```
#RANDOM FOREST
library(randomForest)
library(caTools)
library(readxl)

dataset<-read_excel("C:/Users/Sumanth B S/Desktop/randomforest.xlsx")
head(dataset)
dim(dataset)

split<-sample.split(dataset$species,SplitRatio=0.7)

train<- subset(dataset,split==TRUE)
test<-subset(dataset,split==FALSE)

input<-train[,1:4]
target<-train$species
#target <- as.factor(target)
model<-randomForest(input,target,ntree=20)

pred <- predict(model,test)
pred

cm<-table(pred,test$species)
cm

plot(model)
varImpPlot(model)
importance(model)
summary(model)
```

```
# Install and load the required packages
install.packages(c("arules", "arulesViz"))
library(arules)
library(arulesViz)

# Read data from CSV
data <- read.csv("C:/Users/Sumanth B S/Desktop/association.csv", header = TRUE, stringsAsFactors = FALSE)

# Convert the data to transactions format
transactions <- as(data, "transactions")

# Explore the transactions
summary(transactions)

# Perform association analysis using Apriori
rules <- apriori(transactions, parameter = list(support = 0.005, confidence = 0.2))

# Display the rules
inspect(rules)

itemFrequencyPlot(transactions,topN=5)

# Visualize association rules
plot(rules, method = "graph")
```

```
# Hierarchical Clustering (hclust)
library(dplyr)
library(ggplot2)

# Load and explore mtcars dataset
head(mtcars)
colnames(mtcars)
dim(mtcars)

# Step 1: Create a distance matrix
dist_matrix <- dist(mtcars, method = 'euclidean')

# Step 2: Build the hierarchical clustering model
hclust_model <- hclust(dist_matrix, method = 'average')
summary(hclust_model)
plot(hclust_model)

# Step 3: Group into clusters (let k be the number of clusters)
rect.hclust(hclust_model, k = 3)
fit <- cutree(hclust_model, k = 3)
print(fit)
```

```
# K-means Clustering (kmeans)
library(cluster)
library(ClusterR)  # Capital 'C'

# Load and explore iris dataset
head(iris)
colnames(iris)
dim(iris)

# Separate attributes and target
attributes <- iris[, 1:4]
target <- iris[, 5]

# Build the k-means clustering model
kmeans_model <- kmeans(attributes, centers = 3, nstart = 20)
summary(kmeans_model)
print(kmeans_model$cluster)

# Create a confusion matrix
cm <- table(target, kmeans_model$cluster)
cm

# Plot clusters
clusplot(iris, kmeans_model$cluster)
```

```
#linear regression
install.packages("Metrics")
library(Metrics)
library(ggplot2)
library(caTools)
library(Metrics)

dataset<-read_excel("C:/Users/Sumanth B S/Desktop/linearregression.xlsx")
dim(dataset)
head(dataset)
colnames(dataset)
ggplot(dataset)+geom_point(aes(x=temp,y=sales))
split <- sample.split(dataset$sales,SplitRatio=0.7)
training<-subset(dataset,split==TRUE)
test <- subset(dataset,split==FALSE)
test

model = lm(formula=sales~temp,dataset)
coef(model)

predns <- predict(model,test)
predns

rmse(test$sales,predns)

ggplot()+geom_point(aes(x=training$temp,y=training$sales))+
  geom_line(aes(x=training$temp,y=predict(model,training)))
```

```
#logistic regression
#for mtcars dataset use dplyr
library(dplyr)
library(ROCR)#for ROC AUC
library(caTools)#for split

head(mtcars)
split<- sample.split(mtcars$vs,SplitRatio = 0.7)#vs is target variable
training<-subset(mtcars,split==TRUE)
testing<-subset(mtcars,split==FALSE)

dim(mtcars)
colnames(mtcars)

#model
model<-glm(
  formula=vs~wt+disp,data = mtcars,
  family =  binomial(link="logit")
  )#input variables are wt and disp

summary(model)
coef(model)

predns <- predict(model,testing)
predns #you get probs

predns <- ifelse(predns>0.5,1,0)
predns

table(predns,testing$vs)
ROCPred <- prediction(predns, testing$vs)
```

```
ROCPer <- performance(ROCPred, measure = "tpr",

                      x.measure = "fpr")



auc <- performance(ROCPred, measure = "auc")

auc <- auc@y.values[[1]]
plot(ROCPer)
```