PCA

```
data("iris")
head(iris)
scaled_data <- scale(iris[, 1:4])
pca_result <- princomp(scaled_data)
summary(pca_result)
plot(pca_result, type="lines", main= "Scree Plot")
biplot(pca_result,scale=0)
cumulative_var <- cumsum(pca_result$sdev^2/sum(pca_result$sdev^2))
num_components <- which(cumulative_var >= 0.95)[1]
num_components
reduced_data <- predict(pca_result,newdata=scaled_data)[,1:num_components]
plot(reduced_data[,1],reduced_data[,2],col=iris$Species,main = "PCA OF IRIS DATASET")
legend("topright",legend=levels(iris$Species),col=1:3,pch=1)
```

Clustering

```
data("iris")
head(iris)
names(iris)
newdata = subset(iris,select=c(-Species))
newdata
resl = kmeans(newdata,3)
resl
res2 = sapply(1:15,function(k){kmeans(newdata,k)$tot.withinss})
res2
plot(1:15,res2,type="b",pch="19",frame=FALSE,xlab="no of clusters", ylab="Total clusters")
library(cluster)
clusplot(newdata,resl$cluster,color=TRUE, shade = TRUE, labels = 2, lines = 0)
```

AOV

```
a1 <- c(10,12,24,33,21)
a2 <- c(11,14,25,23,27)
a3 <- c(15,82,64,53,81)
data <- data.frame(cbind(a1,a2,a3))
data
s<-stack(data)
s
res <- aov(values~ind,data=s)
res
summary(res)
```

DT

```
data('mtcars')
head(mtcars)
dt = ctree(mpg~cyl+am,data=mtcars)
dt
plot(dt)
```

HT

```
library(tidyverse)
?airquality
View(airquality)
qplot(airquality$Wind, geom = "histogram")
t.test(airquality$Wind, mu=9)
qplot(airquality$Solar.R, geom= "histogram")
t.test(airquality$Solar.R, mu =175)
```

TSF

```
install.packages('forecast')
library('forecast')
class(AirPassengers)
my_colors <- rainbow(12)
boxplot(split(AirPassengers, cycle(AirPassengers)),
      xlab = "Month", ylab= "Number of Passangers",
      col = my_colors,
      border = "black",
      main = "Monthly Air Passanger Counts by Month",
      names = month.abb,
      outline = FALSE)
model <- auto.arima(AirPassengers)
summary(model)
f<-forecast(model,level=c(95), h=10*12)
plot(f)
```

SLR MLR

```
library(car)
summary(mtcars)
model = lm(mpg~wt, data=mtcars)
summary(model)
plot(mtcars$wt,mtcars$mpg)
abline(model)
residualPlots(model)
model = lm(mpg~wt+hp, data =mtcars)
residualPlots(model)
summary(model)
```

Recursive Partioning

```
library(party)
dt=ctree(mpg~cyl+am,data=mtcars)
dt
index=sample(1:nrow(mtcars),0.75*nrow(mtcars))
train=mtcars[index,]
test=readingSkills[-index]
train
test=readingSkills[-index]
train
train=mtcars[-index]
train
dt=ctree(mpg~hp+wt,data=mtcars)
plot(dt)
```