```
###################################################################################
            ##1-Realization of Markov Chain##
###################################################################################
#Q-1]
TPM=matrix(c(3/4,1/4,0,1/4,1/2,1/4,0,3/4,1/4),nrow=3,ncol=3,byrow=TRUE)
TPM
alpha=c(1/3,1/3,1/3)
alpha
TPM2=TPM%*%TPM
TPM2
TPM3=TPM2%*%TPM
TPM3
p1=TPM2[2,3]
p1
p2=TPM2[3,2]
p2
p3=alpha%*%TPM2
p3
p3[3]
p4=TPM3[3,2]
p4
p5=TPM[1,3]
p5
#Q-2]
TPM=matrix(c(1/2,1/2,1/3,2/3),nrow=2,ncol=2,byrow=TRUE)
TPM
TPM2=TPM%*%TPM
TPM2
TPM3=TPM2%*%TPM
TPM3
TPM4=TPM2%*%TPM2
TPM4
p1=TPM[1]
p1
p2=TPM4[1,2]
p2
p3=TPM3[1]
```

```
p3

#Q-3]

TPM=matrix(c(1/4,2/4,1/4,1/4,1/4,2/4,3/4,1/4,0),nrow=3,ncol=3,byrow=TRUE)

TPM

alpha=c(1/3,1/3,1/3)

alpha

TPM3=TPM%*%TPM%*%TPM

TPM3

p1=alpha%*%TPM3

p1

#Q-4]

TPM=matrix(c(1/2,1/2,0,1/4,1/4,1/2,1/3,1/3,1/3),nrow=3,ncol=3,byrow=TRUE)

TPM

alpha=c(2/6,3/6,1/6)

alpha

TPM2=TPM%*%TPM

TPM2

p1=alpha[1]*TPM[1,2]%*%TPM[2,2]

p1

p2=alpha[2]*TPM[2,2]%*%TPM[2,3]

p2

#Q-5]

TPM=matrix(c(0.1,0.5,0.4,06,0.2,0.2,0.3,0.4,0.3),nrow=3,ncol=3,byrow=TRUE)

TPM

alpha=c(1/3,1/3,1/3)

alpha

TPM2=TPM%*%TPM

TPM2

TPM3=TPM%*%TPM%*%TPM

TPM3

p1=alpha%*%TPM2

p1

p2=alpha[2]*TPM[2,3]%*%TPM[3,2]%*%TPM[2,3]

p2

p3=alpha[1]*TPM[1,2]%*%TPM[2,2]%*%TPM[2,1]

p3

#Q-6]
```

```
TPM=matrix(c(0,0,0,0,1,0,1/3,0,2/3,0,0,0,1/2,0,1/2,0,0,0,1,0,0,0,2/5,0,3/5),nrow=5,ncol=5,byrow=TRUE)

TPM

install.packages("markovchain")

library(markovchain)

TPM_state=c("0","1","2","3","4")

TPM_state

MC=new("markovchain",states=TPM_state,byrow=TRUE,transitionMatrix=TPM,name="TPM Matrix")

MC



##################################################################################

    #2.realization of branching process

##################################################################################

#Q.1)#i)

# Parameters

Initial_individuals <- 1

n_generations <- 6

n_trials <- 1000


# Simulate family tree function

simulate_family_tree <- function() {

 total_individuals <- Initial_individuals

 current_generation <- Initial_individuals


 for (generation in 1:n_generations) {

  next_generation <- 0


  for (individual in 1:current_generation) {

   rand <- runif(1)

   if (rand < 0.2) {

    next_generation <- next_generation + 0

   } else if (rand < 0.7) {

    next_generation <- next_generation + 1

   } else {

    next_generation <- next_generation + 2

   }

  }
```

```
    total_individuals <- total_individuals + next_generation
    current_generation <- next_generation
  }

  return(total_individuals)
}

# Run simulations
results <- replicate(n_trials, simulate_family_tree())

# Calculate statistics
mean_individuals <- mean(results)
variance_individuals <- var(results)

# Display results
cat("Mean number of individuals in the family tree after", n_generations, "generations:", mean_individuals, "\n")
cat("Variance of individuals in the family tree after", n_generations, "generations:", variance_individuals, "\n")

#######₹₹##₹####################################################################
      #3.simulation poission and its limiting #distribution
################################################################################
#Q.1)i)
poisproc <- function(lambda, time) {
  inter <- rexp(20, rate = lambda)
  arr <- cumsum(inter)
  arr <- arr[arr < time]
  n <- length(arr)
  t1 <- c(0, arr)
  t2 <- c(arr, time)
  x <- data.frame(t1, "_" = rep("<=t<", n + 1), t2, Nt = seq(0, n))
  print(x)
  return(x)
}

lambda <- 2
time <- 5
```

```r
x <- poisproc(lambda, time)

###############ii)

poisproc <- function(lambda, time) {

  inter <- rexp(20, rate = lambda)

  arr <- cumsum(inter)

  arr <- arr[arr < time]

  n <- length(arr)

  t1 <- c(0, arr)

  t2 <- c(arr, time)

  x <- data.frame(t1, "_" = rep("<=t<", n + 1), t2, Nt = seq(0, n))

  print(x)

  return(x)

}


lambda <- 4

time <- 9.5

x <- poisproc(lambda, time)

#############iii)

poisproc <- function(lambda, time) {

  inter <- rexp(20, rate = lambda)

  arr <- cumsum(inter)

  arr <- arr[arr < time]

  n <- length(arr)

  t1 <- c(0, arr)

  t2 <- c(arr, time)

  x <- data.frame(t1, "_" = rep("<=t<", n + 1), t2, Nt = seq(0, n))

  print(x)

  return(x)

}


lambda <- 9

time <- 7

x <- poisproc(lambda, time)

#Q.2)ii)

simpoiss <- function(n, t) {

  arr <- sort(runif(n, 0, t))

  t1 <- c(0, arr)
```

```
  t2 <- c(arr, t)
  x <- data.frame(t1, "_" = rep("<=t<", n + 1), t2, Nt = seq(0, n))
  print(x)
  return(x)
}

n <- 10
t <- 7
x <- simpoiss(n, t)
#q.2)iii)
simpoiss <- function(n, t) {
  arr <- sort(runif(n, 0, t))
  t1 <- c(0, arr)
  t2 <- c(arr, t)
  x <- data.frame(t1, "_" = rep("<=t<", n + 1), t2, Nt = seq(0, n))
  print(x)
  return(x)
}

n <- 7
t <- 9
x <- simpoiss(n, t)
#Q.3.i)
simpoiss <- function(lambda, t) {
  n <- length(t)
  Nt <- rep(0, n)
  Nt[1] <- rpois(1, lambda * t[1])
  for (i in 2:n) {
    Nt[i] <- rpois(1, lambda * (t[i] - t[i - 1]))
  }
  Nt <- cumsum(Nt)
  x <- data.frame(t, Nt)
  print(x)
  return(x)
}

lambda <- 1
```

```
t <- c(1.5, 2.2, 3.8, 7.5, 8.8)
x <- simpoiss(lambda, t)
#ii)
simpoiss <- function(lambda, t) {
  n <- length(t)
  Nt <- rep(0, n)
  Nt[1] <- rpois(1, lambda * t[1])
  for (i in 2:n) {
    Nt[i] <- rpois(1, lambda * (t[i] - t[i - 1]))
  }
  Nt <- cumsum(Nt)
  x <- data.frame(t, Nt)
  print(x)
  return(x)
}

lambda <- 1.5
t <- c(1.23, 2.21, 2.83, 6.05, 7.08, 17.8)
x <- simpoiss(lambda, t)



#################################################################################
        ##4-Realization of Birth and Death Process##
#################################################################################
install.packages(c("numDeriv", "DOBAD", "lattice", "Matrix", "somebm","markovchain"))
library(numDeriv)
library(DOBAD)
library(lattice)
library(Matrix)
#Q-1]
t=25
t
x0=17
x0
lambda=0.3
lambda
```

```
mu=0.3
mu
birth=birth.death.simulant(t,x0,lambda,mu)
birth
summary(birth)
BDsummaryStats(birth)
plot(birth)
#Q-2]
t=15
t
x0=10
x0
lambda=2
lambda
mu=3
mu
birth=birth.death.simulant(t,x0,lambda,mu)
birth
summary(birth)
BDsummaryStats(birth)
plot(birth)
#Q-3]
t=20
t
x0=9
x0
lambda=0.5
lambda
mu=0.3
mu
nu=0.2
nu
birth=birth.death.simulant(t,x0,lambda,mu,nu)
birth
summary(birth)
BDsummaryStats(birth)
plot(birth)
```

```
##########################################################################
            ##5-Realization of Brownian Motion Process##
##########################################################################
#Q-1]
install.packages("somebm")
library(somebm)
x0=0
x0
t0=0
t0
t=1
t
n=100
BM=bm(x0,t0,t,n)
BM
#Q-2]
n=1000
n
t0=0
t0
t=1
t
mu=0.05
mu
sigma=0.15
sigma
x0=1
x0
GBM=gbm(x0,mu,sigma,t0,t,n)
GBM
#Q-3]
x0=0
x0
t0=0
t0
t=10
```

```
t
n=100
BM=bm(x0,t0,t,n)
BM
plot(BM)
#Q-4]
n=100
n
t0=0
t0
t=1
t
mu=0.05
mu
sigma=0.1
sigma
x0=10
x0
GBM=gbm(x0,mu,sigma,t0,t,n)
GBM
plot(GBM)
#Q-5]
x0=0
x0
t0=0
t0
t=1
t
n=1000
BM=bm(x0,t0,t,n)
BM
plot(BM)


#####################################################################################
  #6.realization of gambler ruin problem
###########################₹₹########################################################
```

```r
# Gambler's Ruin Simulation
# Parameters
Initial_capital <- 10
target <- 20
n_simulations <- 1000
# Gambler's Ruin Function
gambler_ruin <- function(Initial_capital, target) {
  money <- Initial_capital
  while (money > 0 && money < target) {
    if (runif(1) < 0.5) {
      money <- money + 1
    } else {
      money <- money - 1
    }
  }
  return(money)
}
# Running the Simulation
results <- replicate(n_simulations, gambler_ruin(Initial_capital, target))
# Calculating Probabilities
pro_ruin <- mean(results == 0)
pro_success <- mean(results == target)
# Displaying Results
cat("Probability of Ruin:", pro_ruin, "\n")
cat("Probability of Success:", pro_success, "\n")


##########################################################################################
          ##7-One Way Classification. Multiple Comparisons Test##
##########################################################################################
#Q-1]
c1=c(264,272,268,277,256,295)
c1
c2=c(278,291,297,282,285,277)
c2
c3=c(275,293,278,271,263,276)
c3
```

```
c4=c(255,266,249,264,270,268)
c4
C=c(c1,c2,c3,c4)
C
N=length(C)
N
a=4
a
m1=mean(c1)
m1
m2=mean(c2)
m2
m3=mean(c3)
m3
m4=mean(c4)
m4
y1=sum(c1)
y1
y2=sum(c2)
y2
y3=sum(c3)
y3
y4=sum(c4)
y4
y1.bar=mean(c1)
y1.bar
y2.bar=mean(c2)
y2.bar
y3.bar=mean(c3)
y3.bar
y4.bar=mean(c4)
y4.bar
y..=sum(c1,c2,c3,c4)
y..
cf=(y..^2/N)
cf
ymean=mean(y..)
```

```
ymean
C1=c1^2
C1
C2=c2^2
C2
C3=c3^2
C3
C4=c4^2
C4
yijsq=sum(C1,C2,C3,C4)
yijsq
SST=(yijsq)-cf
SST
yi.sq=(y1^2/6)+(y2^2/6)+(y3^2/6)+(y4^2/6)
yi.sq
SStreat=yi.sq-cf
SStreat
SSE=SST-SStreat
SSE
df1=a-1
df1
df2=N-a
df2
df3=N-1
df3
MStreat=SStreat/df2
MStreat
MSE=SSE/df3
MSE
Ftreat=MStreat/MSE
Ftreat
SV=c("treatment","error","total")
DF=c(df1,df2,df3)
SS=c(SStreat,SSE,SST)
MSS=c(MStreat,MSE,0)
F=c(Ftreat,0,0)
anova=data.frame("SV"=SV,"DF"=DF,"SS"=SS,"MSS"=MSS,"F"=F)
```

```
anova

##Hypothesis

cat("H0:there is no significant difference between different treatment mean","\n")

cat("H1:atleast two treatment means are differ significantly","\n")

##conclusion

Ftv=qf(0.95,3,20)

Ftv

cat("Here Ftreat is less than Ftv,so we accept H0A at 5% l.o.s")


#Q-2]

cc=c(643,655,702)

cc

mc=c(469,427,456)

mc

fc=c(484,456,402)

fc

CS=c(cc,mc,fc)

CS

N=length(CS)

N

a=3

a

m1=mean(cc)

m1

m2=mean(mc)

m2

m3=mean(fc)

m3

y1=sum(cc)

y1

y2=sum(mc)

y2

y3=sum(fc)

y3

y1.bar=mean(cc)

y1.bar

y2.bar=mean(mc)
```

```
y2.bar
y3.bar=mean(fc)
y3.bar
y..=sum(cc,mc,fc)
y..
cf=(y..^2/N)
cf
ymean=mean(y..)
ymean
CC=cc^2
CC
MC=mc^2
MC
FC=fc^2
FC
yijsq=sum(CC,MC,FC)
yijsq
SST=(yijsq)-cf
SST
yi.sq=(y1^2/3)+(y2^2/3)+(y3^2/3)
yi.sq
SStreat=yi.sq-cf
SStreat
SSE=SST-SStreat
SSE
df1=a-1
df1
df2=N-a
df2
df3=N-1
df3
MStreat=SStreat/df2
MStreat
MSE=SSE/df3
MSE
Ftreat=MStreat/MSE
Ftreat
```

```
SV=c("treatment","error","total")

DF=c(df1,df2,df3)

SS=c(SStreat,SSE,SST)

MSS=c(MStreat,MSE,0)

F=c(Ftreat,0,0)

anova=data.frame("SV"=SV,"DF"=DF,"SS"=SS,"MSS"=MSS,"F"=F)

anova

##Hypothesis

cat("H0:the mean pressure applied to the driver’s head during a crash test is equal for each type of car","\n")

cat("H1:the mean pressure applied to the driver’s head during a crash test is not equal for each type of car","\n")

##conclusion

Ftv=qf(0.95,2,8)

Ftv

cat("Here Ftreat is greater than Ftv,so we accept H0A at 5% l.o.s")



############################################################################################

          ##8-Two way Classification with one observation per cell (with interaction)##

################################################################################################

#Q-1]

R1=c(9,10,9,10,11,11)

R2=c(12,11,9,11,10,10)

R3=c(11,10,10,12,11,10)

R4=c(12,13,11,14,12,10)

C1=c(9,12,11,12)

C2=c(10,11,10,13)

C3=c(9,9,10,11)

C4=c(10,11,12,14)

C5=c(11,10,11,12)

C6=c(11,10,10,10)

R1.=sum(R1)

R1.

R2.=sum(R2)

R2.

R3.=sum(R3)

R3.

R4.=sum(R4)
```

```
R4.
C1.=sum(C1)
C1.
C2.=sum(C2)
C2.
C3.=sum(C3)
C3.
C4.=sum(C4)
C4.
C5.=sum(C5)
C5.
C6.=sum(C6)
C6.
N=24
Y..=sum(R1.,R2.,R3.,R4.)
Y..
CF=(Y..*Y..)/N
CF
SST=sum(R1^2,R2^2,R3^2,R4^2)-CF
SST
SSA=((1/6)*sum(R1.^2,R2.^2,R3.^2,R4.^2))-CF
SSA
SSB=((1/4)*sum(C1.^2,C2.^2,C3.^2,C4.^2,C5.^2,C6.^2))-CF
SSB
SSE=SST-SSA-SSB
SSE
MSA=SSA/3
MSB=SSB/5
MSE=SSE/15
MST=SST/23
FA=MSA/MSE
FB=MSB/MSE
SV=c("row","column","error","total")
DF=c(3,5,15,23)
SS=c(SSA,SSB,SSE,SST)
MSS=c(MSA,MSB,MSE,0)
F=c(FA,FB,0,0)
```

```
anova=data.frame("SV"=SV,"DF"=DF,"SS"=SS,"MSS"=MSS,"F"=F)

anova

##HYPOTHESIS

cat("HO: There is no significant difference between different levels of factor A")

cat("H1:At least two levels of factor A differ significantly")

cat("HO': There is no significant difference between different levels of factor B")

cat("H1':At least two levels of factor B differ significantly")

##CONCLUSION

FAcritical=qf(0.95,df1=3,df2=15)

FAcritical

FBcritical=qf(0.95,df1=5,df2=15)

FBcritical

cat("Here FA is less than FAcritical so we accept HO at 5%  l.o.s.")

cat("Here FB is greater than FBcritical so we reject HO' at 5% l.o.s.")


#Q-2]

r1=c(5.1,5.0,4.8,5.0,5.1,5.3,5.1,5.1,4.9,4.9,4.9,5.0,5.0,5.0,5.0)

r2=c(5.2,5.2,5.4,5.3,5.3,5.5,5.3,5.2,5.2,5.2,5.0,5.5,5.1,5.3,5.9)

r3=c(5.8,5.7,5.9,6,5.9,6.2,5.8,5.9,5.9,5.8,5.5,5.5,5.9,5.4,5.5)

r4=c(6,6,5.9,6.2,6.5,6,6,6.1,6,6,5.8,5.5,5.8,5.6,5.5)

r5=c(6,6,6,6,6.1,6.3,5.9,6,5.8,5.9,6,5.5,5.5,6,6.2)

R1=sum(c(5.1,5.0,4.8))

R1

R2=sum(c(5.0,5.1,5.3))

R2

R3=sum(c(5.1,5.1,4.9))

R3

R4=sum(c(4.9,4.9,5.0))

R4

R5=sum(c(5.0,5.0,5.0))

R5

R6=sum(c(5.2,5.2,5.4))

R6

R7=sum(c(5.3,5.3,5.5))

R7

R8=sum(c(5.3,5.2,5.2))

R8
```

```
R9=sum(c(5.2,5.0,5.5))
R9
R10=sum(c(5.1,5.3,5.9))
R10
R11=sum(c(5.8,5.7,5.9))
R11
R12=sum(c(6,5.9,6.2))
R12
R13=sum(c(5.8,5.9,5.9))
R13
R14=sum(c(5.8,5.5,5.5))
R14
R15=sum(c(5.9,5.4,5.5))
R15
R16=sum(c(6,6,5.9))
R16
R17=sum(c(6.2,6.5,6))
R17
R18=sum(c(6,6.1,6))
R18
R19=sum(c(6,5.8,5.5))
R19
R20=sum(c(5.8,5.6,5.5))
R20
R21=sum(c(6,6,6))
R21
R22=sum(c(6,6.1,6.3))
R22
R23=sum(c(5.9,6,5.8))
R23
R24=sum(c(5.9,6,5.5))
R24
R25=sum(c(5.5,6,6.2))
R25
RR=sum(R1^2,R2^2,R3^2,R4^2,R5^2,R6^2,R7^2,R8^2,R9^2,R10^2,R11^2,R12^2,R13^2,R14^2,R15^2,R16^2,R17^2,R18^2,R19^2,R20^2,R21^2,R22^2,R23^2,R24^2,R25^2)
RR
```

```
R1.=sum(R1,R2,R3,R4,R5)

R1.

R2.=sum(R6,R7,R8,R9,R10)

R2.

R3.=sum(R11,R12,R13,R14,R15)

R3.

R4.=sum(R16,R17,R18,R19,R20)

R4.

R5.=sum(R21,R22,R23,R24,R25)

R5.

C1.=sum(R1,R6,R11,R16,R21)

C1.

C2.=sum(R2,R7,R12,R17,R22)

C2.

C3.=sum(R3,R8,R13,R18,R23)

C3.

C4.=sum(R4,R9,R14,R19,R24)

C4.

C5.=sum(R5,R10,R15,R20,R25)

C5.

Y...=sum(R1.,R2.,R3.,R4.,R5.)

Y...

N=75

N

m=3

m

a=5

a

b=5

b

CF=(Y...)^2/N

CF

SST=sum(r1^2,r2^2,r3^2,r4^2,r5^2)-CF

SST

SSA=((1/(m*b))*sum(R1.^2,R2.^2,R3.^2,R4.^2,R5.^2))-CF

SSA

SSB=((1/(m*a))*sum(C1.^2,C2.^2,C3.^2,C4.^2,C5.^2))-CF
```

```
SSB
SSM=((1/m)*RR)-CF
SSM
SSAB=SSM-SSA-SSB
SSAB
SSE=SST-SSA-SSB-SSAB
SSE
df1=a-1
df1
df2=b-1
df1
df3=(a-1)*(b-1)
df3
df4=a*b*(m-1)
df4
df5=(a*b*m)-1
df5
MSA=SSA/df1
MSA
MSB=SSB/df2
MSB
MSAB=SSAB/df3
MSAB
MSE=SSE/df4
MSE
FA=MSA/MSE
FA
FB=MSB/MSE
FB
FAB=MSAB/MSE
FAB
SV=c("row","column","treatment","error","total")
DF=c(df1,df2,df3,df4,df5)
SS=c(SSA,SSB,SSAB,SSE,SST)
MSS=c(MSA,MSB,MSAB,MSE,0)
Fratio=c(FA,FB,FAB,0,0)
anov=data.frame("SV"=SV,"DF"=DF,"SS"=SS,"MSS"=MSS,"Fratio"=Fratio)
```

```
anov

##Hypothesis

cat("H0A:there is no significance difference between different level of factor A")

cat("H1A:atleast two level of factor A differ significantly")

cat("H0B:there is no significance difference between different level of factor B")

cat("H1B:atleast two level of factor B differ significantly")

cat("H0AB:factor A&B are independent")

cat("H1AB:factor A&B are not independent")


##Conclusion

FAcritical=qf(0.95,df1,df4)

FAcritical

FBcritical=qf(0.95,df2,df4)

FBcritical

FABcritical=qf(0.95,df3,df4)

FABcritical

cat("Here FA is greater than Fcritical so we reject HOA at 5% l.o.s.")

cat("Here FB is greater than Fcritical2 so we reject HOB' at 5% l.o.s.")

cat("Here FAB is less than Fcritical3 so we accept H0AB' at 5% l.o.s.")


#################################################################################

              ##9-Analysis of LSD and BIBD##

#################################################################################

#Q-1]LSD

r1=c(29.1,18.9,29.4,5.7)

r2=c(16.4,10.2,21.2,19.1)

r3=c(5.4,38.8,24.0,37.0)

r4=c(24.9,41.7,9.5,28.9)

c1=c(29.1,16.4,5.4,24.9)

c2=c(18.9,10.2,38.8,41.7)

c3=c(29.4,21.2,24.0,9.5)

c4=c(5.7,19.1,37.0,28.9)

t1=c(5.4,10.2,9.5,5.7)

t2=c(24.9,18.9,24.0,19.1)

t3=c(16.4,41.7,29.4,37.0)

t4=c(29.1,38.8,21.2,28.9)

N=16
```

```
N
y1..=sum(r1)
y1..
y2..=sum(r2)
y2..
y3..=sum(r3)
y3..
y4..=sum(r4)
y4..
y.1.=sum(c1)
y.1.
y.2.=sum(c2)
y.2.
y.3.=sum(c3)
y.3.
y.4.=sum(c4)
y.4.
y..1=sum(t1)
y..1
y..2=sum(t2)
y..2
y..3=sum(t3)
y..3
y..4=sum(t4)
y..4
y...=sum(y1..,y2..,y3..,y4..)
y...
N=16
N
m=4
m
CF=(y...)^2/N
CF
SSr=(1/4)%*%(sum(y1..^2,y2..^2,y3..^2,y4..^2))-CF
SSr
SSc=(1/4)%*%(sum(y.1.^2,y.2.^2,y.3.^2,y.4.^2))-CF
SSc
```

```
SStreat=(1/4)%*%(sum(y..1^2,y..2^2,y..3^2,y..4^2))-CF

SStreat

Yijksq=sum(r1^2,r2^2,r3^2,r4^2)

Yijksq

SSt=Yijksq-CF

SSt

SSe=SSt-SSr-SSc-SStreat

SSe

df1=m-1

df1

df2=m-1

df2

df3=m-1

df3

df4=(m-1)*(m-2)

df4

df5=N-1

df5

SV=c("Row","Column","Treatment","Error","Total")

DF=c(df1,df2,df3,df4,df5)

DF

SS=c(SSr,SSc,SStreat,SSe,SSt)

MSr=SSr/df2

MSr

MSc=SSc/df2

MSc

MStreat=SStreat/df3

MStreat

MSe=SSe/df4

MSe

MSS=c(MSr,MSc,MStreat,MSe,0)

Fr=MSr/MSe

Fr

Fc=MSc/MSe

Fc

Ftreat=MStreat/MSe

Ftreat
```

```
Fratio=c(Fr,Fc,Ftreat,0,0)

Fratio

anova=data.frame("SV"=SV,"DF"=DF,"SS"=SS,"MSS"=MSS,"Fratio"=Fratio)

anova

##Hypothesis

cat("H0R:there is no significance difference between different row effect")

cat("H1R:atleast two row effect differ significantly")

cat("H0C:there is no significance difference between different column effect")

cat("H1C:atleast two column effect differ significantly")

cat("H0treat:there is no significance difference between different treatment effect")

cat("H1treat:atleast two treatment effect differ significantly")


##Conclusion

print("Fratio>F(m-1,((m-1)*(m-2)),alpha),then Reject H0")

#Fr>F(3,6,0.05),i.e.,3.316653<4.76.Accept H0R at 5% of los.There is no significance difference between different row effect)

#Fc>F(3,6,0.05),i.e.,1.985963<4.76.Accept H0C at 5% of los.There is no significance difference between different column effect)

#Ftreat>F(3,6,0.05),i.e.,17.549690>4.76.Reject H0treat at 5% of los.Atleast two treatment effect differ significantly)


##Q-2)BIBD

V1=c(73,74,0,71)

V2=c(0,75,67,72)

V3=c(73,75,68,0)

V4=c(75,0,72,75)

B1=c(73,0,73,75)

B2=c(74,75,75,0)

B3=c(0,67,68,72)

B4=c(71,72,0,75)

X1.=sum(V1)

X1.

X2.=sum(V2)

X2.

X3.=sum(V3)

X3.

X4.=sum(V4)

X4.

X.1=sum(B1)
```

```
X.1

X.2=sum(B2)

X.2

X.3=sum(B3)

X.3

X.4=sum(B4)

X.4

V1.=V1^2

V1.

V2.=V2^2

V2.

V3.=V3^2

V3.

V4.=V4^2

V4.

X..=sum(X1.,X2.,X3.,X4.)

X..

v=4

b=4

k=3

r=3

lambda=2

N=12

CF=(X..^2)/N

CF

Xijsq=sum(V1.,V2.,V3.,V4.)

Xijsq

SST=Xijsq-CF

SST

SSblock=(1/k)*(X.1^2+X.2^2+X.3^2+X.4^2)-CF

SSblock

Q1=X1.-(1/k)%*%(1*X.1+1*X.2+0*X.3+1*X.4)

Q1

Q2=X2.-(1/k)%*%(0*X.1+1*X.2+1*X.3+1*X.4)

Q2

Q3=X3.-(1/k)%*%(1*X.1+1*X.2+1*X.3+0*X.4)

Q3
```

```
Q4=X4.-(1/k)%*%(1*X.1+0*X.2+1*X.3+1*X.4)

Q4

SStreat=((3)%*%(sum(Q1^2,Q2^2,Q3^2,Q4^2)))/(lambda*v)

SStreat

SSE=SST-SSblock-SStreat

SSE

df1=v-1

df1

df2=b-1

df2

df3=N-v-b+1

df3

df4=N-1

df4

MStreat=SStreat/df1

MStreat

MSblock=SSblock/df2

MSblock

MSE=SSE/df3

MSE

F=MStreat/MSE

F

SV=c("treatment","block","error","total")

DF=c(df1,df2,df3,df4)

SS=c(SStreat,SSblock,SSE,SST)

MSS=c(MStreat,MSblock,MSE,0)

Fratio=c(F,0,0,0)

anov=data.frame("SV"=SV,"DF"=DF,"SS"=SS,"MSS"=MSS,"Fratio"=Fratio)

anov


#Hypothesis

cat("H0:there is no significant difference between different treatment effects","\n")

cat("H1:atleast two treatment effects are differ significantly","\n")


#Conclusion:

print("Fratio>F(v-1,N-v-b+1,alpha),then Reject H0")

#F>F(3,5,0.05),i.e.,11.66667>5.41.Reject H0 at 5% of los.Atleast two treatment effects are differ significantly)
```

```
###########################################################################################

          ##10-Analysis of covariance in one way and two way model##

###########################################################################################

#Question 1)

t = 3

n = 4

N = 12

Y1 = c(27, 44, 33, 11)

Y2 = c(25, 35, 46, 26)

Y3 = c(40, 22, 42, 25)

X1 = c(24, 40, 35, 40)

X2 = c(26, 32, 42, 25)

X3 = c(38, 26, 50, 26)


Y1. = sum(Y1)

Y2. = sum(Y2)

Y3. = sum(Y3)

X1. = sum(X1)

X2. = sum(X2)

X3. = sum(X3)


Y.. = sum(Y1., Y2., Y3.)

X.. = sum(X1., X2., X3.)


cfy = Y..^2 / N

cfx = X..^2 / N


Syy = sum(Y1^2, Y2^2, Y3^2) - cfy

Sxx = sum(X1^2, X2^2, X3^2) - cfx

Sxy = sum(Y1 * X1, Y2 * X2, Y3 * X3) - (Y.. * X..) / N


Tyy = 1 / 4 * (Y1.^2 + Y2.^2 + Y3.^2) - cfy

Txx = 1 / 4 * (X1.^2 + X2.^2 + X3.^2) - cfx

Txy = 1 / 4 * ((Y1. * X1.) + (Y2. * X2.) + (Y3. * X3.)) - X.. * Y.. / N


Eyy = Syy - Tyy
```

```
Exx = Sxx - Txx

Exy = Sxy - Txy


Bhat = Exy / Exx

SSE = Eyy - (Exy^2 / Exx)

Bhatdsh = Sxy / Sxx

SSEst = Syy - (Sxy^2 / Sxx)


F = ((SSEst - SSE) / (t - 1)) / (SSE / (N - t - 1))

Fcritical = qf(0.95, df1 = t - 1, df2 = N - t - 1)


if (F >= Fcritical) {

  cat("Reject H0 at 5% l.o.s, Therefore at least two treatment effects differ significantly")

} else {

  cat("Accept H0 at 5% l.o.s, Therefore all the treatments are equally effective.")

}


F1 = (Exy^2 / Exx) / (SSE / (N - t - 1))

F1critical = qf(0.95, df1 = 1, df2 = N - t - 1)


if (F1 >= F1critical) {

  cat("Reject H0 at 5% l.o.s, Therefore B is significant")

} else {

  cat("Accept H0 at 5% l.o.s, Therefore B is not significant.")

}


SV = c("Treatment", "Error", "Total", "Difference")

DF = c(t - 1, N - t, N - 1, 0)

SSandSP = data.frame(xx = c(Txx, Exx, Sxx, 0), xy = c(Txy, Exy, Sxy, 0), yy = c(Tyy, Eyy, Syy, 0))

EstimateB = c(0, Bhat, Bhatdsh, 0)

AdjSS = c(0, SSE, SSEst, (SSEst - SSE))

Adjdf = c(0, N - t - 1, N - 2, t - 1)


ANCOV = data.frame("SV" = SV, "DF" = DF, "SSandSP" = SSandSP, "EstimateB" = EstimateB, "AdjSS" = AdjSS, "Adjdf" = Adjdf)

ANCOV


# Question 2
```

```
a = 3

b = 5

N = 15

Y1 = c(68, 90, 98, 77, 88)

Y2 = c(112, 94, 65, 74, 88)

Y3 = c(118, 82, 73, 92, 80)

X1 = c(120, 140, 150, 125, 136)

X2 = c(165, 140, 120, 125, 133)

X3 = c(175, 132, 124, 141, 130)


B1y = c(68, 112, 118)

B2y = c(90, 94, 82)

B3y = c(98, 65, 73)

B4y = c(77, 74, 92)

B5y = c(88, 88, 80)


B1x = c(120, 165, 175)

B2x = c(140, 140, 132)

B3x = c(150, 120, 124)

B4x = c(125, 125, 141)

B5x = c(136, 133, 130)


Y1. = sum(Y1)

Y2. = sum(Y2)

Y3. = sum(Y3)

X1. = sum(X1)

X2. = sum(X2)

X3. = sum(X3)


B1y. = sum(B1y)

B2y. = sum(B2y)

B3y. = sum(B3y)

B4y. = sum(B4y)

B5y. = sum(B5y)

B1x. = sum(B1x)

B2x. = sum(B2x)

B3x. = sum(B3x)
```

```
B4x. = sum(B4x)
B5x. = sum(B5x)

Y.. = sum(Y1., Y2., Y3.)
X.. = sum(X1., X2., X3.)

cfy = Y..^2 / N
cfx = X..^2 / N

Syy = sum(Y1^2, Y2^2, Y3^2) - cfy
Sxx = sum(X1^2, X2^2, X3^2) - cfx
Sxy = sum(Y1 * X1, Y2 * X2, Y3 * X3) - (Y.. * X..) / N

Tyy = 1 / b * (Y1.^2 + Y2.^2 + Y3.^2) - cfy
Txx = 1 / b * (X1.^2 + X2.^2 + X3.^2) - cfx
Txy = 1 / b * ((Y1. * X1.) + (Y2. * X2.) + (Y3. * X3.)) - X.. * Y.. / N

Byy = 1 / a * (B1y.^2 + B2y.^2 + B3y.^2 + B4y.^2 + B5y.^2) - cfy
Bxx = 1 / a * (B1x.^2 + B2x.^2 + B3x.^2 + B4x.^2 + B5x.^2) - cfx
Bxy = 1 / a * (B1y. * B1x. + B2y. * B2x. + B3y. * B3x. + B4y. * B4x. + B5y. * B5x.) - (X.. * Y..) / N

Eyy = Syy - Tyy - Byy
Exx = Sxx - Txx - Bxx
Exy = Sxy - Txy - Bxy

Bhat = Exy / Exx
SSE = Eyy - (Exy^2 / Exx)
Exxdsh = Txx + Exx
Eyydsh = Tyy + Eyy
Exydsh = Txy + Exy
Bhatdsh = Exydsh / Exxdsh
SSEst = Eyydsh - (Exydsh^2 / Exxdsh)

F = ((SSEst - SSE) / (a - 1)) / (SSE / ((a - 1) * (b - 1) - 1))
Fcritical = qf(0.95, df1 = a - 1, df2 = (a - 1) * (b - 1) - 1)

if (F >= Fcritical) {
```

```
  cat("Reject H0 at 5% l.o.s, Therefore at least two treatment effects differ significantly")

} else {

  cat("Accept H0 at 5% l.o.s, Therefore all the treatments are equally effective.")

}


F1 = (Exy^2 / Exx) / (SSE / ((a - 1) * (b - 1) - 1))

F1critical = qf(0.95, df1 = 1, df2 = (a - 1) * (b - 1) - 1)


if (F1 >= F1critical) {

  cat("Reject H0 at 5%)}










####################################################################################################

                    ##11-2^k factorial analysis with single replicate##

####################################################################################################

#Q-1]

k=4

r=1

a=rep(c(-1,1),8)

a

b=rep(c(-1,-1,1,1),4)

b

c=rep((rep(c(-1,1),each=4)),2)

c

d=rep(c(-1,1),each=8)

d

resp=c(44,70,49,66,68,60,80,65,42,100,45,102,77,85,72,94)

resp

A=a*resp

A

B=b*resp

B

C=c*resp
```

```
C
D=d*resp
D
AB=a*b*resp
AB
AC=a*c*resp
AC
AD=a*d*resp
AD
BC=b*c*resp
BC
BD=b*d*resp
BD
CD=c*d*resp
CD
ABC=a*b*c*resp
ABC
ABD=a*b*d*resp
ABD
ACD=a*c*d*resp
ACD
BCD=b*c*d*resp
BCD
ABCD=a*b*c*d*resp
ABCD
d1=data.frame(A,B,C,D,AB,AC,AD,BC,BD,CD,ABC,ABD,ACD,BCD,ABCD)
d1
ct=colSums(d1)
ct
ct2=ct^2
ct2
eff=ct/(2^(k-1)*r)
eff
SS=ct2/(2^k*r)
SS
SST=sum(SS)
SST
```

```
percont=SS/SST*100
percont
modtm=c("A","B","C","D","AB","AC","AD","BC","BD","CD","ABC","ABD","ACD","BCD","ABCD")
modtm
d2=data.frame("SR.NO"=1:15,"Modelterm"=modtm,SS,percont)
d2
dt=c(2,5,8,9,11,12,14,15)
dt
SSE=sum(SS[dt])
SSE
SV=c(modtm[-dt],"Error","Total")
SV
S_S=c(SS[-dt],SSE,SST)
S_S
df=c(1,1,1,1,1,1,1,8,15)
MS=S_S/df
MS
Fratio=MS/MS[8]
Fratio
anova=data.frame(SV,df,S_S,MS,Fratio)
anova
tv=qf(0.95,1,8)
tv
##Hypothesis
cat("H0A:main effect A is not significant")
cat("H1A:main effect A is significant")
cat("H0C:main effect C is not significant")
cat("H1C:main effect C is significant")
cat("H0D:main effect D is not significant")
cat("H1D:main effect D is significant")
cat("H0AC:interaction effect AC is not significant")
cat("H1AC:interaction effect AC is significant")
cat("H0AD:interaction effect AD is not significant")
cat("H1AD:interaction effect AD is significant")
cat("H0CD:interaction effect CD is not significant")
cat("H1CD:interaction effect CD is significant")
cat("H0ACD:interaction effect ACD is not significant")
```

```
cat("H1ACD:interaction effect ACD is significant")


##Conclusion

cat("Here FA>tv,so we reject H0A at 5% l.o.s")

cat("Here FC>tv,so we reject H0A at 5% l.o.s")

cat("Here FD>tv,so we reject H0A at 5% l.o.s")

cat("Here FAC>tv,so we reject H0A at 5% l.o.s")

cat("Here FAD>tv,so we reject H0A at 5% l.o.s")

cat("Here FCD>tv,so we reject H0A at 5% l.o.s")

cat("Here FACD>tv,so we reject H0A at 5% l.o.s")


########################################################################################################

            ##12-Total and partial confounding in 2^k factorial experiment##

########################################################################################################

#Q-1]

m=matrix(c(3,1,4,3,6,3,3,4,2,4,1,0,3,4,4,3,0,0,3,0,2,5,3,5,4,5,2,2,1,2,4,0),nrow=4,ncol=8,byrow=TRUE)

m

r=4

r

x..=sum(m)

x..

cf=(x..^2)/(8*r)

cf

x.=m^2

x.

SST=sum(x.)-cf

SST

x.j=colSums(m)

x.j

x.jsq=x.j^2

x.jsq

SSb=1/4*(sum(x.j^2))-cf

SSb

T1=c(2,4,6,4)

Ta=c(4,3,3,4)

Tb=c(1,0,3,3)

Tab=c(3,3,5,5)
```

```
Tc=c(5,1,4,3)

Tac=c(0,0,2,3)

Tbc=c(4,2,2,0)

Tabc=c(0,2,1,4)

dataF=rbind(T1,Ta,Tb,Tab,Tc,Tac,Tbc,Tabc)

dataF

T=rowSums(dataF)

T

A=-T[1]+T[2]-T[3]+T[4]-T[5]+T[6]-T[7]+T[8]

A

B=-T[1]-T[2]+T[3]+T[4]-T[5]-T[6]+T[7]+T[8]

B

C=-T[1]-T[2]-T[3]-T[4]+T[5]+T[6]+T[7]+T[8]

C

SSA=A^2/(8*r)

SSA

SSB=B^2/(8*r)

SSB

SSC=C^2/(8*r)

SSC

AB=rowSums(dataF[,-2])

AB

TAB=AB[1]-AB[2]-AB[3]+AB[4]+AB[5]-AB[6]-AB[7]+AB[8]

TAB

SSAB=TAB^2/(8*(r-1))

SSAB

AC=rowSums(dataF[,-4])

AC

TAC=AC[1]-AC[2]+AC[3]-AC[4]-AC[5]+AC[6]-AC[7]+AC[8]

TAC

SSAC=TAC^2/(8*(r-1))

SSAC

BC=rowSums(dataF[,-3])

BC

TBC=BC[1]+BC[2]-BC[3]-BC[4]-BC[5]-BC[6]+BC[7]+BC[8]

TBC

SSBC=TBC^2/(8*(r-1))
```

```
SSBC

ABC=rowSums(dataF[,-1])

ABC

TABC=-ABC[1]+ABC[2]+ABC[3]-ABC[4]+ABC[5]-ABC[6]-ABC[7]+ABC[8]

TABC

SSABC=TABC^2/(8*(r-1))

SSABC

SSE=SST-SSb-SSA-SSB-SSC-SSAB-SSAC-SSBC-SSABC

SSE

df1=2*r-1

df1

df2=1

df2

df3=1

df3

df4=1

df4

df5=1

df5

df6=1

df6

df7=1

df7

df8=1

df8

df9=6*(r-1)

df9

df10=8*r-1

df10

MSb=SSb/df1

MSb

MSA=SSA/df2

MSA

MSB=SSB/df3

MSB

MSC=SSC/df4

MSC
```

```
MSAB=SSAB/df5

MSAB

MSAC=SSAC/df6

MSAC

MSBC=SSBC/df7

MSBC

MSABC=SSABC/df8

MSABC

MSE=SSE/df9

MSE

Fb=MSb/MSE

Fb

FA=MSA/MSE

FA

FB=MSB/MSE

FB

FC=MSC/MSE

FC

FAB=MSAB/MSE

FAB

FAC=MSAC/MSE

FAC

FBC=MSBC/MSE

FBC

FABC=MSABC/MSE

FABC

SV=c("block","A","B","C","AB","AC","BC","ABC","Error","Total")

DF=c(df1,df2,df3,df4,df5,df6,df7,df8,df9,df10)

SS=c("SSb","SSA","SSB","SSC","SSAB","SSAC","SSBC","SSABC","SSE","SST")

MSS=c("MSb","MSA","MSB","MSC","MSAB","MSAC","MSBC","MSABC","MSE","0")

Fratio=c(Fb,FA,FB,FC,FAB,FAC,FBC,FABC,0,0)

anov=data.frame("SV"=SV,"DF"=DF,"SS"=SS,"MSS"=MSS,"Fratio"=Fratio)

anov

##Hypothesis

cat("H0:confounding is not effective")

cat("H1:confounding is effective")

cat("HOA:main effect A is not significant")
```

```
cat("H1A:main effect A is significant")

cat("HOB:main effect B is not significant")

cat("H1B:main effect B is significant")

cat("HOC:main effect C is not significant")

cat("H1C:main effect C is significant")

cat("H0AB:interaction effect AB is not significant")

cat("H1AB:interaction effect AB is not significant")

cat("H0AC:interaction effect AC is not significant")

cat("H1AC:interaction effect AC is not significant")

cat("H0BC:interaction effect BC is not significant")

cat("H1BC:interaction effect BC is not significant")

cat("H0ABC:interaction effect ABC is not significant")

cat("H1ABC:interaction effect ABC is not significant")

##Conclusion

Ftv1=qf(0.95,7,18)

Ftv1

Ftv2=qf(0.95,1,18)

Ftv2

cat("Here Fb is less than Ftv1, so accept H0 at 5% of l.o.s")

cat("Here FA is less than Ftv2, so accept H0A at 5% of l.o.s")

cat("Here FB is less than Ftv2, so accept H0B at 5% of l.o.s")

cat("Here FC is greater than Ftv2, so reject H0C at 5% of l.o.s")

cat("Here FAB is less than Ftv2, so accept H0AB at 5% of l.o.s")

cat("Here FAC is greater than Ftv2, so reject H0AC at 5% of l.o.s")

cat("Here FBC is less than Ftv2, so accept H0BC at 5% of l.o.s")

cat("Here FABC is less than Ftv2, so accept H0ABC at 5% of l.o.s")


#Q-2]

data=matrix(c(101,450,106,449,87,471,131,437,291,106,306,89,334,128,272,103,373,265,338,272,324,279,361,302,391,31
2,407,324,423,323,445,324),nrow=4,ncol=8,byrow=TRUE)

data

T1=c(101,106,87,131)

Tn=c(106,89,128,103)

Tk=c(265,272,279,302)

Tnk=c(291,306,334,272)

Tp=c(312,324,323,324)

Tnp=c(373,338,324,361)
```

```
Tkp=c(391,407,423,445)

Tnkp=c(450,449,471,437)

ST1=sum(T1)

ST1

STn=sum(Tn)

STn

STk=sum(Tk)

STk

STnk=sum(Tnk)

STnk

STp=sum(Tp)

STp

STnp=sum(Tnp)

STnp

STkp=sum(Tkp)

STkp

STnkp=sum(Tnkp)

STnkp

total=c(ST1,STn,STk,STnk,STp,STnp,STkp,STnkp)

total

yate=function(x)

{

c1=c(x[1]+x[2],x[3]+x[4],x[5]+x[6],x[7]+x[8],x[2]-x[1],x[4]-x[3],x[6]-x[5],x[8]-x[7])

return(c1)

}

c1=yate(total)

c1

c2=yate(c1)

c2

c3=yate(c2)

c3

SS=c3^2/32

SS

treat=c(1,"n","k","nk","p","np","kp","nkp")

treat

C=data.frame(treat,total,c1,c2,c3,SS)

C
```

```
SStreat=SS[2]+SS[3]+SS[4]+SS[5]+SS[6]+SS[7]

SStreat

rss=sum(data^2)

rss

y..=sum(data)

y..

CF=y..^2/32

CF

SST=rss-CF

SST

y.j=colSums(data)

SSblock=(sum(y.j^2)/4)-CF

SSblock

SSE=SST-SSblock-SStreat

SSE

SV=c("block","N","K","NK","P","NP","KP","Error","Total")

SV

DF=c(7,1,1,1,1,1,1,18,31)

DF

S_S=c(SSblock,SS[2],SS[3],SS[4],SS[5],SS[6],SS[7],SSE,SST)

S_S

MSS=S_S/DF

MSS

Fratio=MSS/MSS[8]

Fratio

anova=data.frame(SV,DF,S_S,MSS,Fratio)

anova

tvblock=qf(0.95,7,18)

tvblock

##Hypothesis

cat("H0:confounding is not effective")

cat("H1:confounding is effective")

cat("HON:N is not significant")

cat("H1N:N is significant")

cat("HOK:K is not significant")

cat("H1K:K is significant")

cat("HONK:NK is not significant")
```

```
cat("H1NK:NK is significant")

cat("HOP:P is not significant")

cat("H1P:P is significant")

cat("HONP:NP is not significant")

cat("H1NP:NP is significant")

cat("HOKP:KP is not significant")

cat("H1KP:KP is significant")

##Conclusion

tvblock=qf(0.95,7,18)

tvblock

tv1=qf(0.95,1,118)

tv1

cat("Here Fblock is greater than tvblock, so reject H0 at 5% of l.o.s")

cat("Here FN is greater than tv1, so reject H0N at 5% of l.o.s")

cat("Here FK is greater than tv1, so reject H0K at 5% of l.o.s")

cat("Here FNK is less than tv1, so accept H0NK at 5% of l.o.s")

cat("Here FP is greater than tv1, so reject H0P at 5% of l.o.s")

cat("Here FNP is less than tv1, so accept H0NP at 5% of l.o.s")

cat("Here FKP is greater than t1, so reject H0 at 5% of l.o.s")




#######################################################################################

    ##13.random effect and mix model

#######################################################################################

#Q.1)

a=5

b=5

N=a*b

m=matrix(c(23.46,23.59,23.51,23.28,23.29, 23.48,23.46,23.64,23.4,23.46,23.58,23.42,23
48,23.37,23.37,23.39,23.49,23.52,23.46.23. 32,23.4,23.5,23.49,23.39,23.38), nrow=5,nco 1=5,byrow=T)

m

Y..=sum(m)

Y..

cf=Y..^2/N

cf

Y.j-colSums(m)
```

```
Y.j
SSbatch=sum(Y.j^2)/a-cf
SSbatch
SST=sum(m^2)-cf
SST
SSE-SST-SSbatch
SSE
#Hypothesis
cat("H0: There is no significant difference between calsium content from batch to batch
against
H1: At least two batch have significance variation in calcium content")
SV=c("Batch", "Error", "Total")
DF=c(a-1,N-a,N-1)
SS=c(SSbatch, SSE, SST)
MS=SS/DF
MS
F=MS/MS[2]
F
anova data.frame("SV"=SV,"DF"=DF,"SS" =SS,"MS"=MS, "Fratio"=F)
anova
Fcritical=qf(0.95,df1=a-1,df2=N-a)
Fcritical
if (F[1]>=Fcritical)
 {

cat("Reject H0 at 5% 1.0.s, Therefore At least two batch have significance variation in calcium content")
}else
{
cat("Accept H0 at 5% 1.0.s, Therefore there is no sigificant difference between calsium content from batch to batch")
}



############Q.2)
a=4
b=4
N=a*b
m=matrix(c(98,97,99,96,91,90,93,92,96,95, 99,95,95,96,97,98),nrow=4,ncol=4,byrow=T )
```

```
m

Y..=sum(m)

Y..

cf=Y..^2/N

cf

Yi.=rowSums(m)

Yi.

SSL=sum(Yi.^2)/b-cf

SSL

Y.j=colSums(m)

Y.j

SSO=sum(Y.j^2)/a-cf

SSO

SST=sum(m^2)-cf

SST

SSE=SST-SSL-SSO

SSE

#Hypothesis

cat("HOL:There is no significant variation in strength of fabric manufactured on different looms.

against

HIL:At least two looms have significant variation in strength of fabric manufactured.

HOO:There is no significant variation in strength of fabric manufactured on different Observations.

against

H1O:At least two Observations have significant variation in strength of fabric manufactured.")

SV=c("Looms", "Observations", "Error","Tot al")

DF=c( a - 1, b - 1 ,(a-1)^ * (b-1),N-1) > SS =

c(SSL, SSO, SSE, SST) > MS = SS / D * F > MS

F=MS/MS[3]

F

anova= data.frame " SV "=SV "DF"-DF,"SS"

SS MS^ prime prime = MS "Fratio"=F)

anova

FcriticalL=qf(0.95,dfl=a-1,df2=(a-1)*(b- 1))

FcriticalL

FcriticalO=qf(0.95,dfl=b-1,df2=(a-1)*(b- 1))

FcriticalO

if (F[1]>=FcriticalL)
```

```
{
cat("Reject HOL at 5% l.o.s, Therefore There is no significant variation in strength of fabric manufactured on different looms.")
 }else
{
cat("Accept HOL at 5% l.o.s, At least two


looms have significant variation in strength of fabric manufactured")
}
Reject HOL at 5% 1.0.s, Therefore There is no significant variation in strength of fabric manufactured on different looms.> if (F[2]>=FcriticalO)
{
cat("Reject HOO at 5% 1.0.s, Therefore There is no significant variation in strength of fabric manufactured on different Observations.")
 }else
{ cat("Accept H0O at 5% 1.o.s, At least two Observations have significant variation in strength of fabric manufactured")


}



####################################################################################
        ##14.Anlysis first and second order surface model
####################################################################################
#Q.1)
install.packages("dplyr")
library(dplyr)
install.packages("ggplot2")
library(ggplot2)
install.packages("rsm")
library(rsm)
# Create data frame
data <- data.frame(
  Run = 1:10,
  X1 = c(-1, -1, 1, 1, 0, 0, 0, -1, 1, 0),
  X2 = c(-1, 1, -1, 1, 0, -1, 1, 0, 0, 0),
  Yield = c(70, 80, 85, 90, 88, 82, 87, 75, 92, 89)
)
```

```
# Fit the response model
library(rsm)
model <- rsm(Yield ~ SO(X1, X2), data = data)

# Display the summary of the model
summary(model)

# View coefficients
coefficients <- coefficients(model)
print(coefficients)

# Find optimal conditions
optimal_conditions <- canonical(model)$xs
cat("Optimal conditions: X1 =", optimal_conditions[1], "X2 =", optimal_conditions[2], "\n")

# Predict optimal yield
optimal_yield <- predict(model, newdata = data.frame(X1 = optimal_conditions[1], X2 = optimal_conditions[2]))
cat("Maximum yield =", optimal_yield, "\n")

# Create a grid of values for X1 and X2
X1_seq <- seq(-1, 1, length.out = 100)
X2_seq <- seq(-1, 1, length.out = 100)
grid <- expand.grid(X1 = X1_seq, X2 = X2_seq)

# Predict the yield on the grid
grid$yield <- predict(model, newdata = grid)

# 3D surface plot
persp(
  x = X1_seq,
  y = X2_seq,
  z = matrix(grid$yield, nrow = 100),
  theta = 30,
  phi = 30,
  expand = 0.5,
  col = "lightblue",
  xlab = "X1 (temperature)",
```

```
  ylab = "X2 (time)",

  zlab = "Yield",

  main = "Response Surface Plot"

)

#################################################################################

          ##15.central composite design,counter surface plot

#################################################################################

# Load necessary libraries

library(rsm)

library(ggplot2)


# Define the data

y = c(54, 45, 32, 477, 50, 53, 47, 51, 41, 39, 44, 42, 40)

x1 = c(-1, -1, 1, 1, -1.414, 1.414, 0, 0, 0, 0, 0, 0, 0)

x2 = c(-1, 1, -1, 1, 0, 0, -1.414, 1.414, 0, 0, 0, 0, 0)

d = data.frame(y, x1, x2)


# Fit a second-order response surface model

model = rsm(y ~ SO(x1, x2), data = d)


# Summarize the model

summary(model)


# Analysis of Variance

anova_result = anova(model)

print(anova_result)


# Create a grid of values for x1 and x2

x1_seq = seq(from = min(x1), to = max(x1), length.out = 100)

x2_seq = seq(from = min(x2), to = max(x2), length.out = 100)

grid = expand.grid(x1 = x1_seq, x2 = x2_seq)


# Predict values of y on the grid

grid$y = predict(model, newdata = grid)


# Generate a contour plot

contour_matrix = matrix(grid$y, nrow = 100, ncol = 100)
```

```r
contour(x1_seq, x2_seq, contour_matrix, xlab = "x1", ylab = "x2",
    main = "Contour Plot of y vs x1 and x2")

# Generate a surface plot
persp(x1_seq, x2_seq, contour_matrix, xlab = "x1", ylab = "x2",
   zlab = "y", main = "Surface Plot of y vs x1 and x2",
   theta = 30, phi = 30, expand = 0.5, col = "lightblue", ltheta = 120)



##################################################################################
      #16.Taguchi methods
##################################################################################
#Q.1)
# Load the dplyr library
library(dplyr)

# Define the data frame
Temperature <- c(180, 180, 180, 200, 200, 200, 220, 220, 220)
Cooling_Time <- c(10, 20, 30, 10, 20, 30, 10, 20, 30)
Pressure <- c(5, 10, 15, 10, 15, 5, 15, 5, 10)
Tensile_Strength <- c(40, 45, 50, 55, 60, 52, 62, 58, 61)
data <- data.frame(Temperature, Cooling_Time, Pressure, Tensile_Strength)

# Group and summarize data by Temperature
grouped_data1 <- group_by(data, Temperature)
avg_strength_A <- summarize(grouped_data1, Average_Strength = mean(Tensile_Strength))

# Group and summarize data by Cooling_Time
grouped_data2 <- group_by(data, Cooling_Time)
avg_strength_B <- summarize(grouped_data2, Average_Strength = mean(Tensile_Strength))

# Group and summarize data by Pressure
grouped_data3 <- group_by(data, Pressure)
avg_strength_C <- summarize(grouped_data3, Average_Strength = mean(Tensile_Strength))

# Determine optimal levels based on average strengths
optimal_A <- avg_strength_A[which.max(avg_strength_A$Average_Strength), ]
```

```
optimal_B <- avg_strength_B[which.max(avg_strength_B$Average_Strength), ]

optimal_C <- avg_strength_C[which.max(avg_strength_C$Average_Strength), ]


# Print the results

cat("Temperature:", optimal_A$Temperature, "°C\n")

cat("Cooling Time:", optimal_B$Cooling_Time, "minutes\n")

cat("Pressure:", optimal_C$Pressure, "bar\n")


###############################################################################################

        ##17-Application of central limit theorem and weak law of large number##

###############################################################################################

#Q-1]

#Weak law of large number:-

#P(|X̅-μ|>=ϵ)→0 as n→∞

#or

#P(|X̅-μ|<ϵ)=1

#P(x>n/2)>=1-α

#α=0.1

#P(x>n/2)>=0.9

#P(x<n/2)>=0.1

#x~B(n,ɵ)

#E(x)=n*ɵ

#V(x)=n*ɵ(1-ɵ)

#P[(x-μ/σ)<((n/2-n*ɵ)/√(n*ɵ(1-ɵ)))]

#[(n/2-n*ɵ)/√(n*ɵ(1-ɵ)]=-Z0.1

#[(n/2-n*ɵ)/√(n*ɵ(1-ɵ)]=-1.28

#[n(1/2-ɵ)/√(n*ɵ(1-ɵ)]=-1.28

#[√n(1/2-ɵ)/√ɵ(1-ɵ)]=-1.28

#√n=[-1.28√ɵ(1-ɵ)]/(1/2-ɵ)        [ɵ=0.45]

#n=162.1404

#n=162


#Q-2]

#TO find p(s>=300)=1-p(s<300)

l=3

n=100

p=1-ppois(300,n*1)
```

```
x=rpois(n,1)

x

m=mean(x)

m

v=(n-1/n)*(var(x))

v

mean=n*m

mean

var=n*v

var

z=(300-mean)/sqrt(var)

z

p1=pnorm(z,0,1)

p1

#Q-3]

n=50

theta=0.45

p=pbinom(n,1,theta)

p

x=rbinom(n,1,theta)

x

m=mean(x)

m

v=(n-1/n)*(var(x))

v

mean=n*m

mean

var=n*v

var

#prob(s>=30)

s=(30-mean)/sqrt(var)

s

p=1-pnorm(s,n,theta)

p

#prob(s<=10)

t=(10-mean)/sqrt(var)

t
```

```
p=pnorm(t,n,theta)

p

p1=pbinom(10,n,theta)

p1


###################################################################################

        ##18.application and verification of weak law large number

###################################################################################

# Q.1: Simulating coin tosses

set.seed(123)

n <- 1000

tosses <- sample(c(0, 1), size = n, replace = TRUE, prob = c(0.5, 0.5))


# Calculating the sample mean (proportion of heads)

sample_mean <- cumsum(tosses) / (1:n)


# Plotting the sample mean as n increases

plot(1:n, sample_mean, type = "l", ylab = "Proportion of Heads", xlab = "Number of Tosses")

abline(h = 0.5, col = "red") # True mean line


# Probability of deviation by more than 0.05

epsilon <- 0.05

prob_dev <- mean(abs(sample_mean - 0.5) >= epsilon)

cat("Probability of deviation by more than 0.05:", prob_dev, "\n")


# Q.2: Simulating customer arrivals using Poisson distribution

n <- 1000

lambda <- 5

arrivals <- rpois(n, lambda)


# a) Calculate the cumulative sample mean over time

cumulative_mean <- cumsum(arrivals) / (1:n)


# b) Plot the cumulative mean to show convergence to the expected mean

plot(1:n, cumulative_mean, type = "l",

  xlab = "Minutes Observed", ylab = "Cumulative Sample Mean",

  main = "Convergence of Sample Mean to Expected Mean")
```

```
abline(h = lambda, col = "red") # Expected mean line


# c) Calculate the final sample mean and print it

final_sample_mean <- mean(arrivals)

cat("Final Sample Mean:", final_sample_mean, "\n")


# Q.3: Simulating student heights using normal distribution

set.seed(123)

n <- 1000

mean_height <- 170

sd_height <- 10

heights <- rnorm(n, mean_height, sd_height)


# a) Calculate the cumulative sample mean over time

cumulative_mean_height <- cumsum(heights) / (1:n)


# b) Plot the cumulative sample mean to show convergence to the true mean

plot(1:n, cumulative_mean_height, type = "l",

    xlab = "Number of Students Measured", ylab = "Cumulative Average Height",

    main = "Convergence of Sample Mean to True Mean Height")

abline(h = mean_height, col = "red") # True mean height line


# c) Calculate and print the final sample mean

final_sample_mean_height <- mean(heights)

cat("Final Sample Mean Height:", final_sample_mean_height, "cm\n")


# Q.4: Simulating defect rates

n <- 5000

TDR <- 0.02 # True defect rate (2%)


# Generate defect data: 1 for defect, 0 for no defect (Bernoulli trials)

defects <- rbinom(n, 1, TDR)


# a) Calculate the cumulative defect rate

CDR <- cumsum(defects) / (1:n)


# b) Plot the cumulative defect rate and compare with the 2% true rate
```

```
plot(1:n, CDR, type = "l",
    xlab = "Number of Components Inspected",
    ylab = "Cumulative Defect Rate",
    main = "WLLN: Convergence of Defect Rate")
abline(h = TDR, col = "red") # True defect rate line

# c) Probability of deviation by more than 0.5%
epsilon <- 0.005 # 0.5% deviation threshold
prob_deviation <- mean(abs(CDR - TDR) >= epsilon)
cat("Probability of deviation by more than 0.5%:", prob_deviation, "\n")

####################################################################################
              #19.modes of convergence
####################################################################################
# Q.1) Pointwise Convergence

# Define the function
fn = function(n, x) {
  x / n
}

# Calculate f_n for n = 1/5 and x = 0, 1, 2
n_values = 1/5
x_values = c(0, 1, 2)
results = outer(n_values, x_values, Vectorize(fn))

# Plot the pointwise convergence
matplot(n_values, results, type = "b", pch = 19, col = 1:5,
      xlab = "n", ylab = "f_n(x)", main = "Pointwise Convergence of f_n(x) = x / n")

# As n tends to infinity, f_n tends to zero for each value of x, hence this is pointwise convergence.



# Q.2) Uniform Convergence

# Define the function
g_n = function(n, x) {
```

```
  x^2 / n
}

# Calculate g_n for n = 1, 2, 3 and x in [0,1] at intervals
x_values = seq(0, 1, length.out = 100)
n_values = 1:3
results = outer(n_values, x_values, Vectorize(g_n))

# Generate a fine grid of x values in [0,1]
x_values = seq(0, 1, length.out = 100)

# Calculate the maximum value of |g_n(x) - 0| for each n
max_values = sapply(n_values, function(n) max(g_n(n, x_values)))

# Display the maximum values for different n
cat("Max values of |g_n(x) - 0| for n = 1, 2, ..., 10:\n")
print(max_values)

# Plot max |g_n(x) - 0| vs n to observe convergence
plot(n_values, max_values, type = "b", pch = 19, col = "blue",
   xlab = "n", ylab = "Max |g_n(x) - 0|", main = "Checking Uniform Convergence of g_n(x) = x^2 / n")

# Hence, the sequence of functions converge to 0 uniformly over [0,1]


# Q.3) Almost Sure Convergence

# Define probabilities for almost sure convergence
n_values = 1:3

# Calculate P(X_n = 1)
p1 = 1 / n_values
cat("P(X_n = 1) for n = 1, 2, 3: \n")
print(p1)

# Calculate P(X_n = 0)
p2 = 1 - p1
```

```
cat("P(X_n = 0) for n = 1, 2, 3: \n")

print(p2)


# As n tends to infinity, P1 tends to zero, hence it is almost sure convergence.

#but for p2 as n tends to infinity p2 does not tends to zero hence it is not almost sure convergence
```