data\_iris<-iris[1:4]

#calculating the covariance martix

Cov\_data<-cov(data\_iris)

#find out the eigenvector and eighenvalue using the covarient

Eigen\_data<- eigen(Cov\_data)

#using the inbuild function PCA calculation

PCA\_data<- princomp(data\_iris,cor="False")

Eigen\_data$values

PCA\_data$sdev^2

#there is slight difference due to squring in PCA\_data ....we can also compare the eighenvector of both models

PCA\_data$loading[,1:4]

Eigen\_data$vectors

summary(PCA\_data)

biplot(PCA\_data)

screeplot(PCA\_data,type="lines")

#we calcute the frst principle com

model2=PCA\_data$loading[,1]

#for the second model, we need to calculate score by multiplying our loading with the data

model2\_scores<-as.matrix(data\_iris)%\*%model2

#loading libraries for naiveBayes model2

library(class)

install.packages("e1071")

library(e1071)

#fitting the first model over the entire data\_iris

mod1<-naiveBayes(iris[,1:4],iris[,5])

#fitting the second model using the first principal component

mod2<-naiveBayes(model2\_scores,iris[,5])

#accuracy for the first model2\_scores

table(predict(mod1,iris[,1:4]),iris[,5])

#accuracy fir the second model

table(predict(mod2,model2\_scores,iris[,5])

**#practicle of clustering**

**install.packages("ggplot2")**

**library(ggplot2)**

**scatter<-ggplot(data=iris,aes(x=Sepal.Length,y=Sepal.Width))**

**scatter+geom\_point(aes(color=Species,shape=Species))+theme\_bw()+xlab("sepal Lenght")+ylab("Sepal Width")+ ggtitle("sepal length-Width")**

**ggplot(data=iris,aes(Sepal.Length,fill=Species))+ theme\_bw()+ geom\_density(alpha=0.25)+labs(x="Sepal.Length",title="species vs sepal Length")**

**vol<-ggplot(data=iris,aes(x=Sepal.Length))**

**vol+stat\_density(aes(ymax=..density..,ymin=-..density..,fill=Species,color=Species),geom="ribbon",position="identity")+facet\_grid(.~Species)+coord\_flip()+theme\_bw()+labs(x="Sepal Length",title="Species vs sepal length")**

**vol<-ggplot(data=iris,aes(x=Sepal.Width))**

**#clustering method1**

**irisData<-iris[,1:4]**

**totalSS-c()**

**#for method1 kmean clustering for 15 time in a loop**

**for(i in 1:15)**

**{**

**clusterIRIS<-kmeans(irisData,centers=i)**

**totalwSS[i]<-clusterIRIS$tot.withinss**

**}**

**#use plot function to plot value of tot\_wss against no-of-coure**

**plot(x=1:15,y=totalwSS,type="b",xlab="Number of Clusters",ylab="Within groups sum-of-squares")**

**#method2**

**install.packages("NbClust")**

**library(NbClust)**

**#set margin as:**

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

**nb<-NbClust(irisData,method="kmeans")**

**hist(nb$Best.nc[1,], breaks=15,main="Histogram for number of cluster")**

**practical:5**

**#load Data Airpassenger**

**data("AirPassengers")**

**#finding the class name**

**class(AirPassengers)**

**#Data In time series format**

**#start of time series**

**start(AirPassengers)**

**Exp:05**

**#start of time series**

**end(AirPassengers)**

**frequency(AirPassengers)**

**#The cycle of this time series is 12 month in a year**

**summary(AirPassengers)**

**#the number of passenger are distributed across the scetrum**

**plot(AirPassengers)**

**#this will plot the time series**

**abline(reg=lm(AirPassengers~time(AirPassengers)))**

**#This will print the cycle across years**

**cycle(AirPassengers)**

**#this wil aggreate the cycles and display a year on year trend**

**plot(aggregate(AirPassengers,FUN=mean))**

**#box plot across month will give us a sense on seasonal effect**

**boxplot(AirPassengers~cycle(AirPassengers))**

**#following are the Acf plots for the series:**

**acf(log(AirPassengers))**

**acf(diff(log(AirPassengers)))**

**(fit<- arima(log(AirPassengers), c(0,1,1),seasonal=list(order=c(0,1,1), period=12)))**

**pred<-predict(fit,n.ahead=10\*12)**

**ts.plot(AirPassengers,2.718^pred$pred, log="y", lty=c(1,3))**