

Experiment 1

#expt 1

#Basic

```
x = c(1,6,2)
```

```
x
```

```
y = c(1,4,3)
```

```
length(x)
```

```
length(y)
```

```
x+y
```

```
ls()
```

```
rm(x,y)
```

```
ls()
```

```
rm(list=ls())
```

#matrix

```
x=matrix(data=c(1,2,3,4), nrow=2, ncol=2)
```

```
x
```

```
x=matrix(c(1,2,3,4) ,2,2)
```

```
matrix(c(1,2,3,4) ,2,2,byrow=TRUE)
```

```
sqrt(x)
```

```
x^2
```

```
x=rnorm(50)
```

```
y=x+rnorm(50,mean=50,sd=.1)
```

```
cor(x,y)
```

```
set.seed(1303)
```

```
rnorm(50)
```

```
set.seed(3)
```

```
y=rnorm(100)
```

```
mean(y)
```

```
var(y)
```

```
sqrt(var(y))
```

```
sd(y)
```

#graphics

```
y=x
f=outer(x,y,function(x,y)cos(y)/(1+x^2))
contour(x,y,f)
z=(f-t(f))/2
contour(x,y,z,nlevels=45)
image(x,y,z)
persp(x,y,z,theta=30,phi=70)
```

```
#loading data
```

```
Auto=read.csv("auto-mpg.csv",header=T,na.strings = "?")
str(Auto)
Auto[1:4,]
Auto=na.omit(Auto)
dim(Auto)
names(Auto)
attach(Auto)
plot(cylinders,mpg)
cylinders=as.factor(cylinders)
plot(cylinders,mpg,col="cyan",varwidth=T,xlab="Cylinders",ylab="mpg",main="plot")
hist(mpg,col=6,breaks = 15)
pairs(~mpg + horsepower + weight + year + name ,Auto)
plot(horsepower,mpg)
identify(horsepower,mpg,name)
summary(mpg)
```

```
#indexing
```

```
a=matrix(1:16,4,4,byrow = TRUE)
a
a[-c(1,3),]
a[-c(1,3),-c(1,3)]
dim(a)
```

Experiment 2

```
library(ISLR)
library(MASS)
data("Auto")
head(Auto)
lm.fit<-lm(mpg~horsepower,data=Auto)
summary(lm.fit)

predict(lm.fit,data.frame(horsepower=c(98)),interval="prediction")
predict(lm.fit,data.frame(horsepower=c(98)),interval="confidence")
plot(Auto$horsepower,Auto$mpg)

abline(lm.fit,col="blue",lwd=5)

which.max(hatvalues(lm.fit))

par(mfrow=c(2,2))

plot(lm.fit)
```

Experiment 3

```
pairs(Auto)
```

```
Auto$name<-NULL  
cor(Auto,method = c("pearson"))
```

```
lm.fit<-lm(mpg~.,data=Auto)  
summary(lm.fit)
```

```
which.max(hatvalues(lm.fit))
```

```
par(mfrows=c(2,2))  
plot(lm.fit)
```

```
lm.fit = lm(mpg ~.+displacement:weight, data = Auto)  
summary(lm.fit)
```

```
lm.fit = lm(mpg  
~.+l((displacement)^2)+log(displacement)+displacement:weight,  
data=Auto)  
summary(lm.fit)
```

4th experiment

#EXPT 4 KNN Breast cancer

```
library(readr)
library(class)
library(gmodels)
library(dplyr)

bcd<- read.csv("WDBC-data.csv")
str(bcd)
bcd<- dplyr::select(bcd,-id,-X)
table(bcd$diagnosis)
round(prop.table(table(bcd$diagnosis)) * 100, digits = 1)
sum(is.na(bcd))
head(bcd)
cor(select(bcd,-diagnosis))
summary(bcd)
#Data Transformation
normalize <- function(x) { return ((x - min(x)) / (max(x) - min(x))) }
new_bcd <- as.data.frame(lapply(select(bcd,-diagnosis), normalize))
summary(select(new_bcd,radius_mean,smoothness_mean))
#Data Preparation
bcd_train <- new_bcd[1:429,]
bcd_test <- new_bcd[430:569,]

bcd_train_labels <- bcd[1:429, 1]
bcd_test_labels <- bcd[430:569, 1]

#Train Model
bcd_test_pred <- knn(train = bcd_train, test = bcd_test, cl = bcd_train_labels,
                     k = 20)
cm = CrossTable(x = bcd_test_labels, y = bcd_test_pred, prop.chisq = FALSE)
cm
```

5th experiment

#EXPT 5

Load required libraries

library(quantmod)

library(MASS)

library(e1071) library(caret)

library(ggplot2)

library(pROC)

Fetch stock market data symbols<-

c("AAPL","MSFT") start_date<-"2020-01-01"

end_date<-"2023-01-01"

getSymbols(symbols,from=start_date,to=end_date)

stock_data<-merge(Ad(AAPL),Ad(MSFT))

colnames(stock_data)<-symbols

**# Calculate daily returns stock_returns<-diff(log(stock_data)) # Create labels based
#on return threshold**

return_threshold<-0.01

stock_labels<-

ifelse(apply(abs(stock_returns),1,max)>return_threshold,"High","Low")

Combine returns and labels into a data frame stock_df<-

data.frame(stock_returns,stock_labels)

Split data into training and testing sets set.seed(123)

train_indices<-

sample(1:nrow(stock_df),0.75*nrow(stock_df)) train_data<-

stock_df[train_indices,] test_data<-stock_df[-train_indices,]

any_na<-

apply(test_data,2,function(column) any(is.na(column))) print(any_na)

test_data<-test_data[complete.cases(test_data),]

lda_model<-lda(stock_labels ~ .,data=train_data)

```
lda_predictions<-predict(lda_model,newdata=test_data)$class
```

```
qda_model<-qda(stock_labels~ .,data=train_data)
```

```
qda_predictions<- predict(qda_model,newdata=test_data)$class
```

```
nb_model<-naiveBayes(stock_labels ~ .,data=train_data)
```

```
nb_predictions<-predict(nb_model,newdata=test_data)
```

```
lda_accuracy<-mean(lda_predictions==test_data$stock_labels)
```

```
qda_accuracy<-mean(qda_predictions==test_data$stock_labels)
```

```
nb_accuracy<-mean(nb_predictions==test_data$stock_labels)
```

```
# Print the accuracy results
```

```
cat("LDA Accuracy:",lda_accuracy,"\n")
```

```
cat("QDA Accuracy:",qda_accuracy,"\n")
```

```
cat("Naive Bayes Accuracy:",nb_accuracy,"\n")
```

```
test_data$stock_labels=as.factor(test_data$stock_labels)
```

```
lda_cm<- confusionMatrix(lda_predictions,test_data$stock_labels)
```

```
qda_cm<-confusionMatrix(qda_predictions,test_data$stock_labels)
```

```
nb_cm<-confusionMatrix(nb_predictions,test_data$stock_labels)
```

```
print(lda_cm) print(qda_cm) print(nb_cm)
```

```
# Create data frame for accuracy comparison
```

```
accuracy_df<-data.frame(Classifier=c("LDA","QDA","Naïve  
Bayes"),Accuracy=c(lda_accuracy,qda_accuracy,nb_accuracy))
```

```
# Create accuracy plot
```

```
accuracy_plot<-ggplot(accuracy_df,aes(x=Classifier,y=Accuracy,fill=Classifier))+  
  geom_bar(stat="identity",position="dodge")+  
  labs(y="Accuracy",title="Classifier Comparison")+ theme_minimal()+  
  theme(legend.position="bottom")  
print(accuracy_plot)
```

```

roc(test_data$stock_labels,as.numeric(lda_predictions=="High"))
roc_qda<-roc(test_data$stock_labels,as.numeric(qda_predictions=="High"))
roc_nb<-roc(test_data$stock_labels,as.numeric(nb_predictions=="High"))

min_length<-min(length(roc_lda$sensitivities),length(roc_qda$sensitivities),
                length(roc_nb$sensitivities))

roc_data <- data.frame(
  FPR = c(roc_lda$specificities[1:min_length], roc_qda$specificities[1:min_length],
          roc_nb$specificities[1:min_length]),
  TPR = c(roc_lda$sensitivities[1:min_length], roc_qda$sensitivities[1:min_length],
          roc_nb$sensitivities[1:min_length]),
  Classifier = rep(c("LDA", "QDA", "Naive Bayes"), each = min_length))

roc_combined_plot <- ggplot(roc_data, aes(x = FPR, y = TPR, color = Classifier,
linetype =
                                Classifier)) +
  geom_line(linewidth = 1) +
  labs(x = "False Positive Rate (1 - Specificity)", y = "True Positive Rate
(Sensitivity)", title =
  "ROC Curves - Classifier Comparison") +
  scale_color_manual(values = c("blue", "green", "red")) +
  scale_linetype_manual(values = c("solid", "dashed", "dotted"))
+ theme_minimal() + theme(legend.position = "bottom")
print(roc_combined_plot)

```


6th experiment

```
library(ggplot2)
library(tidyverse)
df=data.frame(sugar_content=c(10.9,10.9,10.6,10,8,8.2,8.6,10.9,10.7,8,7.7,7.8,8.4,11.
5),samples=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14))
colnames(df)          # Print variable names
plot_df <- ggplot(data = df, aes(x = sugar_content, y = 0)) +geom_point() +
  geom_text(aes(label = sugar_content), size = 2.5, vjust = 2, hjust =0.5)
plot_df
mm_separator <- (8.9+10)/2
separator <- data.frame(sep = mm_separator)
plot_=plot_df
#add separator to sugar content scatterplot
plot_sep <- plot_ + geom_point(data = separator, aes(x = mm_separator, y = 0),
color ="blue")
plot_sep              #display plot

set.seed(42)          #set seed
#set number of data points.
n <- 600
df <- data.frame(x1 = runif(n),x2 = runif(n))
df$y <- factor(ifelse(df$x2-1.4*df$x1 < 0, -1, 1),
  levels = c(-1, 1))
#set margin
delta <- 0.07
# retain only those points that lie outside the margin
df1 <- df[abs(1.4*df$x1 - df$x2) > delta, ]
#build plot
plot_margins <- ggplot(data = df1, aes(x = x1, y = x2, color = y)) + geom_point() +
  scale_color_manual(values = c("red", "blue")) +
  geom_abline(slope = 1.4, intercept = 0)+
  geom_abline(slope = 1.4, intercept = delta, linetype = "dashed") +
  geom_abline(slope = 1.4, intercept = -delta, linetype = "dashed")
#display plot
plot_margins
```

Experiment 7

```
library(ISLR)
library(e1071)
names(Khan)
dim(Khan$xtrain )
dim(Khan$xtest )
length(Khan$ytrain )
length(Khan$ytest )
table(Khan$ytrain)
table(Khan$ytest)
dat=data.frame(x=Khan$xtrain, y=as.factor(Khan$ytrain))
out=svm(y ~ ., data=dat, kernel ="linear",cost=10)
summary(out)
table(out$fitted, dat$y)
dat.te=data.frame(x=Khan$xtest, y=as.factor(Khan$ytest))
pred.te=predict(out, newdata=dat.te)
table(pred.te, dat.te$y)
```

Experiment 8

```
library(ISLR)
USArrests[1:10,]
prcomp(USArrests)
plot(prcomp(USArrests))
summary(prcomp(USArrests))
plot(prcomp(USArrests,scale=T))
summary(prcomp(USArrests,scale=T))
prcomp(USArrests,scale=T)
plot(prcomp(USArrests,scale=T)$x[,1:2])
plot(prcomp(USArrests,scale=T)$x[,1:2],type="n")
text(prcomp(USArrests,scale=T)$x[,1:2],rownames(USArrests))
biplot(prcomp(USArrests,scale=T))
```

Experiment 9

```
library(ggplot2)
head(iris)
str(iris)
ggplot(iris, aes(Petal.Length, Petal.Width)) + geom_point(aes(col=Species), size=4)
set.seed(100)
irisCluster <- kmeans(iris[,1:4], center=3, nstart=20)
irisCluster
tot.withinss <- vector(mode="character", length=10)
for (i in 1:10){
  irisCluster <- kmeans(iris[,1:4], center=i, nstart=20)
  tot.withinss[i] <- irisCluster$tot.withinss
}
plot(1:10, tot.withinss, type="b", pch=19)
irisCluster <- kmeans(iris[,1:4], center=3, nstart=20)
table(irisCluster$cluster, iris$Species)
library(cluster)
clusplot(iris, irisCluster$cluster, color=T, shade=T, labels=0, lines=0)
```

Experiment 10

```
library(ISLR)
nci.labs=NCI60$labs
nci.data=NCI60$data
```

```
dim(nci.data)
```

```
nci.labs[1:4]
```

```
table(nci.labs)
```

```
pr.out=prcomp(nci.data , scale = TRUE)
```

```
Cols=function(vec){
  cols=rainbow(length(unique(vec)))
  return(cols[as.numeric(as.factor(vec))])
}
```

```
par(mfrow=c(1,2))
plot(pr.out$x[,1:2],col=Cols(nci.labs),pch=19,xlab="Z1",ylab="z2")
plot(pr.out$x[,c(1,3)],col=Cols(nci.labs),pch=19,xlab="Z1",ylab="Z3")
```

```
summary(pr.out)
```

```
plot(pr.out)
```

```
pve=100*pr.out$sdev^2/sum(pr.out$sdev^2)
plot(pve,type="o",xlab="Principal component",ylab="PVE",col="blue")
plot(cumsum(pve),type="o",xlab="Principal Component",ylab="cumulative
PVE",col="brown")
```

```
sd.data=scale(nci.data)
```

```
par(mfrow=c(1,3))
data.dist=dist(sd.data)
plot(hclust(data.dist), labels=nci.labs , main="Complete Linkage", xlab="", sub="",
ylab = "")
plot(hclust(data.dist,method="average"),labels=nci.labs,main="Average",xlab="",y
lab = "",sub="")
plot(hclust(data.dist,method="single"),labels=nci.labs,main="single",xlab="",ylab
= "",sub="")
```

```
hc.out=hclust(dist(sd.data))
hc.clusters=cutree(hc.out,4)
table(hc.clusters ,nci.labs)
```

```
par(mfrow=c(1,1))
plot(hc.out , labels=nci.labs)
```

```
abline(h=139,col="red")
```

```
hc.out
```

```
set.seed(2)
km.out=kmeans(sd.data,4,nstart = 20)
kmclusters=km.out$cluster
table(kmclusters,hc.clusters)
```

```
hc.out=hclust(dist(pr.out$x[,1:5]))
plot(hc.out,labels=nci.labs,main="hdsfko")
```

```
table(cutree(hc.out,4),nci.labs)
```