**Basic Visualizations in R**

There are 4 important basic visualizations in R which we came across daily

* Histogram
* Bar plot
* Boxplot
* Scatter plot

**Histogram:**

Histogram plot explains the frequency distribution of a set of continuous data.

It will breaks the data into bins and show frequency distribution of these bins.

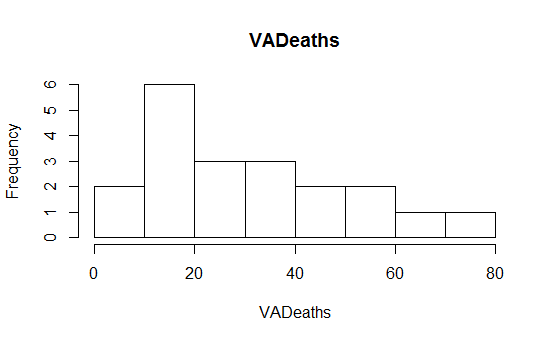
Interesting things that we can infer from the histogram plot**:**

* Normal Distribution
* Skewness
* Outliers
* Kurtosis

**RCode:**

>data(VADeaths)

>hist(VADeaths)



Y-axis: Frequency

X-axis: VADeaths(Continuous data)

**Colored Histogram:**

>Install.packages(“RColorBrewer”)

>library(RColorBrewer)

> help(brewer.pal) # Creates colored bins

>par(mfrow=c(3,3))

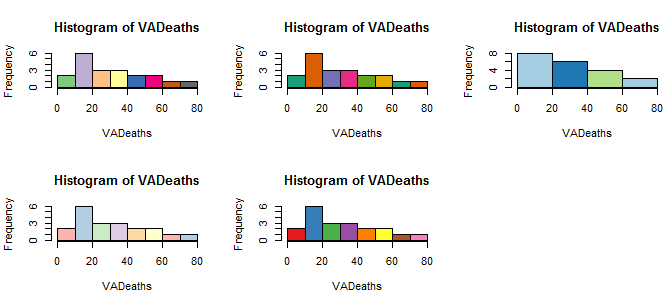
>hist(VADeaths,breaks=8,col=brewer.pal(10,"Accent"))

>hist(VADeaths,breaks=5,col=brewer.pal(6,"Dark2"))

>hist(VADeaths,breaks=3,col=brewer.pal(3,"Paired"))

>hist(VADeaths,breaks=6,col=brewer.pal(6,"Pastel1"))

>hist(VADeaths,breaks=8,col=brewer.pal(10,"Set1"))



**Bar Plot:**

Bar plot will be helpful for making comparisons between cumulative totals across different variables.

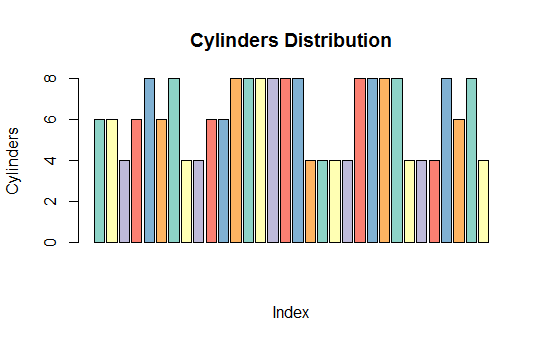
Bar plot for Single variable is a plot between magnitude of variables and its index

**Bar plot for single variable:**

For Continuous data it simply shows rectangular bars(length of each bar represents the magnitude of the variable).

>data(mtcars)

>barplot(mtcars$cyl,col=brewer.pal(6,”Set3”),xlab=”Index”,ylab=”Cylinders”,main=”Cylinders Distribution”)



**For categorical type of data:**

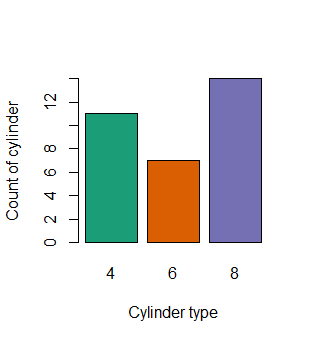
>table(mtcars$cyl)

Mtcars$cyl

4 6 8

11 7 14

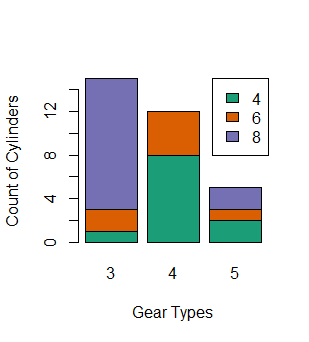
>barplot(table(mtcars$cyl),col=brewer.pal(3,”Dark2”),names.arg=c(“4”,”6”,”8”),xlab=”Cylinder type”,ylab=”Count of Cylinder”)



2 different visualizations with bar plots (For Comparing the cumulative totals across different variables)

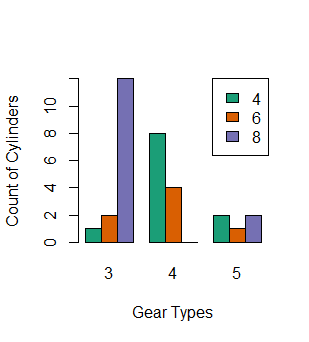
* **Stacked bar plot**

>barplot(table(cyl,gear),names.arg=c(“3”,”4”,”5”),xlab=”Gear Type”,ylab=”Count ofCylinders”,col=brewer.pal(3,”Dark2”)); legend(“topright”,c(“4”,”6”,”8”),fill=brewer.pal(3,”Dark2”),cex=1)



* **Unstacked bar plot**

>barplot(table(cyl,gear),names.arg=c(“3”,”4”,”5”),xlab=”Gear Type”,ylab=”Count ofCylinders”,col=brewer.pal(3,”Dark2”),beside=T); legend(“topright”,c(“4”,”6”,”8”),fill=brewer.pal(3,”Dark2”),cex=1)

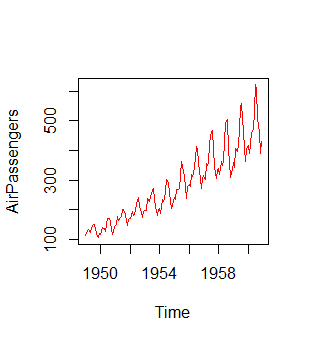


**Line Chart:**

It will be helpful when we have data, which is collected over certain period of time (time series data). To observe any trend, seasonality etc.

>data(“Airpassengers”)

>plot(Airpassengers,type=”l”,col=”red”)



**Box plot:**

Box plot is useful for visualizing the spread of data and to detect any outliers also to know whether the data is skewed or not.

It will display five statistical values

* Minimum
* 1st Quartile (median of 1st 50% of data)
* Median line
* 3rd Quartile (median of last 50% of data)
* Maximum

Inter Quartile Range (IQR) is the spread of the data present inside the rectangular box of Box plot.

Box plot Visualizations:

>data()

>library(datasets)

>data(mtcars)

>data(AirPassengers)

>data(iris)

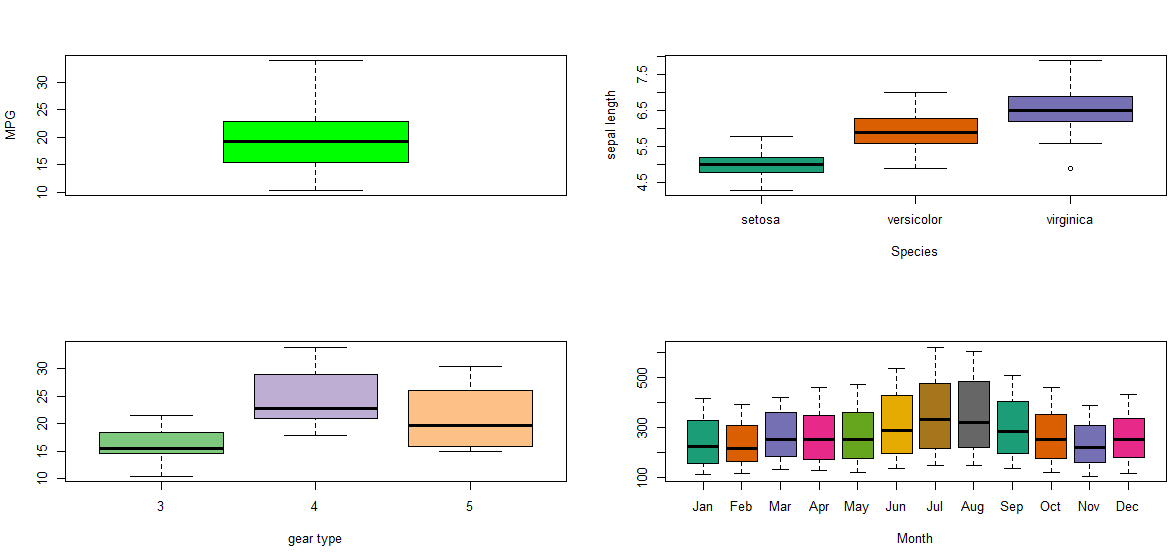
>par(mfrow=c(2,2)

>boxplot(mtcars$mpg,col=”green”,ylab=”MPG”)

>boxplot(iris$Sepal.Length~iris$Species,col=brewer.pal(9,”Dark2”),xlab=”Species”,ylab=”sepal length”)

>boxplot(mtcars$mpg~mtcars$gear,col=brewer.pal(4,”Accent”),xlab=”Gear Type”)

>boxplot(AirPassengers~cycle(AirPassengers),col=brewer.pal(12,”Dark2”),xlab=”Month”,names=c(“Jan”,”Feb”,”Mar”,”Apr”,”May”,”Jun”,”Jul”,”Aug”,”Sep”,”Oct”,”Nov”,”Dec”))



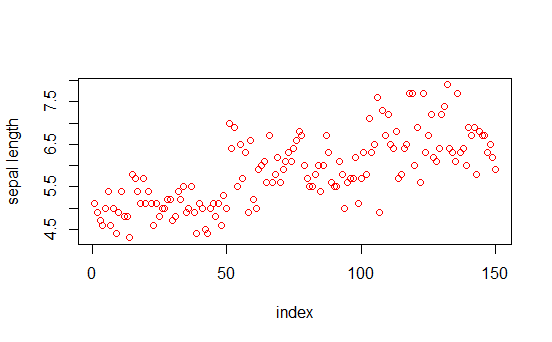
**Scatter Plot:**

For single variable, it is a simple visualization about the spread of the data.

If the scatter plot is between two variables then it shows the extent of correlation between those two variables

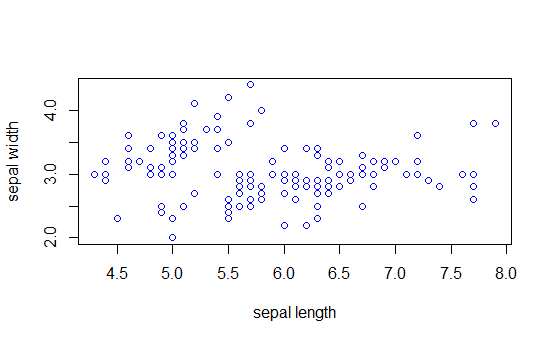
Scatter plot for single variable

>plot(iris$Sepal.Length,xlab=”index”,ylab=”sepal length”,col=”red”,type=”o”)



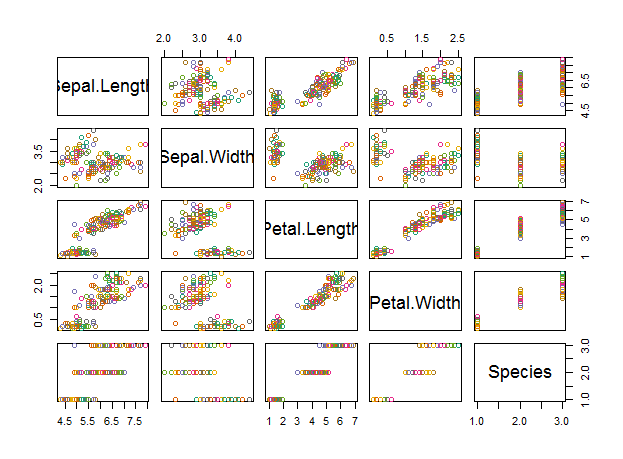
Scatter plot between 2 variables

>plot(iris$Sepal.Length, iris$Sepal.Width, xlab=”sepal length”, ylab=”sepal width”, col=”blue”, type=”p”)



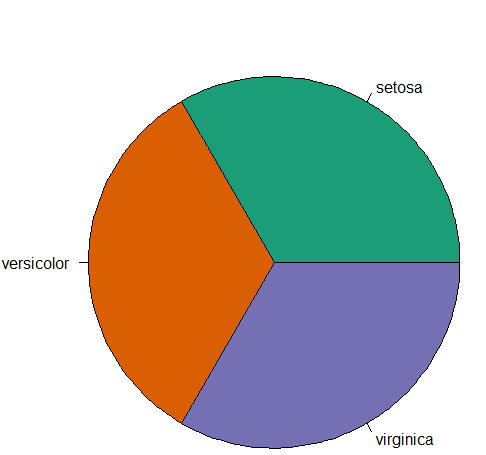
Scatter plot Matrix will help visualize the variables across each other

> Plot(iris,col=brewer.par(10,”Dark2”))



Pie Chart:

>pie(table(iris$Species),col=brewer.pal(3,”Dark2”))

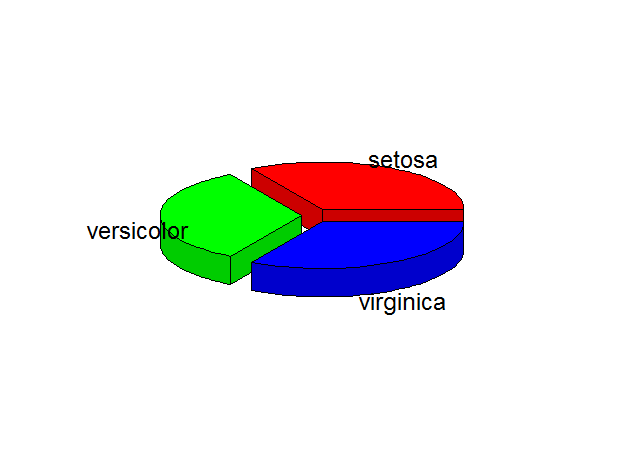


**3D Pie chart**

>install.packages(“plotrix”)

>library(plotrix)

>pie3D(table(iris$Species),labels=c(“setosa”,”versicolor”,”virginica”),explode=0.2)



**Correlogram:**

This is used to test the level of co-relation among the variables present in the data set.

The cells of correlogram are shaded or colored to show the co-relation value.

**Darker the color higher is correlation between the two variables.**

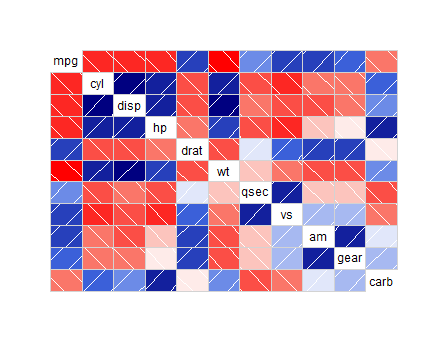
**>data(mtcars)**

**>install.packages(“corrgram”)**

**>library(corrgram)**

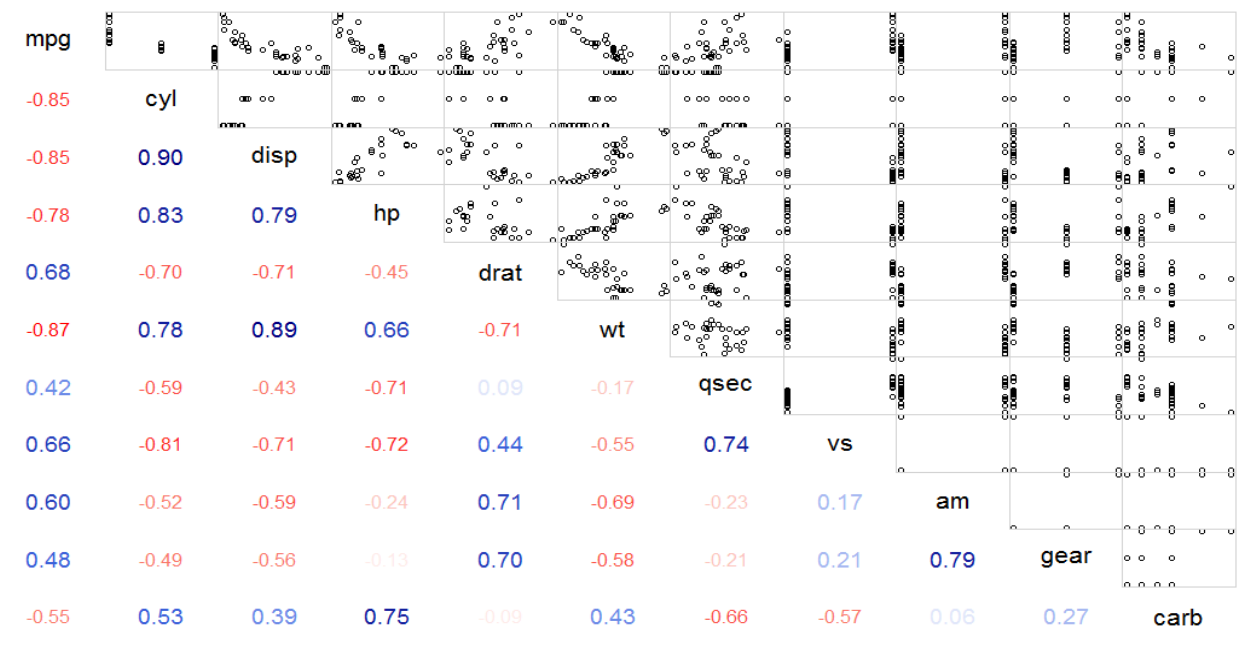
**# Normal Correlogram plot**

**>corrgram(mtcars)**

****

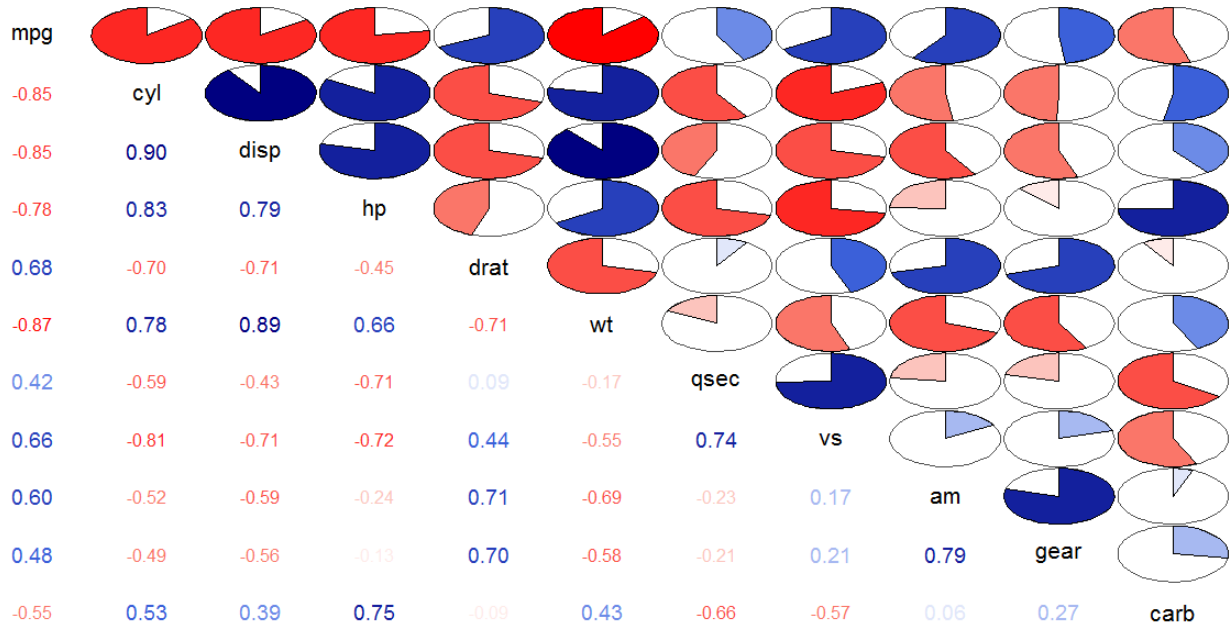
**# Correlogram with Scatter plot between variables**

**>corrgram(mtcars,lower.panel=panel.cor,upper.panel=panel.pts)**



**# Correlogram with correlation values and pie plot**

**>corrgram(mtcars,lower.panel=panel.pie,upper.panel=panel.pts)**



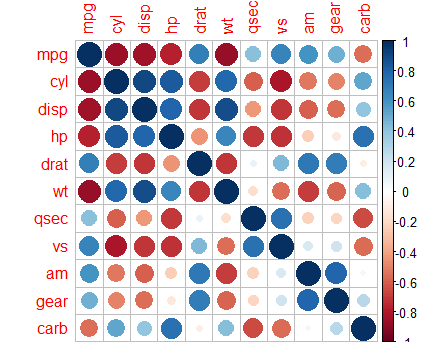
**# Correlation and Correlogram plot**

**>install.packages(“corrplot”)**

**>library(corrplot)**

**Cor\_matrix<-corr(mtcars) # Correlation values between variables**

**>corrplot(cor\_matrix) # Normal Correlation Correlogram plot**



**# To display Correlation coefficient values between any 2 variables**

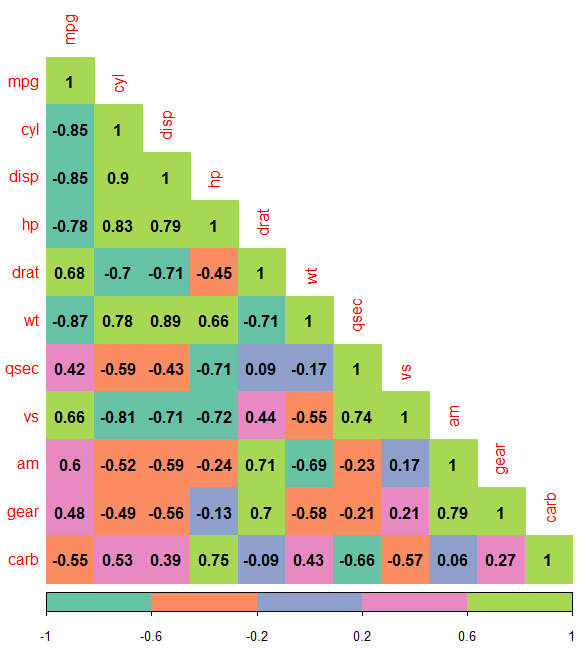
**>corrplot(cor\_matrix,method=”number”)**



**# Specialized the insignificant value according to the significance level**

**>corrplot(cor\_matrix,type=”lower”,**

**addCoef.col=”black”,method=”color”,col=brewer.pal(5,”Set2”))**



**# 3 Dimensional View of Scatter plot**

**This can be helpful in identifying the clusters formed in the scatter plot across the variables.**

**>install.packages(“scatterplot3d”)**

**>library(scatterplot3d)**

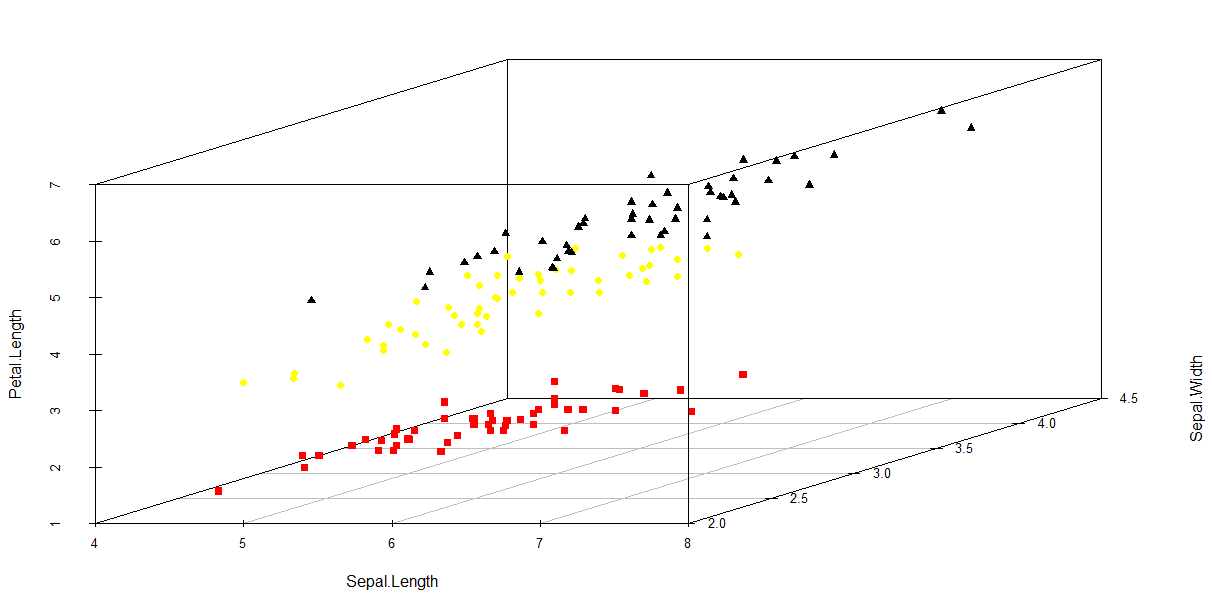
**>shapes<-c(15,16,17)**

**>shapes<-shapes[as.numeric(iris$Species)]**

**>colors<-c(“red”,”yellow”,”black”)**

**>colors<-colors[as.numeric(iris$Species)]**

**>scatterplot3d(iris[,1:3], pch=shapes, color=colors)**



# **To Display the Scatter plot, Histogram and Correlation Coefficient values of data across any two variables in a single plot**

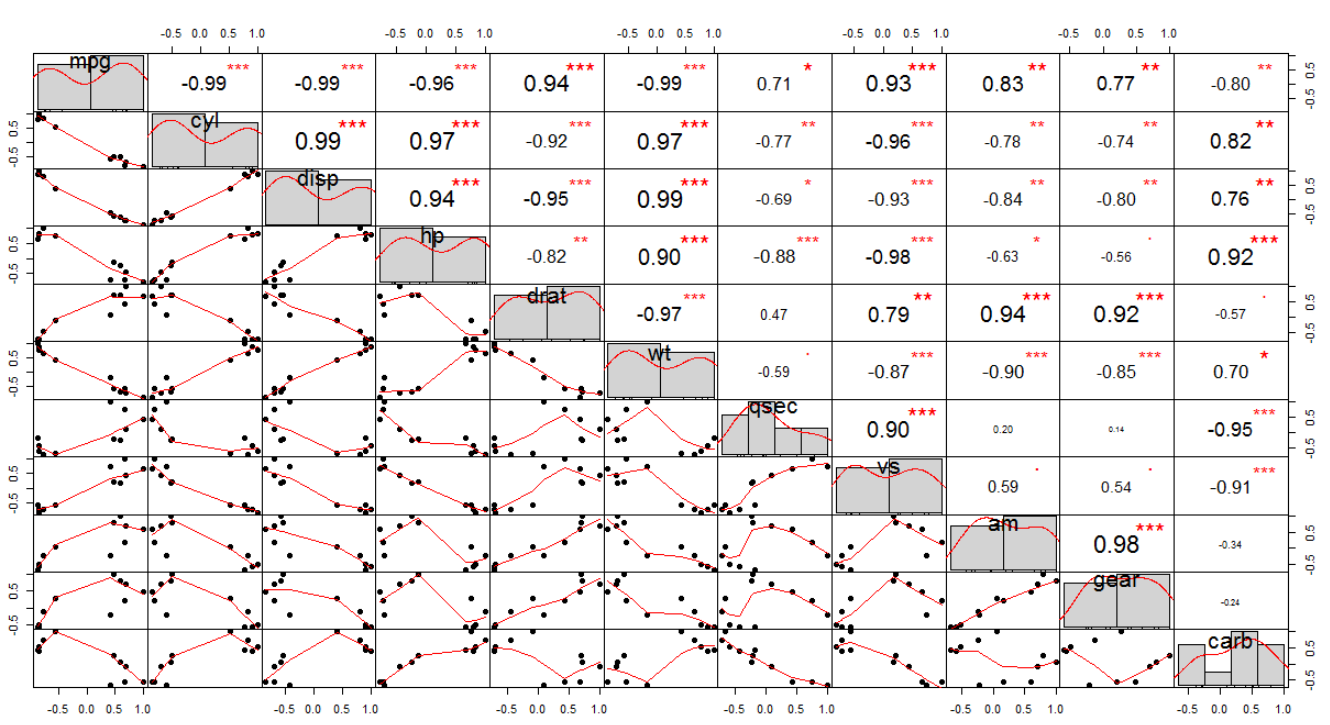
**>data(mtcars)**

**>cor\_matrix<-corr(mtcars)**

**>install.packages(PerformanceAnalytics)**

**>library(PerformanceAnalytics)**

**>chart.Correlation(cor\_matrix,histogram=T,pch=19)**



Hexabins plot:

What is Hexabin Binning?

It is a great alternative technique for visualizing the density of data of large data sets

It will create colored hexagonal figures over a gridded network, the intensity of the color of an hexagon will tells us the spread of the data at that region. More is the intensity, more is the overlapping or spread (located) of data in that region.

**Normal Hexbin plot:**

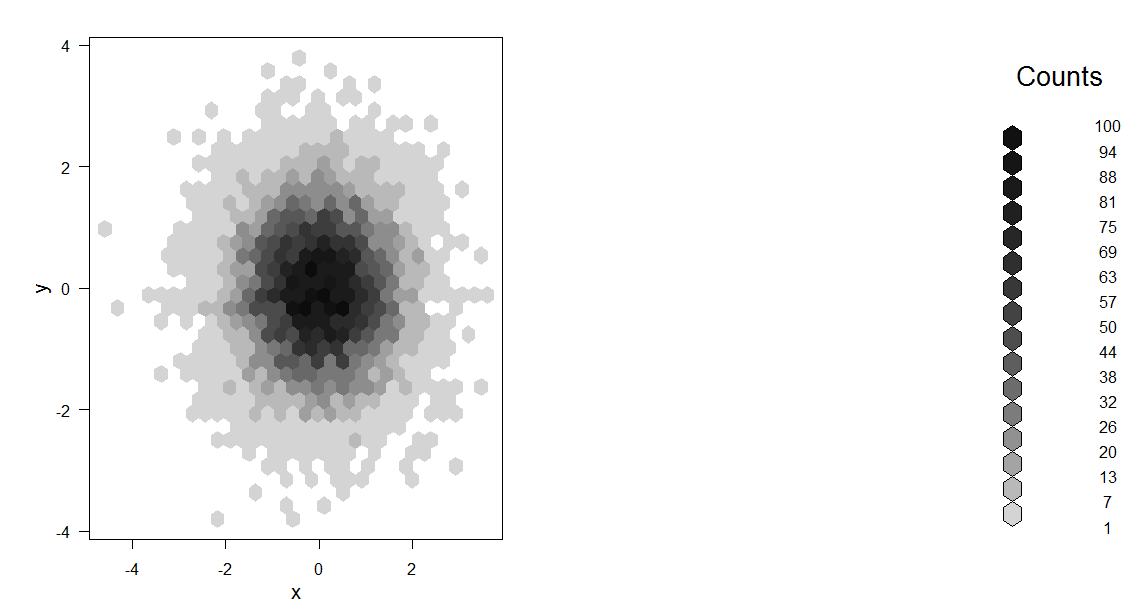
>x<-rnorm(20000) # selecting random 20000 numbers

>y<-rnorm(20000) # selecting random 20000 numbers

>install.packages(“hexbin”)

>BIN<-bin(x,y)

>plot(BIN)

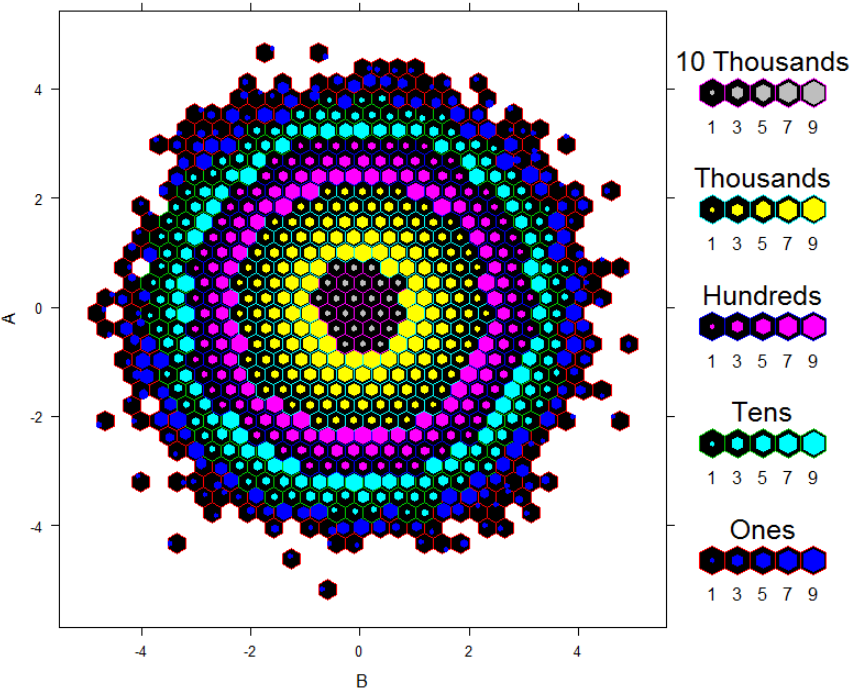


**Colored Hexbin Plot:**

**>A<-rnorm(1e6)**

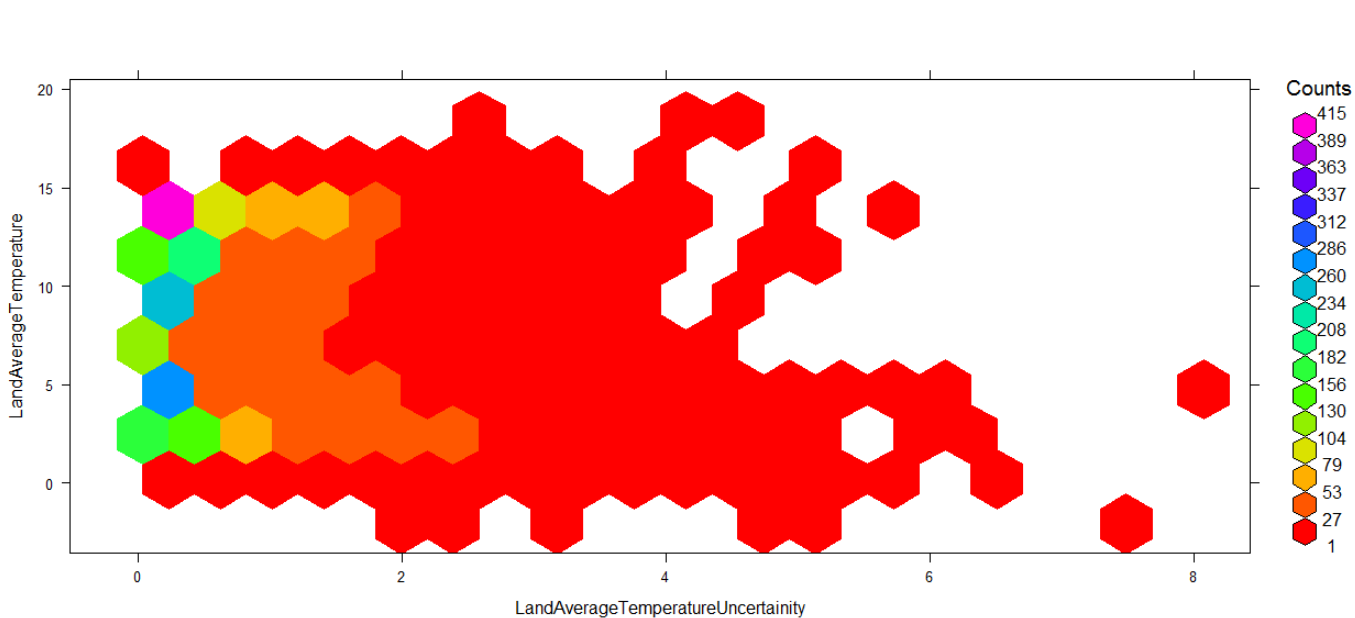
**>B<-rnorm(1e6)**

**>hexbinplot(A~B,colramp = colorRampPalette(brewer.pal(4,"Dark2")),style="nested.centroids")**



**# Hexbin plot for LandAverageTemperature Vs LandAverageTemperatureUncertainity**

**hexbinplot(LandAverageTemperature~LandAverageTemperatureUncertainity,colramp=colorRampPalette(rainbow(7)),style="nested.centroids",xbins=20)**

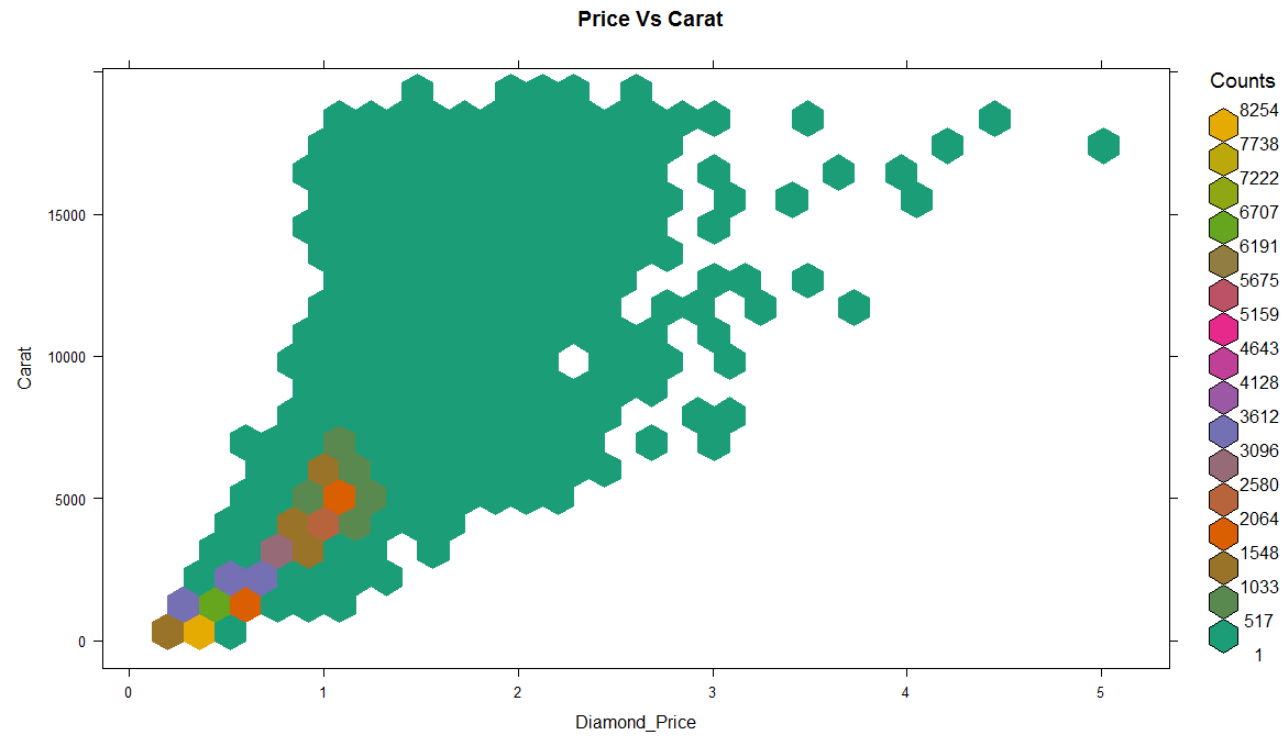


**# Hexbin plot for Diamond Price Vs Diamond Carats**

**>library(“ggplot”)**

**>data(“diamonds”)**

**>hexbinplot(diamonds$price,diamonds$price,xlab=”Diamond\_Price”,ylab=”Diamond\_Carats”,main=”Diamonds Price Vs Diamonds\_Carat”,colramp =colorRampPalette(brewer.pal(6,”Dark2”))**



**Mosaic Plot :**

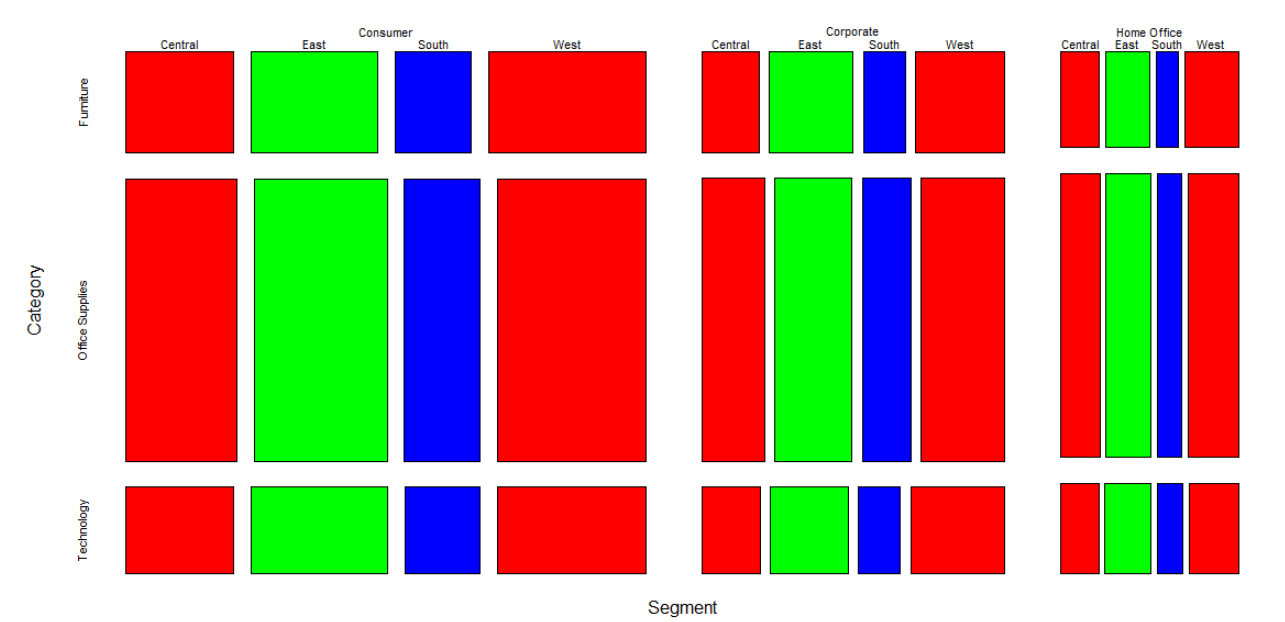
Graphical method for visualizing the data for two or more qualitative variables. The visualization will helps us to identify the relation between the variables, rectangular bin size indicates the number of observations for the given qualitative variable.

# Loading super sample store

# Considering Category,Region,Segment - columns

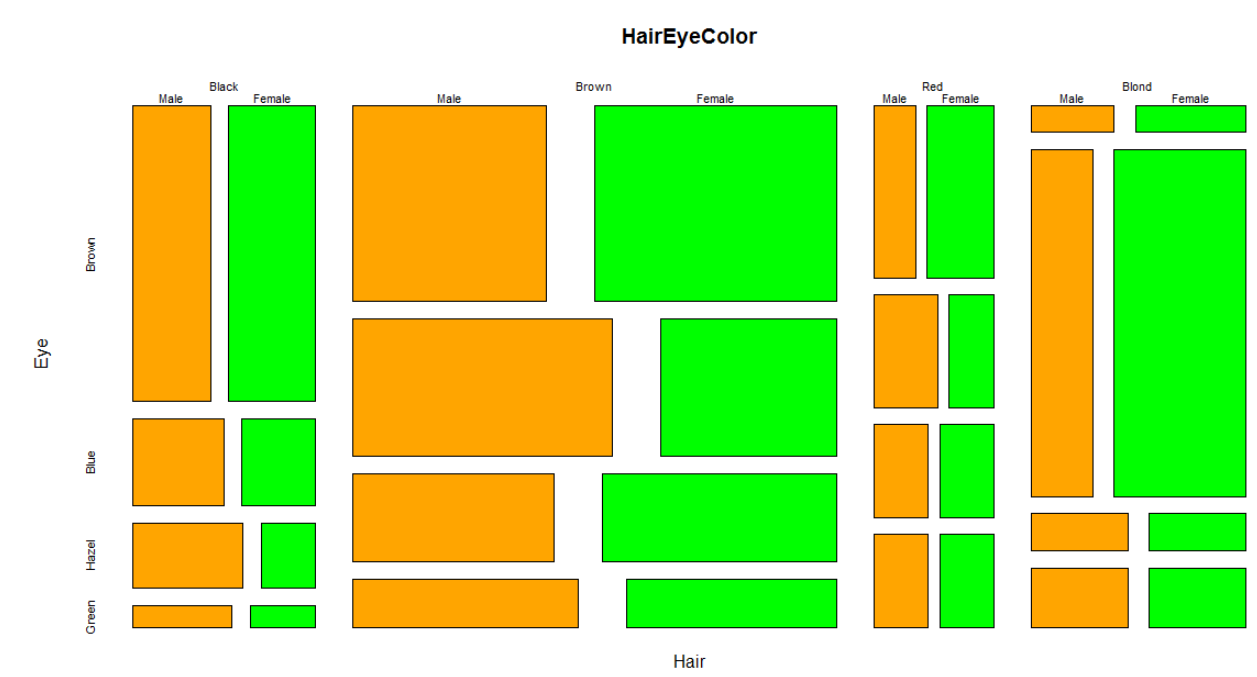
>mosaicplot(Segment~Category+Region,legend=T,

col=c(“red”,”green”, ”blue”)



**>data(“HairEyeColor”)**

**>mosaicplot(HairEyeColor,col=c(“orange”,”green”))**

****

**Package: “raster”**

**>install.packages(“raster”)**

**>library(raster)**

**>ccodes() # to get the country codes**

**# Displaying a country’s global boundaries**

**>c\_ind\_0<-getData(‘GADM’,country=”IND”,level=0)**

**# getData() will acquire the data about global amd boundaries and**

**# It will directly download into R.**

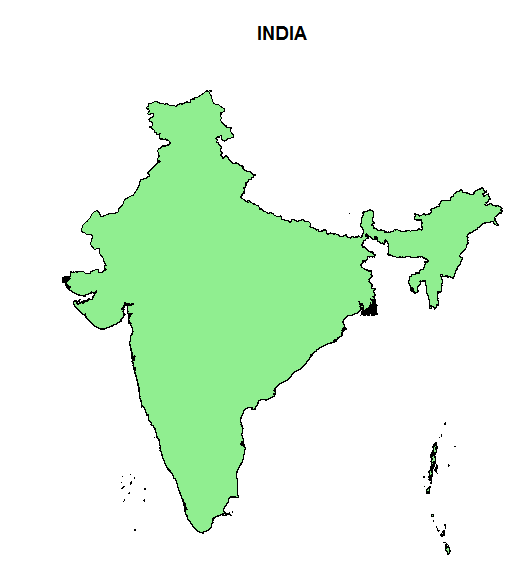
**#GADM => Global administrative boundaries**

**# Country = “country code”**

**# To display all country codes => ccodes()**

**# level => level of administrative sub division ( 0 or 1 or 2)**

**>plot(c\_ind\_0,col=”lightgreen”,main=”INDIA”)**

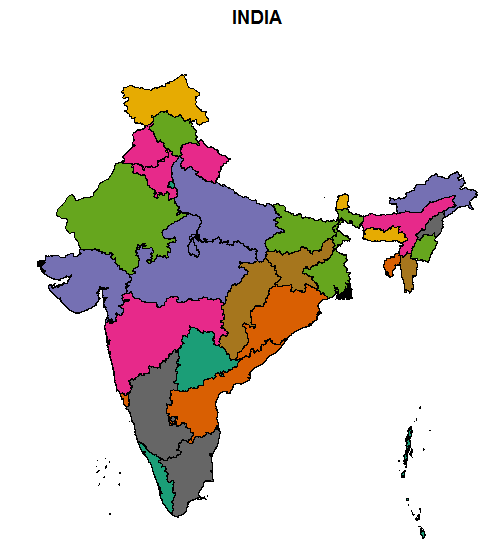


**# To display the states**

**>library(RColorBrewer)**

**>c\_ind\_1<-getData(“GADM”,country=”IND”,level=1)**

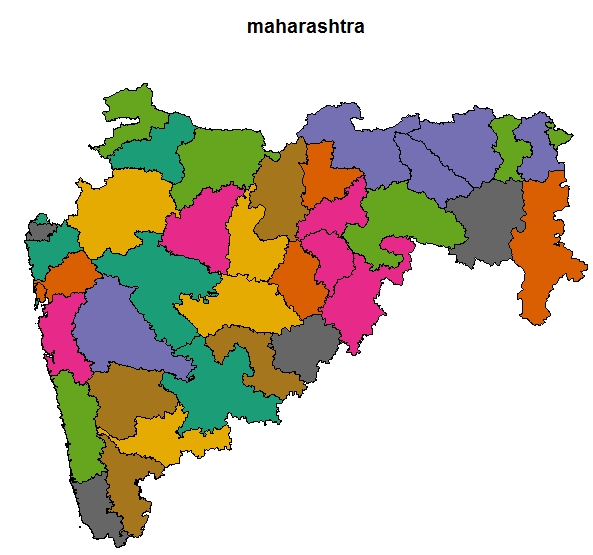
**>plot(c\_ind\_1,col=brewer.pal(12,”Dark2”),main=”INDIA”)**



**>c\_ind\_2<-getData(“GADM”,country=”IND”,level=2)**

**>c\_ind\_2\_maharashtra<- subset(c\_ind\_2,c\_ind\_2$Name\_1==”Maharashtra”)**

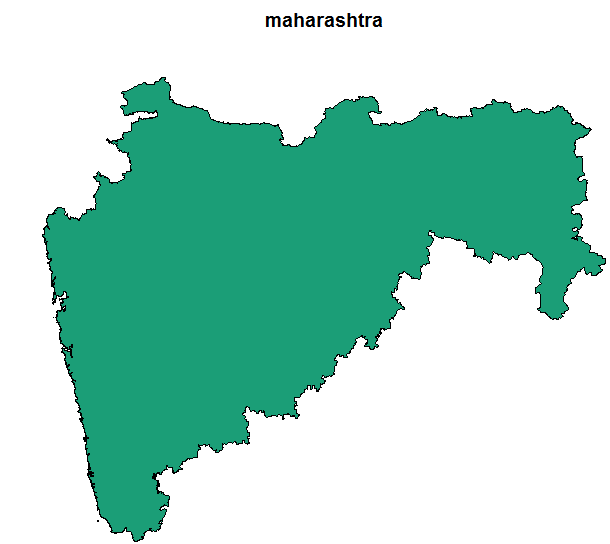
**>plot(c\_ind\_2\_maharashtra,col=brewer.pal(12,”Dark2”))**



**>c\_ind\_1<-getData(“GADM”,country=”IND”,level=1)**

**>c\_ind\_1\_maharashtra<- subset(c\_ind\_1,c\_ind\_1$Name\_1==”Maharashtra”)**

**>plot(c\_ind\_1\_maharashtra,col=brewer.pal(12,”Dark2”))**



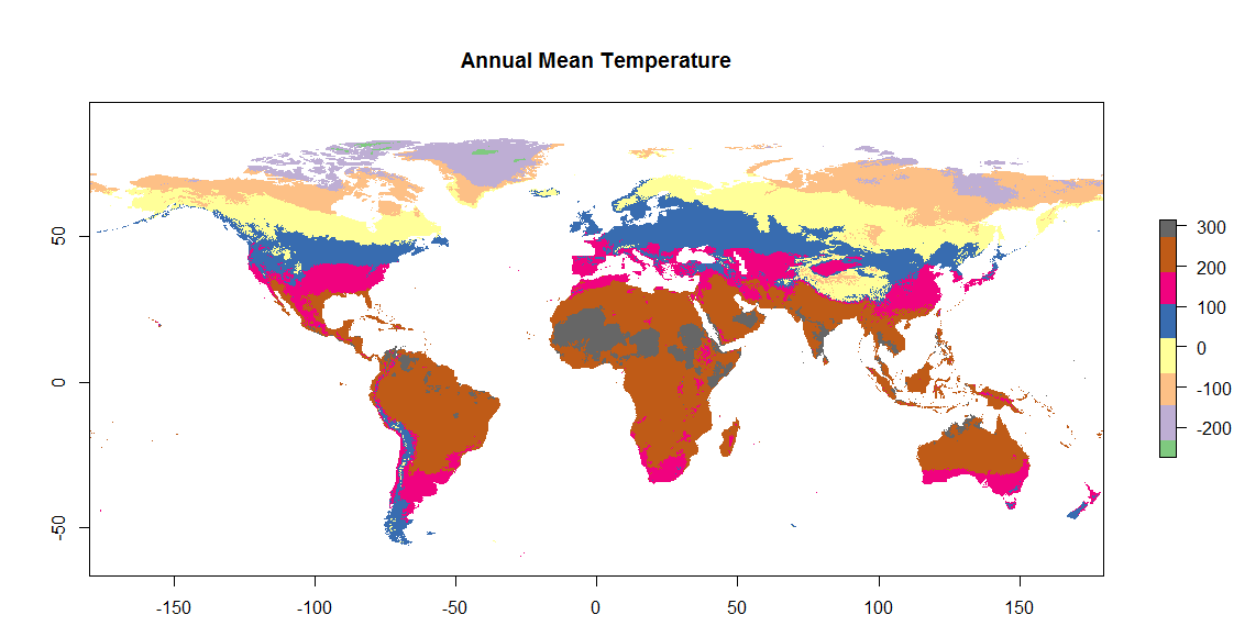
**# World Climate**

**>Climate<-getData(“worldclim”,var=”bio”,res=2.5)**

**# “worldclim” => returns world climate data**

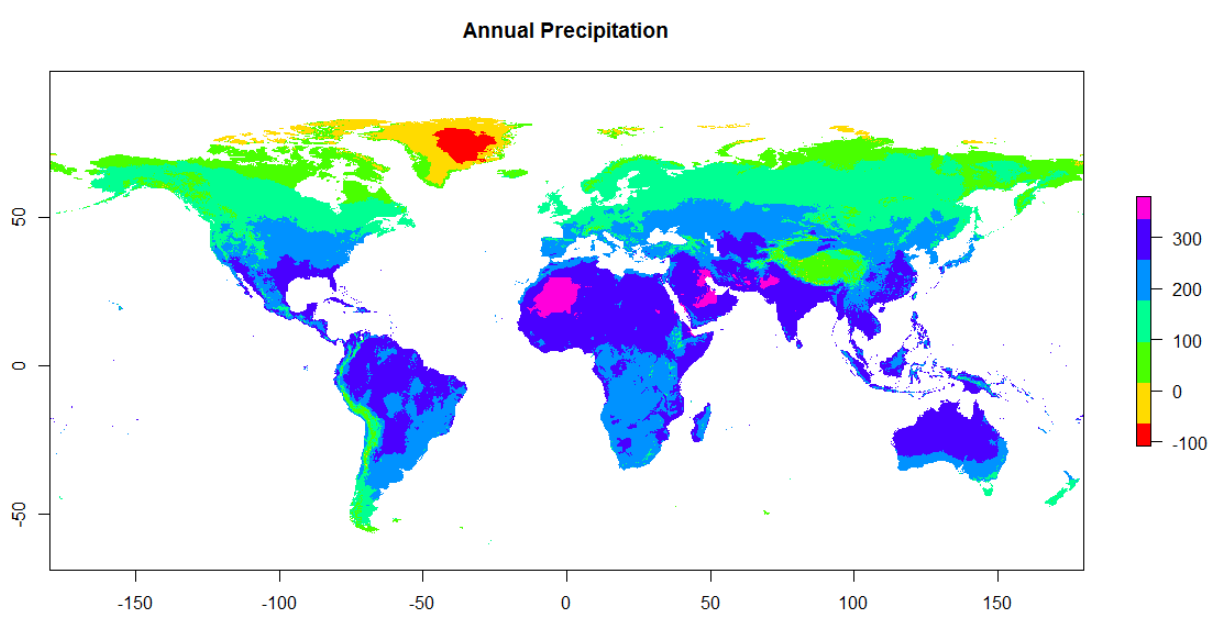
**# Annual Mean Temperature across world**

**>Plot(Climate$bio1,col=brewer.pal(8,”Dark2”),main=”Annual Mean Temperature”)**

****

**# Annual Precipitation**

**>plot(Climate$bio10,col=brewer.pal(8,”Accent”),main=”Annual Mean Temperature”)**



**# Geographical Elevation plot**

**>library(“raster”)**

**>Srtm<-getData(‘SRTM’,lon=16,lat=48)**

**>**

**# Displaying Maps**

**>install.packages("maps")**

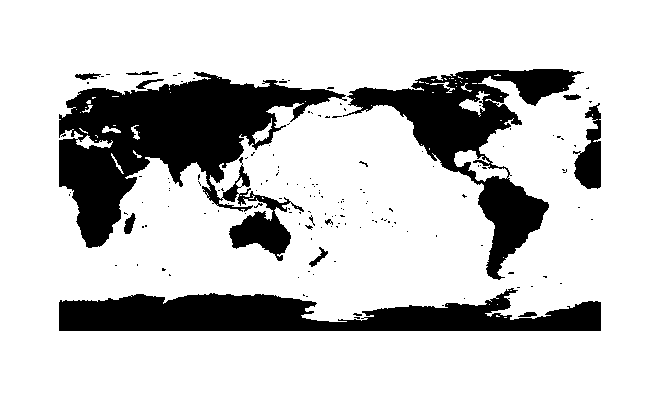
**>install.packages("mapdata")**

**>library(maps)**

**>library(mapdata)**

**>map(“world2”,fill=T)**

**# World Map**



**# Visualization for Large data sets**

* **To explore and analyze large multivariate data sets we will use table plot, which is a very powerful visualization method.**
* **Tableplots are used to explore the the relationships between variables and to discover different patterns and occurrence of missing values.**

**>install.packages(ggplot2)**

**>library(ggplot2)**

**>data(diamonds)**

**>table(is.na(diamonds$price)**

**FALSE  
 53940**

**# Means there are no NA values**

**# Lets add some NA values to the diamonds data set**

**>is.na(diamonds$price)<-diamonds$cut == “IDEAL”**

**>table(is.na(diamonds$price))**

**FALSE TRUE**

**32389 21551**

**# Now we have added 21551 missing values under price column**

**>table(is.na(diamonds$cut))**

**FALSE**

**53940**

**# adding missing values to cut column**

**>is.na(diamonds$cut)<-(runif(nrow(diamonds))>0.8)**

**>table(is.na(diamonds$cut))**

**FALSE TRUE**

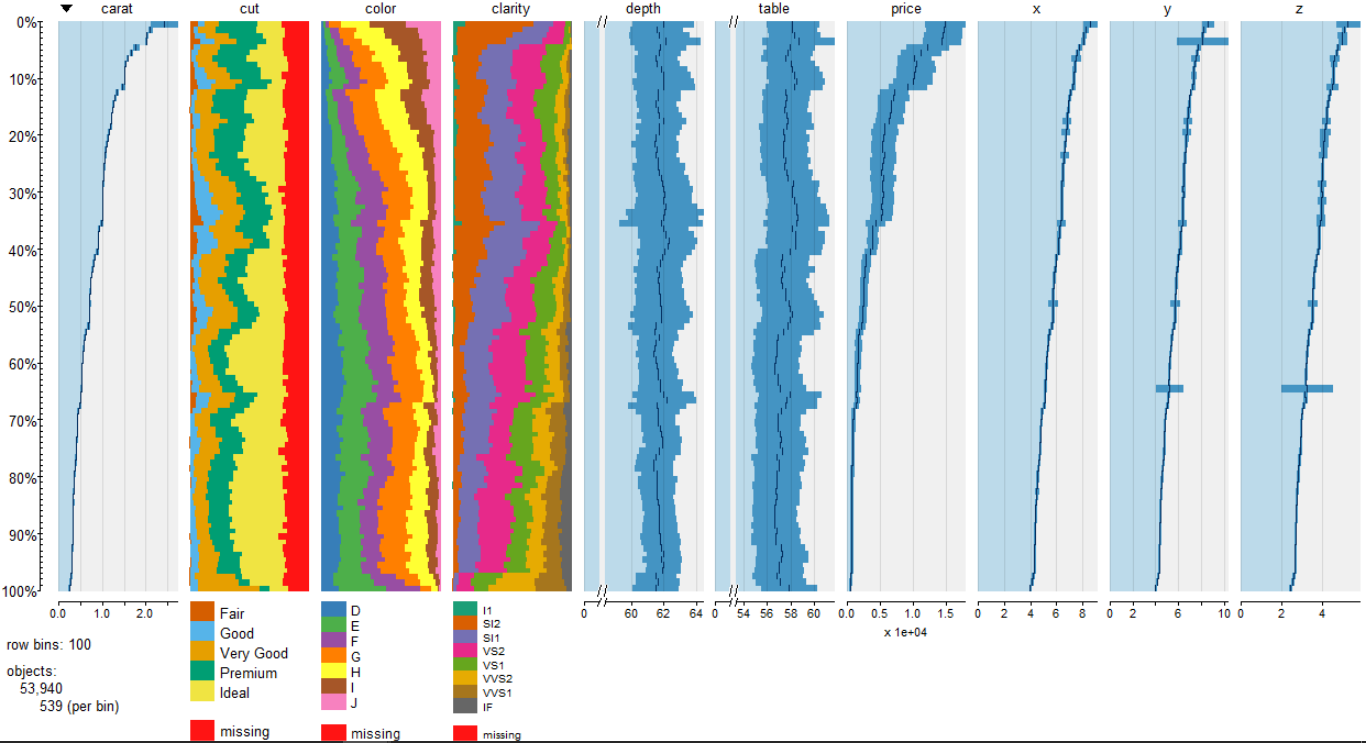
**32389 21551**

**# When we apply tableplot to the data set by default all the values are sorted according the first column. With argument sortCol we can specify on which column the data is sorted (descending order).**

**>install.packages(“tabplot”)**

**>library(tabplot)**

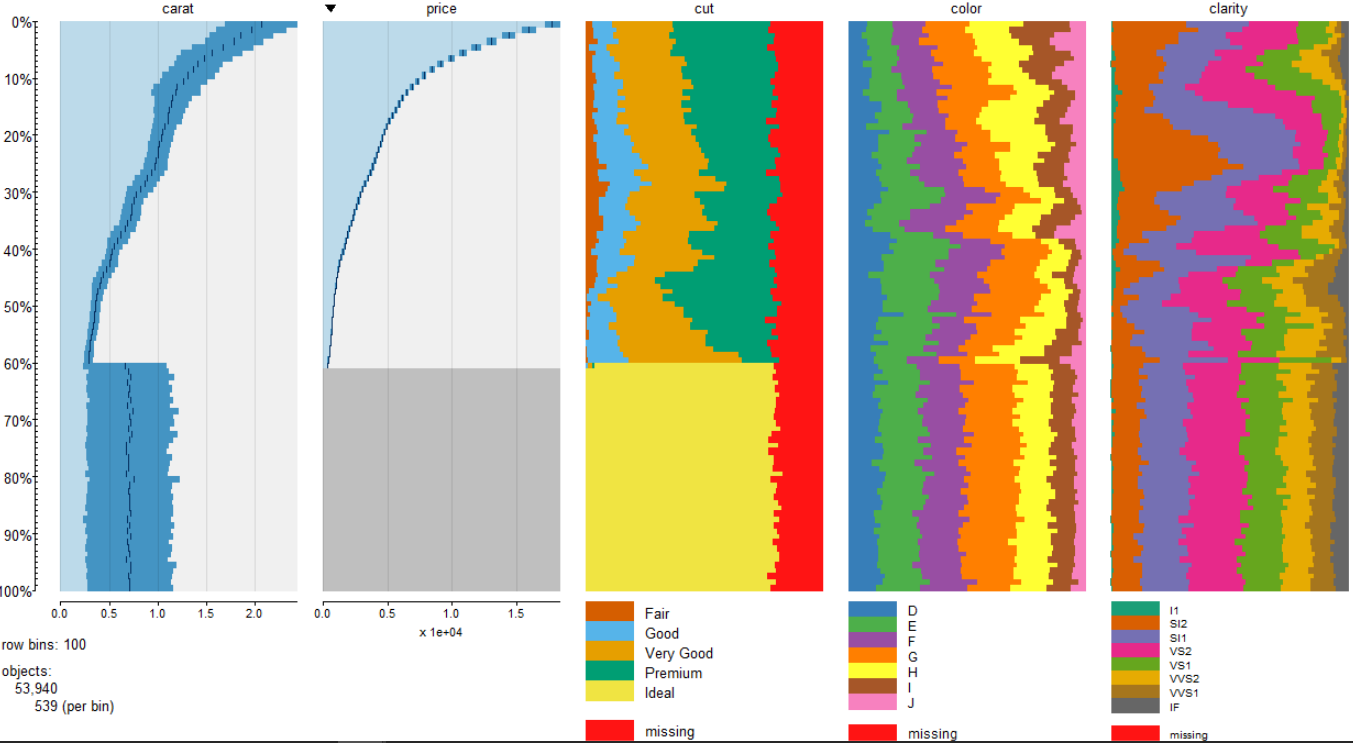
**>tableplot(diamonds)**



**# missing values are shown in Red color and by default it will selects all the columns**

**# To select specific columns use select parameter**

**>tableplot(diamonds,select=c(carat,price,cut,color,clarity),sortCol=price)**

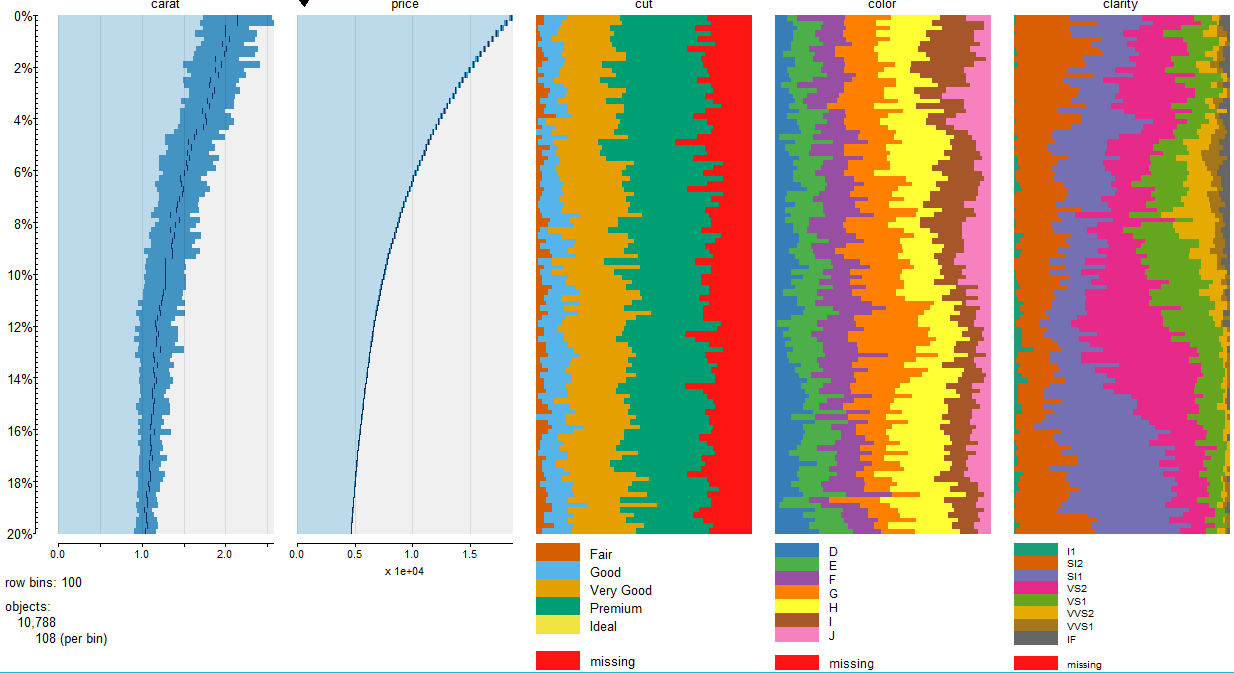


**# Lets visualize the data for top 20% of diamonds price**

**# for this we have to mention two parameters**

**# from =0, to=10**

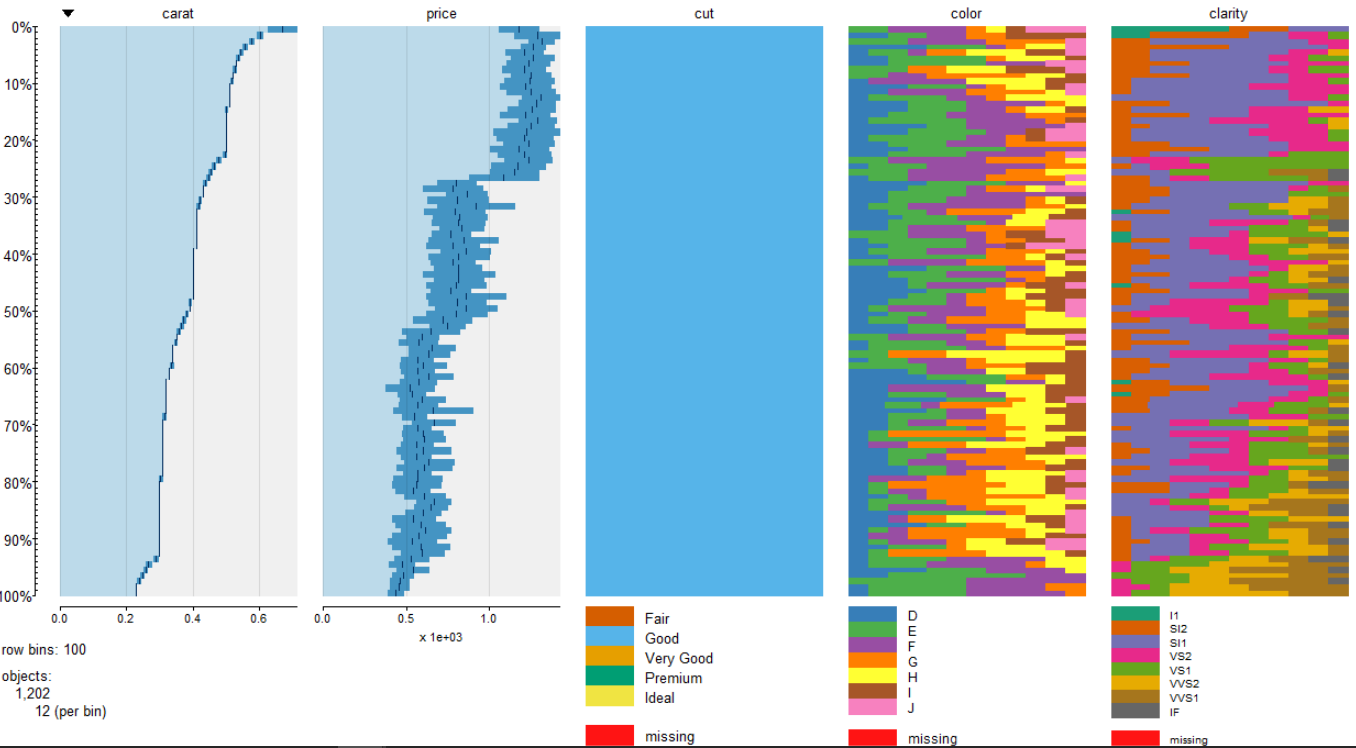
**>tableplot(diamonds,select=c(carat,price,cut,color,clarity),sortCol=price,from=0,to=20)**



**# Lets do some filtering. For this we have to use subset parameter**

**>tableplot(diamonds,select=c(carat,price,cut,color,clarity),subset = price<1500 & cut==”Good)**

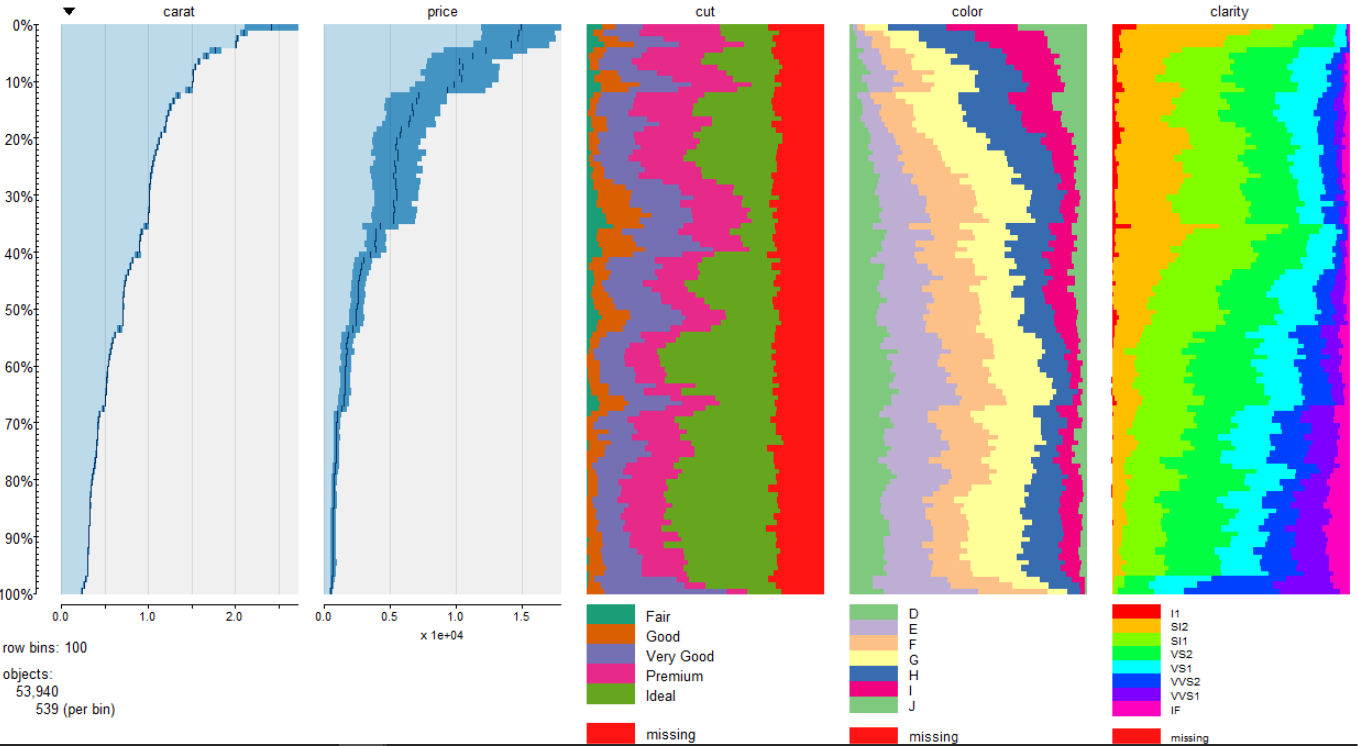
**# We are selecting only value “Good” under cut column and diamonds price which are under 1500**



**# To change the colors of categorical variables with different colors we use pals parameter**

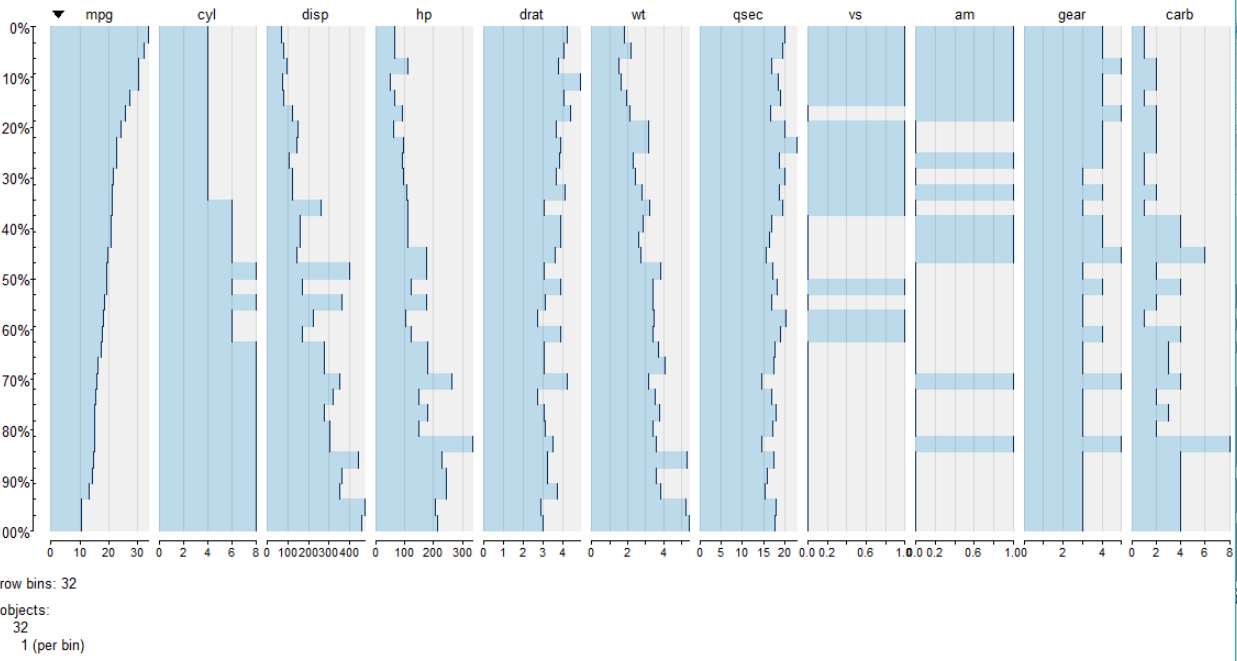
**>** **library(RColorBrewer)**

**>** **tableplot(diamonds,select=c(carat,price,cut,color,clarity),pals = list(cut=brewer.pal(5,"Dark2"),color=brewer.pal(6,"Accent"),clarity=rainbow(8)))**



**>data(mtcars)**

**>tableplot(mtcars)**



**# It seems like there are no missing values present in the mtcars data set**

**# Visualization of missing values by missmap function defined in “Amelia” package**

**>data(mtcars)**

**# This data contains no NA values, lets add some NA values to the data set**

**>is.na(mtcars$gear)<-(mtcars$gear==3)**

**>table(is.na(mtcars$gear))**

**FALSE TRUE**

**17 15**

**>is.na(mtcars$mpg)<-(mtcars$mpg<15)**

**>table(is.na(mtcars$mpg))**

**FALSE TRUE**

**27 5**

**>is.na(mtcars$wt)<-(mtcars$wt<2.4)**

**>table(is.na(mtcars$wt))**

**FALSE TRUE**

**27 5**

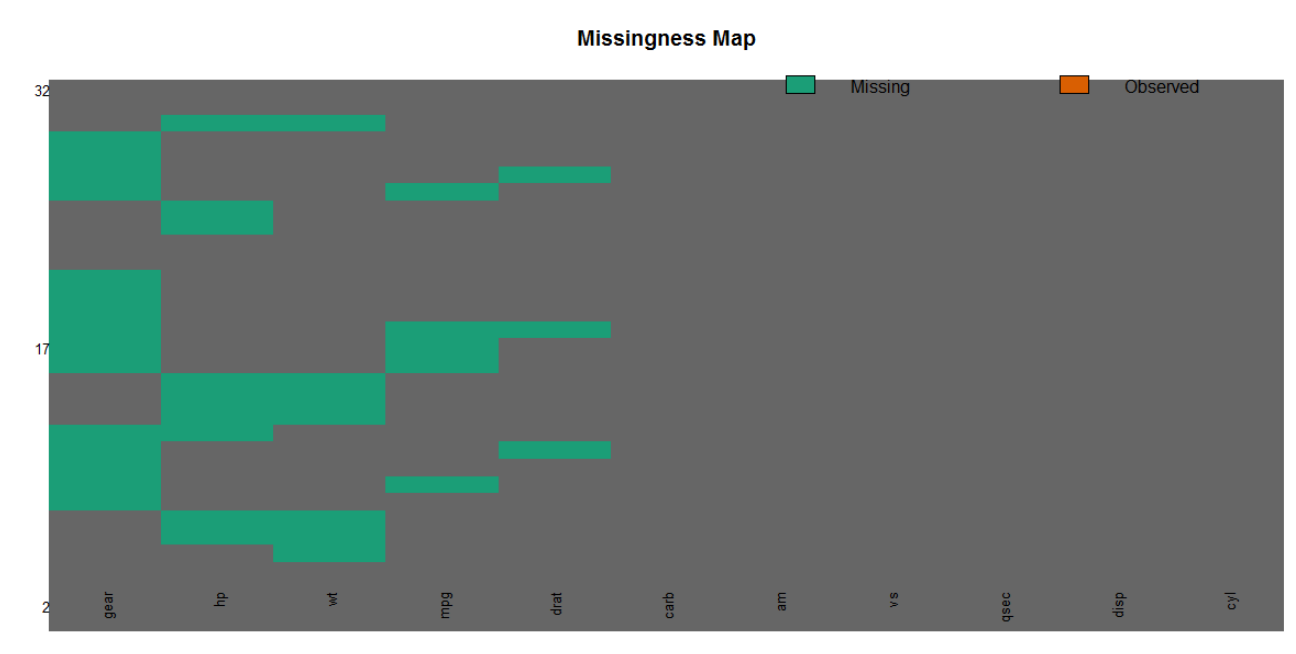
**>is.na(mtcars$hp)<-(mtcars$hp<100)**

**>is.na(mtcars$drat)<-(mtcars$drat<3)**

**>install.packages(“Amelia”)**

**>library(Amelia)**

**>missmap(mtcars)**



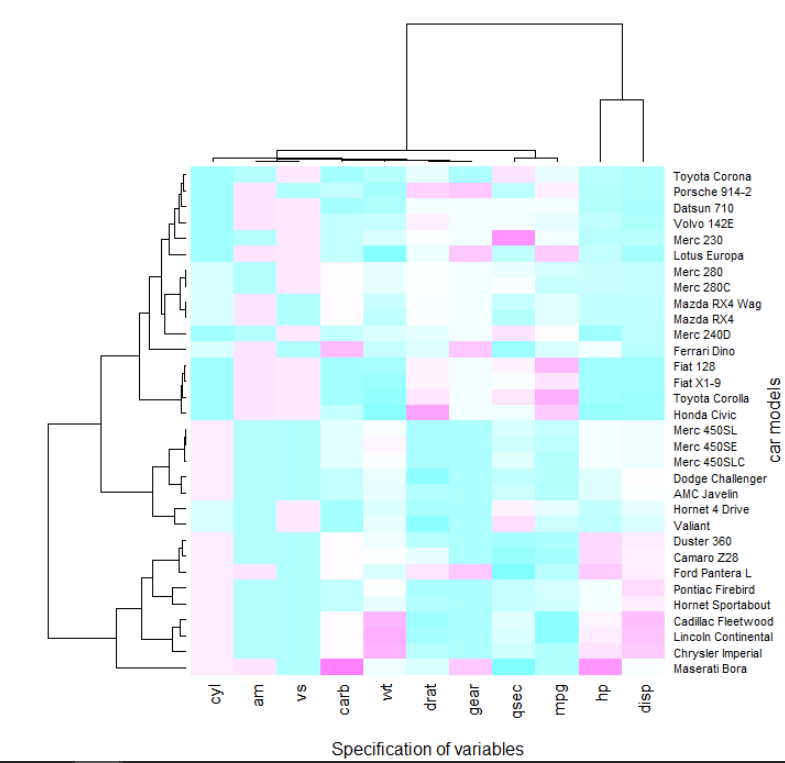
As we can see from the visualizations that green color indicates the missing values of the data.

**Heat maps:**

It is used to visualize a table of numbers where numbers are substituted with colored cells. Intensity of colors correspond the level of measurement magnitude objects in each column.

>data(“mtcars”)

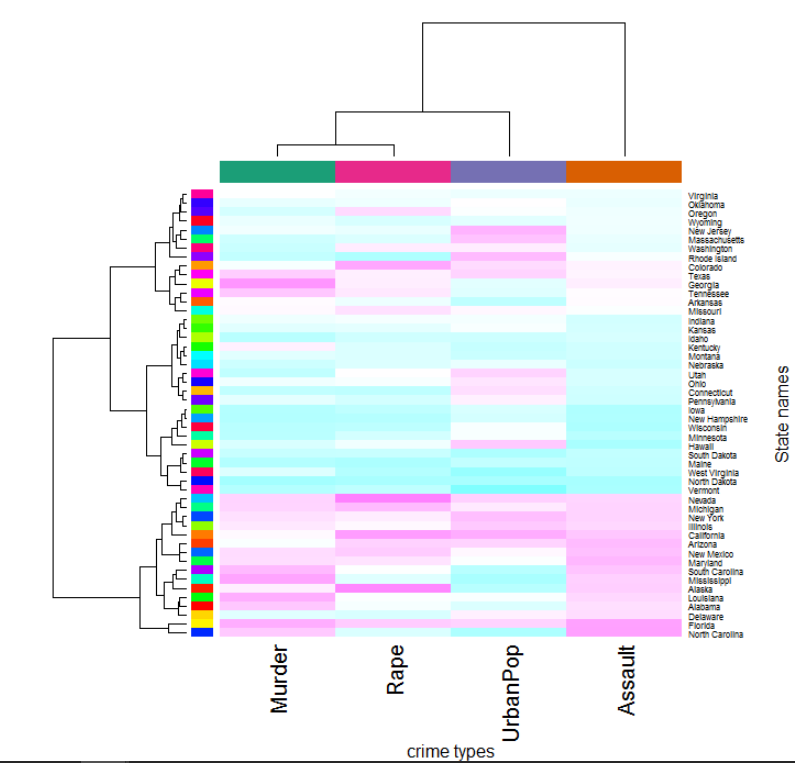
>install.packages(“gplots”)  
>library(gplots)  
>heatmap(data.matrix(mtcars,col=cm.colors(256),xlab=”car models”,ylab=”specification variables”)

****

>data(“USArrest”)

>heatmap(data.matrix(USArrests), col = cm.colors(256), scale = "column",

RowSideColors = rainbow(50), ColSideColors = brewer.pal(4,"Dark2"), margins = c(9,8),xlab = "crime type", ylab = "state names")



**# Visualization of unemployment in US using “blscrapeR” package**

**>install.packages(“blscrapeR”)**

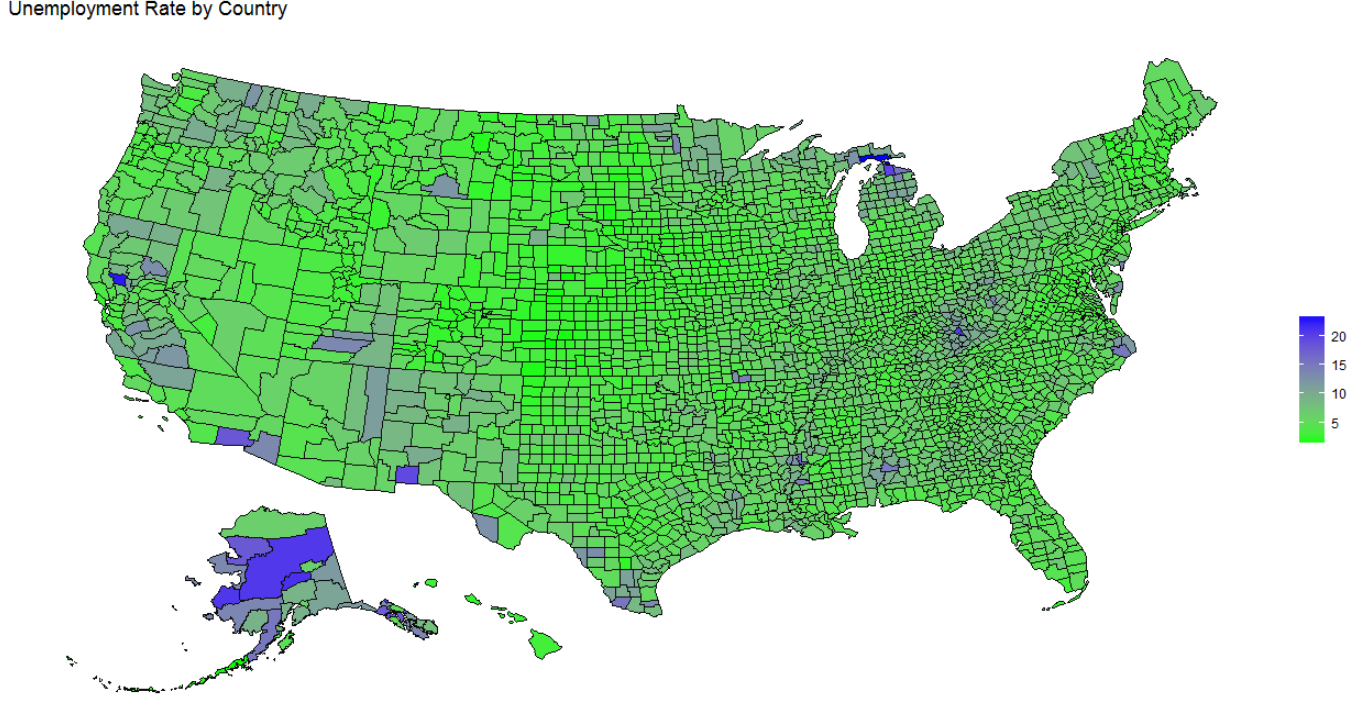
**>library(blscrapeR)**

**# To get the past of labor**

**>unemployment<-get\_bls\_county()**

**#Unemployment rate by country**

**bls\_map\_county(map\_data = df, fill\_rate = "unemployed\_rate", labtitle = "Unemployment Rate by County",highFill = "blue",lowFill = "green")**



**# To get the unemployment rate of states Indiana and Florida**

**Unemployment\_indiana\_Florida<-get\_bls\_county(statename=c(”Indiana”,”Florida”))**

**>bls\_map\_county(map\_data = df, fill\_rate = "unemployed\_rate", labtitle = "Unemployment Rate in Indiana",highFill = "blue",lowFill = "green",stateName = c("Indiana",”Florida”))**

