Nigeria Covid-19 Analysis using R

Akolade Sofiyyah Iwalewa,22201441

2022-12-20

Introduction to dataset

This a Covid-19 data set gotten from the Nigeria Center for Disease Control and Prevention NCDC’s website. It provides information about the number of confirmed cases, recovered patients, deaths due to Covid, number of active cases according to each state in Nigeria. This is the link to their website: https://www.ncdc.gov.ng (https://www.ncdc.gov.ng)

PART 1: ANALYSIS

For my analysis, I would be providing answers to the following questions as a guide.

- TASK 1: MANIPULATION

1. Load the data set (R project dataset)
2. Covert the geographical zone column to a factor.
3. What is the size (number of rows and columns) and the structure of this dataset?
4. Are there any missing values in the dataset?

- TASK 2: ANALYSIS

5. Which state has the highest number of confirmed cases, hence determine:

i. The percentage of patients who recovered from that state
ii. The percentage of patients who died from that state

6. which state has the lowest number of confirmed cases, hence determine:

i. The percentage of patients who recovered from that state
ii. The percentage of patients who died from that state

7. With the aid of a chart, can you discuss the relationship between the number of screened individuals and those confirmed for Covid-19

8. With the aid of a chart, can you discuss the relationship between the number of confirmed cases of Covid-19 and the number of people who recovered from Covid-19.

9. Create a summary tables for the data set and interpret your result where necessary.

```
# 1.Loading the dataset into R
covid_df <- read.csv("R project dataset.csv",
                     header = TRUE, # first row contains column names
                     row.names = 1) # first column contains state names
```

```
# Having a look at the data set
head(covid_df)
```

```
##           CONFIRMED RECOVERED DEATHS ACTIVE.CASES SCREENED  Latitude Longitude
## Abia           2030      1990     31            9    37748  5.453302  7.523190
## Adamawa        1157      1098     32           27    29724  9.323227 12.400241
## Akwa Ibom      4348      4076     44          228    46007  4.907245  7.846395
## Anambra        2405      2386     19            0    49787  6.222776  6.932186
## Bauchi         1802      1736     23           43    35826 10.796647  9.990588
## Bayelsa        1250      1208     28           14    34322  4.766315  6.080419
##           AREA.SQUARE.METER GEOGRAPHICAL.ZONE
## Abia               4858.882        South East
## Adamawa           37924.988        North East
## Akwa Ibom          6723.203       South South
## Anambra            4807.933        South East
## Bauchi            48496.401        North East
## Bayelsa            9546.418       South South
```

```
#2.  Converting the geographical zone column to a factor
covid_df$GEOGRAPHICAL.ZONE <- factor(covid_df$GEOGRAPHICAL.ZONE)
```

```
# 3.i Checking the size of the dataset
dim(covid_df)
```

```
## [1] 37  9
```

> This shows that there are 37 rows and 9 columns in the dataset.

```
# 3.ii Checking the dimension of the data set
str(covid_df)
```

```
## 'data.frame':    37 obs. of  9 variables:
##  $ CONFIRMED        : int  2030 1157 4348 2405 1802 1250 1907 1356 662 4149 ...
##  $ RECOVERED        : int  1990 1098 4076 2386 1736 1208 1512 1317 622 2556 ...
##  $ DEATHS           : int  31 32 44 19 23 28 25 38 25 110 ...
##  $ ACTIVE.CASES     : int  9 27 228 0 43 14 370 1 15 1483 ...
##  $ SCREENED         : int  37748 29724 46007 49787 35826 34322 45909 24184 18025 72468 ...
##  $ Latitude         : num  5.45 9.32 4.91 6.22 10.8 ...
##  $ Longitude        : num  7.52 12.4 7.85 6.93 9.99 ...
##  $ AREA.SQUARE.METER: num  4859 37925 6723 4808 48496 ...
##  $ GEOGRAPHICAL.ZONE: Factor w/ 6 levels "North Central",..: 4 2 5 4 2 5 1 2 5 5 ...
```

> The structure shows that the object is a data frame with 37 rows and 9 columns.It also shows that all the columns are either int or numeric except geographical zone which is a factor.

```
#4. Checking if there are missing values in the dataset
is.null(covid_df)
```

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

> This dataset contains no missing values

```
#5a. writing a code  that determines the state with the highest number of confirmed cases.
rownames(covid_df)[which.max(covid_df$CONFIRMED)]
```

```
## [1] "Lagos"
```

> The state with the highest number of confirmed cases is Lagos

```
#5bi. Getting the percentage of patients who recovered from covid in Lagos
covid_df["Lagos",]$RECOVERED / covid_df["Lagos",]$CONFIRMED * 100
```

```
## [1] 98.40973
```

> Approximately 98.4% of the patients recovered from Covid-19 in Lagos state. The recovery rate is high which is a good one.

```
#5bii. Getting the percentage of patients who died from covid in Lagos state
covid_df["Lagos",]$DEATHS / covid_df["Lagos",]$CONFIRMED * 100
```

```
## [1] 0.9633714
```

> Approximately 0.96% of the patients died from Covid-19 in Lagos state.

```
#6a. writing a code that determines the state with the lowest number of confirmed cases.
rownames(covid_df)[which.min(covid_df$CONFIRMED)]
```

```
## [1] "Kogi"
```

> The state with the lowest number of confirmed cases is Kogi

```
#6bi. Getting the percentage of patients who recovered from covid in Kogi state
covid_df["Kogi",]$RECOVERED / covid_df["Kogi",]$CONFIRMED * 100
```

```
## [1] 60
```

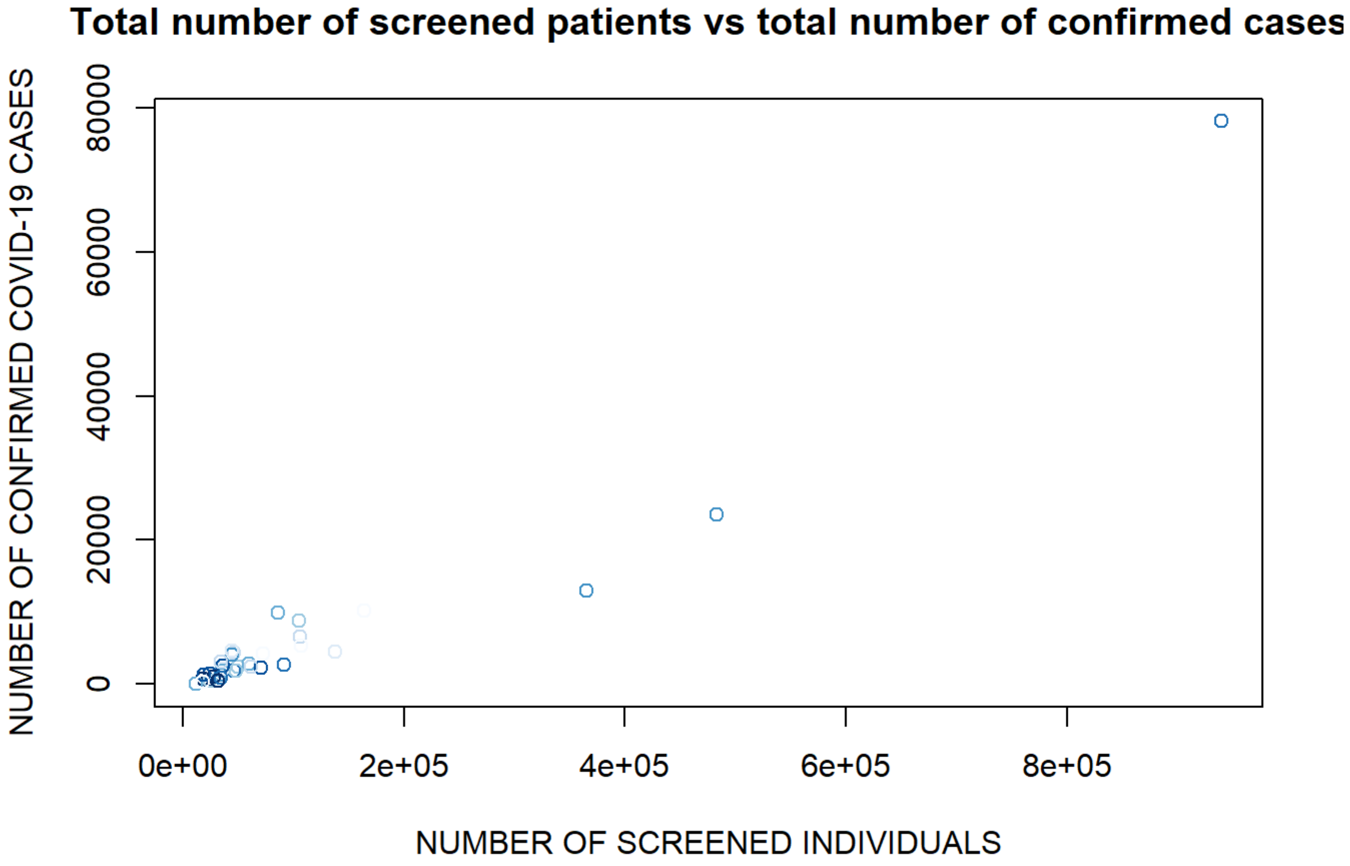
> 60% of the patients recovered from Covid-19 in Kogi state. It could be noted that the recovery rate is not quite high as that of Lagos state but it is fairly above average.

```
#6bii. Getting the percentage of patients who died from covid in Kogi
covid_df["Kogi",]$DEATHS / covid_df["Kogi",]$CONFIRMED * 100
```

```
## [1] 40
```

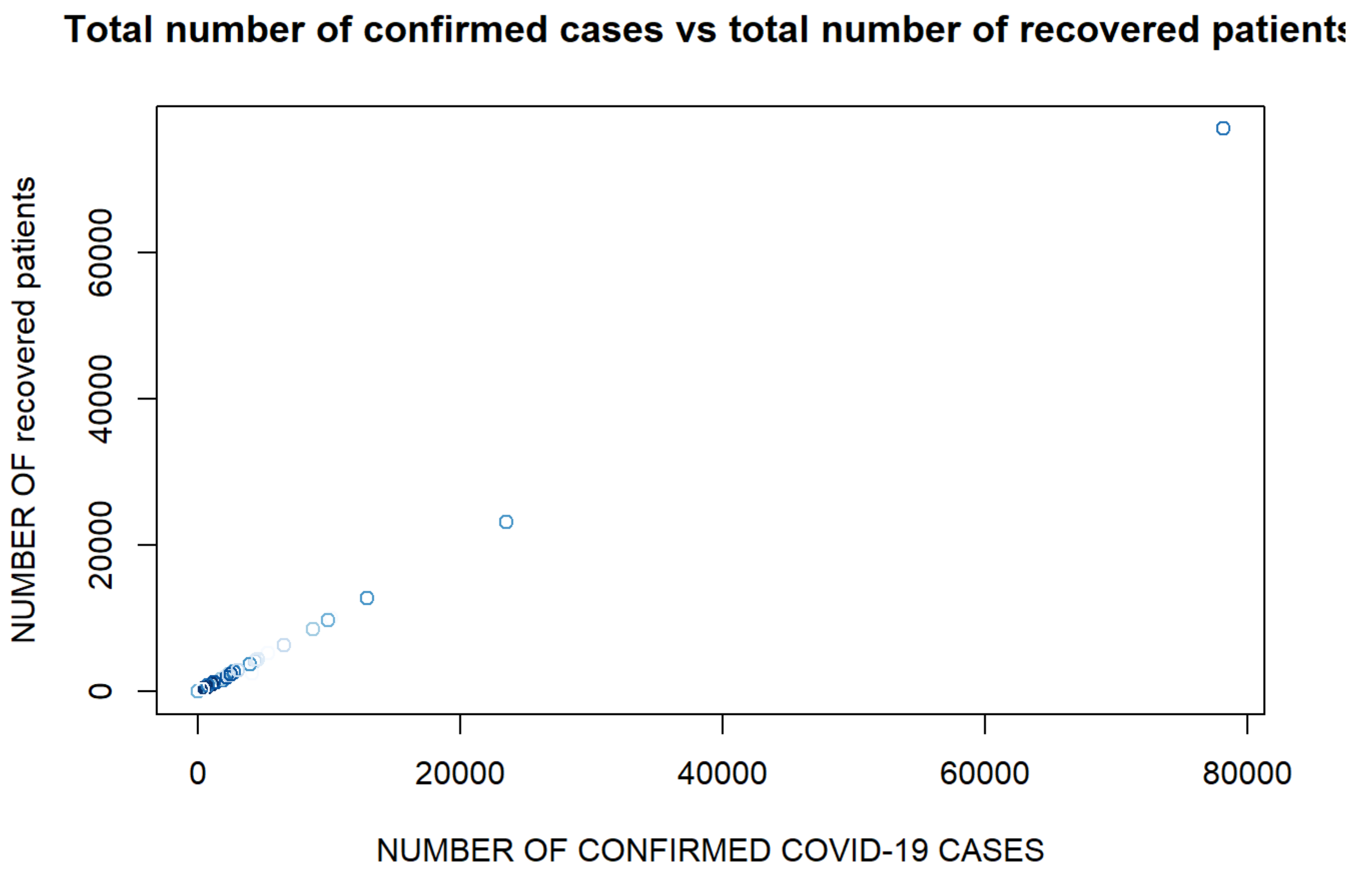
> 40% of the patients died from covid-19 in Kogi state.

```
#7. Creating a scatter plot to check the relationship between number of screened individuals and number of confirmed cases
plot(x=covid_df$SCREENED, y=covid_df$CONFIRMED, main = "Total number of screened patients vs total number of confirmed cases", xlab = "NUMBER OF SCREENED INDIVIDUALS", ylab = "NUMBER OF CONFIRMED COVID-19 CASES", col = blues9 )
```



> There seem to be a strong positive relationship between the number of screened individuals and the number of confirmed cases, this means that most people who were screened for Covid-19 turned out to be postive.

```
#8. Creating a scatter plot to check the relationship between the number of confirmed cases of Covid-19 and those who recovered from it.
plot(x=covid_df$CONFIRMED, y=covid_df$RECOVERED, main = "Total number of confirmed cases vs total number of recovered patients", xlab = "NUMBER OF CONFIRMED COVID-19 CASES", ylab = "NUMBER OF recovered patients", col = blues9 )
```



> There is a very strong positive relationship between the number of confirmed cases of Covid-19 and the number of patients who recovered from it. This implies that the recovery rate accross the country is very high.

```
#9. Creating a summary table for the dataset
summary(covid_df)
```

```
##    CONFIRMED       RECOVERED        DEATHS       ACTIVE.CASES
##  Min.   :    5   Min.   :    3   Min.   :  2.00   Min.   :   0.0
##  1st Qu.: 1250   1st Qu.: 1208   1st Qu.: 25.00   1st Qu.:   5.0
##  Median : 2356   Median : 2210   Median : 37.00   Median :  18.0
##  Mean   : 5786   Mean   : 5601   Mean   : 80.43   Mean   : 104.4
##  3rd Qu.: 4429   3rd Qu.: 4264   3rd Qu.: 81.00   3rd Qu.:  71.0
##  Max.   :78163   Max.   :76920   Max.   :753.00   Max.   :1483.0
##     SCREENED         Latitude        Longitude      AREA.SQUARE.METER
##  Min.   : 11759   Min.   : 4.766   Min.   : 3.473   Min.   : 3701
##  1st Qu.: 31387   1st Qu.: 6.522   1st Qu.: 5.590   1st Qu.: 8644
##  Median : 44788   Median : 8.023   Median : 7.196   Median :21418
##  Mean   : 97477   Mean   : 8.477   Mean   : 7.368   Mean   :24653
##  3rd Qu.: 85856   3rd Qu.:10.392   3rd Qu.: 8.599   3rd Qu.:33562
##  Max.   :939598   Max.   :13.038   Max.   :13.099   Max.   :75950
##      GEOGRAPHICAL.ZONE
##  North Central:7
##  North East   :6
##  North West   :7
##  South East   :5
##  South South  :6
##  South West   :6
```

- The state(s) which has the lowest number of confirmed cases has a record of 5 people, while the state which has the highest number of confirmed cases has a record of 78,163
- The state(s) which has the lowest number of recovered patients has a record of 3 people, while the state which has the highest number of recovered patients has a record of 76,920.
- The state(s) which has the lowest number of deaths due to covid has a record of 2 people, while the state which has the highest number of deaths due to covid has a record of 753.
- The state(s) with the lowest number of active cases has a record of 0 people, while the state with the highest number of active cases has a record of 1483 people.
- The state(s) with the lowest number of screened patients has a record of 11,759 , while the state the highest number of screened patients has a record of 939,598
- The following columns has a right skewed distribution: CONFIRMED, RECOVERED,DEATHS,ACTIVE CASES and SCREENED.
- 7 states belong to north central, 6 states belong to north east, 5 states belong to south east, 6 states belong to south south, and 6 states belong to south west

PART 2: PLOTLY PACKAGE

Introduction to package

Plotly is a package in R that makes quality web graphs that are interactive. The advantage of using plotly over ggplot2 is because of it is interactive. It provides functions for creating different types of charts like bar chart, scatter plot, box plot, line chart , different types of maps and so on.

Package installation and usage

To install the plotly package you can make use of this code "install.packages("plotly")" and to make use of the package when it is already installed, you will you the library function to call it. that is: "library(plotly)".

Package argument

Lets begin by knowing some of the arguments for creating charts using plotly.

plot_ly(data = data.frame(), …, type = NULL, width = NULL, height = NULL,)

- data is the data set you want to work with
- type allows you specify the type of chart you want to create.
- height is the height in pixels, it is optional and defaults to automatic sizing if it is not specified.
- width is the width in pixels, it is also optional and defaults to automatic sizing if it is not specified.

You might also want to look into other arguments of plotly which are: color, colors, alpha, stroke, strokes, alpha_stroke, size, sizes, span, spans, symbol,symbols,linetype, linetypes , split, frame, and source.

Note that we can also initialize a plotly object by using plot_geo() or plot_mapbox() instead of plotly(). To add a tittle to a chart using plotly, we can make use of the layout function, “%>% layout(title =”Text Title”)”

Practice

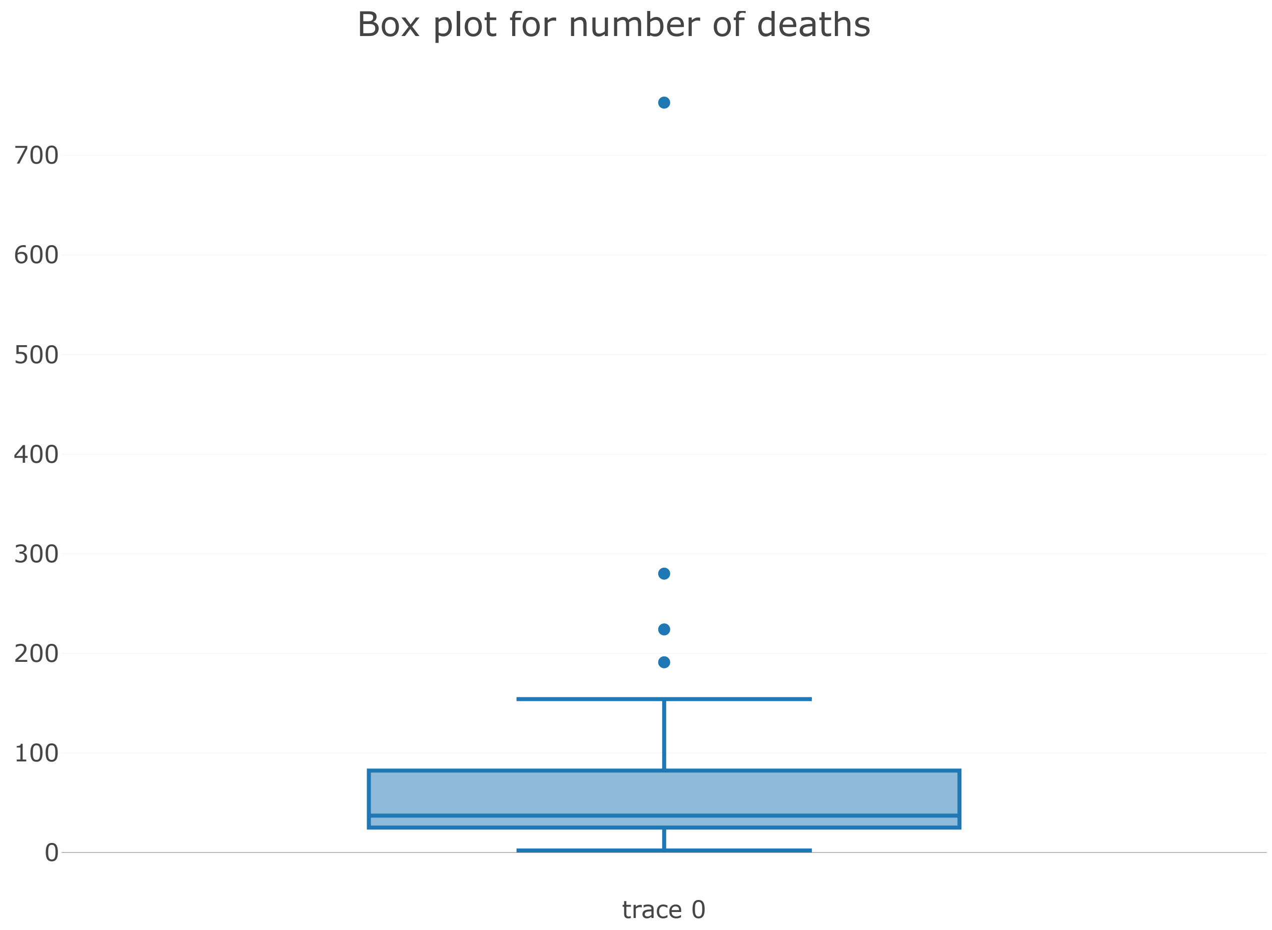
Let’s get into practice! . I will be using the covid_df I already loaded into R .

```
# Calling the library so I can make use of it.
library(plotly)
```

BOX PLOT WITH PLOTLY

I want to plot a box plot for the number of deaths

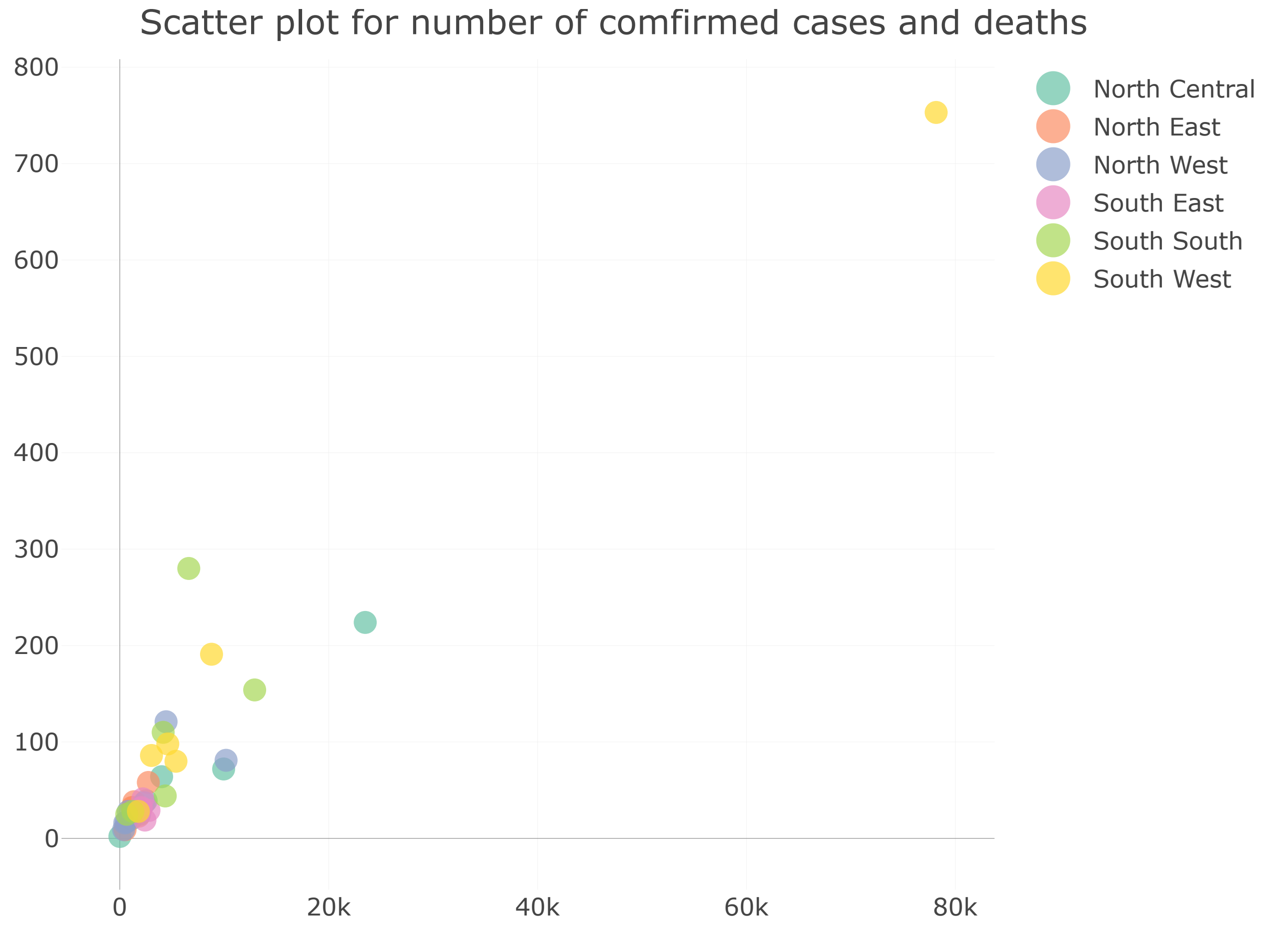
```
plot_ly(
  data = covid_df,# specify the data set
  y = covid_df$DEATHS, # y is the numeric column, y has to be numeric since its a box plot
  # type specifies that I want to create a box plot
  type = "box") %>% layout(title = "Box plot for number of deaths")
```



SCATTER PLOT WITH PLOTLY

I want to create a scatter for the number of of comfirmed cases and deaths

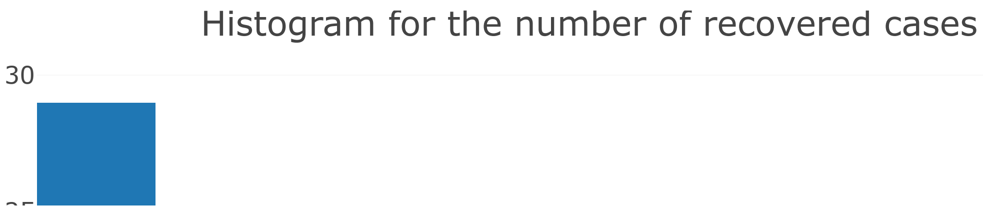
```
plot_ly(
  data = covid_df,# specify the data set
  x = covid_df$CONFIRMED, # x is the first numeric column
  y = covid_df$DEATHS, # y is the second numeric column, note that both x and y has to be num
eric.
  type = "scatter",  # This specifies that I want to create a scatter plot
  color = covid_df$GEOGRAPHICAL.ZONE, # including an additional argument, because I want the
 dots to be colored by the geographical zones
   # size specifies the size of dots I want for the scatter plot
  size = 3) %>% layout(title = "Scatter plot for number of comfirmed cases and deaths")
```

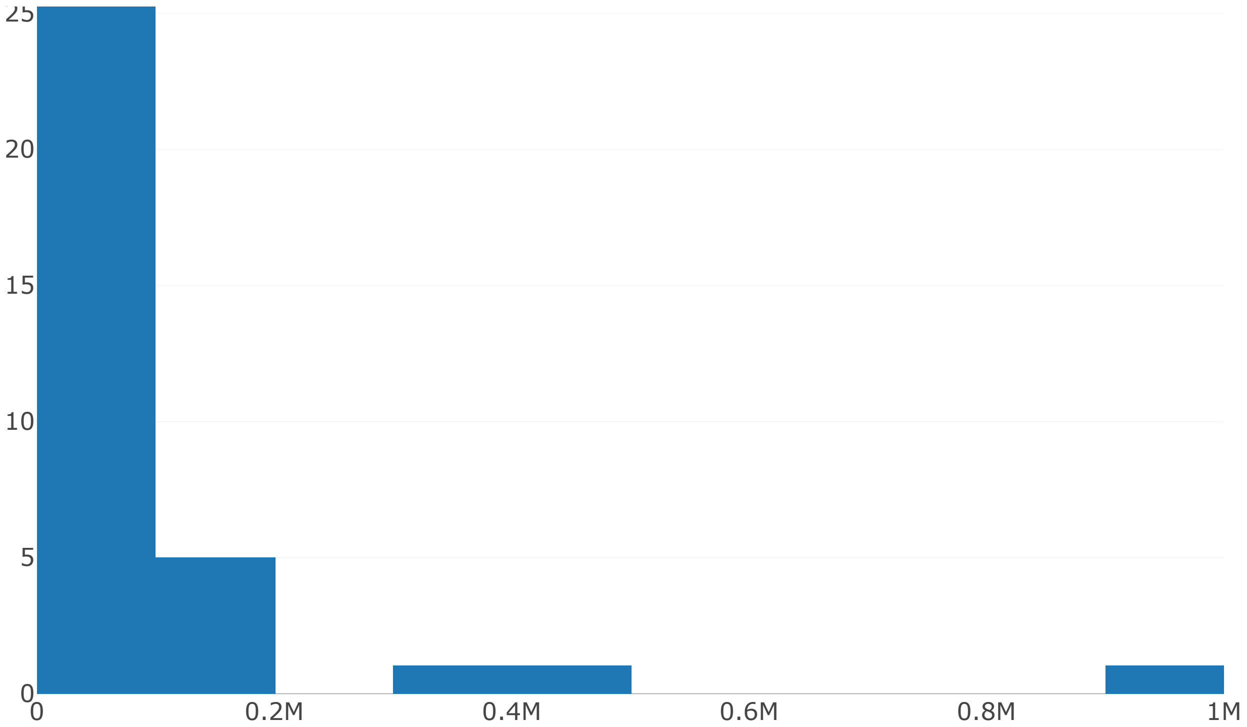


HISTOGRAM WITH PLOTLY

I want to create an histogram for the number of recovered cases

```
plot_ly(
  data = covid_df,# specify the data set
  x = covid_df$SCREENED, # x is the numeric column, x has to be numeric since its an histogra
m
   # type specifies that I want to create an histogram
  type = "histogram")  %>% layout(title = "Histogram for the number of recovered cases")
```





PART 3: FUNCTIONS/PROGRAMMING

S3 class I will be testing my function on the covid dataset but this time the states won't be specified as row names.

```
covid_df1 <- read.csv("R project dataset.csv",
                     header = TRUE) # first row contains column names
```

```
#Turn the object containing the covid data into an S3 class called covid
class(covid_df1) <- "covid"
```

Print function

I want to write a print function that prints out any data set assigned to the covid class as a data frame

```
print.covid <- function(df){ # defining a function called print.covid
  print(head(as.data.frame(do.call(cbind, df)))) # This combines all the columns in the data
 set as a data frame and print out the first 6 rows
}
print.covid(covid_df1) #testing if the function works
```

```
##       STATE CONFIRMED RECOVERED DEATHS ACTIVE.CASES SCREENED    Latitude
## 1      Abia      2030      1990     31            9    37748 5.453302119
## 2   Adamawa      1157      1098     32           27    29724 9.323227332
## 3 Akwa Ibom      4348      4076     44          228    46007 4.907245026
## 4   Anambra      2405      2386     19            0    49787 6.222775877
## 5    Bauchi      1802      1736     23           43    35826 10.79664717
## 6   Bayelsa      1250      1208     28           14    34322 4.766315328
##     Longitude AREA.SQUARE.METER GEOGRAPHICAL.ZONE
## 1 7.523189982       4858.882335        South East
## 2 12.40024078       37924.98786        North East
## 3 7.846394928       6723.202769       South South
## 4 6.932186089       4807.933352        South East
## 5 9.990588234       48496.40051        North East
## 6  6.08041884       9546.418182       South South
```

summary function

I want to write a summary function that returns back the total number for each of these columns: SCREENED, CONFIRMED,RECOVERED,DEATH, and ACTIVE CASES as a list.

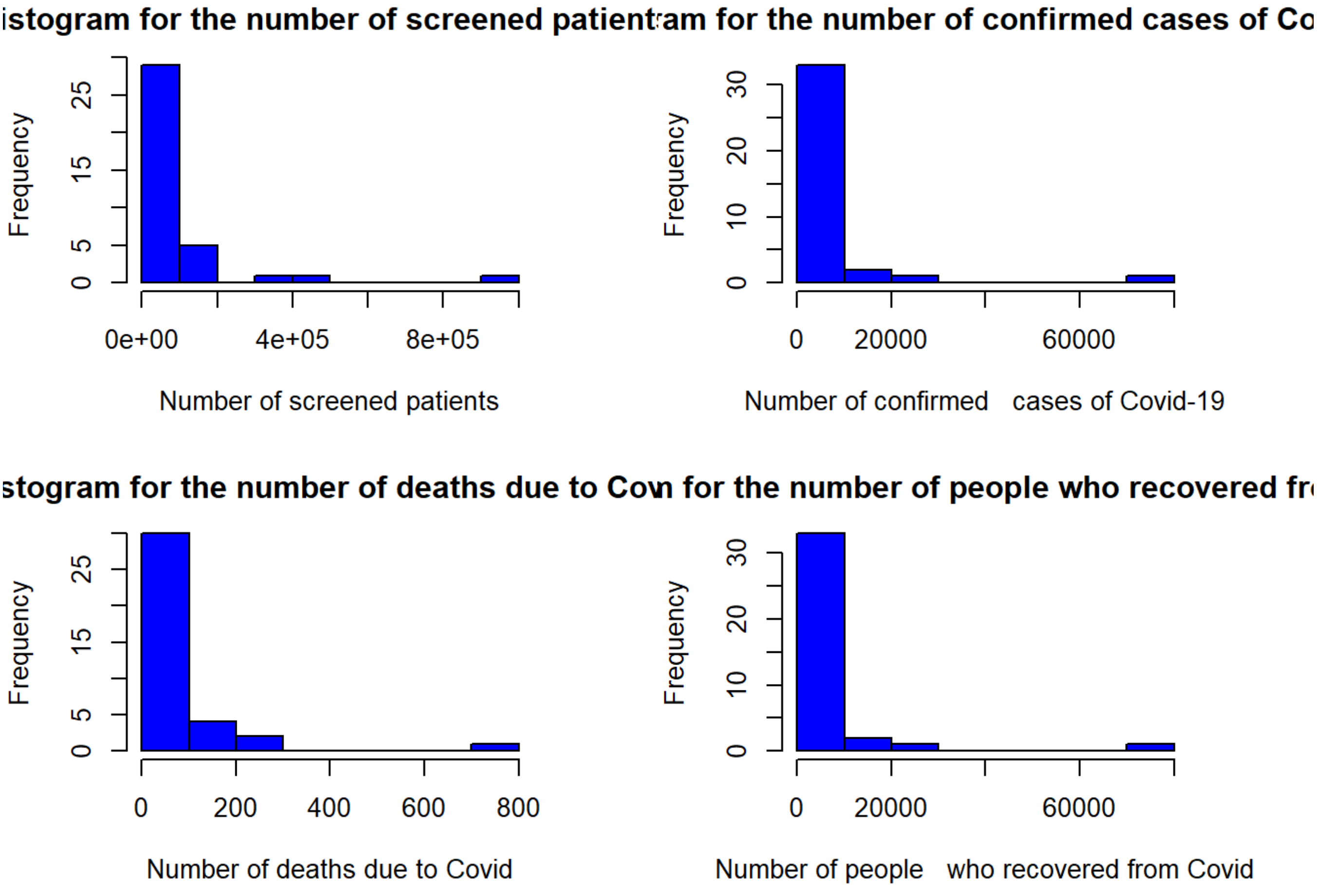
```
summary.covid<-function(df){
  s_1 = sum(df$SCREENED) # sum of the screened column
  s_2 = sum(df$CONFIRMED) # sum of the confirmed column
  s_3 = sum(df$RECOVERED) # sum of the recovered column
  s_4 = sum(df$DEATH) #sum of the death column
  s_5 = sum(df$ACTIVE.CASES) # sum of the active cases column

  list("Total number of screened patients in Nigeria" = s_1,
        "Total number of confirmed cases of Covid"= s_2,
       "Total number of patients who recovered from Covid" = s_3,
       "Total number of patients who died from Covid" = s_4,
       "Total number of active cases" = s_5
  ) # This combines all the sums as a list.
}
summary.covid(covid_df1) #testing if the function works
```

```
## $`Total number of screened patients in Nigeria`
## [1] 3606664
##
## $`Total number of confirmed cases of Covid`
## [1] 214092
##
## $`Total number of patients who recovered from Covid`
## [1] 207254
##
## $`Total number of patients who died from Covid`
## [1] 2976
##
## $`Total number of active cases`
## [1] 3862
```

Plot function

I want to write a plot function that returns back an histogram for the following columns: SCREENED, DEATHS,and RECOVERED.

```
plot.covid<-function(df){
  par(mfrow=c(2, 2), # Creating multiple plots
  mar=c(5, 4, 2.5, 4)) #leaves space for titles under plot
  hist(df$SCREENED, main = "Histogram for the number of screened patients ", xlab = "Number o
f screened patients",   col = "blue", border = "black")
  hist(df$CONFIRMED,main = "Histogram for the number of confirmed cases of Covid-19 ", xlab =
"Number of confirmed   cases of Covid-19",   col = "blue", border = "black")
  hist(df$DEATHS, main = "Histogram for the number of deaths due to Covid", xlab = "Number of
deaths due to Covid",   col = "blue", border = "black")
  hist(df$RECOVERED, main = "Histogram for the number of people who recovered from Covid", xl
ab = "Number of people   who recovered from Covid", col = "blue", border = "black")
 # The above code creates an histogram for all the columns mentioned above, main is giving ti
tle to each histogram,  xlab is for labeling the x-axis, col specified the color for the hist
ogram, border is used to specify the border    color between the bars.
}
plot.covid(covid_df1) #testing if the function works
```


istogram for the number of screened patient
am for the number of confirmed cases of Co
Frequency
0e+00
4e+05
8e+05
Number of screened patients
Frequency
Number of confirmed   cases of Covid-19
stogram for the number of deaths due to Co
n for the number of people who recovered fr
Frequency
Number of deaths due to Covid
Frequency
Number of people   who recovered from Covid