**Exercise Number – 1**

**Acquiring and Plotting Data**

**Q1. Load the in-build dataset from R and draw various basic plot in R using grid (Horizontal bar plot, Vertical bar plot, box plot, multiple box plot, plot with point and lines etc.,)**

**Aim:** To draw various basic plots in R like line plot, scatterplot, boxplot, multiple boxplots, horizontal and vertical bar plots, pie chart and histogram of the built-in dataset names Airquality and showing them in a 3x3 grid.

**Code:**

#20BDS0162 - SHOBHIT AGRAWAL

#Exercise 1

#Q1

#Loading the in-built dataset airquality

data = airquality

data

#Analyzing the data using various functions

str(data)

head(data)

tail(data)

summary(data)

#1.Using basic plot function to plot Solar.R vs Temp

plot(data$Solar.R,data$Temp,main = "20BDS0162")

#2.Using the barplot function

#Horizontal barplot

barplot(data$Temp, horiz = TRUE, main = "Horizontal Barplot 20BDS0162", col

="red", xlab = "Temperature", ylab = "Observations" )

#Vertical barplot

barplot(data$Temp, horiz = FALSE, main = "Vertical Barplot 20BDS0162", col ="green",

xlab = "Observations", ylab = "Temperature")

#3.Boxplot

#One box plot

boxplot(data$Ozone, main="Single Boxplot 20BDS0162", col = "blue", notch = TRUE,

outline = TRUE, border = "red")

#Multiple boxplot

boxplot(data$Ozone,data$Solar.R,data$Wind,data$Temp, notch = TRUE, col =

c("red","green","blue","orange"), main = "Multiple Boxplots 20BDS0162")

#4.Plot with both points and lines

plot(data$Wind,type = "b",main = "Points and Lines 20BDS0162", col = "red", bg = "blue",pch = 24)

#5.Histogram

hist(data$Wind, col = "yellow", border = "green", main = "Histogram 20BDS0162")

#6.Pie Chart

pie(data$Month,labels = data$Month, main = "Pie Chart 20BDS0162")

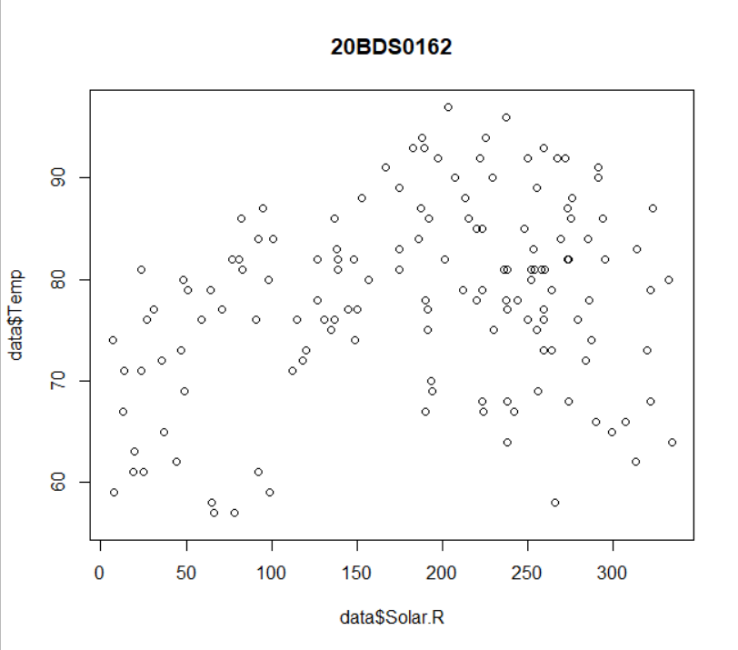
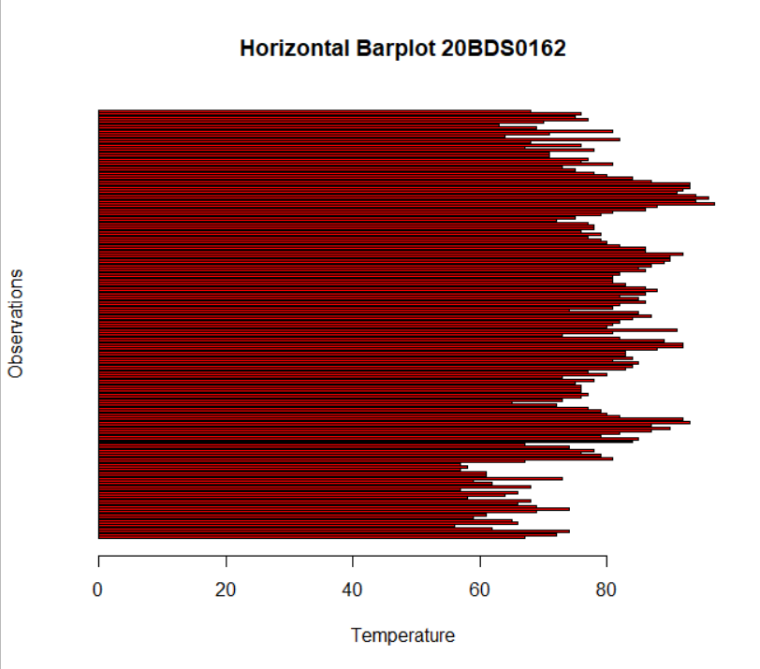
#7.Visualizing all the above plots in a 3x3 grid

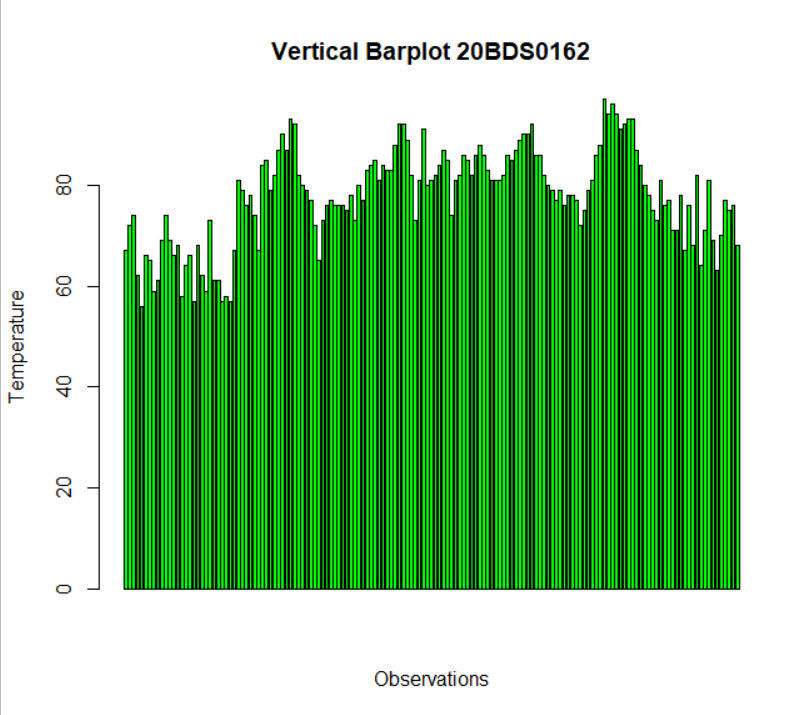
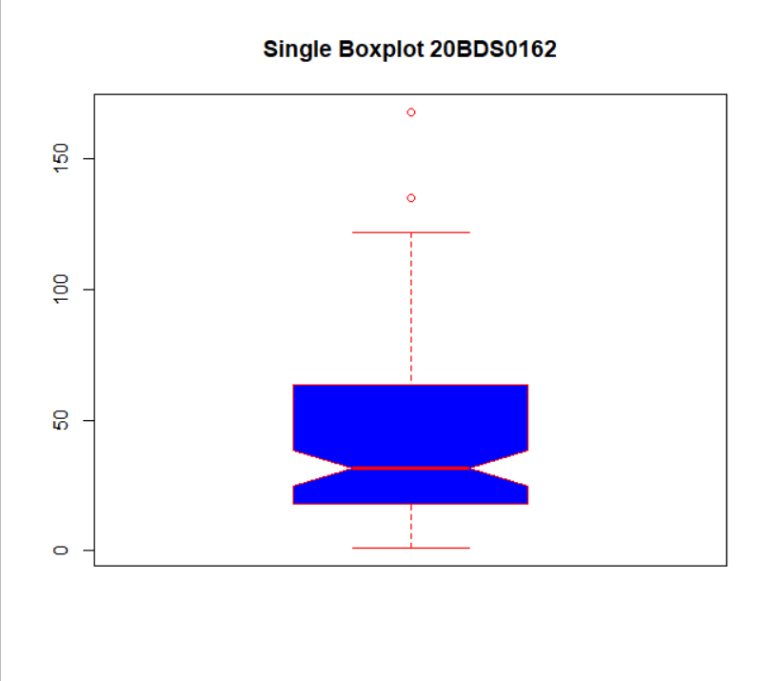
par(mfrow=c(3,3))

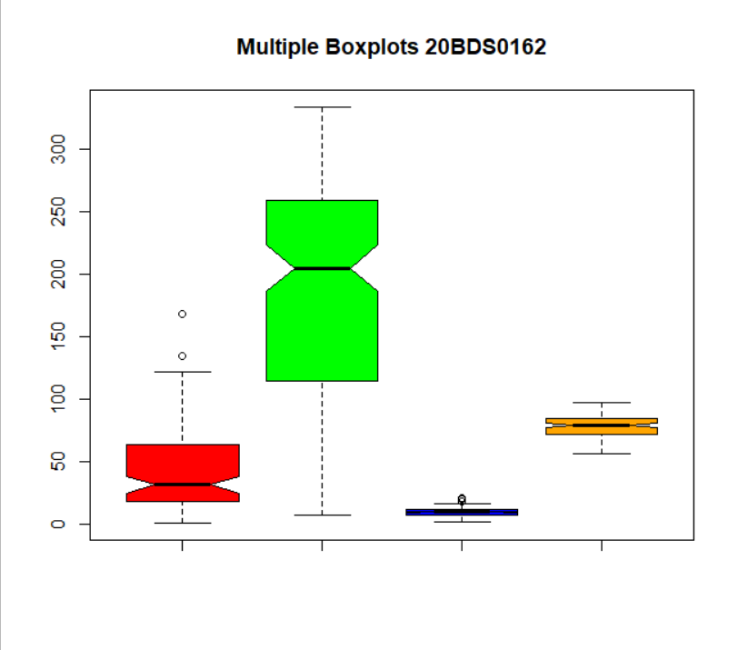
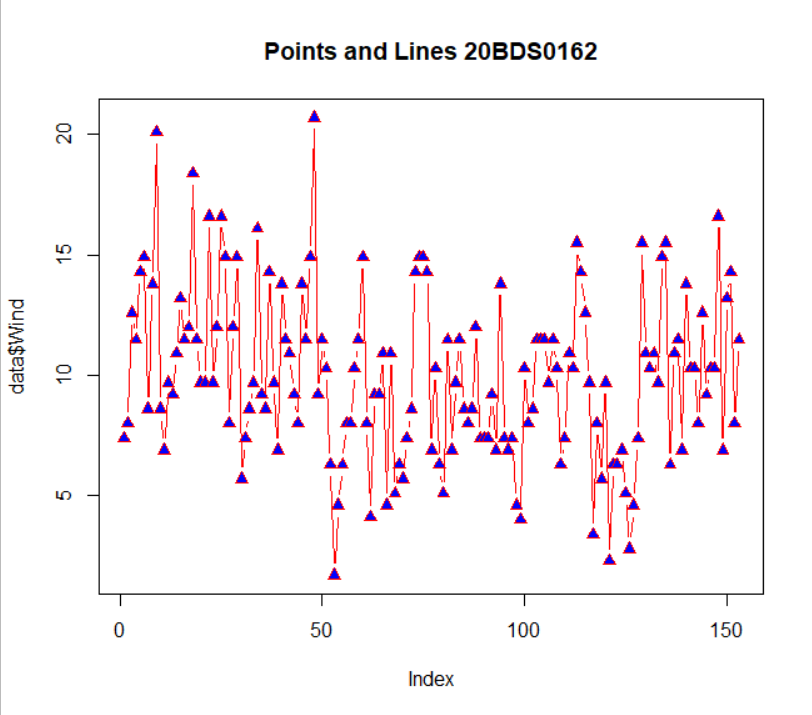
#8.Scatter Plot Matrix

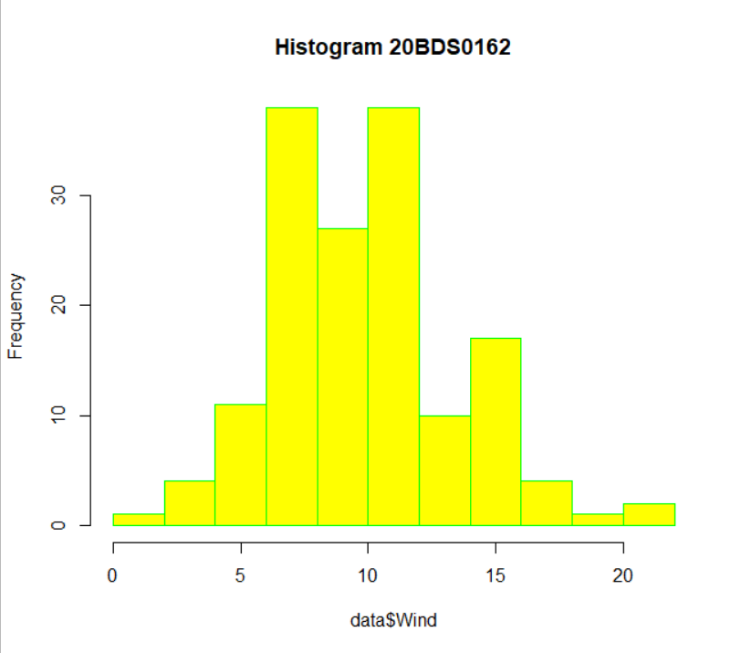
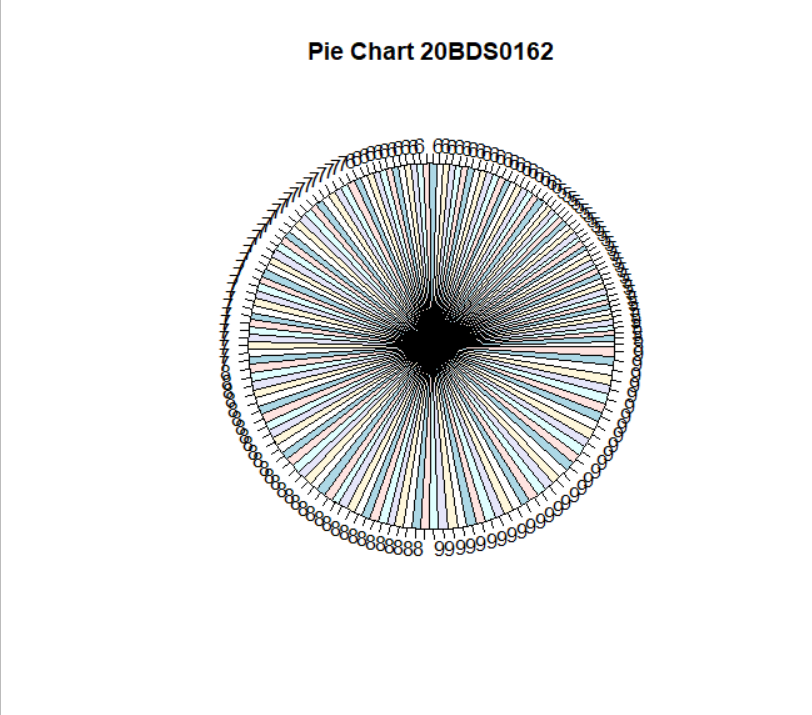
plot(data, main = "Scatter Plot Matrix 20BDS0162")

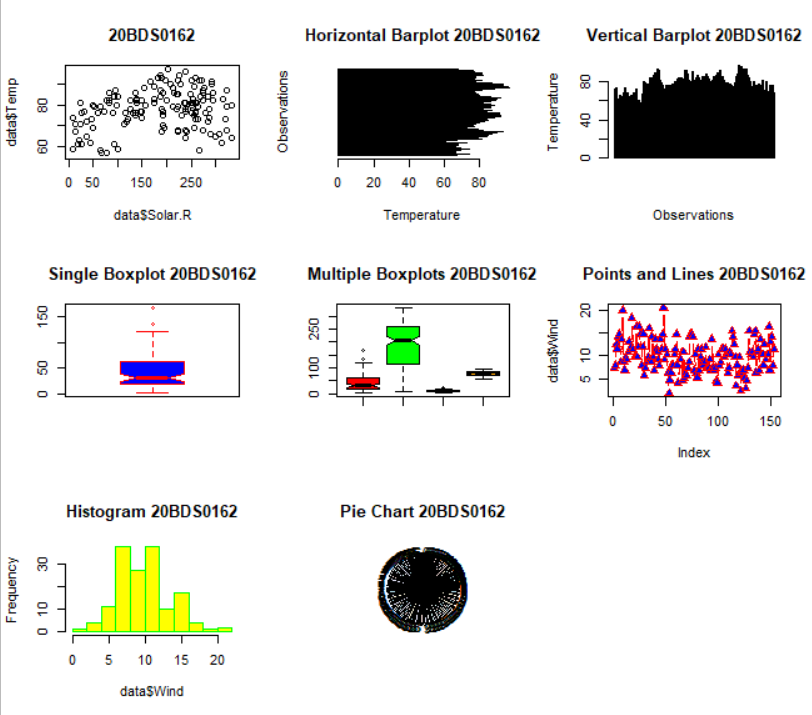
**Output:**

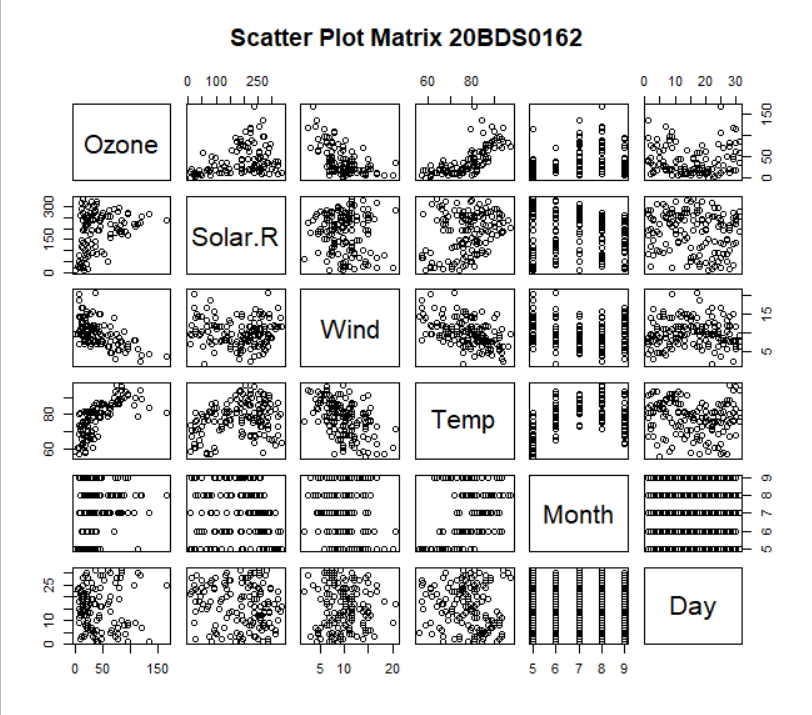












**Result:**

We were able to plot all the basic plots in R for a built-in dataset available in R language.

**Q2. Load in-built dataset mtcars and visualize data using visualization library ggplot.**

**Aim:** To use the visualization library ggplot2 and plot various plots using it on the built-in dataset mtcars in R.

**Code:**

#Q2

library(lattice)

library(ggplot2)

data = mtcars

data

#1.Scatter Plot

ggplot(data = data, aes(x = wt,y = mpg,color = as.factor(cyl)))+

geom\_point()+xlab("Weight")+ylab("Mileage")+ggtitle("WT vs MPG 20BDS0162")

#2.Line Plot

ggplot(data = data, aes(x = wt,y = mpg,color = as.factor(cyl)))+

geom\_line()+xlab("Weight")+ylab("Mileage")+ggtitle("WT vs MPG 20BDS0162")

#3.Plot with both line and point

ggplot(data = data, aes(x = wt,y = mpg,color = as.factor(cyl)))+geom\_point()+

geom\_line()+xlab("Weight")+ylab("Mileage")+ggtitle("WT vs MPG 20BDS0162")

#4.Plot with size attribute

ggplot(data = data, aes(x = wt,y = mpg,color = as.factor(cyl),size = qsec))+

geom\_point()+xlab("Weight")+ylab("Mileage")+ggtitle("WT vs MPG 20BDS0162")

#5.Plot with shape attribute

ggplot(data = data, aes(x = wt,y = mpg,shape = as.factor(cyl),color = as.factor(cyl),size = qsec))+

geom\_point()+xlab("Weight")+ylab("Mileage")+ggtitle("WT vs MPG 20BDS0162")

#6.Histogram

ggplot(data = data, aes(x = wt))+geom\_histogram(color="blue",fill="yellow",bins=7)+ggtitle("WT vs MPG 20BDS0162")+xlab("Weight")+ylab("Count")

#7.Boxplot

data$gear = as.factor(data$gear)

ggplot(data = data, aes(x = gear,y = disp))+geom\_boxplot(color = "red", fill = "yellow")+ggtitle("WT vs MPG 20BDS0162")+xlab("Weight")

data

#8.Column Chart

ggplot(data = data,

aes(x = gear,y = mpg,color = gear))+

geom\_col() + xlab("Gear") + ylab("Mpg")+

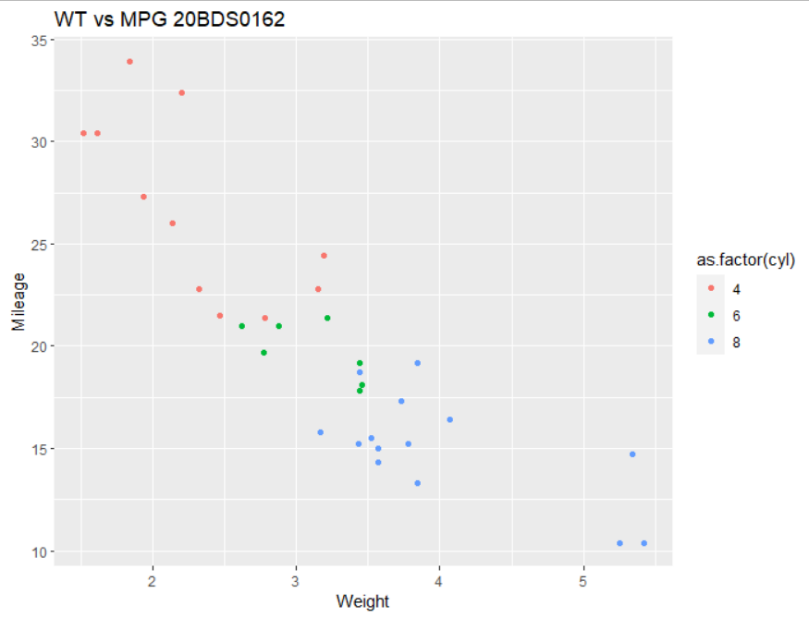
ggtitle("Gear vs Mpg 20BDS0162")

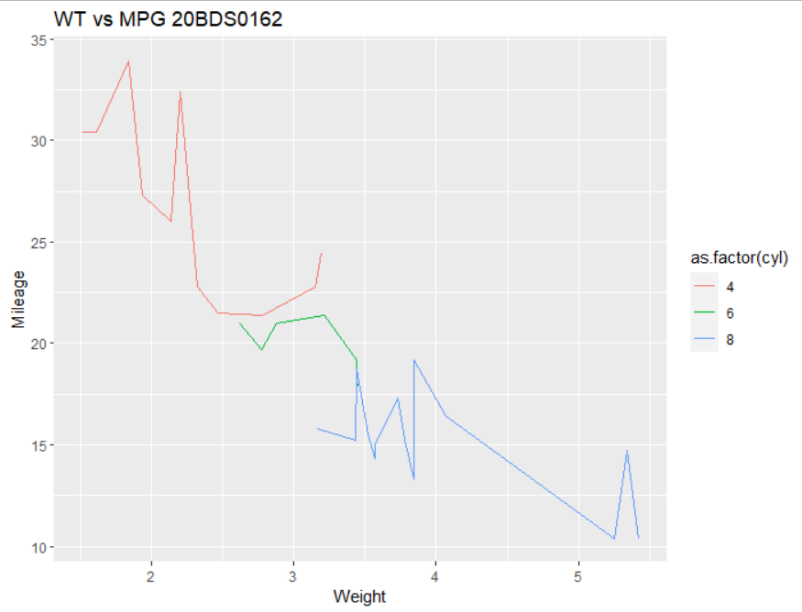
#9.Line Plot with weight

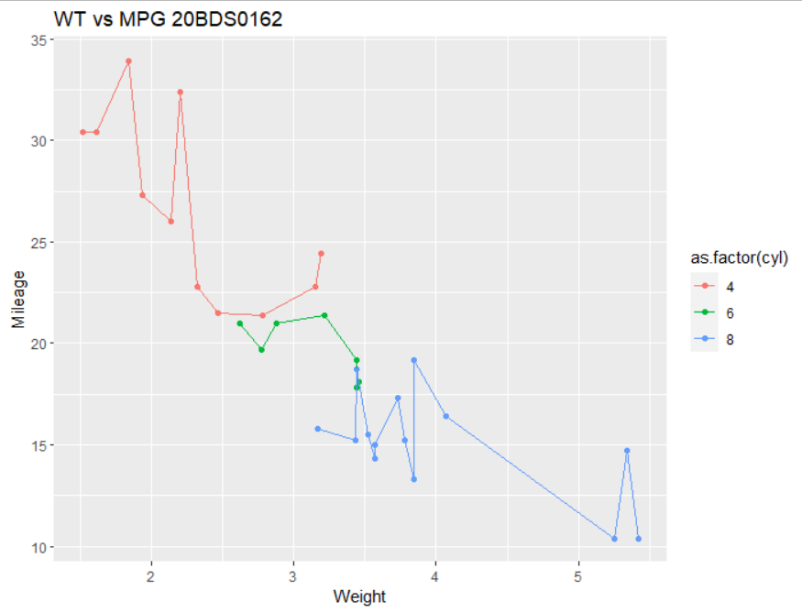
ggplot(data = data, aes(x = wt,y = mpg,color = as.factor(cyl), size = qsec))+

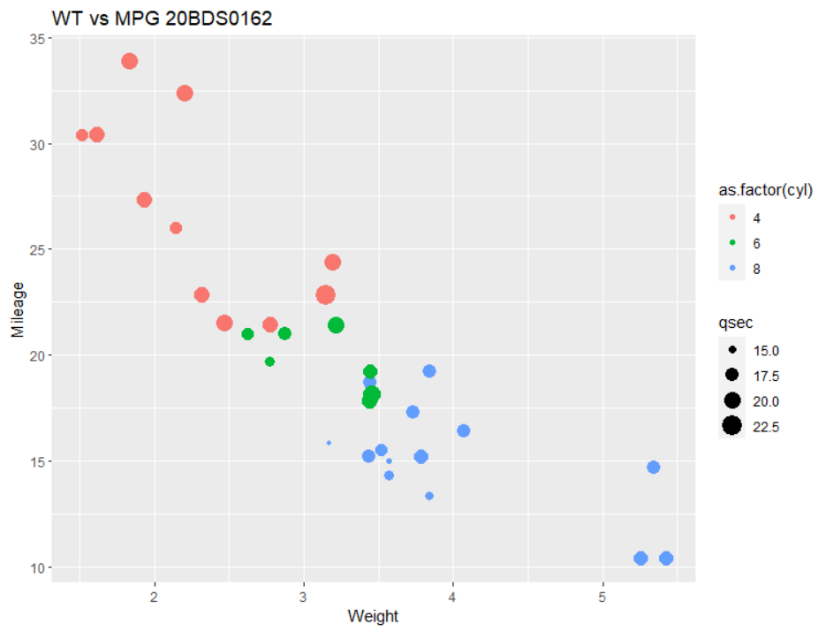
geom\_line()+xlab("Weight")+ylab("Mileage")+ggtitle("WT vs MPG 20BDS0162")

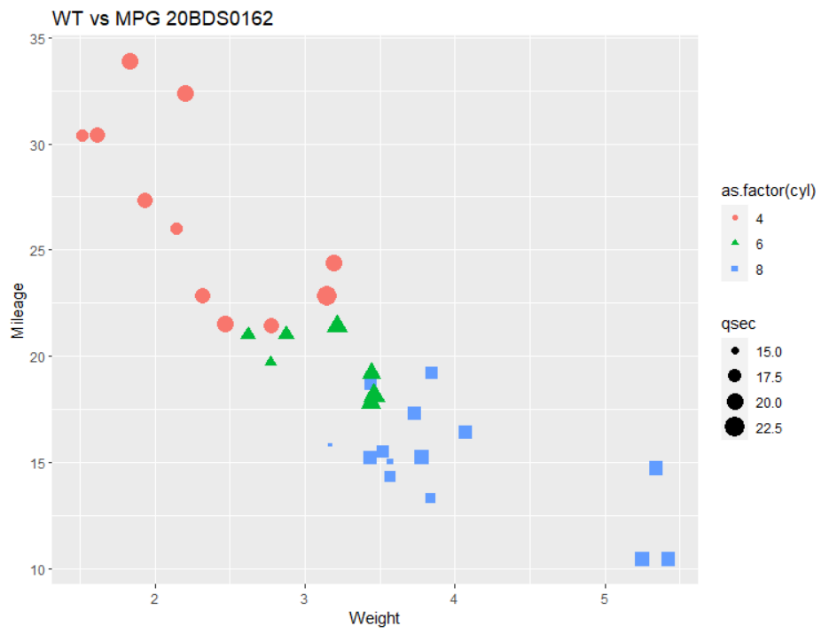
**Output:**

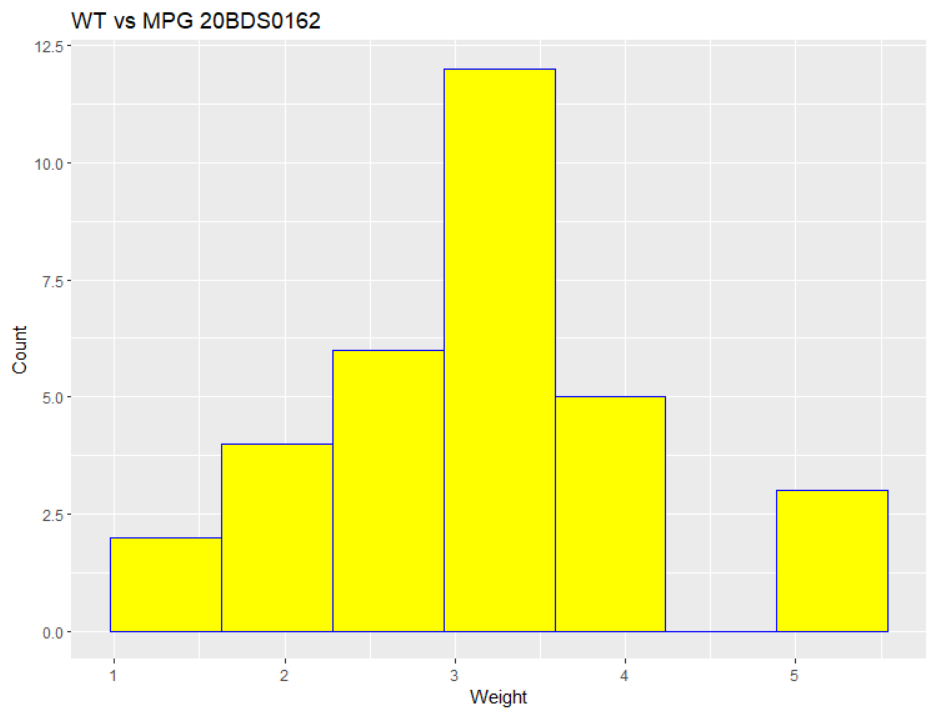


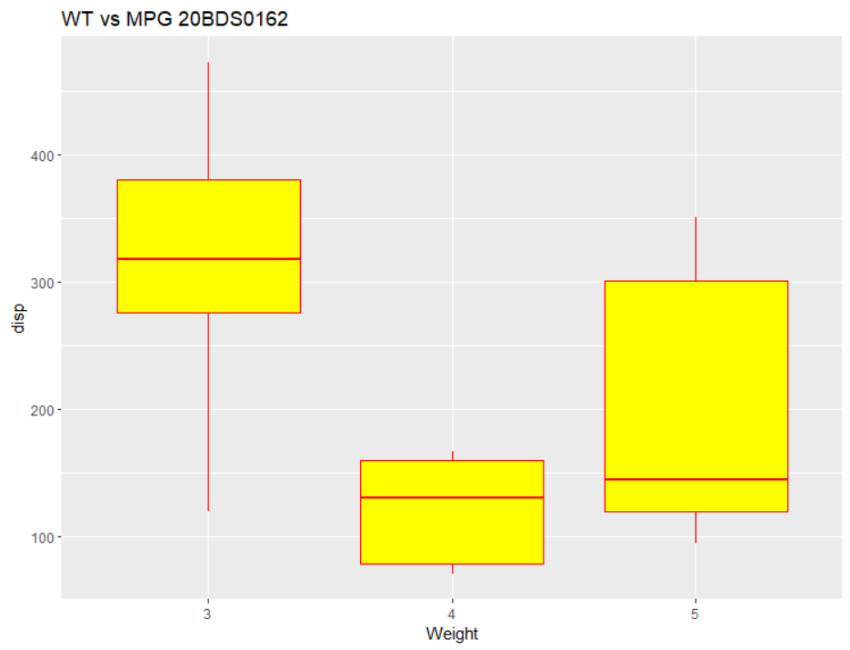


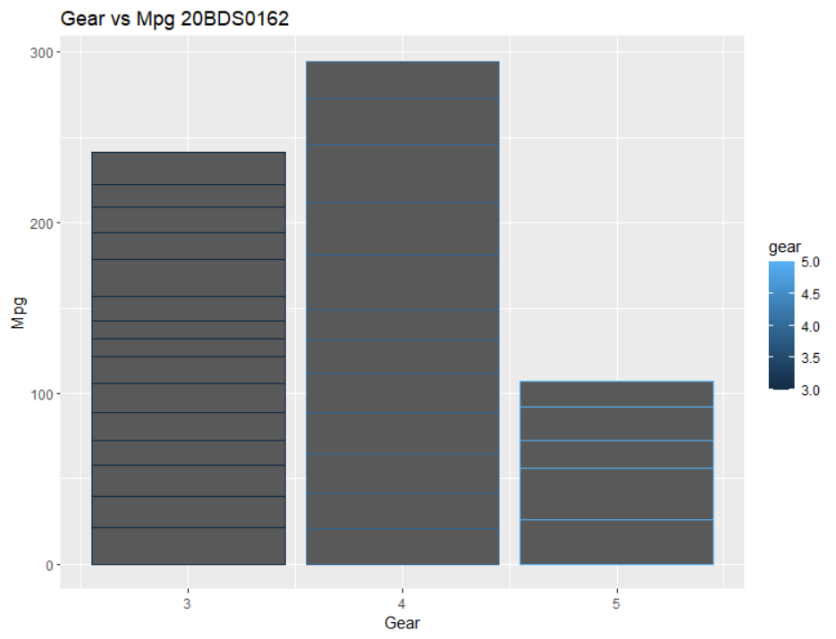


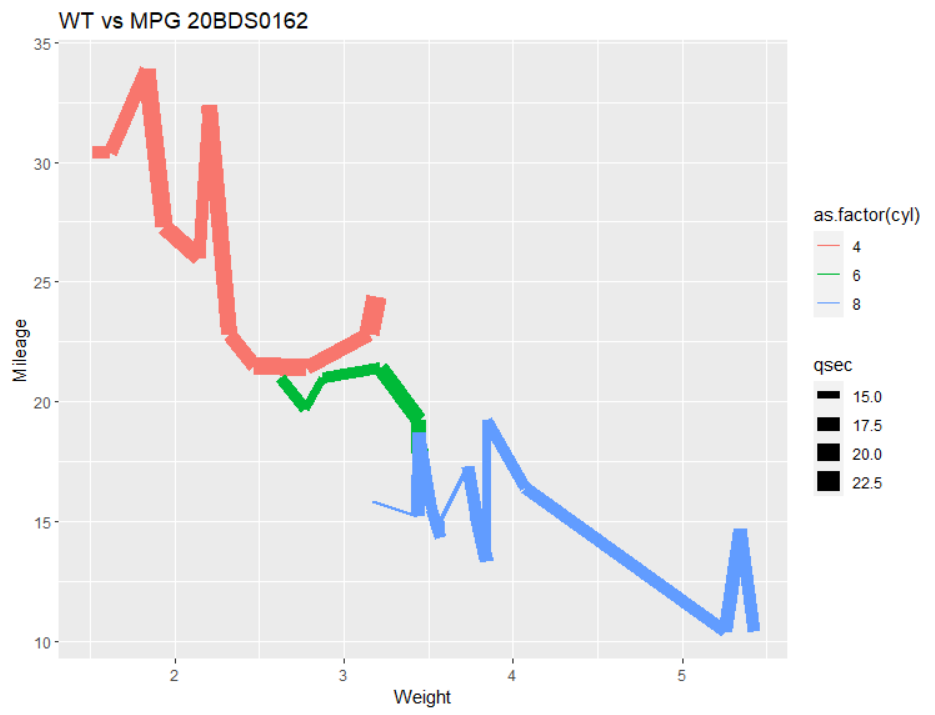












**Result:** We were able to successfully plot various plots for the mtcars dataset in R using the ggplot2 library.

**Exercise Number - 2**

**Statistical Analysis – Univariate, Bivariate, Multivariate – plotting and coloring for maps**

**Q1. Load the gapminder dataset and perform statistical analysis using tidyverse and dplyr libraries.**

**Aim:** To use the gapminder dataset in R and use various function of dplyr and tidyr libraries like filter, sort, arrange, etc… and perform statistical analysis.

**Code:**

#20BDS0162 - SHOBHIT AGRAWAL

#Exercise 2

#Q1

#Loading the libraries

library(dplyr)

library(tidyr)

library(gapminder)

library(ggplot2)

View(gapminder)

str(gapminder)

glimpse(gapminder) #dplyr for glimpse

#1.Extract continent 'Asia'

filter(gapminder,continent == "Asia")

gapminder %>% filter(continent == "Asia")

#2.Extract year 1957

gapminder %>% filter(year == "1957")

#3.Extract year as 2002 and country china

gapminder %>% filter(year == "2002" & country == "China")

gapminder %>% filter(year == "2002", country == "China")

#4.Sort lifeExp in desc order

sort(gapminder$lifeExp,decreasing = TRUE)

arrange(gapminder,desc(lifeExp))

#5.1957 and pop in desc order

arrange(filter(gapminder,year == "1957"),desc(pop))

#6.a.lifeExp in months

data = gapminder

data

#updating the same column

data$lifeExp = data$lifeExp\*12

#6.b.adding a new column

data = mutate(data,lifeExpMonths = lifeExp\*12)

data

#gapminder\_1952

gapminder\_1952 = gapminder %>% filter(year == "1952")

gapminder\_1952

View(gapminder\_1952)

#7.Visualize scatter plot for pop and gdpPercap for gapminder\_1952

ggplot(data = gapminder\_1952,

aes(x = pop,y = gdpPercap,color = as.factor(continent))) +

geom\_point() + scale\_x\_log10() + scale\_y\_log10() + xlab("Pop") + ylab("GDP Per Capita") +

ggtitle("Pop vs GDP Per Capita 20BDS0162")

#8.Scatter plot pop and lifeExp group by continent based on population size

ggplot(data = gapminder\_1952,

aes(x = pop,y = lifeExp,color = as.factor(continent),size = gdpPercap))+

geom\_point() + scale\_x\_log10() + xlab("Pop") + ylab("Life Exp") +

ggtitle("Pop vs Life Exp 20BDS0162")

#9.Sub-graph

ggplot(data = gapminder\_1952,

aes(x = pop,y = lifeExp, color = continent))+

geom\_point() + scale\_x\_log10() + facet\_wrap(~continent)+ xlab("Pop") + ylab("Life Exp") +

ggtitle("Pop vs Life Exp 20BDS0162")

#10.Sub-graph for year-DS:gapminder

ggplot(data = gapminder,

aes(x = pop,y = lifeExp, color = continent))+

geom\_point() + scale\_x\_log10() + facet\_wrap(~year) + xlab("Pop") + ylab("Life Exp") +

ggtitle("Pop vs Life Exp 20BDS0162")

#11.Summarize - median lifeExp, DS:gapminder

gapminder%>%summarize(MedianLifeExp = median(lifeExp))

#12.1957, median - lifeExp, max - gdpPercap

gapminder %>% filter(year == 1957) %>%

summarize(MedianLifeExp = median(lifeExp),MaxgdpPercap = max(gdpPercap))

#13.group by year, median lifeExp

#store in object by\_year

data = gapminder

by\_year <- data %>% group\_by(year)%>% summarize(MedianLifeExp = median(lifeExp))

by\_year

#14.Visualize year vs MedianLifeExp:by\_year

ggplot(data = by\_year,

aes(x = year,y = MedianLifeExp))+

geom\_point() + expand\_limits(y = 0) + xlab("Year") + ylab("Median Life Exp")+

ggtitle("Year vs Median Life Exp 20BDS0162")

#15.summarize the median gdpPercap by year and continent and save it in

#by\_year\_continent

data = gapminder

by\_year\_continent <- data %>%

group\_by(year,continent)%>%

summarize(MedianGdpPercap = median(gdpPercap))

by\_year\_continent

#16.Visualise year vs MedianGdpPercap

#16.1.Line plot

ggplot(data = by\_year\_continent,

aes(x = year,y = MedianGdpPercap,color = continent))+

geom\_line() + geom\_point() + xlab("Year") + ylab("Median GDP Per Capita")+

ggtitle("Year vs Median GDP Per Capita 20BDS0162")

#16.2.Boxplot

ggplot(data = by\_year\_continent,

aes(x = year,y = MedianGdpPercap,color = continent))+

geom\_boxplot() + xlab("Year") + ylab("Median GDP Per Capita")+

ggtitle("Year vs Median GDP Per Capita 20BDS0162")

#16.3.Column chart

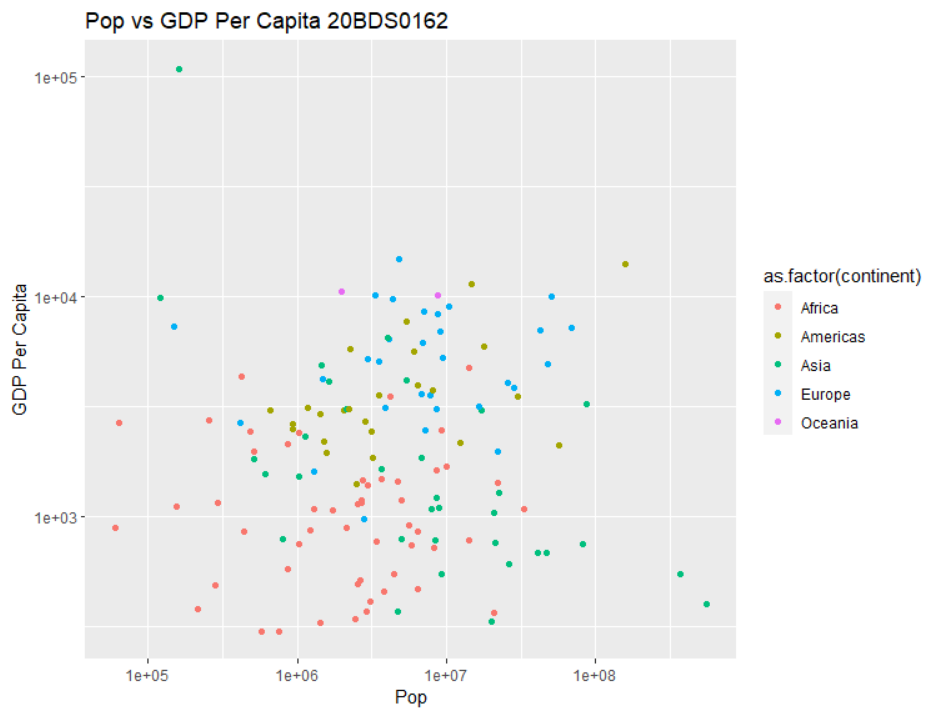
ggplot(data = by\_year\_continent,

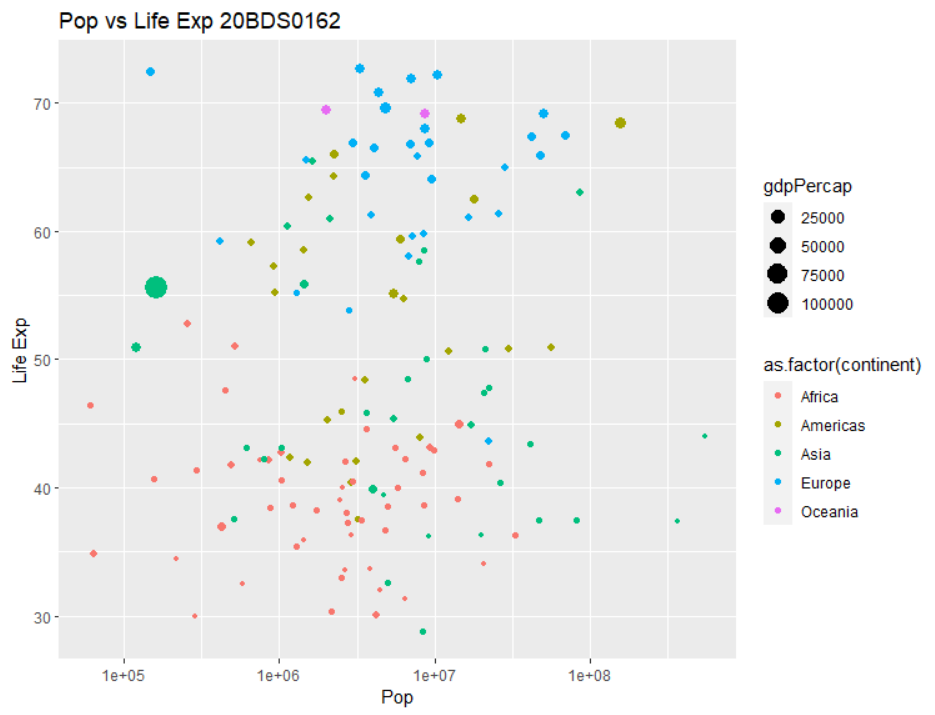
aes(x = year,y = MedianGdpPercap,color = continent))+

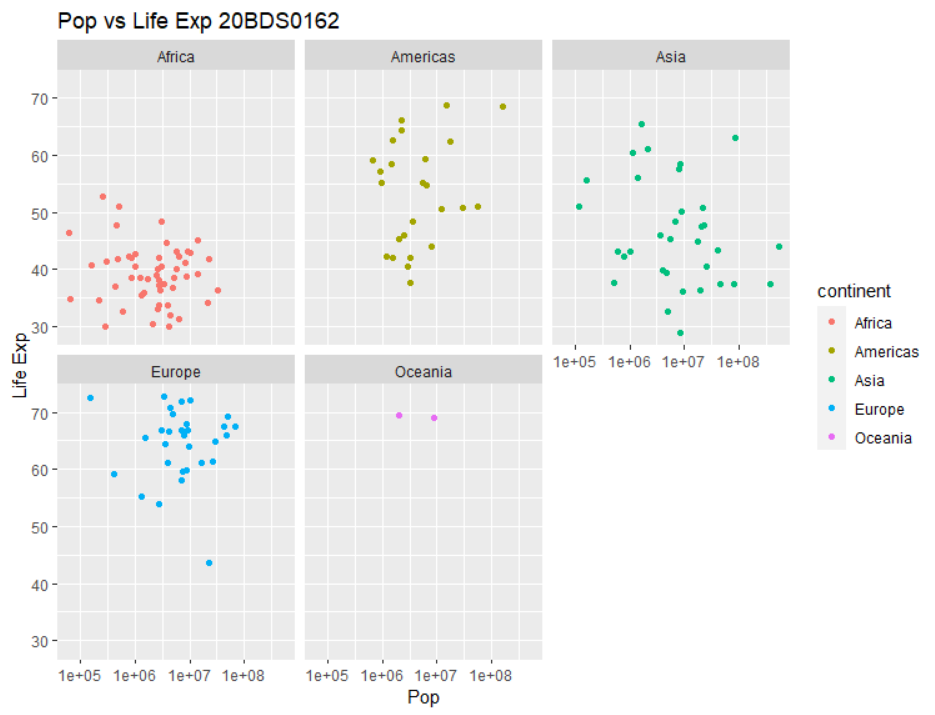
geom\_col() + xlab("Year") + ylab("Median GDP Per Capita")+

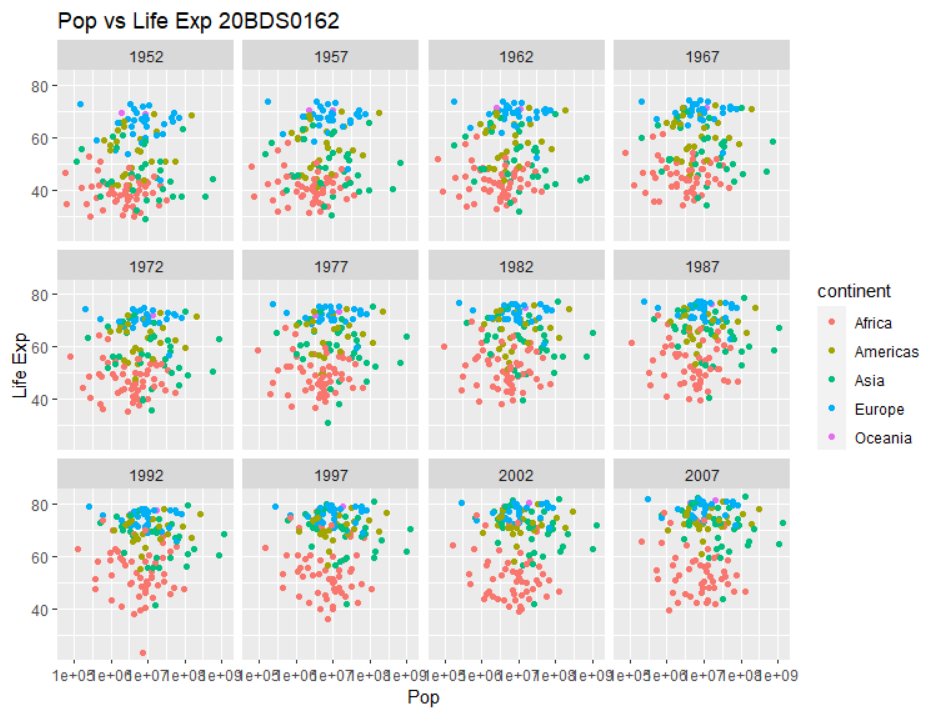
ggtitle("Year vs Median GDP Per Capita 20BDS0162")

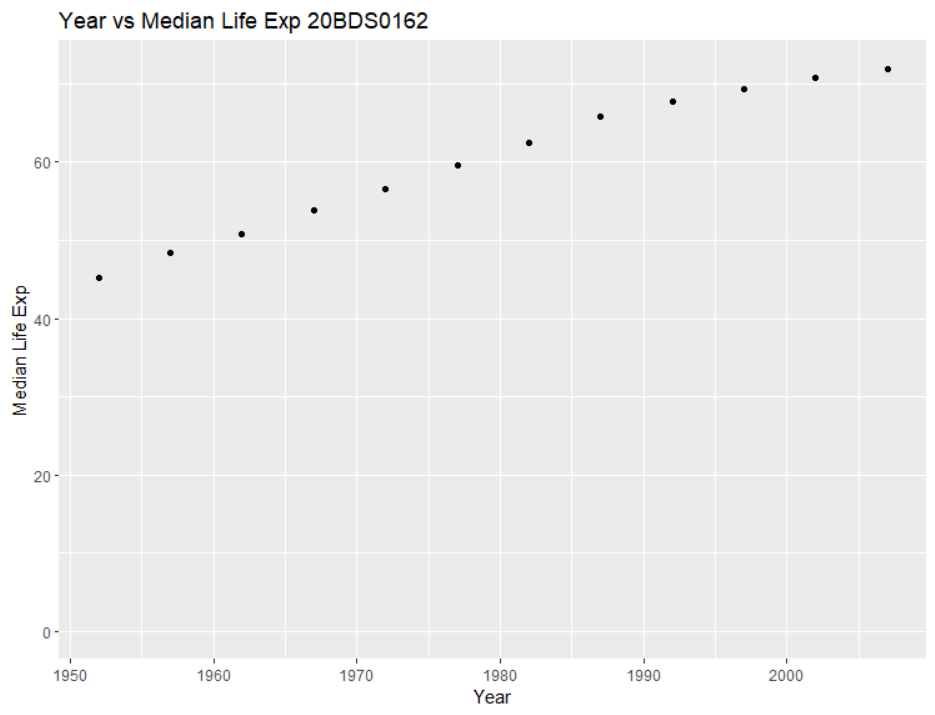
**Output:**

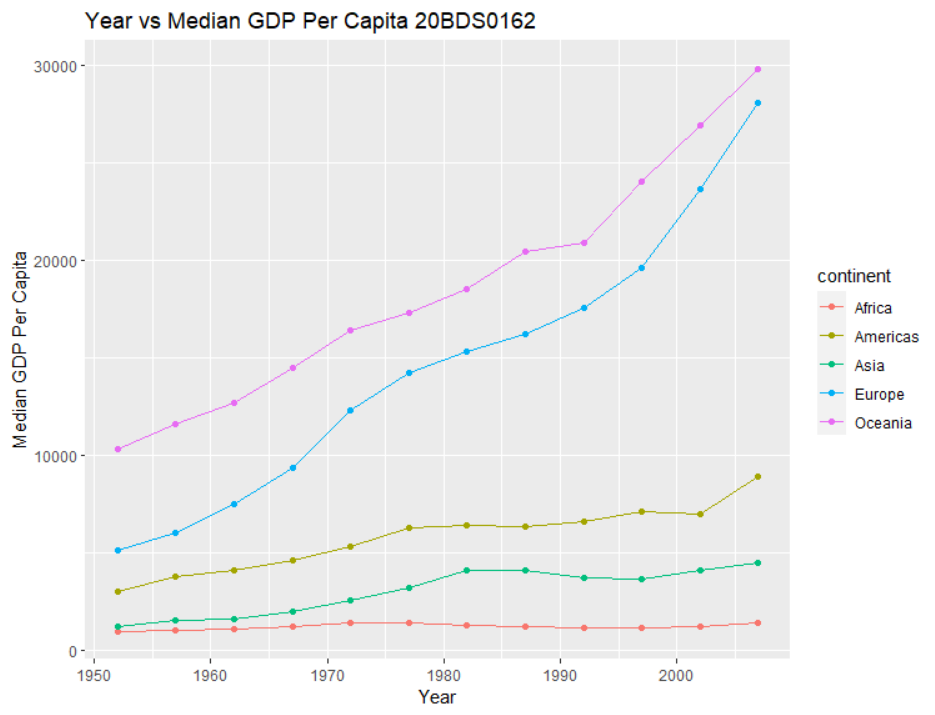


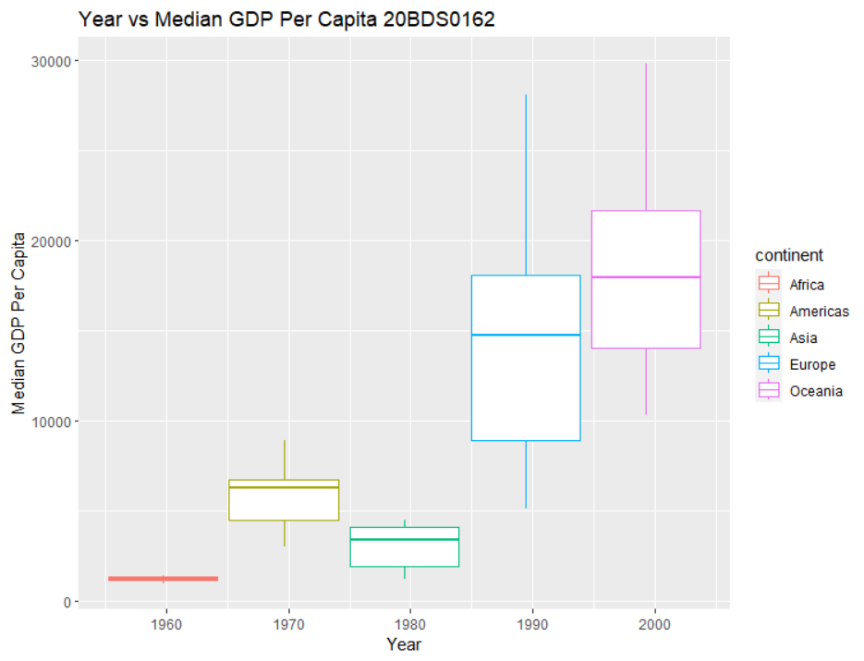


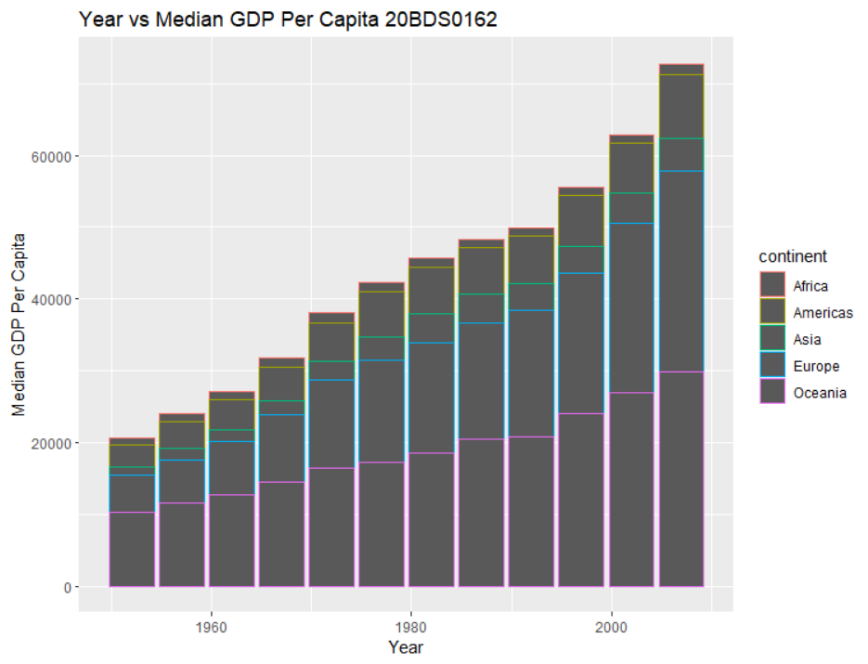












**Result:** We were able to successfully perform the statistical analysis of gapminder dataset using the in-built functions of the dplyr and tidyr libraries.

**Q2. Using RColorBrewer visualize mpg data.**

**Aim:** To visualize the mpg data in R using the RColorBrewer library

**Code:**

#Q2

#Color Visualization using RColorBrewer

library(RColorBrewer)

library(viridis)

library(ggplot2)

library(gridExtra)

str(mpg)

factor(mpg$cyl)

display.brewer.all()

display.brewer.all(colorblindFriendly=T)

#mpg dataset

View(mpg)

#1.Density Plot

ggplot(data = mpg,aes(x = cty))+geom\_density(aes(fill = factor(cyl))) + ggtitle("Density Plot 20BDS0162")

#2.Transparency

ggplot(data = mpg,aes(x = cty))+geom\_density(aes(fill = factor(cyl),alpha=0.7)) + ggtitle("Density Plot (Transparency) 20BDS0162")

#3.Labels

ggplot(data = mpg,aes(x = cty))+geom\_density(aes(fill = factor(cyl),alpha=0.7))+labs(title = "Density Plot 20BDS0162",x = "City Mileage",fill = "#Cylinder")

#4.Brewer

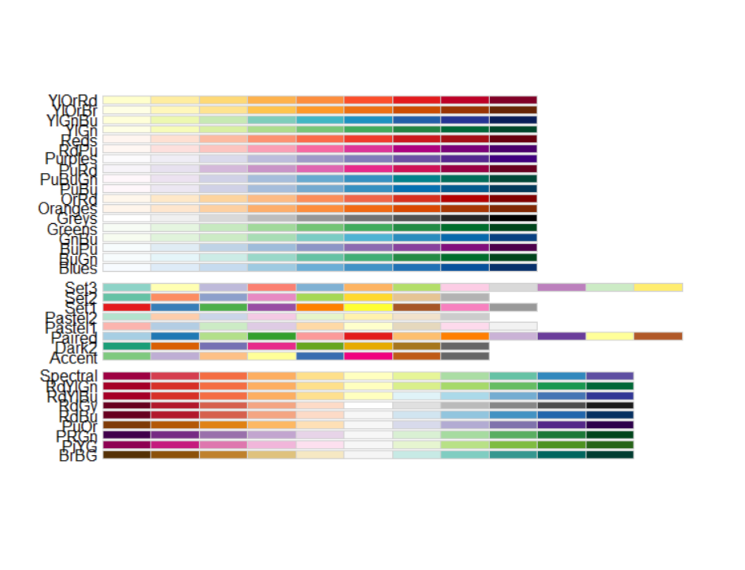
p1<-ggplot(data = mpg,aes(x = cty))+geom\_density(aes(fill = factor(cyl),alpha=0.7))+labs(title = "D1 20BDS0162",x = "City Mileage",fill = "#Cylinder")+scale\_fill\_brewer(palette = "YlOrRd")

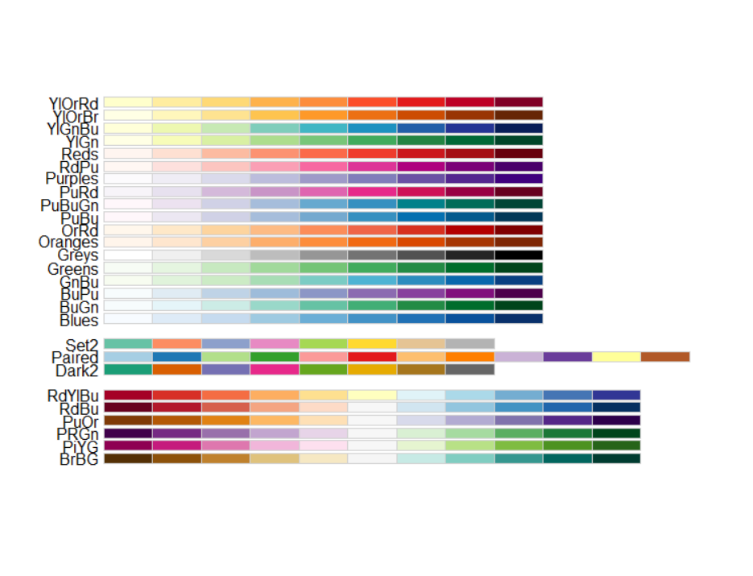
p2<-ggplot(data = mpg,aes(x = cty))+geom\_density(aes(fill = factor(cyl),alpha=0.7))+labs(title = "D2 20BDS0162",x = "City Mileage",fill = "#Cylinder")+scale\_fill\_brewer(palette = "Set2")

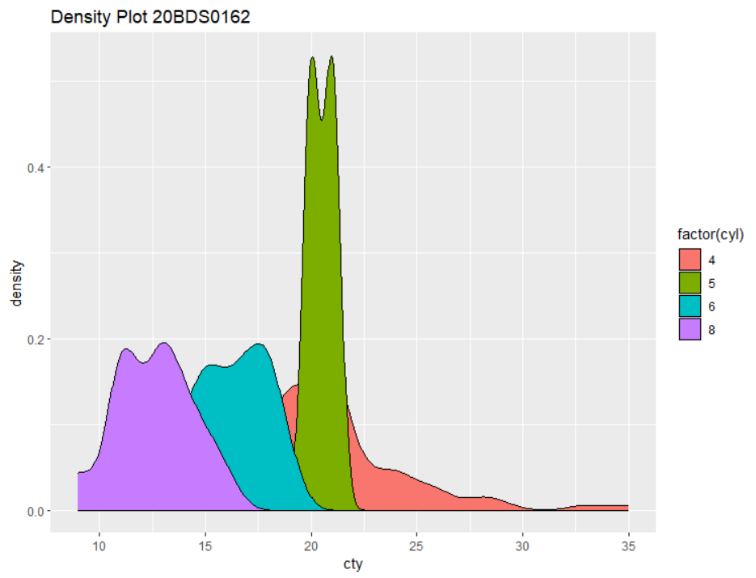
p3<-ggplot(data = mpg,aes(x = cty))+geom\_density(aes(fill = factor(cyl),alpha=0.7))+labs(title = "D3 20BDS0162",x = "City Mileage",fill = "#Cylinder")+scale\_fill\_brewer(palette = "BrBG")

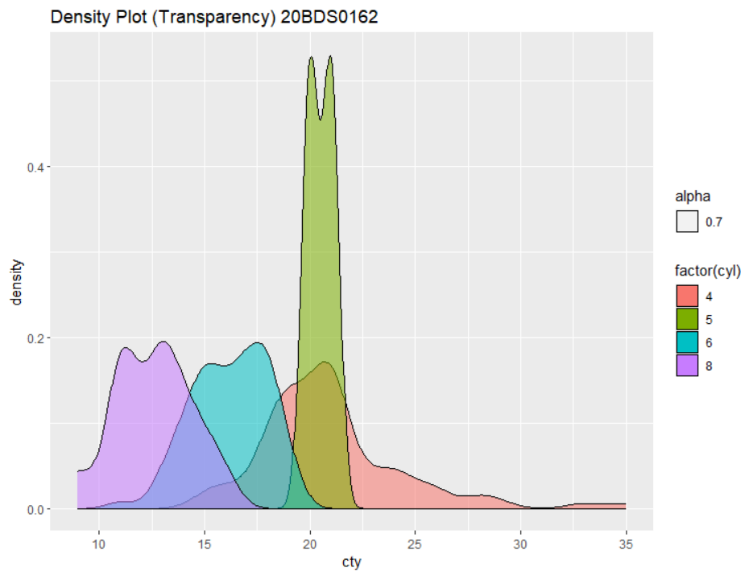
p4<-ggplot(data = mpg,aes(x = cty))+geom\_density(aes(fill = factor(cyl),alpha=0.7))+labs(title = "D4 20BDS0162",x = "City Mileage",fill = "#Cylinder")+scale\_fill\_brewer(palette = "Spectral")

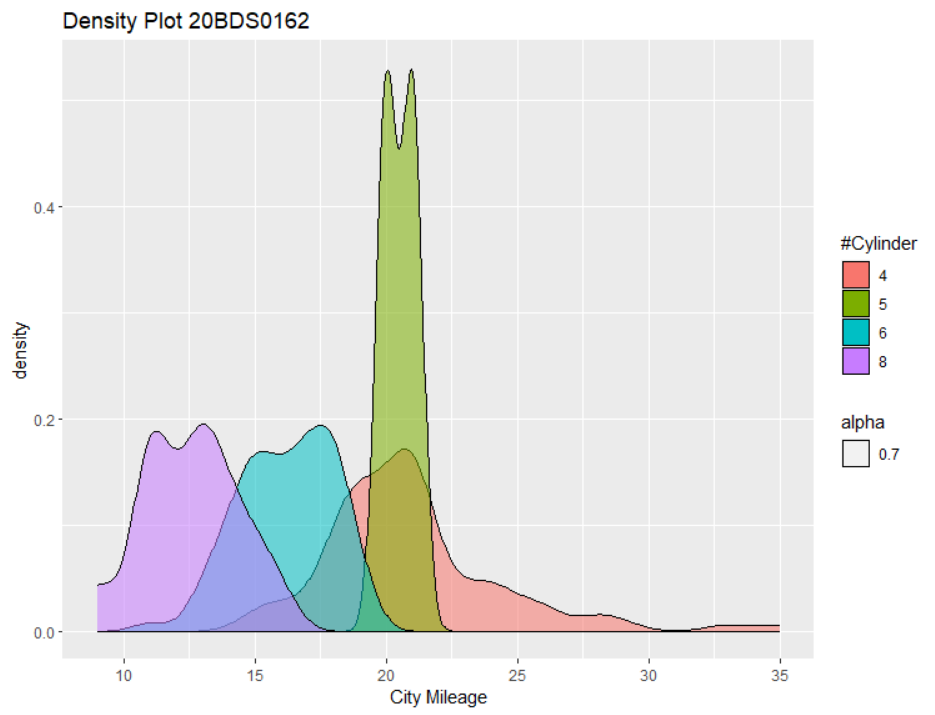
grid.arrange(p1,p2,p3,p4,nrow = 2,top = "Grid Graphs 20BDS0162")

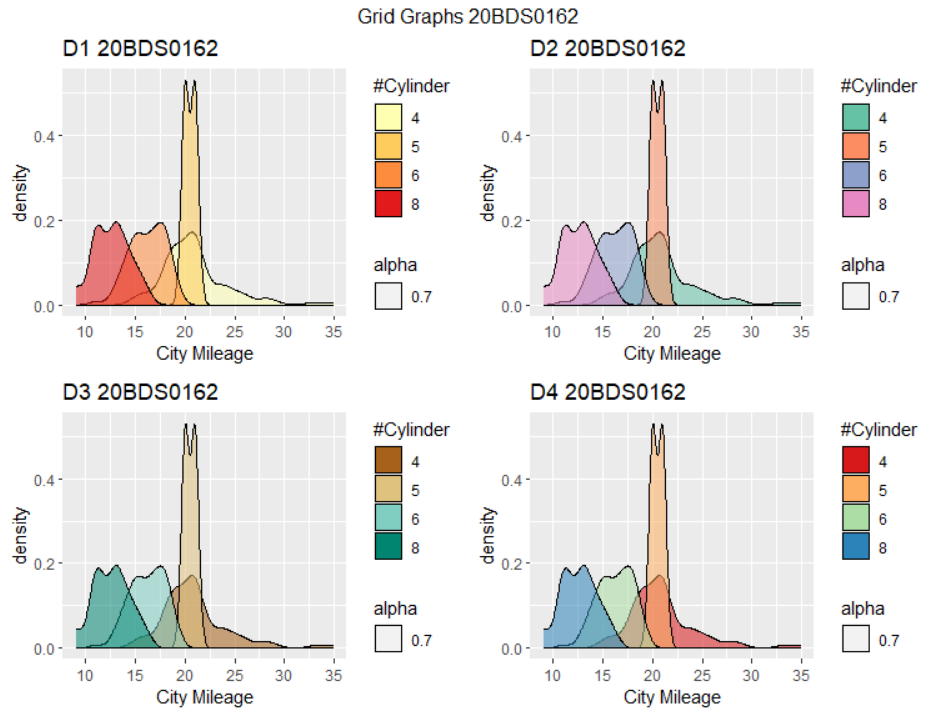
**Output:**











**Result:** We were able to visualize the mpg dataset using the RColorBrewer and plotted various density plots and also shown graphs in a 2x2 grid using the library gridExtra.

**Q3. Load USArrests in-built dataset and correlate in the maps with anyone fields. Display the maps using colormapping.**

**Aim:** To use the USArrests dataset in R and correlate in the maps with anyone field and use color mapping to display the maps.

**Code:**

#Q4

#Colorful map using Viridis

library(maps)

library(ggmap)

library(viridis)

library(dplyr)

View(USArrests)

arrests = USArrests

arrests$region <- tolower(rownames(USArrests))

View(arrests)

#Retrieve the states map

states\_map <- map\_data("state")

View(states\_map)

arrests\_map<-left\_join(x=states\_map,

y=arrests,

by ='region')

View(arrests\_map)

#Create the map

#1.Assault

p1<-ggplot(arrests\_map,aes(x=long,y=lat,group=group))+

geom\_polygon(aes(fill=Assault),color='white')+

labs(title="Assaults Arrests Per State 20BDS0162")+

scale\_fill\_viridis\_c(option='F',direction=1)

#2.Murder

p2<-ggplot(arrests\_map,aes(x=long,y=lat,group=group))+

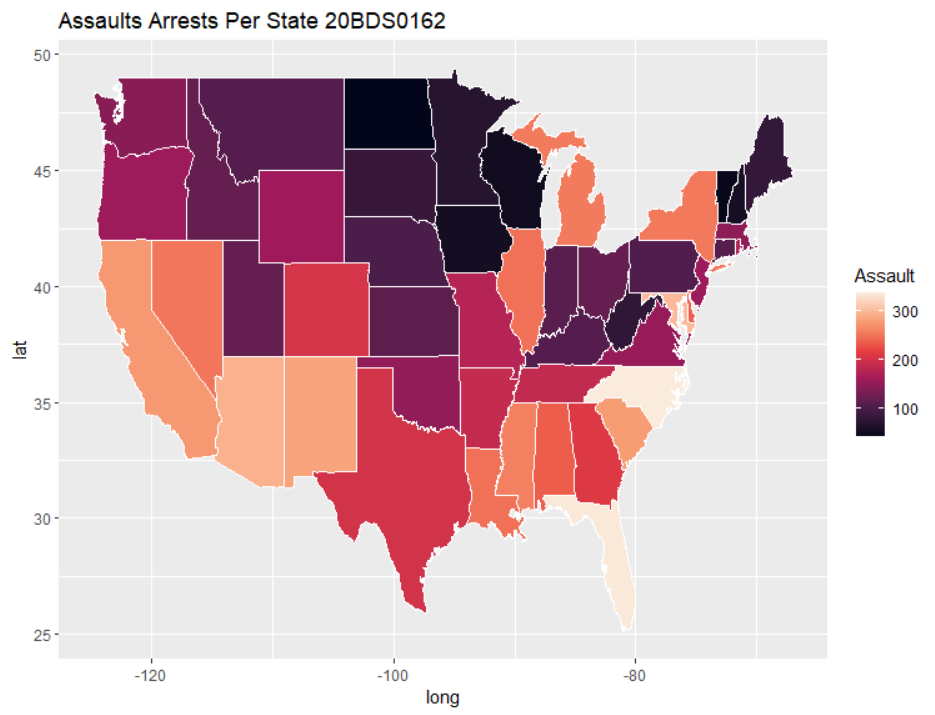
geom\_polygon(aes(fill=Murder),color='white')+

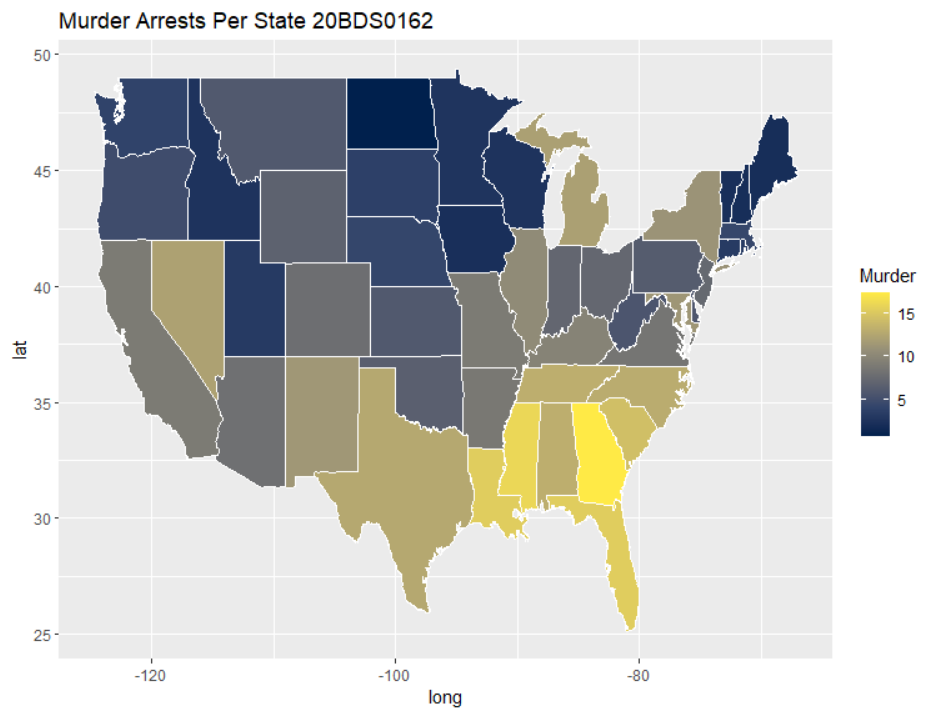
labs(title="Murder Arrests Per State 20BDS0162")+

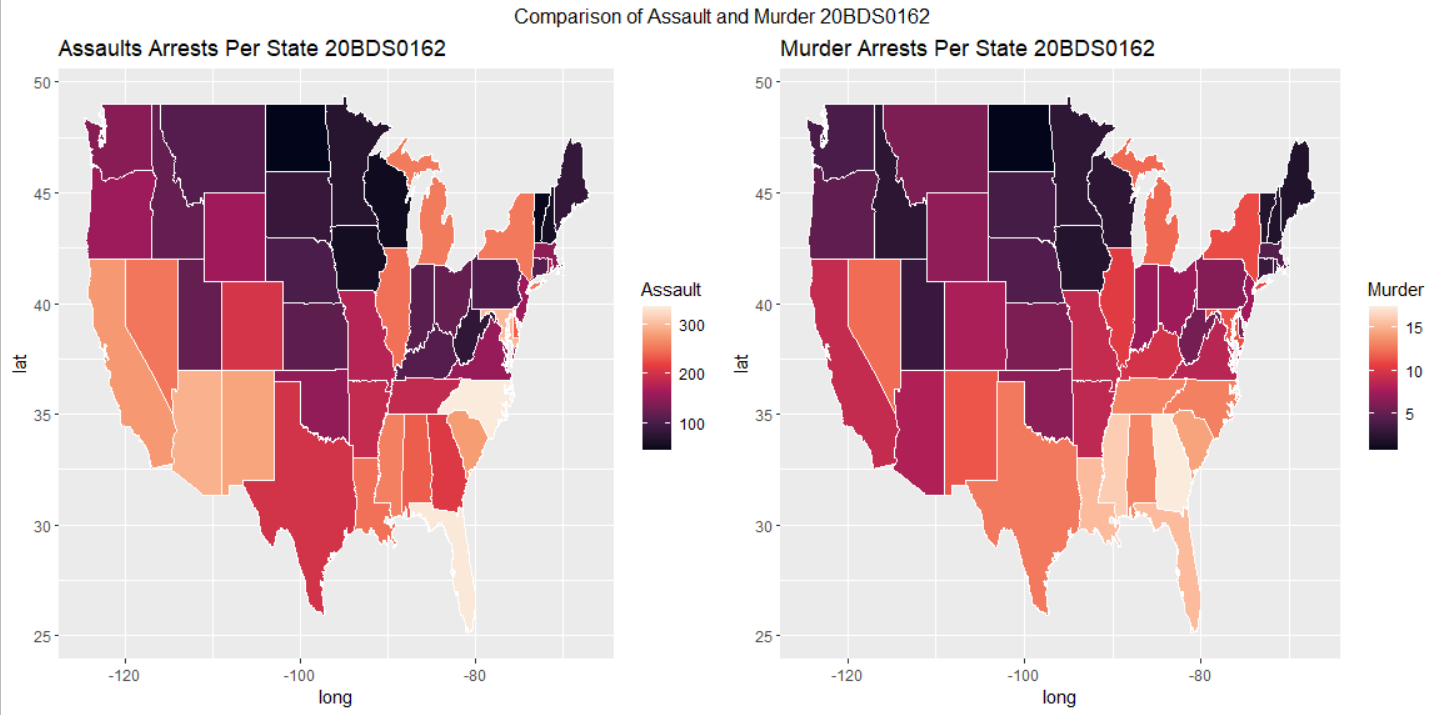
scale\_fill\_viridis\_c(option='F',direction=1)

grid.arrange(p1,p2,nrow = 1,top = "Comparison of Assault and Murder 20BDS0162")

**Output:**







**Result:** We were able to compare Murder and Assaults Per State of the USArrests dataset in R and correlate in the maps and used color mapping to display the maps.