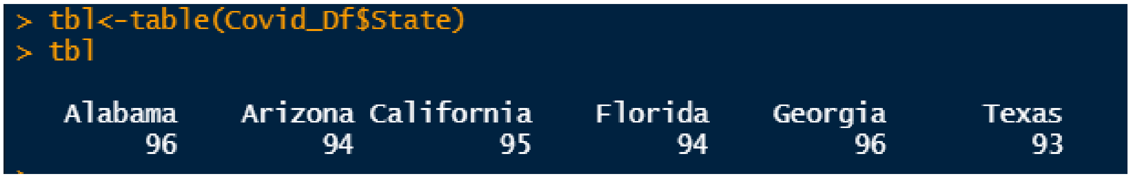
```
#Q2.what is the distribution of States? ******************************************

Covid_Df <- original_Covid_Df #taking a backup
Covid_Df[Covid_Df=='']<-NA #converting Null to NA

sum(is.na(Covid_Df$State))
r1<-which(is.na(Covid_Df$State))
Covid_Df<-Covid_Df[-r1,]

#since target is categorical variable, in univariaite Analysis for summarization

tbl<-table(Covid_Df$State)
prop.table(table(Covid_Df$State)) #Gives probability of each item
tbl

# Simple Bar Plot

barplot(c(tbl[1], tbl[2],tbl[3],tbl[4],tbl[5],tbl[6]), main="State Distribution",
        ylab="Number",col = rainbow(length(tbl)),horiz = FALSE,ylim=c(0,120))
```

```
> prop.table(table(Covid_Df$State)) #Gives probability of each item

  Alabama   Arizona California   Florida   Georgia     Texas 
0.1690141 0.1654930 0.1672535 0.1654930 0.1690141 0.1637324 
```

There are some NA values in the state column, and we have removed those rows from the data frame.

#Summarization

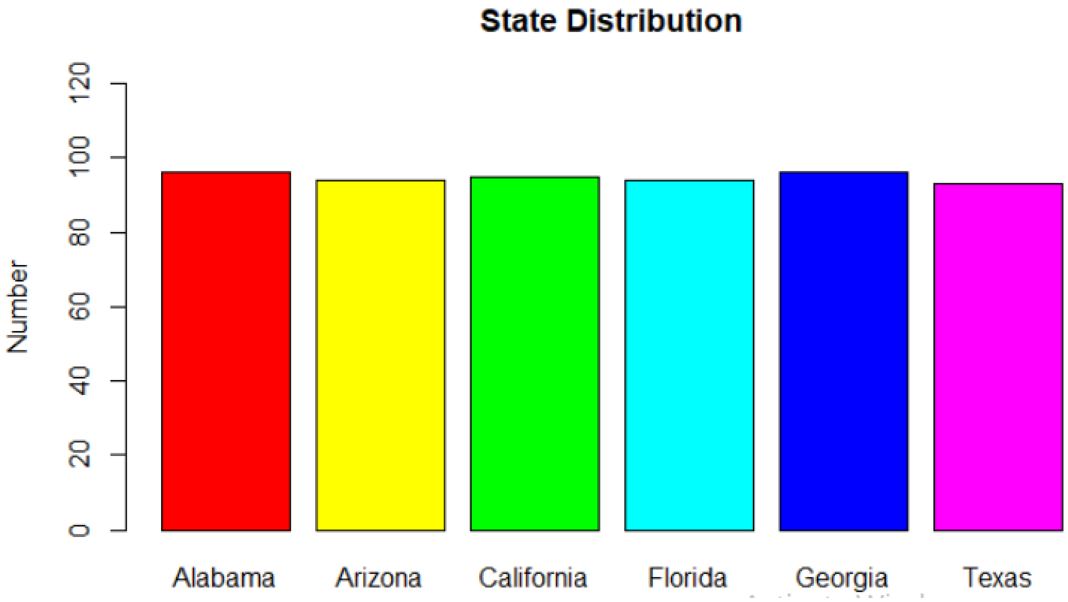
For summarization we use the table function, and the distribution of the categorical variable is as shown below.



```
> tbl<-table(Covid_Df$State)
> tbl

   Alabama    Arizona California    Florida    Georgia      Texas 
        96         94         95         94         96         93 
```

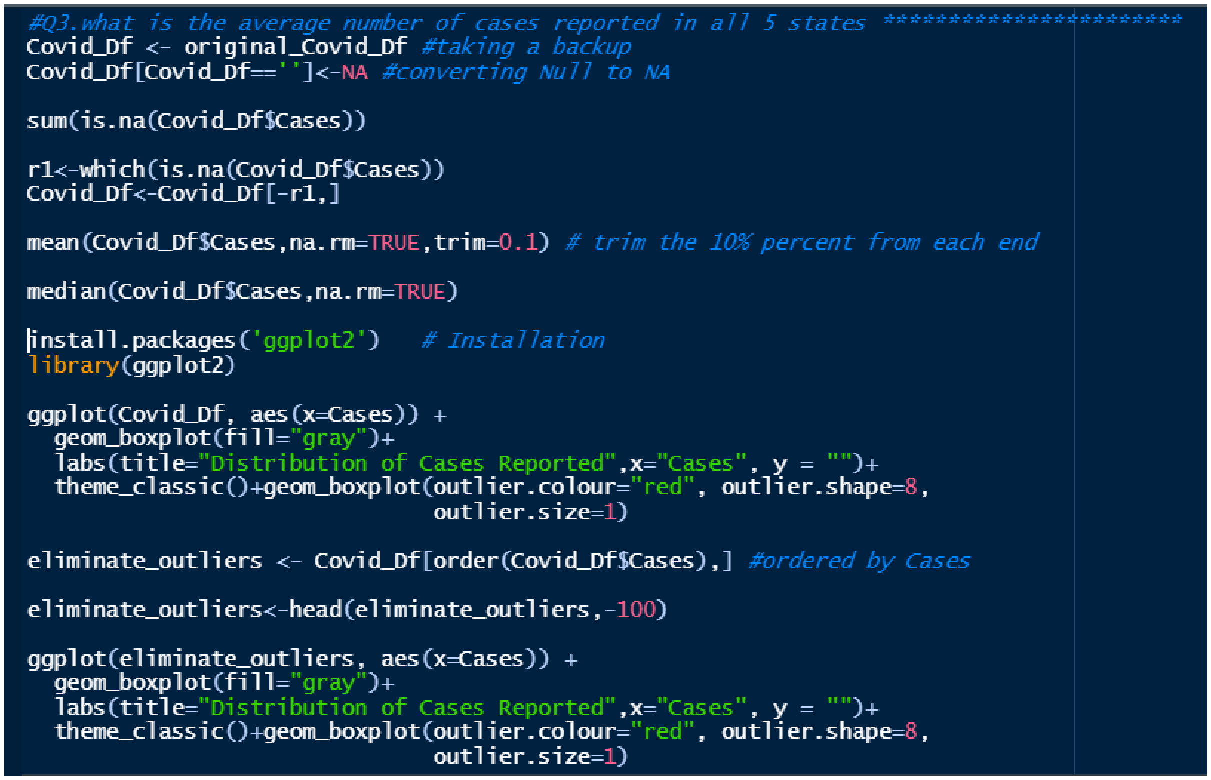
#Visualization

We use a bar chart for viewing the distribution of each level of the categorical column and the result obtained from our data set is as shown below.



3. what is the distribution of cases reported in all states?

We need to find the distribution of total COVID-19 cases reported over the period from October 2020 to December 2020 in the data set.

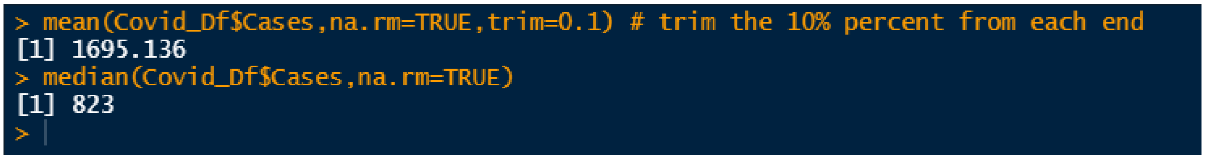


```
#Q3.what is the average number of cases reported in all 5 states *************************
Covid_Df <- original_Covid_Df #taking a backup
Covid_Df[Covid_Df=='']<-NA #converting Null to NA

sum(is.na(Covid_Df$Cases))

r1<-which(is.na(Covid_Df$Cases))
Covid_Df<-Covid_Df[-r1,]

mean(Covid_Df$Cases,na.rm=TRUE,trim=0.1) # trim the 10% percent from each end

median(Covid_Df$Cases,na.rm=TRUE)

install.packages('ggplot2')    # Installation
library(ggplot2)

ggplot(Covid_Df, aes(x=Cases)) +
  geom_boxplot(fill="gray")+
  labs(title="Distribution of Cases Reported",x="Cases", y = "")+
  theme_classic()+geom_boxplot(outlier.colour="red", outlier.shape=8,
                               outlier.size=1)

eliminate_outliers <- Covid_Df[order(Covid_Df$Cases),] #ordered by Cases

eliminate_outliers<-head(eliminate_outliers,-100)

ggplot(eliminate_outliers, aes(x=Cases)) +
  geom_boxplot(fill="gray")+
  labs(title="Distribution of Cases Reported",x="Cases", y = "")+
  theme_classic()+geom_boxplot(outlier.colour="red", outlier.shape=8,
                               outlier.size=1)
```

There are certain rows with the NA values, and they are removed from the data frame.

#Summarization

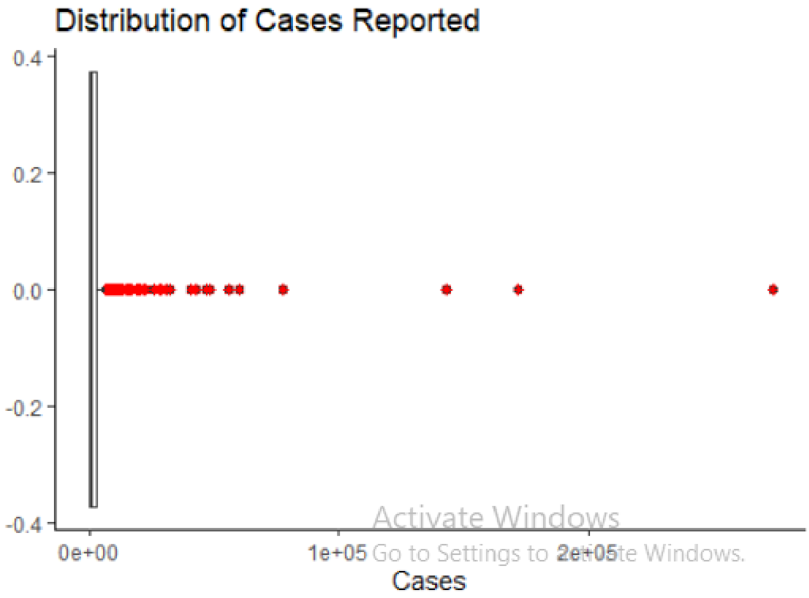
For summarization we use the Central tendency, and the distribution of the continuous variable is as shown below.



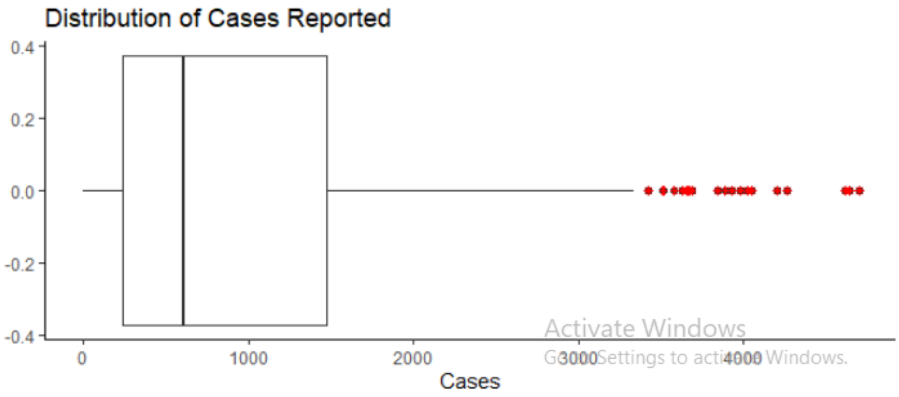
```
> mean(Covid_Df$Cases,na.rm=TRUE,trim=0.1) # trim the 10% percent from each end
[1] 1695.136
> median(Covid_Df$Cases,na.rm=TRUE)
[1] 823
> 
```

#Visualization

We use a box plot for viewing the distribution of each level of the continuous variable and the result obtained from our data set is as shown below.

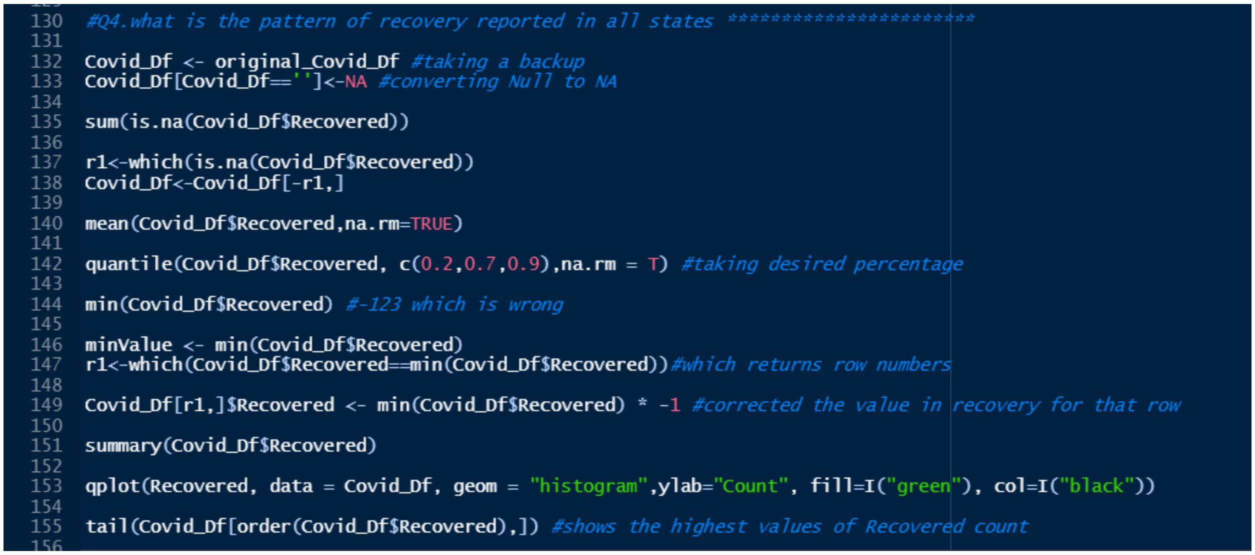


To get a clear sample distribution, certain outliers are removed and the distribution is plotted as shown below.



4. what is the pattern of recovery reported in all states?

We are to find the pattern of the COVID-19 cases recovered over the period from October 2020 to December 2020 in the data set.

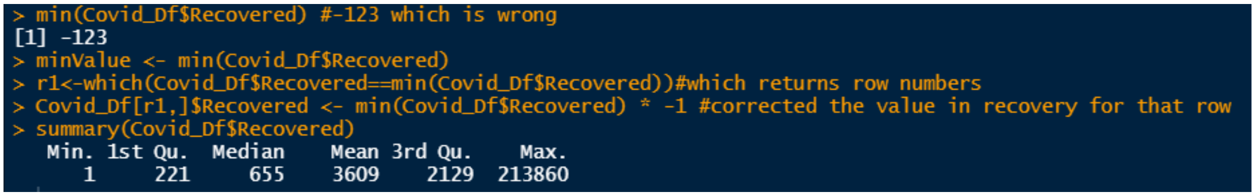


```
#Q4.what is the pattern of recovery reported in all states ************************

Covid_Df <- original_Covid_Df #taking a backup
Covid_Df[Covid_Df=='']<-NA #converting Null to NA

sum(is.na(Covid_Df$Recovered))

r1<-which(is.na(Covid_Df$Recovered))
Covid_Df<-Covid_Df[-r1,]

mean(Covid_Df$Recovered,na.rm=TRUE)

quantile(Covid_Df$Recovered, c(0.2,0.7,0.9),na.rm = T) #taking desired percentage

min(Covid_Df$Recovered) #-123 which is wrong

minValue <- min(Covid_Df$Recovered)
r1<-which(Covid_Df$Recovered==min(Covid_Df$Recovered))#which returns row numbers

Covid_Df[r1,]$Recovered <- min(Covid_Df$Recovered) * -1 #corrected the value in recovery for that row

summary(Covid_Df$Recovered)

qplot(Recovered, data = Covid_Df, geom = "histogram",ylab="Count", fill=I("green"), col=I("black"))

tail(Covid_Df[order(Covid_Df$Recovered),]) #shows the highest values of Recovered count
```

There are certain rows with the NA values, which are removed from the data frame.
The minimum value when analysed, was an outlier because the number of recovered cases can never be a negative value which is -123.
Hence, we have corrected the outlier by making it positive integer my multiplying with -1.

#Summarization

For summarization we use the Central tendency, and the distribution of the continuous variable is as shown below.



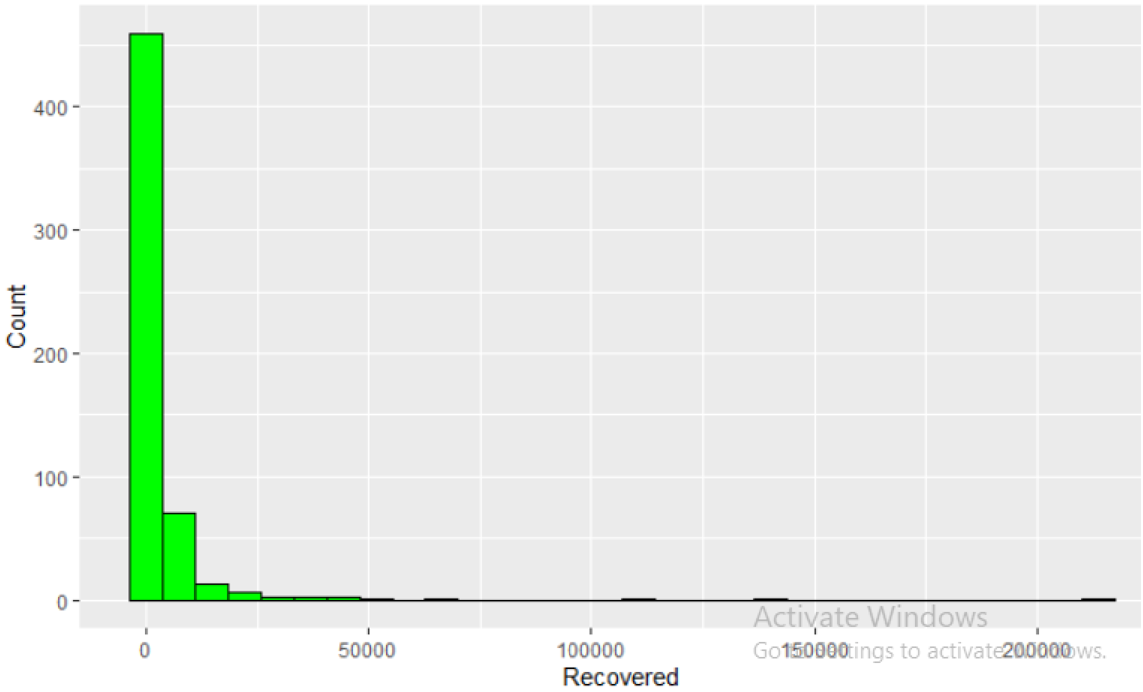
```
> min(Covid_Df$Recovered) #-123 which is wrong
[1] -123
> minValue <- min(Covid_Df$Recovered)
> r1<-which(Covid_Df$Recovered==min(Covid_Df$Recovered))#which returns row numbers
> Covid_Df[r1,]$Recovered <- min(Covid_Df$Recovered) * -1 #corrected the value in recovery for that row
> summary(Covid_Df$Recovered)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      1     221     655    3609    2129  213860 
```

#Visualization

We use a histogram for viewing the distribution of the continuous variable and the result obtained from our data set is as shown below.

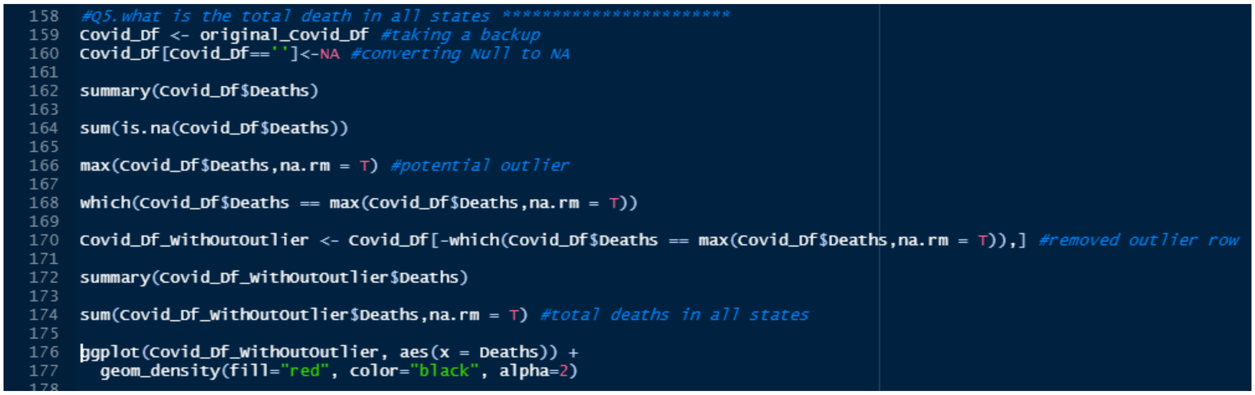
The trend shows that majority of recovered cases falls within the range of 0 to 25000.

However, we could see some small population of recovery at farther points of 200K.



5. what is the total death in all states?

We are to find the total people died out of this pandemic in all the states over the period.



```
#Q5.what is the total death in all states ************************
Covid_Df <- original_Covid_Df #taking a backup
Covid_Df[Covid_Df=='']<-NA #converting Null to NA

summary(Covid_Df$Deaths)

sum(is.na(Covid_Df$Deaths))

max(Covid_Df$Deaths,na.rm = T) #potential outlier

which(Covid_Df$Deaths == max(Covid_Df$Deaths,na.rm = T))

Covid_Df_WithoutOutlier <- Covid_Df[-which(Covid_Df$Deaths == max(Covid_Df$Deaths,na.rm = T)),] #removed outlier row

summary(Covid_Df_WithoutOutlier$Deaths)

sum(Covid_Df_WithoutOutlier$Deaths,na.rm = T) #total deaths in all states

ggplot(Covid_Df_WithoutOutlier, aes(x = Deaths)) +
  geom_density(fill="red", color="black", alpha=2)
```

The number of deaths is analyzed by taking the basic aggregate functions as well as summary function.

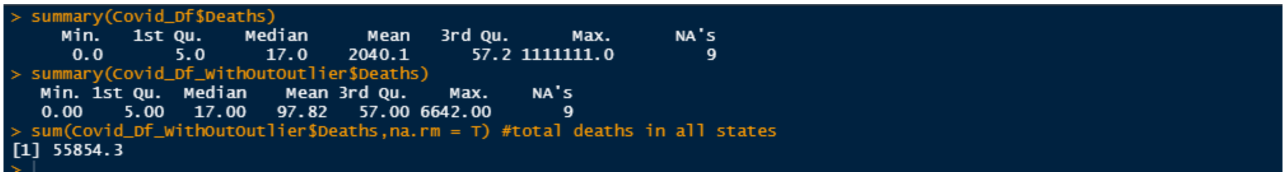
From the initial summary it was clear that max value of the deaths is an outlier as the value is incorrect.

Hence, we have removed the row containing the outlier as it is just 1 row and it would not potentially impact our analysis.

The cleaned data frame is then summarized as shown below.

#Summarization

For summarization we use the sum function that returns the sum of all deaths from the column and is shown in the below figure.

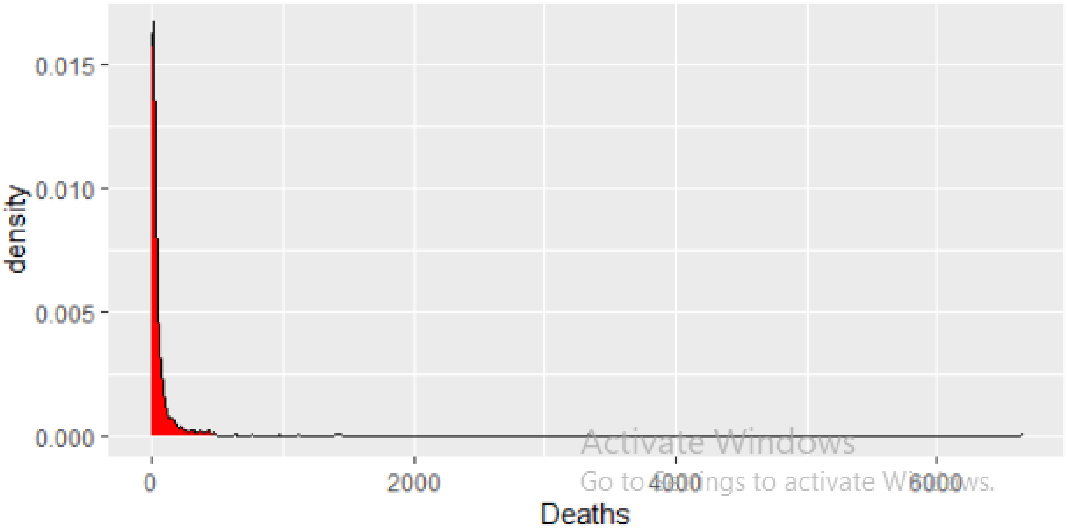


```
> summary(Covid_Df$Deaths)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
      0.0       5.0      17.0    2040.1      57.2 1111111.0         9
> summary(Covid_Df_WithOutOutlier$Deaths)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   0.00    5.00   17.00   97.82   57.00 6642.00       9
> sum(Covid_Df_WithOutOutlier$Deaths,na.rm = T) #total deaths in all states
[1] 55854.3
>
```

The sum of the column is taken by avoiding the NA values in the column.

#Visualization

We use a density plot (*using ggplot*) for viewing the distribution of the continuous variable and the result obtained from our data set is as shown below.



BIVARIATE ANALYSIS

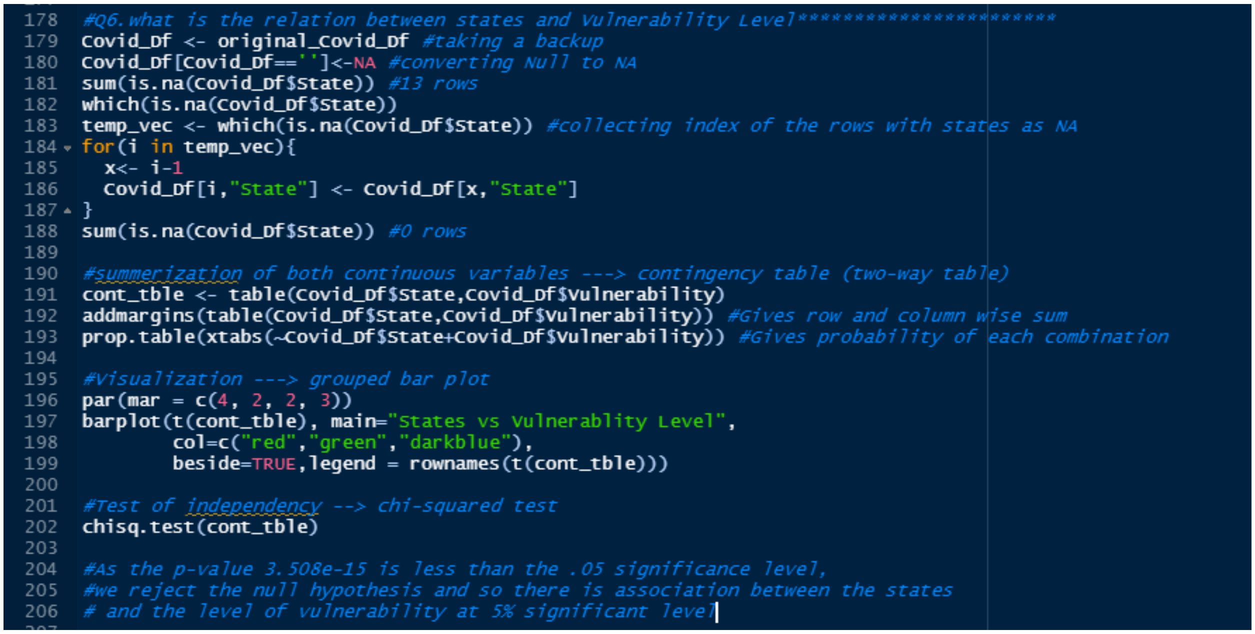
6. what is the relation between states and Vulnerability Level?

Here is the relationship between 2 categorical columns.

The Vulnerability column being the target, doesn't have any "NA" values.

We have 13 rows in the data that has "NA" for the state column.

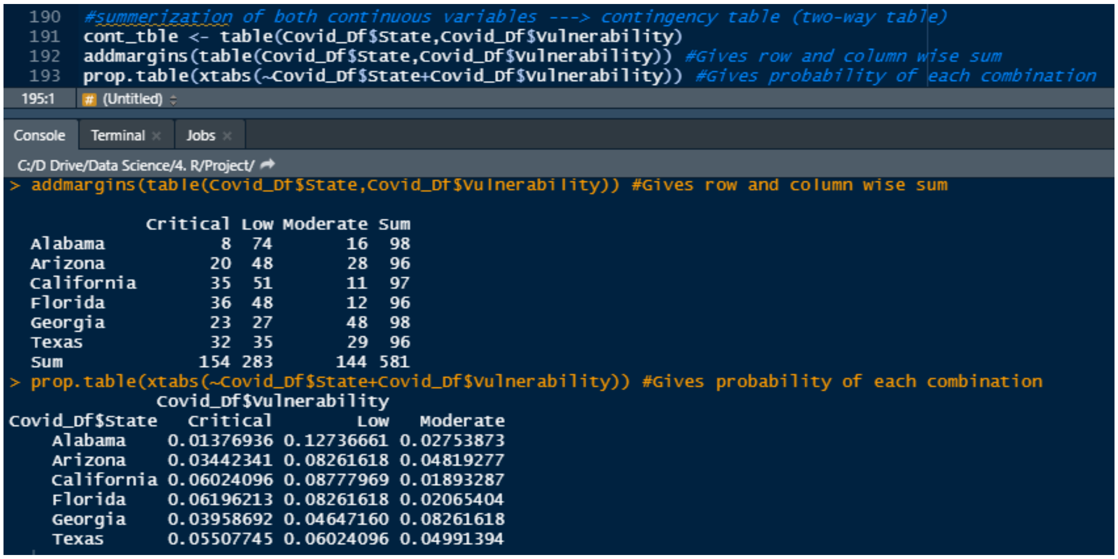
We have filled the missing values from the value of the adjacent row to it.



```
#Q6.what is the relation between states and Vulnerability Level***********************
Covid_Df <- original_Covid_Df #taking a backup
Covid_Df[Covid_Df=='']<-NA #converting Null to NA
sum(is.na(Covid_Df$State)) #13 rows
which(is.na(Covid_Df$State))
temp_vec <- which(is.na(Covid_Df$State)) #collecting index of the rows with states as NA
for(i in temp_vec){
  x<- i-1
  Covid_Df[i,"State"] <- Covid_Df[x,"State"]
}
sum(is.na(Covid_Df$State)) #0 rows

#summerization of both continuous variables ---> contingency table (two-way table)
cont_tble <- table(Covid_Df$State,Covid_Df$Vulnerability)
addmargins(table(Covid_Df$State,Covid_Df$Vulnerability)) #Gives row and column wise sum
prop.table(xtabs(~Covid_Df$State+Covid_Df$Vulnerability)) #Gives probability of each combination

#Visualization ---> grouped bar plot
par(mar = c(4, 2, 2, 3))
barplot(t(cont_tble), main="States vs Vulnerablity Level",
        col=c("red","green","darkblue"),
        beside=TRUE,legend = rownames(t(cont_tble)))

#Test of independency --> chi-squared test
chisq.test(cont_tble)

#As the p-value 3.508e-15 is less than the .05 significance level,
#we reject the null hypothesis and so there is association between the states
# and the level of vulnerability at 5% significant level
```

#Summarization

For summarization we use the contingency table (two-way table).

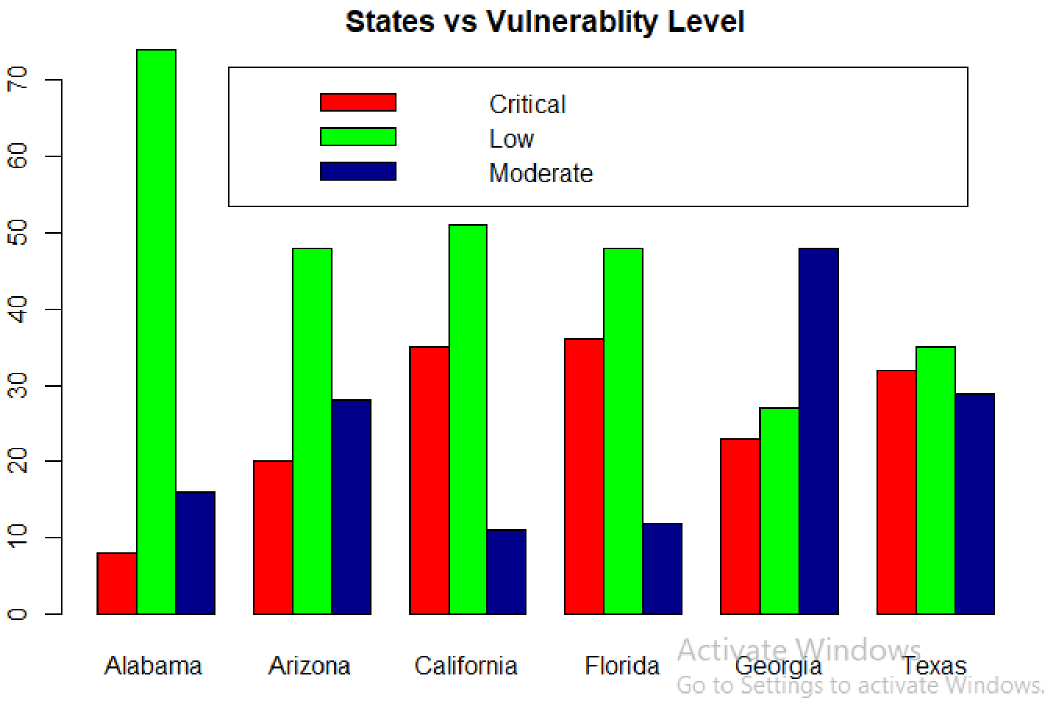


```
#summerization of both continuous variables ---> contingency table (two-way table)
cont_tble <- table(Covid_Df$State,Covid_Df$Vulnerability)
addmargins(table(Covid_Df$State,Covid_Df$Vulnerability)) #Gives row and column wise sum
prop.table(xtabs(~Covid_Df$State+Covid_Df$Vulnerability)) #Gives probability of each combination
```

```
> addmargins(table(Covid_Df$State,Covid_Df$Vulnerability)) #Gives row and column wise sum

             Critical Low Moderate Sum
  Alabama           8  74       16  98
  Arizona          20  48       28  96
  California       35  51       11  97
  Florida          36  48       12  96
  Georgia          23  27       48  98
  Texas            32  35       29  96
  Sum             154 283      144 581
> prop.table(xtabs(~Covid_Df$State+Covid_Df$Vulnerability)) #Gives probability of each combination
              Covid_Df$Vulnerability
Covid_Df$State   Critical        Low   Moderate
    Alabama    0.01376936 0.12736661 0.02753873
    Arizona    0.03442341 0.08261618 0.04819277
    California 0.06024096 0.08777969 0.01893287
    Florida    0.06196213 0.08261618 0.02065404
    Georgia    0.03958692 0.04647160 0.08261618
    Texas      0.05507745 0.06024096 0.04991394
```

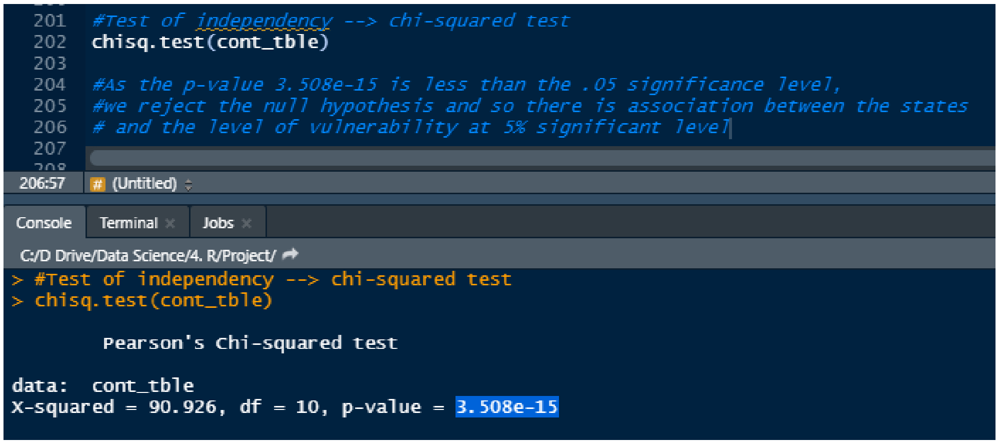
#Visualization

We use a group bar char for plotting categorical vs categorical variable and the result obtained from our data set is as shown below.



#Test of Independence

For categorical vs categorical, chi-squared test is conducted with the two-way table with the frequency is given as the input.

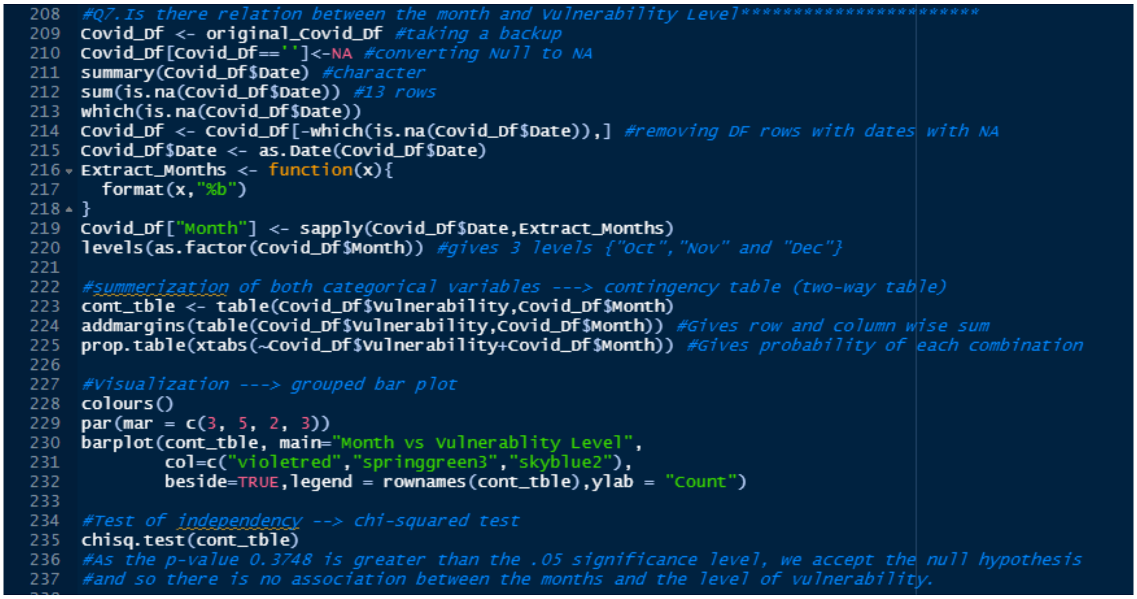

```
#Test of independency --> chi-squared test
chisq.test(cont_tble)

#As the p-value 3.508e-15 is less than the .05 significance level,
#we reject the null hypothesis and so there is association between the states
# and the level of vulnerability at 5% significant level
```

```
> #Test of independency --> chi-squared test
> chisq.test(cont_tble)

	Pearson's Chi-squared test

data:  cont_tble
X-squared = 90.926, df = 10, p-value = 3.508e-15
```

As the p-value 3.508e-15 is less than the .05 significance level, we reject the null hypothesis and so there is association between the states and the level of vulnerability at 5% significant level.

7. Is there relation between the month and Vulnerability Level?

The relation between months (October, November, and December) with the vulnerability level is to be found.



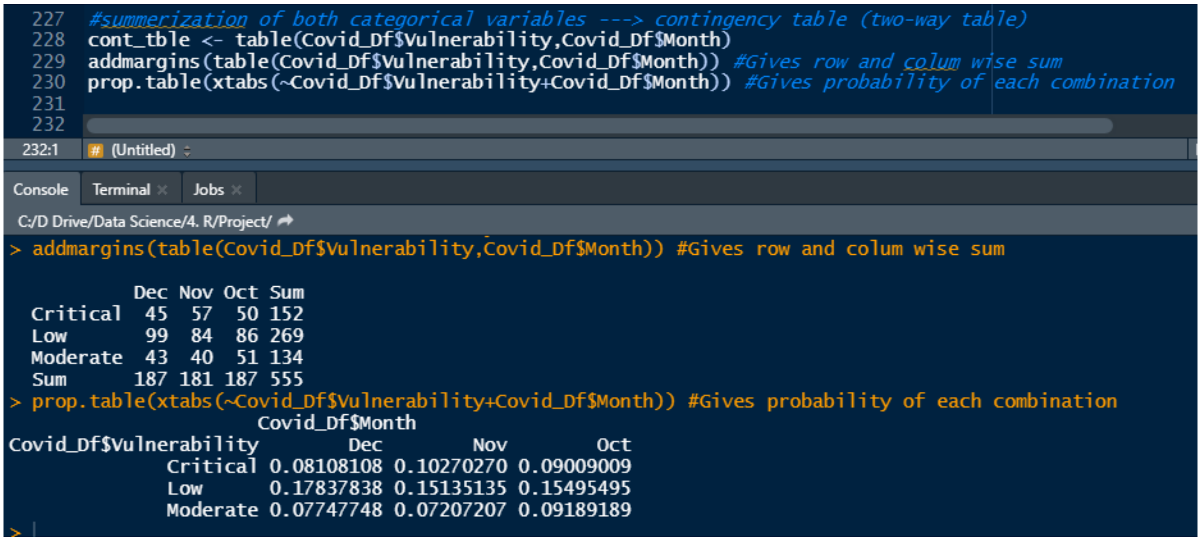
```
#Q7.Is there relation between the month and Vulnerability Level************************
Covid_Df <- original_Covid_Df #taking a backup
Covid_Df[Covid_Df=='']<-NA #converting Null to NA
summary(Covid_Df$Date) #character
sum(is.na(Covid_Df$Date)) #13 rows
which(is.na(Covid_Df$Date))
Covid_Df <- Covid_Df[-which(is.na(Covid_Df$Date)),] #removing DF rows with dates with NA
Covid_Df$Date <- as.Date(Covid_Df$Date)
Extract_Months <- function(x){
  format(x,"%b")
}
Covid_Df["Month"] <- sapply(Covid_Df$Date,Extract_Months)
levels(as.factor(Covid_Df$Month)) #gives 3 levels {"Oct","Nov" and "Dec"}

#summerization of both categorical variables ---> contingency table (two-way table)
cont_tble <- table(Covid_Df$Vulnerability,Covid_Df$Month)
addmargins(table(Covid_Df$Vulnerability,Covid_Df$Month)) #Gives row and column wise sum
prop.table(xtabs(~Covid_Df$Vulnerability+Covid_Df$Month)) #Gives probability of each combination

#Visualization ---> grouped bar plot
colours()
par(mar = c(3, 5, 2, 3))
barplot(cont_tble, main="Month vs Vulnerablity Level",
        col=c("violetred","springgreen3","skyblue2"),
        beside=TRUE,legend = rownames(cont_tble),ylab = "Count")

#Test of independency --> chi-squared test
chisq.test(cont_tble)
#As the p-value 0.3748 is greater than the .05 significance level, we accept the null hypothesis
#and so there is no association between the months and the level of vulnerability.
```

There are 13 rows with dates values as “NA” and they are removed.

The month is extracted using format function and new Month column is added the data frame.

#Summarization

For summarization we use the contingency table (two-way table).

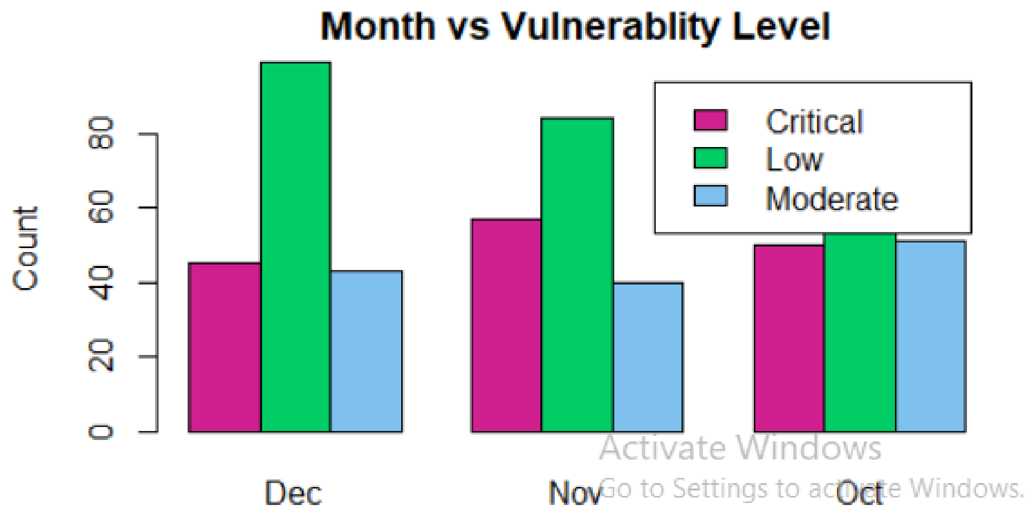


```
#summerization of both categorical variables ---> contingency table (two-way table)
cont_tble <- table(Covid_Df$Vulnerability,Covid_Df$Month)
addmargins(table(Covid_Df$Vulnerability,Covid_Df$Month)) #Gives row and colum wise sum
prop.table(xtabs(~Covid_Df$Vulnerability+Covid_Df$Month)) #Gives probability of each combination
```

```
> addmargins(table(Covid_Df$Vulnerability,Covid_Df$Month)) #Gives row and colum wise sum
          
           Dec Nov Oct Sum
  Critical  45  57  50 152
  Low       99  84  86 269
  Moderate  43  40  51 134
  Sum      187 181 187 555
> prop.table(xtabs(~Covid_Df$Vulnerability+Covid_Df$Month)) #Gives probability of each combination
                      Covid_Df$Month
Covid_Df$Vulnerability        Dec        Nov        Oct
              Critical 0.08108108 0.10270270 0.09009009
              Low      0.17837838 0.15135135 0.15495495
              Moderate 0.07747748 0.07207207 0.09189189
> 
```

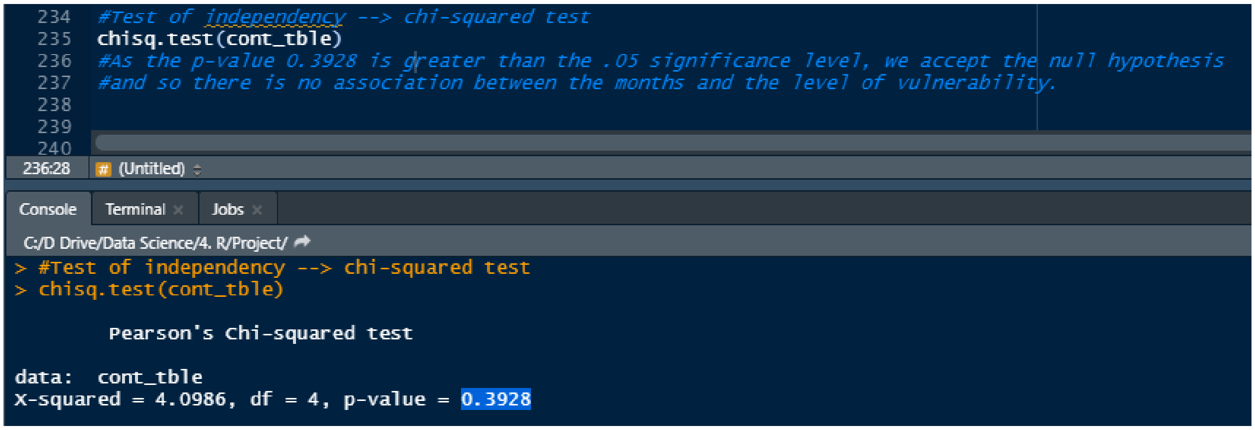
#Visualization

We use a group bar char for plotting categorical vs categorical variable and the frequency obtained from our data set is as shown below.



#Test of Independence

For categorical vs categorical, chi-squared test is conducted with the two-way table with the frequency is given as the input.



```
#Test of independency --> chi-squared test
chisq.test(cont_tble)
#As the p-value 0.3928 is greater than the .05 significance level, we accept the null hypothesis
#and so there is no association between the months and the level of vulnerability.
```

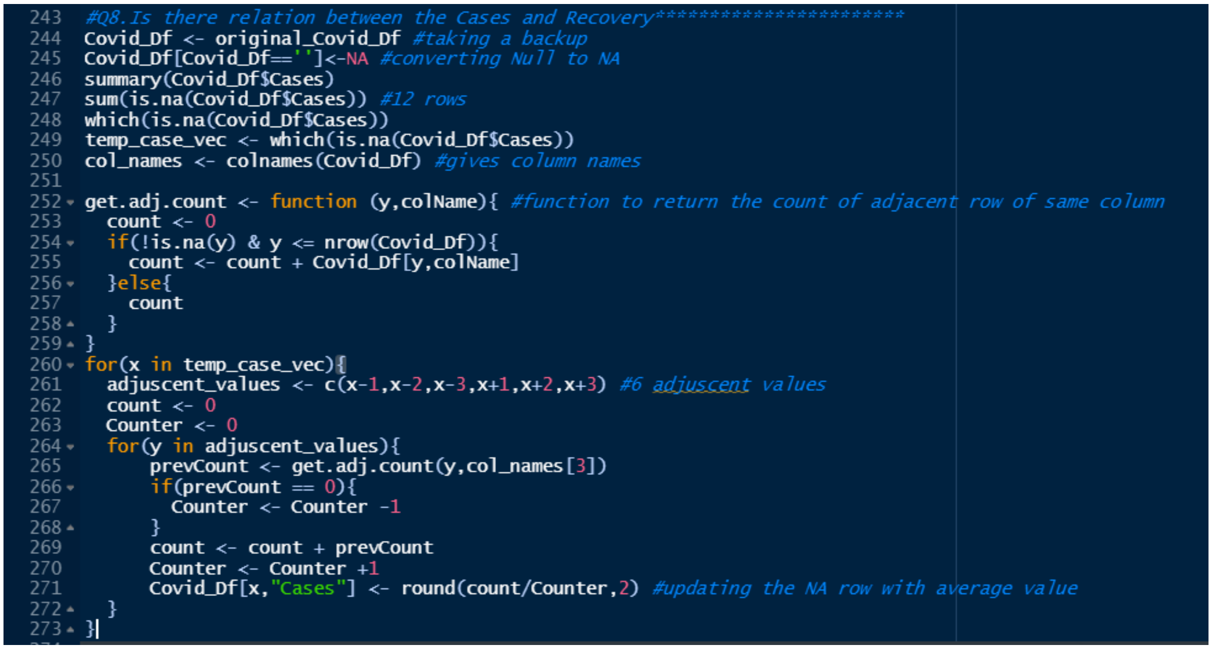
```
> #Test of independency --> chi-squared test
> chisq.test(cont_tble)

	Pearson's Chi-squared test

data:  cont_tble
X-squared = 4.0986, df = 4, p-value = 0.3928
```

As the p-value 0.3928 is greater than the .05 significance level, we accept the null hypothesis and so there is no association between the months and the level of vulnerability.

We might need a bigger data set to do further test and figure out any other relation between them.

8. Is there relation between the Cases and Recovery?

The task is to find is the COVID cases reported and the number of recovered cases have any relation with each other.

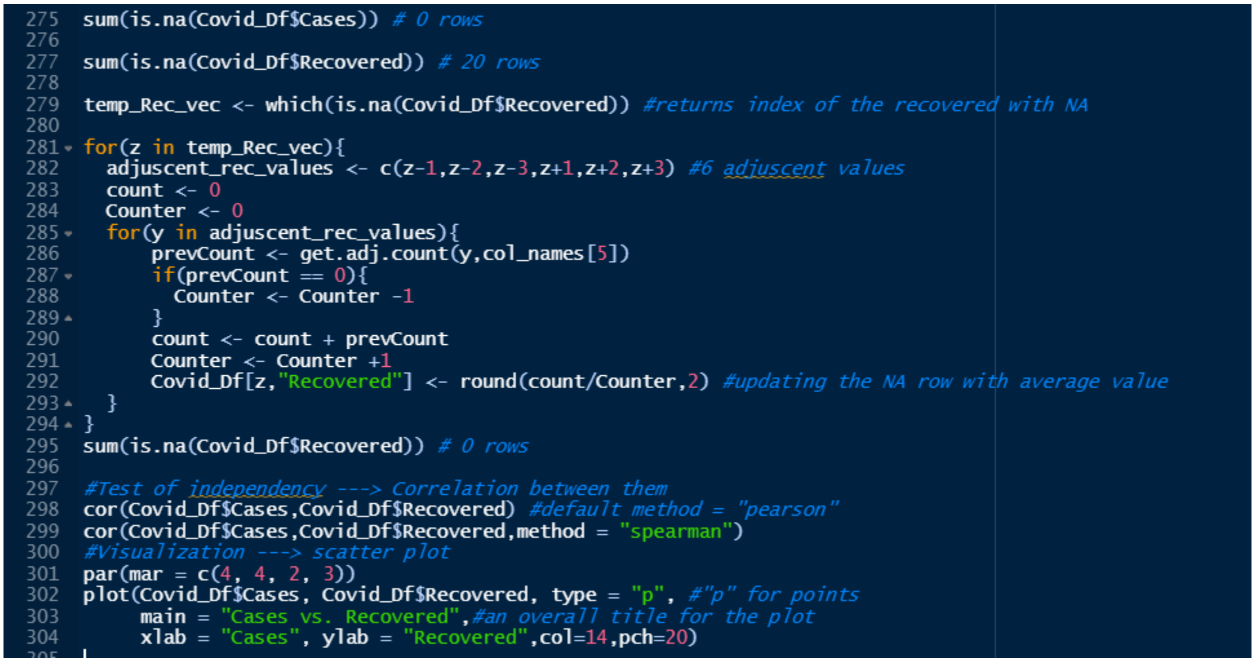


```r
#Q8.Is there relation between the Cases and Recovery*************************
Covid_Df <- original_Covid_Df #taking a backup
Covid_Df[Covid_Df=='']<-NA #converting Null to NA
summary(Covid_Df$Cases)
sum(is.na(Covid_Df$Cases)) #12 rows
which(is.na(Covid_Df$Cases))
temp_case_vec <- which(is.na(Covid_Df$Cases))
col_names <- colnames(Covid_Df) #gives column names

get.adj.count <- function (y,colName){ #function to return the count of adjacent row of same column
  count <- 0
  if(!is.na(y) & y <= nrow(Covid_Df)){
    count <- count + Covid_Df[y,colName]
  }else{
    count
  }
}
for(x in temp_case_vec){
  adjuscent_values <- c(x-1,x-2,x-3,x+1,x+2,x+3) #6 adjuscent values
  count <- 0
  Counter <- 0
  for(y in adjuscent_values){
      prevCount <- get.adj.count(y,col_names[3])
      if(prevCount == 0){
        Counter <- Counter -1
      }
      count <- count + prevCount
      Counter <- Counter +1
      Covid_Df[x,"Cases"] <- round(count/Counter,2) #updating the NA row with average value
  }
}
```

There are 13 rows of cases with “NA” and 20 rows of Recovered with “NA”.

To get the best value to be filled in them, a code is written that will take the mean of the 6 adjacent rows of that “NA”.

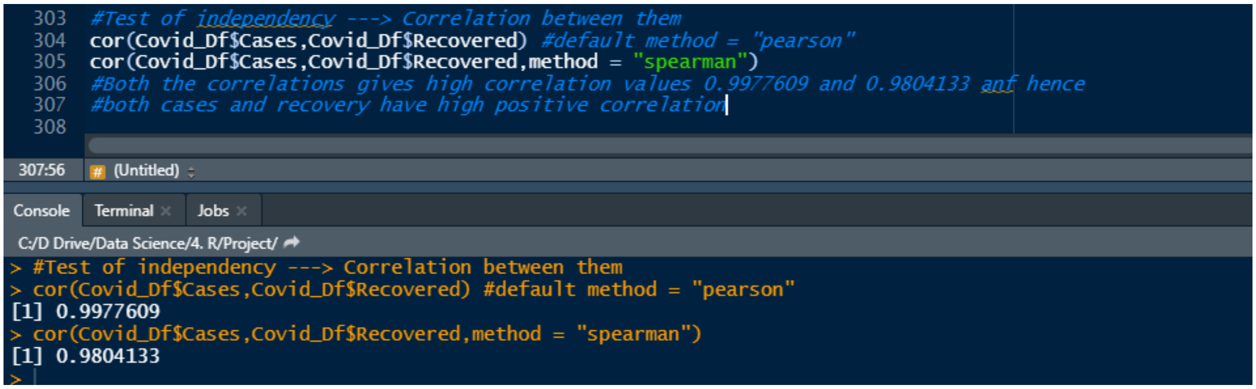
The data frame is cleaned and is used for finding correlation between the variables.



```r
sum(is.na(Covid_Df$Cases)) # 0 rows

sum(is.na(Covid_Df$Recovered)) # 20 rows

temp_Rec_vec <- which(is.na(Covid_Df$Recovered)) #returns index of the recovered with NA

for(z in temp_Rec_vec){
  adjuscent_rec_values <- c(z-1,z-2,z-3,z+1,z+2,z+3) #6 adjuscent values
  count <- 0
  Counter <- 0
  for(y in adjuscent_rec_values){
      prevCount <- get.adj.count(y,col_names[5])
      if(prevCount == 0){
        Counter <- Counter -1
      }
      count <- count + prevCount
      Counter <- Counter +1
      Covid_Df[z,"Recovered"] <- round(count/Counter,2) #updating the NA row with average value
  }
}
sum(is.na(Covid_Df$Recovered)) # 0 rows

#Test of independency ---> Correlation between them
cor(Covid_Df$Cases,Covid_Df$Recovered) #default method = "pearson"
cor(Covid_Df$Cases,Covid_Df$Recovered,method = "spearman")
#Visualization ---> scatter plot
par(mar = c(4, 4, 2, 3))
plot(Covid_Df$Cases, Covid_Df$Recovered, type = "p", #"p" for points
     main = "Cases vs. Recovered",#an overall title for the plot
     xlab = "Cases", ylab = "Recovered",col=14,pch=20)
```

#Test of Independence

For continuous vs continuous, correlation test (Pearson correlation by default) is conducted and the correlation value (between -1 and 1) decides the association between them.



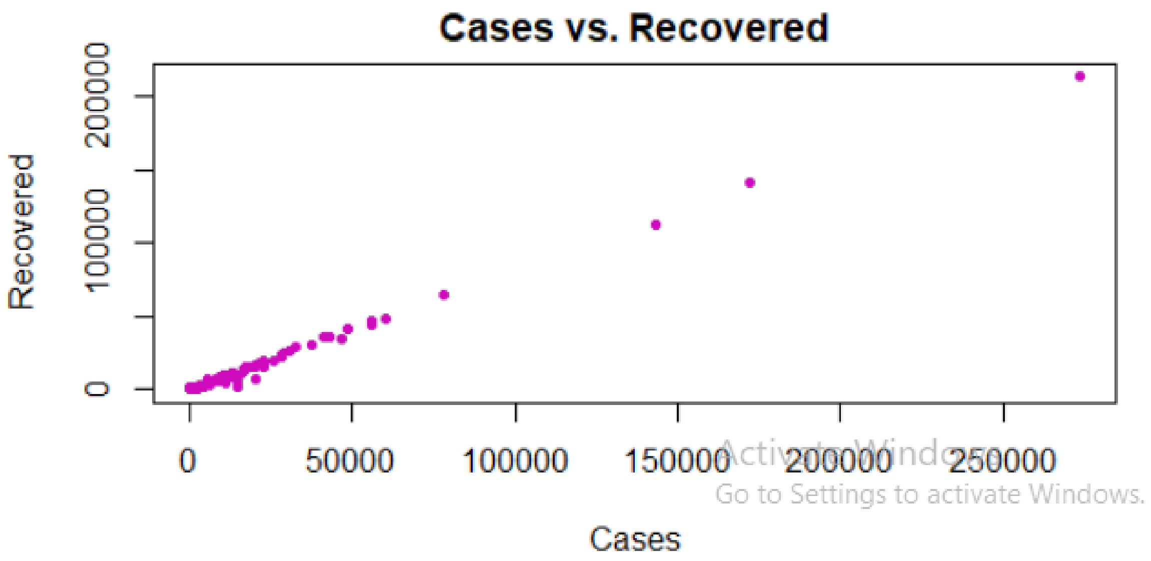
```
#Test of independency ---> Correlation between them
cor(Covid_Df$Cases,Covid_Df$Recovered) #default method = "pearson"
cor(Covid_Df$Cases,Covid_Df$Recovered,method = "spearman")
#Both the correlations gives high correlation values 0.9977609 and 0.9804133 anf hence
#both cases and recovery have high positive correlation
```

```
> #Test of independency ---> Correlation between them
> cor(Covid_Df$Cases,Covid_Df$Recovered) #default method = "pearson"
[1] 0.9977609
> cor(Covid_Df$Cases,Covid_Df$Recovered,method = "spearman")
[1] 0.9804133
>
```

Both the correlations give high correlation values 0.9977609 and 0.9804133 and hence both cases and recovery have high positive correlation

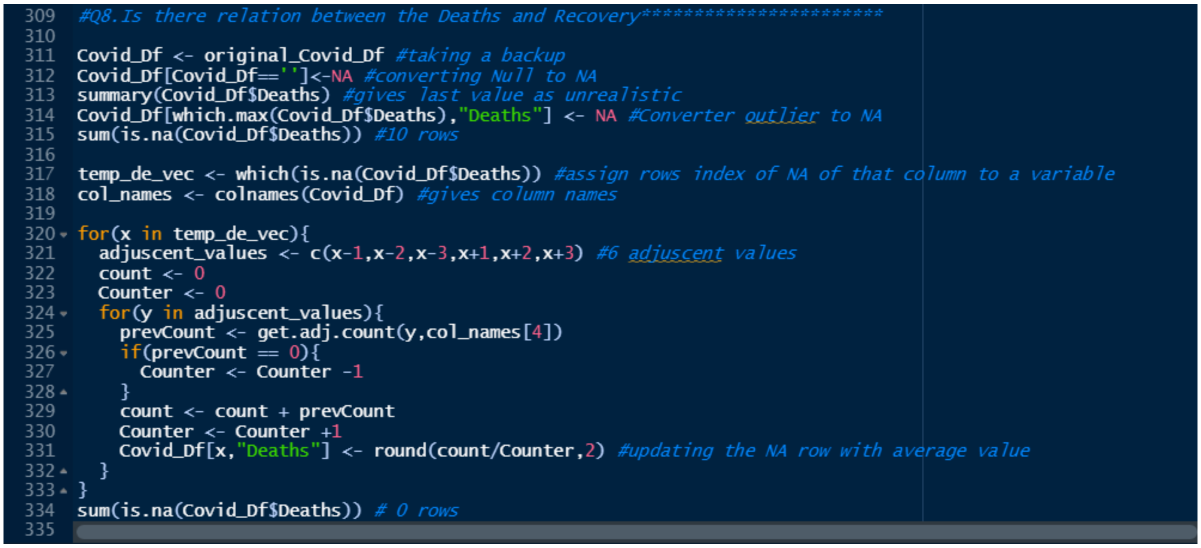
#Visualization

We use a scatter plot to visualize the relation between 2 continuous columns. The high positive correlation between them is well displayed in the plot.

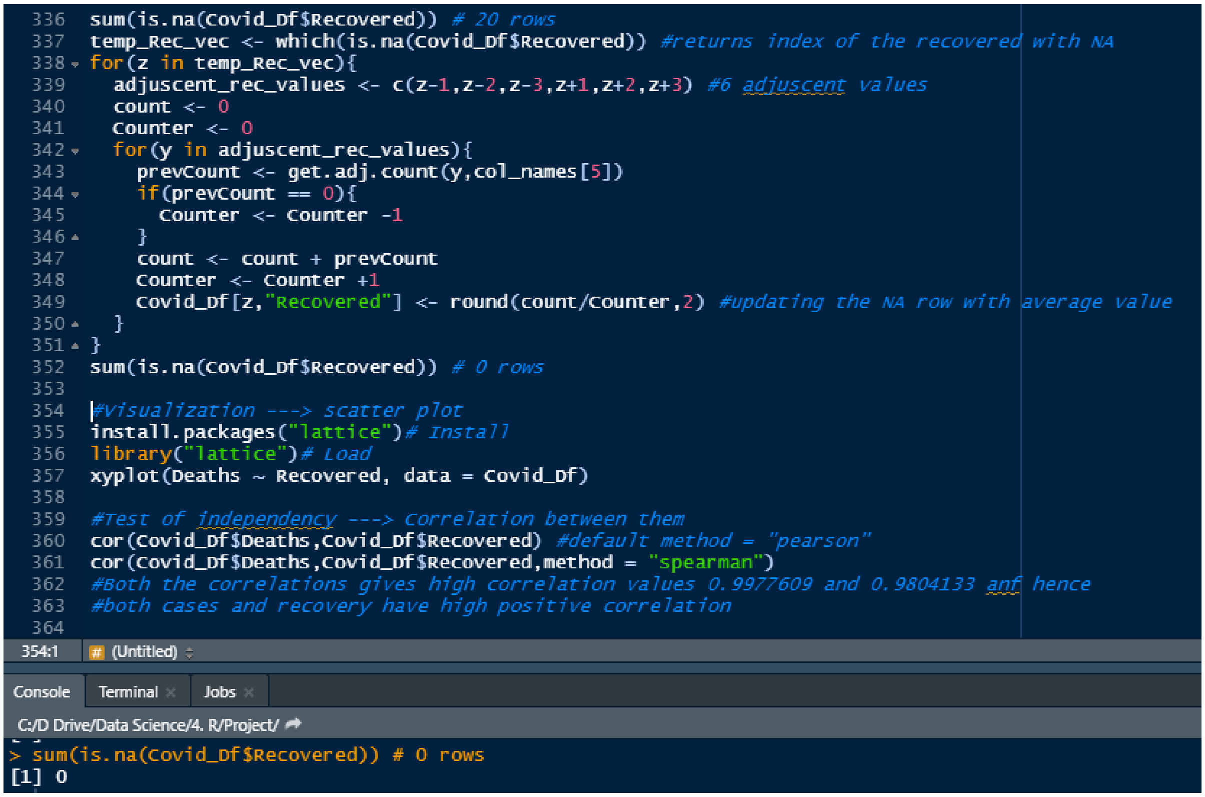


9. Is there relation between the Deaths and Recovery?

The task is to find is the COVID deaths reported and the number of recovered cases have any relation with each other.



```r
#Q8.Is there relation between the Deaths and Recovery************************

Covid_Df <- original_Covid_Df #taking a backup
Covid_Df[Covid_Df=='']<-NA #converting Null to NA
summary(Covid_Df$Deaths) #gives last value as unrealistic
Covid_Df[which.max(Covid_Df$Deaths),"Deaths"] <- NA #Converter outlier to NA
sum(is.na(Covid_Df$Deaths)) #10 rows

temp_de_vec <- which(is.na(Covid_Df$Deaths)) #assign rows index of NA of that column to a variable
col_names <- colnames(Covid_Df) #gives column names

for(x in temp_de_vec){
  adjuscent_values <- c(x-1,x-2,x-3,x+1,x+2,x+3) #6 adjuscent values
  count <- 0
  Counter <- 0
  for(y in adjuscent_values){
    prevCount <- get.adj.count(y,col_names[4])
    if(prevCount == 0){
      Counter <- Counter -1
    }
    count <- count + prevCount
    Counter <- Counter +1
    Covid_Df[x,"Deaths"] <- round(count/Counter,2) #updating the NA row with average value
  }
}
sum(is.na(Covid_Df$Deaths)) # 0 rows

```



```r
sum(is.na(Covid_Df$Recovered)) # 20 rows
temp_Rec_vec <- which(is.na(Covid_Df$Recovered)) #returns index of the recovered with NA
for(z in temp_Rec_vec){
  adjuscent_rec_values <- c(z-1,z-2,z-3,z+1,z+2,z+3) #6 adjuscent values
  count <- 0
  Counter <- 0
  for(y in adjuscent_rec_values){
    prevCount <- get.adj.count(y,col_names[5])
    if(prevCount == 0){
      Counter <- Counter -1
    }
    count <- count + prevCount
    Counter <- Counter +1
    Covid_Df[z,"Recovered"] <- round(count/Counter,2) #updating the NA row with average value
  }
}
sum(is.na(Covid_Df$Recovered)) # 0 rows

#Visualization ---> scatter plot
install.packages("lattice")# Install
library("lattice")# Load
xyplot(Deaths ~ Recovered, data = Covid_Df)

#Test of independency ---> Correlation between them
cor(Covid_Df$Deaths,Covid_Df$Recovered) #default method = "pearson"
cor(Covid_Df$Deaths,Covid_Df$Recovered,method = "spearman")
#Both the correlations gives high correlation values 0.9977609 and 0.9804133 anf hence
#both cases and recovery have high positive correlation

```

```
C:/D Drive/Data Science/4. R/Project/
> sum(is.na(Covid_Df$Recovered)) # 0 rows
[1] 0
```

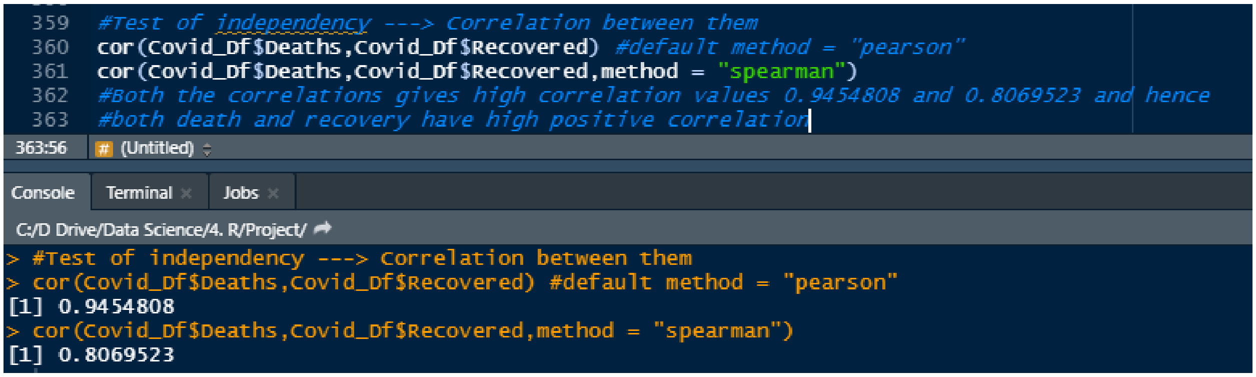
There are 10 “NA” values in the death column and 20 “NA” values in the recovered column.

To get the best value to be filled in them, a code is written that will take the mean of the 6 adjacent rows of that “NA”.

The data frame is cleaned and is used for finding correlation between the variables.

#Test of Independence

For continuous vs continuous, correlation test (Pearson correlation by default) is conducted and the correlation value (between -1 and 1) decides the association between them.



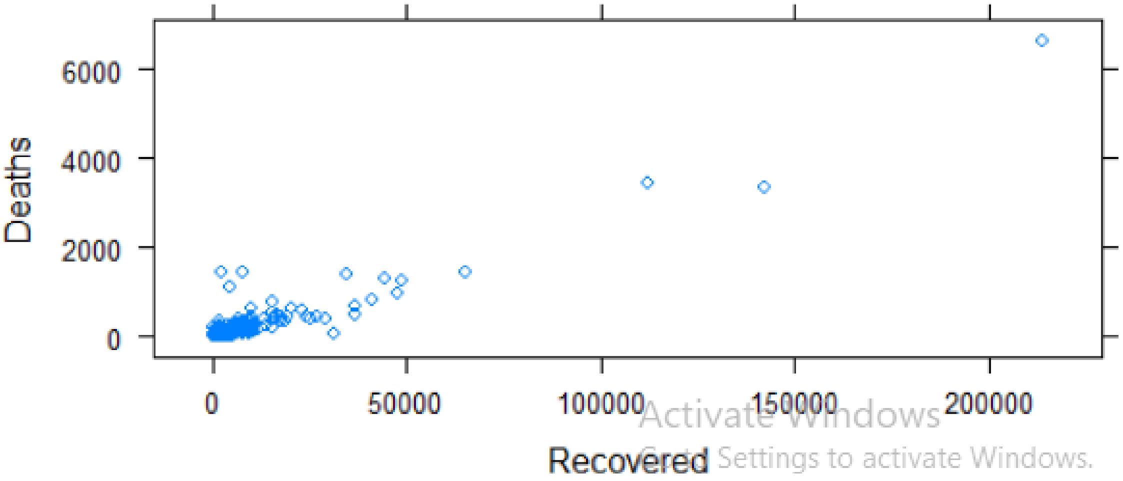
```
#Test of independency ---> Correlation between them
cor(Covid_Df$Deaths,Covid_Df$Recovered) #default method = "pearson"
cor(Covid_Df$Deaths,Covid_Df$Recovered,method = "spearman")
#Both the correlations gives high correlation values 0.9454808 and 0.8069523 and hence
#both death and recovery have high positive correlation
```

```
> #Test of independency ---> Correlation between them
> cor(Covid_Df$Deaths,Covid_Df$Recovered) #default method = "pearson"
[1] 0.9454808
> cor(Covid_Df$Deaths,Covid_Df$Recovered,method = "spearman")
[1] 0.8069523
```

Both the correlations give high correlation values 0.9454808 and 0.8069523 and hence both death and recovery have high positive correlation.

#Visualization

We use a scatter plot to visualize the relation between 2 continuous columns. The high positive correlation between them is well displayed in the plot.

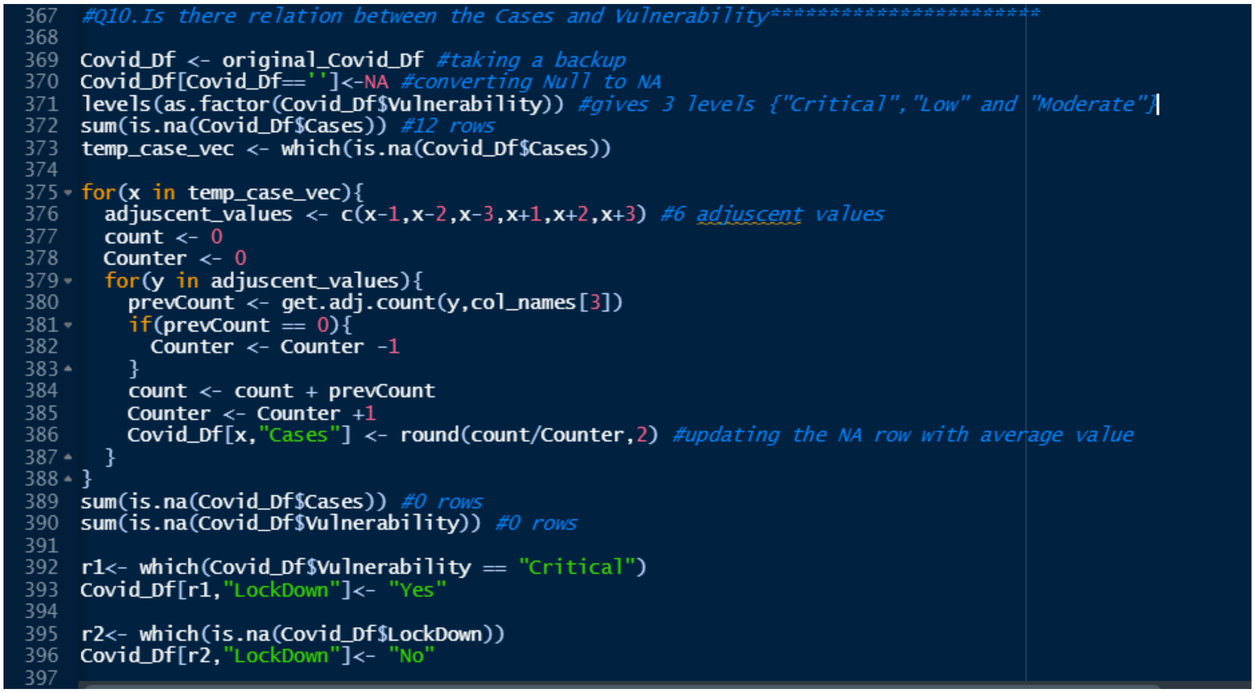


Library lattice and XY plot is used to obtain the above pattern of scatter plot.

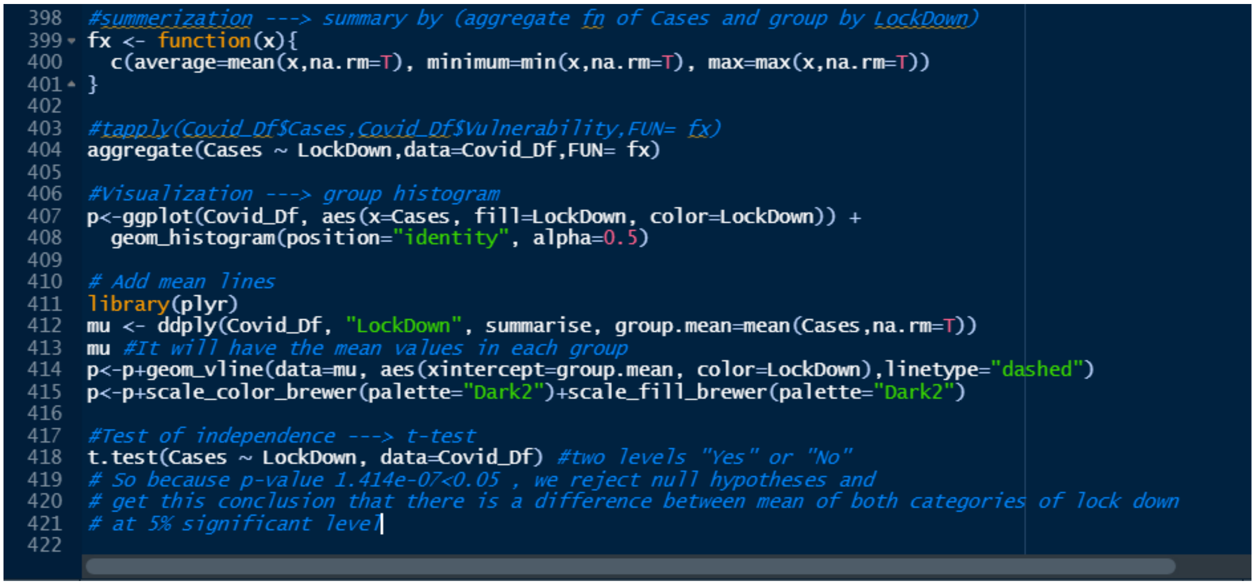
10. Is there relation between the Cases and Vulnerability?

The goal is to find the relation between the total number of cases reported and vulnerability level.

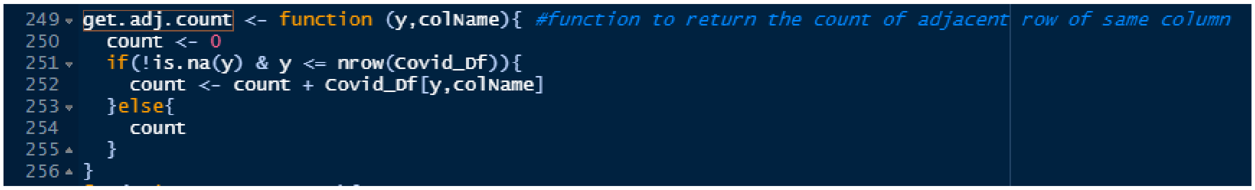
This is relation between continuous vs categorical column.



```r
#Q10.Is there relation between the Cases and Vulnerability***********************

Covid_Df <- original_Covid_Df #taking a backup
Covid_Df[Covid_Df=='']<-NA #converting Null to NA
levels(as.factor(Covid_Df$Vulnerability)) #gives 3 levels {"Critical","Low" and "Moderate"}
sum(is.na(Covid_Df$Cases)) #12 rows
temp_case_vec <- which(is.na(Covid_Df$Cases))

for(x in temp_case_vec){
  adjuscent_values <- c(x-1,x-2,x-3,x+1,x+2,x+3) #6 adjuscent values
  count <- 0
  Counter <- 0
  for(y in adjuscent_values){
    prevCount <- get.adj.count(y,col_names[3])
    if(prevCount == 0){
      Counter <- Counter -1
    }
    count <- count + prevCount
    Counter <- Counter +1
    Covid_Df[x,"Cases"] <- round(count/Counter,2) #updating the NA row with average value
  }
}
sum(is.na(Covid_Df$Cases)) #0 rows
sum(is.na(Covid_Df$Vulnerability)) #0 rows

r1<- which(Covid_Df$Vulnerability == "Critical")
Covid_Df[r1,"LockDown"]<- "Yes"

r2<- which(is.na(Covid_Df$LockDown))
Covid_Df[r2,"LockDown"]<- "No"

```



```r
#summerization ---> summary by (aggregate fn of Cases and group by LockDown)
fx <- function(x){
  c(average=mean(x,na.rm=T), minimum=min(x,na.rm=T), max=max(x,na.rm=T))
}

#tapply(Covid_Df$Cases,Covid_Df$Vulnerability,FUN= fx)
aggregate(Cases ~ LockDown,data=Covid_Df,FUN= fx)

#Visualization ---> group histogram
p<-ggplot(Covid_Df, aes(x=Cases, fill=LockDown, color=LockDown)) +
  geom_histogram(position="identity", alpha=0.5)

# Add mean lines
library(plyr)
mu <- ddply(Covid_Df, "LockDown", summarise, group.mean=mean(Cases,na.rm=T))
mu #It will have the mean values in each group
p<-p+geom_vline(data=mu, aes(xintercept=group.mean, color=LockDown),linetype="dashed")
p<-p+scale_color_brewer(palette="Dark2")+scale_fill_brewer(palette="Dark2")

#Test of independence ---> t-test
t.test(Cases ~ LockDown, data=Covid_Df) #two levels "Yes" or "No"
# So because p-value 1.414e-07<0.05 , we reject null hypotheses and
# get this conclusion that there is a difference between mean of both categories of lock down
# at 5% significant level

```



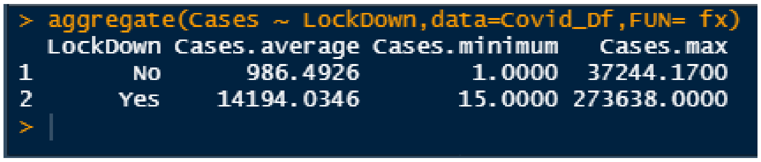
```r
get.adj.count <- function (y,colName){ #function to return the count of adjacent row of same column
  count <- 0
  if(!is.na(y) & y <= nrow(Covid_Df)){
    count <- count + Covid_Df[y,colName]
  }else{
    count
  }
}
```

We have replaced the 12 “NA” values in cases column with mean of adjacent rows.

A new column is created with name “LockDown” and the value of it is “Yes” for critical vulnerability and “No” for low and moderate vulnerability.

#Summarization

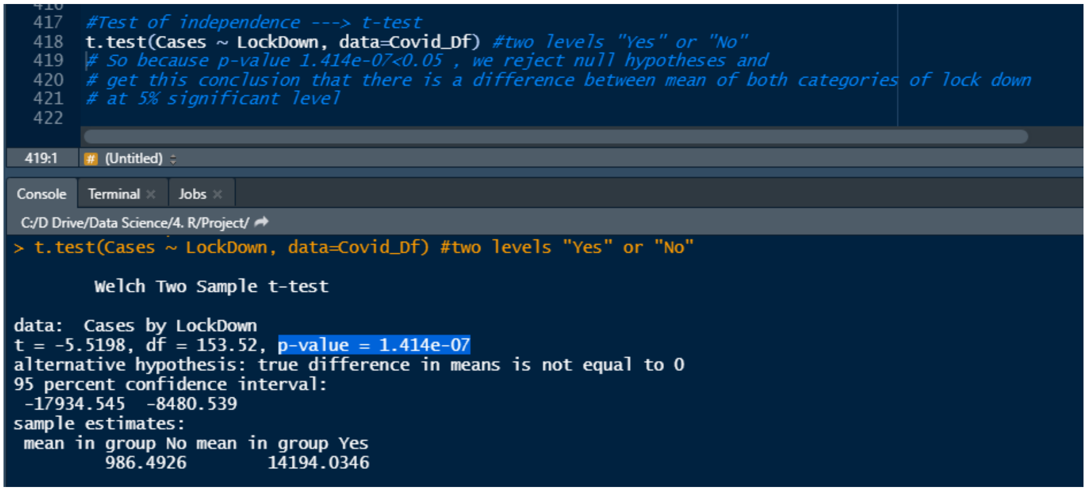
For summarization we use the aggregate function. It groups by the categorical column and takes the aggregate of the continuous column.



```
> aggregate(Cases ~ LockDown,data=Covid_Df,FUN= fx)
  LockDown Cases.average Cases.minimum   Cases.max
1       No      986.4926        1.0000  37244.1700
2      Yes    14194.0346       15.0000 273638.0000
> |
```

#Test of Independence

For continuous vs categorical column, we run t-test to get the relation between them. The t-test is used as the there are only 2 levels in the categorical column.



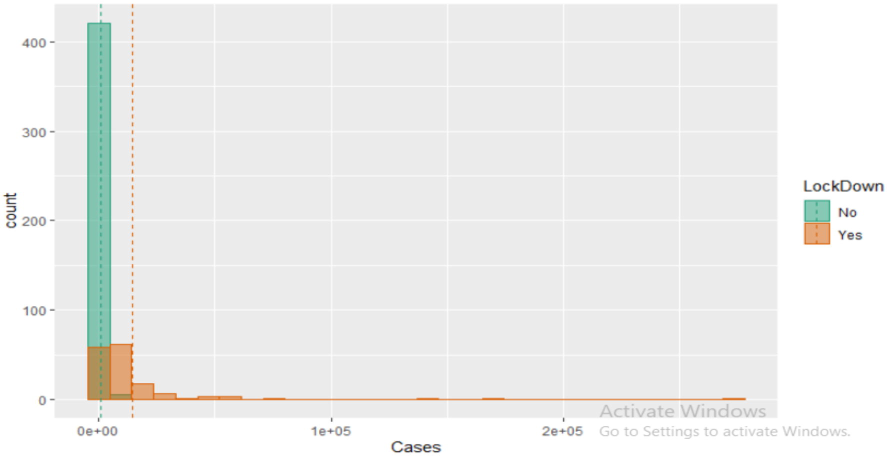
```
#Test of independence ---> t-test
t.test(Cases ~ LockDown, data=Covid_Df) #two levels "Yes" or "No"
# So because p-value 1.414e-07<0.05 , we reject null hypotheses and
# get this conclusion that there is a difference between mean of both categories of lock down
# at 5% significant level

```

```
> t.test(Cases ~ LockDown, data=Covid_Df) #two levels "Yes" or "No"

	Welch Two Sample t-test

data:  Cases by LockDown
t = -5.5198, df = 153.52, p-value = 1.414e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -17934.545  -8480.539
sample estimates:
 mean in group No mean in group Yes
         986.4926        14194.0346
```

As the p-value 1.414e-07<0.05, we reject null hypotheses and get this conclusion that there is a difference between mean of both categories of lock down at 5% significant level.

#Visualization

For visualization we use a group histogram as below and a mean line between two categories is drawn as vertical line.



CONCLUSION

To conclude, the project gave in depth knowledge on the following:

- Nearly half of the vulnerability levels were Low.
- Critical vulnerability was slightly higher than the moderate vulnerability level.
- All 6 states were almost equally distributed over the period.
- There is strong association between states and vulnerability.
- The months and vulnerability are not associated with each other.
- There is high linear correlation between cases and recovery and death levels.
- There is strong association between cases and vulnerability.

Recommendation:

- When critical vulnerability and number of cases are high, there should be lockdown.