

Week – 1

Aim: To understand the basics of R- Programming Language.

Program Code:

```
2+3  
1+2*3  
c(1,2,3)  
2^3^5  
2^(3^5)  
(2^3)^5  
sqrt(2)  
1:10  
-5:5  
x=22  
y=x+25  
y  
a=c(1,2,3,4)  
b=c(1,3,5,7)  
a+b  
a-b  
5*a  
2*a+1  
u=c(10,20,30)  
v=c(1,2,3,4,5,6,7,8,9)  
u+v  
n=c(2,3,5)  
s=c("aa","bb","cc")  
b=c(TRUE,FALSE,TRUE)  
df=data.frame(n,s,b)  
df
```

```
1 2+3
2 1+2*3
3 c(1,2,3)
4 2^3^5
5 2^(3^5)
6 (2^3)^5
7 sqrt(2)
8 1:10
9 -5:5
10 x=22
11 y=x+25
12 y
13 a=c(1,2,3,4)
14 b=c(1,3,5,7)
15 a+b
16 a-b
17 5*a
18 2*a+1
19 u=c(10,20,30)
20 v=c(1,2,3,4,5,6,7,8,9)
21 u+v
```

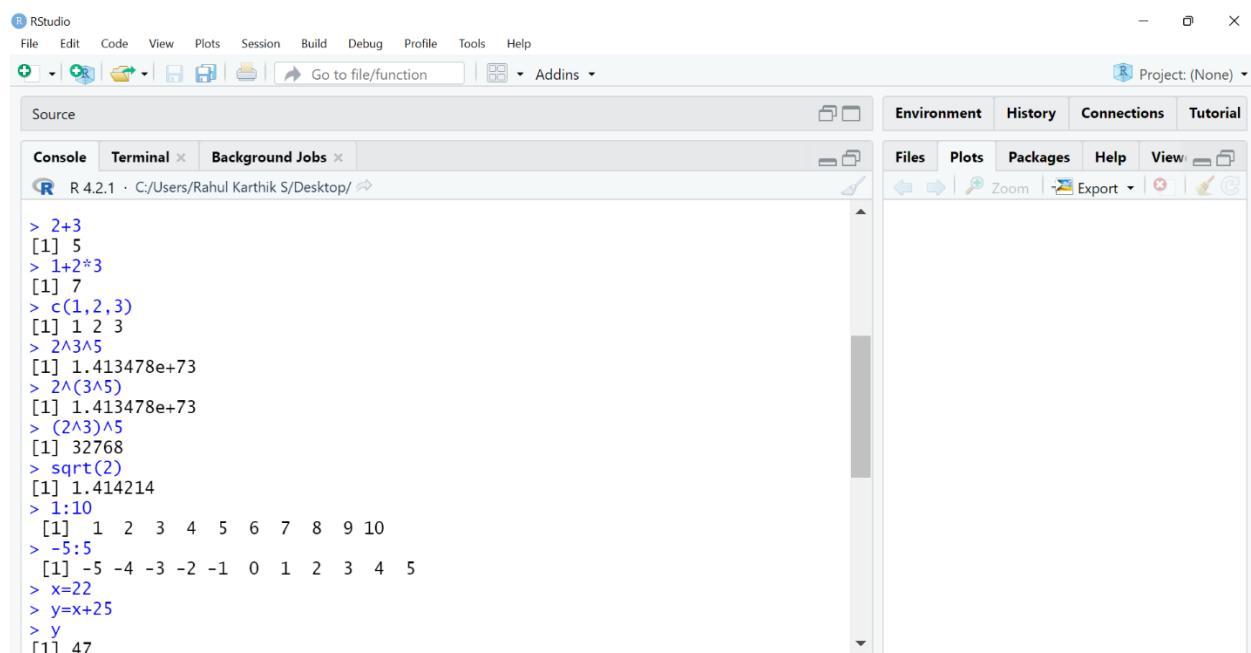
```
6 (2^3)^5
7 sqrt(2)
8 1:10
9 -5:5
10 x=22
11 y=x+25
12 y
13 a=c(1,2,3,4)
14 b=c(1,3,5,7)
15 a+b
16 a-b
17 5*a
18 2*a+1
19 u=c(10,20,30)
20 v=c(1,2,3,4,5,6,7,8,9)
21 u+v
22 n=c(2,3,5)
23 s=c("aa","bb","cc")
24 b=c(TRUE,FALSE,TRUE)
25 df=data.frame(n,s,b)
26 df
```

Output:

```
> 2+3
[1] 5
> 1+2*3
[1] 7
> c(1,2,3)
```

```
[1] 1 2 3
> 2^3^5
[1] 1.413478e+73
> 2^(3^5)
[1] 1.413478e+73
> (2^3)^5
[1] 32768
> sqrt(2)
[1] 1.414214
> 1:10
[1] 1 2 3 4 5 6 7 8 9 10
> -5:5
[1] -5 -4 -3 -2 -1 0 1 2 3 4 5
> x=22
> y=x+25
> y
[1] 47
> a=c(1,2,3,4)
> b=c(1,3,5,7)
>a+b
[1] 2 5 8 11
> a-b
[1] 0 -1 -2 -3
> 5*a
[1] 5 10 15 20
> 2*a+1
[1] 3 5 7 9
> u=c(10,20,30)
> v=c(1,2,3,4,5,6,7,8,9)
>u+v
```

```
[1] 11 22 33 14 25 36 17 28 39
> n=c(2,3,5)
> s=c("aa","bb","cc")
> b=c(TRUE, FALSE, TRUE)
> df=data.frame(n,s,b)
> df
   n   s     b
1 2 aa  TRUE
2 3 bb FALSE
3 5 cc  TRUE
```



The image displays two side-by-side screenshots of the RStudio interface, both showing the same R session history and environment.

Session 1 (Left):

```
> y  
[1] 47  
> a=c(1,2,3,4)  
> b=c(1,3,5,7)  
> a+b  
[1] 2 5 8 11  
> a-b  
[1] 0 -1 -2 -3  
> 5*a  
[1] 5 10 15 20  
> 2*a+1  
[1] 3 5 7 9  
> u=c(10,20,30)  
> v=c(1,2,3,4,5,6,7,8,9)  
> u+v  
[1] 11 22 33 14 25 36 17 28 39  
> n=c(2,3,5)  
> s=c("aa","bb","cc")  
> b=c(TRUE,FALSE,TRUE)  
> df=data.frame(n,s,b)  
> df  
   n   s     b  
1 2  aa  TRUE  
2 3  bb FALSE  
3 5  cc  TRUE  
> |
```

Session 2 (Right):

```
> a+b  
[1] 2 5 8 11  
> a-b  
[1] 0 -1 -2 -3  
> 5*a  
[1] 5 10 15 20  
> 2*a+1  
[1] 3 5 7 9  
> u=c(10,20,30)  
> v=c(1,2,3,4,5,6,7,8,9)  
> u+v  
[1] 11 22 33 14 25 36 17 28 39  
> n=c(2,3,5)  
> s=c("aa","bb","cc")  
> b=c(TRUE,FALSE,TRUE)  
> df=data.frame(n,s,b)  
> df  
   n   s     b  
1 2  aa  TRUE  
2 3  bb FALSE  
3 5  cc  TRUE  
> |
```

Week – 2

Aim: Computation of tables and graphs – summary statistics for employee data.

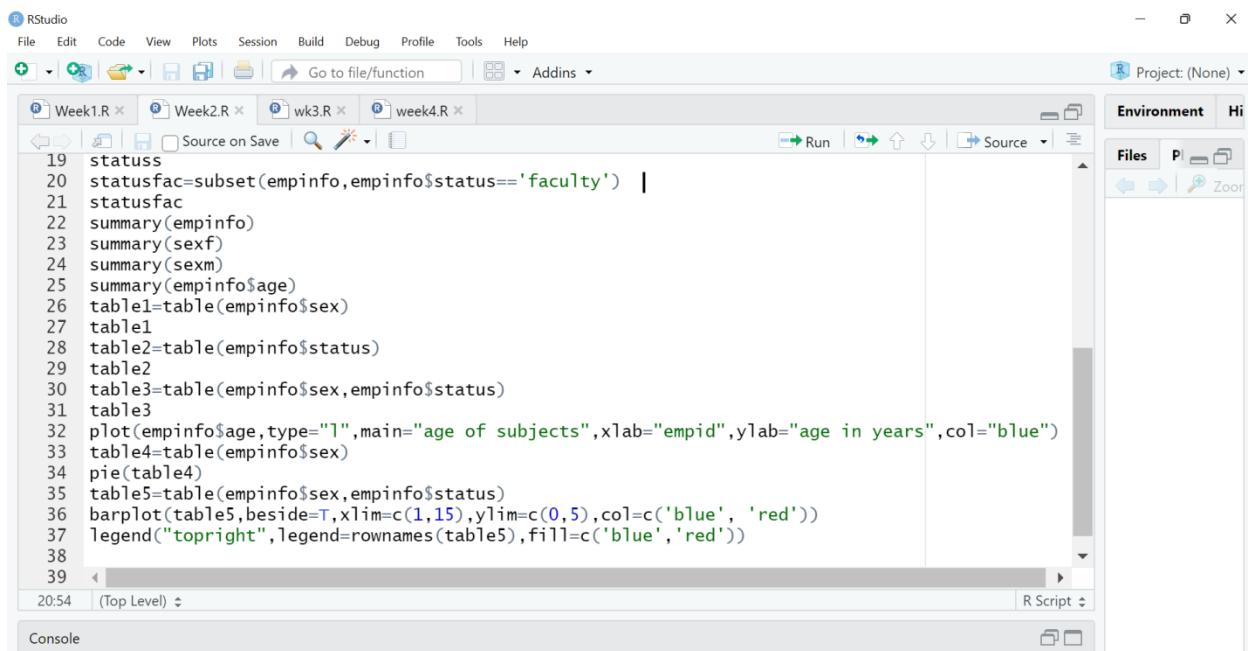
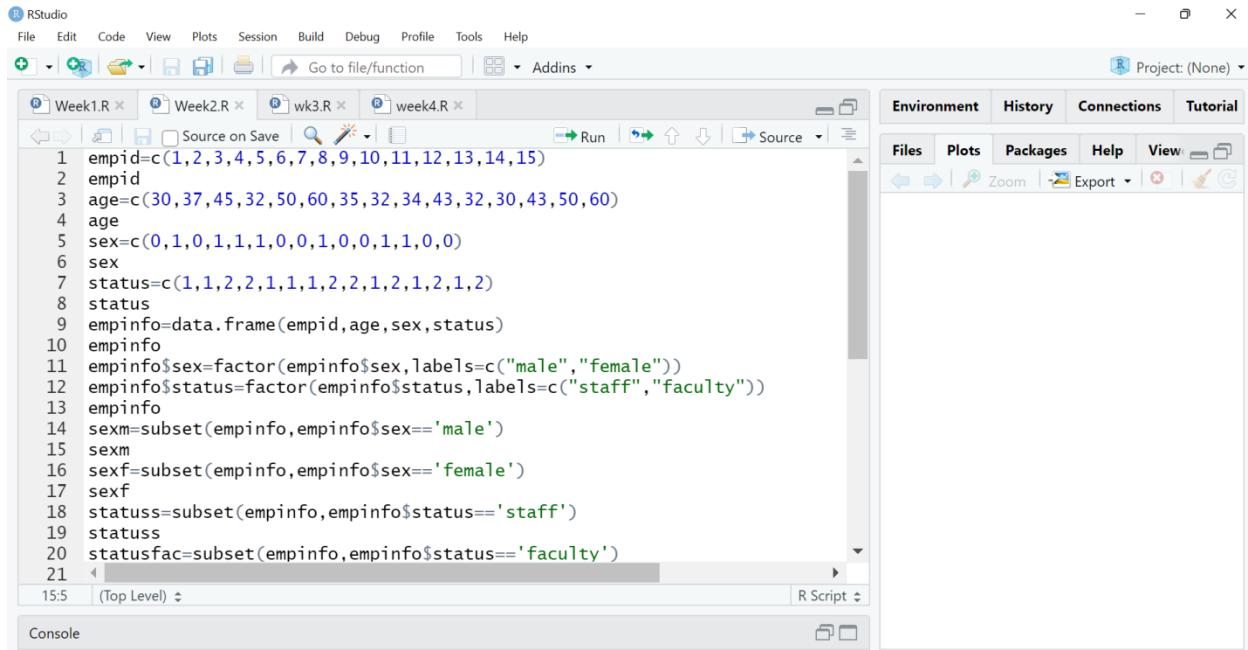
Program Code:

```
empid=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
empid
age=c(30,37,45,32,50,60,35,32,34,43,32,30,43,50,60)
age
sex=c(0,1,0,1,1,1,0,0,1,0,0,1,1,0,0)
sex
status=c(1,1,2,2,1,1,1,2,2,1,2,1,2,1,2)
status
empinfo=data.frame(empid,age,sex,status)
empinfo
empinfo$sex=factor(empinfo$sex,labels=c("male","female"))
empinfo$status=factor(empinfo$status,labels=c("staff","faculty"))
empinfo
sexm=subset(empinfo,empinfo$sex=='male')
sexm
sexf=subset(empinfo,empinfo$sex=='female')
sexf
statuss=subset(empinfo,empinfo$status=='staff')
statuss
statusfac=subset(empinfo,empinfo$status=='faculty')
statusfac
summary(empinfo)
summary(sexf)
summary(sexm)
summary(empinfo$age)
table1=table(empinfo$sex)
table1
table2=table(empinfo$status)
table2
table3=table(empinfo$sex,empinfo$status)
table3
```

```

plot(empinfo$age,type="l",main="age of
subjects",xlab="empid",ylab="age in years",col="blue")
table4=table(empinfo$sex)
pie(table4)
table5=table(empinfo$sex,empinfo$status)
barplot(table5,beside=T,xlim=c(1,15),ylim=c(0,5),col=c('blue', 'red'))
legend("topright",legend=rownames(table5),fill=c('blue','red'))

```



Output:

```
> empid=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
> empid
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15
>
age=c(30,37,45,32,50,60,35,32,34,43,32,30,43,5
0,60)
> age
[1] 30 37 45 32 50 60 35 32 34 43 32 30 43 50
60
> sex=c(0,1,0,1,1,1,0,0,1,0,0,1,1,0,0)
> sex
[1] 0 1 0 1 1 1 0 0 1 0 0 1 1 0 0
> status=c(1,1,2,2,1,1,1,2,2,1,2,1,2,1,2)
> status
[1] 1 1 2 2 1 1 1 2 2 1 2 1 2 1 2
> empinfo=data.frame(empid,age,sex,status)
> empinfo
  empid age sex status
1 1 30 0 1
2 2 37 1 1
3 3 45 0 2
4 4 32 1 2
5 5 50 1 1
6 6 60 1 1
7 7 35 0 1
8 8 32 0 2
9 9 34 1 2
10 10 43 0 1
11 11 32 0 2
12 12 30 1 1
13 13 43 1 2
14 14 50 0 1
15 15 60 0 2
>
```

```
empinfo$sex=factor(empinfo$sex,labels=c("male"
,"female"))
>
empinfo$status=factor(empinfo$status,labels=c(
"staff","faculty"))
> empinfo
  empid age sex status
1 1 30 male staff
2 2 37 female staff
3 3 45 male faculty
4 4 32 female faculty
5 5 50 female staff
6 6 60 female staff
7 7 35 male staff
8 8 32 male faculty
9 9 34 female faculty
10 10 43 male staff
11 11 32 male faculty
12 12 30 female staff
13 13 43 female faculty
14 14 50 male staff
15 15 60 male faculty
> sexm=subset(empinfo,empinfo$sex=='male')

> sexm
  empid age sex status
1 1 30 male staff
3 3 45 male faculty
7 7 35 male staff
8 8 32 male faculty
10 10 43 male staff
11 11 32 male faculty
14 14 50 male staff
15 15 60 male faculty
> sexf=subset(empinfo,empinfo$sex=='female')
```

```
> sexf
  empid age sex status
  2 2 37 female staff
  4 4 32 female faculty
  5 5 50 female staff
  6 6 60 female staff
  9 9 34 female faculty
 12 12 30 female staff
 13 13 43 female faculty
>
statusss=subset(empinfo,empinfo$status=='staff'
)
> statusss
  empid age sex status
  1 1 30 male staff
  2 2 37 female staff
  5 5 50 female staff
  6 6 60 female staff
  7 7 35 male staff
 10 10 43 male staff
 12 12 30 female staff
 14 14 50 male staff
>
statusfac=subset(empinfo,empinfo$status=='faculty')
> statusfac
  empid age sex status
  3 3 45 male faculty
  4 4 32 female faculty
  8 8 32 male faculty
  9 9 34 female faculty
 11 11 32 male faculty
 13 13 43 female faculty
 15 15 60 male faculty
```

```
> summary(empinfo)
  empid age sex
  status
  Min. : 1.0 Min. :30.00 male :8
  staff :8
  1st Qu.: 4.5 1st Qu.:32.00 female:7
  faculty:7
  Median : 8.0 Median :37.00

  Mean : 8.0 Mean :40.87

  3rd Qu.:11.5 3rd Qu.:47.50

  Max. :15.0 Max. :60.00

> summary(sexf)
  empid age sex
  status
  Min. : 2.000 Min. :30.00 male :0
  staff :4
  1st Qu.: 4.500 1st Qu.:33.00 female:7
  faculty:3
  Median : 6.000 Median :37.00

  Mean : 7.286 Mean :40.86

  3rd Qu.:10.500 3rd Qu.:46.50

  Max. :13.000 Max. :60.00

> summary(sexm)
  empid age sex
  status
  Min. : 1.000 Min. :30.00 male :8
  staff :4
```

```
1st Qu.: 6.000 1st Qu.:32.00 female:0
faculty:4
Median : 9.000 Median :39.00

Mean : 8.625 Mean :40.88

3rd Qu.:11.750 3rd Qu.:46.25

Max. :15.000 Max. :60.00

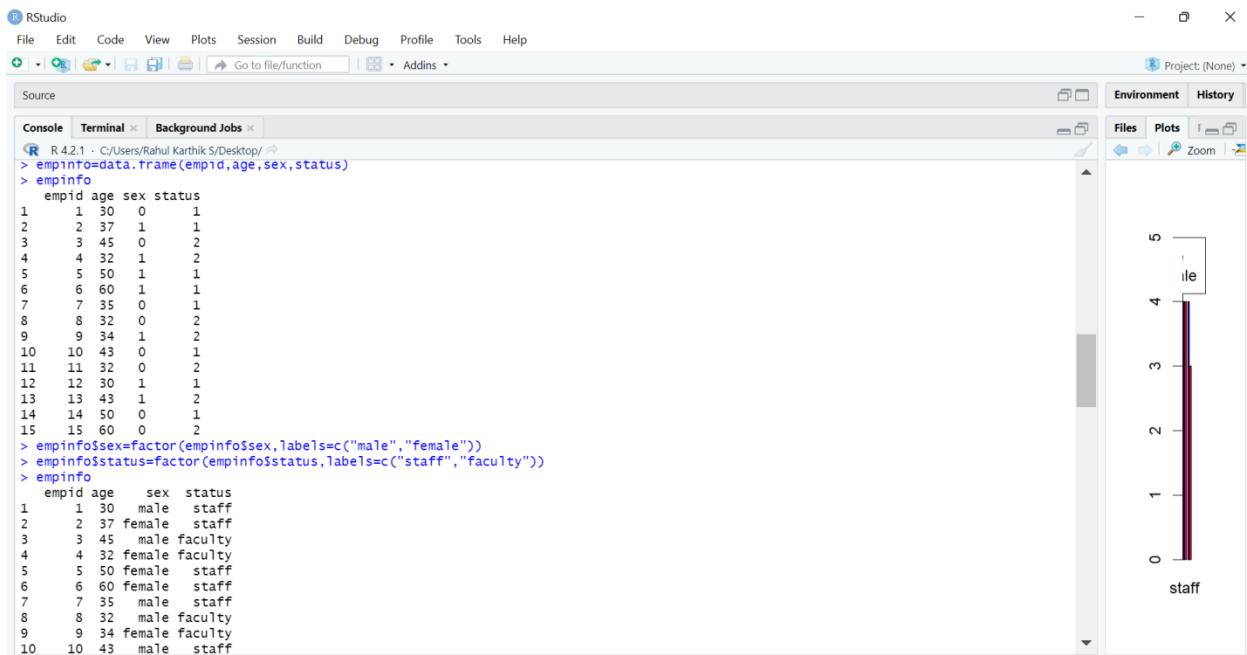
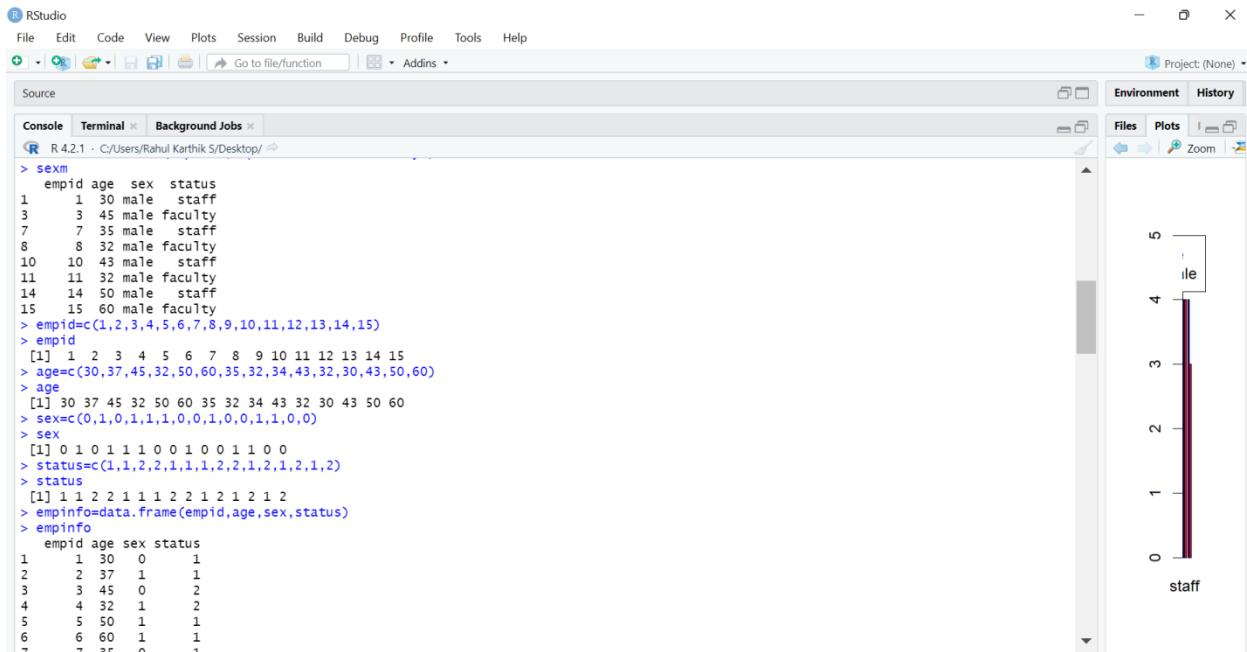
> summary(empinfo$age)
Min. 1st Qu. Median Mean 3rd Qu.
Max.
30.00 32.00 37.00 40.87 47.50
60.00
> table1=table(empinfo$sex)
> table1
male female
8 7
> table2=table(empinfo$status)
> table2
staff faculty
8 7
> table3=table(empinfo$sex,empinfo$status)
> table3

staff faculty
male 4 4
female 4 3
> plot(empinfo$age,type="l",main="age of
subjects",xlab="empid",ylab="age in
years",col="blue")
> table4=table(empinfo$sex)
> pie(table4)
> table5=table(empinfo$sex,empinfo$status)
```

```

>
barplot(table5,beside=T,xlim=c(1,15),ylim=c(0,
5),col=c('blue', 'red'))
>
legend("topright",legend=rownames(table5),fill
=c('blue','red'))
>

```



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Project: (None)

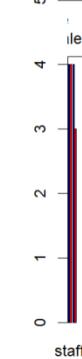
Source

Console Terminal Background Jobs

R 4.2.1 - C:/Users/Rahul Karthik S/Desktop/

```
empid age sex status
1 1 30 male staff
2 2 37 female staff
3 3 45 male faculty
4 4 32 female faculty
5 5 50 female staff
6 6 60 female staff
7 7 35 male staff
8 8 32 male faculty
9 9 34 female faculty
10 10 43 male staff
11 11 32 male faculty
12 12 30 female staff
13 13 43 female faculty
14 14 50 male staff
15 15 60 male faculty
> sexm<-subset(empinfo,empinfo$sex=='male')
> sexm
  empid age sex status
1 1 30 male staff
3 3 45 male faculty
7 7 35 male staff
8 8 32 male faculty
10 10 43 male staff
11 11 32 male faculty
14 14 50 male staff
15 15 60 male faculty
> sexf<-subset(empinfo,empinfo$sex=='female')
> sexf
  empid age sex status
2 2 37 female staff
4 4 32 female faculty
5 5 50 female staff
6 6 60 female staff
7 7 35 male staff
10 10 43 male staff
12 12 30 female staff
13 13 43 female faculty
> statusss<-subset(empinfo,empinfo$status=='staff')
> statusss
  empid age sex status
1 1 30 male staff
2 2 37 female staff
5 5 50 female staff
6 6 60 female staff
7 7 35 male staff
10 10 43 male staff
12 12 30 female staff
14 14 50 male staff
> statussfac<-subset(empinfo,empinfo$status=='faculty')
> statussfac
  empid age sex status
3 3 45 male faculty
4 4 32 female faculty
8 8 32 male faculty
9 9 34 female faculty
11 11 32 male faculty
13 13 43 female faculty
15 15 60 male faculty
> summary(empinfo)
```

Files Plots Zoom



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Project: (None)

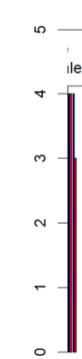
Source

Console Terminal Background Jobs

R 4.2.1 - C:/Users/Rahul Karthik S/Desktop/

```
> sexf<-subset(empinfo,empinfo$sex=='female')
> sexf
  empid age sex status
2 2 37 female staff
4 4 32 female faculty
5 5 50 female staff
6 6 60 female staff
7 7 35 male staff
10 10 43 male staff
12 12 30 female staff
13 13 43 female faculty
> statusss<-subset(empinfo,empinfo$status=='staff')
> statusss
  empid age sex status
1 1 30 male staff
2 2 37 female staff
5 5 50 female staff
6 6 60 female staff
7 7 35 male staff
10 10 43 male staff
12 12 30 female staff
14 14 50 male staff
> statussfac<-subset(empinfo,empinfo$status=='faculty')
> statussfac
  empid age sex status
3 3 45 male faculty
4 4 32 female faculty
8 8 32 male faculty
9 9 34 female faculty
11 11 32 male faculty
13 13 43 female faculty
15 15 60 male faculty
> summary(empinfo)
```

Files Plots Zoom



RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Project: (None)

Source

Console Terminal Background Jobs

```
R 4.2.1 - C:/Users/Rahul Karthik S/Desktop/ 
> summary(empinfo)
  empid      age       sex     status
Min. : 1.00  Min. :30.00  male :8   staff :8
1st Qu.: 4.50  1st Qu.:32.00 female:7  faculty:7
Median : 8.00  Median :37.00
Mean  : 8.00  Mean  :40.87
3rd Qu.:11.50 3rd Qu.:47.50
Max.  :15.00  Max.  :60.00
> summary(sexf)
  empid      age       sex     status
Min. : 2.000  Min. :30.00  male :0   staff :4
1st Qu.: 4.500  1st Qu.:33.00 female:7  faculty:3
Median : 6.000  Median :37.00
Mean  : 7.286  Mean  :40.86
3rd Qu.:10.500 3rd Qu.:46.50
Max.  :13.000  Max.  :60.00
> summary(sexm)
  empid      age       sex     status
Min. : 1.000  Min. :30.00  male :8   staff :4
1st Qu.: 6.000  1st Qu.:32.00 female:0  faculty:4
Median : 9.000  Median :39.00
Mean  : 8.625  Mean  :40.88
3rd Qu.:11.750 3rd Qu.:46.25
Max.  :15.000  Max.  :60.00
> summary(empinfo$age)
  Min. 1st Qu. Median  Mean 3rd Qu.  Max.
30.00 32.00 37.00 40.87 47.50 60.00
> table1=table(empinfo$sex)
> table1

  male female
    8      7
```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Project: (None)

Source

Console Terminal Background Jobs

```
R 4.2.1 - C:/Users/Rahul Karthik S/Desktop/ 
1st Qu.: 6.000  1st Qu.:32.00  female:0  faculty:4
Median : 9.000  Median :39.00
Mean  : 8.625  Mean  :40.88
3rd Qu.:11.750 3rd Qu.:46.25
Max.  :15.000  Max.  :60.00
> summary(empinfo$age)
  Min. 1st Qu. Median  Mean 3rd Qu.  Max.
30.00 32.00 37.00 40.87 47.50 60.00
> table1=table(empinfo$sex)
> table1

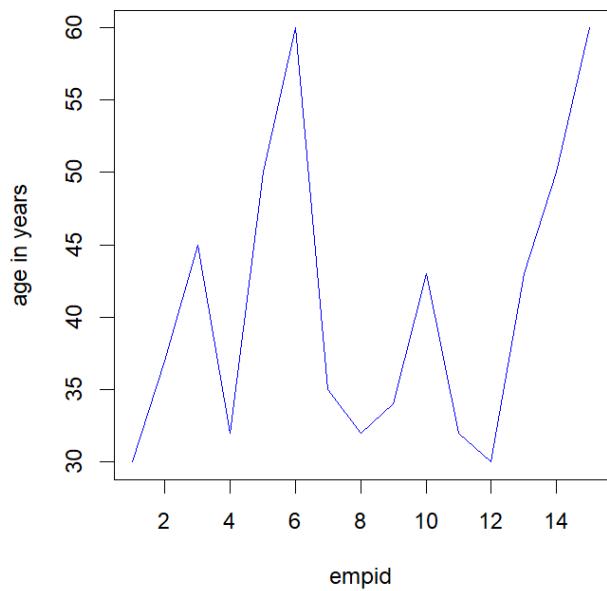
  male female
    8      7
> table2=table(empinfo$status)
> table2

  staff faculty
    8      7
> table3=table(empinfo$sex,empinfo$status)
> table3

  staff faculty
male     4      4
female   4      3
> plot(empinfo$age,type="l",main="age of subjects",xlab="empid",ylab="age in years",col="blue")
> table4=table(empinfo$sex)
> pie(table4)
> table5=table(empinfo$sex,empinfo$status)
> barplot(table5,beside=T,xlim=c(1,15),ylim=c(0,5),col=c('blue', 'red'))
> legend("topright",legend=rownames(table5),fill=c('blue','red'))
```

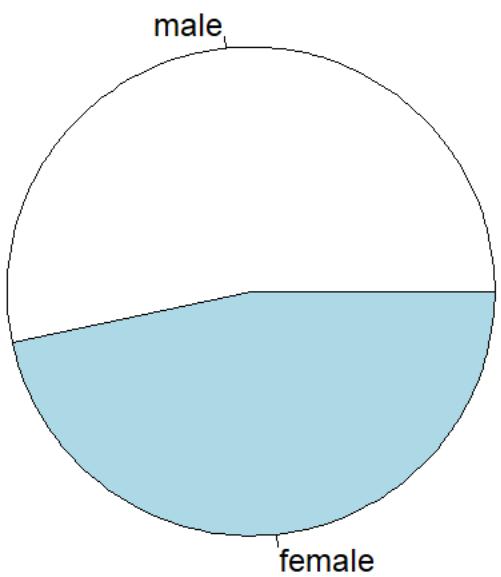
Graph:

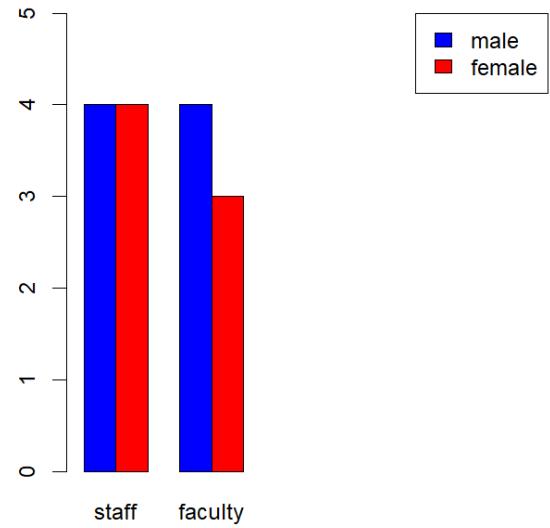
age of subjects



male

female





Week – 3

Aim: To calculate the correlation coefficient using R- Programming Language.

Program Code:

```
x=c(65, 66, 67, 67, 68, 69, 70, 72)
y=c(67, 68, 65, 68, 72, 72, 69, 71)
var(x)
var(y)
var(x,y)
r=var(x,y)/sqrt(var(x)*var(y) )
r
r=cor(x,y)
r
cor.test(x,y,method="pearson")
cor(x,y)
model=lm(x~y)
summary.lm(model)
selection=c(44, 49, 52, 54, 47, 76, 65, 60, 63, 58, 50, 67)
proficiency=c(48, 55, 45, 60, 43, 80, 58, 50, 77, 46, 47, 65)
cor.test(selection,proficiency,method="spearman")
```

The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with various icons for file operations like Open, Save, and Run. The main area displays an R script with the following code:

```
1 x=c(65,66,67,67,68,69,70,72)
2 y=c(67,68,65,68,72,72,69,71)
3 var(x)
4 var(y)
5 var(x,y)
6 r=var(x,y)/sqrt(var(x)*var(y))
7 r
8 r=cor(x,y)
9 r
10 cor.test(x,y,method="pearson")
11 cor(x,y)
12 model=lm(x~y)
13 summary.lm(model)
14 selection=c(44,49,52,54,47,76,65,60,63,58,50,67)
15 proficiency=c(48,55,45,60,43,80,58,50,77,46,47,65)
16 cor.test(selection,proficiency,method="spearman")
```

The status bar at the bottom shows "16:50 (Top Level) R Script".

Output:

```
> x=c(65,66,67,67,68,69,70,72)

> y=c(67,68,65,68,72,72,69,71)

> var(x)

[1] 5.142857

> var(y)

[1] 6.285714

> var(x,y)

[1] 3.428571

> r=var(x,y)/sqrt(var(x)*var(y))

> r

[1] 0.6030227

> r=cor(x,y)

> r

[1] 0.6030227

> cor.test(x,y,method="pearson")
```

Pearson's product-moment correlation

data: x and y
t = 1.8516, df = 6, p-value = 0.1135
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.1767627 0.9177238
sample estimates:

cor

0.6030227

```
> cor(x,y)  
[1] 0.6030227  
> model=lm(x~y)  
> summary.lm(model)
```

Call:

lm(formula = x ~ y)

Residuals:

Min	1Q	Median	3Q	Max
-1.9091	-1.5000	-0.5454	1.3864	2.9091

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	30.3636	20.3377	1.493	0.186
y	0.5455	0.2946	1.852	0.114

```
Residual standard error: 1.954 on 6 degrees of freedom  
Multiple R-squared:  0.3636,  Adjusted R-squared:  0.2576  
F-statistic: 3.429 on 1 and 6 DF,  p-value: 0.1135
```

```
> selection=c(44,49,52,54,47,76,65,60,63,58,50,67)  
> proficiency=c(48,55,45,60,43,80,58,50,77,46,47,65)  
> cor.test(selection,proficiency,method="spearman")
```

Spearman's rank correlation rho

```
data: selection and proficiency  
S = 80, p-value = 0.01102  
alternative hypothesis: true rho is not equal to 0  
sample estimates:  
rho  
0.7202797
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins Project: (None)

Source

Console Terminal Background Jobs

```
R 4.2.1 : C:/Users/Rahul Karthik S/Desktop/r
> x=c(65,66,67,67,68,69,70,72)
> y=c(67,68,65,68,72,72,69,71)
> var(x)
[1] 5.142857
> var(y)
[1] 6.285714
> var(x,y)
[1] 3.428571
> r=var(x,y)/sqrt(var(x)*var(y))
> r
[1] 0.6030227
> r<-cor(x,y)
> r
[1] 0.6030227
> cor.test(x,y,method="pearson")

Pearson's product-moment correlation

data: x and y
t = 1.8516, df = 6, p-value = 0.1135
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.1767627 0.9177238
sample estimates:
cor
0.6030227

> cor(x,y)
[1] 0.6030227
> model=lm(x~y)
> summary.lm(model)
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins Project: (None)

Source

Console Terminal Background Jobs

```
R 4.2.1 : C:/Users/Rahul Karthik S/Desktop/r
> model=lm(x~y)
> summary.lm(model)

Call:
lm(formula = x ~ y)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.9091 -1.5000 -0.5454  1.3864  2.9091 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 30.3636   20.3377   1.493   0.186    
y           0.5455    0.2946   1.852   0.114    
                                                        
Residual standard error: 1.954 on 6 degrees of freedom
Multiple R-squared:  0.3636, Adjusted R-squared:  0.2576 
F-statistic: 3.429 on 1 and 6 DF,  p-value: 0.1135

> selection=c(44,49,52,54,47,76,65,60,63,58,50,67)
> proficiency=c(48,55,45,60,43,80,58,50,77,46,47,65)
> cor.test(selection,proficiency,method="spearman")

Spearman's rank correlation rho

data: selection and proficiency
S = 80, p-value = 0.01102
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.7202797
```

Week – 4

Definition: Multiple regression generally explains the relationship between multiple independent variables and one dependent variable.

The multiple regression equation takes the following form:

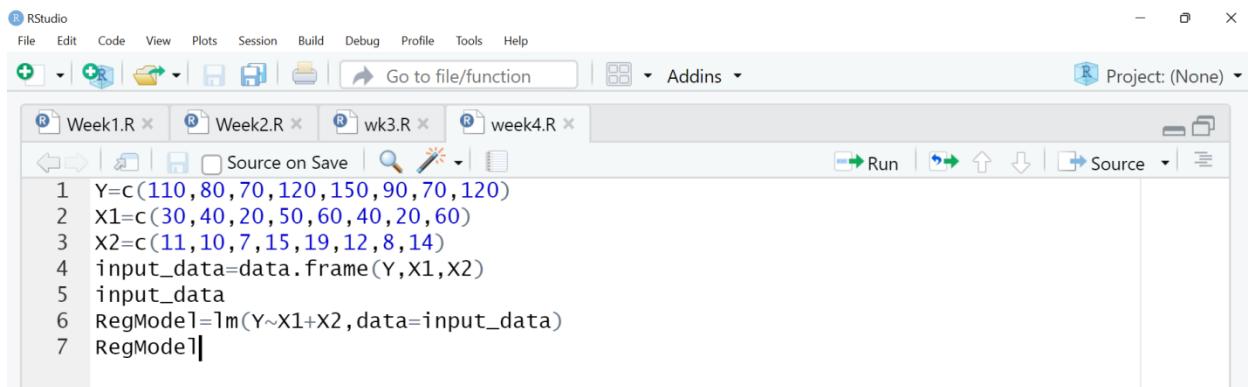
$$y = b_1x_1 + b_2x_2 + \dots + b_nx_n + c$$

Problem 1: The sale of a product in lakhs of rupees (Y) is expected to be influenced by two variables namely the advertising expenditure (X_1) and the number of sales persons (X_2) in a region. Sample data on 8 regions of a state has given the following results.

Area	Y	X1	X2
1	110	30	11
2	80	40	10
3	70	20	7
4	120	50	15
5	150	60	19
6	90	40	12
7	70	20	8
8	120	60	14

Program Code:

```
Y=c(110,80,70,120,150,90,70,120)
X1=c(30,40,20,50,60,40,20,60)
X2=c(11,10,7,15,19,12,8,14)
input_data=data.frame(Y,X1,X2)
input_data
RegModel=lm(Y~X1+X2,data=input_data)
RegModel
```



The screenshot shows the RStudio interface with the following details:

- Toolbar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- File Explorer:** Shows files Week1.R, Week2.R, wk3.R, and week4.R.
- Code Editor:** Displays the R code for the multiple regression model.
- Run Buttons:** Run, Source, etc.
- Project:** Project (None).

```
1 Y=c(110,80,70,120,150,90,70,120)
2 X1=c(30,40,20,50,60,40,20,60)
3 X2=c(11,10,7,15,19,12,8,14)
4 input_data=data.frame(Y,X1,X2)
5 input_data
6 RegModel=lm(Y~X1+X2,data=input_data)
7 RegModel
```

Output:

```
> Y=c(110,80,70,120,150,90,70,120)
> X1=c(30,40,20,50,60,40,20,60)
> X2=c(11,10,7,15,19,12,8,14)
> input_data=data.frame(Y,X1,X2)
> input_data
   Y  X1  X2
1 110 30 11
2   80 40 10
3   70 20  7
4 120 50 15
5 150 60 19
6   90 40 12
7   70 20  8
8 120 60 14
> RegModel=lm(Y~X1+X2,data=input_data)
> RegModel
```

Call:

```
lm(formula = Y ~ X1 + X2, data = input_data)
```

Coefficients:

(Intercept)	X1	X2
16.8314	-0.2442	7.8488

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Project: (None)

Source

Console Terminal × Background Jobs ×

R 4.2.1 · C:/Users/Rahul Karthik S/Desktop/ ↗

```
> Y=c(110,80,70,120,150,90,70,120)
> X1=c(30,40,20,50,60,40,20,60)
> X2=c(11,10,7,15,19,12,8,14)
> input_data=data.frame(Y,X1,X2)
> input_data
   Y X1 X2
1 110 30 11
2  80 40 10
3  70 20  7
4 120 50 15
5 150 60 19
6  90 40 12
7  70 20  8
8 120 60 14
> RegModel=lm(Y~X1+X2,data=input_data)
> RegModel
```

Call:

```
lm(formula = Y ~ X1 + X2, data = input_data)
```

Coefficients:

(Intercept)	X1	X2
16.8314	-0.2442	7.8488

> |

Week – 5

Aim: To conduct random experiments with the probability concepts using R- Programming Language.

Program Code:

```
sample(1:50, 5)

sample(1:6,10,replace = TRUE)

sample(c('H','T'),10,replace = TRUE)

sample(c('success', 'failure'),10, replace = TRUE)

choose(5,2)

n = 10; k = 5;

p = factorial(10)/factorial(5)

p

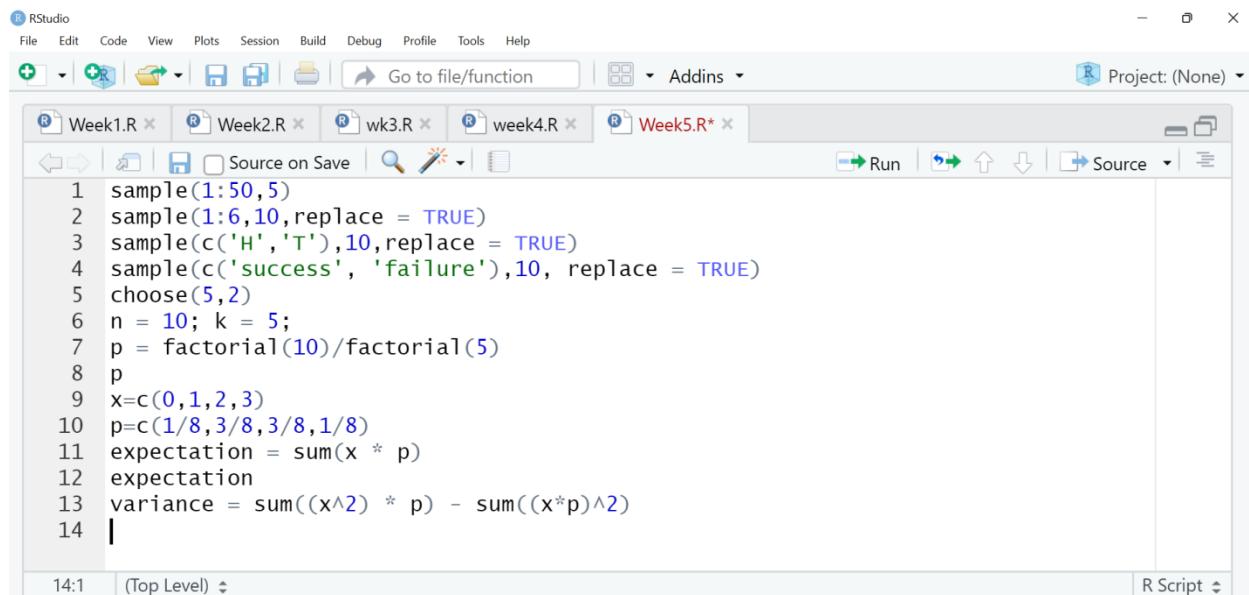
x=c(0,1,2,3)

p=c(1/8,3/8,3/8,1/8)

expectation = sum(x * p)

expectation

variance = sum((x^2) * p) - sum((x*p)^2)
```



The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar below the menu has icons for file operations like Open, Save, and Run, along with a Go to file/function search bar and an Addins dropdown. The Project pane on the right shows '(None)'. The code editor window displays the R script with numbered lines from 1 to 14. The status bar at the bottom indicates line 14:1 and Top Level, with a dropdown for R Script.

```
1 sample(1:50, 5)
2 sample(1:6,10,replace = TRUE)
3 sample(c('H','T'),10,replace = TRUE)
4 sample(c('success', 'failure'),10, replace = TRUE)
5 choose(5,2)
6 n = 10; k = 5;
7 p = factorial(10)/factorial(5)
8 p
9 x=c(0,1,2,3)
10 p=c(1/8,3/8,3/8,1/8)
11 expectation = sum(x * p)
12 expectation
13 variance = sum((x^2) * p) - sum((x*p)^2)
14 |
```

Output:

```
> sample(1:50,5)
[1] 13 39 42 26 44

> sample(1:6,10,replace = TRUE)
[1] 1 2 3 3 5 3 5 3 1 1

> sample(c('H','T'),10,replace = TRUE)
[1] "H" "T" "T" "H" "T" "H" "H" "T" "T" "T"

> sample(c('success', 'failure'),10, replace = TRUE)
[1] "failure" "success" "failure" "success" "success" "failure"
"failure" "success" "success" "success"

> choose(5,2)
[1] 10

> n = 10; k = 5;

> p = factorial(10)/factorial(5)

> p
[1] 30240

> x=c(0,1,2,3)

> p=c(1/8,3/8,3/8,1/8)

> expectation = sum(x * p)

> expectation
[1] 1.5

> variance = sum((x^2) * p) - sum((x*p)^2)

>
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Source

Console Terminal Background Jobs

R 4.2.1 · C:/Users/Rahul Karthik S/Desktop/ ↗

```
> sample(1:50,5)
[1] 13 39 42 26 44
> sample(1:6,10,replace = TRUE)
[1] 1 2 3 3 5 3 5 3 1 1
> sample(c('H','T'),10,replace = TRUE)
[1] "H" "T" "T" "H" "T" "H" "H" "T" "T" "T"
> sample(c('success', 'failure'),10, replace = TRUE)
[1] "failure" "success" "failure" "success" "success" "failure" "failure" "success" "success" "success"
> choose(5,2)
[1] 10
> n = 10; k = 5;
> p = factorial(10)/factorial(5)
> p
[1] 30240
> x=c(0,1,2,3)
> p=c(1/8,3/8,3/8,1/8)
> expectation = sum(x * p)
> expectation
[1] 1.5
> variance = sum((x^2) * p) - sum((x*p)^2)
> |
```

Week - 6

Name: Rahul Karthik S

Register Number: 21BEC1851

Binomial Distribution: A rv X is said to follow binomial distribution if it assumed only non-negative values and its pmf is given by:

$$P(X=x) = p(x) = \begin{cases} \binom{n}{x} p^x q^{n-x}; & x = 0, 1, 2, \dots n; q = 1 - p \\ 0, & \text{otherwise} \end{cases}$$

`dbinom(x,size=n,prob=p)`

Aim: Computation and Plotting Binomial Distributions.

PROBLEM 1: Find the probability of getting two "2" among ten dice.

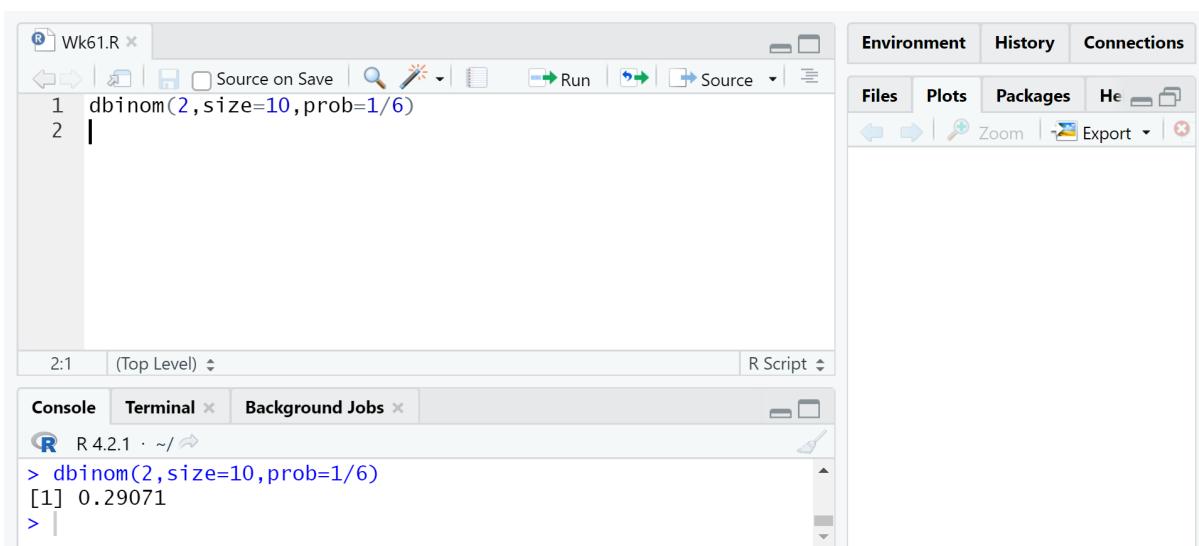
PROGRAM CODE:

```
dbinom(2,size=10,prob=1/6)
```

OUTPUT:

```
> dbinom(2,size=10,prob=1/6)
```

```
[1] 0.29071
```



PROBLEM 2: Find BIN(n=10,P=1/6) and visualise binomial plots.

PROGRAM CODE:

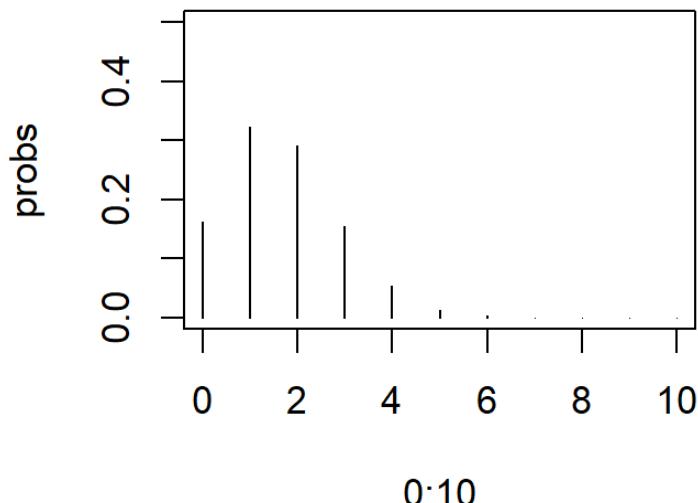
```
n=10;p=1/6
probs=dbinom(x=c(0:n),size=n,prob=p)
probs=round(probs,4);
x=0:n;
```

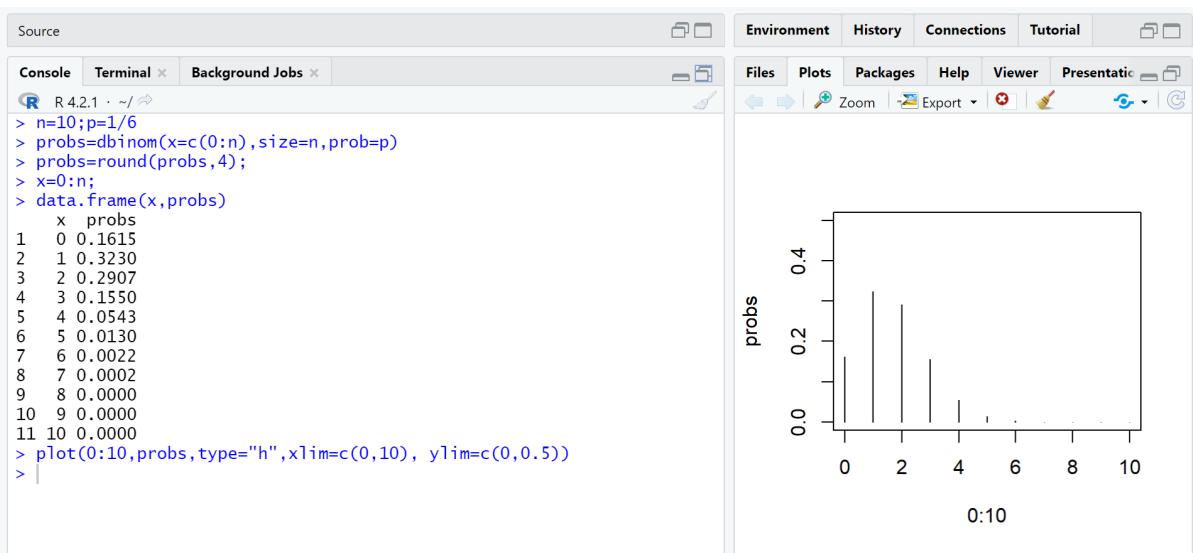
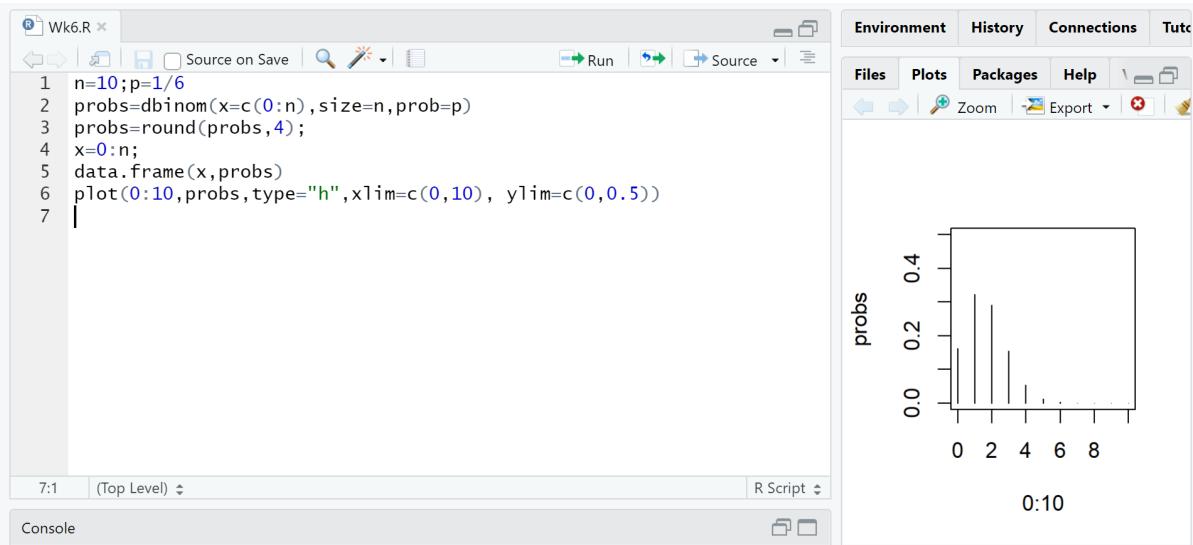
```
data.frame(x,probs)
plot(0:10,probs,type="h",xlim=c(0,10), ylim=c(0,0.5))
```

OUTPUT:

```
> n=10;p=1/6
> probs=dbinom(x=c(0:n),size=n,prob=p)
> probs=round(probs,4);
> x=0:n;
> data.frame(x,probs)
  x   probs
1 0 0.1615
2 1 0.3230
3 2 0.2907
4 3 0.1550
5 4 0.0543
6 5 0.0130
7 6 0.0022
8 7 0.0002
9 8 0.0000
10 9 0.0000
11 10 0.0000
> plot(0:10,probs,type="h",xlim=c(0,10), ylim=c(0,0.5))
>
```

GRAPH:





PROBLEM 3: For a Binomial(7,1/4) random variable named X,

- Compute the probability of two success**
- Compute the Probabilities for whole space**
- Display those probabilities in a table**
- Show the shape of this binomial Distribution**

PROGRAM CODE:

(i)
`dbinom(2,7,1/4)`

(ii)
`p=dbinom(0:7,7,1/4)`
`round(p,4)`

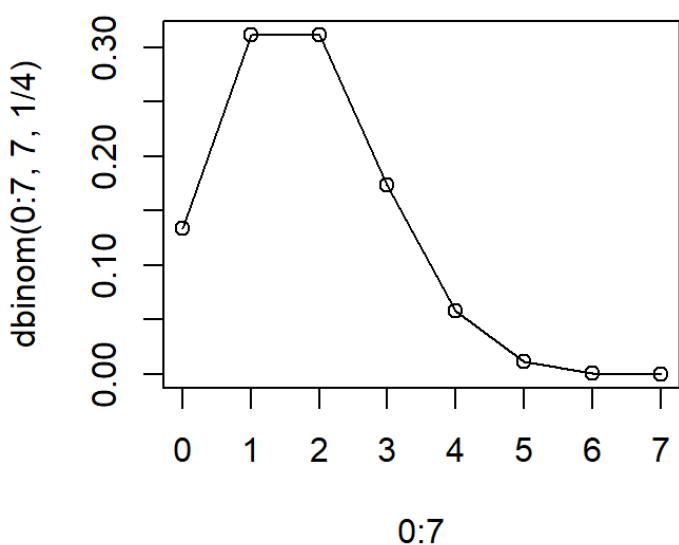
(iii)
`p=data.frame(0:7,dbinom(0:7,7,1/4))`
`round(p,4)`

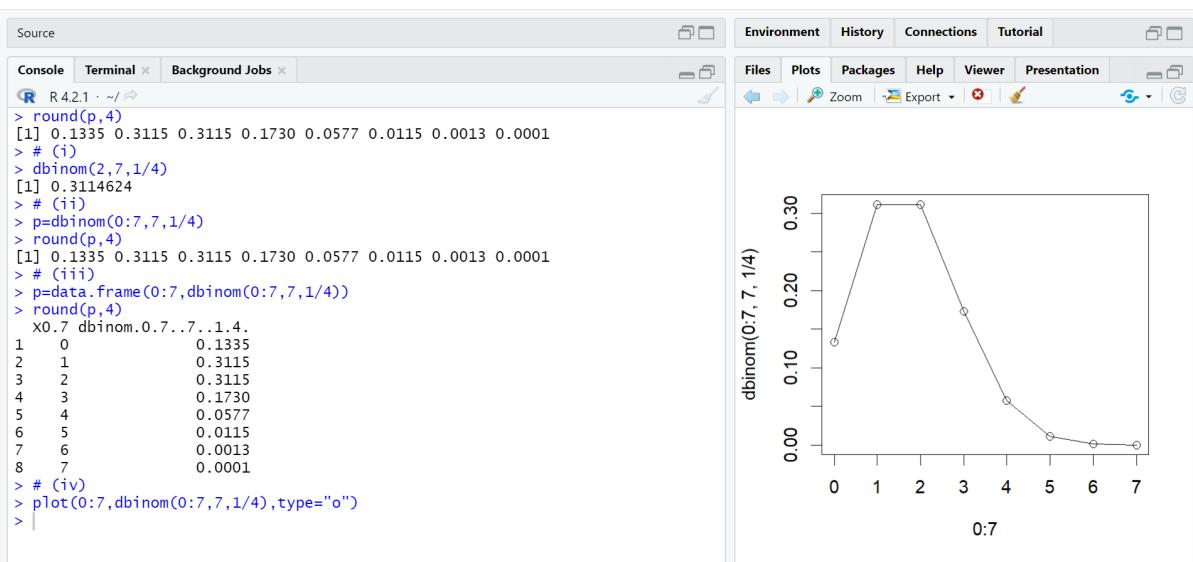
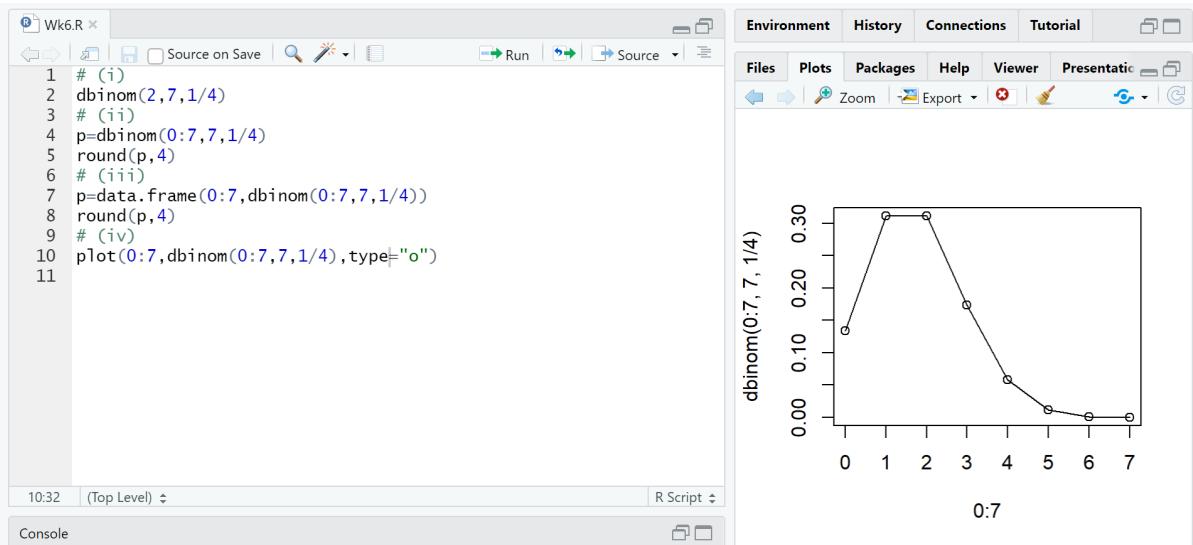
(iv)
plot(0:7,dbinom(0:7,7,1/4),type="o")

OUTPUT:

```
> round(p,4)
[1] 0.1335 0.3115 0.3115 0.1730 0.0577 0.0115 0.0013 0.0001
> # (i)
> dbinom(2,7,1/4)
[1] 0.3114624
> # (ii)
> p=dbinom(0:7,7,1/4)
> round(p,4)
[1] 0.1335 0.3115 0.3115 0.1730 0.0577 0.0115 0.0013 0.0001
> # (iii)
> p=data.frame(0:7,dbinom(0:7,7,1/4))
> round(p,4)
X0.7 dbinom.0.7..7..1.4.
1 0 0.1335
2 1 0.3115
3 2 0.3115
4 3 0.1730
5 4 0.0577
6 5 0.0115
7 6 0.0013
8 7 0.0001
> # (iv)
> plot(0:7,dbinom(0:7,7,1/4),type="o")
```

GRAPH:





PROBLEM 4: Suppose there are twelve multiple choice questions in an English class quiz. Each question has five possible answers, and only one of them is correct. Find the probability of having four or less correct answers if a student attempts to answer every question at random.

PROGRAM CODE:

```
sum(dbinom(1:4,12,0.2))
```

OUTPUT:

```
> sum(dbinom(1:4,12,0.2))
[1] 0.858725
```

The screenshot shows the RStudio interface. The top panel is a script editor titled "Wk6.R" containing the R code: `sum(dbinom(1:4,12,0.2))`. The bottom panel is a console window titled "Console" showing the output of the R code: `> sum(dbinom(1:4,12,0.2))` followed by the result `[1] 0.858725`.

PROBLEM 5: If 10% of the Screws produced by an automatic machine are defective, find the probability that out of 20 screws selected at random, there are

- (i) Exactly 2 defective
- (ii) At least 2 defectives
- (iii) Between 1 and 3 defectives (inclusive)

PROGRAM CODE:

(i)

`dbinom(2,20,0.10)`

(ii)

`1-dbinom(1,20,0.1)`

(iii)

`x=sum(dbinom(1:3,20,0.10))`

OUTPUT:

`> # (i)`

```

> dbinom(2,20,0.10)
[1] 0.2851798
> # (ii)
> 1-dbinom(1,20,0.1)
[1] 0.7298297
> # (iii)
> x=sum(dbinom(1:3,20,0.10))

```

The screenshot shows the RStudio interface. The top panel is a script editor titled "Wk6.R" containing the following R code:

```

1 # (i)
2 dbinom(2,20,0.10)
3 # (ii)
4 1-dbinom(1,20,0.1)
5 # (iii)
6 x=sum(dbinom(1:3,20,0.10))
7

```

The bottom panel is a console window titled "Console" showing the execution of the script. The output is:

```

R 4.2.1 · ~/R
> # (i)
> dbinom(2,20,0.10)
[1] 0.2851798
> # (ii)
> 1-dbinom(1,20,0.1)
[1] 0.7298297
> # (iii)
> x=sum(dbinom(1:3,20,0.10))
>

```

PROBLEM 6: Show that in Binomial distribution variance is less than mean with Binomial variable follows (7,1/4)

PROGRAM CODE:

```

n=7
p=1/4
px=dbinom(0:n,n,p)
x=0:n
Ex=sum(x*px)
Ex
var=sum(x^2*px)-Ex^2
var

```

The screenshot shows the RStudio interface. The top panel is a script editor titled "Wk6.R" containing R code. The code defines variables n=7, p=1/4, calculates probabilities px using dbinom(0:n, n, p), computes the expected value Ex = sum(x*px), and calculates the variance var = sum(x^2*px) - Ex^2. The bottom panel is a console window titled "Console" showing the execution of the script. The R environment is R 4.2.1. The output from the console includes intermediate results like dbinom(2,20,0.10) [1] 0.2851798, and the final result for variance var [1] 1.3125.

```
1 n=7
2 p=1/4
3 px=dbinom(0:n,n,p)
4 x=0:n
5 Ex=sum(x*px)
6 Ex
7 var=sum(x^2*px)-Ex^2
8 var

6:3 (Top Level) ◆ R Script ◆

Console Terminal × Background Jobs ×
R 4.2.1 · ~/🔗
```

```
> # (i)
> dbinom(2,20,0.10)
[1] 0.2851798
> # (ii)
> 1-dbinom(1,20,0.1)
[1] 0.7298297
> # (iii)
> x=sum(dbinom(1:3,20,0.10))
> n=7
> p=1/4
> px=dbinom(0:n,n,p)
> x=0:n
> Ex=sum(x*px)
> Ex
[1] 1.75
> var=sum(x^2*px)-Ex^2
> var
[1] 1.3125
> |
```

Result: The code ran successfully and the output is noted.

Week 7:

Name: Rahul Karthik S

Register Number: 21BEC1851

Poisson Distribution: The random variable X is said to follow the poisson distribution if and only if

$$p[X=x] = \frac{e^{-\lambda} \lambda^x}{x!}, x=0,1,2,\dots$$

Problem 1

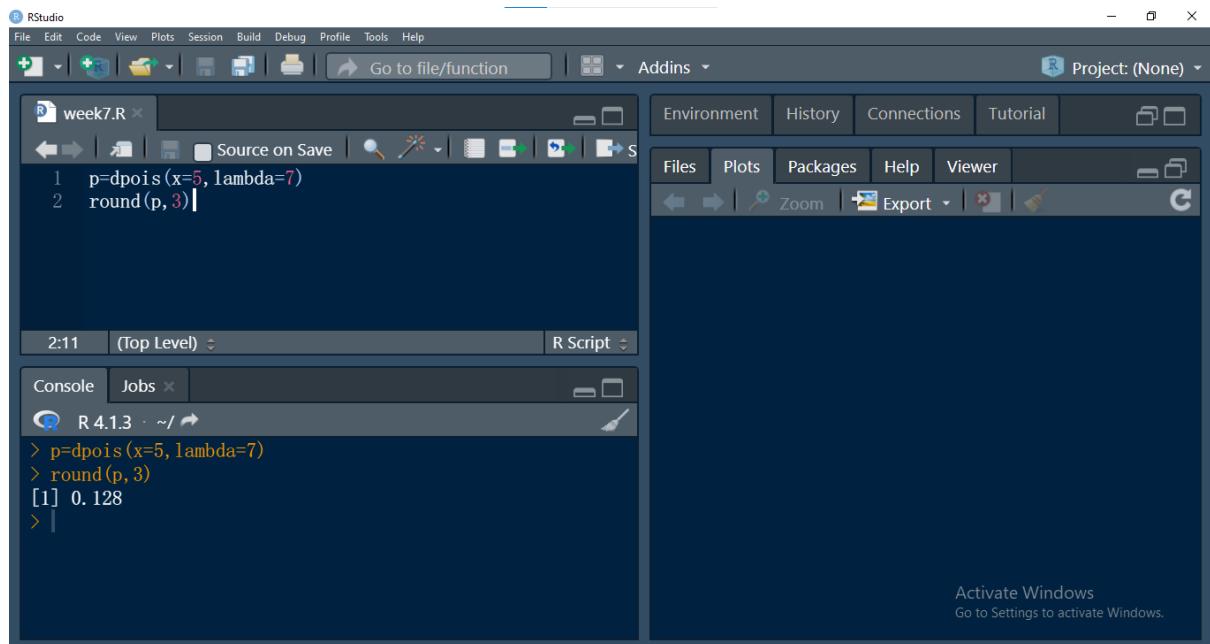
Find P(X=5) with parameter 7 upto 3 decimal places.

Program Code:

```
p=dpois(x=5,lambda=7)
round(p,3)
```

Output:

```
> p=dpois(x=5,lambda=7)
> round(p,3)
[1] 0.128
```



Problem 2

Find P(X=0)+P(X=1)+...+P(X=5) with parameter 7

Program Code:

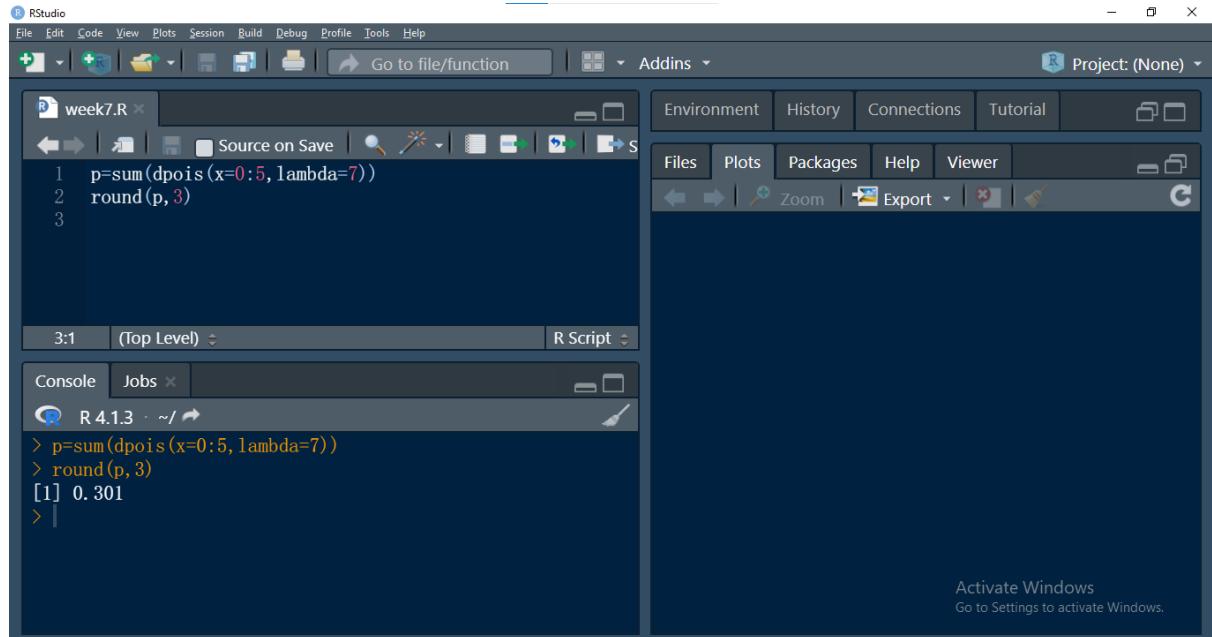
```
p=sum(dpois(x=0:5,lambda=7))
round(p,3)
```

Output:

```

> p=sum(dpois(x=0:5,lambda=7))
> round(p,3)
[1] 0.301

```



Problem 3

Find $P(X \leq 5)$ with parameter 7

Program Code:

```
round(ppois(q=5,lambda=7,lower.tail=T),4)
```

Output:

```

> round(ppois(q=5,lambda=7,lower.tail=T),4)
[1] 0.3007

```

The screenshot shows the RStudio interface. In the top-left pane, a script file named 'week7.R' is open with the following code:

```
1 round(ppois(q=5, lambda=7, lower.tail=F), 4)
```

In the bottom-left pane, the 'Console' tab is active, showing the output of the R session:

```
R 4.1.3 ~ / 
> round(ppois(q=5, lambda=7, lower.tail=F), 4)
[1] 0.3007
```

The right side of the interface contains various toolbars and panels, including 'Environment', 'History', 'Connections', 'Tutorial', 'Files', 'Plots', 'Packages', 'Help', and 'Viewer'.

Problem4

Find $P(X > 5)$ with parameter 7

Program Code:

```
round(ppois(q=5,lambda=7,lower.tail=F),4)
```

Output:

```
> round(ppois(q=5,lambda=7,lower.tail=F),4)
[1] 0.6993
```

The screenshot shows the RStudio interface. In the top-left pane, a script file named 'week7.R' is open with the following code:

```
1 round(ppois(q=5, lambda=7, lower.tail=F), 4)
```

In the bottom-left pane, the 'Console' tab is active, showing the output of the R session:

```
R 4.1.3 ~ / 
> round(ppois(q=5, lambda=7, lower.tail=F), 4)
[1] 0.6993
```

The right side of the interface contains various toolbars and panels, including 'Environment', 'History', 'Connections', 'Tutorial', 'Files', 'Plots', 'Packages', 'Help', and 'Viewer'.

Problem 5

Poisson distribution with parameter 2

1. How to obtain a sequence from 0 to 10
2. Calculate $P(0), (1) \dots (10)$ when $\lambda = 2$ and Make the output prettier

3. Find $P(X \leq 6)$
4. Sum all probabilities
5. Find $P(X > 6)$
6. Make a table of the first 11 Poisson probabilities and cumulative probabilities when $\lambda=2$ obtains the output prettier.
7. Plot the probabilities Put some labels on the axes and give the plot a title

Program Code:

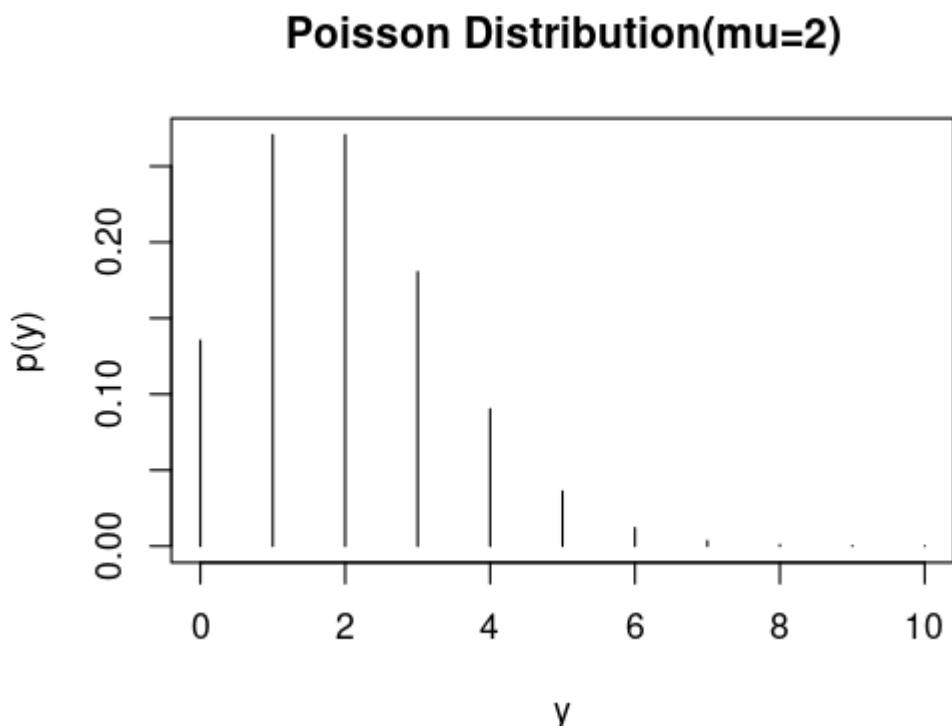
```
#1
0:10
#2
round(dpois(0:10, 2), 3)
#3
ppois(6, 2)
#4
sum(dpois(0:6, 2))
#5
1-ppois(6, 2)
#6
round(cbind(0:10,dpois(0:10,2),ppois (0:10,2))),3)
#7
plot(0:10,dpois(0:10,2),type="h",xlab="y",ylab="p(y)",main="Poisson Distribution(mu=2)")
```

Output:

```
> #1
> 0:10
[1] 0 1 2 3 4 5 6 7 8 9 10
> #2
> round(dpois(0:10, 2), 3)
[1] 0.135 0.271 0.271 0.180 0.090 0.036 0.012 0.003 0.001 0.000 0.000
> #3
> ppois(6, 2)
[1] 0.9954662
> #4
> sum(dpois(0:6, 2))
[1] 0.9954662
> #5
> 1-ppois(6, 2)
[1] 0.004533806
> #6
> round(cbind(0:10,dpois(0:10,2),ppois (0:10,2))),3)
 [,1] [,2] [,3]
[1,] 0 0.135 0.135
[2,] 1 0.271 0.406
[3,] 2 0.271 0.677
[4,] 3 0.180 0.857
[5,] 4 0.090 0.947
[6,] 5 0.036 0.983
```

```
[7,] 6 0.012 0.995  
[8,] 7 0.003 0.999  
[9,] 8 0.001 1.000  
[10,] 9 0.000 1.000  
[11,] 10 0.000 1.000  
> #7  
> plot(0:10,dpois(0:10,2),type="h",xlab="y",ylab="p(y)",main="Poisson Distribution(mu=2)")
```

Plot:



```

File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins R 4.2.1
week7.R x
1 #1
2 0:10
3 #2
4 round(dpois(0:10, 2), 3)
5 #3
6 ppois(6, 2)
7 #4
8 sum(dpois(0:6, 2))
9 #5
10 1-ppois(6, 2)
11 #6
12 round(cbind(0:10,dpois(0:10,2),ppois (0:10,2)),3)
13 #7
14 plot(0:10,dpois(0:10,2),type="h",xlab="y",ylab="p(y)",main="Poisson Distribution(mu=2)")| Run Source
14.89 (Top Level) R Script

```

```

Console Terminal x Background Jobs x
R 4.2.1 ./cloud/project/
> #1
> 0:10
[1] 0 1 2 3 4 5 6 7 8 9 10
> #2
> round(dpois(0:10, 2), 3)
[1] 0.135 0.271 0.271 0.180 0.090 0.036 0.012 0.003 0.001 0.000 0.000
> #3
> ppois(6, 2)
[1] 0.9954662
> #4
> sum(dpois(0:6, 2))
[1] 0.9954662
> #5
> 1-ppois(6, 2)
[1] 0.004533806
> #6
> round(cbind(0:10,dpois(0:10,2),ppois (0:10,2)),3)
[,1] [,2] [,3]
[1,] 0 0.135 0.135
[2,] 1 0.271 0.406
[3,] 2 0.271 0.677
[4,] 3 0.180 0.857
[5,] 4 0.090 0.947
[6,] 5 0.036 0.983
[7,] 6 0.012 0.995
[8,] 7 0.003 0.999
[9,] 8 0.001 1.000
[10,] 9 0.000 1.000
[11,] 10 0.000 1.000
> #7
> plot(0:10,dpois(0:10,2),type="h",xlab="y",ylab="p(y)",main="Poisson Distribution(mu=2)")

```

Problem 6

If there are twelve cars crossing a bridge per minute on average, Find the probability of having 17 or more cars crossing the bridge in a particular minute

Program Code:

`1-ppois(17,12)`

Output:

[1] 0.0629663

The screenshot shows the RStudio IDE interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with various icons for file operations like Open, Save, and Print. A search bar labeled "Go to file/function" is followed by a dropdown for "Addins". On the right side, a status bar indicates "R 4.2.1".

The main area consists of two panes. The left pane is a script editor titled "week7.R" containing the code:

```
1 1-ppois(17,12)
```

The right pane is a "Console" window showing the R session output:

```
R 4.2.1 . /cloud/project/
> 1-ppois(17,12)
[1] 0.0629663
>
```

At the bottom, a status bar shows "1:15" and "(Top Level)".

Week - 8

Normal Distribution

Name: Rahul Karthik S
Register Number: 21BEC1851

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < +\infty$$

R has 4 different built-in functions to generate normal distribution.

dnorm() - It gives height of the probability distribution at each point for a given mean and standard deviation.

`y=dnorm(x, mean=mu, sd=sigma)`

pnorm() - It gives the probability of normally distributed r.v to be less than the value of a given number. It is also called cumulative distribution function.

`y=pnorm(x, mean=mu, sd=sigma)`

qnorm() - It takes probability value and gives a number whose cumulative value matches the probability value.

`y=qnorm(x, mean=mu, sd=sigma)`

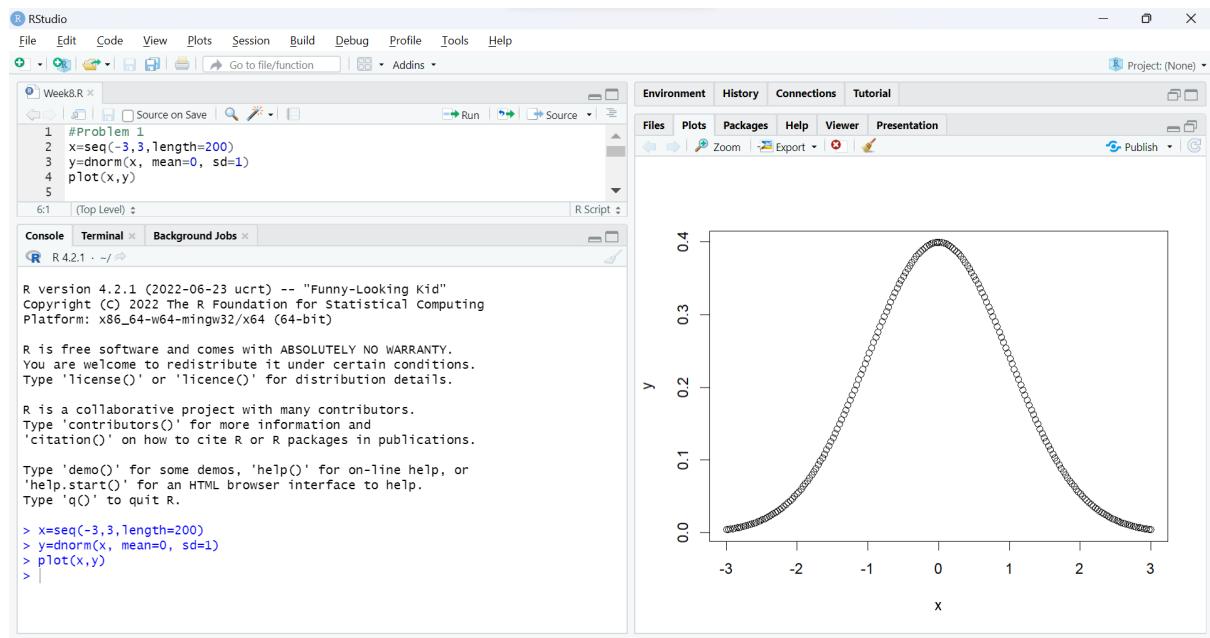
rnorm() - It is used to generate random numbers whose distribution is normal.

`y=rnorm(n)`

Problem 1: Plot a sequence of 200 numbers with x=-3 to 3 with mean 0 and sd 1

Program Code:

```
x=seq(-3,3,length=200)
y=dnorm(x, mean=0, sd=1)
plot(x,y)
```



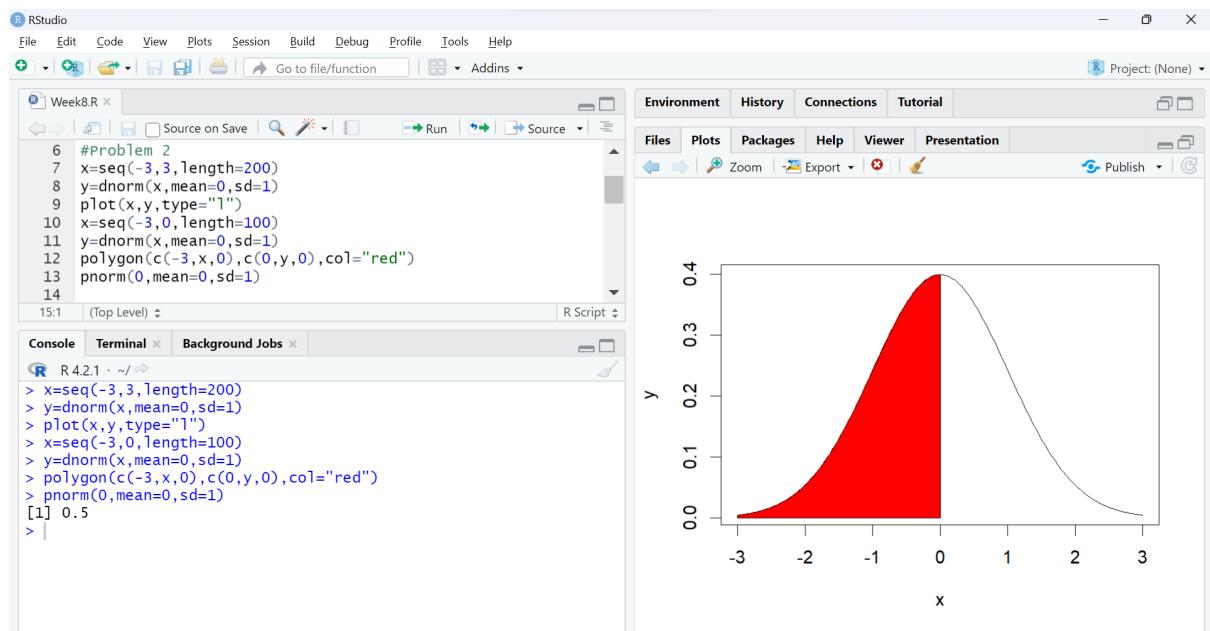
Problem 2: To find the area under the curve to the left of the mean 0 and sd 1 with a sequence of 200 numbers with x=-3 to3.

Program Code:

```

x=seq(-3,3,length=200)
y=dnorm(x,mean=0,sd=1)
plot(x,y,type="l")
x=seq(-3,0,length=100)
y=dnorm(x,mean=0,sd=1)
polygon(c(-3,x,0),c(0,y,0),col="red")
pnorm(0,mean=0,sd=1)

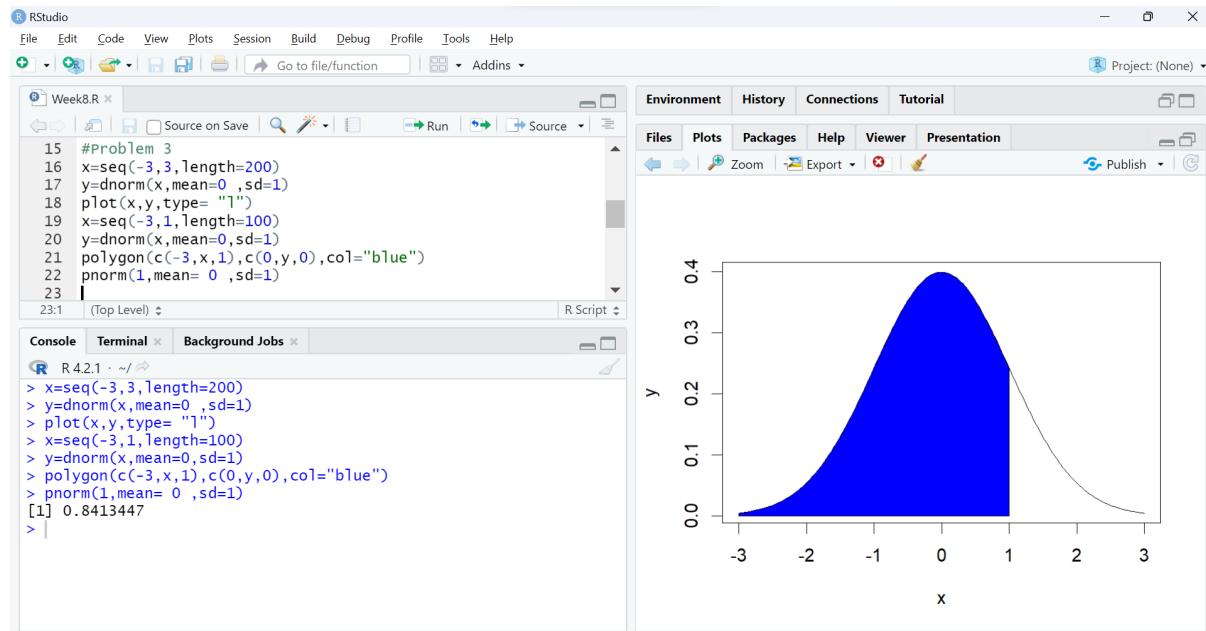
```



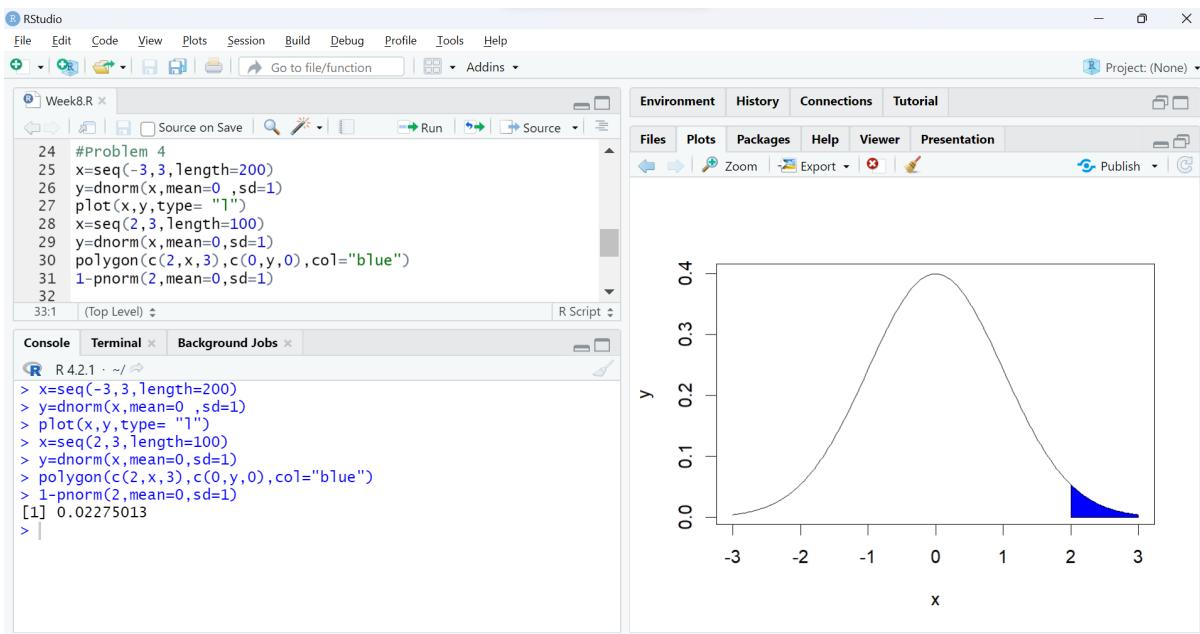
Problem 3: To find area to the left of x=1

Program Code:

```
x=seq(-3,3,length=200)
y=dnorm(x,mean=0 ,sd=1)
plot(x,y,type= "l")
x=seq(-3,1,length=100)
y=dnorm(x,mean=0, sd=1)
polygon(c(-3,x,1),c(0,y,0),col="blue")
pnorm(1,mean= 0 ,sd=1)
```

**Problem 4: Find the area to the right of x=2.****Program Code:**

```
x=seq(-3,3,length=200)
y=dnorm(x,mean=0 ,sd=1)
plot(x,y,type= "l")
x=seq(2,3,length=100)
y=dnorm(x,mean=0, sd=1)
polygon(c(2,x,3),c(0,y,0),col="blue")
1-pnorm(2,mean=0, sd=1)
(# 1-pnorm give the area of the right extreme portion of the normal curve with x=2)
```



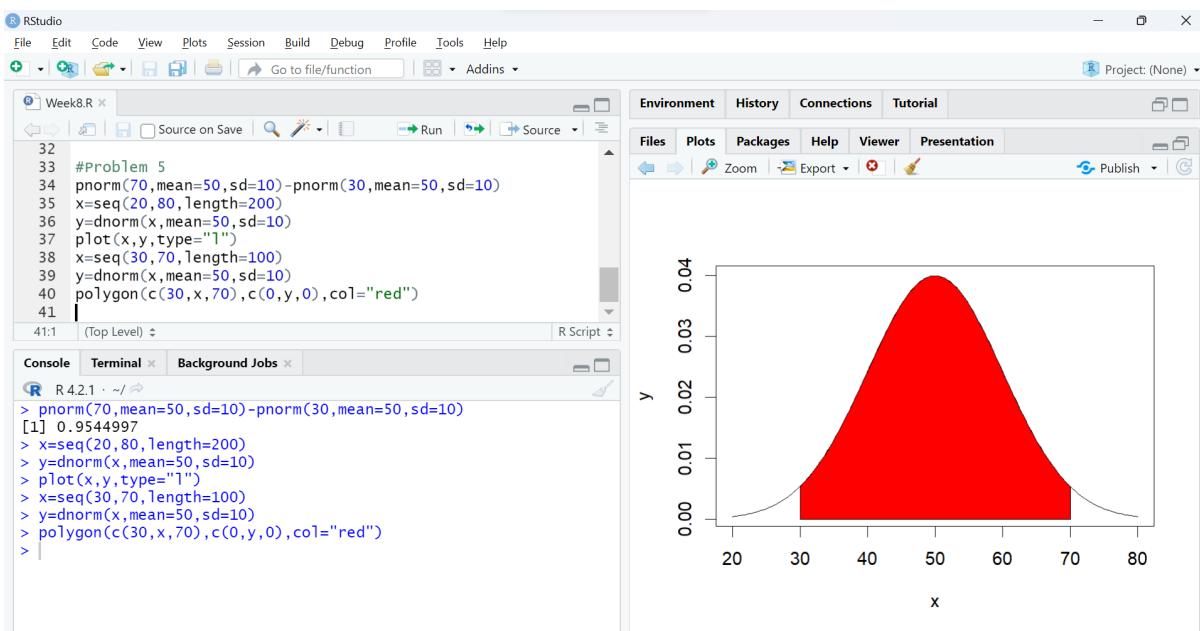
Problem 5: Find $P(30 < X < 70)$ with mean=50 and standard deviation=10.

Program Code:

```

x=seq(20,80,length=200)
y=dnorm(x,mean=50,sd=10)
plot(x,y,type="l")
x=seq(30,70,length=100)
y=dnorm(x,mean=50,sd=10)
polygon(c(30,x,70),c(0,y,0),col="red")
pnorm(70,mean=50,sd=10)-pnorm(30,mean=50,sd=10)

```



Week 9 & 10

Name : Rahul Karthik S
Register Number : 21BEC1851

Week 9:

Problem 1: Suppose that 12% of apples harvested in an orchard last year was rotten. 30 out of 214 apples in a harvest sample this year turns out to be rotten. At .05 significance level, can we reject the null hypothesis that the proportion of rotten apples in harvest stays below 12% this year?

The null hypothesis is that $p \leq 0.12$. $H_0: p = P_0$ $H_1: p > P_0$

Program Code:

```
p=30/214
p0=12/100
q0=1-p0
n=214
z=(p-p0)/sqrt(p0*q0/n)
z
alpha=0.05
zalpha=qnorm(1-alpha)
zalpha
pval=pnorm(z,lower.tail=FALSE)
pval
```

Output:

```
> p=30/214
> p0=12/100
> q0=1-p0
> n=214
> z=(p-p0)/sqrt(p0*q0/n)
> z
[1] 0.908751
> alpha=0.05
> zalpha=qnorm(1-alpha)
> zalpha
[1] 1.644854
> pval=pnorm(z,lower.tail=FALSE)
> pval
[1] 0.1817408
```

(The p-value, or probability value, tells you how likely it is that your data could have occurred under the null hypothesis.)

The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Print, along with a 'Go to file/function' search bar and an 'Addins' dropdown. The main area has tabs for 'Week9.R' (active) and 'Console'. The 'Week9.R' tab contains the following R code:

```

1 p=30/214
2 p0=12/100
3 q0=1-p0
4 n=214
5 z=(p-p0)/sqrt(p0*q0/n)
6 z
7 alpha=0.05
8 zalpha=qnorm(1-alpha)
9 zalpha
10 pval=pnorm(z,lower.tail=FALSE)
11 pval
12

```

The 'Console' tab shows the execution of this code in R version 4.2.1:

```

> p=30/214
> p0=12/100
> q0=1-p0
> n=214
> z=(p-p0)/sqrt(p0*q0/n)
> z
[1] 0.908751
> alpha=0.05
> zalpha=qnorm(1-alpha)
> zalpha
[1] 1.644854
> pval=pnorm(z,lower.tail=FALSE)
> pval
[1] 0.1817408
>

```

Interpretation:

The test statistic 0.90875 is not greater than the critical value of 1.6449. Hence, at 0.05 significance level, we do not reject the null hypothesis that the proportion of rotten apples in harvest stays below 12% this year.

p-value is greater than the 0.05 significance level, we do not reject the null hypothesis that the proportion of rotten apples in harvest stays below 12% this year.

Problem 2: The mean breaking strength of the cables supplied by a manufacturer is 1800 with a s.d of 100. By a new technique in the manufacturing process, it is claimed that the breaking strength of the cable has increased. In order to test this claim, a sample of 50 cables is tested and it is found that the mean breaking strength is 1850. Can we support the claim?

H0:xbar=mu H1: xbar~mu

Program Code:

```

xbar=1850
mu=1800
sigma=100
n=50
z=(xbar-mu)/(sigma/sqrt(n))
z
alpha=0.05
zalpha=qnorm(1-alpha)
zalpha
pval=pnorm(z,lower.tail=FALSE)
pval

```

Output:

```

> xbar=1850
> mu=1800
> sigma=100
> n=50

```

```

> z=(xbar-mu)/(sigma/sqrt(n))
> z
[1] 3.535534
> alpha=0.05
> zalpha=qnorm(1-alpha)
> zalpha
[1] 1.644854
> pval=pnorm(z,lower.tail=FALSE)
> pval
[1] 0.000203476

```

The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with various icons. The main area has tabs for Week9.R (selected), Source on Save, Go to file/function, and Addins. The code editor window contains the R script with syntax highlighting. The console tab at the bottom shows the R session history with colored code and blue output text.

```

1 xbar=1850
2 mu=1800
3 sigma=100
4 n=50
5 z=(xbar-mu)/(sigma/sqrt(n))
6 z
7 alpha=0.05
8 zalpha=qnorm(1-alpha)
9 zalpha
10 pval=pnorm(z,lower.tail=FALSE)
11 pval
12:1 (Top Level) R Script

Console Terminal Background Jobs
R 4.2.1 · ~/R
> xbar=1850
> mu=1800
> sigma=100
> n=50
> z=(xbar-mu)/(sigma/sqrt(n))
> z
[1] 3.535534
> alpha=0.05
> zalpha=qnorm(1-alpha)
> zalpha
[1] 1.644854
> pval=pnorm(z,lower.tail=FALSE)
> pval
[1] 0.000203476
>

```

Week 10:

Problem 1: Suppose the food label on a cookie bag states that there is at most 2 grams of saturated fat in a single cookie. In a sample of 35 cookies, it is found that the mean amount of saturated fat per cookie is 2.1 grams. Assume that the population standard deviation is 0.25 grams. At a 0.05 significance level, can we reject the claim on the food label?

Program Code:

```

xbar=2.1
mu0=2
sigma=0.25
n=35
z=(xbar-mu0)/(sigma/sqrt(n))
z
alpha=0.05
zalpha=qnorm(1-alpha)
zalpha
pval=pnorm(z)
pval

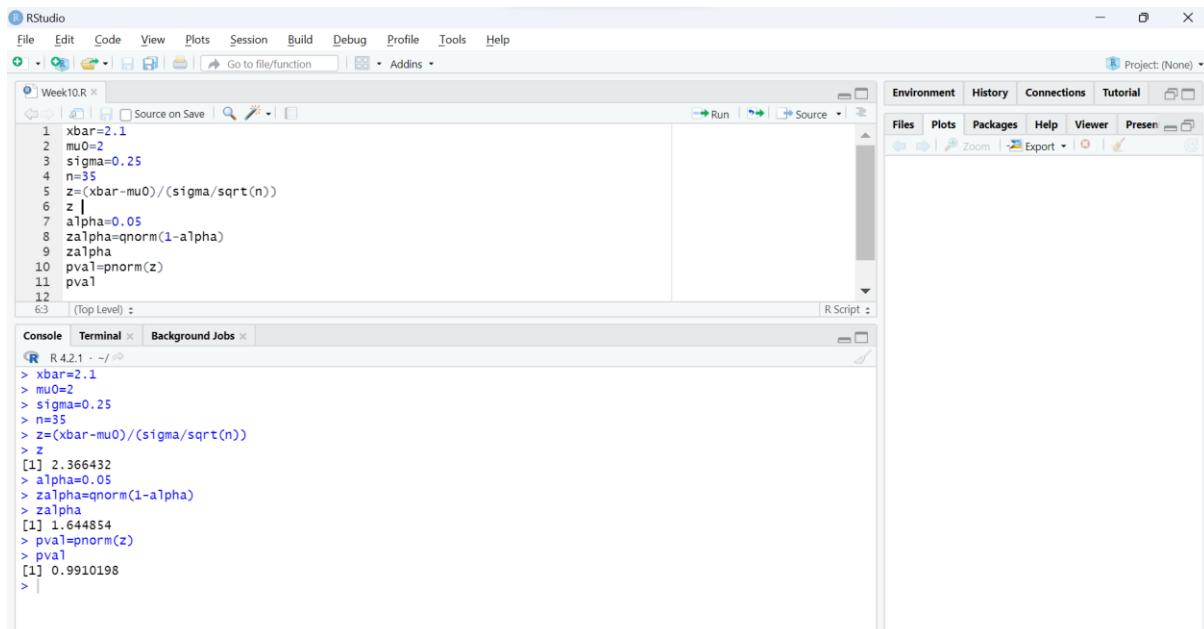
```

Output:

```

> xbar=2.1
> mu0=2
> sigma=0.25
> n=35
> z=(xbar-mu0)/(sigma/sqrt(n))
> z
[1] 2.366432
> alpha=0.05
> zalpha=qnorm(1-alpha)
> zalpha
[1] 1.644854
> pval=pnorm(z)
> pval
[1] 0.9910198
>

```



Interpretation:

The test statistic 2.3664 is greater than the critical value of 1.6449. Hence, at 0.05 significance level, we reject the claim that there is at most 2 grams of saturated fat in a cookie. The upper tail p-value of the test statistic is less than the significance level 0.05, we reject the null hypothesis that $\mu \leq 2$.

Problem 2: An outbreak of salmonella-related illness was attributed to ice produced at a certain factory. Scientists measured the level of Salmonella in 9 randomly sampled batches of ice cream. The levels(in MPN/g) were:

0.593 0.142 0.329 0.691 0.231 0.793 0.519 0.392 0.418

0.593	0.142	0.329	0.691	0.231	0.793	0.519	0.392	0.418
-------	-------	-------	-------	-------	-------	-------	-------	-------

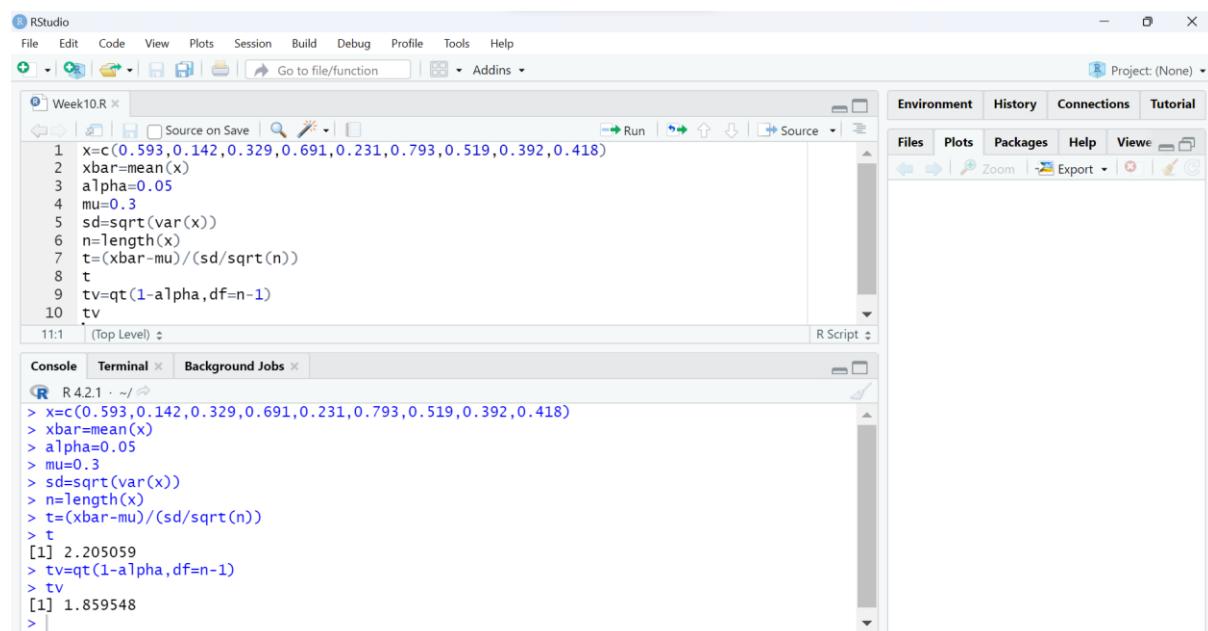
Is there evidence that the mean level of Salmonella in ice cream is greater than 0.3 MPN/g?

Program Code:

```
x=c(0.593,0.142,0.329,0.691,0.231,0.793,0.519,0.392,0.418)
xbar=mean(x)
alpha=0.05
mu=0.3
sd=sqrt(var(x))
n=length(x)
t=(xbar-mu)/(sd/sqrt(n))
t
tv=qt(1-alpha,df=n-1)
tv
```

Output:

```
> x=c(0.593,0.142,0.329,0.691,0.231,0.793,0.519,0.392,0.418)
> xbar=mean(x)
> alpha=0.05
> mu=0.3
> sd=sqrt(var(x))
> n=length(x)
> t=(xbar-mu)/(sd/sqrt(n))
> t
[1] 2.205059
> tv=qt(1-alpha,df=n-1)
> tv
[1] 1.859548
>
```



The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The top right corner shows "Project: (None)". The left pane contains the R script "Week10.R" with the code provided above. The right pane has tabs for Environment, History, Connections, and Tutorial, with Environment selected. The bottom pane contains the "Console" tab, which displays the R session history with the commands and their outputs.

```
1 x=c(0.593,0.142,0.329,0.691,0.231,0.793,0.519,0.392,0.418)
2 xbar=mean(x)
3 alpha=0.05
4 mu=0.3
5 sd=sqrt(var(x))
6 n=length(x)
7 t=(xbar-mu)/(sd/sqrt(n))
8 t
9 tv=qt(1-alpha,df=n-1)
10 tv
11:1 (Top Level) :>
R 4.2.1 - ~/R
> x=c(0.593,0.142,0.329,0.691,0.231,0.793,0.519,0.392,0.418)
> xbar=mean(x)
> alpha=0.05
> mu=0.3
> sd=sqrt(var(x))
> n=length(x)
> t=(xbar-mu)/(sd/sqrt(n))
> t
[1] 2.205059
> tv=qt(1-alpha,df=n-1)
> tv
[1] 1.859548
> |
```

Interpretation:

From the output we see that the $t = 2.205059 > 1.859548$. Reject H_0 . Hence, there is moderately strong evidence that the mean Salmonella level in the ice cream is above 0.3MPN/g.

Result/ Inference:

The code ran successfully and output was verified..