**EXP -1**

#1-1 OPERATION ON VARIABLES

#1-1-1 Assigning values to variables

x=c(1,2,5)

cat("x=",x,"\n")

x<-c(1,2,5)

cat("x=",x,"\n")

x<-c("a","b")

cat("x=",x,"\n")

x<-as.numeric(1:5)

cat("x=",x,"\n")

x<-seq(1,5)

cat("x=",x,"\n")

x=1:5

cat("x=",x,"\n")

#1-1-2 Scalar Multiplication

x=c(1,2,3,4,5,2,3,-1)

cat("x=",x,"\n")

cat("2\*x=",2\*x,"\n")

#1-1-3 Addition of two variables

x=c(1,2,3,4,5,2,3,-1)

y=c(x,0,1,2,x)

cat("x=",x,"\n")

cat("y=",y,"\n")

cat("x+y=",x+y,"\n")

#1-1-4 Difference of two variables

x=c(1,2,3,4,5,2,3,-1)

y=c(2,0,1,2,3,1,4,0)

cat("x-y=",x-y,"\n")

#1-1-5 Product of two variables

x=c(1,2,3)

y=c(4,5,6)

cat("x=",x,"\n")

cat("y=",y,"\n")

cat("x\*y=",x\*y,"\n")

#1-1-6 Division of two variables

x=c(1,2,3)

y=c(4,5,6)

cat("x/y=",x/y,"\n")

#1-1-7 Find the Sum of the variables

x=c(1,2,3,4,5,2,3,-1)

cat("sum(x)=",sum(x),"\n")

y=c(2,0,1,2,3,1,4,0)

cat("sum(y)=",sum(y),"\n")

#1-1-8 Find the Length of the variable

x=c(1,5,3,2,4,6,7,8)

cat("length(x)=",length(x),"\n")

#1-1-9 Find the Product value of the variable

x=c(1,2,3,4,5,2,3,-1)

cat("prod(x)=",prod(x),"\n")

#1-1-10 Find the Minimum, Maximum value of the variable

x=c(25,32,22,15,27)

cat("min(x)=",min(x),"\n")

cat("max(x)=",max(x),"\n")

#1-1-11 Find the Range of the variable

x=c(1,2,3,4,5,2,3,-1)

cat("range(x)=",range(x),"\n")

#1-1-12 Find the Square root value

x=c(52,49,16,40)

cat("sqrt(x)=",sqrt(x),"\n")

#1-1-13 Arrange the value either/ascending/decending of the variable

x=c(1,5,7,9,-1,-5)

cat("sort(x)=",sort(x),"\n")

cat("sort(x),decreasing=T)=",sort(x,decreasing = T),"\n")

#1-1-14 Find the Cummulative sum/product/min/max of the variable

a=c(3:5,-3:-1,11:13)

cat("a=",a,"\n")

cat("cumsum(a)=",cumsum(a),"\n")

cat("cumprod(a)=",cumprod(a),"\n")

cat("cummin(a)=",cummin(a),"\n")

cat("cummax(a)=",cummax(a),"\n")

#1-1-15 Sequence & repitition functions

cat("seq(1,5)=",seq(1,5),"\n")

cat("seq(from=-5,to=5,by=2)=",seq(from=-5,to=5,by=2),"\n")

cat("seq(length=10,from=0,by=2)=",seq(length=10,from=0,by=2),"\n")

x=c(1:3,5:9)

cat("x=",x,"\n")

cat("rep(x,times=2)=",rep(x,times=2),"\n")

cat("rep(x,each=2)=",rep(x,each=2), "\n")

**EXP -2**

#1-2-1 Assigning values to the matrix

A=matrix(c(1,2,3,4),2,2)

B=matrix(c(1,2,3,4),2,2,byrow = T)

Print(A)

#1-2-2 Assigning Labels to Rows & Columns

A=matrix(c(1,2,3,4),2,2,dimnames = list(c("A","B"),c("C","D")))

B=matrix(c(1:4),1,4,dimnames = list(c(1),LETTERS[1:4]))

print(B)

#1-2-3 Add Row & Column of the matrix

A=matrix(c(1,2,3,4,5,6),3,2,byrow = T)

print(A)

B=matrix(c(1,2),1,2)

print(B)

C=rbind(A,B)

print(C)

D=cbind(C,c(2,3,4,5))

print(D)

#1-2-4 Find Transpose of the matrix

A=matrix(c(1,3,5,2,4,6,2,3,4,3,5,4),3,4)

print(A)

print(t(A))

#1-2-5 Find Inverse of the matrix

B=matrix(c(1,2,3,4),2,2)

print(B)

I<-solve(B)

print(I)

#1-2-6 Round the values of the matrix

A=matrix(c(1,2,3,4),2,2)

print(A)

print(round(A%\*%solve(A)))

#1-2-7 Add,Subtract and Multipliation the matrix

A=matrix(c(1,2,3,4),2,2)

B=matrix(c(5,6,7,8),2,2)

print(A+B)

print(A-B)

print(A%\*%B)

#1-2-8 Assign Diagonal value of the matrix

D=diag(1:4)

print(D)

**EXP -3**

A=var(iris[,-5])

A=A[-5,]

A

B=A[1:2,1:2]

C=A[1:2,3:4]

D=A[3:4,1:2]

E=A[3:4,3:4]

X=D%\*%solve(B)%\*%C

da=det(B)\*det(E-X)

da

det(A)

**EXP -5**

set.seed(2022)

x=sample(20:80,100,replace=T)

x

y=vector()

for(i in 1:length(x))

{ if(x[i]>=50)

{

y[i]=c("the student is pass")

}

else

{

y[i]=c("the student is fail")

}

}

y

da=data.frame(x,y)

da

set.seed(2025)

x = sample(20:80, 100, replace = TRUE)

y = ifelse(x >= 50, "The student is pass", "The student is fail")

da = data.frame(x,y)

print(da)

**EXP -6**

set.seed(2022)

x=sample(30:100,200,replace=T)

x

y=vector()

for( i in 1:length(x))

{

if(x[i]<50)

{

y[i]=c("fail")

}

else

{

if(x[i]>=50&x[i]<60)

{

y[i]=c("second class")

}

else

{

if(x[i]>=60 & x[i]<70)

{

y[i]=c("first class")

}

else

{

if(x[i]>=70 & x[i]<80)

{

y[i]=c("first class with distinction")

}

else

{

if(x[i]>=80)

{

y[i]=c("outstanding")

}}}}}}

y

a=data.frame(x,y)

gender=c(sample(c("m","f"),size=200,rep=T))

gender

a=data.frame(x,y,gender)

a[which(a[,3]==”m”),] or f for female

**EXP -7**

set.seed(2025)

coin <- c("H","T")

result <- sample(coin,10, replace = TRUE)

table(result)

dftotal=data.frame(table(result))

dfprob <- cbind(dftotal, prop.table(dftotal$Freq))

names(dfprob) <- c("result","Freq", "probability")

dfprob

##Obtaining the results graphically

barplot(dfprob$Freq, names.arg = dftotal$result, main = "Head & Tail count of Tossing a coin 10

times",col=c('red','blue'))

barplot(dfprob$probability, names.arg = dftotal$result, main = "Probability of Tossing a coin 10 times",col=c('red','blue'))

**EXP -8**

set.seed(2022)

tosscoin = function(numToss) {

coin = c("H", "T")

result = sample(coin, numToss, replace = TRUE)

dftotal = data.frame(table(result))

dfprob = cbind(dftotal, prop.table(dftotal$Freq))

names(dfprob) = c("result", "Freq", "Probability")

return(dfprob)

}

n = 50

coin = tosscoin(n)

coin

barplot(coin$Freq, names.arg = coin$result, main = paste("tossing a coin", n, "times"), col = c("red", "blue"))

barplot(coin$Probability, names.arg = coin$result, main = paste("Probability of tossing a coin", n, "times"), col= c('red', 'blue'))

#9

set.seed(2022)

tossdice=function(numtoss)

{dice=c(1:6)

result=sample(dice,numtoss,replace=TRUE)

dftotal=data.frame(table(result))

return(dftotal) }

#function to plot

plotDice= function(dice)

{ par(mfrow =c(2,1))

barplot(dice$Freq,names.arg = dice$result, main=paste("count of tossing a dice",n,"times"),col=sample(657,6))

barplot(prop.table(dice$Freq),names.arg=dice$result,main=paste("probability of tossing a coin",n,"times"),col=sample(657,6))

}

set.seed(2022)

n=50

dice = tossDice (n)

dice

plotDice(dice)

exp10

#what is the probability that a five cards hand selected from a well shuffled pack of cards contains 3 aces

cn=function(n,x)

{ c1=factorial(n)/(factorial(x)\*factorial(n-x))

return(c1) }

pron=(cn(4,3)\*cn(4,2))/cn(52,5)

pron

#a commitee of university teachers consist of -------- three readers and one lecture?

prob=(cn(3,2)\*cn(5,3)\*cn(2,1))/cn(10,6)

prob

exp11

set.seed(3000)

p=1/6

n1=100

d1=sample(1:6,n1,rep=T)

d1

f1=table(d1)

f1

p1=f1/n1

paste("random samples of size 100 is:",p1)

set.seed(3000)

p=1/6

n2=200

d2=sample(1:6,n2,rep=T)

d2

f2=table(d2)

f2

p2=f2/n2

paste("random samples of size 200 is:",p2)

set.seed(3000)

p=1/6

n3=300

d3=sample(1:6,n3,rep=T)

d3

f3=table(d3)

f3

p3=f3/n3

paste("random samples of size 500 is:",p3)

exp12

#i)

x=0:8

fx=c(0,2/34,6/34,7/34,9/34,4/34,1/34,4/34,1/34)

plot(x,fx,type="h",main="no. of flood occurences")

points(x,fx,pch=10)

#ii)

Fx= cumsum(fx)

cdf= data.frame (x,Fx)

cdf

plot(x, Fx, type="s", main= "CDF of flood occurences")

point3 (x,Fx,pch=16)

exp4

(i)name and suite of every card

Cbind(c(“ace”,”king”,”queen”,”jack”,”ten”),C(“spades”,”spades”,”spades”,”spades”,”spades”))

(ii) list to matrix

Mylist=c(1,2,3,4,5,6,7,8,9,10)

Myslist

Ls1=list( list(1,2,3),list(4,5,6),list(7,8,9))

Mt1=matrix(unlist(ls1),nrow=3,byrow=TRUE)

Mt1

Students=list(1:10)

H=list(c(height of students))

W=list(c(weight of students))

Af=data.Frame(unlist(students),unlist(h),unlist(w))

Names(af)=c(“students”,”height”,”weight”)

Cat(“the dataframe is:\n”)

Print(af)

Final\_list=c(mylist,mt1,af)

Final\_list

OR

A=1:10

B=matrix(1:9,3,3)

C=data.frame(a,b,height=c(---),weight=c(---))

Out=list(a.b.c)

out