```
getwd()

setwd("C:\\Users\\Swapnil bandekar\\Downloads\\Swapnil\\Data Analytics\\My
Work\\R\\Datasets")

getwd()

#### Base Plotting


### Using plot() to study 2 continuous variables

IR <- iris


View(IR)

dim(IR)

str(IR)


## Syntax

## plot(X = variable to be displayed on X-axis , Y = variable to be displayed on
Y-axis )

## Scatter Plot : 2 Continuous variables ( Decimal values )

plot( x = IR$Petal.Width , y = IR$Petal.Length )

plot( IR$Petal.Width , IR$Petal.Length )


# "x=' and "y=" are optional arguments

plot( y = IR$Petal.Length , x = IR$Petal.Width )


# "x=' and "y=" are arguments can be used in reverse order


## Adding xlabels , ylabels and Title

plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal
Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length"))

plot( x = IR$Petal.Width , y = IR$Petal.Length , main = "Petal Width v/s Petal
Length" ,
```

```
      xlab = "Petal Width" , ylab = "Petal Length")


# Main fn is for Title , xlab is for Labelling X-axis , ylab is for Labelling
Y-axis


### Variable Types

## 1. Discrete Variables

# Whole no or Count ( 1 , 2 , 3 , 4 , 5 )

# e.g No of customers , age of person etc



## 2. Continuous Variables

# Decimal values ( 0.1 , 0.2 , 0.3 ,0.4 )

# Revenue , Price , Income etc


## Adding colors

plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal
Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , col = "red")

# col = "red" => for red colour


## Adding different plotting symbols


plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal
Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , col = "red" , pch = 2)

# pch = 2 => for different symbol
# pch is used for plotting symbols (plotting characters)
# values ( 0 : 18)

plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal
Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , col = "red" , pch = 2)


## Adding More Options
```

```
plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal
Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , col = "red" , pch = 2 ,
type = "b")

plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal
Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , col = "red" , pch = 2 ,
type = "l")


# type => type of plot
# p => point , l => line , b => point and line



## Making a Conditional Bivariate Plot

## Seeing a relationship accross different species

plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal
Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , col = IR$Species)


# Will get colour according to the species group

as.numeric(IR$Species)

plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal
Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , col = IR$Species , pch =
as.numeric(IR$Species))


# pch can take only numeric values. Hence , "as.numeric" fn is used to get
different symbols for different species


plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal
Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , cex =
as.numeric(IR$Species))


# cex fn is used to have variable symbol size. Here , we have got different symbol
size for different species
```

```
plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , pch = as.numeric(IR$Species) ,
      cex = as.numeric(IR$Species) , col = IR$Species)


## Adding a Legend

plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , pch = as.numeric(IR$Species))

unique(IR$Species)

plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , pch = as.numeric(IR$Species))


legend(0.2 , 7 , c("setosa" , "versicolor" , "virginica") , pch = 1:3 )

plot( x = IR$Petal.Width , y = IR$Petal.Length , main = c("Petal Width v/s Petal Length") ,
      xlab = c("Petal Width") , ylab = c("Petal Length") , col = IR$Species , pch = as.numeric(IR$Species))

legend(0 , 7 , c("setosa" , "versicolor" , "virginica") , pch = 1:3 , col = 1:3 )


## Summary


# Bivariate Outliers

# Heteroskedasticity will be high

# This will impact the modelling output

# Heteroskedasticity : It refers to the circumstances in which variability of the variable is unequal
#                       across the range of values of a second variable that predicts it


### Univariate Analysis
```

```
## Histogram and boxplot are used to understand the distribution of the continuous
variable

## Box Plots


boxplot(IR$Petal.Length)

summary(IR$Petal.Length)


Box = boxplot(IR$Sepal.Width)

summary(IR$Sepal.Width)

Box$out


# Box$out => gives outliers


## Improving the Aesthetics of boxplot

boxplot(IR$Petal.Length , col = "red" , main = "Distribution of Petal Length")


class(IR$Species)

class(IR$Sepal.Width)


## In a Plot function, if x is factor variable and y is continuous variable then
the result will be boxplot

head(IR)

plot( x = IR$Species , y = IR$Sepal.Width , xlab = "Species" , main = "Sepal Length
accross Species" ,
      col = "red" , horizontal = TRUE)


plot( x = IR$Species , y = IR$Sepal.Width , xlab = "Species" , main = "Sepal Length
accross Species" , col = "red")


## Summary

# Box plot gives possible outliers

# Median is the better representative of the missing value as compared to Mean
```

```
# Outlier can be replaced with 95th percentile which is called as capping

# Setting a cap at 99% and replacing the outlier with 99%



### Histograms : Frequency Distribution

hist(IR$Sepal.Width , col = "Orange")

hist(IR$Sepal.Width , col = "Orange" , labels = TRUE)

hist(IR$Sepal.Width , col = "Orange" , freq = FALSE)

hist(IR$Sepal.Width , col = "Orange" , labels = TRUE , freq = FALSE)

# freq = FALSE => gives me the density


## Summary

# Histogram gives the frequency of the data

# Binning of data : Category wise ( like bucketing for e.g Age )

# For continuous variables , creating a bucketing to understand the data ( e.g
Income )


density(IR$Sepal.Width)

lines(density(IR$Sepal.Width))

# Density fn will give me the density summary

# But to add density plot to the existing histogram I have to use the lines fn

## Adding multiple plots in single plotting window

par('mar')
```

```
# par('mar') gives me the dimensions of the plotting window (default margin)

# I want to draw 4 different charts in the same plotting window

# I have to resize the margin

par( mfrow = c(1,2))

# par( mfrow = c(1,2)) => 1 Row , 2 Plots

# Arrange plots in Matrix format

plot( x = IR$Species , y = IR$Sepal.Width , xlab = "species" , main = "Sepal Width
accross Species" , col = "red")

plot( x = IR$Species , y = IR$Sepal.Length , xlab = "species" , main = "Sepal
Length accross Species" , col = "red")

par('mar')

Data = iris[ , 1:4]

list = names(Data)

par(mfrow = c(2,2))

par('mar')

par(mar = c(2,2,2,2))


### Market Mix Data ( Visualization using ggplot )


MIx <- read.csv("MMix.csv")

View(MIx)

dim(MIx)

head(MIx)

library(dplyr)

library(ggplot2)


summary(MIx$NewVolSales)
```

```
## Univariate Analysis

## Distribution of Sales


quantile( MIx$NewVolSales , p = c(1:100)/100 )

# Divides data into 100 parts

quantile( MIx$NewVolSales , p = 0.75 )

# Returns the value which lie at 75th percentile = 20942.75
#That means 75% of NewVolSales is less than or equal to 20942.75


## Histogram

plot1 = ggplot( MIx , aes( x = NewVolSales))

summary(MIx$NewVolSales)

# Checking the summary for min and Max value to create histogram accordingly
# Min.    1st Qu  Median  Mean    3rd Qu  Max.
# 17431   19049   19944   20171   20943   24944


plot1 + geom_histogram(breaks = seq(17000,25000 , by=1000) , col = "blue" , fill =
"green" )

# creating a histogram for range 17000 to 25000 with the interval of 1000

plot1 + geom_histogram(breaks = seq(17000,25000 , by=1000) , col = "blue" , fill =
"green" ) +
  labs ( title = "Distribution of sales")

# Adding title for the plot

plot1 + geom_histogram(breaks = seq(17000,25000 , by=1000) , col = "blue" , fill =
"green" ) +
  labs ( title = "Distribution of sales") + labs ( x = "Sales" , y = "Count")

# Adding x labels and y labels

plot1 + geom_histogram(breaks = seq(17000,25000 , by=1000) , col = "blue" , fill =
"green"  , alpha = 0.2) +
  labs ( title = "Distribution of sales") + labs ( x = "Sales" , y = "Count")

# alpha = 0.2 => Gives the transparency to the plot

plot1 + geom_histogram(aes(y = ..density..),breaks = seq(17000,25000 , by=1000) ,
```

```
col = "blue" , fill = "green"  , alpha = 0.2)

# Gives the histogram in terms of density
# aes(y = ..density..) => Syntax

plot1 + geom_histogram(aes(y = ..density..),breaks = seq(17000,25000 , by=1000) ,
col = "blue" , fill = "green"  , alpha = 0.2)+
  geom_density()

options(scipen = 999)

plot1 + geom_histogram(aes(y = ..density..),breaks = seq(17000,25000 , by=1000) ,
col = "blue" , fill = "green"  , alpha = 0.2)+
  geom_density()


# options(scipen = 999) => converts the density values from scientific to numeric



## Understanding "Website.campaign" wise distribution

unique(MIx$Website.Campaign)

plot1 + geom_histogram(aes( fill = Website.Campaign ),breaks = seq(17000,25000 ,
by=1000) , alpha =0.5 )


# Modifying the Position

plot1 + geom_histogram( aes( fill = Website.Campaign , colour = Website.Campaign) ,
position = "stack" , alpha = 0.2)



## Two Continuous variables ( Bivariate)

## We will use scatter plot

## Understanding the relationship between sales and price


P <- ggplot( MIx , aes( x = Base.Price , y = NewVolSales ))

P + geom_point()


# geom_point() => It is used to draw the scatter plot
```

```
## Understanding the conditional relationship based on Website.Campiagn

Q <- P + geom_point( aes( colour = Website.Campaign ))

Q

Q + labs( x = "Price" , y = "Voulme Sales" )

## Creating Grid

Q + facet_grid("Website.Campaign")

Q + facet_grid(Website.Campaign~.)

head(mtcars)

ggplot( mtcars , aes( x = mpg)) + geom_histogram() + facet_grid(vs~am)

# 2d heatmaps

P1 <- ggplot(MIx, aes(x = MIx$Base.Price, y=MIx$NewVolSales))

P1 + geom_bin2d()

## Box Plots

R <- ggplot( MIx ,aes( x = Website.Campaign , y = NewVolSales , fill =
Website.Campaign ) )

R

R + geom_boxplot()

## Density Plots

S <- ggplot( MIx , aes( x = NewVolSales ))

S

S + geom_density( aes( fill = Website.Campaign , colour = Website.Campaign ))
```

```
# fill is for filling the colour inside the plotting area
# colour is for coloring the outline of plotting area

S + geom_density( aes( fill = Website.Campaign , colour = Website.Campaign ) ,
alpha = 0.2 )



## ggplot is more suitable with data.frame


## finding average sales by Website.Campaign


MIx %>% group_by(Website.Campaign) %>% summarise(Avg_Sales = mean(NewVolSales)) %>%
as.data.frame() -> df

df

fd <- MIx %>% group_by(Website.Campaign) %>% summarise(Avg_Sales =
mean(NewVolSales)) %>% as.data.frame()

fd

# fd<- and -> df both can be used to store the values


aggregate( MIx$NewVolSales , by = list(MIx$Website.Campaign) , mean)


## Bar Chart

plot2 = ggplot( df , aes( x = Website.Campaign , y = Avg_Sales , fill =
Website.Campaign ))

plot2 + geom_bar( stat = "identity")

plot2 + geom_bar( stat = "identity") + geom_text( label = round(df$Avg_Sales))

plot2 + geom_bar( stat = "identity") + geom_text( label = round(df$Avg_Sales) ,
vjust = -1 )


# geom_text() is used to define the labels
# vjust is used to adjust the position of the labels


## Correlation : Relationship between continuous variables

# 3 continuous variables
```

```
names(MIx[c(1,2,4)])

cor(MIx[c(1,2,4)])

# Cor fn tells relationship strength between 3 variables

# Retlationship between 2 continuous variables can be explained using correlation
and scatter plot

# Using correlation and scatter plot in same plot

# If we compare more than 3 variable , the result will be difficult to understand

library(corrgram)


plot3 <- corrgram( MIx[c(1,2,4)] , lower.panel = panel.cor , upper.panel =
panel.pts )


# panel.cor => to find correlation matrix

# panel.pts => to draw scatter plot


## leaflet : TO visualize data on the map ( Geo Spatial Data )

install.packages('leaflet')

library(leaflet)

schools <- read.csv('schools.csv')

View(schools)

leaflet(schools) %>% addTiles() %>% addCircles(lng = ~long, lat = ~lat, popup =
~schoolname )


# addTiles() : to overlay the data on the map
# lng : longitude
# lat : latitude


## Extracting Lat-Long data from the shape File using rgdal() package

# most of the geospatial data is stored in shape file
# shapefile = data + location data
# Types of shape files : SpatialPointsDataFrame, SpatialPolygonsDataFrame,
```

```
SpatialLinesDataFrame, SpatialPixelsDataFrame , SpatialGridDataFrame etc.

install.packages('rgdal')

library(rgdal)
library(sp)

shape1 <- readOGR('Subway')

# readOGR command is used to extract the shape file

class(shape1)

# class of shape1 SpatialPointsDataFrame

shape1@data %>% head()

# to view data part of the shape file

shape1@coords %>% head()

# location info is in the form of Northing and Easting
# need to convert it into the decimal degrees form

shape1_1 <- spTransform(shape1,CRS('+init=epsg:4326'))

# CRS('+init=epsg:4326') : code to convert lat and long

leaflet(shape1) %>% addTiles() %>% addCircles()


shape2 <- readOGR('nyha_15a')

class(shape2)

# class of shape2 SpatialPolygonDataFrame

shape2@data %>% head()

class(shape2@polygons)

# Lat _long data is stored in polygons slot

shape2@polygons[1]


shape2_2 <- spTransform(shape2, CRS('+init=epsg:4326'))

leaflet(shape2_2) %>% addTiles() %>%addPolygons(weight = 0.9, popup =
```

```
~as.character(HealthArea))

# Add polygons : shape files contains polygon data
# weight : to change the width of the polygons
# popup : to display information ( should be in character form )


# to display the polygons like heatmaps we can use inbuild color functions and
create a palette

pal = colorBin(palette = 'Reds',bins = 4, domain = c(100,9300))

# colorbin : to produce the heatmaps
# palette : name of sthe palette
# Bin : no of required bins
# domain : min and max value for the bins

leaflet(shape2_2) %>% addTiles() %>%
  addPolygons(weight = 0.9, popup = ~as.character(HealthArea),fillColor =
~pal(HealthArea), fillOpacity = 1 ) %>%
  addLegend(pal = pal , values = ~HealthArea)

# fillOpacity : to display the heatmap prominently



### Case Study


hd <- read.csv('case_study_heart_disease_data_set.csv')
View(hd)
str(hd)


# Sex :- gender
#        1 : female , 0 : male
# cp :- chest pain type -
#       1: typical angina , 2: atypical angina , 3: non-anginal pain, 4:
asymptomatic
# trestbps :- resting blood pressure (in mm Hg on admission to the hospital)
# chol :- serum cholestoral in mg/dl
# fbs :- fasting blood sugar > 120 mg/dl
#         1 = yes; 0 = no
# restecg :- resting electrocardiographic results
#         0: normal , 1: having ST-T wave abnormality (T wave inversions and/or ST
elevation or depression of > 0.05 mV),
#         2: showing probable or definite left ventricular hypertrophy by Estes'
criteria
# thalach :- maximum heart rate achieved
# exang :- exercise induced angina
```

```
#         1 = yes; 0 = no
# oldpeak :- ST depression induced by exercise relative to rest
# slope :- the slope of the peak exercise ST segment
#         1: upsloping , 2: flat , 3: downsloping
# ca :- number of major vessels (0-3) colored by flourosopy
# thal :- heart condition
#         3 : normal, 6 : fixed defect , 7 : reversable defect
# num :-
# DV :- diagnosis of heart disease (angiographic disease status)
#       0 :  No presence (< 50% diameter narrowing) , 1 : Presence (> 50% diameter
narrowing)


### Histogram

# trestbps : wrt Sex, thal, cp and exang

ggplot(hd, aes(x =trestbps)) + geom_histogram(aes(fill = as.factor(DV)), position =
'dodge') +
  facet_grid(Sex+thal~cp+exang)



# chol : wrt Sex, thal, cp and exang


ggplot(hd, aes(x=chol)) + geom_histogram(aes(fill = as.factor(DV)), position =
'dodge') +
  facet_grid(Sex+thal~cp+exang)


# thalach : wrt Sex, thal, cp and exang


ggplot(hd, aes(x=thalach)) + geom_histogram(aes(fill = as.factor(DV)), position =
'dodge') +
  facet_grid(Sex+thal~cp+exang)


# oldpeak : wrt Sex, thal, cp and exang


ggplot(hd, aes(x=oldpeak)) + geom_histogram(aes(fill = as.factor(DV)), position =
'dodge') +
  facet_grid(Sex+thal~cp+exang)



#### Boxplot
```

```
# trestbps : wrt Sex and cp

ggplot(hd, aes(y = trestbps, x =as.factor(DV), fill = as.factor(DV))) +
geom_boxplot() +
  facet_grid(Sex~cp)


# thalach : wrt Sex and cp


ggplot(hd, aes(y =thalach, x =as.factor(DV), fill =as.factor(DV))) + geom_boxplot()
+
  facet_grid(Sex~cp)


# oldpeak : wrt Sex and cp


ggplot(hd, aes(y =oldpeak, x =as.factor(DV), fill =as.factor(DV))) + geom_boxplot()
+
  facet_grid(Sex~cp)



# chol : wrt Sex and cp


ggplot(hd, aes(y =chol, x =as.factor(DV), fill =as.factor(DV))) + geom_boxplot() +
  facet_grid(Sex~cp)




### Scatter Plot


# thalach vs age : wrt DV


hd1 <- ggplot(hd, aes(x = Age, y =thalach, color = as.factor(DV)))

hd1 + geom_point() + facet_grid(.~DV)


# thalach vs age : wrt thal , sex and fbs


hd1 + geom_point() + facet_grid(thal ~ Sex+fbs)
```

```
# chol vs age : wrt DV

hd2 <- ggplot(hd, aes(x =Age, y =chol, color =as.factor(DV))) + geom_point()

hd2 + facet_grid(.~DV)

# chol vs age : wrt thal , sex and fbs

hd2 + facet_grid(thal~Sex+fbs)

# thalach vc trestbps : wrt DV

hd3 <- ggplot(hd, aes(x =trestbps, y =thalach, color =as.factor(DV))) +
geom_point()

hd3 + facet_grid(.~DV)

# thalach vc trestbps : wrt thal , sex and fbs

hd3 + facet_grid(thal ~ Sex+fbs)

### Case Study (Practice Assignment)

data <- read.csv('dataF.csv')

View(data)

str(data)

dim(data)

unique(data$Industry)

library(ggplot2)
library(dplyr)
```

```
# Plot 1 : Technology


tech <- data %>% filter(Industry =='Technology')
View(tech)


t1 <- ggplot(tech , aes(x = tech$Company.Advertising, y =tech$Brand.Revenue,col =
tech$Brand, size = tech$Brand.Value))


t1 + geom_point() +
  labs( title = 'Technology', x = 'Company Advertising in Billions of $', y = '
Brand Revenue in Billions of $') +
  scale_size(range = c(2,4), breaks = c(30,60,100) , name = 'Brand Value $
(Billions)') +
  geom_text(aes(label = tech$Brand), hjust = 0.5, vjust = 1) + guides(col = F) +
theme_light() +
  theme(legend.key = element_rect(fill = 'light blue', color = 'black'), plot.title
= element_text(hjust = 0.5, size = 20, face = 'bold'),
        axis.title = element_text(face = 'bold'), legend.title = element_text(face
= 'bold'))




# Plot 2 : Luxury


lux <- data %>% filter(Industry == 'Luxury')


l1 <- ggplot(lux, aes(x = lux$Company.Advertising, y = lux$Brand.Revenue, col =
lux$Brand, size = lux$Brand.Value))

l1 + geom_point() + labs(title = 'Luxury', x = 'Company Advertising in Billions of
$', y = 'Brand Revenue in Billions of $ ') +
  scale_size(range = c(3,5), breaks = c(10,28.1), name = 'Brand Value $
(Billions)') + guides( col = F) + theme_light()+
  geom_text(aes(label = lux$Brand), hjust =0.5, vjust =1.5) +
  theme(legend.key = element_rect(fill = 'light blue', color = 'black'), plot.title
= element_text(hjust = 0.5, face = 'bold', size = 20),
        axis.title = element_text(face = 'bold'), legend.title = element_text(face
= 'bold')) +
  scale_x_continuous(breaks = seq(0,5,0.1))




# Plot 3 : Financial Services
```

```
fin <- data %>% filter(Industry =='Financial Services')
fin


f1 <- ggplot(fin, aes(x = fin$Company.Advertising, y = fin$Brand.Revenue, col =
fin$Brand, size = fin$Brand.Value))

f1 + geom_point() + theme_light() + scale_size(range = c(2,5), breaks =
c(7,12,23.4), name = 'Brand Values $ (Billions)') +
  labs(title = 'Financial Services', x = 'Company Advertising in Billions of $', y
= 'Brand Value in Billions of $') + guides(col = F) +
  geom_text(aes(label = fin$Brand), vjust = 1.5, hjust = 0.7 ) +
scale_x_continuous(breaks =  seq(0,5,0.1)) +
  scale_y_continuous(breaks = seq(0,100,10)) +
  theme(legend.key = element_rect(fill = 'light blue', color = 'black'),
legend.title = element_text(face = 'bold'),
        axis.title = element_text(face = 'bold'), plot.title = element_text(face =
'bold', hjust = 0.5, size = 20))




# Plot 4 : Automotive

unique(data$Industry)
auto <- data %>% filter(Industry == 'Automotive')

summary(auto$Brand.Value)


a1 <- ggplot(auto, aes(x=auto$Company.Advertising, y=auto$Brand.Revenue, color =
auto$Brand, size = auto$Brand.Value))


a1 + geom_point() + theme_light() + scale_size(range = c(2,6), breaks =
c(6.2,15,37), name = 'Brand Value $ (Billions)') +
  labs(title = 'Automotive', x ='Company Advertising in Billions of $', y = 'Brand
Value in Billions of $') + guides(color =F) +
  scale_x_continuous(breaks = seq(0,6,0.1)) + scale_y_continuous(breaks =
seq(0,200,10)) +
  theme(legend.key = element_rect(fill = 'light blue', color = 'black'),
legend.title = element_text(face = 'bold'),
        axis.title = element_text(face = 'bold'), plot.title = element_text(face =
'bold', hjust = 0.5, vjust = 1.5)) +
  geom_text(aes(label = auto$Brand), hjust = 0.5, vjust = 1.5)
```