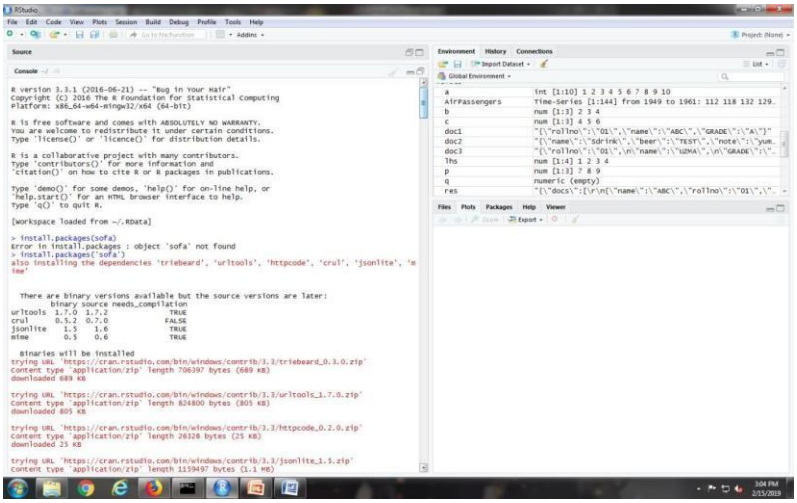
PRACTICAL NO:-1

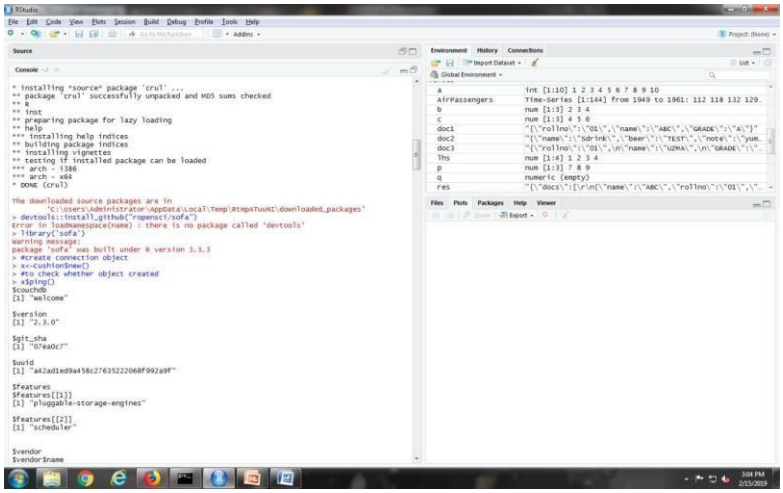
Couch dB database -
-- rscript Install
couch dB first

Rscript code

```
install.packages('sofa')
#devtools::install_github("ropensci/so
fa") library('sofa')
#create connection
object x<-
Cushion$new()
#to check whether object
createdx$ping()
#create database ty
db_create(x,dbnam
e = 'ty') db_list(x)
#create json doc
doc1<-'{"rollno":"01","name":"ABC","GRADE":"A"}'
doc_create(x,doc1,dbname =
"ty",docid = "a_1") doc2<-
'{"rollno":"02","name":"PQR","GRA
DE":"A"}'
doc_create(x,doc2,dbname = "ty",docid = "a_2")
doc3<-'{"rollno":"03","name":"xyz","GRADE":"B","REMARK":"PASS"}'
doc_create(x,doc3,dbname = "ty",docid = "a_3")

#CHANGES FEED
db_changes(x,"ty")

#search for id > null so all
docs will display
db_query(x,dbname = "ty",
    selector = list('_id'=list('$gt'=NULL)))$docs

#search for students with grade is A
db_query(x,dbname = "ty",selector = list(GRADE="A"))$docs

#search for students with remark =pass
db_query(x,dbname = "ty",selector = list(REMARK="PASS"))$docs

#return only certain fields where rollno>2
db_query(x,dbname = "ty",selector = list(rollno=list('$gt'='02')),fields=c("name","GRADE"))$docs

#convert the result of a query into a data frame using
jsonlitelibrary("jsonlite")
res<-db_query(x,dbname = "ty",selector =
list('_id'=list('$gt'=NULL)),fields=c("name","rollno","GRADE","REMARK"),as="json")
```

```
#display json doc
fromJSON(res)$docs
#doc_delete(cushion,dbname,d
ocid) doc_delete(x,dbname =
"ty",docid = "a_2")
doc_get(x,dbname = "ty",docid
= "a_2")

doc2<-'{"name":"Sdrink","beer":"TEST","note":"yummy","note2":"yay"}'
doc_update(x,dbname = "ty",doc=doc2,docid="a_3",rev = "3-b1fb56db955b142c6efd3b3c52fe9e1b")

doc3<-
'{"rollno":"01",
"name":"UZM
A",
"GRADE":"A"}'

doc_update(x,dbname = "ty",doc=doc3,docid = "a_1",rev = "1-be7c98bddf8ea7c46f4f401ff387593d")
```



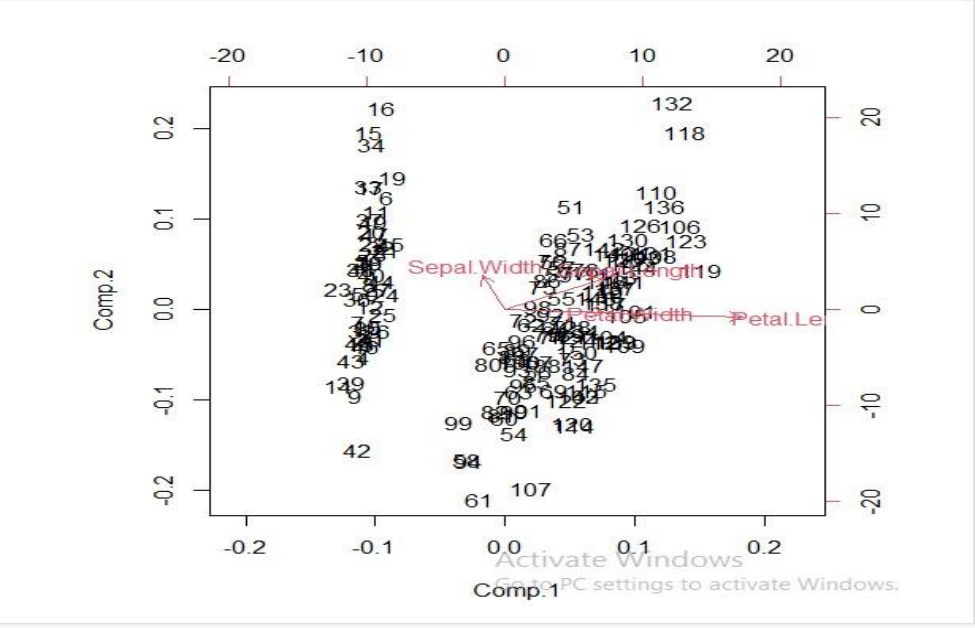


PRACTICAL NO:- 3

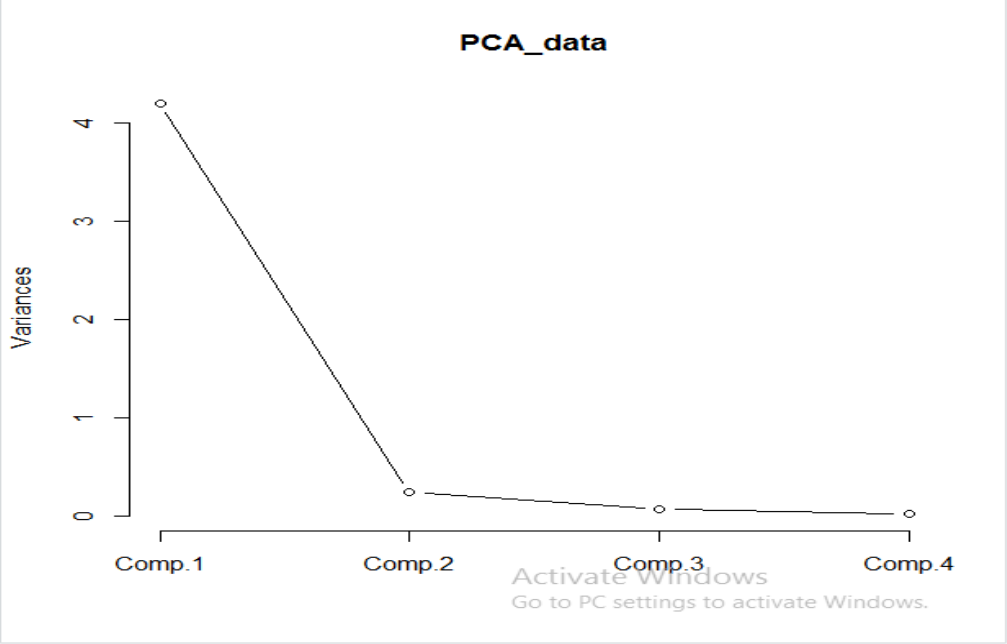
Aim:- Practical of Principal Component Analysis(PCA)

Code:-

```
data_iris<-iris[1:4]
Cov_data<-cov(data_iris)
Eigen_data<-eigen(Cov_data)
PCA_data<-princomp(data_iris,cor="False")
Eigen_data$values
PCA_data$dev^2
PCA_data$loadings[,1:4]
Eigen_data$vectors
summary(PCA_data)
biplot(PCA_data)
```




```
screeplot(PCA_data,type="lines")
```



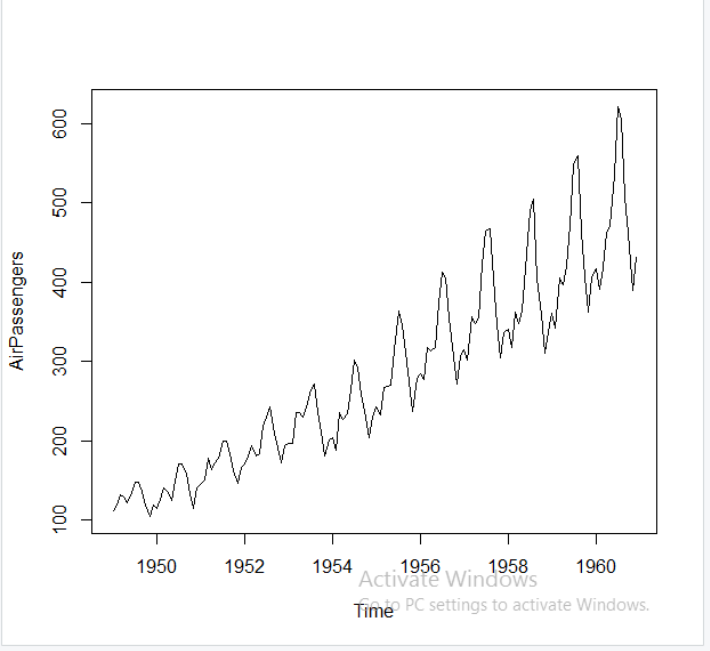

```
model2=PCA_data$loadings[,1]
model2_scores<-as.matrix(data_iris)%*%model2
library(class)
install.packages("e1071")
library(e1071)
mod1<-naiveBayes(iris[,1:4],iris[,5])
mod2<-naiveBayes(model2_scores,iris[,5])
table(predict(mod1,iris[,1:4]),iris[,5])
table(predict(mod1,iris[,1:4]),iris[,5])
table(predict(mod2,model2_scores),iris[,5])
```

PRACTICAL NO:- 5

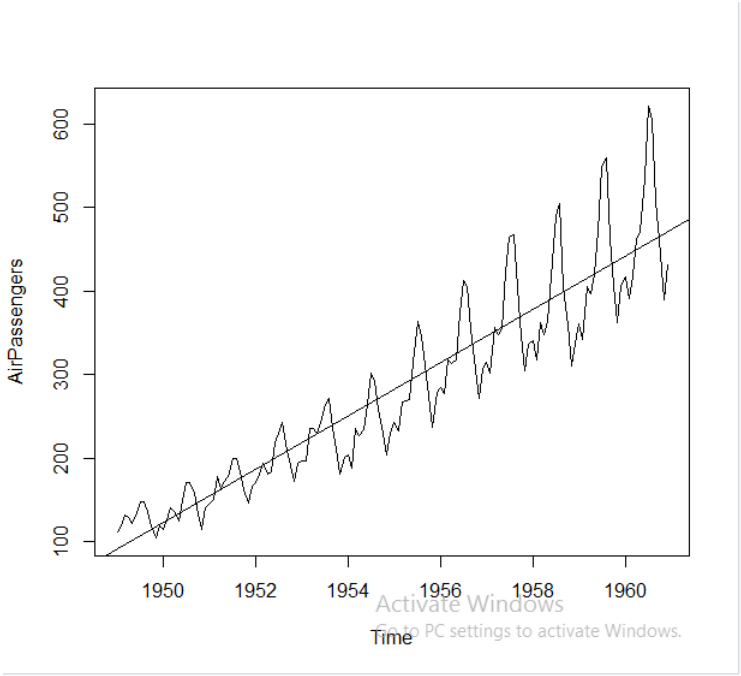
Aim:- Practical of Time-series forecasting

Code:-

```
data(AirPassengers)
class(AirPassengers)
start(AirPassengers)
end(AirPassengers)
frequency(AirPassengers)
summary(AirPassengers)
plot(AirPassengers)
```

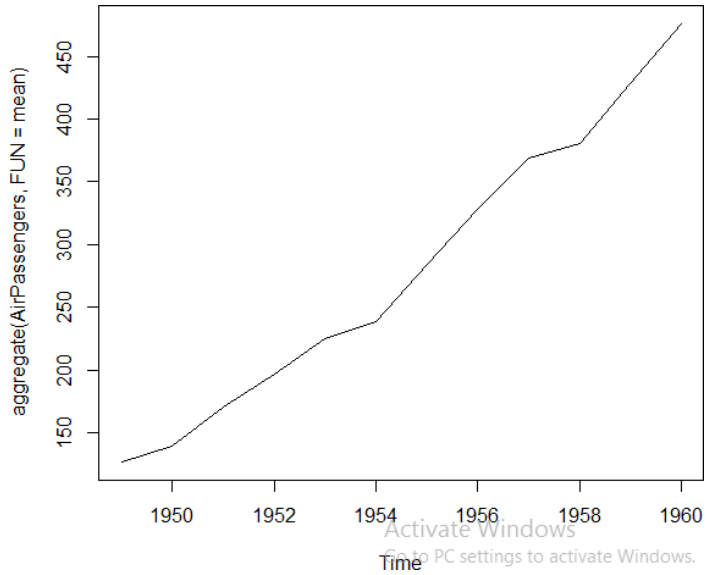



```
abline(reg=lm(AirPassengers~time(AirPassengers)))
```

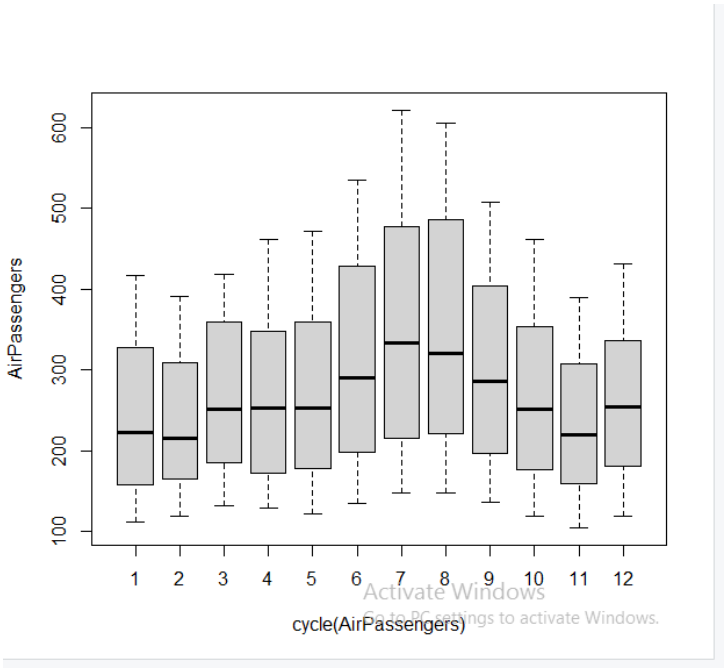


cycle(AirPassengers)

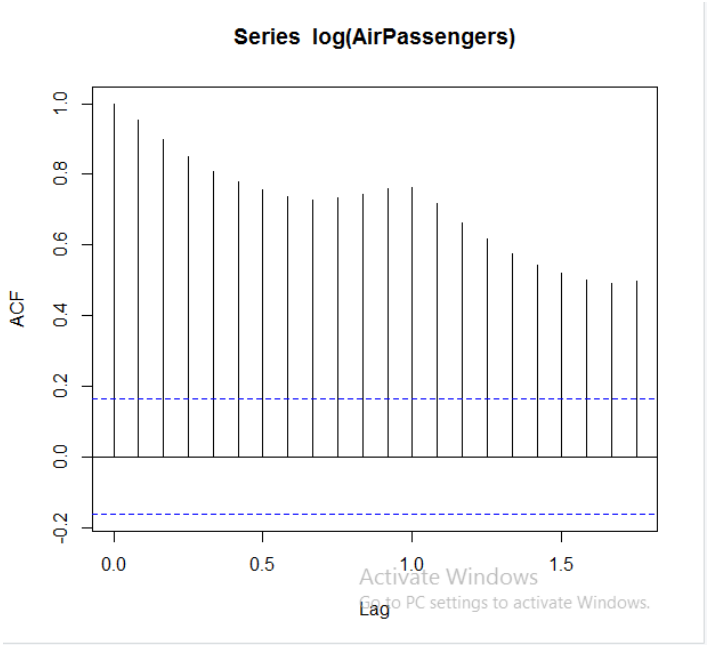
plot(aggregate(AirPassengers,FUN=mean))



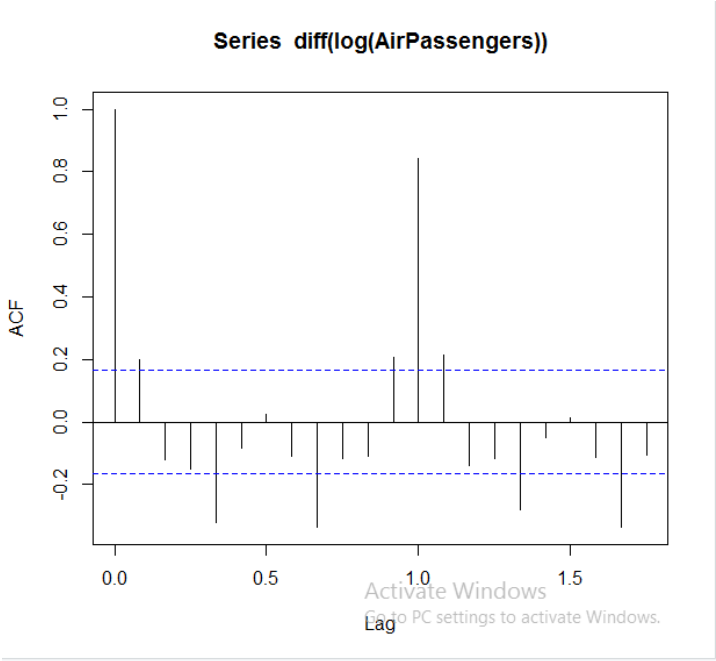
boxplot(AirPassengers~cycle(AirPassengers))



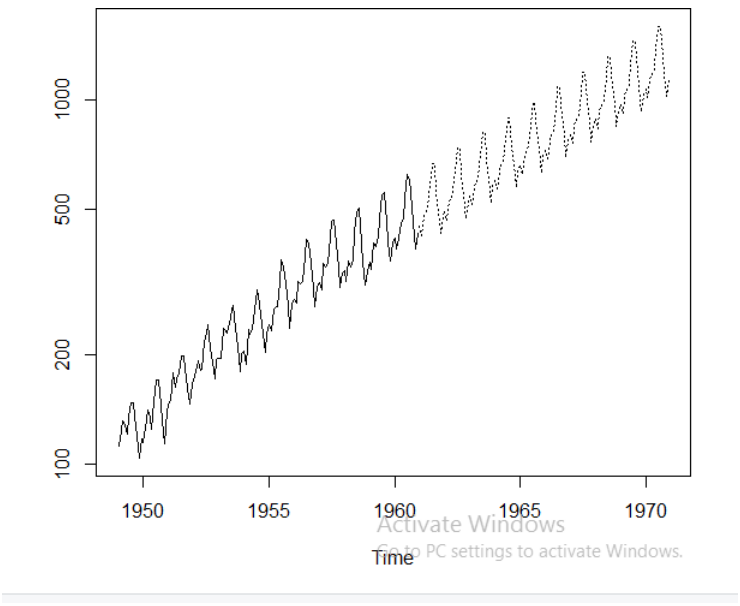
acf(log(AirPassengers))



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.plo(AirPassengers,2.718^pred$pred,log="y",lty=c(1,3))
```

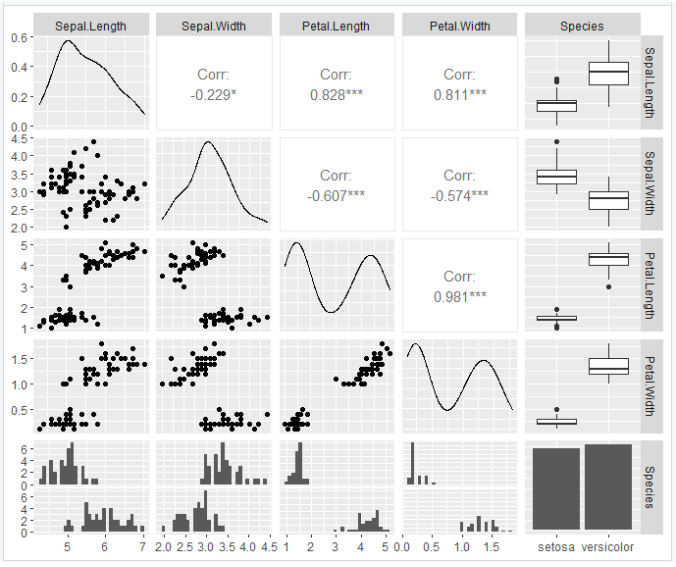


PRACTICAL NO:-7

Aim:- Practical of Logistics Regression

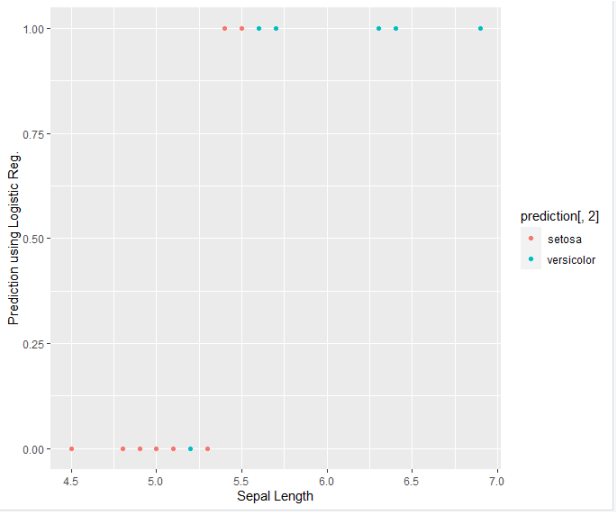
Code:-

```
library(datasets)
ir_data<- iris
head(ir_data)
str(ir_data)
levels(ir_data$Species)
sum(is.na(ir_data))
ir_data<-ir_data[1:100,]
set.seed(100)
samp<-sample(1:100,80)
ir_test<-ir_data[samp,]
ir_ctrl<-ir_data[-samp,]
install.packages("ggplot2")
library(ggplot2)
install.packages("GGally")
library(GGally)
ggpairs(ir_test)
```




```
y<-ir_test$Species; x<-ir_test$Sepal.Length
```

```
glfit<-glm(y~x, family = 'binomial')

summary(glfit)

newdata<- data.frame(x=ir_ctrl$Sepal.Length)

predicted_val<-predict(glfit, newdata, type="response")

prediction<-data.frame(ir_ctrl$Sepal.Length, ir_ctrl$Species,predicted_val)

prediction

qplot(prediction[,1], round(prediction[,3]), col=prediction[,2], xlab = 'Sepal Length', ylab = 'Prediction using Logistic Reg.')
```



PRACTICAL NO:- 8

Aim:- Practical of Hypothesis testing

Code:-

One-sampel hypothesis test:

```
x= c(6.2, 6.6, 7.1, 7.4, 7.6, 7.9, 8, 8.3, 8.4, 8.5, 8.6, + 8.8, 8.8, 9.1, 9.2, 9.4, 9.4, 9.7, 9.9, 10.2, 10.4, 10.8,
+ 11.3, 11.9)

t.test(x-9,alternative="two.sided",conf.level=0.95)

        One Sample t-test

data: x - 9

t = -0.35687, df = 23, p-value =

0.7244

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

 -0.7079827 0.4996494

sample estimates:

 mean of x

-0.1041667
```

Two-sample hypothesis test:

```
x=c(418,421,421,422,425,427,431,434,437,439,446,447,448,453,454,463,465)

y=c(429,430,430,431,36,437,440,441,445,446,447)

test2<-t.test(x,y,alternative="two.sided",mu=0,var.equal=F,conf.level=0.95)

test2

Welch Two Sample t-test

data: x and y

t = 1.0123, df = 10.202, p-value =

0.3348

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

 -44.46343 118.86984

sample estimates:
 mean of x mean of y 438.2941 401.090
```

PRACTICAL NO:-9

Aim:- Practical of Anlysis of Variance

Code:-

```
y1 = c(18.2, 20.1, 17.6, 16.8, 18.8, 19.7, 19.1)
y2 = c(17.4, 18.7, 19.1, 16.4, 15.9, 18.4, 17.7)
y3 = c(15.2, 18.8, 17.7, 16.5, 15.9, 17.1, 16.7)
y = c(y1, y2, y3)
n = rep(7, 3)
n
[1] 7 7 7
group = rep(1:3, n)
group
[1] 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3
tmp = tapply(y, group, stem)
 The decimal point is at the |
 16 | 8
 17 | 6
 18 | 28
 19 | 17
 20 | 1
 The decimal point is at the |
 15 | 9
 16 | 4
 17 | 47
 18 | 47
 19 | 1
 The decimal point is at the |
 15 | 29
 16 | 57
 17 | 17
```

```
 18 | 8
stem(y)
 The decimal point is at the |
 15 | 299
 16 | 4578
 17 | 14677
 18 | 24788
 19 | 117
 20 | 1
 tmpfn = function(x) c(sum = sum(x), mean = mean(x), var = var(x),
+                      + n = length(x))
Error: unexpected '=' in:
"tmpfn = function(x) c(sum = sum(x), mean = mean(x), var = var(x),
                    + n ="
 tmpfn = function(x) c(sum = sum(x), mean = mean(x), var = var(x), n = length(x))
 tapply(y, group, tmpfn)
$`1`
      sum       mean        var         n
130.300000  18.614286   1.358095   7.000000
$`2`
      sum       mean        var         n
123.600000  17.657143   1.409524   7.000000
$`3`
      sum       mean        var         n
117.900000  16.842857   1.392857   7.000000
 tmpfn(y)
      sum       mean        var         n
371.800000  17.704762   1.798476  21.000000
 data = data.frame(y = y, group = factor(group))
 fit = lm(y ~ group, data)
```

```
 anova(fit)
Analysis of Variance Table
Response: y
          Df Sum Sq Mean Sq F value Pr(>F)
group      2 11.007  5.5033  3.9683 0.03735 *
Residuals 18 24.963  1.3868
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
 df = anova(fit)[, "Df"]
 names(df) = c("trt", "err")
 df
trt err
  2  18
 alpha = c(0.05, 0.01)
 qf(alpha, df["trt"], df["err"], lower.tail = FALSE)
[1] 3.554557 6.012905
 anova(fit)["Residuals", "Sum Sq"]
[1] 24.96286
 anova(fit)["Residuals", "Sum Sq"]/qchisq(c(0.025, 0.975), 18,lower.tail = FALSE)
[1] 0.7918086 3.0328790
```