```
# Load Libraries
library("ggplot2")
library("randomForest")
#marketing cost of 120 months
#Random Data Generation using Sample.int
a = sample.int(300,120)
# sort data using sort function to bring a trend inside the data
a = sort(a)
b = sample(60,120, replace = TRUE)
b = sort(b)
#Random uniform number generation using RUNIF
c = runif(120,2,12)
# adding some random noise to the data using jitter function
c = jitter(c)
c = sort(c)
d = runif(120,3,20)
d = sort(d)

#checking if the length is equal for all the data columns
length(a)
length(b)
length(c)
length(d)

# UNIVARIATE OUTLIER DETECTION ALGORITHMS
# IQR range detection using 25p and 75p method
IQR_method = function(x){
        Q1 = quantile(x, 0.25)
        Q3 = quantile(x, 0.75)
        IQR = Q3 - Q1
        lower = Q1 - 1.5*IQR
        higher = Q3 + 1.5*IQR
        return(c(lower, higher))}

print("Interquantile Ranges for 4 variables:")
IQR_method(a)
IQR_method(b)
IQR_method(c)
IQR_method(d)

# UNIVARIATE OUTLIER DETECTION USING MEAN ABSOLUTE DEVIATION (MAD)
MAD_method = function(x){
        med = median(x)
        MAD = mad(x)
        lower = med - 2*(MAD/0.6745)
        higher = med + 2*(MAD/0.6745)
        return(c(lower, higher))}
print("Permitted Ranged using MAD Method:")
MAD_method(a)
MAD_method(b)
MAD_method(c)
MAD_method(d)

#changing the column names to real names so that analysis becomes easy
col_names = c('Mkt_cost', 'Sales_cost', 'Sales', 'Profit')
df = data.frame(a,b,c,d)
```

```r
colnames(df) = col_names
df

# see summary statistics and some descriptive analysis of the dataset
summary(df)
boxplot(df$Mkt_cost, main = "Boxplot of Marketing Cost",
        ylab = "Values",
        col = "orange",
        border = "brown",
        horizontal = FALSE,
        notch = TRUE)

boxplot(df$Sales_cost, main = "Boxplot of Sales Cost",
        ylab = "Values",
        col = "green",
        border = "brown",
        horizontal = FALSE,
        notch = TRUE)

boxplot(df$Sales, main = "Boxplot of Sales Volume",
        ylab = "Values",
        col = "blue",
        border = "brown",
        horizontal = FALSE,
        notch = TRUE)

boxplot(df$Profit, main = "Distribution of Gross Profit",
        ylab = "Values",
        col = "orange",
        border = "brown",
        horizontal = FALSE,
        notch = TRUE)
plot(df$Mkt_cost, df$Profit, main="Profit vs Marketing Cost",
     xlab="Marketing Cost ", ylab="Profit ", pch=19, col='red')

plot(df$Sales_cost, df$Profit, main="Profit vs Sales Cost",
     xlab="Sales Cost ", ylab="Profit ", pch=19, col='green')

plot(df$Sales, df$Profit, main="Profit vs Sales Volume",
     xlab="Sales Volume", ylab="Profit ", pch=19, col='blue')

#Linear Modeling and analysis of the dataset

#Trying model-1 and check summary and if residuals is normally distributed
model1 = lm(Profit ~ Mkt_cost, data = df)
summary(model1)
# Check normality of Residuals
qqnorm(model1$residuals, pch = 1, frame = FALSE)
qqline(model1$residuals, col = "steelblue", lwd = 2)
ggplot(data.frame(value = model1$residuals), aes(x=value)) + geom_histogram()

#Trying model-2 and check summary and if residuals is normally distributed
model2 = lm(Profit ~ Sales_cost, data = df)
summary(model2)
# Check normality of Residuals
qqnorm(model2$residuals, pch = 1, frame = FALSE)
```

```r
qqline(model2$residuals, col = "steelblue", lwd = 2)
ggplot(data.frame(value = model2$residuals), aes(x=value)) + geom_histogram()

#Trying model-3 and check summary and if residuals is normally distributed
model3 = lm(Profit ~ Sales, data = df)
summary(model3)
# Check normality of Residuals
qqnorm(model3$residuals, pch = 1, frame = FALSE)
qqline(model3$residuals, col = "steelblue", lwd = 2)
ggplot(data.frame(value = model3$residuals), aes(x=value)) + geom_histogram()

#Trying model-4 and check summary and if residuals is normally distributed
model4 = lm(Profit ~ Mkt_cost+Sales_cost+Sales, data = df)
summary(model4)
# Check normality of Residuals
qqnorm(model4$residuals, pch = 1, frame = FALSE)
qqline(model4$residuals, col = "steelblue", lwd = 2)
qplot(model4$residuals, geom="histogram", main = "Residuals of model-4")

#Trying model-5 and check summary and if residuals is normally distributed
model5 = lm(Profit ~ Mkt_cost*Sales_cost*Sales, data = df)
summary(model5)
# Check normality of Residuals
qqnorm(model5$residuals, pch = 1, frame = FALSE)
qqline(model5$residuals, col = "steelblue", lwd = 2)
qplot(model5$residuals, geom="histogram", main = "Residuals of model-5")

#Some Simple Statistical Analysis on the generated dataset
# Data frame to Matrix
df_mat = data.matrix(df[,1:3])
df_mat

# Covariance and Corr Matrix formation
cov(df_mat)
cor(df_mat)

# PCA Analysis of the dataset
pca.df = prcomp(df_mat)
summary(pca.df)

#Linear modeling using first two PCs
pca_dataset = pca.df$x[,1:2]
pca_dataset = data.frame(pca_dataset)
pca_dataset
pca_dataset$Profit = df$Profit
pca_dataset
model6 = lm(Profit ~ PC1+PC2, data = pca_dataset)
summary(model6)
# Check normality of Residuals
qqnorm(model6$residuals, pch = 1, frame = FALSE)
qqline(model6$residuals, col = "steelblue", lwd = 2)
qplot(model6$residuals, geom="histogram", main = "Residuals of model-6")

# TRYING TO FIT SOME POLYNOMIAL or NON-LINEAR MODELS

model7 = lm(Profit ~ Mkt_cost + I(Mkt_cost^2), data = df)
```

```
summary(model7)
# Check normality of Residuals
qqnorm(model7$residuals, pch = 1, frame = FALSE)
qqline(model7$residuals, col = "steelblue", lwd = 2)
qplot(model7$residuals, geom="histogram", main = "Residuals of model-7", bins
= 30)

model8 = lm(Profit ~ poly(Mkt_cost, 5, raw = TRUE), data = df)
summary(model8)
# Check normality of Residuals
qqnorm(model8$residuals, pch = 1, frame = FALSE)
qqline(model8$residuals, col = "steelblue", lwd = 2)
qplot(model8$residuals, geom="histogram", main = "Residuals of model-8", bins
= 30)

model9 = lm(Profit ~ Mkt_cost + I(Mkt_cost^2) + Sales_cost + I(Sales_cost^2),
data = df)
summary(model9)
# Check normality of Residuals
qqnorm(model9$residuals, pch = 1, frame = FALSE)
qqline(model9$residuals, col = "steelblue", lwd = 2)
qplot(model9$residuals, geom="histogram", main = "Residuals of model-9", bins
= 30)

model10 = lm(Profit ~ Mkt_cost + I(Mkt_cost^2) + Sales_cost + I(Sales_cost^2)
+
                     Sales + I(Sales^2), data = df)
summary(model10)
# Check normality of Residuals
qqnorm(model10$residuals, pch = 1, frame = FALSE)
qqline(model10$residuals, col = "steelblue", lwd = 2)
qplot(model10$residuals, geom="histogram", main = "Residuals of model-10",
bins = 30)

model11 = lm(Profit ~ poly(Mkt_cost, 4, raw = TRUE) +
                     poly(Sales_cost, 4, raw = TRUE) , data = df)
summary(model11)
# Check normality of Residuals
qqnorm(model11$residuals, pch = 1, frame = FALSE)
qqline(model11$residuals, col = "steelblue", lwd = 2)
qplot(model11$residuals, geom="histogram", main = "Residuals of model-11",
bins = 30)
model12 = lm(Profit ~ poly(Mkt_cost, 5, raw = TRUE) +
                      poly(Sales_cost, 5, raw = TRUE) +
                      poly(Sales, 5, raw = TRUE), data = df)
summary(model12)
qplot(model12$residuals, geom="histogram", main = "Residuals of model-12",
bins = 30)
# Check normality of Residuals
qqnorm(model12$residuals, pch = 1, frame = FALSE)
qqline(model12$residuals, col = "steelblue", lwd = 2)

# TRYING TREE BASED MODELS NOW (Random Forest Algorithm)
model13 = randomForest(Profit ~ Mkt_cost+Sales_cost+Sales, data = df,
                       mtry = 2, importance = TRUE, na.action = na.omit)
summary(model13)
```

```
print(model13)
plot(model13)
importance(model13)

#ANOVA of all models we have created till now
anova(model1, model2, model3, model4,
      model5, model6, model7, model8,
      model9, model10, model11, model12)

print("Thank you!")
```