PROBABILITY AND STATISTICS

EXPERIMENT-1

(1)
```
c<-c(5,10,15,20,25,30);
MAX<-max(c)
MIN<-min(c)
print(paste("The max value is ",MAX))
print(paste("The min value is ",MIN))
```

```
> c<-c(5,10,15,20,25,30);
> MAX<-max(c)
> MIN<-min(c)
> print(paste("The max value is ",MAX))
[1] "The max value is  30"
> print(paste("The min value is ",MIN))
[1] "The min value is  5"
```

(2)
```
n=as.integer(readline(prompt = "enter a number:"))
fact=1
if(n<0){
 print(paste("The number is negative"))
}else if(n==0){
 print("The factorial of 0 is 1")
}else{
 for(i in 1:n){
  fact=fact*i
 }
 print(paste("The factorial of ",n," is ",fact))
}
```

a) Positive Number as input:

```
> n=as.integer(readline(prompt = "enter a number:"))
enter a number:8
> fact=1
> if(n<0){
+   print(paste("The number is negative"))
+ }else if(n==0){
+   print("The factorial of 0 is 1")
+ }else{
+   for(i in 1:n){
+     fact=fact*i
+   }
+   print(paste("The factorial of ",n," is ",fact))
+ }
[1] "The factorial of  8  is  40320"
```

b) Negative Number as input:

```
> n=as.integer(readline(prompt = "enter a number:"))
enter a number:-4
> fact=1
> if(n<0){
+   print(paste("The number is negative"))
+ }else if(n==0){
+   print("The factorial of 0 is 1")
+ }else{
+   for(i in 1:n){
+     fact=fact*i
+   }
+   print(paste("The factorial of ",n," is ",fact))
+ }
[1] "The number is negative"
```

(3)
```
m=as.integer(readline(prompt = "enter a number:"))
fibo<-vector();
fibo[1]=0;
fibo[2]=1;
for(i in 3:m){
```

```
> m=as.integer(readline(prompt = "enter a number:"))
enter a number:7
> fibo<-vector();
> fibo[1]=0;
> fibo[2]=1;
> for(i in 3:m){
+   fibo[i]=fibo[i-1]+fibo[i-2];
+ }
> print("The fibonaci sequence is ")
[1] "The fibonaci sequence is "
> print(fibo)
[1] 0 1 1 2 3 5 8
```

```
 fibo[i]=fibo[i-1]+fibo[i-2];
}
print("The fibonaci sequence is ")
print(fibo)
```

(4)
```
add<-function(x,y){
 return(x+y)
}
sub<-function(x,y){
 return(x-y)
}
mult<-function(x,y){
 return(x*y)
}
div<-function(x,y){
 return(x/y)
}
num1=as.integer(readline(prompt = "enter a number:"))
num2=as.integer(readline(prompt = "enter a number:"))
print(" Enter Choice")
print("1. Add")
print("2. Subtraction")
print("3. Multiplication")
print("4. Division")
choice=as.integer(readline(prompt = "enter a number:"))
result<-switch(choice,add(num1,num2),sub(num1,num2),mult(num1,num2),div(num1,num2
))
print(paste("Result is: ",result))
```

a) Addition:

```
> num1=as.integer(readline(prompt = "enter a number:"))
enter a number:10
> num2=as.integer(readline(prompt = "enter a number:"))
enter a number:2
> print("1. Add")
[1] "1. Add"
> print("2. Subtraction")
[1] "2. Subtraction"
> print("3. Multiplication")
[1] "3. Multiplication"
> print("4. Division")
[1] "4. Division"
> print(" Enter Choice")
[1] " Enter Choice"
> choice=as.integer(readline(prompt = "enter a number:"))
enter a number:1
> result<-switch(choice,add(num1,num2),sub(num1,num2),mult(num1,num2),div(n
um2))
> print(paste("Result is: ",result))
[1] "Result is:  12"
```

b) Subtraction:

```
> num1=as.integer(readline(prompt = "enter a number:"))
enter a number:5
> num2=as.integer(readline(prompt = "enter a number:"))
enter a number:9
> print(" Enter Choice")
[1] " Enter Choice"
> print("1. Add")
[1] "1. Add"
> print("2. Subtraction")
[1] "2. Subtraction"
> print("3. Multiplication")
[1] "3. Multiplication"
> print("4. Division")
[1] "4. Division"
> choice=as.integer(readline(prompt = "enter a number:"))
enter a number:2
> result<-switch(choice,add(num1,num2),sub(num1,num2),mult(num1,num2),div(num1,n
um2))
> print(paste("Result is: ",result))
[1] "Result is:  -4"
```

c)Multiplication:

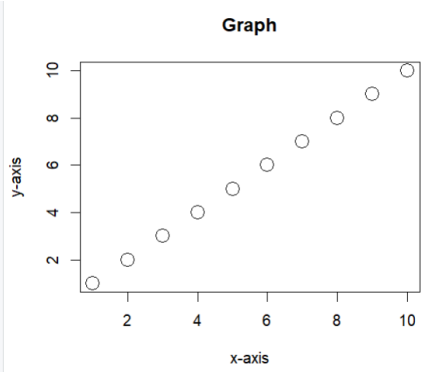
d) Division:

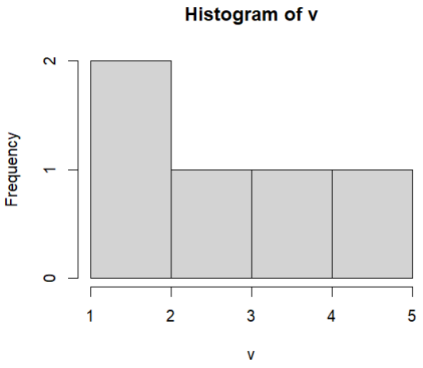
```
> num1=as.integer(readline(prompt = "enter a number:"))
enter a number:23
> num2=as.integer(readline(prompt = "enter a number:"))
enter a number:-2
> print(" Enter Choice")
[1] " Enter Choice"
> print("1. Add")
[1] "1. Add"
> print("2. Subtraction")
[1] "2. Subtraction"
> print("3. Multiplication")
[1] "3. Multiplication"
> print("4. Division")
[1] "4. Division"
> choice=as.integer(readline(prompt = "enter a number:"))
enter a number:3
> result<-switch(choice,add(num1,num2),sub(num1,num2),mult(num1,num2),div(num1,n
um2))
> print(paste("Result is: ",result))
[1] "Result is:  -46"
```

```
> num1=as.integer(readline(prompt = "enter a number:"))
enter a number:9
> num2=as.integer(readline(prompt = "enter a number:"))
enter a number:6
> print(" Enter Choice")
[1] " Enter Choice"
> print("1. Add")
[1] "1. Add"
> print("2. Subtraction")
[1] "2. Subtraction"
> print("3. Multiplication")
[1] "3. Multiplication"
> print("4. Division")
[1] "4. Division"
> choice=as.integer(readline(prompt = "enter a number:"))
enter a number:4
> result<-switch(choice,add(num1,num2),sub(num1,num2),mult(num1,num2),div(num1,n
um2))
> print(paste("Result is: ",result))
[1] "Result is:  1.5"
```

(5) #Histogram

```
v<-c(1,2,3,4,5)
hist(v)
#plot()
plot(1:10,main="Graph",xlab="x-axis",ylab="y-axis",cex=2)
```





PROBABILITY AND STATISTICS

EXPERIMENT-2

(1) chest<-c(rep("gold",1),rep("silver",1),rep("bronze",1))
sample(chest,10,replace=TRUE,prob=c(0.2,0.3,0.5)) #--> by default its without replacement
#so we will write in it if we want replacement of objects true
sample(c("success","failure"),10,replace=TRUE,prob=c(0.9,0.1))

```
> chest<-c(rep("gold",1),rep("silver",1),rep("bronze",1))
> sample(chest,10,replace=TRUE,prob=c(0.2,0.3,0.5))  #--> by default its without replacement
so we will write in it if we want replacement of objects true
 [1] "silver" "bronze" "silver" "silver" "bronze" "bronze" "gold"   "bronze" "bronze"
[10] "bronze"
> sample(c("success","failure"),10,replace=TRUE,prob=c(0.9,0.1))
 [1] "failure" "failure" "success" "success" "success" "success" "success" "success"
 [9] "success" "success"
```

(2) n<-2
while(n>=2){
prod<-1
for(i in 1:(n-1)){
prod=prod*(1-(i/365))
}

prob<-1-prod

if(prob>0.5){

break

}

n=n+1

}

print(paste(prob))

print(paste(n))

```
> print(paste(prob))
[1] "0.507297234323985"
> print(paste(n))
[1] "23"
```

(3) func<-function(c,r,cr){
rc<-r*cr/c
return(rc)
}
c<-0.4
r<-0.2

```
> result<-func(c,r,cr)
> print(paste("Probability that it will rain that day: ",result))
[1] "Probability that it will rain that day:  0.425"
```

```
cr<-0.85
result<-func(c,r,cr)
print(paste("Probability that it will rain that day: ",result))
```

(4) data1<-iris

a) head(data1,5)

```
> data1<-iris
> head(data1,5) #by default gives first 6
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
```

b) str(data1)

```
> str(data1)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

c) range(data1$Sepal.Length)

d) mean(data1$Sepal.Length)

e) median(data1$Sepal.Length)

```
> range(data1$Sepal.Length)
[1] 4.3 7.9
> mean(data1$Sepal.Length)
[1] 5.843333
> median(data1$Sepal.Length)
[1] 5.8
```

f) quantile(data1$Sepal.Length,0.25)
quantile(data1$Sepal.Length,0.75)
IQR(data1$Sepal.Length) #for inter quartile range

```
> quantile(data1$Sepal.Length,0.25)
25%
5.1
> quantile(data1$Sepal.Length,0.75)
75%
6.4
> IQR(data1$Sepal.Length)  #for inter quartile range
[1] 1.3
```

g) sd(data1$Sepal.Length)
var(data1$Sepal.Length)

```
> sd(data1$Sepal.Length)
[1] 0.8280661
> var(data1$Sepal.Length)
[1] 0.6856935
```

h) lapply(data1[,2:4],sd)
lapply(data1[,2:4],var)
lapply(data1[,2:4],mean)

```
> lapply(data1[,2:4],sd)
$Sepal.Width
[1] 0.4358663

$Petal.Length
[1] 1.765298

$Petal.Width
[1] 0.7622377

> lapply(data1[,2:4],var)
$Sepal.Width
[1] 0.1899794

$Petal.Length
[1] 3.116278

$Petal.Width
[1] 0.5810063

> lapply(data1[,2:4],mean)
$Sepal.Width
[1] 3.057333

$Petal.Length
[1] 3.758

$Petal.Width
[1] 1.199333
```

i) summary(data1)

```
> summary(data1)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

(5)
```
mode<-function(v){
uniq<-unique(v)
uniq[which.max(tabulate(match(v,uniq)))]
}
v<-c(2,1,2,3,4,2,3,1,2)
modeval<-mode(v)
print(paste("Mode Value: ",modeval))
```

```
>   uniq<-unique(v)
> match(v,uniq)
[1] 1 2 1 3 4 1 3 2 1
> tabulate(match(v,uniq))
[1] 4 2 2 1
> max(tabulate(match(v,uniq)))
[1] 4
>   uniq[which.max(tabulate(match(v,uniq)))]
[1] 2
> print(paste("Mode Value: ",modeval))
[1] "Mode Value:  2"
```

PROBABILITY AND STATISTICS

EXPERIMENT-3

(1) n<-12

p<-1/6

y<-dbinom(6,n,p)

print(paste("The probability of getting 6: ",y))

px<-dbinom(7,n,p)+dbinom(8,n,p)+dbinom(9,n,p)

print(paste("probability of getting 7,8 or 9 using dbinom: ",px))

vx<-pbinom(9,n,p,lower.tail=T)-pbinom(6,n,p,lower.tail=T)

print(paste("probability of getting 7,8 or 9 using pbinom: ",vx))

```
17  n<-12
18  p<-1/6
19  y<-dbinom(6,n,p)
20  print(paste("The probability of getting 6: ",y))
21  px<-dbinom(7,n,p)+dbinom(8,n,p)+dbinom(9,n,p)
22  print(paste("probability of getting 7,8 or 9 using dbinom: ",px))
23  vx<-pbinom(9,n,p,lower.tail=T)-pbinom(6,n,p,lower.tail=T)
24  print(paste("probability of getting 7,8 or 9 using pbinom: ",vx))
```

5:1 (Top Level) R S

Console Terminal Background Jobs

R 4.3.1 · ~/3rdYear/Prob and Stats/Assignments/

```
> n<-12
> p<-1/6
> y<-dbinom(6,n,p)
> print(paste("The probability of getting 6: ",y))
[1] "The probability of getting 6:  0.006632495937343"
> px<-dbinom(7,n,p)+dbinom(8,n,p)+dbinom(9,n,p)
> print(paste("probability of getting 7,8 or 9 using dbinom: ",px))
[1] "probability of getting 7,8 or 9 using dbinom:  0.00129175754208252"
> vx<-pbinom(9,n,p,lower.tail=T)-pbinom(6,n,p,lower.tail=T)
> print(paste("probability of getting 7,8 or 9 using pbinom: ",vx))
[1] "probability of getting 7,8 or 9 using pbinom:  0.00129175754208255"
```

(2) u<-72

s<-15.2

pn<-pnorm(84,u,s,lower.tail=F) #for 84 or more, lower.tail=false

print(paste("% of students scoring 84 or more: ",pn))

```
> u<-72
> s<-15.2
> pn<-pnorm(84,u,s,lower.tail=F) #for 84 or more, lower.tail=false
> print(paste("% of students scoring 84 or more: ",pn))
[1] "% of students scoring 84 or more:  0.214917602311272"
>
```

(3) l<-5

```
q<-dpois(0,l)
print(paste("Probability of no cars arriving: ",q))
l1<-50
q1<-dpois(48,l1)+dpois(49,l1)+dpois(50,l1)
q2<-ppois(50,l1,lower.tail=T)-ppois(47,l1,lower.tail=T)
print(paste("probability of 48<=customers<=50 using dpois: ",q1))
print(paste("probability of 48<=customers<=50 using ppois: ",q2))
```

```
[1] "Probability of no cars arriving:  0.00673794699908547"
> l1<-50
> q1<-dpois(48,l1)+dpois(49,l1)+dpois(50,l1)
> q2<-ppois(50,l1,lower.tail=T)-ppois(47,l1,lower.tail=T)
> print(paste("probability of 48<=customers<=50 using dpois: ",q1))
[1] "probability of 48<=customers<=50 using dpois:  0.167848518849069"
> print(paste("probability of 48<=customers<=50 using ppois: ",q2))
[1] "probability of 48<=customers<=50 using ppois:  0.167848518849069"
```

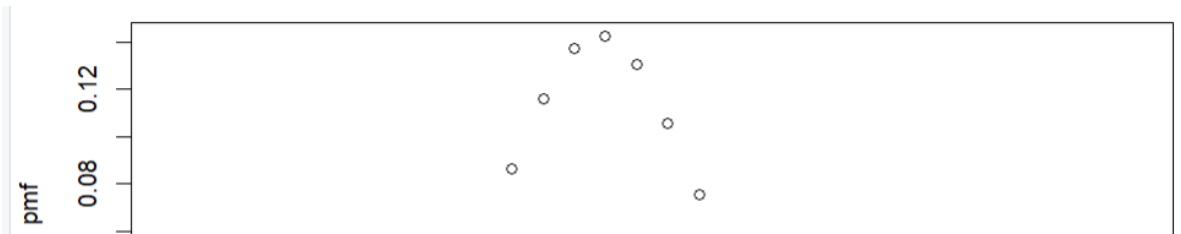
(4)
```
x<-3
M<-17
N<-250
m<-5
h<-dhyper(x,M,N-M,m)
print(paste("Probability of exactly 3 defectives: ",h))
```

```
> h<-dhyper(x,M,N-M,m)
> print(paste("Probability of exactly 3 defectives: ",h))
[1] "Probability of exactly 3 defectives:  0.00235115343595976"
```

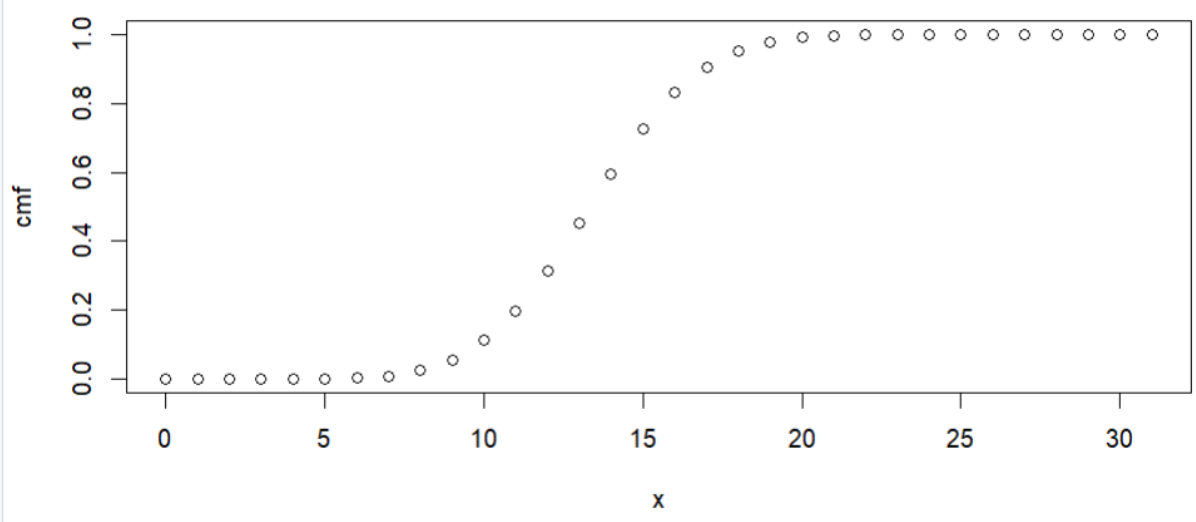
(5)
```
prob<-0.447
n1<-31
x<-seq(0,31,1)
pmf<-c()
for(i in 1:length(x)){
  pmf[i]=dbinom(x[i],n1,prob)
}
plot(x,pmf)
cmf<-c()
for(i in 1:length(x)){
  cmf[i]=pbinom(x[i],n1,prob)
}
plot(x,cmf)
mean<-n1*prob
var<-n1*prob*(1-prob)
sd<-sqrt(var)
print(paste("mean,variance and std deviation are: ",mean,var,sd))
```

(a) Binomial distribution.

(b) Probability Mass Function:



(c) Cumulative Distribution Function:



(d) Mean, variance and standard deviation of X:

```
> mean<-n1*prob
> var<-n1*prob*(1-prob)
> sd<-sqrt(var)
> print(paste("mean,variance and std deviation are: ",mean,var,sd))
[1] "mean,variance and std deviation are:  13.857 7.662921 2.76819815042204"
```

PROBABILITY AND STATISTICS

EXPERIMENT-4

(1)
```
x=c(0,1,2,3,4)
p=c(0.41,0.37,0.16,0.05,0.01)
average<-sum(x*p)
weighted.mean(x,p)
```

```
> average<-sum(x*p)
> weighted.mean(x,p)
[1] 0.88
>
```

(2)
```
g<-function(t){
t*0.1*exp(-0.1*t)
}
expected=integrate(g,lower=0,upper=Inf)
print(expected)
print(paste("The expected value of T: ",expected$value))
```

```
> print(expected)
10 with absolute error < 6.7e-05
> print(paste("The expected value of T: ",expected$value))
[1] "The expected value of T:  9.99999999969787"
>
```

(3)
```
x1=c(0,1,2,3)
px=c(0.1,0.2,0.2,0.5)
y<-(12*x1)+2*(3-x1)-18
Avg<-sum(y*px)
print(Avg)
weighted.mean(y*px)
```

```
> Avg<-sum(y*px)
> print(Avg)
[1] 9
> weighted.mean(y*px)
[1] 2.25
>
```

(4)
```
g1=function(x){
x*0.5*exp(-abs(x)) }
M1=integrate(g1,lower=1,upper=10)
print(M1)
M3=(M1$value)
print(paste("First Moment or Mean: ",M3))
g2=function(x){
x*x*0.5*exp(-abs(x))
}
M2=integrate(g2,lower=1,upper=10)
print(M2)
M4=(M2$value)
print(paste("Second Moment: ",M4))
variance<-M4-(M3^2)
```

print(paste("Variance using the moments: ",variance))

```
> print(paste("First Moment or Mean: ",M3))
[1] "First Moment or Mean:  0.367629741557749"
> g2=function(x){
+   x*x*0.5*exp(-abs(x))
+ }
> M2=integrate(g2,lower=1,upper=10)
> print(M2)
0.9169292 with absolute error < 6e-13
> M4=(M2$value)
> print(paste("Second Moment: ",M4))
[1] "Second Moment:  0.916929207213094"
> variance<-M4-(M3^2)
> print(paste("Variance using the moments: ",variance))
[1] "Variance using the moments:  0.781777580335277"
```

(5) f<-function(y){ (3/4)*((1/4)^(sqrt(y)-1)) }
ans<-f(3*3)
print(ans)
x<-c(1,2,3,4,5)
y=x*x
mean<-sum(y*f(y))
print(mean)
z<-(y-mean)^2
var<-sum(z*f(z))
print(var)

```
> ans<-f(3*3)
> print(ans)
[1] 0.046875
> x<-c(1,2,3,4,5)
> y=x*x
> mean<-sum(y*f(y))
> print(mean)
[1] 2.182617
> z<-(y-mean)^2
> var<-sum(z*f(z))
> print(var)
[1] 1.623002
>
```