NAME: K. LIKITHA CHOWDARY

CLASS:IT-II-A

ROLLNO:198W1A1225

R LAB-01

TASK-01:

1.PERFORMING BASIC R COMMANDS

```
1+1
1+2+3
3*7*2
4/2
4/3
4*6+5
(4*6)+5
4*(6+5)
x<-2
x
y=5
y
3->z
z
a<-b<-7
a
b
a=b=8
a
b
assign("j",4)
j
rm(j)
```

```
j
i<-5L
i
is.numeric(i)
is.integer(i)
class(4L)
class(2.8)
4L*2.8
class(4L*2.8)
class(5L)
class(2L)
5L/2L
class(5L/2L)
x<-"data"
x
y<-factor("data")
y
z<-factor("hello world")
z
nchar(z)
nchar(x)
nchar("hello")
nchar(3)
date<-as.Date("2012-06-28")
date
class(date)
as.numeric(date)
d<-as.POSIXct("2012-06-28 5:45")
d
class(d)
as.numeric(d)
class(as.numeric(d))
TRUE*5
FALSE*10
```

```
k<-TRUE
class(k)
is.logical(k)
TRUE
T
T<-7
W<-0
W
F<-0
F
class(T)
2==3
7<10
2!=3
2>3
"data"=="stats"
"data"<="stats"
x<-c(1,2,3,4,5,6,7,8,9,10)
x
x*3
x+2
x-2
x^2
x/2
sqrt(x)
1:10
-2:8
10:1
5:-2
x<-1:10
y<- -5:4
x+y
x-y
x*y
```

```
x/y
length(x)
length(y)
length(x*y)
x+c(1,2)
x+c(1,2,3)
x<=5
x<10:1
any(x<y)
all(x<y)
q<-c("hockey","football","baseball","curling","rugby","lacrosse")
nchar(q)
q[0]
c(one="a",two="b",last="c")
w<-1:3
names(w)<-c("a","g","p")
w
q2<-c(q,"waterpool","lacrosse","cricket","football")
q2Factor<-as.factor(q2)
q2Factor
```

OUTPUT:

```
 1+1
[1] 2
> 1+2+3
[1] 6
> 3*7*2
[1] 42
> 4/2
[1] 2
> 4/3
[1] 1.333333
> 4*6+5
[1] 29
```

```
> (4*6)+5
[1] 29
> 4*(6+5)
[1] 44
> 7%3
Error: unexpected input in "7%3"
>
> x<-2
> x
[1] 2
> y
Error: object 'y' not found
>
> y=5
> y
[1] 5
> 3->z
> z
[1] 3
> a<-b<-7
> a
[1] 7
> b
[1] 7
> a=b=8
> a
[1] 8
> b
[1] 8
> assign("j",4)
> j
[1] 4
> rm(j)
> j
```

```
Error: object 'j' not found
> i<-5L
> i
[1] 5
> is.numeric(i)
[1] TRUE
> is.integer(i)
[1] TRUE
> class(4L)
[1] "integer"
> class(2.8)
[1] "numeric"
> 4L*2.8
[1] 11.2
> class(4L*2.8)
[1] "numeric"
> class(5L)
[1] "integer"
> class(2L)
[1] "integer"
> 5L/2L
[1] 2.5
> class(5L/2L)
[1] "numeric"
> x<-"data"
> x
[1] "data"
> y<-factor("data")
> y
[1] data
Levels: data
> z<-factor("hello world")
> z
[1] hello world
```

```
Levels: hello world
> nchar(z)
Error in nchar(z) : 'nchar()' requires a character vector
> nchar(x)
[1] 4
> nchar("hello")
[1] 5
> nchar(3)
[1] 1
> date<-as.Date("2012-06-28")
> date
[1] "2012-06-28"
> class(date)
[1] "Date"
> as.numeric(date)
[1] 15519
> d<-as.POSIXct("2012-06-28 5:45")
> d
[1] "2012-06-28 05:45:00 IST"
> class(d)
[1] "POSIXct" "POSIXt"
> as.numeric(d)
[1] 1340842500
> class(as.numeric(d))
[1] "numeric"
> TRUE*5
[1] 5
> FALSE*10
[1] 0
> k<-TRUE
> class(k)
[1] "logical"
> is.logical(k)
[1] TRUE
```

```
> TRUE
[1] TRUE
> T
[1] TRUE
> T<-7
> W<-0
> W
[1] 0
> F<-0
> F
[1] 0
> class(T)
[1] "numeric"
> 2==3
[1] FALSE
> 7<10
[1] TRUE
> 2!=3
[1] TRUE
> 2>3
[1] FALSE
> "data"=="stats"
[1] FALSE
> "data"<="stats"
[1] TRUE
> x<-c(1,2,3,4,5,6,7,8,9,10)
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> x*3
 [1]  3  6  9 12 15 18 21 24 27 30
> x+2
 [1]  3  4  5  6  7  8  9 10 11 12
> x-2
 [1] -1  0  1  2  3  4  5  6  7  8
```

```
> x^2
 [1]   1   4   9  16  25  36  49  64  81 100
> x/2
 [1] 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> sqrt(x)
 [1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751
 [8] 2.828427 3.000000 3.162278
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
> -2:8
 [1] -2 -1  0  1  2  3  4  5  6  7  8
> 10:1
 [1] 10  9  8  7  6  5  4  3  2  1
> 5:-2
[1]  5  4  3  2  1  0 -1 -2
> x<-1:10
> y<- -5:4
> x+y
 [1] -4 -2  0  2  4  6  8 10 12 14
> x-y
 [1] 6 6 6 6 6 6 6 6 6 6
> x*y
 [1] -5 -8 -9 -8 -5  0  7 16 27 40
> x/y
 [1] -0.2 -0.5 -1.0 -2.0 -5.0  Inf  7.0  4.0  3.0  2.5
> length(x)
[1] 10
> length(y)
[1] 10
> length(x*y)
[1] 10
> x+c(1,2)
 [1]  2  4  4  6  6  8  8 10 10 12
> x+c(1,2,3)
```

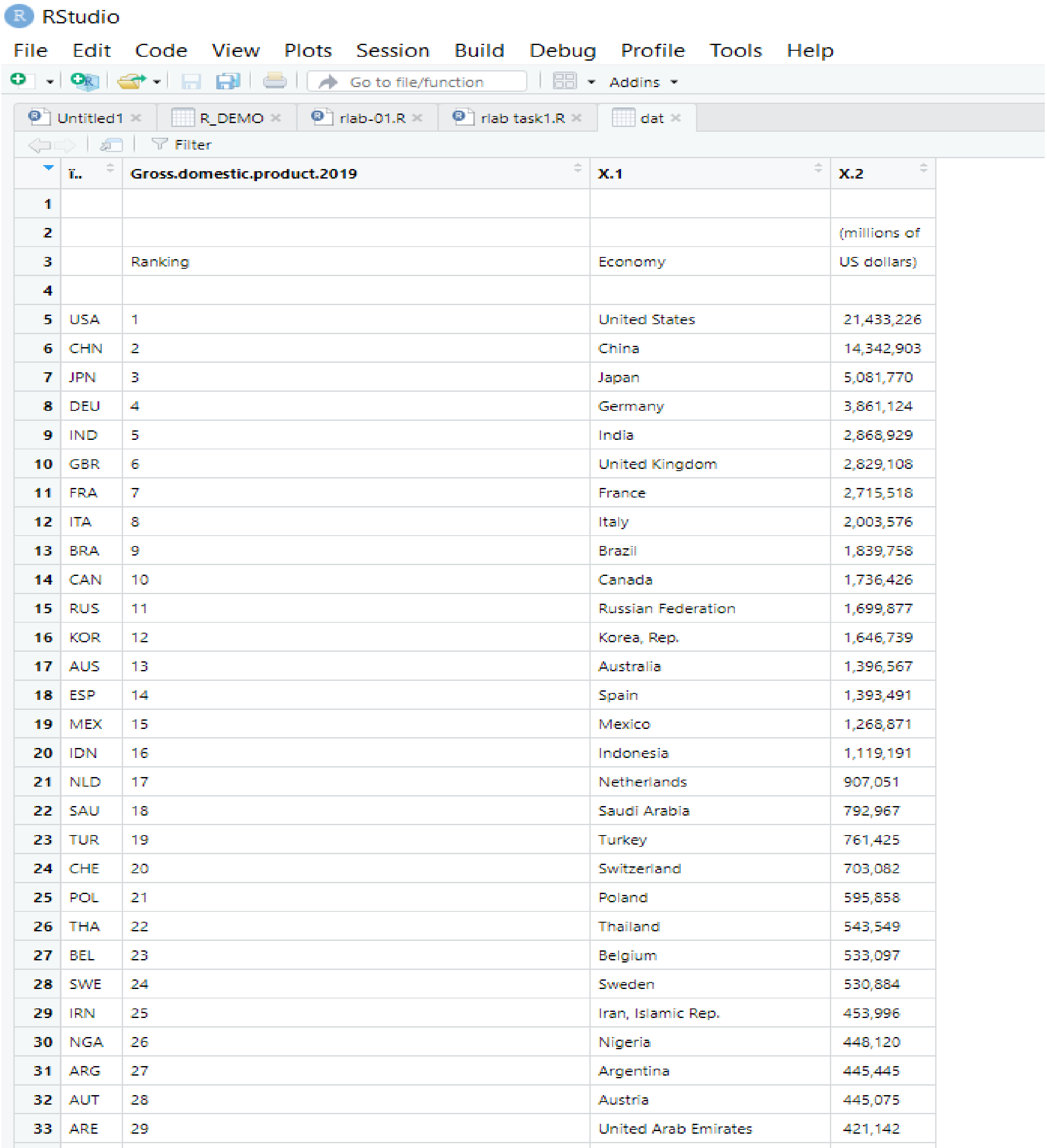
```
 [1]  2  4  6  5  7  9  8 10 12 11
Warning message:
In x + c(1, 2, 3) :
  longer object length is not a multiple of shorter object length
> x<=5
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
> x<10:1
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
> any(x<y)
[1] FALSE
> all(x<y)
[1] FALSE
> q<-c("hockey","football","baseball","curling","rugby","lacrosse")
> nchar(q)
[1] 6 8 8 7 5 8
> q[0]
character(0)
> c=(one="a",two="b",last="c")
Error: unexpected ',' in "c=(one="a","
> c(one="a",two="b",last="c")
 one  two last
 "a"  "b"  "c"
> w<-1:3
> names(w)<-c("a","g","p")
> w
a g p
1 2 3
> q2<-c(q,"waterpool","lacrosse","cricket","football")
> q2Factor<-as.factor(q2)
> q2Factor
 [1] hockey    football  baseball  curling   rugby     lacrosse
 [7] waterpool lacrosse  cricket   football
8 Levels: baseball cricket curling football hockey lacrosse ... waterpool
```

TASK-02:

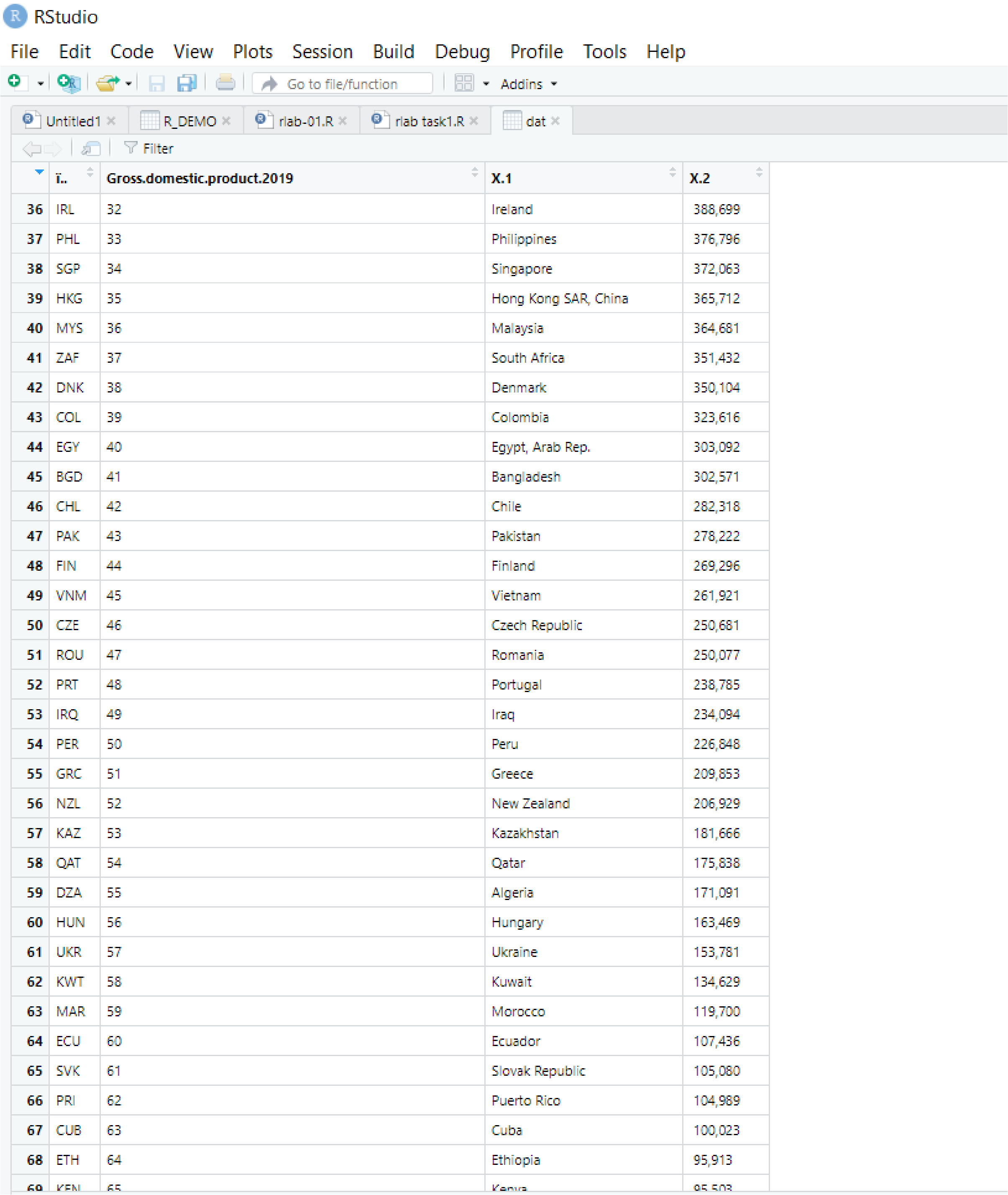
EXTRACTING CSV FILE FROM WEB BROWSER URL

```
dat<-read.csv("https://databank.worldbank.org/data/download/GDP.csv",header=T)
View(dat)


dat<-dat[,-c(3,6,7,8,9,10)]
dat<-dat[-(100:329),]
```

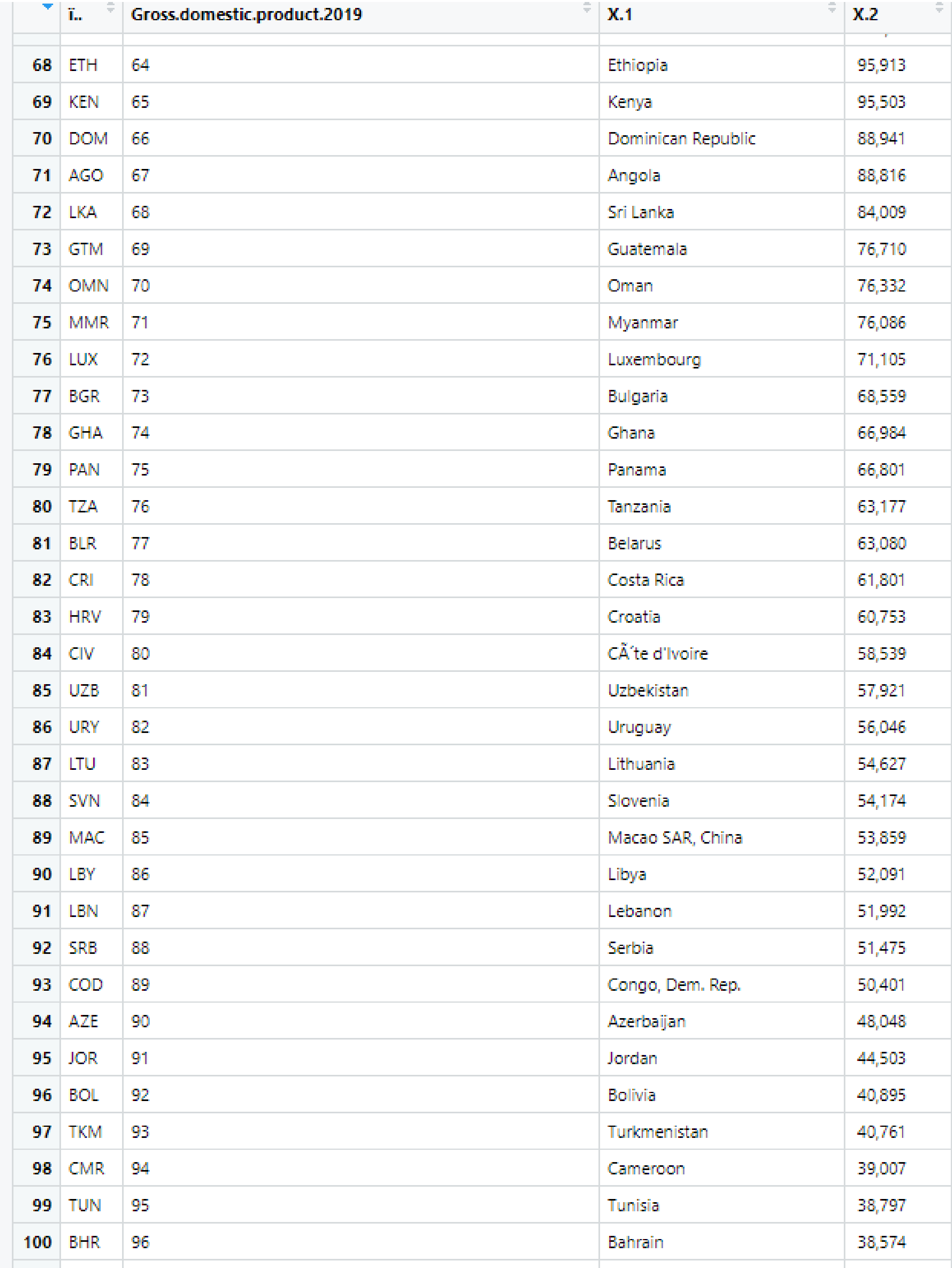
OUTPUT:



	ï..	Gross.domestic.product.2019	X.1	X.2
1				
2				(millions of
3		Ranking	Economy	US dollars)
4				
5	USA	1	United States	21,433,226
6	CHN	2	China	14,342,903
7	JPN	3	Japan	5,081,770
8	DEU	4	Germany	3,861,124
9	IND	5	India	2,868,929
10	GBR	6	United Kingdom	2,829,108
11	FRA	7	France	2,715,518
12	ITA	8	Italy	2,003,576
13	BRA	9	Brazil	1,839,758
14	CAN	10	Canada	1,736,426
15	RUS	11	Russian Federation	1,699,877
16	KOR	12	Korea, Rep.	1,646,739
17	AUS	13	Australia	1,396,567
18	ESP	14	Spain	1,393,491
19	MEX	15	Mexico	1,268,871
20	IDN	16	Indonesia	1,119,191
21	NLD	17	Netherlands	907,051
22	SAU	18	Saudi Arabia	792,967
23	TUR	19	Turkey	761,425
24	CHE	20	Switzerland	703,082
25	POL	21	Poland	595,858
26	THA	22	Thailand	543,549
27	BEL	23	Belgium	533,097
28	SWE	24	Sweden	530,884
29	IRN	25	Iran, Islamic Rep.	453,996
30	NGA	26	Nigeria	448,120
31	ARG	27	Argentina	445,445
32	AUT	28	Austria	445,075
33	ARE	29	United Arab Emirates	421,142



	ï..	Gross.domestic.product.2019	X.1	X.2
36	IRL	32	Ireland	388,699
37	PHL	33	Philippines	376,796
38	SGP	34	Singapore	372,063
39	HKG	35	Hong Kong SAR, China	365,712
40	MYS	36	Malaysia	364,681
41	ZAF	37	South Africa	351,432
42	DNK	38	Denmark	350,104
43	COL	39	Colombia	323,616
44	EGY	40	Egypt, Arab Rep.	303,092
45	BGD	41	Bangladesh	302,571
46	CHL	42	Chile	282,318
47	PAK	43	Pakistan	278,222
48	FIN	44	Finland	269,296
49	VNM	45	Vietnam	261,921
50	CZE	46	Czech Republic	250,681
51	ROU	47	Romania	250,077
52	PRT	48	Portugal	238,785
53	IRQ	49	Iraq	234,094
54	PER	50	Peru	226,848
55	GRC	51	Greece	209,853
56	NZL	52	New Zealand	206,929
57	KAZ	53	Kazakhstan	181,666
58	QAT	54	Qatar	175,838
59	DZA	55	Algeria	171,091
60	HUN	56	Hungary	163,469
61	UKR	57	Ukraine	153,781
62	KWT	58	Kuwait	134,629
63	MAR	59	Morocco	119,700
64	ECU	60	Ecuador	107,436
65	SVK	61	Slovak Republic	105,080
66	PRI	62	Puerto Rico	104,989
67	CUB	63	Cuba	100,023
68	ETH	64	Ethiopia	95,913
69	KEN	65	Kenya	95,503



	ï..	Gross.domestic.product.2019	X.1	X.2
68	ETH	64	Ethiopia	95,913
69	KEN	65	Kenya	95,503
70	DOM	66	Dominican Republic	88,941
71	AGO	67	Angola	88,816
72	LKA	68	Sri Lanka	84,009
73	GTM	69	Guatemala	76,710
74	OMN	70	Oman	76,332
75	MMR	71	Myanmar	76,086
76	LUX	72	Luxembourg	71,105
77	BGR	73	Bulgaria	68,559
78	GHA	74	Ghana	66,984
79	PAN	75	Panama	66,801
80	TZA	76	Tanzania	63,177
81	BLR	77	Belarus	63,080
82	CRI	78	Costa Rica	61,801
83	HRV	79	Croatia	60,753
84	CIV	80	CÃ´te d'Ivoire	58,539
85	UZB	81	Uzbekistan	57,921
86	URY	82	Uruguay	56,046
87	LTU	83	Lithuania	54,627
88	SVN	84	Slovenia	54,174
89	MAC	85	Macao SAR, China	53,859
90	LBY	86	Libya	52,091
91	LBN	87	Lebanon	51,992
92	SRB	88	Serbia	51,475
93	COD	89	Congo, Dem. Rep.	50,401
94	AZE	90	Azerbaijan	48,048
95	JOR	91	Jordan	44,503
96	BOL	92	Bolivia	40,895
97	TKM	93	Turkmenistan	40,761
98	CMR	94	Cameroon	39,007
99	TUN	95	Tunisia	38,797
100	BHR	96	Bahrain	38,574

TASK-03:

EXTRACTING DATA FROM MYSQL:

create table student (rollno integer, name char (100), marks integer);

insert into student(rollno,name,marks) values(1,"ABC",90);

insert into student(rollno,name,marks) values(2,"XYZ",80);

insert into student (rollno, name,marks) values(3,"RES",70);

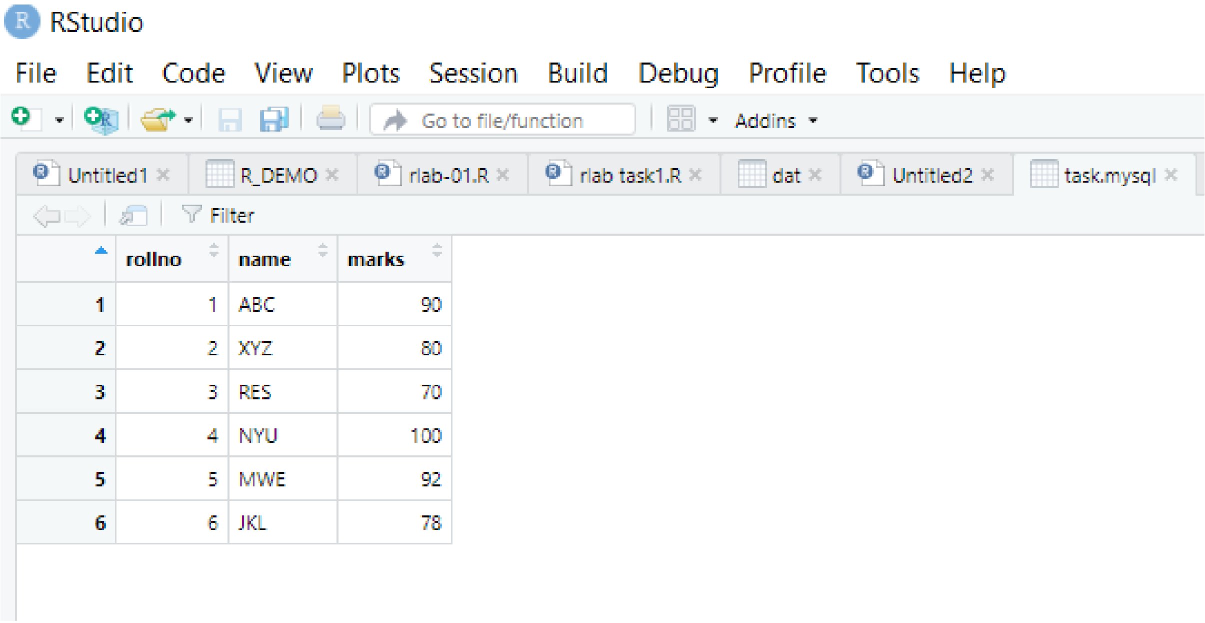
insert into student(rollno,name,marks) values(4,"NYU",100);

insert into student(rollno,name,marks) values(5,"MWE",92);

insert into student(rollno,name,marks) values(6,"JKL",78);

select * from student

OUTPUT:



	rollno	name	marks
1	1	ABC	90
2	2	XYZ	80
3	3	RES	70
4	4	NYU	100
5	5	MWE	92
6	6	JKL	78

R LAB-02

TASK-01:

1.PERFORMING BASIC R COMMANDS:

```
x<-c (10,20,30,40,50)
x
mean(x)
? mean
apropos("mean")
z<-c (1,2, NA,4,5, NA,6)
z
is.na(z)
zchar<-c ("hello”, “world”, NA)
zchar
is.na(zchar)
w<-c (1, NULL,3)
w
is. null (w)
d<-NULL
d
is. null(d)
x<-10:6
y<- -4:0
q<-c("Hockey","Football","baseball","curling","cricket")
thedf<-data. frame (x, y, q)
thedf
thedf<-data. frame (first=x, second=y, sport=q)
thedf
nrow(thedf)
ncol(thedf)
dim(thedf)
names(thedf)
names(thedf) [2]
```

```
rownames(thedf)
rownames(thedf)<-c("one","two","three","four","five")
rownames(thedf)
thedf
rownames(thedf)<-NULL
rownames(thedf)
head(thedf)
head (thedf, n=3)
head (thedf, n=7)
tail(thedf)
tail (thedf, n=2)
class(thedf)
thedf$sport
thedf [3,2]
thedf [3,2:3]
thedf [c (2,3),3]
thedf [c (2,3),2:3]
```

OUTPUT:

```
x<-c (10,20,30,40,50)
> x
[1] 10 20 30 40 50
> mean(x)
[1] 30
>? mean
> apropos("mean")
 [1] “. colMeans"  “. rowMeans"     "colMeans"      "kmeans"
 [5] "mean"          "mean. Date"     "mean. Default"  "mean.difftime"
 [9] "mean.POSIXct"  "mean.POSIXlt"  "rowMeans"      "weighted.mean"
> z<-c (1,2, NA,4,5, NA,6)
> z
[1] 1 2 NA  4 5 NA  6
> is.na(z)
[1] FALSE FALSE TRUE FALSE FALSE TRUE FALSE
```

```
> zchar<-c ("hello”, “world”, NA)
> zchar
[1] "hello" "world" NA
> is.na(zchar)
[1] FALSE FALSE TRUE
> w<-c (1, NULL,3)
> W
[1] 0
> w
[1] 1 3
> is. null(w)
[1] FALSE
> d<-NULL
> d
NULL
> is. null(d)
[1] TRUE
> x<-10:5
> y<- -4:0
> x<-10:6
> x<-10:6
> y<- -4:0
> q<-c("Hockey","Football","baseball","curling","cricket")
> thedf<-data. frame (x, y, q)
> thedf
   x y        q
1 10 -4   Hockey
2 9 -3 Football
3 8 -2 baseball
4 7 -1 curling
5 6 0 cricket
> thedf<-data. Frame (first=x, second=y, sport=q)
> thedf
  first second    sport
```

```
1    10     -4   Hockey
2     9     -3 Football
3     8     -2 baseball
4     7     -1 curling
5     6      0 cricket
> nrow(thedf)
[1] 5
> nrow(thedf)
[1] 5
> ncol(thedf)
[1] 3
> dim(thedf)
[1] 5 3
> names(thedf)
[1] "first" "second" "sport"
> names(thedf)[2]
[1] "second"
> rownames(thedf)
[1] "1" "2" "3" "4" "5"
> rownames(thedf)<-c("one","two","three","four","five")
> rownames(thedf)
[1] "one"   "two"   "three" "four" "five"
> thedf
      first second    sport
one      10     -4   Hockey
two       9     -3 Football
three     8     -2 baseball
four      7     -1 curling
five      6      0 cricket
> rownames(thedf)<-NULL
> rownames(thedf)
[1] "1" "2" "3" "4" "5"
> head(thedf)
  first second    sport
```

```
1    10     -4   Hockey
2     9     -3 Football
3     8     -2 baseball
4     7     -1 curling
5     6      0 cricket
> head (thedf, n=3)
  first second    sport
1    10     -4   Hockey
2     9     -3 Football
3     8     -2 baseball
> head (thedf, n=7)
  first second    sport
1    10     -4   Hockey
2     9     -3 Football
3     8     -2 baseball
4     7     -1 curling
5     6      0 cricket
> tail(thedf)
  first second    sport
1    10     -4   Hockey
2     9     -3 Football
3     8     -2 baseball
4     7     -1 curling
5     6      0 cricket
> tail (thedf, n=2)
  first second   sport
4     7     -1 curling
5     6      0 cricket
> class(thedf)
[1] "data. frame"
thedf$sport
[1] "Hockey"   "Football" "baseball" "curling"  "cricket"
> thedf [3,2]
[1] -2
```

```
> thedf [3,2:3]
  second    sport
3     -2 baseball
> thedf [c (2,3),3]
[1] "Football" "baseball"
> thedf [c (2,3),2:3]
  second    sport
2     -3 Football
3     -2 baseball
```

R LAB-03

TASK-01:

1.PERFORMING BASIC R COMMANDS:

```
> list(1,2,3)
[[1]]
[1] 1

[[2]]
[1] 2

[[3]]
[1]

> list(c(1,2,3))
[[1]]
[1] 1 2 3

> list(1:6)
[[1]]
[1] 1 2 3 4 5 6

> list3<-list(c(1,2,3),3:7)
> list3
[[1]]
[1] 1 2 3

[[2]]
[1] 3 4 5 6 7

> list3<-list(c(1,2,3),3:7)
> list3
[[1]]
[1] 1 2 3

[[2]]
[1] 3 4 5 6 7

> x<-10:1
> y<- -4:5
> q<-c("A","B","C","D","E","F","G","H","I","J")
> theDF<- data.frame(x, y, q)
> theDF
    x  y q
1  10 -4 A
2   9 -3 B
3   8 -2 C
4   7 -1 D
5   6  0 E
6   5  1 F
7   4  2 G
8   3  3 H
9   2  4 I
10  1  5 J
> list(theDF, 1:10)
[[1]]
    x  y q
1  10 -4 A
2   9 -3 B
3   8 -2 C
4   7 -1 D
5   6  0 E
```

```
6   5  1 F
7   4  2 G
8   3  3 H
9   2  4 I
10  1  5 J

[[2]]
 [1]  1  2  3  4  5  6  7  8  9 10

> list5<- list(theDF, 1:10, list3)
> list5
[[1]]
    x  y q
1  10 -4 A
2   9 -3 B
3   8 -2 C
4   7 -1 D
5   6  0 E
6   5  1 F
7   4  2 G
8   3  3 H
9   2  4 I
10  1  5 J

[[2]]
 [1]  1  2  3  4  5  6  7  8  9 10

[[3]]
[[3]][[1]]
[1] 1 2 3
```

```
[[3]][[2]]
[1] 3 4 5 6 7

> names(list5)
[1] "data,frame" "vector"     "list"
> names(list5) <- c("data,frame","vector","list")
> names(list5)
[1] "data,frame" "vector"     "list"
> list5
$`data,frame`
    x  y q
1  10 -4 A
2   9 -3 B
3   8 -2 C
4   7 -1 D
5   6  0 E
6   5  1 F
7   4  2 G
8   3  3 H
9   2  4 I
10  1  5 J

$vector
 [1]  1  2  3  4  5  6  7  8  9 10

$list
$list[[1]]
[1] 1 2 3

$list[[2]]
[1] 3 4 5 6 7
```

```
> list6 <- list(TheDataFrame = theDF, TheVector= 1:10, TheList= list3)
> names(list6)
[1] "TheDataFrame" "TheVector"    "TheList"

> list6
$TheDataFrame
    x  y q
1  10 -4 A
2   9 -3 B
3   8 -2 C
4   7 -1 D
5   6  0 E
6   5  1 F
7   4  2 G
8   3  3 H
9   2  4 I
10  1  5 J

$TheVector
 [1]  1  2  3  4  5  6  7  8  9 10

$TheList
$TheList[[1]]
[1] 1 2 3

$TheList[[2]]
[1] 3 4 5 6 7

> (emptyList <- vector(mode= "list", length=4 ))
```

```
[[1]]
NULL

[[2]]
NULL

[[3]]
NULL

[[4]]
NULL

> list5[[1]]
    x  y q
1  10 -4 A
2   9 -3 B
3   8 -2 C
4   7 -1 D
5   6  0 E
6   5  1 F
7   4  2 G
8   3  3 H
9   2  4 I
10  1  5 J

> list5[["data.frame"]]
NULL
> list5[[1]]$q
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J"

> list5[[1]][ ,"y"]
```

```
 [1] -4 -3 -2 -1  0  1  2  3  4  5

> list5 [[1]][ ,"y", drop= FALSE]
    y
1  -4
2  -3
3  -2
4  -1
5   0
6   1
7   2
8   3
9   4
10  5

> length(list5)
[1] 3
> length(list6)
[1] 3

> list5[[4]] <-2
> length(list5)
[1] 4

> list5[["NewElement"]] <- 3:6
> length(list5)
[1] 5

> names(list5)
[1] "data,frame" "vector"     "list"       ""           "NewElement"
```

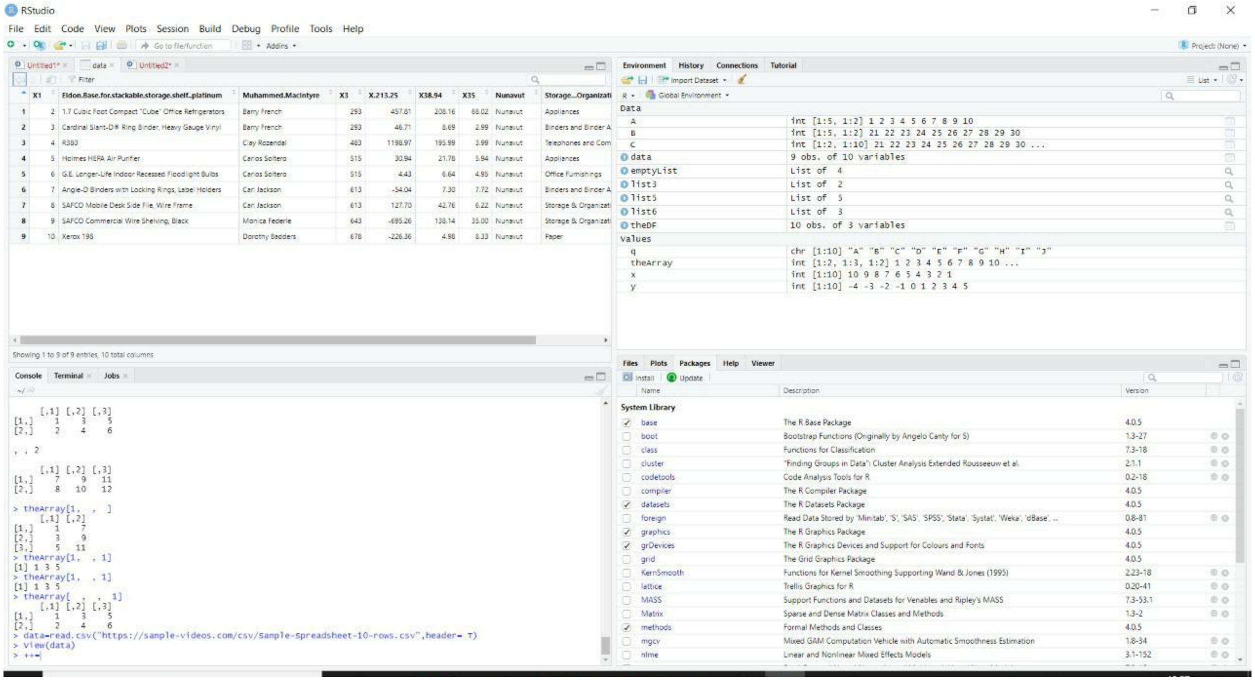
```
> list5
$`data,frame`
    x  y q
1  10 -4 A
2   9 -3 B
3   8 -2 C
4   7 -1 D
5   6  0 E
6   5  1 F
7   4  2 G
8   3  3 H
9   2  4 I
10  1  5 J

$vector
 [1]  1  2  3  4  5  6  7  8  9 10

$list
$list[[1]]
[1] 1 2 3

$list[[2]]
[1] 3 4 5 6 7



[[4]]
[1] 2

$NewElement
[1] 3 4 5 6
```

```
> A<-matrix(1:10, nrow=5)
> B<-matrix(21:30, nrow=5)
> C<-matrix(21:40, nrow=2)
> A
     [,1] [,2]
[1,]    1    6
[2,]    2    7
[3,]    3    8
[4,]    4    9
[5,]    5   10
> B
     [,1] [,2]
[1,]   21   26
[2,]   22   27
[3,]   23   28
[4,]   24   29
[5,]   25   30
> C
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]   21   23   25   27   29   31   33   35   37    39
[2,]   22   24   26   28   30   32   34   36   38    40

> nrow(B)
[1] 5
> ncol(B)
[1] 2
> dim(A)
[1] 5 2
> A+B
     [,1] [,2]
[1,]   22   32
```

```
[2,]   24   34
[3,]   26   36
[4,]   28   38
[5,]   30   40

> A*B
     [,1] [,2]
[1,]   21  156
[2,]   44  189
[3,]   69  224
[4,]   96  261
[5,]  125  300
> A-B
     [,1] [,2]
[1,]  -20  -20
[2,]  -20  -20
[3,]  -20  -20
[4,]  -20  -20
[5,]  -20  -20
> A==B
      [,1]  [,2]
[1,] FALSE FALSE
[2,] FALSE FALSE
[3,] FALSE FALSE
[4,] FALSE FALSE
[5,] FALSE FALSE

> A%*%t(B)
     [,1] [,2] [,3] [,4] [,5]
[1,]  177  184  191  198  205
[2,]  224  233  242  251  260
```

```
[3,]  271  282  293  304  315
[4,]  318  331  344  357  370
[5,]  365  380  395  410  425

> colnames(A)
NULL
> rownames(A)
NULL
> colnames(A) <- c("left","right")
> rownames(A) <- c("list","2nd","3rd","4th","5th")
> colnames(B)
NULL
> rownames(B)
NULL
> colnames(B) <- c("first","second")
> rownames(B) <- c("one","two","three","four","five")
> colnames(C)
NULL

t(A)
      list 2nd 3rd 4th 5th
left     1   2   3   4   5
right    6   7   8   9  10

> A%*%C
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
list  153  167  181  195  209  223  237  251  265   279
2nd   196  214  232  250  268  286  304  322  340   358
3rd   239  261  283  305  327  349  371  393  415   437
4th   282  308  334  360  386  412  438  464  490   516
5th   325  355  385  415  445  475  505  535  565   595
```

```
> theArray <-array(1:12, dim=c(2,3,2))
> theArray
, , 1

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2

     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12

> theArray[1,  ,  ]
     [,1] [,2]
[1,]    1    7
[2,]    3    9
[3,]    5   11
> theArray[1,  , 1]
[1] 1 3 5
> theArray[1,  , 1]
[1] 1 3 5
> theArray[  ,  ,  1]
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

> data=read.csv("https://sample-videos.com/csv/Sample-Spreadsheet-10-rows.csv",header= T)
> View(data)
```



	X1	Eldon.Base.for.stackable.storage.shelf..platinum	Muhammed.MacIntyre	X3	X.213.25	X38.94	X35	Nunavut	Storage...Organizati
1	2	1.7 Cubic Foot Compact "Cube" Office Refrigerators	Barry French	293	457.81	208.16	68.02	Nunavut	Appliances
2	3	Cardinal Slant-D® Ring Binder, Heavy Gauge Vinyl	Barry French	293	46.71	8.69	2.99	Nunavut	Binders and Binder A
3	4	R380	Clay Rozendal	483	1198.97	195.99	3.99	Nunavut	Telephones and Com
4	5	Holmes HEPA Air Purifier	Carlos Soltero	515	30.94	21.78	5.94	Nunavut	Appliances
5	6	G.E. Longer-Life Indoor Recessed Floodlight Bulbs	Carlos Soltero	515	4.43	6.64	4.95	Nunavut	Office Furnishings
6	7	Angle-D Binders with Locking Rings, Label Holders	Carl Jackson	613	-54.04	7.30	7.72	Nunavut	Binders and Binder A
7	8	SAFCO Mobile Desk Side File, Wire Frame	Carl Jackson	613	127.70	42.76	6.22	Nunavut	Storage & Organizati
8	9	SAFCO Commercial Wire Shelving, Black	Monica Federle	643	-695.26	138.14	35.00	Nunavut	Storage & Organizati
9	10	Xerox 198	Dorothy Badders	678	-226.36	4.98	8.33	Nunavut	Paper

Showing 1 to 9 of 9 entries, 10 total columns

```
Data
  A            int [1:5, 1:2] 1 2 3 4 5 6 7 8 9 10
  B            int [1:5, 1:2] 21 22 23 24 25 26 27 28 29 30
  C            int [1:2, 1:10] 21 22 23 24 25 26 27 28 29 30 ...
 data          9 obs. of 10 variables
 emptyList     List of  4
 list3         List of  2
 list5         List of  5
 list6         List of  3
 theDF         10 obs. of 3 variables
Values
  q            chr [1:10] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J"
  theArray     int [1:2, 1:3, 1:2] 1 2 3 4 5 6 7 8 9 10 ...
  x            int [1:10] 10 9 8 7 6 5 4 3 2 1
  y            int [1:10] -4 -3 -2 -1 0 1 2 3 4 5
```

```
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2

     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12

> theArray[1,  ,  ]
     [,1] [,2]
[1,]    1    7
[2,]    3    9
[3,]    5   11
> theArray[1,  , 1]
[1] 1 3 5
> theArray[1,  , 1]
[1] 1 3 5
> theArray[  ,  ,  1]
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> data=read.csv("https://sample-videos.com/csv/Sample-Spreadsheet-10-rows.csv",header= T)
> View(data)
> ++=
```

	Name	Description	Version
	System Library		
☑	base	The R Base Package	4.0.5
□	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-27
□	class	Functions for Classification	7.3-18
□	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.1.1
□	codetools	Code Analysis Tools for R	0.2-18
□	compiler	The R Compiler Package	4.0.5
☑	datasets	The R Datasets Package	4.0.5
□	foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-81
☑	graphics	The R Graphics Package	4.0.5
☑	grDevices	The R Graphics Devices and Support for Colours and Fonts	4.0.5
□	grid	The Grid Graphics Package	4.0.5
□	KernSmooth	Functions for Kernel Smoothing Supporting Wand & Jones (1995)	2.23-18
□	lattice	Trellis Graphics for R	0.20-41
□	MASS	Support Functions and Datasets for Venables and Ripley's MASS	7.3-53.1
□	Matrix	Sparse and Dense Matrix Classes and Methods	1.3-2
☑	methods	Formal Methods and Classes	4.0.5
□	mgcv	Mixed GAM Computation Vehicle with Automatic Smoothness Estimation	1.8-34
□	nlme	Linear and Nonlinear Mixed Effects Models	3.1-152

R LAB-04

TASK-01:

WRITE A R-PROGRAM TO CHECK WHETHER A NUMBER IS EVEN OR ODD

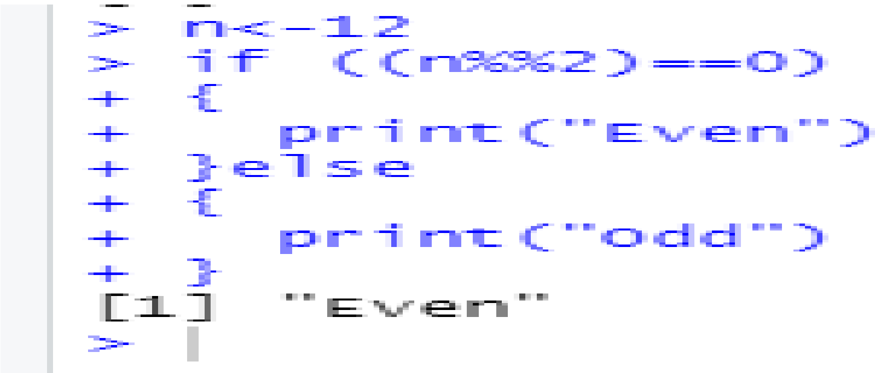
PROGRAM:

```
n<-12
if ((n%%2) ==0)
{
  print("Even")
} else
{
  print("Odd")
}
```

OUTPUT:

```
[1] "Even"
```

EDITOR CODE:



```
> n<-12
> if ((n%%2)==0)
+ {
+     print("Even")
+ }else
+ {
+     print("odd")
+ }
 [1] "Even"
> 
```

RESULT:

Hence the code is executed successfully.

TASK-02:

WRITE THE R-PROGRAM TO CHECK WHETHER NUMBER IS PALINDROME OR NOT.

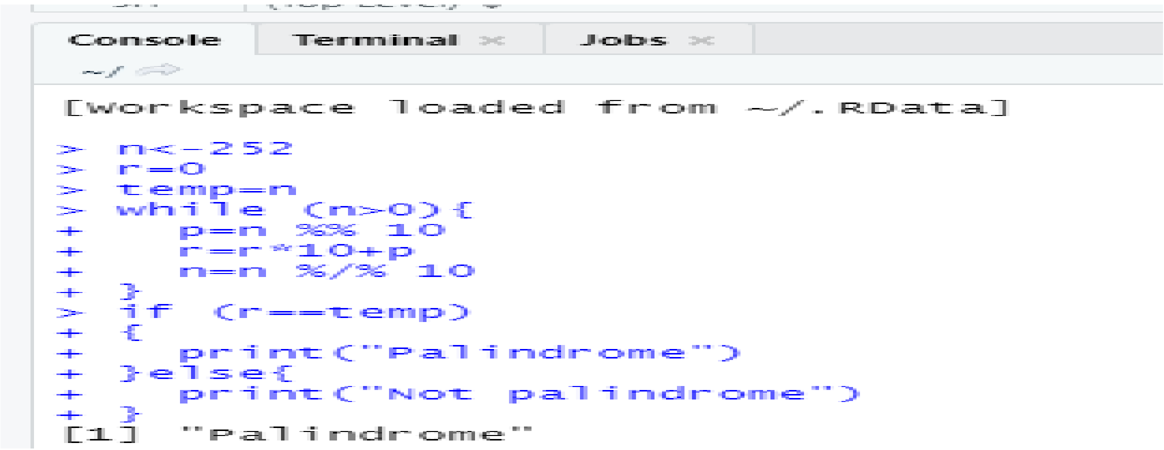
PROGRAM:

```
n<-252
r=0
temp=n
while (n>0) {
  p=n %% 10
  r=r*10+p
  n=n %/% 10
}
if (r==temp)
{
  print("Palindrome")
} else {
  Print ("Not palindrome")
}
```

OUTPUT:

```
[1] "Palindrome"
```

EDITOR CODE:



```
[Workspace loaded from ~/.RData]

> n<-252
> r=0
> temp=n
> while (n>0){
+     p=n %% 10
+     r=r*10+p
+     n=n %/% 10
+ }
> if (r==temp)
+ {
+     print("Palindrome")
+ }else{
+     print("Not palindrome")
+ }
[1] "Palindrome"
```

RESULT:

Hence the code is executed successfully.

TASK-03:

WRITE A R-PROGRAM TO PRINT FIBONACCI SERIES

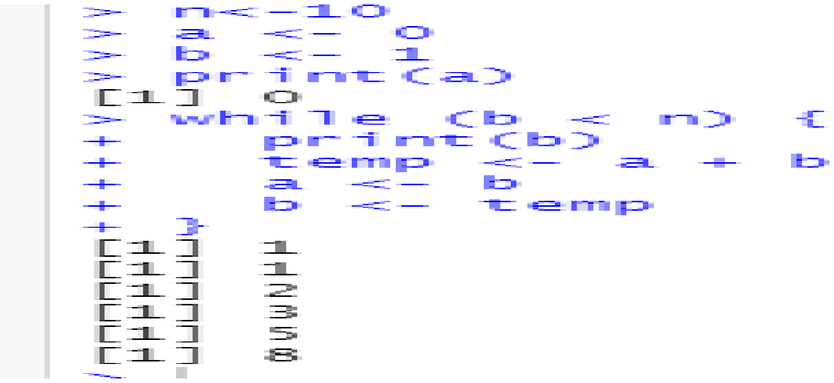
PROGRAM:

```
n<-10
a <- 0
b <- 1
print(a)
while (b < n) {
  print(b)
  temp <- a + b
  a <- b
  b <- temp
}
```

OUTPUT:

```
[1] 0
[1] 1
[1] 1
[1] 2
[1] 3
[1] 5
[1] 8
```

EDITOR CODE:



```
> n<-10
> a <- 0
> b <- 1
> print(a)
[1] 0
> while (b < n) {
+     print(b)
+     temp <- a + b
+     a <- b
+     b <- temp
+ }
[1] 1
[1] 1
[1] 2
[1] 3
[1] 5
[1] 8
>
```

RESULT:

Hence the code is executed successfully.

TASK-4:

WRITE A R-PROGRAM TO CHECK WHETHER IS ARMSTRONG OR NOT.

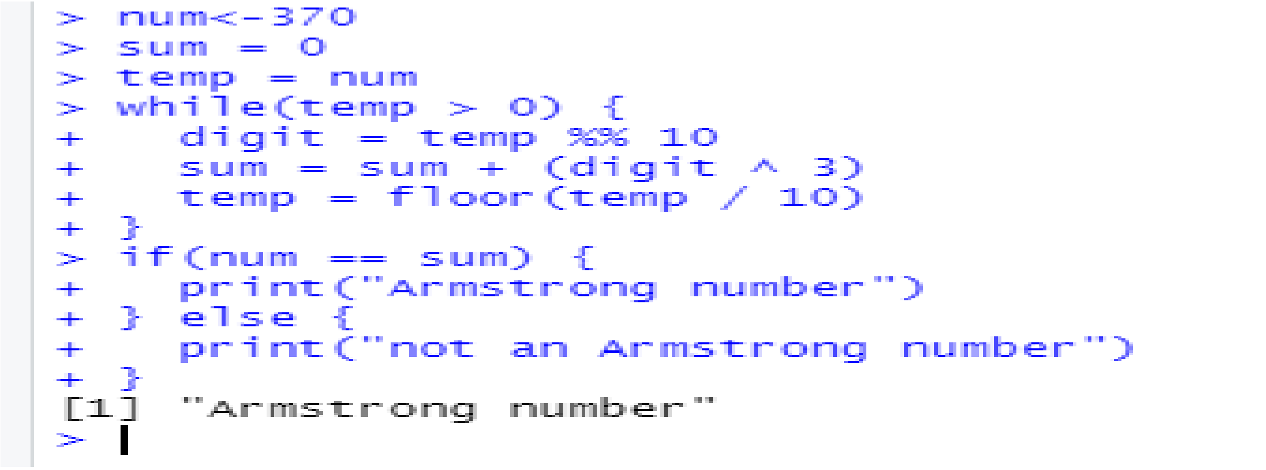
PROGRAM:

```
num<-370
sum = 0
temp = num
while (temp > 0) {
  digit = temp %% 10
  sum = sum + (digit ^ 3)
  temp = floor (temp / 10)
}
If (num == sum) {
  Print ("Armstrong number")
} else {
  Print ("not an Armstrong number")
}
```

OUTPUT:

```
[1] "Armstrong number"
```

EDITOR CODE:



```
> num<-370
> sum = 0
> temp = num
> while(temp > 0) {
+    digit = temp %% 10
+    sum = sum + (digit ^ 3)
+    temp = floor(temp / 10)
+ }
> if(num == sum) {
+    print("Armstrong number")
+ } else {
+    print("not an Armstrong number")
+ }
[1] "Armstrong number"
> 
```

Result:

Hence code is executed successfully

Task-05:

WRITE A R PROGRAM TO REVERSE A NUMBER.

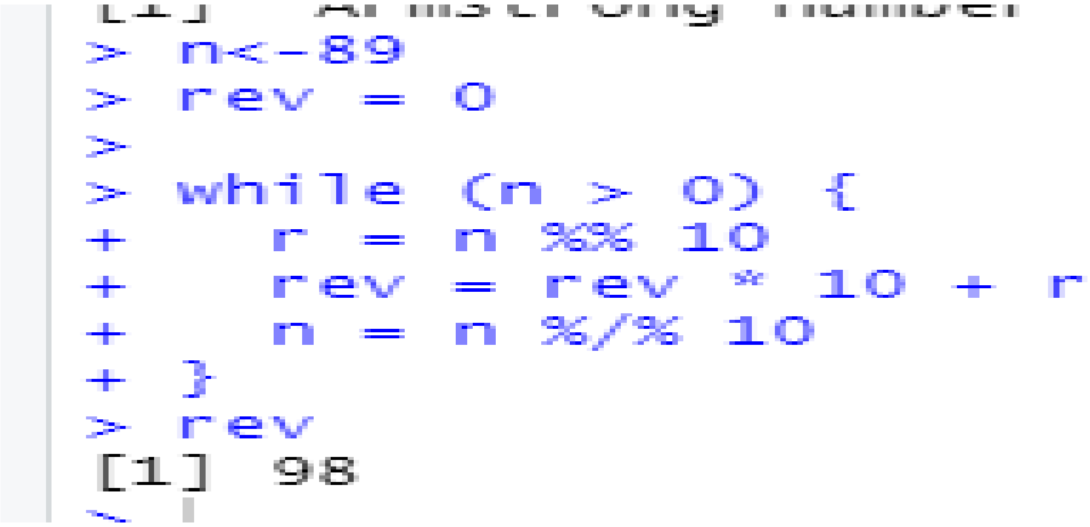
PROGRAM:

```
n<-89
rev = 0
while (n > 0) {
  r = n %% 10
  rev = rev * 10 + r
  n = n %/% 10
}
rev
```

OUTPUT:

```
[1] 98
```

EDITOR CODE:



```
[1] "Armstrong number"
> n<-89
> rev = 0
>
> while (n > 0) {
+     r = n %% 10
+     rev = rev * 10 + r
+     n = n %/% 10
+ }
> rev
[1] 98
>
```

RESULT:

HENCE THE CODE IS EXECUTED SUCCESSFULLY.

TASK-06:

WRITE A R PROGRAM TO FIND FACTORIAL OF A NUMBER.

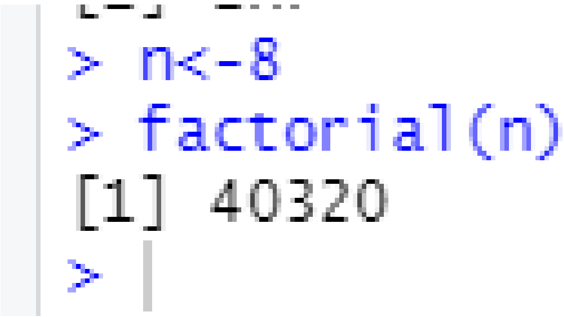
PROGRAM:

```
n<-8
factorial(n)
```

OUTPUT:

```
[1] 40320
```

EDITOR CODE:



```
> n<-8
> factorial(n)
[1] 40320
> 
```

RESULT:

HENCE THE CODE IS EXECUTED SUCCESSFULLY.

TASK-07:

WRITE A R PROGRAM TO SWAP TWO NUMBERS

PROGRAM:

WITH THIRD VARIABLE:

```
a<-10
b<-20
temp=a
a=b
b=temp
a
b
```

WITHOUT THIRD VARIABLE:

```
a<-10
b<-20
b=a+b
```

a=b-a

b=b-a

a

b

OUTPUT:

> a

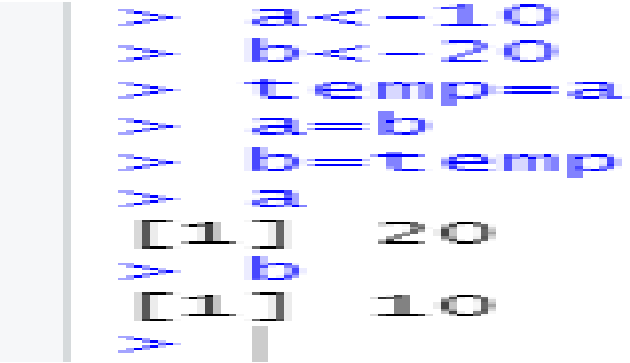
[1] 20

> b

[1] 10

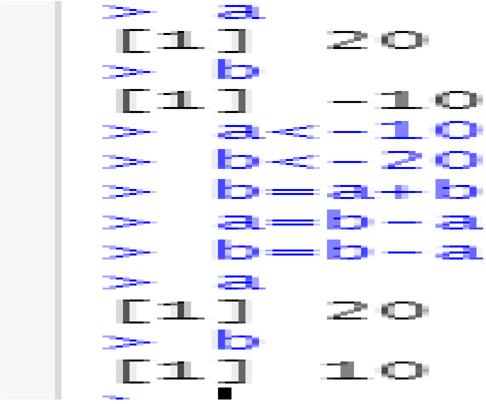
EDITOR CODE:

WITH 3RD VARIABLE:



```
> a<-10
> b<-20
> temp=a
> a=b
> b=temp
> a
[1] 20
> b
[1] 10
>
```

WITHOUT 3RD VARIABLE:



```
> a
[1] 20
> b
[1] -10
> a<-10
> b<-20
> b=a+b
> a=b-a
> b=b-a
> a
[1] 20
> b
[1] 10
```

RESULT:

HENCE THE CODE IS EXECUTED SUCCESSFULLY.

TASK-08:

WRITE A R PROGRAM TO FIND SUM OF DIGITS

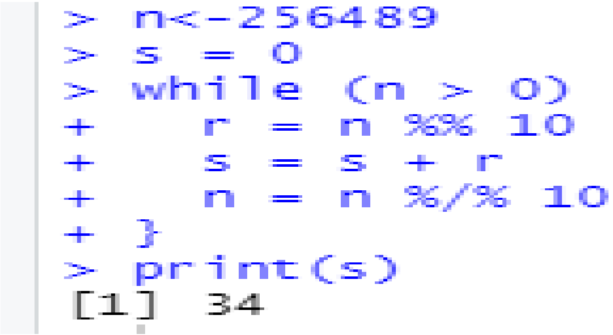
PROGRAM:

```
n<-256489
s = 0
while (n > 0) {
  r = n %% 10
  s = s + r
  n = n %/% 10
}
print(s)
```

OUTPUT:

```
[1] 34
```

EDITOR CODE:



```
> n<-256489
> s = 0
> while (n > 0) {
+     r = n %% 10
+     s = s + r
+     n = n %/% 10
+ }
> print(s)
[1] 34
```

RESULT:

HENCE THE CODE IS EXECUTED SUCCESSFULLY.

TASK-9:

WRITE A R PROGRAM TO PRINT MULTIPLICATION TABLE.

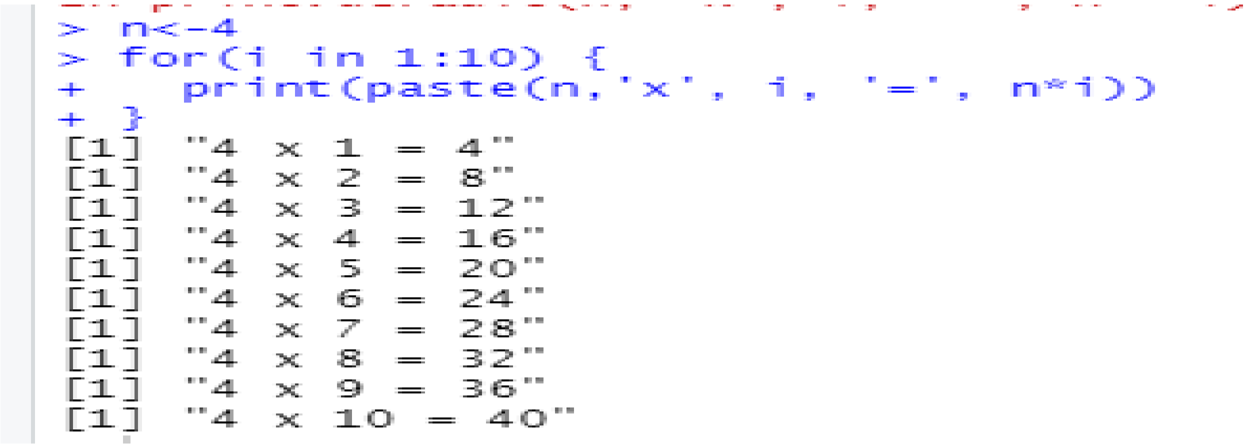
PROGRAM:

```
n<-4
for (i in 1:10) {
  print (paste (n,'x', i, '=', n*i))
}
```

OUTPUT:

```
[1] "4 x 1 = 4"
[1] "4 x 2 = 8"
[1] "4 x 3 = 12"
[1] "4 x 4 = 16"
[1] "4 x 5 = 20"
[1] "4 x 6 = 24"
[1] "4 x 7 = 28"
[1] "4 x 8 = 32"
[1] "4 x 9 = 36"
[1] "4 x 10 = 40"
```

EDITOR CODE:



```
> n<-4
> for(i in 1:10) {
+     print(paste(n,'x', i, '=', n*i))
+ }
[1] "4 x 1 = 4"
[1] "4 x 2 = 8"
[1] "4 x 3 = 12"
[1] "4 x 4 = 16"
[1] "4 x 5 = 20"
[1] "4 x 6 = 24"
[1] "4 x 7 = 28"
[1] "4 x 8 = 32"
[1] "4 x 9 = 36"
[1] "4 x 10 = 40"
```

RESULT:

HENCE THE CODE IS EXECUTED SUCCESSFULLY.

TASK-10:

WRITE A PROGRAM TO CHECK WHETHER NUMBER IS PRIME OR NOT

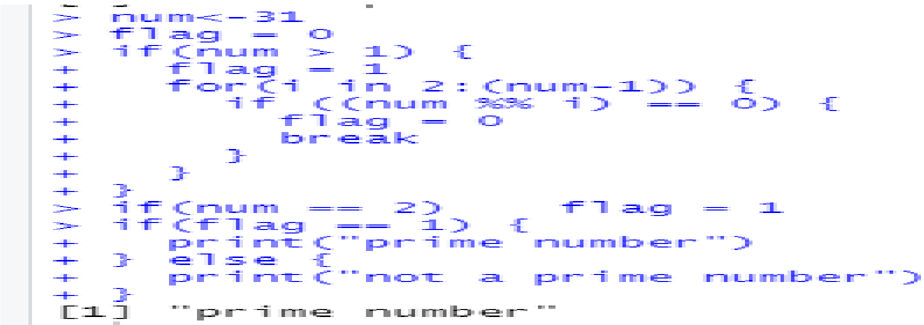
PROGRAM:

```
num<-31
flag = 0
if (num > 1) {
  flag = 1
  for (i in 2:(num-1)) {
    if ((num %% i) == 0) {
      flag = 0
      break
    }
  }
}
If (num == 2)    flag = 1
If (flag == 1) {
  Print ("prime number")
} else {
  Print ("not a prime number")
}
```

OUTPUT:

```
[1] "prime number"
```

EDITOR CODE:



```
> num<-31
> flag = 0
> if(num > 1) {
+     flag = 1
+     for(i in 2:(num-1)) {
+         if ((num %% i) == 0) {
+             flag = 0
+             break
+         }
+     }
+ }
> if(num == 2)     flag = 1
> if(flag == 1) {
+     print("prime number")
+ } else {
+     print("not a prime number")
+ }
[1] "prime number"
```

RESULT:

HENCE THE CODE IS EXECUTED SUCCESSFULLY.

R LAB-05

TASK-01:

Implement function calls and commands:

```
say. hello <-function ()
{
  print ("HELLO WORLD")
}
say. hello ()
sprintf ("HELLO %s","LIKITHA")
sprintf ("HELLO! %s today is %s”,” LIKITHA”, Monday")
hello. person<-function(name)
{
  sprintf ("Likitha and %s are friends”, name)
}
hello. person("Vidya")
hello. person("Harshitha")
two.arg<-function (first, last)
{
  sprintf ("Welcome %s %s”, first, last)
}
two.arg ("Likitha”, Chowdary")
two.arg (first="vidya”, last="Chowdary")
two.arg (last="Lakshmi”, first="Harshitha")
two.arg (first="vrsec”, students")
two.arg (fir="it”, l="students")
my. prog <-function (first, last= “, it student")
{
  sprintf ("Welcome %s %s”, first, last)
}
my. prog("likitha")
my. prog ("likitha”, Chowdary")
ex. prog<-function (first, last="Chowdary”, …)
```

```
{
  print (sprintf ("Hello! %s %s”, first, last))
}
ex. prog ("Likitha”, extra="Goodbye")
double. say<-function(x)
{
  return (x*5)
}
double. say (2)
do. call ("two.arg”, args=list (first="likitha”, last="Chowdary"))
run. this<-function (x, func="mean")
{
  do. call (func, args=list(x))
}
run. this (1:10)
run. this (1:10, sum)
run. this (1:10, sd)
```

output:

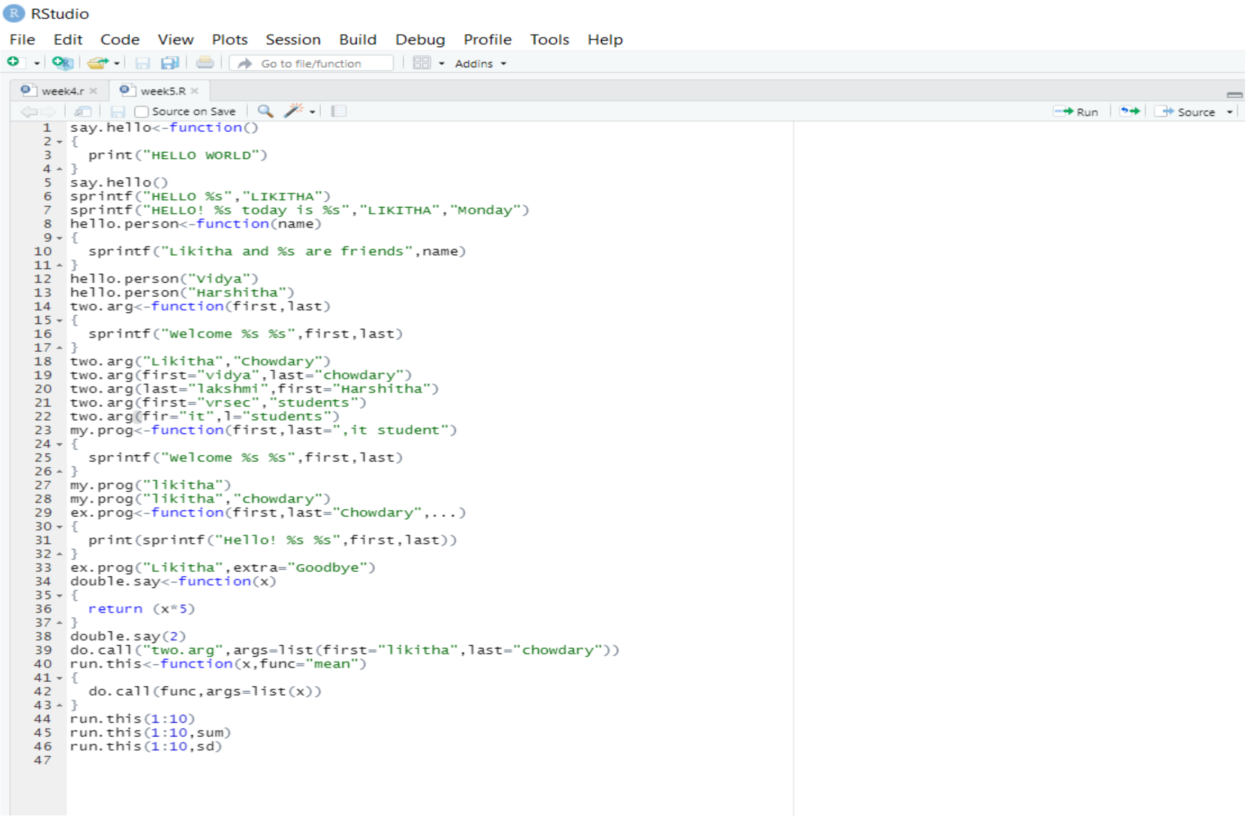
```
> say. hello<-function ()
+ {
+   print ("HELLO WORLD")
+ }
> say. hello ()
[1] "HELLO WORLD"
> sprintf ("HELLO %s","LIKITHA")
[1] "HELLO LIKITHA"
> sprintf ("HELLO! %s today is %s”,” LIKITHA”, Monday")
[1] "HELLO! LIKITHA today is Monday"
> hello. person<-function(name)
+ {
+   sprintf ("Likitha and %s are friends”, name)
+ }
> hello. person("vidya")
```

```
[1] "Likitha and vidya are friends"
> hello. person<-function(name)
+ {
+   sprintf ("Likitha and %s are friends”, name)
+ }
> hello. person ("Vidya")
[1] "Likitha and Vidya are friends"
> hello. person ("Harshitha")
[1] "Likitha and Harshitha are friends"
> two.arg<-function (first, last)
+ {
+   print (sprintf ("Welcome %s %s”, first, last))
+ }
> two.arg ("Likitha”, Chowdary")
[1] "Welcome Likitha Chowdary"
> two.arg<-function (first, last)
+ {
+   sprintf ("Welcome %s %s”, first, last)
+ }
> two.arg ("Likitha”, Chowdary")
[1] "Welcome Likitha Chowdary"
> two.arg<-function (first="Vidya”, last="Chowdary")
+ {
+   sprintf ("Welcome %s %s”, first, last)
+ }
> two.arg<-function (first="Vidya”, last="Chowdary")
+ {
+   sprintf ("Welcome %s %s”, first, last)
+ }
> two.arg<-function (first="Vidya”, last="Chowdary")
+ {
+   sprintf ("Welcome %s %s”, first, last)
+ }
> two.arg("vidya","Chowdary")
```

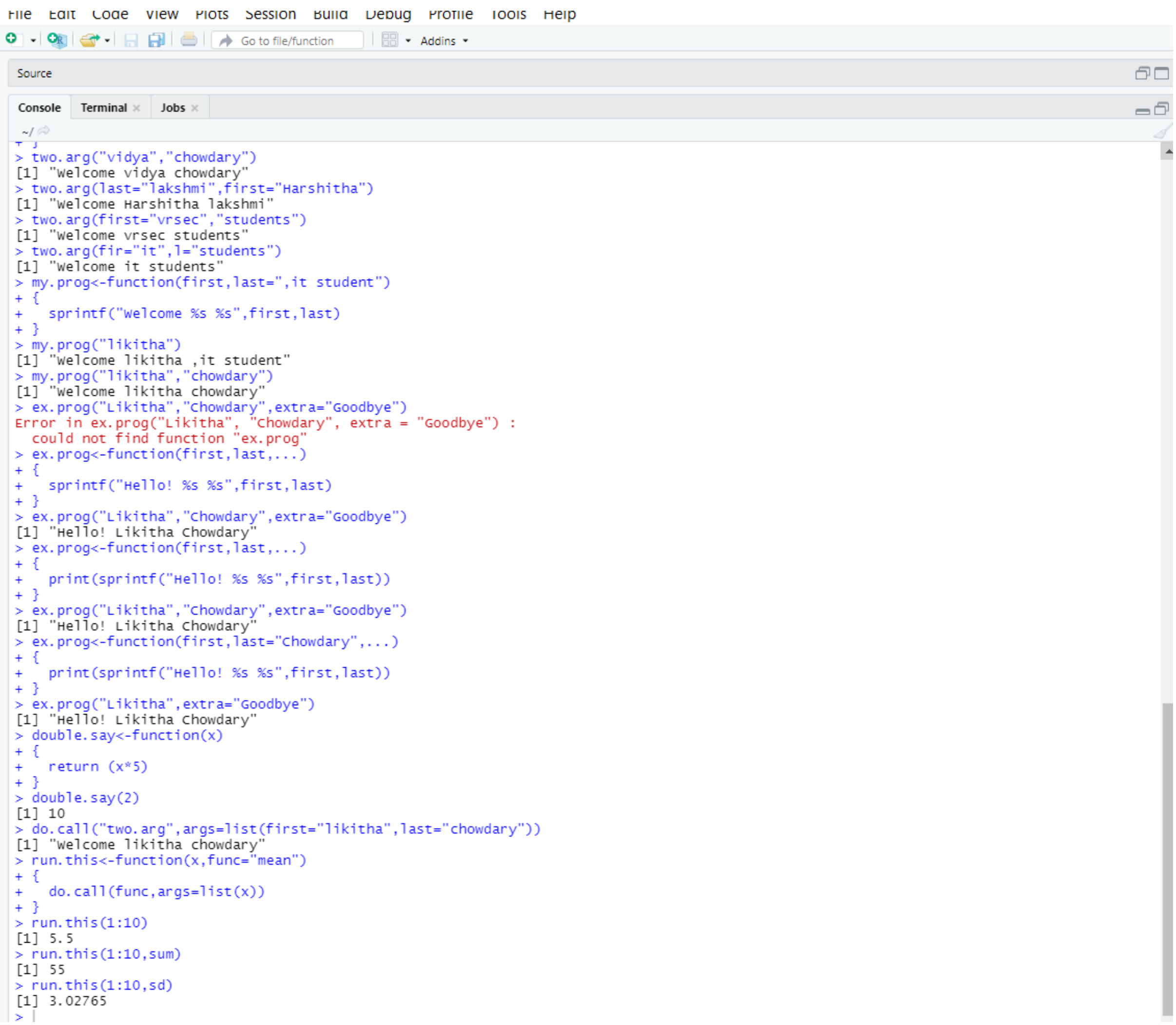
```
[1] "Welcome vidya Chowdary"
> two.arg (last="Lakshmi”, first="Harshitha")
[1] "Welcome Harshitha Lakshmi"
> two.arg (first="vrsec”, students")
[1] "Welcome vrsec students"
> two.arg (fir="it”, l="students")
[1] "Welcome it students"
> my. prog<-function (first, last= “, it student")
+ {
+   sprintf ("Welcome %s %s”, first, last)
+ }
> my. prog ("likitha")
[1] "Welcome likitha, it student"
> my. prog ("likitha”, Chowdary")
[1] "Welcome likitha Chowdary"
> ex. prog<-function (first, last, …)
+ {
+   sprintf ("Hello! %s %s”, first, last)
+ }
> ex. prog ("Likitha”, Chowdary”, extra="Goodbye")
[1] "Hello! Likitha Chowdary"
> ex. prog<-function (first, last, …)
+ {
+   print (sprintf ("Hello! %s %s”, first, last))
+ }
> ex. prog ("Likitha”, Chowdary”, extra="Goodbye")
[1] "Hello! Likitha Chowdary"
> ex. prog<-function (first, last="Chowdary",...)
+ {
+   print (sprintf ("Hello! %s %s”, first, last))
+ }
> ex. prog ("Likitha”, extra="Goodbye")
[1] "Hello! Likitha Chowdary"
> double.say <-function(x)
```

```
+ {
+   return (x*5)
+ }
> double. say (2)
[1] 10
> do.call ("two.arg”, args=list(first="likitha”, last="Chowdary"))
[1] "Welcome likitha Chowdary"
> run. this <-function (x, func="mean")
+ {
+   do. call (func, args=list(x))
+ }
> run. this (1:10)
[1] 5.5
> run. this (1:10, sum)
[1] 55
> run. this (1:10, sd)
[1] 3.02765
```

EDITOR CODE:



```
say.hello<-function()
{
  print("HELLO WORLD")
}
say.hello()
sprintf("HELLO %s","LIKITHA")
sprintf("HELLO! %s today is %s","LIKITHA","Monday")
hello.person<-function(name)
{
  sprintf("Likitha and %s are friends",name)
}
hello.person("vidya")
hello.person("Harshitha")
two.arg<-function(first,last)
{
  sprintf("Welcome %s %s",first,last)
}
two.arg("Likitha","Chowdary")
two.arg(first="vidya",last="chowdary")
two.arg(last="lakshmi",first="Harshitha")
two.arg(first="vrsec","students")
two.arg(fir="it",l="students")
my.prog<-function(first,last=",it student")
{
  sprintf("Welcome %s %s",first,last)
}
my.prog("likitha")
my.prog("likitha","chowdary")
ex.prog<-function(first,last="Chowdary",...)
{
  print(sprintf("Hello! %s %s",first,last))
}
ex.prog("Likitha",extra="Goodbye")
double.say<-function(x)
{
  return (x*5)
}
double.say(2)
do.call("two.arg",args=list(first="likitha",last="chowdary"))
run.this<-function(x,func="mean")
{
  do.call(func,args=list(x))
}
run.this(1:10)
run.this(1:10,sum)
run.this(1:10,sd)
```


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
~/

```
+ }
> two.arg("vidya","chowdary")
[1] "Welcome vidya chowdary"
> two.arg(last="lakshmi",first="Harshitha")
[1] "Welcome Harshitha lakshmi"
> two.arg(first="vrsec","students")
[1] "Welcome vrsec students"
> two.arg(fir="it",l="students")
[1] "Welcome it students"
> my.prog<-function(first,last=",it student")
+ {
+   sprintf("Welcome %s %s",first,last)
+ }
> my.prog("likitha")
[1] "Welcome likitha ,it student"
> my.prog("likitha","chowdary")
[1] "Welcome likitha chowdary"
> ex.prog("Likitha","Chowdary",extra="Goodbye")
Error in ex.prog("Likitha", "Chowdary", extra = "Goodbye") :
  could not find function "ex.prog"
> ex.prog<-function(first,last,...)
+ {
+   sprintf("Hello! %s %s",first,last)
+ }
> ex.prog("Likitha","Chowdary",extra="Goodbye")
[1] "Hello! Likitha Chowdary"
> ex.prog<-function(first,last,...)
+ {
+   print(sprintf("Hello! %s %s",first,last))
+ }
> ex.prog("Likitha","Chowdary",extra="Goodbye")
[1] "Hello! Likitha Chowdary"
> ex.prog<-function(first,last="Chowdary",...)
+ {
+   print(sprintf("Hello! %s %s",first,last))
+ }
> ex.prog("Likitha",extra="Goodbye")
[1] "Hello! Likitha Chowdary"
> double.say<-function(x)
+ {
+   return (x*5)
+ }
> double.say(2)
[1] 10
> do.call("two.arg",args=list(first="likitha",last="chowdary"))
[1] "Welcome likitha chowdary"
> run.this<-function(x,func="mean")
+ {
+   do.call(func,args=list(x))
+ }
> run.this(1:10)
[1] 5.5
> run.this(1:10,sum)
[1] 55
> run.this(1:10,sd)
[1] 3.02765
> 
```

RESULT:

Hence the code is executed successfully.

R LAB-06

TASK-01:

Implement the concept of statistical graphs in r studio.

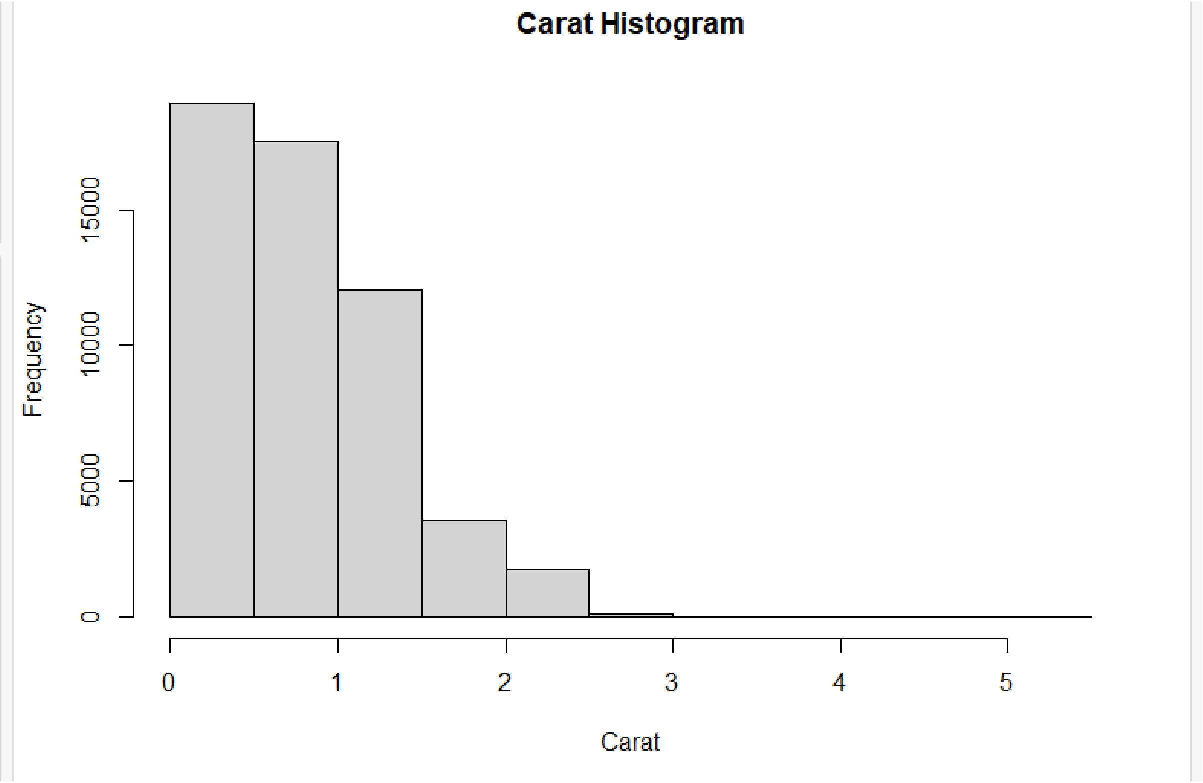
PROGRAM:

```
require(ggplot2)
data(diamonds)
head(diamonds)
hist (diamonds$carat, main="Carat Histogram", xlab= "Carat")
plot (price ~ carat, data=diamonds)
plot (diamonds$carat, diamonds$price)
boxplot(diamonds$carat)
ggplot (data=diamonds) + geom_histogram(aes(x=carat))
ggplot(data=diamonds) + geom_density(aes(x=carat), fill= "red")
ggplot (diamonds, aes (x=carat, y=price,)) +geom_point ()
g<- ggplot (diamonds, aes (x=carat, y=price))
g+geom_point (aes (color=color))
g+geom_point(aes(color=color)) + facet_wrap (~ color)
g+geom_point(aes(color=color)) + facet_grid (cut~clarity)
ggplot (diamonds, aes (y=carat, x=1)) + geom_boxplot ()
ggplot (diamonds, aes (y=carat, x=cut)) + geom_boxplot ()
ggplot (diamonds, aes (y=carat, x=cut)) + geom_violin ()
ggplot (diamonds, aes (y=carat, x=cut)) + geom_point () + geom_violin ()
ggplot (diamonds, aes (y=carat, x=cut)) + geom_violin () + geom_point ()
ggplot (economics, aes (x=date, y= pop)) + geom_line ()

install. packages ("lubridate")
require(lubridate)
economics$year<- year(economics$date)
economics$year
economics$month<- month (economics$date, label=TRUE)
economics$month
econ2000<-economics [which (economics$year >=2000),]
```

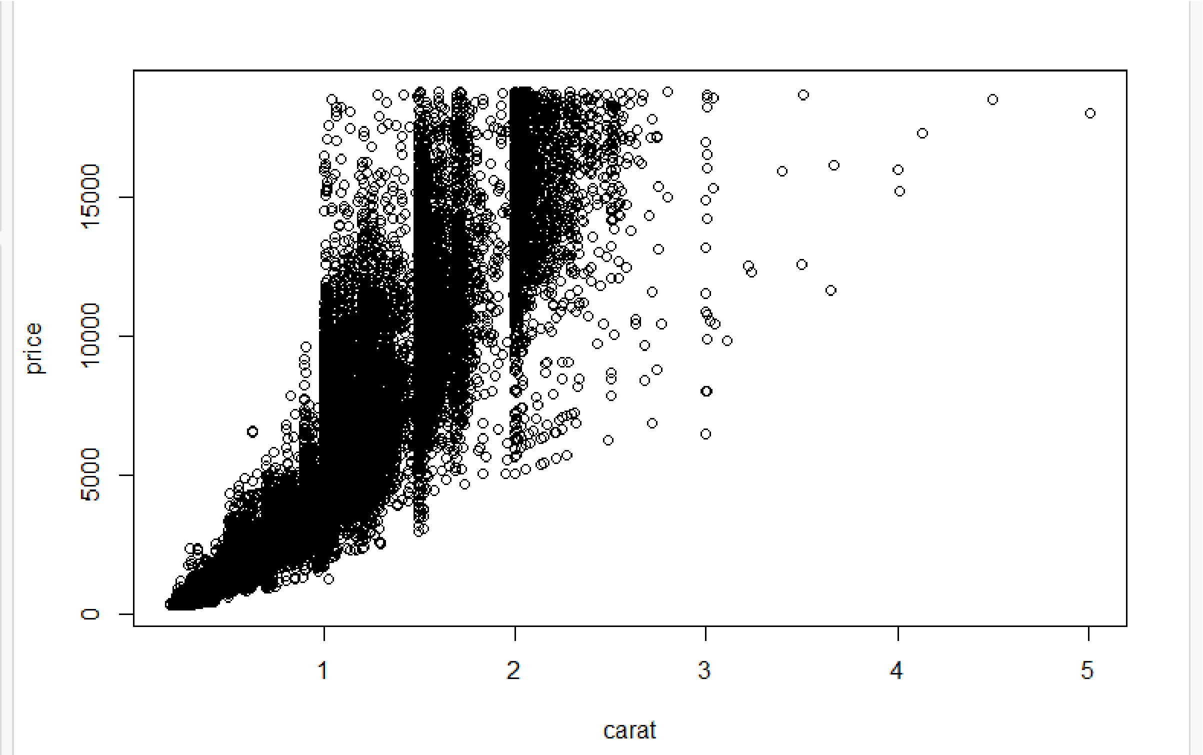
```
econ2000
g<- ggplot (econ2000, aes (x=month, y=pop))
g<-g + geom_line(aes(color=factor(year), group=year))
g<- g+scale_color_discrete(name="Year")
g<- g+ scale_y_continuos(labels=comma)
g<-g+ labs (title="Population Growth", x="Month", y="Population")
g

install. packages("ggthemes")
require(ggthemes)
g2<- ggplot (diamonds, aes (x=carat, y=price)) + geom_point(aes(color=color))
g2 + theme_economist () + scale_colour_economist ()
g2+ theme_excel () + scale_colour_excel ()
g2+ theme_tufte ()
g2+theme_wsj ()
```

OUTPUT:

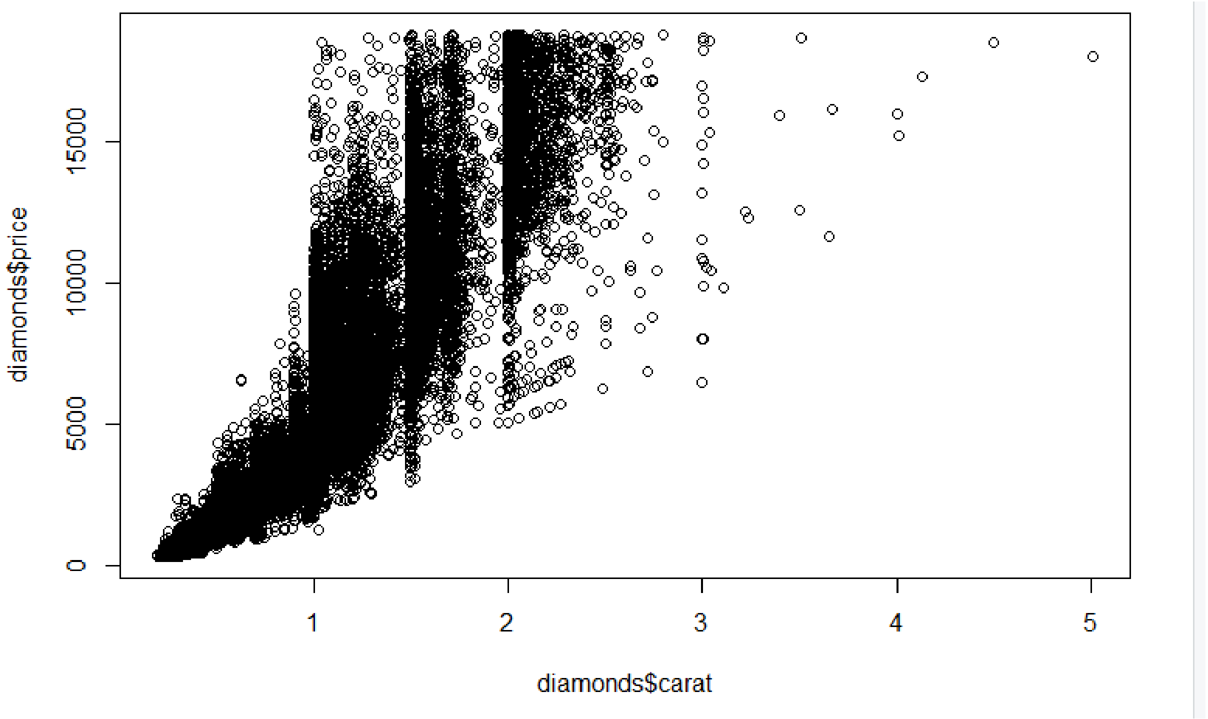
```
require(ggplot2)
Loading required package: ggplot2
> data(diamonds)
> head(diamonds)
# A tibble: 6 x 10
  carat cut       color clarity depth table price     x     y     z
  <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1 0.23 Ideal     E     SI2      61.5    55   326 3.95 3.98 2.43
2 0.21 Premium   E     SI1      59.8    61   326 3.89 3.84 2.31
3 0.23 Good      E     VS1      56.9    65   327 4.05 4.07 2.31
4 0.29 Premium   I     VS2      62.4    58   334 4.2   4.23 2.63
5 0.31 Good      J     SI2      63.3    58   335 4.34 4.35 2.75
6 0.24 Very Good J     VVS2     62.8    57   336 3.94 3.96 2.48
> hist (diamonds$carat, main="Carat Histogram", xlab= "Carat")
```



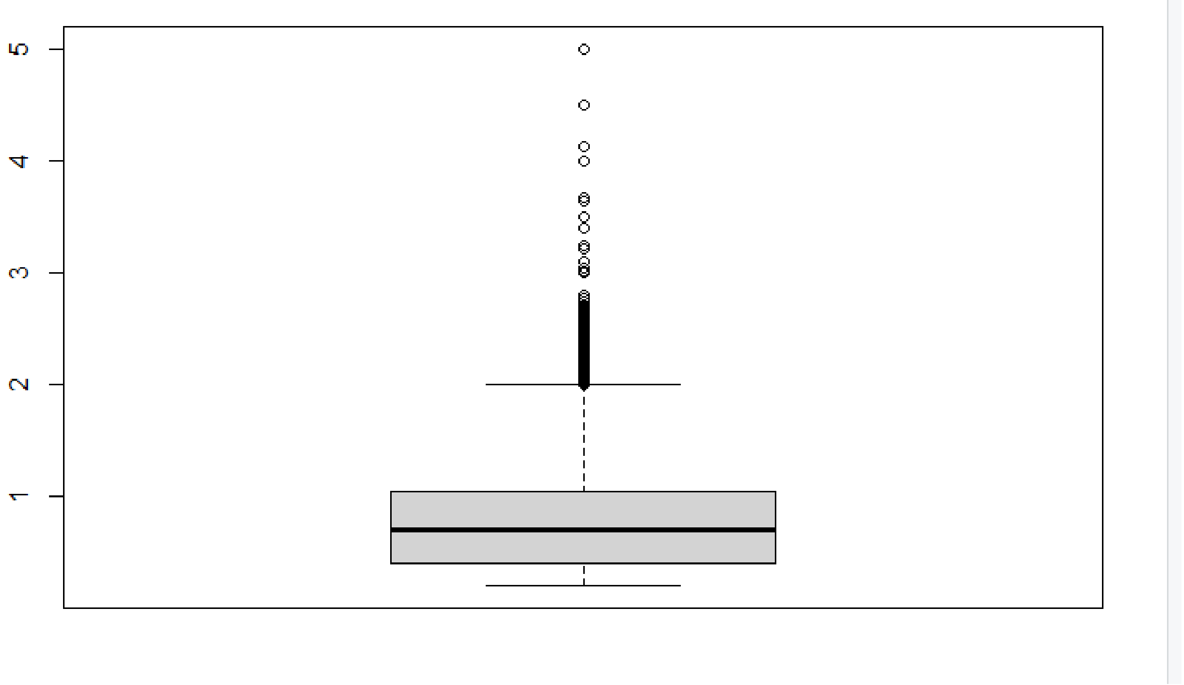
> plot (price ~ carat, data=diamonds)



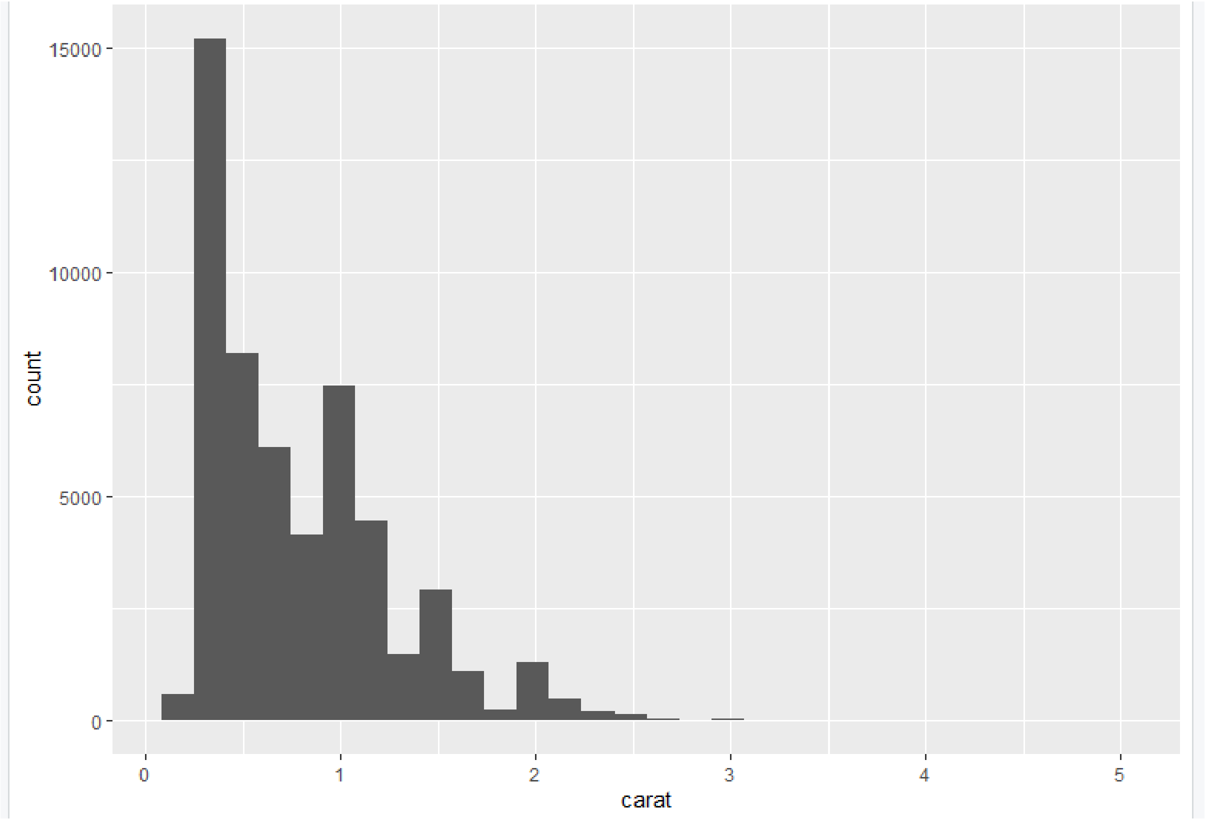
> plot (diamonds$carat, diamonds$price)



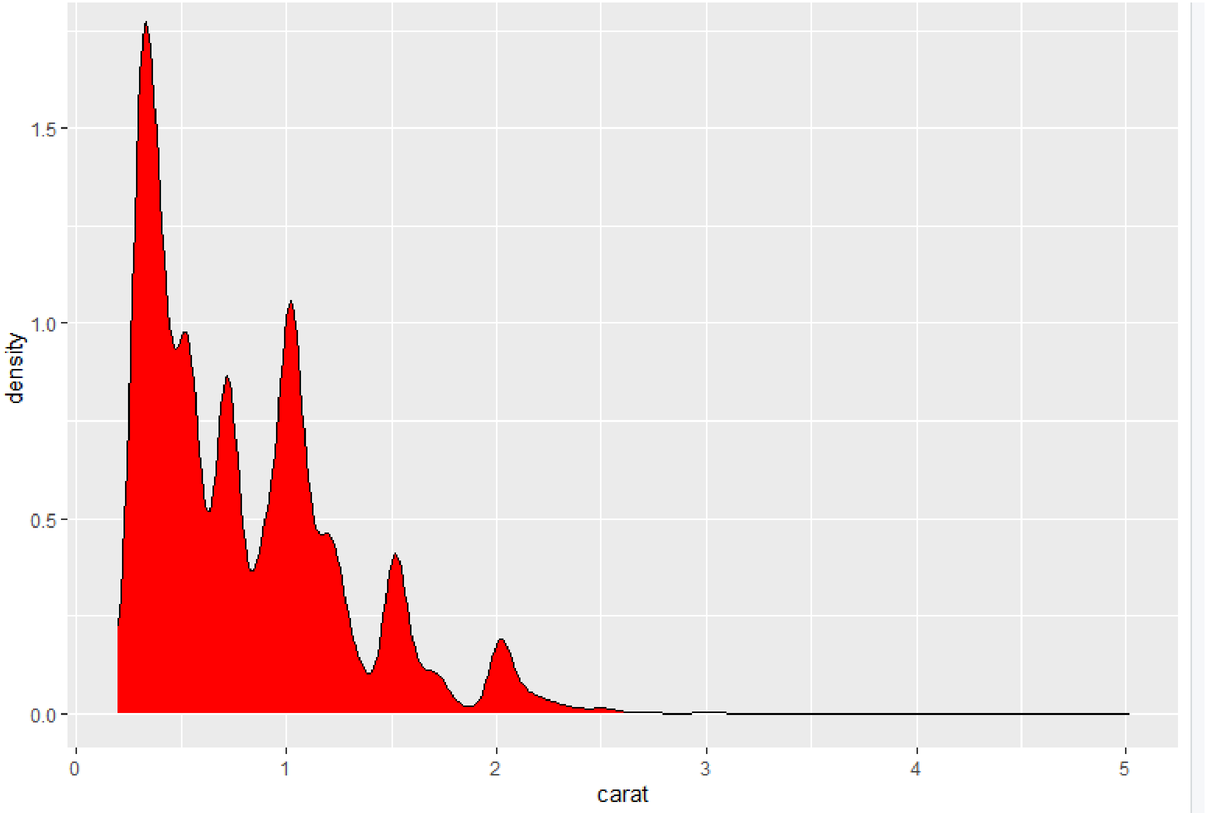
```
> boxplot(diamonds$carat)
```



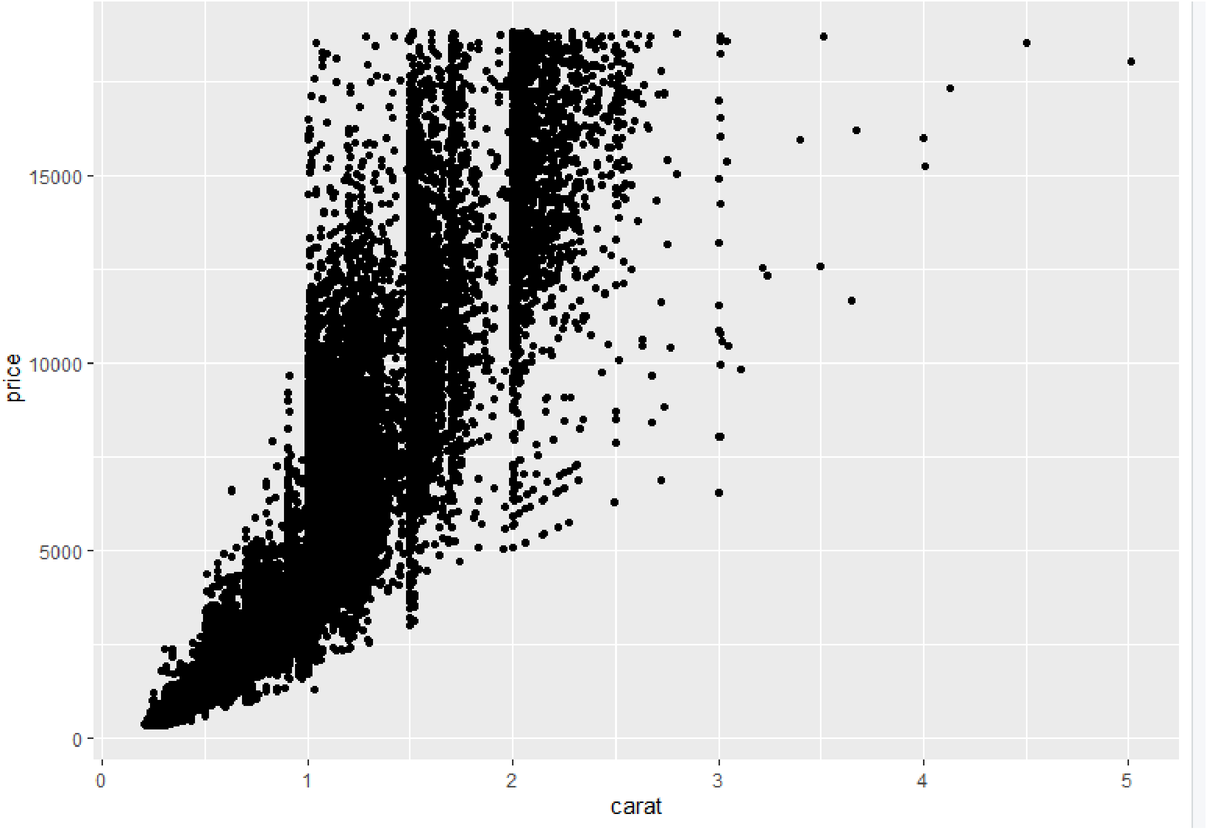
```
> ggplot(data=diamonds) + geom_histogram(aes(x=carat))
```



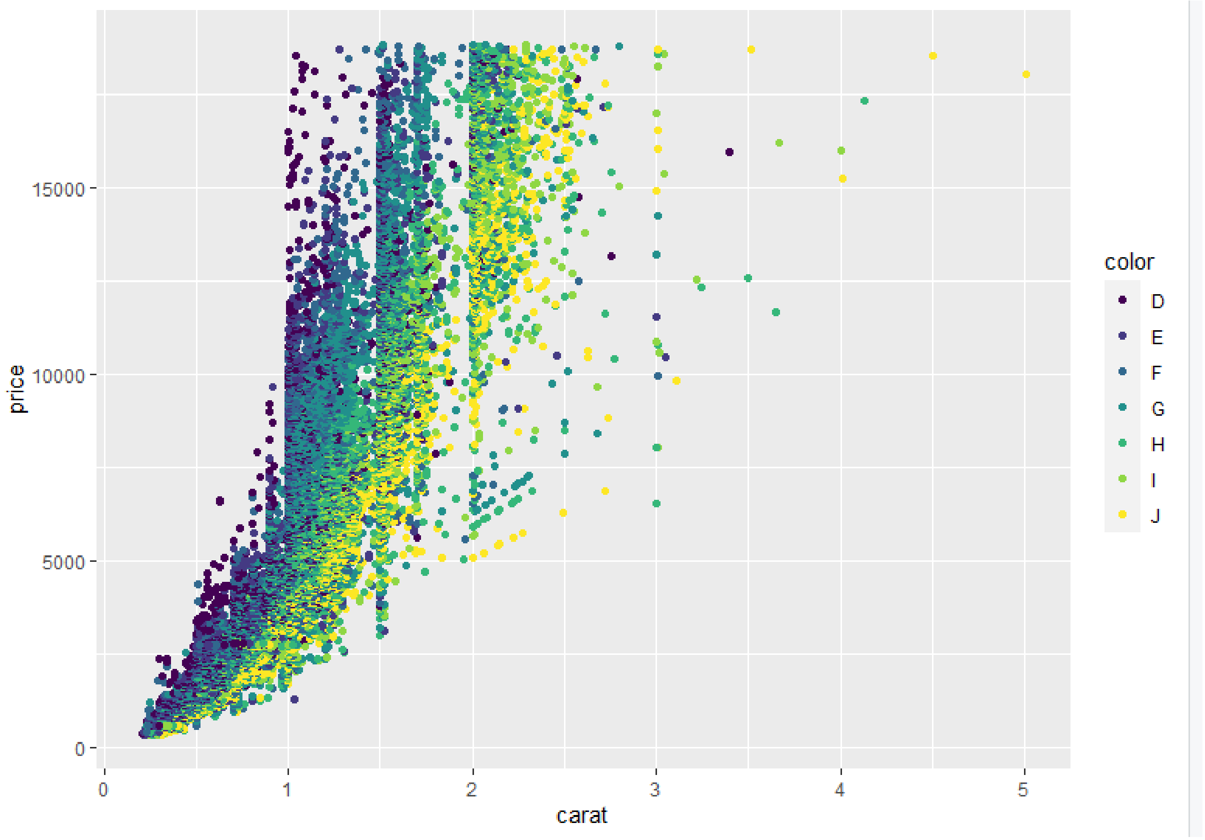
> ggplot(data=diamonds) + geom_density(aes(x=carat), fill= "red")



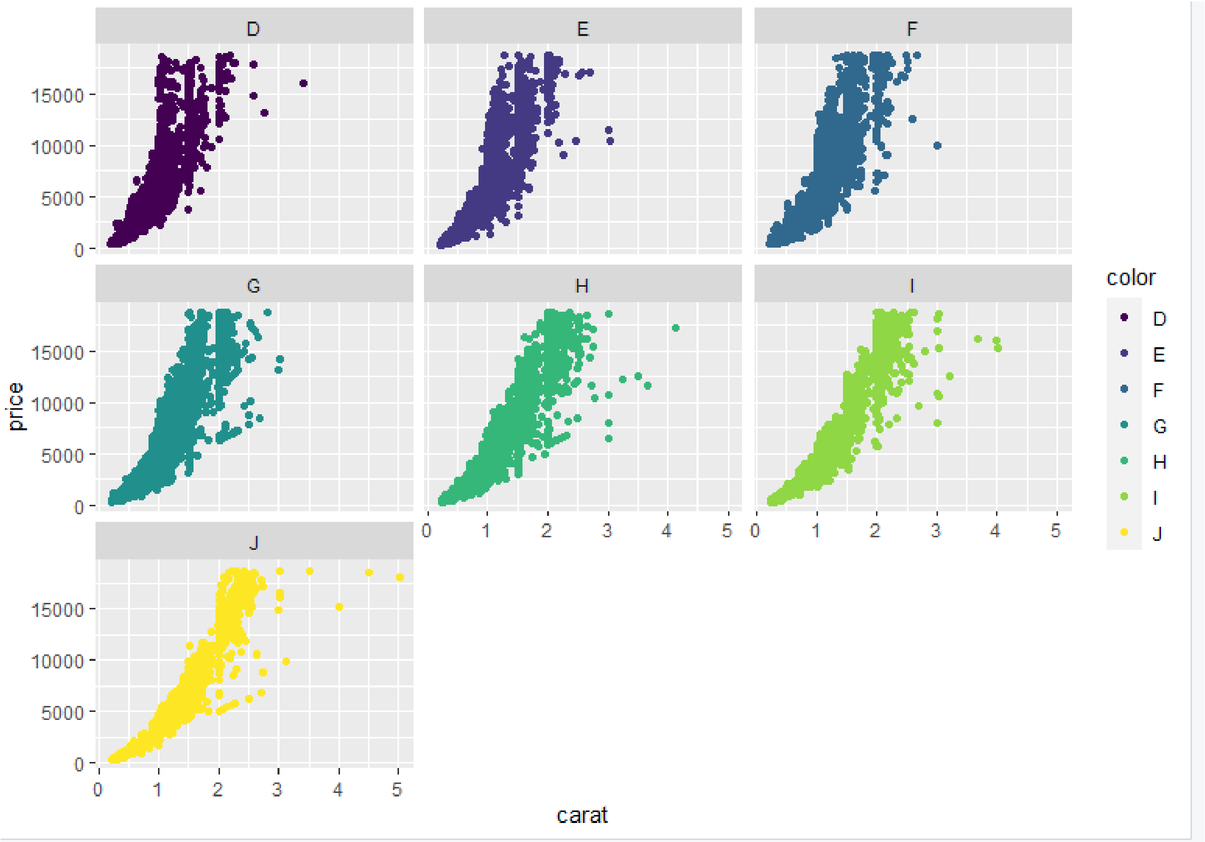
> ggplot (diamonds, aes (x=carat, y=price,)) +geom_point ()



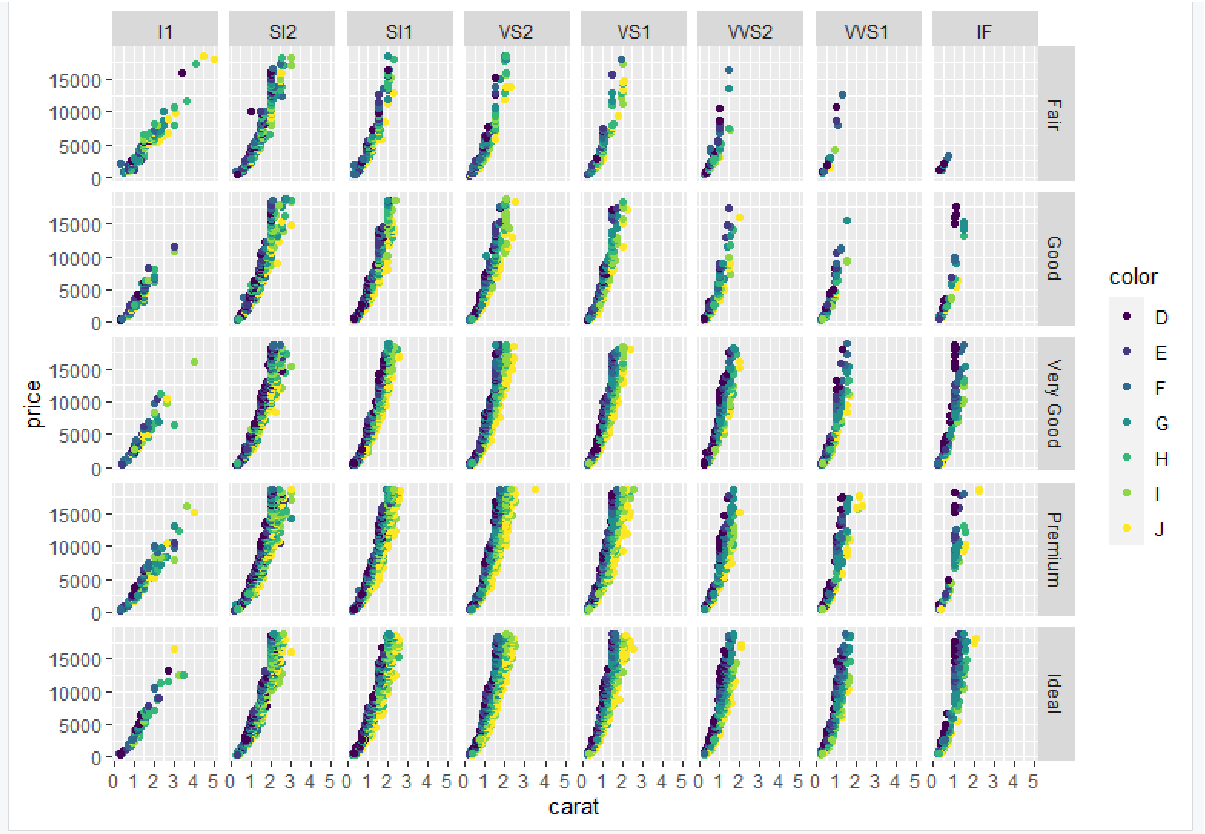
```
> g<- ggplot (diamonds, aes (x=carat, y=price))
> g+geom_point(aes(color=color))
```



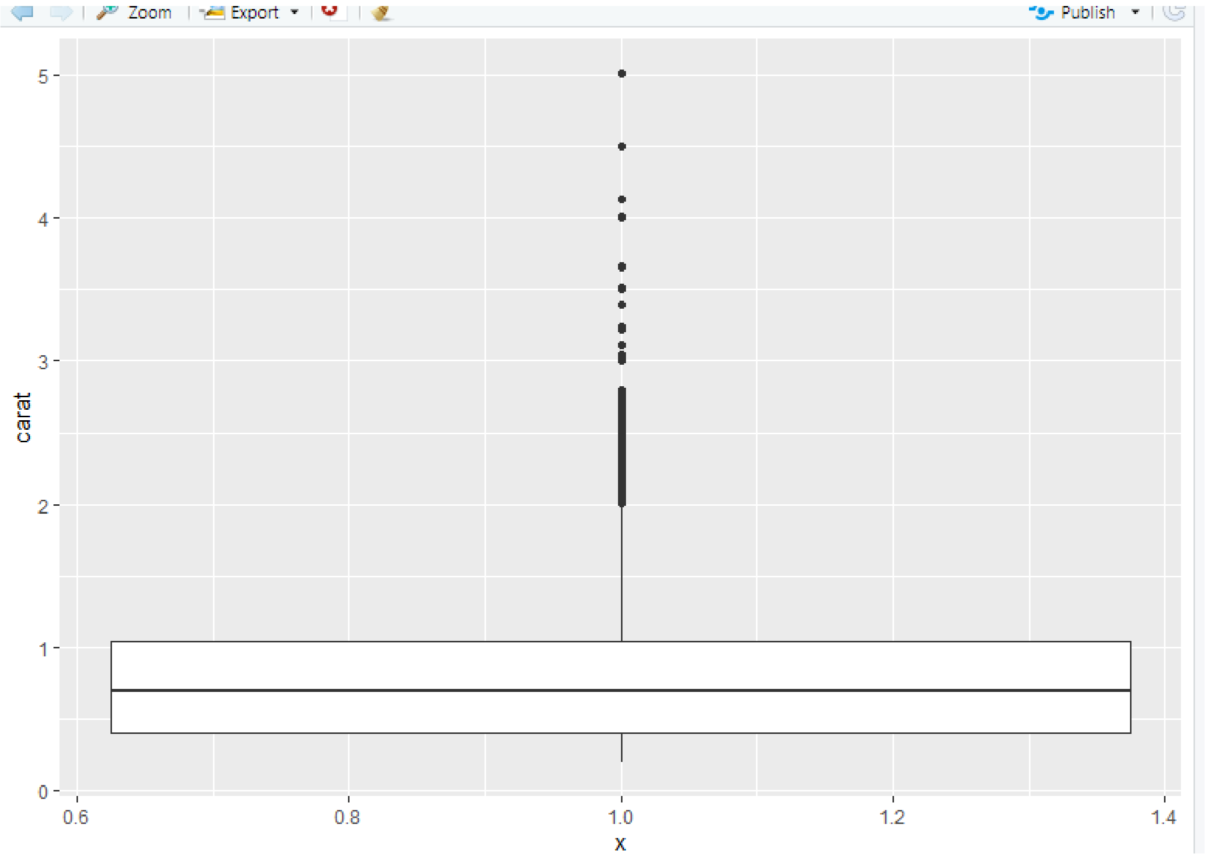
```
> g+geom_point(aes(color=color)) + facet_wrap (~ color)
```



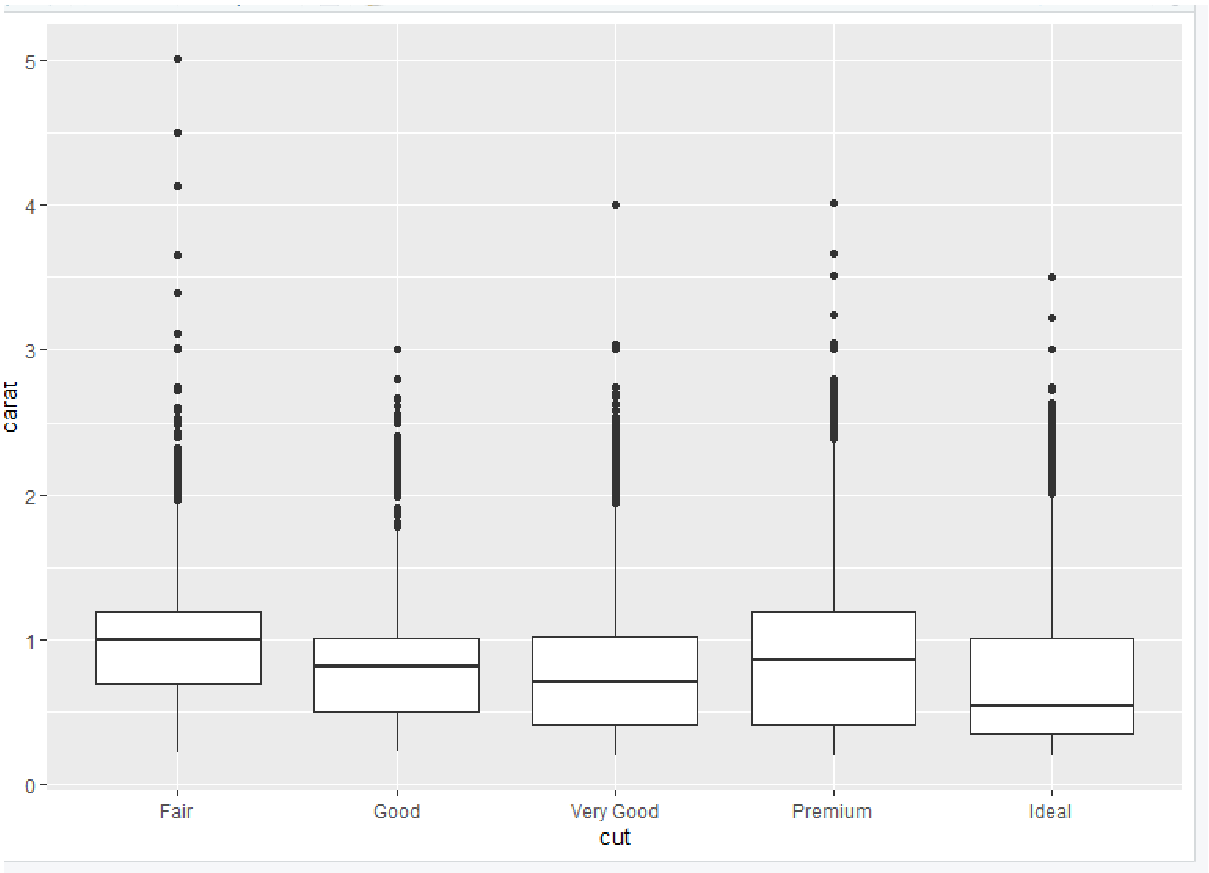
> g+geom_point(aes(color=color)) + facet_grid(cut~clarity)



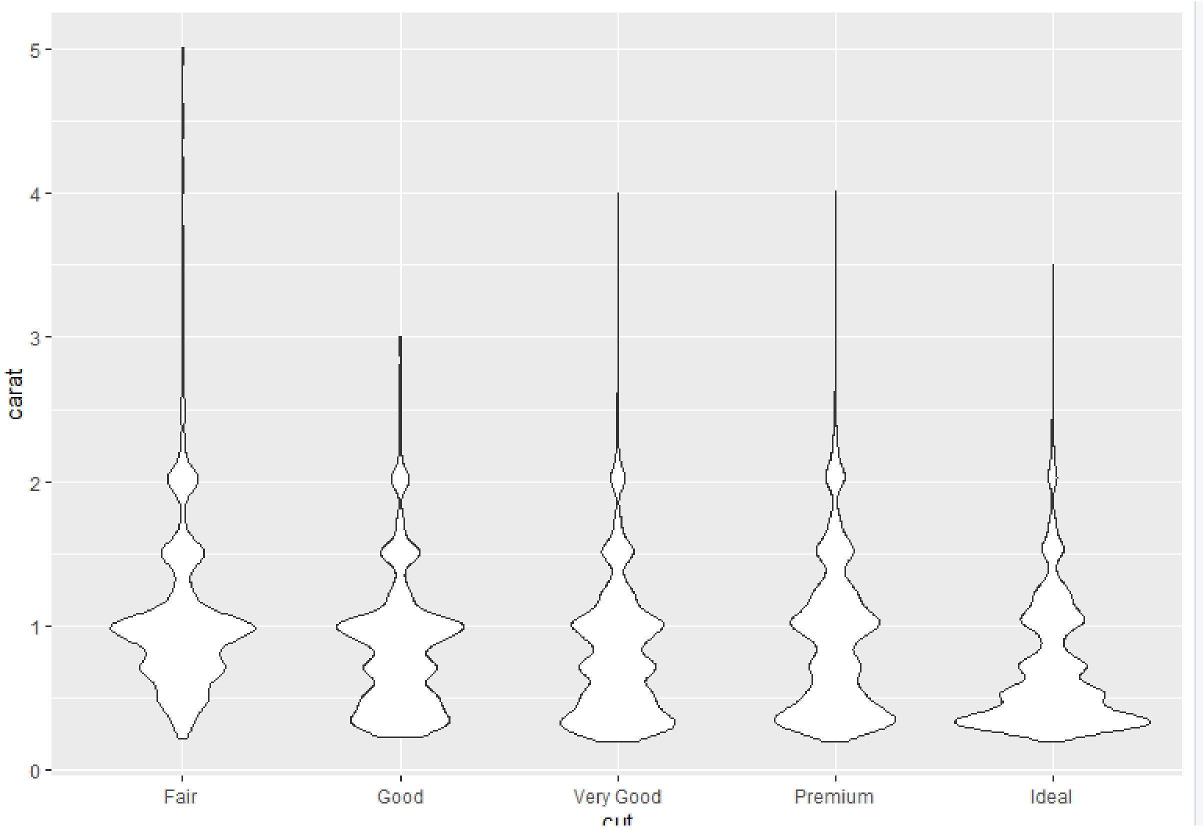
> ggplot (diamonds, aes (y=carat, x=1)) + geom_boxplot ()



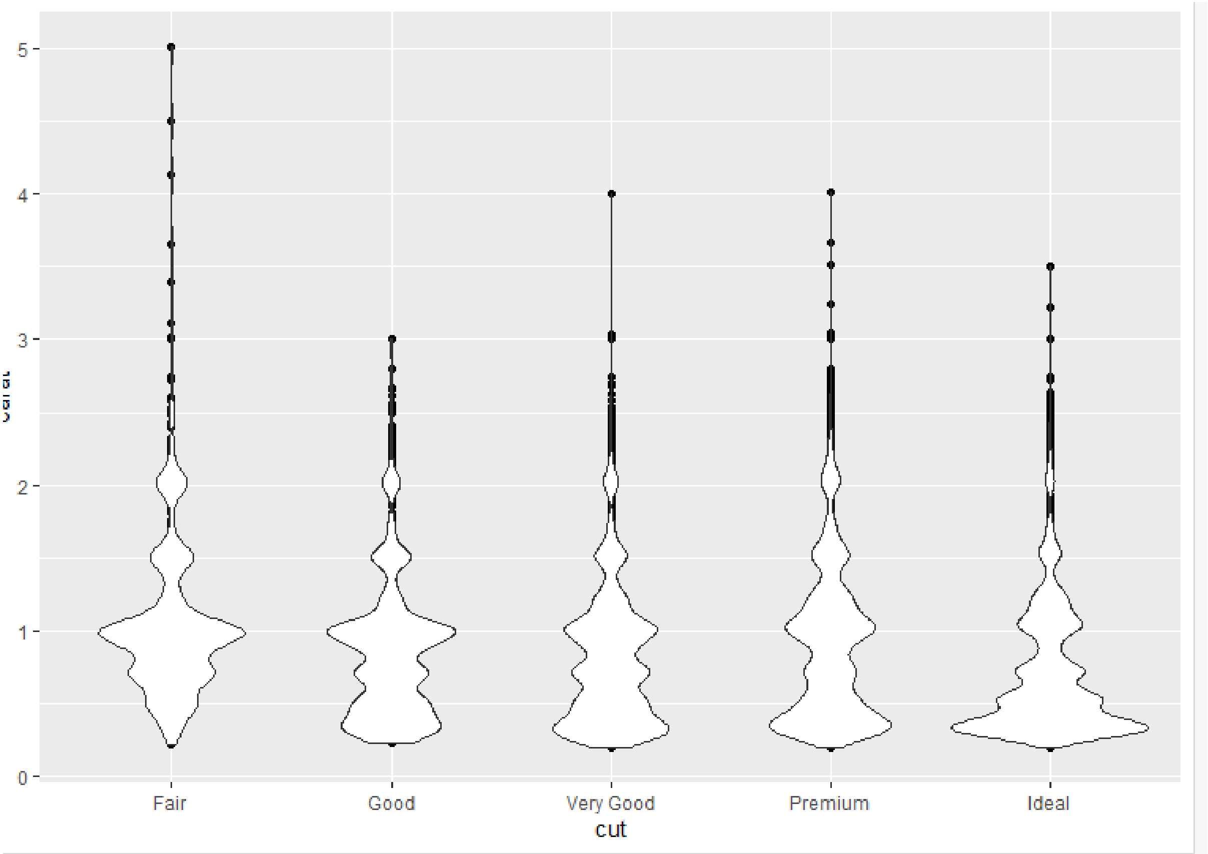
> ggplot (diamonds, aes (y=carat, x=cut)) + geom_boxplot ()



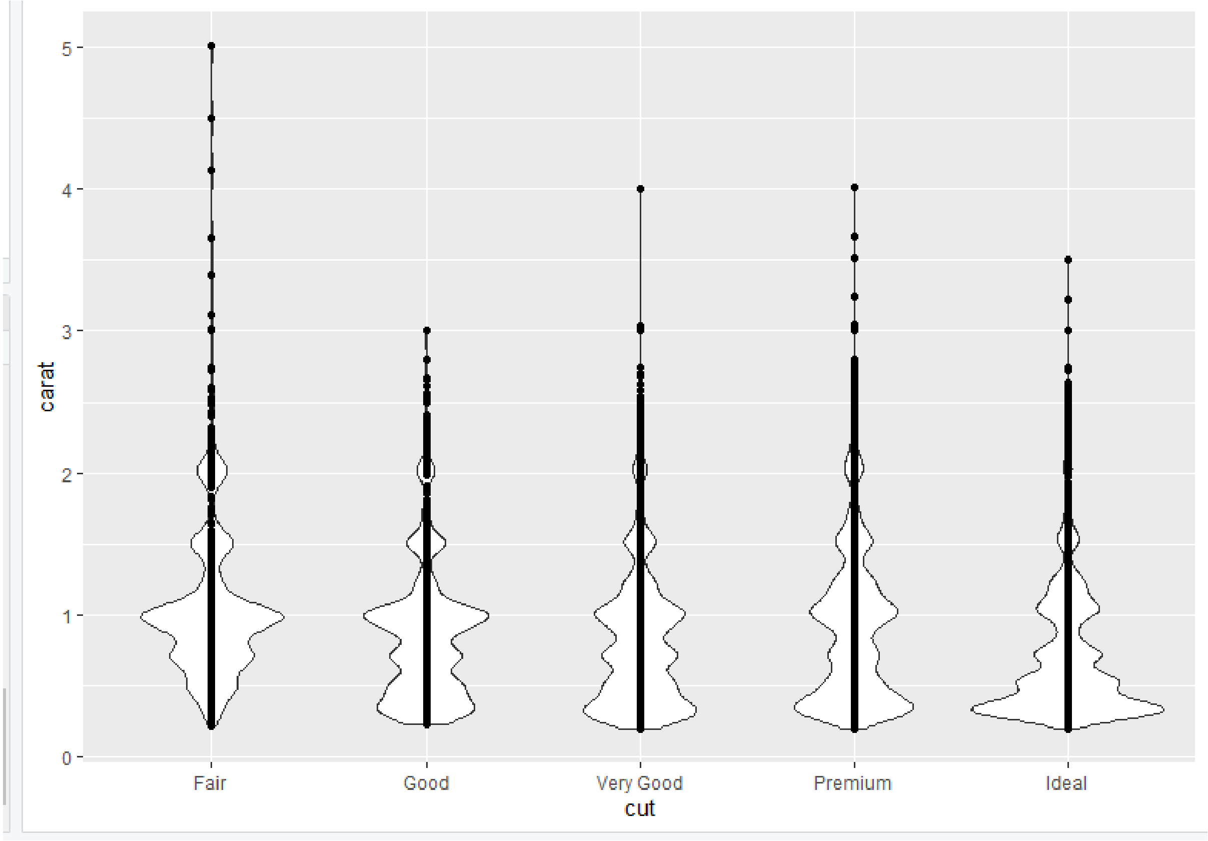
> ggplot (diamonds, aes (y=carat, x=cut)) + geom_violin ()



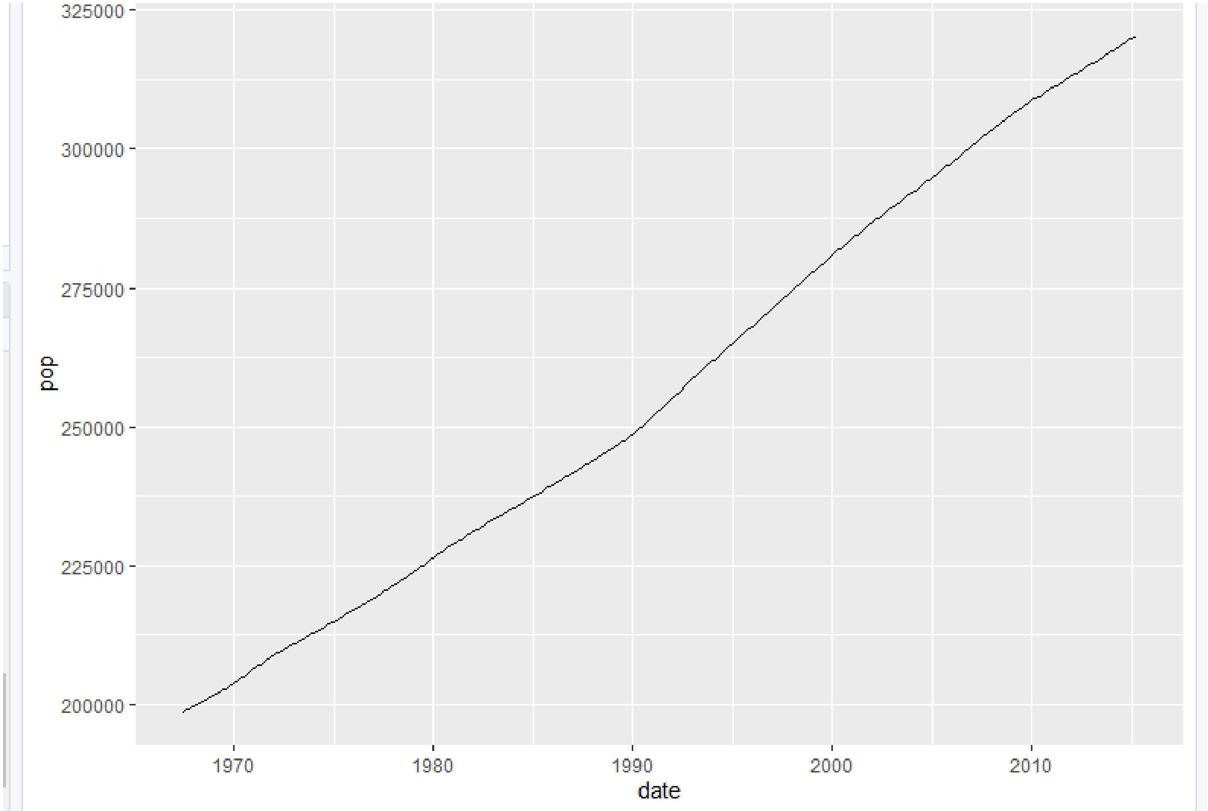
> ggplot (diamonds, aes (y=carat, x=cut)) + geom_point () + geom_violin ()



> ggplot (diamonds, aes (y=carat, x=cut)) + geom_violin () + geom_point ()



```
> ggplot (economics, aes (x=date, y= pop)) + geom_line ()
```

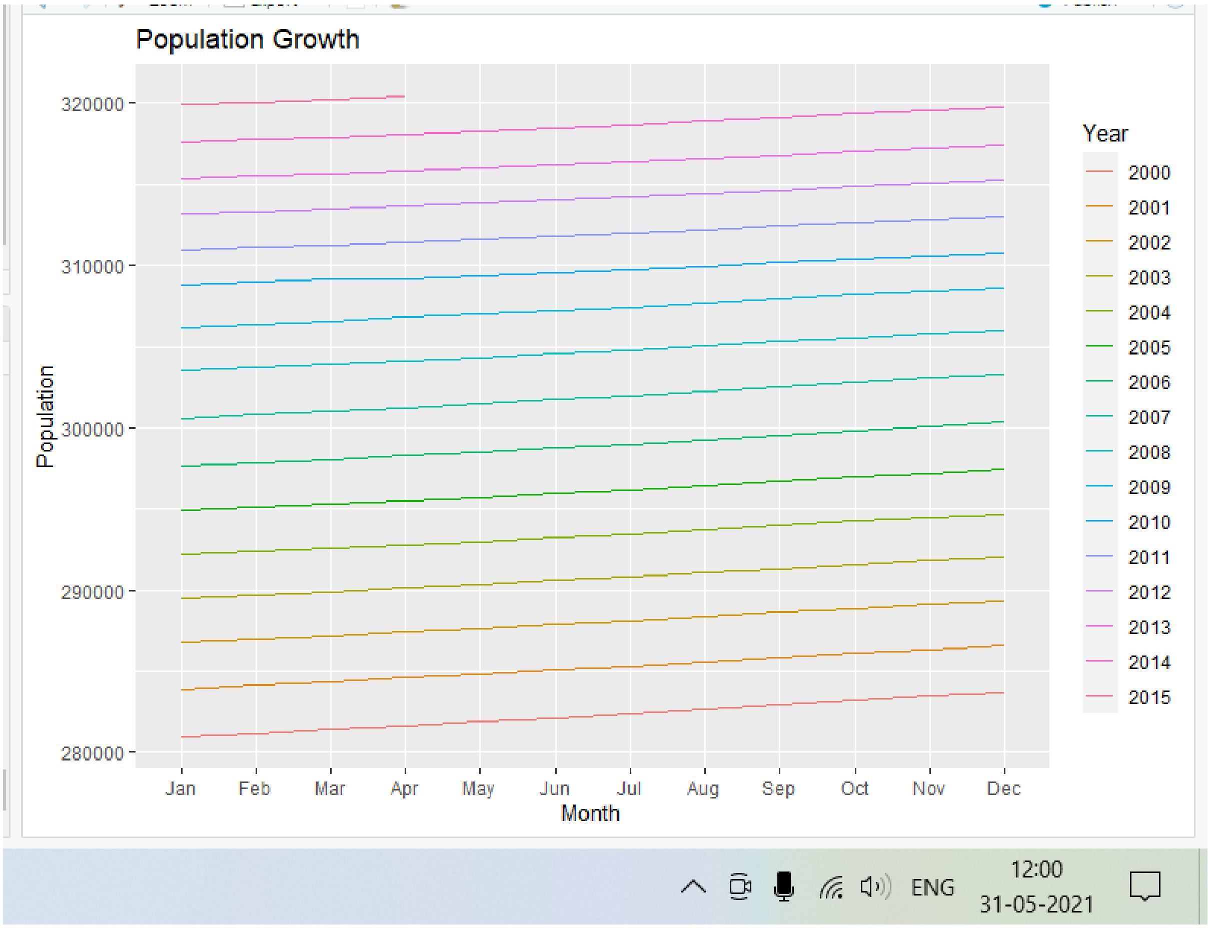


```
> require(lubridate)
> economics$year<- year(economics$date)
> economics$year
  [1] 1967 1967 1967 1967 1967 1967 1968 1968 1968 1968 1968 1968 1968 1968 1968 1968
1968 1968 1969 1969 1969 1969 1969 1969 1969
 [26] 1969 1969 1969 1969 1969 1970 1970 1970 1970 1970 1970 1970 1970 1970 1970 1970
1970 1971 1971 1971 1971 1971 1971 1971 1971
 [51] 1971 1971 1971 1971 1972 1972 1972 1972 1972 1972 1972 1972 1972 1972 1972 1972
1973 1973 1973 1973 1973 1973 1973 1973 1973
 [76] 1973 1973 1973 1974 1974 1974 1974 1974 1974 1974 1974 1974 1974 1974 1974 1975
1975 1975 1975 1975 1975 1975 1975 1975 1975
[101] 1975 1975 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1977 1977
1977 1977 1977 1977 1977 1977 1977 1977 1977
[126] 1977 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1979 1979 1979
1979 1979 1979 1979 1979 1979 1979 1979 1979
[151] 1980 1980 1980 1980 1980 1980 1980 1980 1980 1980 1980 1980 1981 1981 1981 1981
1981 1981 1981 1981 1981 1981 1981 1981 1982
[176] 1982 1982 1982 1982 1982 1982 1982 1982 1982 1982 1982 1983 1983 1983 1983 1983
1983 1983 1983 1983 1983 1983 1983 1984 1984
[201] 1984 1984 1984 1984 1984 1984 1984 1984 1984 1984 1985 1985 1985 1985 1985 1985
1985 1985 1985 1985 1985 1985 1986 1986 1986
[226] 1986 1986 1986 1986 1986 1986 1986 1986 1986 1987 1987 1987 1987 1987 1987 1987
1987 1987 1987 1987 1987 1988 1988 1988 1988
[251] 1988 1988 1988 1988 1988 1988 1988 1988 1989 1989 1989 1989 1989 1989 1989 1989
1989 1989 1989 1989 1990 1990 1990 1990 1990
[276] 1990 1990 1990 1990 1990 1990 1990 1991 1991 1991 1991 1991 1991 1991 1991 1991
1991 1991 1991 1992 1992 1992 1992 1992 1992
[301] 1992 1992 1992 1992 1992 1992 1993 1993 1993 1993 1993 1993 1993 1993 1993 1993
1993 1993 1994 1994 1994 1994 1994 1994 1994
[326] 1994 1994 1994 1994 1994 1995 1995 1995 1995 1995 1995 1995 1995 1995 1995 1995
1995 1996 1996 1996 1996 1996 1996 1996 1996
[351] 1996 1996 1996 1996 1997 1997 1997 1997 1997 1997 1997 1997 1997 1997 1997 1997
1998 1998 1998 1998 1998 1998 1998 1998 1998
[376] 1998 1998 1998 1999 1999 1999 1999 1999 1999 1999 1999 1999 1999 1999 1999 2000
2000 2000 2000 2000 2000 2000 2000 2000 2000
```

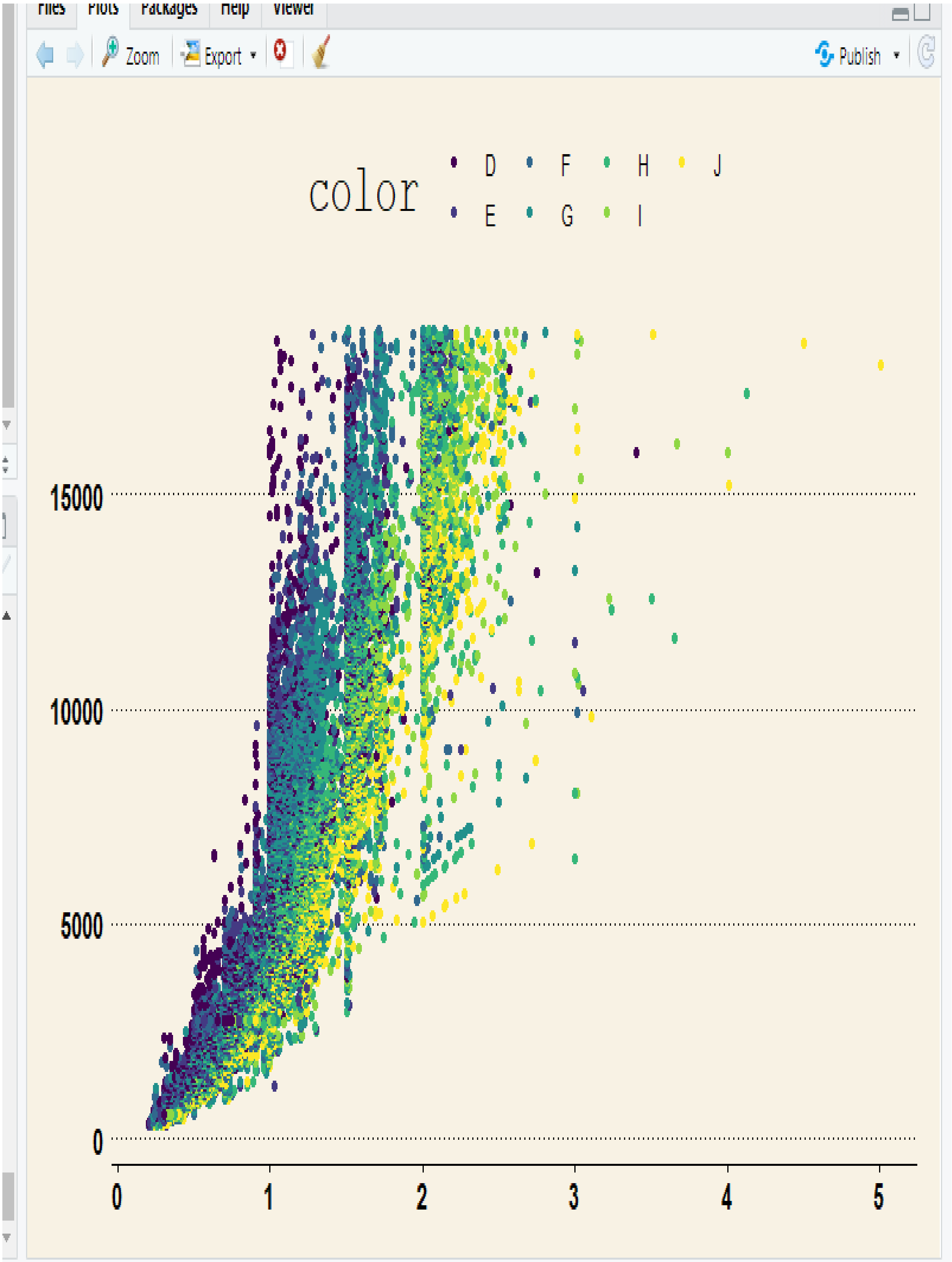
```
[401] 2000 2000 2001 2001 2001 2001 2001 2001 2001 2001 2001 2001 2001 2001 2002 2002
2002 2002 2002 2002 2002 2002 2002 2002 2002
[426] 2002 2003 2003 2003 2003 2003 2003 2003 2003 2003 2003 2003 2003 2004 2004 2004
2004 2004 2004 2004 2004 2004 2004 2004 2004
[451] 2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 2006 2006 2006 2006
2006 2006 2006 2006 2006 2006 2006 2006 2007
[476] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 2008 2008 2008 2008 2008
2008 2008 2008 2008 2008 2008 2008 2009 2009
[501] 2009 2009 2009 2009 2009 2009 2009 2009 2009 2009 2010 2010 2010 2010 2010 2010
2010 2010 2010 2010 2010 2010 2011 2011 2011
[526] 2011 2011 2011 2011 2011 2011 2011 2011 2011 2012 2012 2012 2012 2012 2012 2012
2012 2012 2012 2012 2012 2013 2013 2013 2013
[551] 2013 2013 2013 2013 2013 2013 2013 2013 2014 2014 2014 2014 2014 2014 2014 2014
2014 2014 2014 2014 2015 2015 2015 2015
> economics$month<- month (economics$date, label=TRUE)
> economics$month
  [1] Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
 [33] Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
 [65] Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
 [97] Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
[129] Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
[161] Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May
Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
[193] Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
[225] Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
[257] Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May
Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
```

```
[289] Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
[321] Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
[353] Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May
Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
[385] Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
[417] Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
[449] Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May
Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
[481] Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb
[513] Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct
[545] Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May
Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr
Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < Sep < Oct < Nov < Dec
> econ2000<-economics [which (economics$year >=2000),]
> econ2000
# A tibble: 184 x 8
   date         pce    pop psavert uempmed unemploy year month
   <date>     <dbl> <dbl>   <dbl>   <dbl>    <dbl> <dbl> <ord>
 1 2000-01-01 6535. 280976     5.4     5.8     5708 2000 Jan
 2 2000-02-01 6620. 281190     4.8     6.1     5858 2000 Feb
 3 2000-03-01 6686. 281409     4.5     6       5733 2000 Mar
 4 2000-04-01 6671. 281653     5       6.1     5481 2000 Apr
 5 2000-05-01 6708. 281877     4.9     5.8     5758 2000 May
 6 2000-06-01 6744. 282126     4.9     5.7     5651 2000 Jun
 7 2000-07-01 6764. 282385     5.2     6       5747 2000 Jul
 8 2000-08-01 6799. 282653     5.2     6.3     5853 2000 Aug
 9 2000-09-01 6883. 282932     4.5     5.2     5625 2000 Sep
10 2000-10-01 6888. 283201     4.6     6.1     5534 2000 Oct
```

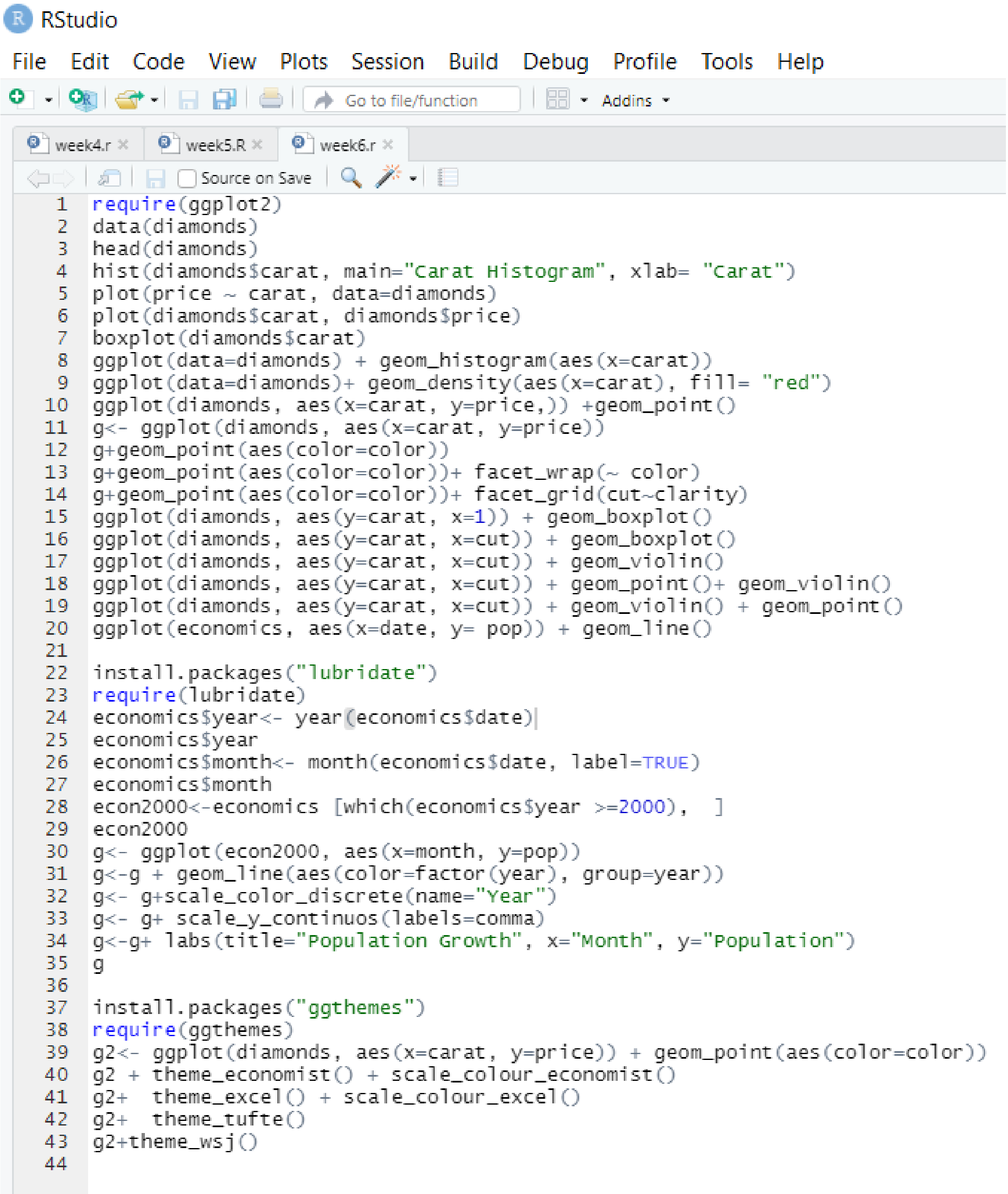
```
# ... with 174 more rows
> g<- ggplot (econ2000, aes (x=month, y=pop))
> g<-g + geom_line(aes(color=factor(year), group=year))
> g<- g+scale_color_discrete(name="Year")
> g<- g+ scale_y_continuos(labels=comma)
> g<-g+ labs (title="Population Growth", x="Month", y="Population")
> g
```



```
> install. packages("ggthemes")
> g2<- ggplot (diamonds, aes (x=carat, y=price)) + geom_point(aes(color=color))
> g2<- ggplot (diamonds, aes (x=carat, y=price)) + geom_point(aes(color=color))
> g2 + theme_economist () + scale_colour_economist ()
> g2+ theme_excel () + scale_colour_excel ()
> g2+ theme_tufte ()
> g2+theme_wsj ()
```


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
color
D
E
F
G
H
I
J
15000
10000
5000
0
0
1
2
3
4
5

EDITOR CODE:



```
require(ggplot2)
data(diamonds)
head(diamonds)
hist(diamonds$carat, main="Carat Histogram", xlab= "Carat")
plot(price ~ carat, data=diamonds)
plot(diamonds$carat, diamonds$price)
boxplot(diamonds$carat)
ggplot(data=diamonds) + geom_histogram(aes(x=carat))
ggplot(data=diamonds)+ geom_density(aes(x=carat), fill= "red")
ggplot(diamonds, aes(x=carat, y=price,)) +geom_point()
g<- ggplot(diamonds, aes(x=carat, y=price))
g+geom_point(aes(color=color))
g+geom_point(aes(color=color))+ facet_wrap(~ color)
g+geom_point(aes(color=color))+ facet_grid(cut~clarity)
ggplot(diamonds, aes(y=carat, x=1)) + geom_boxplot()
ggplot(diamonds, aes(y=carat, x=cut)) + geom_boxplot()
ggplot(diamonds, aes(y=carat, x=cut)) + geom_violin()
ggplot(diamonds, aes(y=carat, x=cut)) + geom_point()+ geom_violin()
ggplot(diamonds, aes(y=carat, x=cut)) + geom_violin() + geom_point()
ggplot(economics, aes(x=date, y= pop)) + geom_line()

install.packages("lubridate")
require(lubridate)
economics$year<- year(economics$date)
economics$year
economics$month<- month(economics$date, label=TRUE)
economics$month
econ2000<-economics [which(economics$year >=2000),  ]
econ2000
g<- ggplot(econ2000, aes(x=month, y=pop))
g<-g + geom_line(aes(color=factor(year), group=year))
g<- g+scale_color_discrete(name="Year")
g<- g+ scale_y_continuos(labels=comma)
g<-g+ labs(title="Population Growth", x="Month", y="Population")
g

install.packages("ggthemes")
require(ggthemes)
g2<- ggplot(diamonds, aes(x=carat, y=price)) + geom_point(aes(color=color))
g2 + theme_economist() + scale_colour_economist()
g2+  theme_excel() + scale_colour_excel()
g2+  theme_tufte()
g2+theme_wsj()

```

RESULT:

HENCE THE CONCEPT OF STATISTICAL GRAPHS ARE IMPLEMENTED.

R LAB-07

TASK-01:

Implement data reshaping concepts in r studio

PROGRAM:

C-BIND AND R-BIND:

```
sport<- c ("Hockey", "Baseball”, “Football")
league<- c ("NHL", "MLB", "NFL")
trophy<- c ("Stanley Cup","Commissioner’s Trophy", "Vince Lombardi Trophy")
trophies1<- cbind (sport, league, trophy)
trophies1
trophies2<- data. frame (sport=c ("Basketball", "Golf"), league=c ("NBA", "PGA"), trophy=c
("Larry O’Brien Championship Trophy","Wanamaker Trophy"), stringsAsFactors=FALSE);
trophies2
trophies<-rbind (trophies1, trophies2)
trophies
cbind (Sport=sport, Association=league, Prize= trophy)
```

JOINTS :

```
df1 = data. frame (StudentId = c (101:106), Product = c ("Hindi", "English","Maths",
"Science","Political Science","Physics"))
df1
df2 = data. frame (StudentId = c (102, 104, 106, 107, 108), State = c ("Mangalore",
"Mysore","Pune", "Dehradun", "Delhi"))
df2
df = merge (x = df1, y = df2, by = "StudentId")
df
install. packages ("plyr")
require(plyr)
eid<-1001:1005
ename=c("SMITH","ALLEN","MARTIN","SCOT","LUTHER")
designation<-c
("PRESIDENT","VICEPRESIDENT","SALES","ACCOUNTANT","SALES")
salary<-c (19000,45000,55000,60000,45000)
```

```
did=c (10,20,20,30,50)
emp<-data. frame (eid, ename, salary, designation, did)
emp
did<-c (10,20,30,40,50)
dname<-c("IT","CSE","ECE","EIE","CIVIL")
dept<-data. frame (did, dname)
dept
college<-merge (x=emp, y=dept, by="did")
college
college1=merge (x=emp, y=dept, by="did”, all.x ="True")
college1
college2=merge (x=emp, y=dept, by="did”, all. y = "True")
college2
college3=merge (x=emp, y=dept, by="did”, all="True")
college3
college4=merge (x=emp, y=dept, by=NULL)
college4
```

OUTPUT:

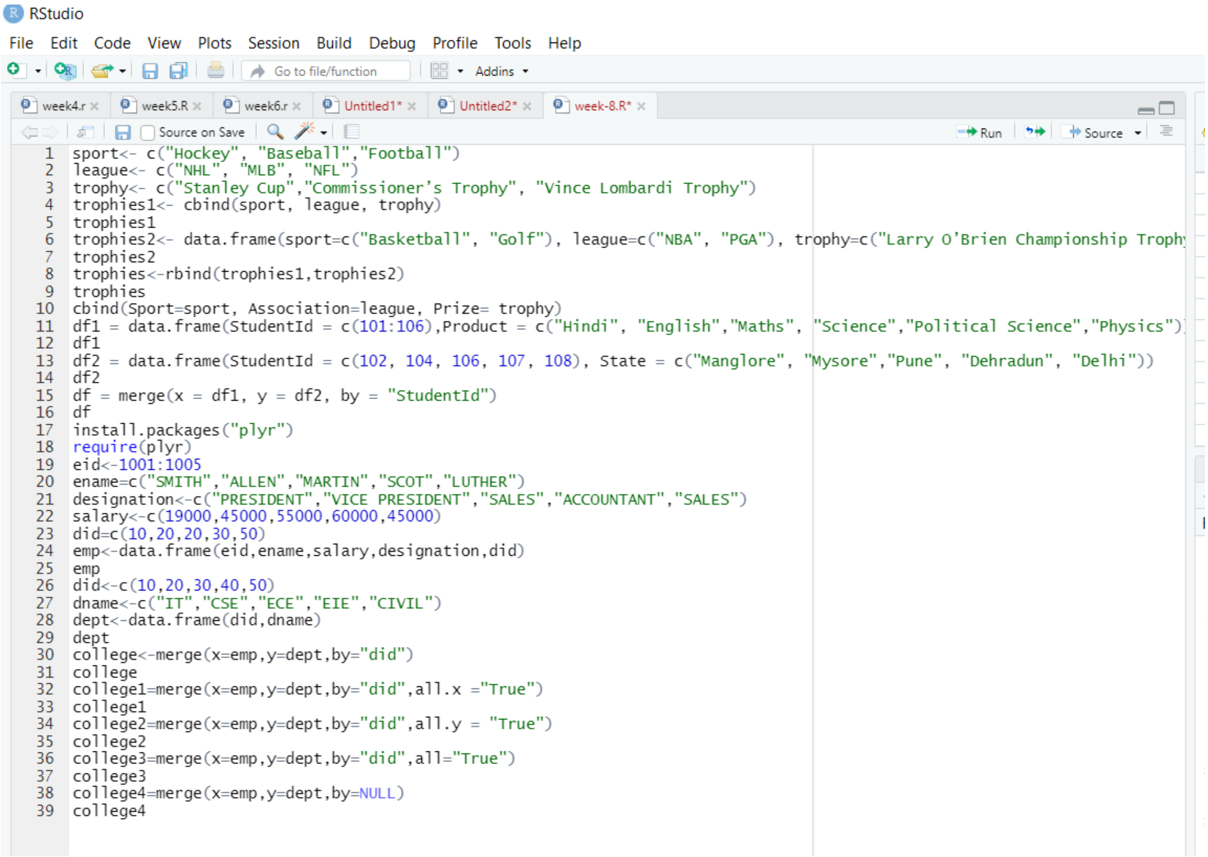
```
sport<- c ("Hockey", "Baseball","Football")
> league<- c ("NHL", "MLB", "NFL")
> trophy<- c ("Stanley Cup","Commissioner’s Trophy", "Vince Lombardi Trophy")
> trophies1<- cbind (sport, league, trophy)
> trophies1
    sport     league    trophy
[1,] "Hockey"   "NHL"  "Stanley Cup"
[2,] "Baseball" "MLB"  "Commissioner’s Trophy"
[3,] "Football" "NFL"  "Vince Lombardi Trophy"
> trophies2<- data. frame (sport=c ("Basketball", "Golf"), league=c ("NBA", "PGA"),
trophy=c ("Larry O’Brien Championship Trophy","Wanamaker Trophy"),
stringsAsFactors=FALSE);
> trophies2
      sport league                          trophy
1 Basketball    NBA Larry O’Brien Championship Trophy
```

```
2       Golf    PGA                 Wanamaker Trophy
> trophies<-rbind (trophies1, trophies2)
> trophies
      sport league                         trophy
1     Hockey    NHL                       Stanley Cup
2   Baseball    MLB            Commissioner’s Trophy
3   Football    NFL            Vince Lombardi Trophy
4 Basketball    NBA Larry O’Brien Championship Trophy
5       Golf    PGA                 Wanamaker Trophy
> cbind (Sport=sport, Association=league, Prize= trophy)
     Sport      Association Prize
[1,] "Hockey"   "NHL"       "Stanley Cup"
[2,] "Baseball" "MLB"       "Commissioner’s Trophy"
[3,] "Football" "NFL"       "Vince Lombardi Trophy"
> df1 = data. frame (StudentId = c (101:106), Product = c ("Hindi", "English","Maths",
"Science","Political Science","Physics"))
> df1
  StudentId           Product
1       101             Hindi
2       102           English
3       103             Maths
4       104           Science
5       105 Political Science
6       106           Physics
> df2 = data. frame (StudentId = c(102, 104, 106, 107, 108), State = c("Mangalore",
"Mysore","Pune", "Dehradun", "Delhi"))
> df2
  StudentId    State
1       102 Mangalore
2       104   Mysore
3       106     Pune
4       107 Dehradun
5       108    Delhi
> df = merge (x = df1, y = df2, by = "StudentId")
> df
  StudentId Product    State
1       102 English Mangalore
2       104 Science   Mysore
3       106 Physics     Pune
> install. packages ("ply")
> require(plyr)
Loading required package: plyr
> eid<-1001:1005
> ename=c("SMITH","ALLEN","MARTIN","SCOT","LUTHER")
>designation<-c
("PRESIDENT","VICEPRESIDENT","SALES","ACCOUNTANT","SALES")
> salary<-c (19000,45000,55000,60000,45000)
> did=c (10,20,20,30,50)
> emp<-data. frame (eid, ename, salary, designation, did)
> emp
   eid ename salary    designation did
1 1001 SMITH  19000      PRESIDENT  10
2 1002 ALLEN  45000 VICE PRESIDENT  20
3 1003 MARTIN  55000          SALES  20
4 1004   SCOT  60000     ACCOUNTANT  30
5 1005 LUTHER  45000          SALES  50
> did<-c (10,20,30,40,50)
> dname<-c("IT","CSE","ECE","EIE","CIVIL")
> dept<-data. frame (did, dname)
> dept
  did dname
1 10    IT
2 20   CSE
3 30   ECE
4 40   EIE
5 50 CIVIL
> college<-merge (x=emp, y=dept, by="did")
> college
```

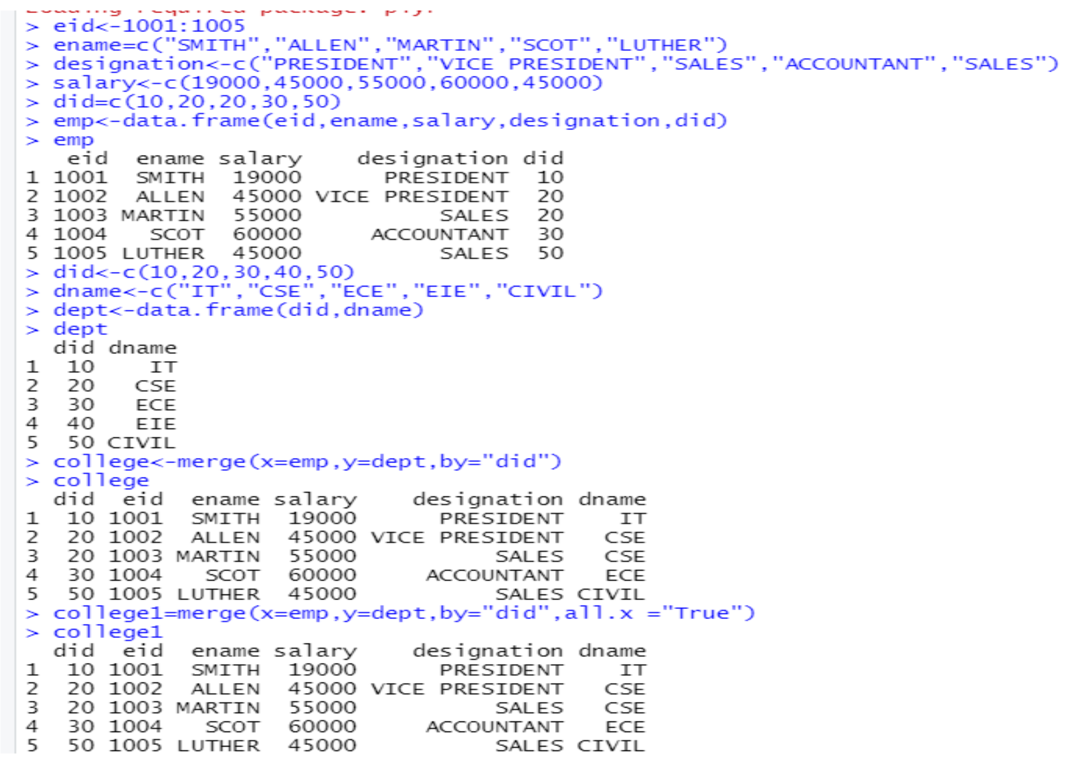
```
  did eid ename salary    designation dname
1 10 1001 SMITH  19000      PRESIDENT    IT
2 20 1002 ALLEN  45000 VICE PRESIDENT   CSE
3 20 1003 MARTIN  55000          SALES   CSE
4 30 1004   SCOT  60000     ACCOUNTANT   ECE
5 50 1005 LUTHER  45000          SALES CIVIL
> college1=merge (x=emp, y=dept, by="did”, all.x ="True")
> college1
  did eid ename salary    designation dname
1 10 1001 SMITH  19000      PRESIDENT    IT
2 20 1002 ALLEN  45000 VICE PRESIDENT   CSE
3 20 1003 MARTIN  55000          SALES   CSE
4 30 1004   SCOT  60000     ACCOUNTANT   ECE
5 50 1005 LUTHER  45000          SALES CIVIL
> college2=merge (x=emp, y=dept, by="did”, all. y = "True")
> college2
  did eid ename salary    designation dname
1 10 1001 SMITH  19000      PRESIDENT    IT
2 20 1002 ALLEN  45000 VICE PRESIDENT   CSE
3 20 1003 MARTIN  55000          SALES   CSE
4 30 1004   SCOT  60000     ACCOUNTANT   ECE
5 40   NA   <NA>     NA           <NA>   EIE
6 50 1005 LUTHER  45000          SALES CIVIL
> college3=merge (x=emp, y=dept, by="did”, all="True")
> college3
  did eid ename salary    designation dname
1 10 1001 SMITH  19000      PRESIDENT    IT
2 20 1002 ALLEN  45000 VICE PRESIDENT   CSE
3 20 1003 MARTIN  55000          SALES   CSE
4 30 1004   SCOT  60000     ACCOUNTANT   ECE
5 40   NA   <NA>     NA           <NA>   EIE
6 50 1005 LUTHER  45000          SALES CIVIL
> college4=merge (x=emp, y=dept, by=NULL)
> college4
```

```
   eid ename salary    designation did.x did. y dname
1 1001 SMITH  19000      PRESIDENT    10    10    IT
2 1002 ALLEN  45000 VICE PRESIDENT    20    10    IT
3 1003 MARTIN  55000          SALES    20    10    IT
4 1004   SCOT  60000     ACCOUNTANT    30    10    IT
5 1005 LUTHER  45000          SALES    50    10    IT
6 1001 SMITH  19000      PRESIDENT    10    20   CSE
7 1002 ALLEN  45000 VICE PRESIDENT    20    20   CSE
8 1003 MARTIN  55000          SALES    20    20   CSE
9 1004   SCOT  60000     ACCOUNTANT    30    20   CSE
10 1005 LUTHER  45000          SALES    50    20   CSE
11 1001 SMITH  19000      PRESIDENT    10    30   ECE
12 1002 ALLEN  45000 VICE PRESIDENT    20    30   ECE
13 1003 MARTIN  55000          SALES    20    30   ECE
14 1004   SCOT  60000     ACCOUNTANT    30    30   ECE
15 1005 LUTHER  45000          SALES    50    30   ECE
16 1001 SMITH  19000      PRESIDENT    10    40   EIE
17 1002 ALLEN  45000 VICE PRESIDENT    20    40   EIE
18 1003 MARTIN  55000          SALES    20    40   EIE
19 1004   SCOT  60000     ACCOUNTANT    30    40   EIE
20 1005 LUTHER  45000          SALES    50    40   EIE
21 1001 SMITH  19000      PRESIDENT    10    50 CIVIL
22 1002 ALLEN  45000 VICE PRESIDENT    20    50 CIVIL
23 1003 MARTIN  55000          SALES    20    50 CIVIL
24 1004   SCOT  60000     ACCOUNTANT    30    50 CIVIL
25 1005 LUTHER  45000          SALES    50    50 CIVIL
```

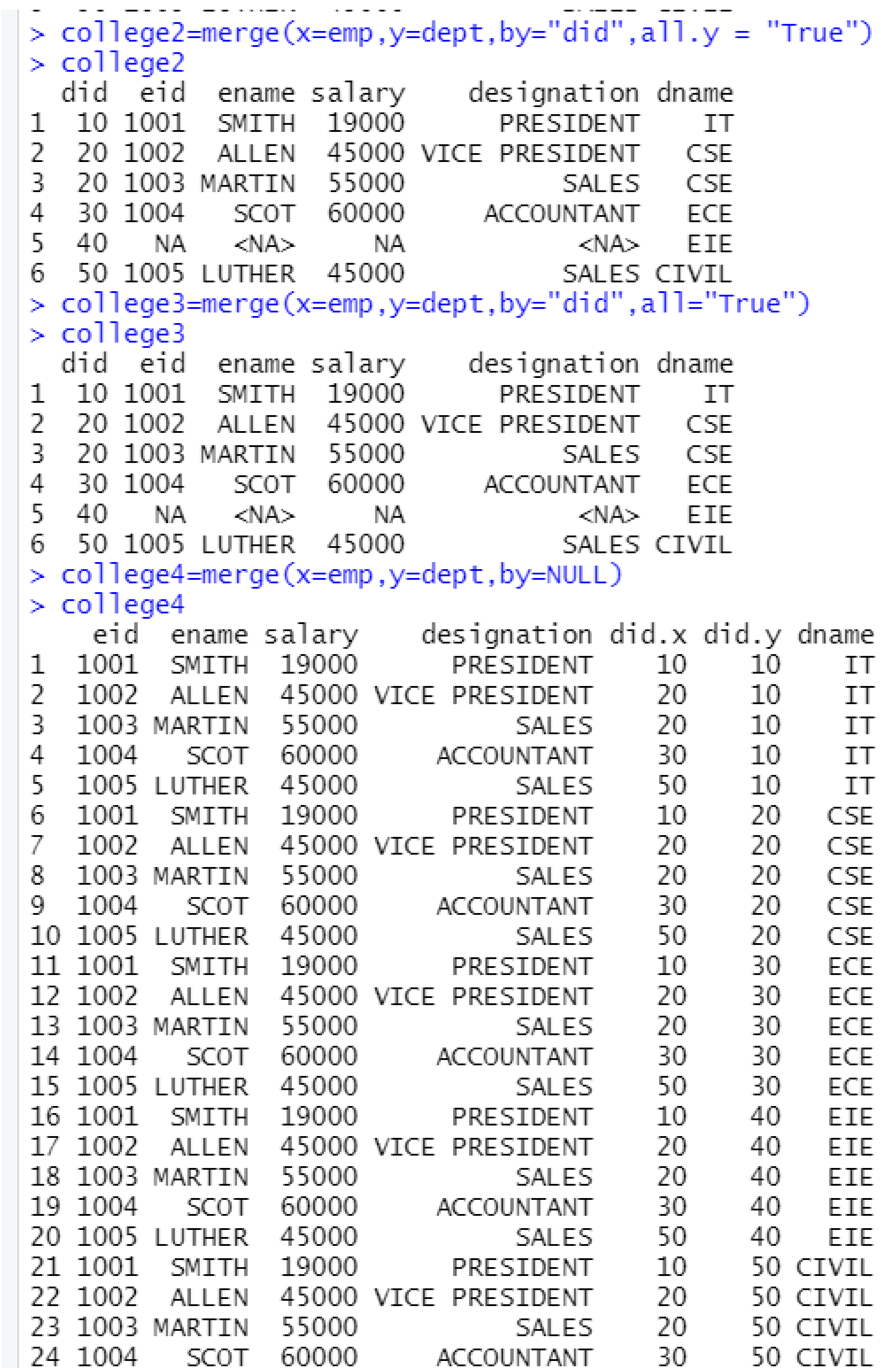
EDITOR CODE:



```
sport<- c("Hockey", "Baseball","Football")
league<- c("NHL", "MLB", "NFL")
trophy<- c("Stanley Cup","Commissioner's Trophy", "Vince Lombardi Trophy")
trophies1<- cbind(sport, league, trophy)
trophies1
trophies2<- data.frame(sport=c("Basketball", "Golf"), league=c("NBA", "PGA"), trophy=c("Larry O'Brien Championship Trophy
trophies2
trophies<-rbind(trophies1,trophies2)
trophies
cbind(Sport=sport, Association=league, Prize= trophy)
df1 = data.frame(StudentId = c(101:106),Product = c("Hindi", "English","Maths", "Science","Political Science","Physics")
df1
df2 = data.frame(StudentId = c(102, 104, 106, 107, 108), State = c("Manglore", "Mysore","Pune", "Dehradun", "Delhi"))
df2
df = merge(x = df1, y = df2, by = "StudentId")
df
install.packages("plyr")
require(plyr)
eid<-1001:1005
ename=c("SMITH","ALLEN","MARTIN","SCOT","LUTHER")
designation<-c("PRESIDENT","VICE PRESIDENT","SALES","ACCOUNTANT","SALES")
salary<-c(19000,45000,55000,60000,45000)
did=c(10,20,20,30,50)
emp<-data.frame(eid,ename,salary,designation,did)
emp
did<-c(10,20,30,40,50)
dname<-c("IT","CSE","ECE","EIE","CIVIL")
dept<-data.frame(did,dname)
dept
college<-merge(x=emp,y=dept,by="did")
college
college1=merge(x=emp,y=dept,by="did",all.x ="True")
college1
college2=merge(x=emp,y=dept,by="did",all.y = "True")
college2
college3=merge(x=emp,y=dept,by="did",all="True")
college3
college4=merge(x=emp,y=dept,by=NULL)
college4
```



```
Loading required package: plyr
> eid<-1001:1005
> ename=c("SMITH","ALLEN","MARTIN","SCOT","LUTHER")
> designation<-c("PRESIDENT","VICE PRESIDENT","SALES","ACCOUNTANT","SALES")
> salary<-c(19000,45000,55000,60000,45000)
> did=c(10,20,20,30,50)
> emp<-data.frame(eid,ename,salary,designation,did)
> emp
   eid  ename salary    designation did
1 1001  SMITH  19000      PRESIDENT  10
2 1002  ALLEN  45000 VICE PRESIDENT  20
3 1003 MARTIN  55000          SALES  20
4 1004   SCOT  60000     ACCOUNTANT  30
5 1005 LUTHER  45000          SALES  50
> did<-c(10,20,30,40,50)
> dname<-c("IT","CSE","ECE","EIE","CIVIL")
> dept<-data.frame(did,dname)
> dept
  did dname
1  10    IT
2  20   CSE
3  30   ECE
4  40   EIE
5  50 CIVIL
> college<-merge(x=emp,y=dept,by="did")
> college
  did  eid  ename salary    designation dname
1  10 1001  SMITH  19000      PRESIDENT    IT
2  20 1002  ALLEN  45000 VICE PRESIDENT   CSE
3  20 1003 MARTIN  55000          SALES   CSE
4  30 1004   SCOT  60000     ACCOUNTANT   ECE
5  50 1005 LUTHER  45000          SALES CIVIL
> college1=merge(x=emp,y=dept,by="did",all.x ="True")
> college1
  did  eid  ename salary    designation dname
1  10 1001  SMITH  19000      PRESIDENT    IT
2  20 1002  ALLEN  45000 VICE PRESIDENT   CSE
3  20 1003 MARTIN  55000          SALES   CSE
4  30 1004   SCOT  60000     ACCOUNTANT   ECE
5  50 1005 LUTHER  45000          SALES CIVIL
```



```
> college2=merge(x=emp,y=dept,by="did",all.y = "True")
> college2
  did  eid  ename salary    designation dname
1  10 1001  SMITH  19000      PRESIDENT    IT
2  20 1002  ALLEN  45000 VICE PRESIDENT   CSE
3  20 1003 MARTIN  55000          SALES   CSE
4  30 1004   SCOT  60000     ACCOUNTANT   ECE
5  40   NA   <NA>     NA           <NA>   EIE
6  50 1005 LUTHER  45000          SALES CIVIL
> college3=merge(x=emp,y=dept,by="did",all="True")
> college3
  did  eid  ename salary    designation dname
1  10 1001  SMITH  19000      PRESIDENT    IT
2  20 1002  ALLEN  45000 VICE PRESIDENT   CSE
3  20 1003 MARTIN  55000          SALES   CSE
4  30 1004   SCOT  60000     ACCOUNTANT   ECE
5  40   NA   <NA>     NA           <NA>   EIE
6  50 1005 LUTHER  45000          SALES CIVIL
> college4=merge(x=emp,y=dept,by=NULL)
> college4
    eid  ename salary    designation did.x did.y dname
1  1001  SMITH  19000      PRESIDENT    10    10    IT
2  1002  ALLEN  45000 VICE PRESIDENT    20    10    IT
3  1003 MARTIN  55000          SALES    20    10    IT
4  1004   SCOT  60000     ACCOUNTANT    30    10    IT
5  1005 LUTHER  45000          SALES    50    10    IT
6  1001  SMITH  19000      PRESIDENT    10    20   CSE
7  1002  ALLEN  45000 VICE PRESIDENT    20    20   CSE
8  1003 MARTIN  55000          SALES    20    20   CSE
9  1004   SCOT  60000     ACCOUNTANT    30    20   CSE
10 1005 LUTHER  45000          SALES    50    20   CSE
11 1001  SMITH  19000      PRESIDENT    10    30   ECE
12 1002  ALLEN  45000 VICE PRESIDENT    20    30   ECE
13 1003 MARTIN  55000          SALES    20    30   ECE
14 1004   SCOT  60000     ACCOUNTANT    30    30   ECE
15 1005 LUTHER  45000          SALES    50    30   ECE
16 1001  SMITH  19000      PRESIDENT    10    40   EIE
17 1002  ALLEN  45000 VICE PRESIDENT    20    40   EIE
18 1003 MARTIN  55000          SALES    20    40   EIE
19 1004   SCOT  60000     ACCOUNTANT    30    40   EIE
20 1005 LUTHER  45000          SALES    50    40   EIE
21 1001  SMITH  19000      PRESIDENT    10    50 CIVIL
22 1002  ALLEN  45000 VICE PRESIDENT    20    50 CIVIL
23 1003 MARTIN  55000          SALES    20    50 CIVIL
24 1004   SCOT  60000     ACCOUNTANT    30    50 CIVIL
```

RESULT:

Hence the code is executed successfully.

R LAB-08

TASK-01:

Implement string manipulation and math functions in r studio

PROGRAM:

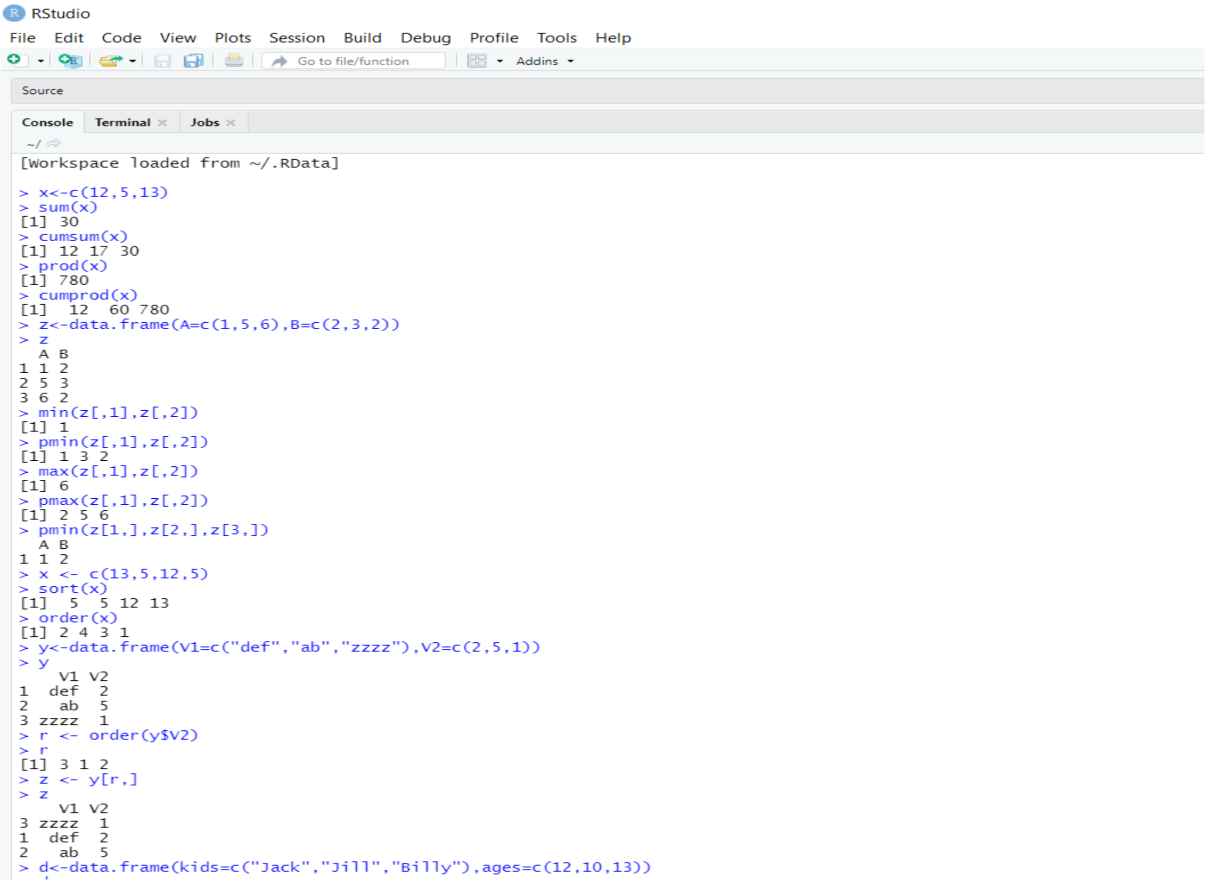
MATH FUNCTIONS:

```
x<-c (12,5,13)
> sum(x)
[1] 30
> cumsum (x)
[1] 12 17 30
> prod(x)
[1] 780
> cumprod (x)
[1] 12 60 780
> z<-data. frame (A=c (1,5,6), B=c (2,3,2))
> z
  A B
1 1 2
2 5 3
3 6 2
> min (z [,1], z[,2])
[1] 1
> pmin(z[,1], z[,2])
[1] 1 3 2
> max(z[,1],z[,2])
[1] 6
> pmax(z[,1],z[,2])
[1] 2 5 6
> pmin(z[1,],z[2,],z[3,])
  A B
1 1 2
> x <- c(13,5,12,5)
> sort(x)
[1]  5  5 12 13
> order(x)
[1] 2 4 3 1
> y<-data.frame(V1=c("def","ab","zzzz"),V2=c(2,5,1))
> y
    V1 V2
1  def  2
2   ab  5
3 zzzz  1
> r <- order(y$V2)
> r
[1] 3 1 2
> z <- y[r,]
> z
    V1 V2
3 zzzz  1
1  def  2
2   ab  5
> d<-data.frame(kids=c("Jack","Jill","Billy"),ages=c(12,10,13))
> d
   kids ages
1  Jack   12
2  Jill   10
3 Billy   13
> d[order(d$kids),]
   kids ages
3 Billy   13
1  Jack   12
2  Jill   10
> d[order(d$ages),]
   kids ages
2  Jill   10
1  Jack   12
```

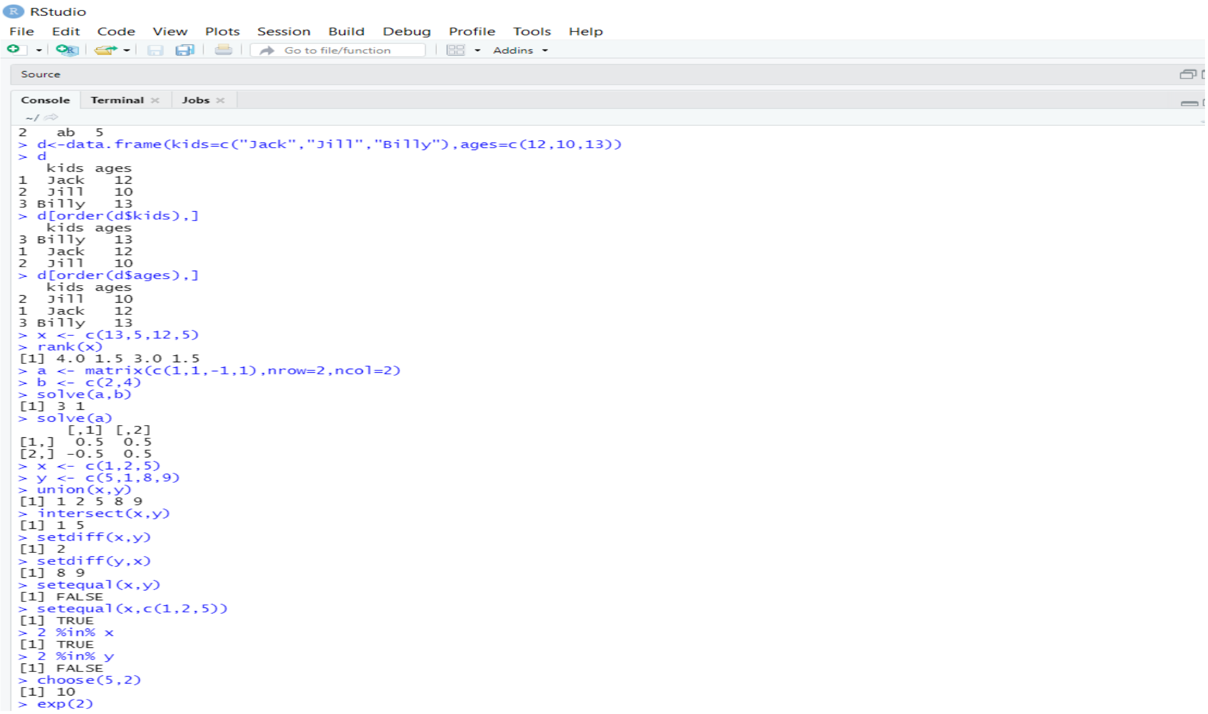
```
3 Billy   13
> x <- c(13,5,12,5)
> rank(x)
[1] 4.0 1.5 3.0 1.5
> a <- matrix(c(1,1,-1,1),nrow=2,ncol=2)
> b <- c(2,4)
> solve(a,b)
[1] 3 1
> solve(a)
     [,1] [,2]
[1,]  0.5  0.5
[2,] -0.5  0.5
> x <- c(1,2,5)
> y <- c(5,1,8,9)
> union(x,y)
[1] 1 2 5 8 9
> intersect(x,y)
[1] 1 5
> setdiff(x,y)
[1] 2
> setdiff(y,x)
[1] 8 9
> setequal(x,y)
[1] FALSE
> setequal(x,c(1,2,5))
[1] TRUE
> 2 %in% x
[1] TRUE
> 2 %in% y
[1] FALSE
> choose(5,2)
[1] 10
> exp(2)
[1] 7.389056
```

```
> log(5)
[1] 1.609438
> log10(2)
[1] 0.30103
> sqrt(25)
[1] 5
> abs(5)
[1] 5
> sin(90)
[1] 0.8939967
> cos(90)
[1] -0.4480736
> round(4.56)
[1] 5
> floor(2.56)
[1] 2
> ceiling(8.97)
[1] 9
> factorial(6)
[1] 720
> exactlyone <- function(p) {
+   notp <-1-p
+   tot <- 0.0
+   for (i in 1:length(p))
+     tot <- tot + p[i] * prod(notp[-i])
+   return(tot)
+ }
> exactlyone(c(1,2,34,56))
[1] -1815
```

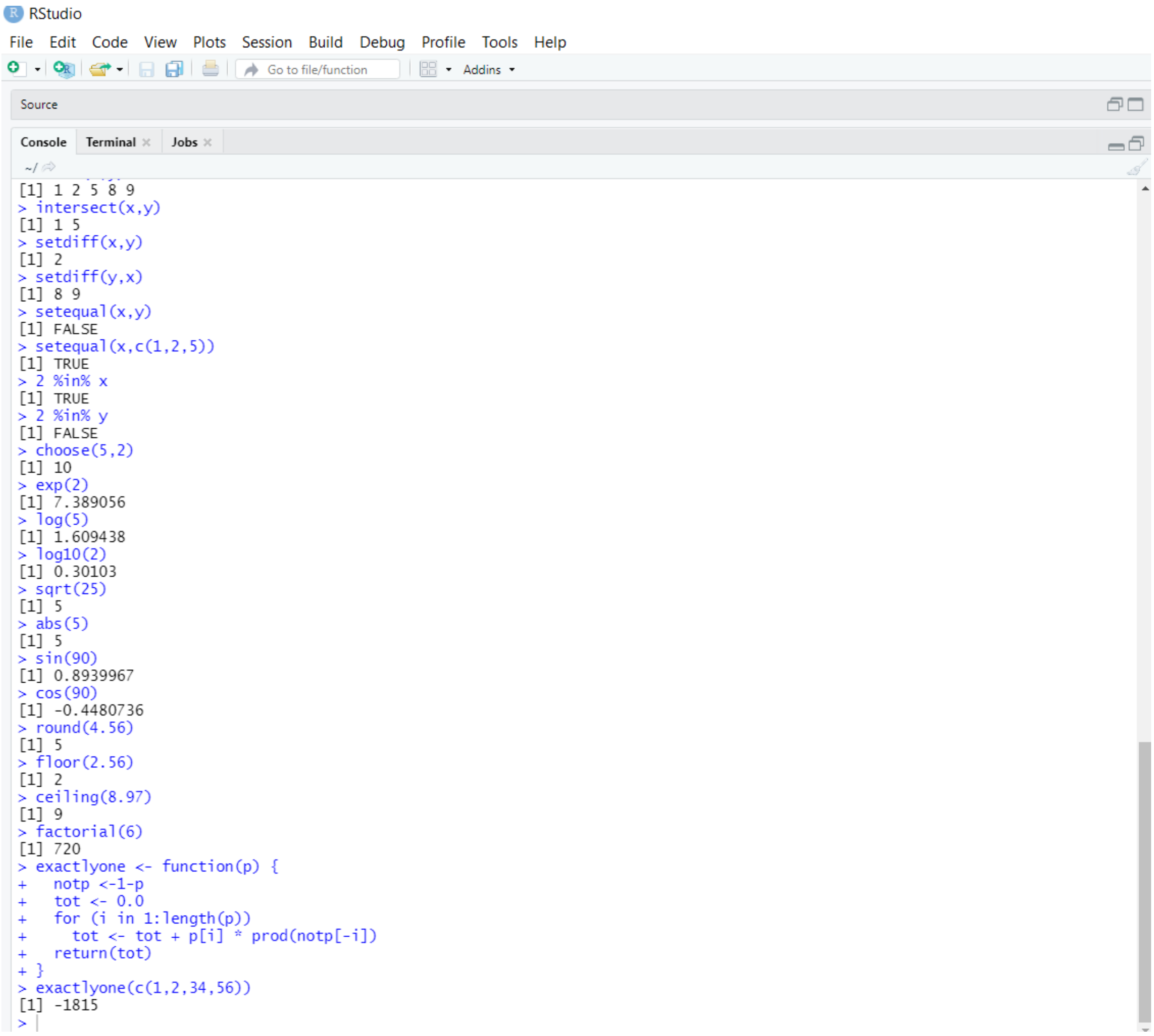
EDITOR CODE:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
~/

```
[Workspace loaded from ~/.RData]

> x<-c(12,5,13)
> sum(x)
[1] 30
> cumsum(x)
[1] 12 17 30
> prod(x)
[1] 780
> cumprod(x)
[1]  12  60 780
> z<-data.frame(A=c(1,5,6),B=c(2,3,2))
> z
  A B
1 1 2
2 5 3
3 6 2
> min(z[,1],z[,2])
[1] 1
> pmin(z[,1],z[,2])
[1] 1 3 2
> max(z[,1],z[,2])
[1] 6
> pmax(z[,1],z[,2])
[1] 2 5 6
> pmin(z[1,],z[2,],z[3,])
  A B
1 1 2
> x <- c(13,5,12,5)
> sort(x)
[1]  5  5 12 13
> order(x)
[1] 2 4 3 1
> y<-data.frame(V1=c("def","ab","zzzz"),V2=c(2,5,1))
> y
    V1 V2
1  def  2
2   ab  5
3 zzzz  1
> r <- order(y$V2)
> r
[1] 3 1 2
> z <- y[r,]
> z
    V1 V2
3 zzzz  1
1  def  2
2   ab  5
> d<-data.frame(kids=c("Jack","Jill","Billy"),ages=c(12,10,13))
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
~/

```
2   ab  5
> d<-data.frame(kids=c("Jack","Jill","Billy"),ages=c(12,10,13))
> d
   kids ages
1  Jack   12
2  Jill   10
3 Billy   13
> d[order(d$kids),]
   kids ages
3 Billy   13
1  Jack   12
2  Jill   10
> d[order(d$ages),]
   kids ages
2  Jill   10
1  Jack   12
3 Billy   13
> x <- c(13,5,12,5)
> rank(x)
[1] 4.0 1.5 3.0 1.5
> a <- matrix(c(1,1,-1,1),nrow=2,ncol=2)
> b <- c(2,4)
> solve(a,b)
[1] 3 1
> solve(a)
     [,1] [,2]
[1,]  0.5  0.5
[2,] -0.5  0.5
> x <- c(1,2,5)
> y <- c(5,1,8,9)
> union(x,y)
[1] 1 2 5 8 9
> intersect(x,y)
[1] 1 5
> setdiff(x,y)
[1] 2
> setdiff(y,x)
[1] 8 9
> setequal(x,y)
[1] FALSE
> setequal(x,c(1,2,5))
[1] TRUE
> 2 %in% x
[1] TRUE
> 2 %in% y
[1] FALSE
> choose(5,2)
[1] 10
> exp(2)
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
~/

```
[1] 1 2 5 8 9
> intersect(x,y)
[1] 1 5
> setdiff(x,y)
[1] 2
> setdiff(y,x)
[1] 8 9
> setequal(x,y)
[1] FALSE
> setequal(x,c(1,2,5))
[1] TRUE
> 2 %in% x
[1] TRUE
> 2 %in% y
[1] FALSE
> choose(5,2)
[1] 10
> exp(2)
[1] 7.389056
> log(5)
[1] 1.609438
> log10(2)
[1] 0.30103
> sqrt(25)
[1] 5
> abs(5)
[1] 5
> sin(90)
[1] 0.8939967
> cos(90)
[1] -0.4480736
> round(4.56)
[1] 5
> floor(2.56)
[1] 2
> ceiling(8.97)
[1] 9
> factorial(6)
[1] 720
> exactlyone <- function(p) {
+   notp <-1-p
+   tot <- 0.0
+   for (i in 1:length(p))
+     tot <- tot + p[i] * prod(notp[-i])
+   return(tot)
+ }
> exactlyone(c(1,2,34,56))
[1] -1815
> 
```

MELT AND CAST:

CODE:

install. packages ("reshape2")

library (MASS)

library(reshape2)

print (head (ships, n=10))

shipdata<-(head (ships, n=10))

molten. ships <- melt (shipdata, id = c("type","year"))

print (molten. ships)

recasted. ship <- dcast (molten. ships, type+year~variable, sum)

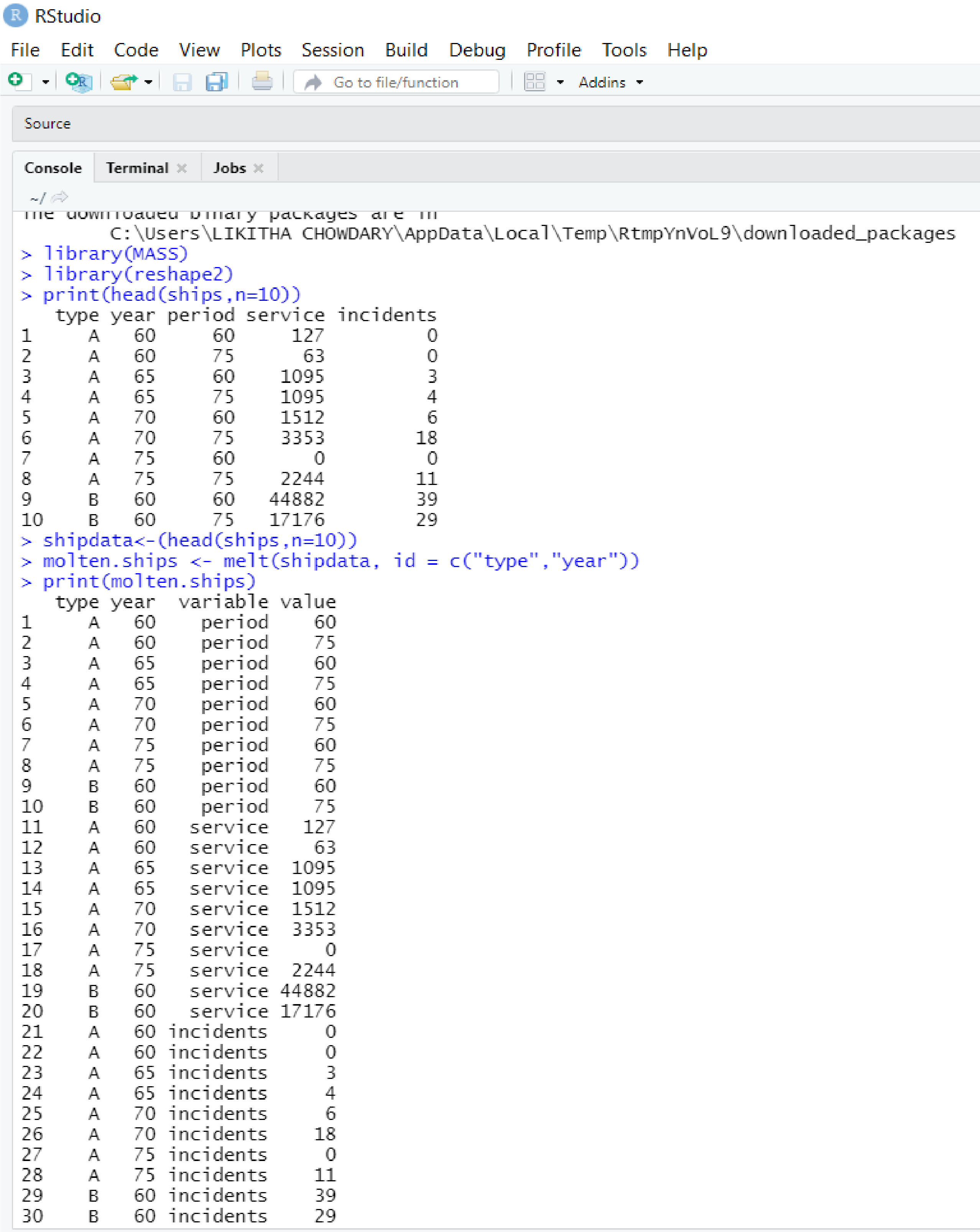
print (recasted. ship)

OUTPUT:

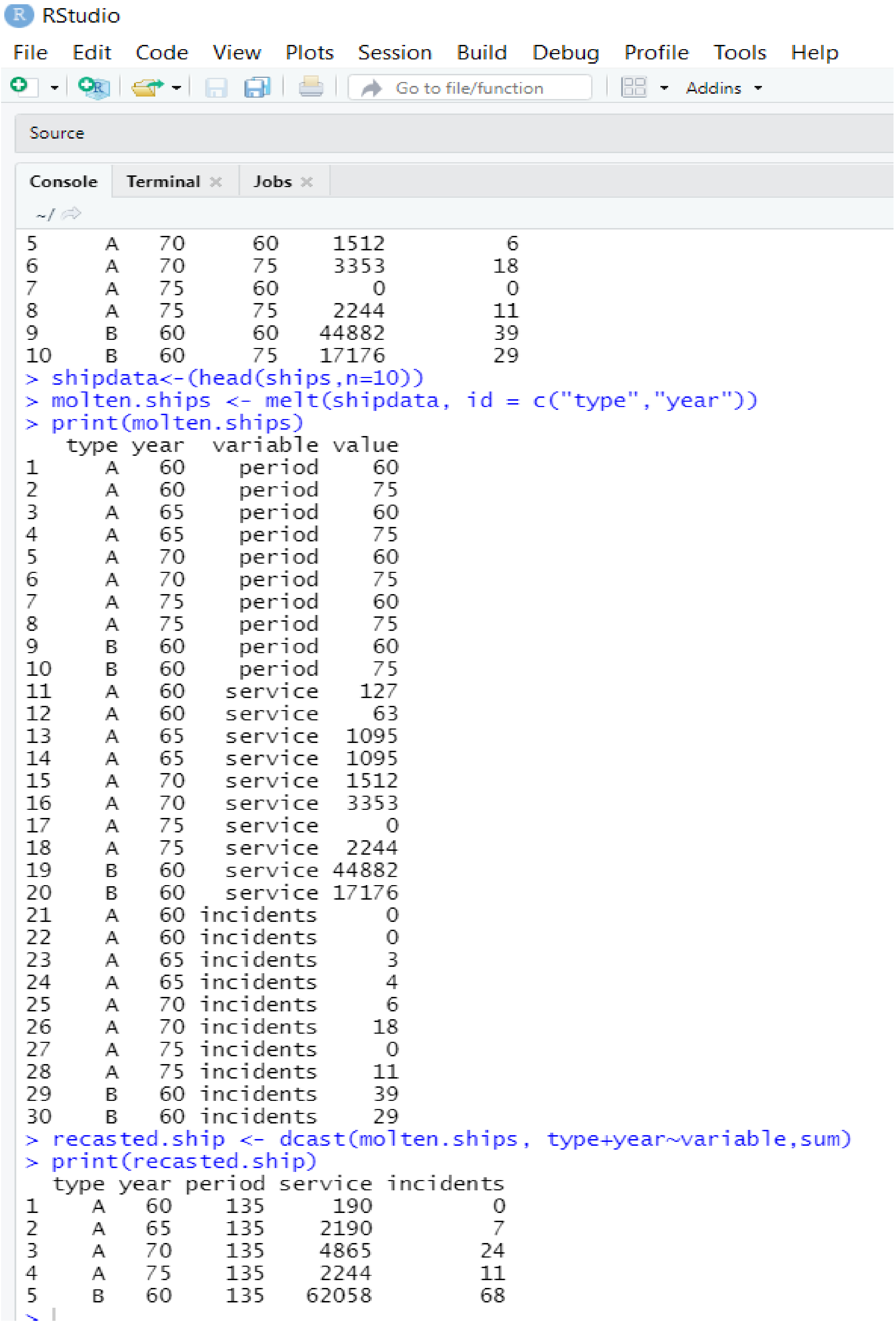
```
Library (MASS)
> library(reshape2)
> print (head (ships, n=10))
   type year period service incidents
1     A   60     60     127         0
2     A   60     75      63         0
3     A   65     60    1095         3
4     A   65     75    1095         4
5     A   70     60    1512         6
6     A   70     75    3353        18
7     A   75     60       0         0
8     A   75     75    2244        11
9     B   60     60   44882        39
10    B   60     75   17176        29
> shipdata<-(head (ships, n=10))
> molten. ships <- melt (shipdata, id = c("type","year"))
> print (molten. ships)
   type year variable value
1     A   60    period    60
2     A   60    period    75
3     A   65    period    60
4     A   65    period    75
5     A   70    period    60
6     A   70    period    75
7     A   75    period    60
8     A   75    period    75
9     B   60    period    60
10    B   60    period    75
11    A   60   service   127
12    A   60   service    63
13    A   65   service 1095
14    A   65   service 1095
```

```
15    A   70   service 1512
16    A   70   service 3353
17    A   75   service     0
18    A   75   service 2244
19    B   60   service 44882
20    B   60   service 17176
21    A   60 incidents     0
22    A   60 incidents     0
23    A   65 incidents     3
24    A   65 incidents     4
25    A   70 incidents     6
26    A   70 incidents    18
27    A   75 incidents     0
28    A   75 incidents    11
29    B   60 incidents    39
30    B   60 incidents    29
> recasted. ship <- dcast (molten. ships, type+year~variable, sum)
> print (recasted. ship)
  type year period service incidents
1    A   60    135     190         0
2    A   65    135    2190         7
3    A   70    135    4865        24
4    A   75    135    2244        11
5    B   60    135   62058        68
```

EDITOR CODE:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
~/

```
The downloaded binary packages are in
        C:\Users\LIKITHA CHOWDARY\AppData\Local\Temp\RtmpYnVoL9\downloaded_packages
> library(MASS)
> library(reshape2)
> print(head(ships,n=10))
   type year period service incidents
1     A   60     60     127         0
2     A   60     75      63         0
3     A   65     60    1095         3
4     A   65     75    1095         4
5     A   70     60    1512         6
6     A   70     75    3353        18
7     A   75     60       0         0
8     A   75     75    2244        11
9     B   60     60   44882        39
10    B   60     75   17176        29
> shipdata<-(head(ships,n=10))
> molten.ships <- melt(shipdata, id = c("type","year"))
> print(molten.ships)
   type year  variable value
1     A   60    period    60
2     A   60    period    75
3     A   65    period    60
4     A   65    period    75
5     A   70    period    60
6     A   70    period    75
7     A   75    period    60
8     A   75    period    75
9     B   60    period    60
10    B   60    period    75
11    A   60   service   127
12    A   60   service    63
13    A   65   service  1095
14    A   65   service  1095
15    A   70   service  1512
16    A   70   service  3353
17    A   75   service     0
18    A   75   service  2244
19    B   60   service 44882
20    B   60   service 17176
21    A   60 incidents     0
22    A   60 incidents     0
23    A   65 incidents     3
24    A   65 incidents     4
25    A   70 incidents     6
26    A   70 incidents    18
27    A   75 incidents     0
28    A   75 incidents    11
29    B   60 incidents    39
30    B   60 incidents    29
```



```
5      A   70     60    1512        6
6      A   70     75    3353       18
7      A   75     60       0        0
8      A   75     75    2244       11
9      B   60     60   44882       39
10     B   60     75   17176       29
> shipdata<-(head(ships,n=10))
> molten.ships <- melt(shipdata, id = c("type","year"))
> print(molten.ships)
   type year  variable value
1     A   60    period    60
2     A   60    period    75
3     A   65    period    60
4     A   65    period    75
5     A   70    period    60
6     A   70    period    75
7     A   75    period    60
8     A   75    period    75
9     B   60    period    60
10    B   60    period    75
11    A   60   service   127
12    A   60   service    63
13    A   65   service  1095
14    A   65   service  1095
15    A   70   service  1512
16    A   70   service  3353
17    A   75   service     0
18    A   75   service  2244
19    B   60   service 44882
20    B   60   service 17176
21    A   60 incidents     0
22    A   60 incidents     0
23    A   65 incidents     3
24    A   65 incidents     4
25    A   70 incidents     6
26    A   70 incidents    18
27    A   75 incidents     0
28    A   75 incidents    11
29    B   60 incidents    39
30    B   60 incidents    29
> recasted.ship <- dcast(molten.ships, type+year~variable,sum)
> print(recasted.ship)
  type year period service incidents
1    A   60    135     190         0
2    A   65    135    2190         7
3    A   70    135    4865        24
4    A   75    135    2244        11
5    B   60    135   62058        68
> 
```

R LAB-9

MANIPULATING STRINGS:

CODE:

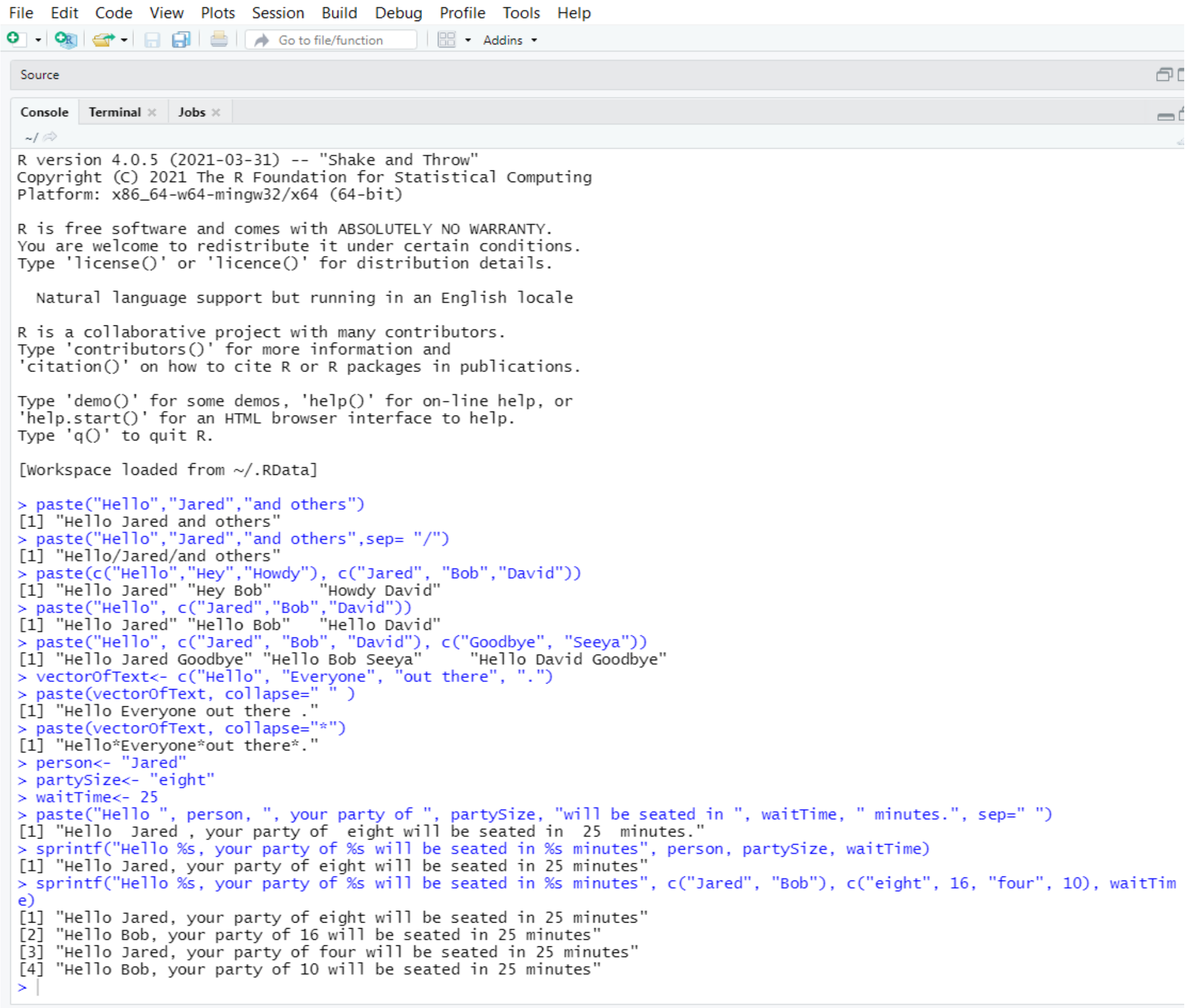
```
paste ("Hello","Jared","and others")
paste ("Hello","Jared","and others”, sep= "/")
paste(c("Hello","Hey","Howdy"), c ("Jared", "Bob","David"))
paste ("Hello", c("Jared","Bob","David"))
paste ("Hello", c ("Jared", "Bob", "David"), c ("Goodbye", "Seeya"))
vectorOfText<- c ("Hello", "Everyone", "out there", ".")
paste (vectorOfText, collapse=" “)
paste (vectorOfText, collapse="*")
person<- "Jared"
partySize<- "eight"
waitTime<- 25
paste ("Hello ", person, ", your party of ", partySize, "will be seated in ", waitTime, " minutes.", sep=" ")
sprintf ("Hello %s, your party of %s will be seated in %s minutes", person, partySize, waitTime)
sprintf ("Hello %s, your party of %s will be seated in %s minutes", c ("Jared", "Bob"), c ("eight", 16, "four", 10), waitTime)
install. packages ("stringr")
library("stringr")
jr = c ("Theo is first", "Esther is second", "Colin - third")
str_detect (jr, "Theo")
str_detect (jr, "is")
str_subset (jr, "Theo")
str_subset (jr, "is")
files = c (
  "tmp-project.csv", "project.csv",
  "project2-csv-specs.csv", "project2.csv2.specs.xlsx",
  "project_cars.ods", "project-houses.csv",
  "Project_Trees.csv","project-cars. R",
  "Project-houses. r", "project-final.xls",
  "Project-final2.xlsx"
)
str_subset (files, "\\.csv")
str_subset (files, "^Proj")
str_subset (files, "\\.csv$")
str_subset (files, "\\.ods$")
```

OUTPUT:

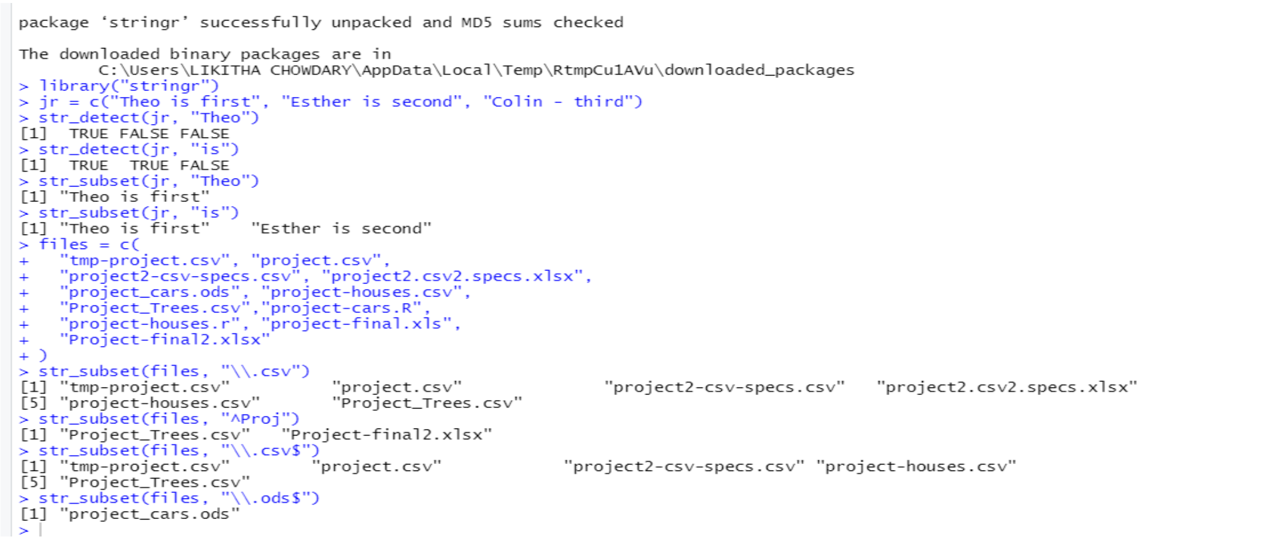
```
paste ("Hello","Jared","and others")
[1] "Hello Jared and others"
> paste ("Hello","Jared","and others”, sep= "/")
[1] "Hello/Jared/and others"
> paste(c("Hello","Hey","Howdy"), c("Jared", "Bob","David"))
[1] "Hello Jared" "Hey Bob"     "Howdy David"
> paste ("Hello", c("Jared","Bob","David"))
[1] "Hello Jared" "Hello Bob"   "Hello David"
> paste ("Hello", c ("Jared", "Bob", "David"), c ("Goodbye", "Seeya"))
[1] "Hello Jared Goodbye" "Hello Bob Seeya"     "Hello David Goodbye"
> vectorOfText<- c ("Hello", "Everyone", "out there", ".")
> paste (vectorOfText, collapse=" “)
[1] "Hello Everyone out there."
> paste (vectorOfText, collapse="*")
[1] "Hello*Everyone*out there*."
> person<- "Jared"
> partySize<- "eight"
> waitTime<- 25
> paste ("Hello ", person, ", your party of ", partySize, "will be seated in ", waitTime, "
minutes.", sep=" ")
[1] "Hello Jared, your party of eight will be seated in 25 minutes."
> sprintf ("Hello %s, your party of %s will be seated in %s minutes", person, partySize,
waitTime)
[1] "Hello Jared, your party of eight will be seated in 25 minutes"
> sprintf ("Hello %s, your party of %s will be seated in %s minutes", c ("Jared", "Bob"),
c("eight", 16, "four", 10), waitTime)
```

```
[1] "Hello Jared, your party of eight will be seated in 25 minutes"
[2] "Hello Bob, your party of 16 will be seated in 25 minutes"
[3] "Hello Jared, your party of four will be seated in 25 minutes"
[4] "Hello Bob, your party of 10 will be seated in 25 minutes"
library("stringr")
> jr = c ("Theo is first", "Esther is second", "Colin - third")
> str_detect (jr, "Theo")
[1] TRUE FALSE FALSE
> str_detect (jr, "is")
[1] TRUE TRUE FALSE
> str_subset (jr, "Theo")
[1] "Theo is first"
> str_subset (jr, "is")
[1] "Theo is first"    "Esther is second"
> files = c (
+   "tmp-project.csv", "project.csv",
+   "project2-csv-specs.csv", "project2.csv2.specs.xlsx",
+   "project_cars.ods", "project-houses.csv",
+   "Project_Trees.csv","project-cars. R",
+   "project-houses. r", "project-final.xls",
+   "Project-final2.xlsx"
+ )
> str_subset (files, "\\.csv")
[1]  "tmp-project.csv"                    "project.csv"                              "project2-csv-specs.csv"
"project2.csv2.specs.xlsx"
[5] "project-houses.csv"       "Project_Trees.csv"
> str_subset (files, "^Proj")
[1] "Project_Trees.csv"   "Project-final2.xlsx"
> str_subset (files, "\\.csv$")
[1] "tmp-project.csv"        "project.csv"           "project2-csv-specs.csv" "project-houses.csv"
[5] "Project_Trees.csv"
> str_subset (files, "\\.ods$")
[1] "project_cars.ods"
>
```

EDITOR CODE:


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
~/

```
R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> paste("Hello","Jared","and others")
[1] "Hello Jared and others"
> paste("Hello","Jared","and others",sep= "/")
[1] "Hello/Jared/and others"
> paste(c("Hello","Hey","Howdy"), c("Jared", "Bob","David"))
[1] "Hello Jared" "Hey Bob"     "Howdy David"
> paste("Hello", c("Jared","Bob","David"))
[1] "Hello Jared" "Hello Bob"   "Hello David"
> paste("Hello", c("Jared", "Bob", "David"), c("Goodbye", "Seeya"))
[1] "Hello Jared Goodbye" "Hello Bob Seeya"     "Hello David Goodbye"
> vectorOfText<- c("Hello", "Everyone", "out there", ".")
> paste(vectorOfText, collapse=" " )
[1] "Hello Everyone out there ."
> paste(vectorOfText, collapse="*")
[1] "Hello*Everyone*out there*."
> person<- "Jared"
> partySize<- "eight"
> waitTime<- 25
> paste("Hello ", person, ", your party of ", partySize, "will be seated in ", waitTime, " minutes.", sep=" ")
[1] "Hello  Jared , your party of  eight will be seated in  25  minutes."
> sprintf("Hello %s, your party of %s will be seated in %s minutes", person, partySize, waitTime)
[1] "Hello Jared, your party of eight will be seated in 25 minutes"
> sprintf("Hello %s, your party of %s will be seated in %s minutes", c("Jared", "Bob"), c("eight", 16, "four", 10), waitTim
e)
[1] "Hello Jared, your party of eight will be seated in 25 minutes"
[2] "Hello Bob, your party of 16 will be seated in 25 minutes"
[3] "Hello Jared, your party of four will be seated in 25 minutes"
[4] "Hello Bob, your party of 10 will be seated in 25 minutes"
> 
```



```
package ‘stringr’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\LIKITHA CHOWDARY\AppData\Local\Temp\RtmpCu1AVu\downloaded_packages
> library("stringr")
> jr = c("Theo is first", "Esther is second", "Colin - third")
> str_detect(jr, "Theo")
[1]  TRUE FALSE FALSE
> str_detect(jr, "is")
[1]  TRUE  TRUE FALSE
> str_subset(jr, "Theo")
[1] "Theo is first"
> str_subset(jr, "is")
[1] "Theo is first"    "Esther is second"
> files = c(
+   "tmp-project.csv", "project.csv",
+   "project2-csv-specs.csv", "project2.csv2.specs.xlsx",
+   "project_cars.ods", "project-houses.csv",
+   "Project_Trees.csv","project-cars.R",
+   "project-houses.r", "project-final.xls",
+   "Project-final2.xlsx"
+ )
> str_subset(files, "\\.csv")
[1] "tmp-project.csv"          "project.csv"              "project2-csv-specs.csv"   "project2.csv2.specs.xlsx"
[5] "project-houses.csv"       "Project_Trees.csv"
> str_subset(files, "^Proj")
[1] "Project_Trees.csv"   "Project-final2.xlsx"
> str_subset(files, "\\.csv$")
[1] "tmp-project.csv"        "project.csv"            "project2-csv-specs.csv" "project-houses.csv"
[5] "Project_Trees.csv"
> str_subset(files, "\\.ods$")
[1] "project_cars.ods"
> 
```

RESULT:

HENCE THE CODE IS EXECUTED SUCCESSFULLY.

R LAB-10

TASK-01:

Implement basic summary statistics concept in r studio

PROGRAM:

```
x<-sample (x=1:100, size=100, replace= TRUE)

x

mean(x)

y<-x

y [sample (x=1:100, size=20, replace=FALSE)] <-NA

y

mean(y)

mean (y, na.rm=TRUE)

grades<-c (95,72,87,66)

weights<-c (1/2,1/4,1/8,1/8)

mean(grades)

mean(weights)

weighted. mean (x=grades, w=weights)

var(x)

sum((x-mean(x)) ^2)/(length(x)-1)

sqrt(var(x))

sd (x)

sd(y)

sd (y, na.rm=TRUE)

min(x)

max(x)

mean(x)

min(y)

max(y)

summary(x)

summary(y)

quantile (x, probs=c (0.25,0.75))

quantile (y, probs=c (0.25,0.75))
quantile (y, probs=c(0.25,0.75), na.rm=TRUE)
```

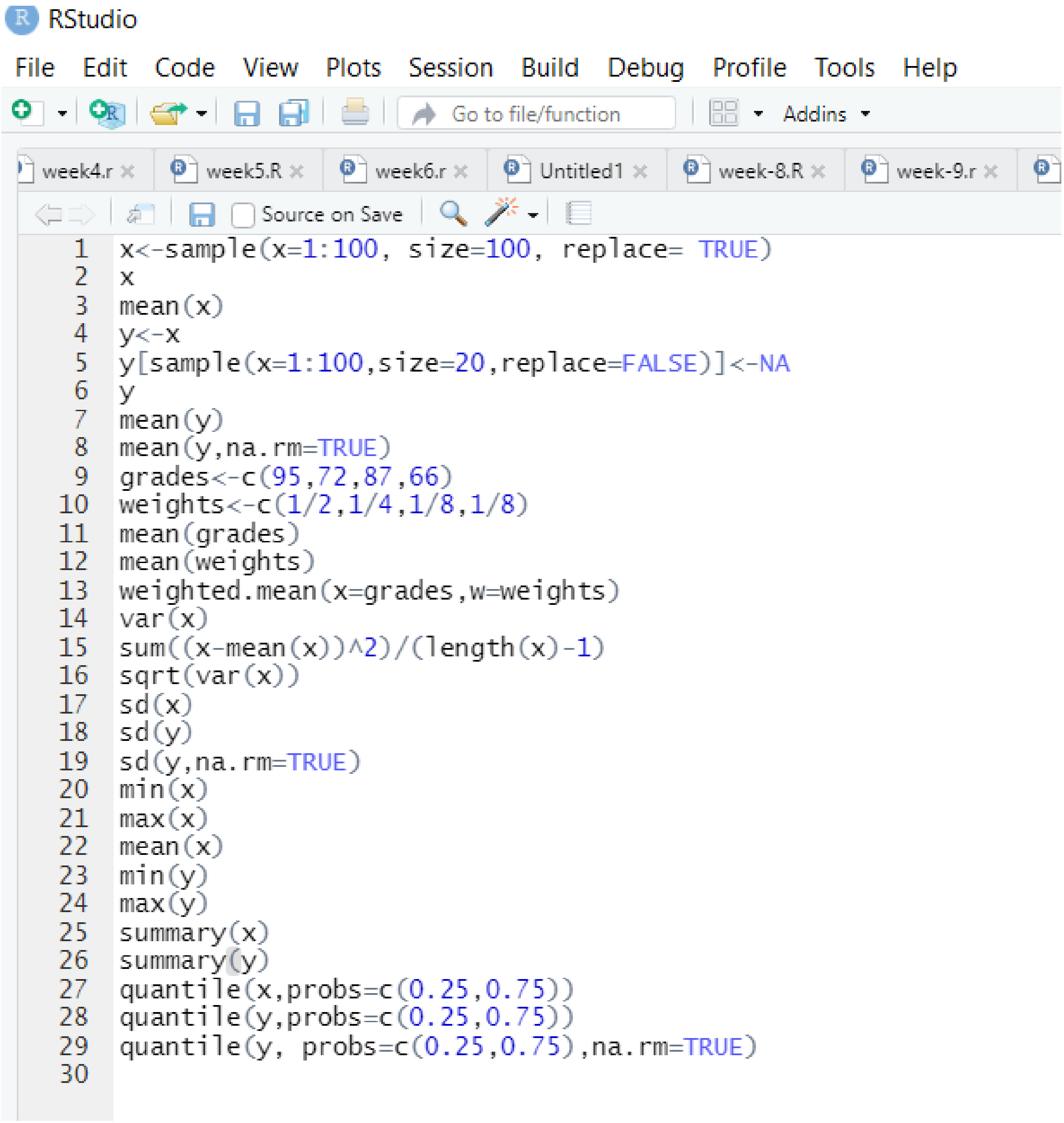
OUTPUT:

```
x<-sample (x=1:100, size=100, replace= TRUE)
> x
  [1] 49 26 69 63 19 70 43 93 39 20 16 26 32 34 70 22 9 12 63 61 84 60 30 63 27 7 87 75 50
80 77 43 48 49 55 94 28 49
 [39] 93 26 34 92 87 15 38 49 10 28 36 35 95 59 69 2 89 19 11 27 34 61 65 18 83 95 21 98 18
5 32 78 30 38 57 49 6 27
 [77] 97 57 18 18 49 57 2 50 98 18 4 97 56 33 43 66 69 74 97 13 18 28 66 99
> mean(x)
[1] 47.98
> y<-x
> y [sample (x=1:100, size=20, replace=FALSE)]<-NA
> y
  [1] 49 NA 69 63 19 70 43 93 39 20 NA NA 32 NA 70 22  9 NA 63 61 NA 60 30 63 27  7 87
75 50 80 NA 43 48 49 55 94 28 49
 [39] 93 26 34 92 NA 15 38 49 10 28 36 35 NA 59 NA  2 89 19 11 27 34 61 65 18 83 95 21 98
NA  5 NA 78 30 38 NA NA  6 27
 [77] NA 57 18 18 49 57 NA NA 98 18  4 NA NA 33 43 66 69 74 97 13 18 28 66 NA
> mean(y)
[1] NA
> mean(y,na.rm=TRUE)
[1] 46.4375
> grades<-c(95,72,87,66)
> weights<-c(1/2,1/4,1/8,1/8)
> mean(grades)
[1] 80
> mean(weights)
[1] 0.25
> weighted.mean(x=grades,w=weights)
[1] 84.625
> var(x)
[1] 807.3935
```

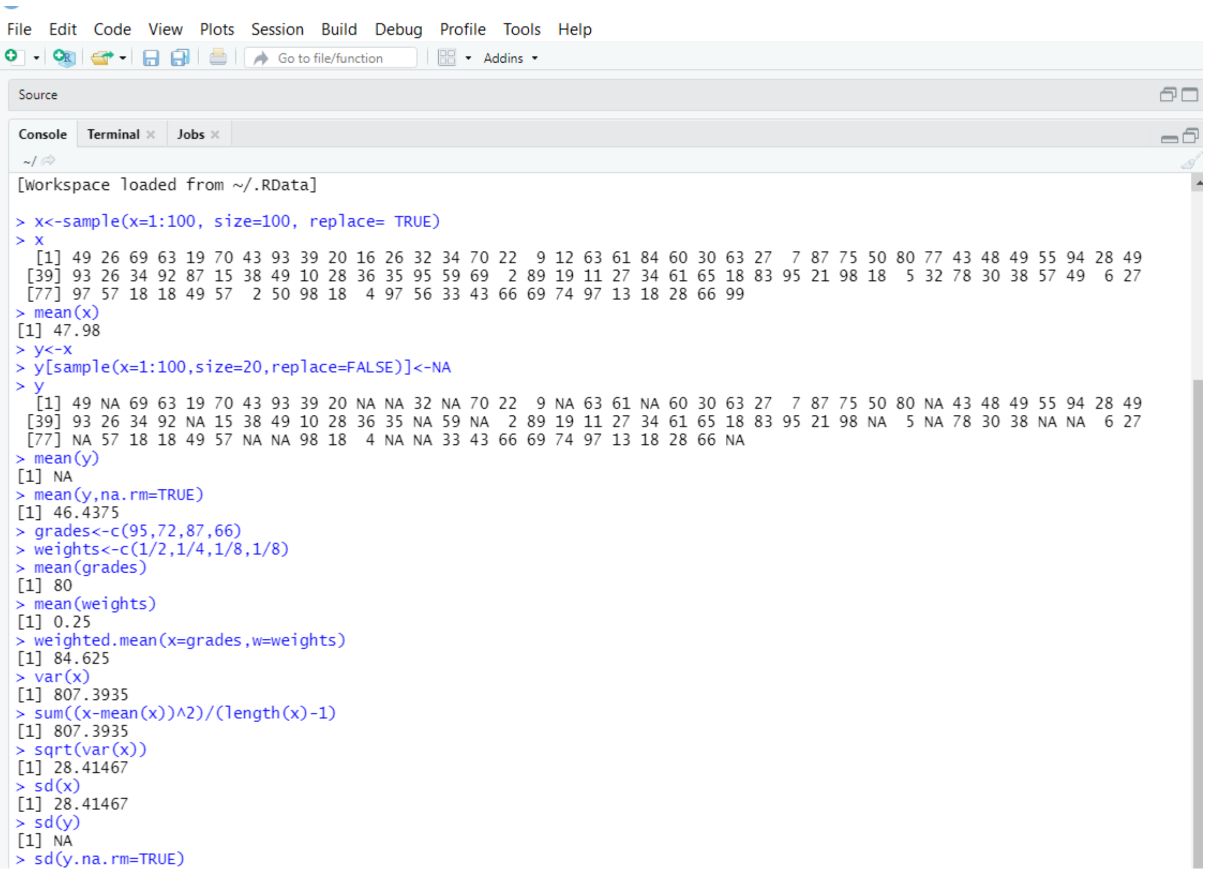
```
> sum((x-mean(x))^2)/(length(x)-1)
[1] 807.3935
> sqrt(var(x))
[1] 28.41467
> sd(x)
[1] 28.41467
> sd(y)
[1] NA
> sd(y.na.rm=TRUE)
> sd(y,na.rm=TRUE)
[1] 27.356
> min(x)
[1] 2
> max(x)
[1] 99
> mean(x)
[1] 47.98
> min(y)
[1] NA
> max(y)
[1] NA
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   2.00   26.00   48.50   47.98   69.00   99.00 
> summary(y)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   2.00   25.00   43.00   46.44   66.00   98.00      20 
> quantile(x,probs=c(0.25,0.75))
25% 75% 
 26  69 
> quantile(y,probs=c(0.25,0.75))
Error in quantile.default(y, probs = c(0.25, 0.75)) : 
  missing values and NaN's not allowed if 'na.rm' is FALSE
> quantile(y, probs=c(0.25,0.75),na.rm=TRUE)
```

```
25% 75%
 25  66
```

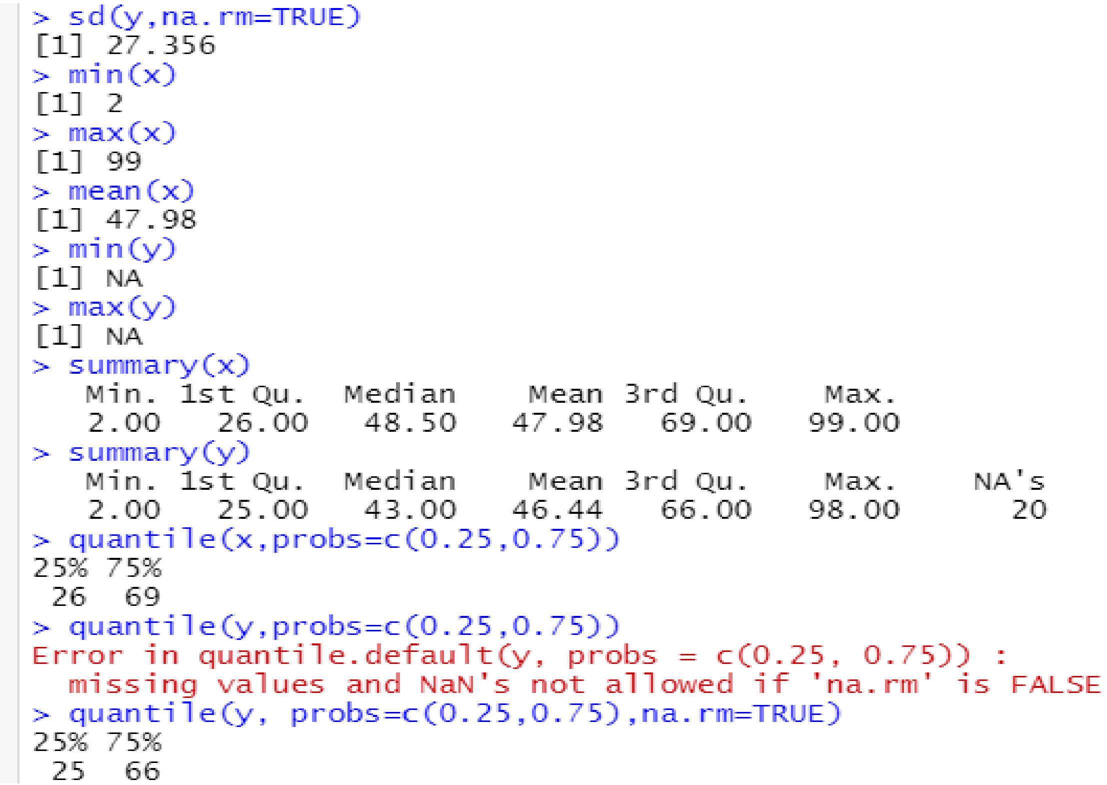
EDITOR CODE:



```r
x<-sample(x=1:100, size=100, replace= TRUE)
x
mean(x)
y<-x
y[sample(x=1:100,size=20,replace=FALSE)]<-NA
y
mean(y)
mean(y,na.rm=TRUE)
grades<-c(95,72,87,66)
weights<-c(1/2,1/4,1/8,1/8)
mean(grades)
mean(weights)
weighted.mean(x=grades,w=weights)
var(x)
sum((x-mean(x))^2)/(length(x)-1)
sqrt(var(x))
sd(x)
sd(y)
sd(y,na.rm=TRUE)
min(x)
max(x)
mean(x)
min(y)
max(y)
summary(x)
summary(y)
quantile(x,probs=c(0.25,0.75))
quantile(y,probs=c(0.25,0.75))
quantile(y, probs=c(0.25,0.75),na.rm=TRUE)

```


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
~/

```
[Workspace loaded from ~/.RData]

> x<-sample(x=1:100, size=100, replace= TRUE)
> x
  [1] 49 26 69 63 19 70 43 93 39 20 16 26 32 34 70 22  9 12 63 61 84 60 30 63 27  7 87 75 50 80 77 43 48 49 55 94 28 49
 [39] 93 26 34 92 87 15 38 49 10 28 36 35 95 59 69  2 89 19 11 27 34 61 65 18 83 95 21 98 18  5 32 78 30 38 57 49  6 27
 [77] 97 57 18 18 49 57  2 50 98 18  4 97 56 33 43 66 69 74 97 13 18 28 66 99
> mean(x)
[1] 47.98
> y<-x
> y[sample(x=1:100,size=20,replace=FALSE)]<-NA
> y
  [1] 49 NA 69 63 19 70 43 93 39 20 NA NA 32 NA 70 22  9 NA 63 61 NA 60 30 63 27  7 87 75 50 80 NA 43 48 49 55 94 28 49
 [39] 93 26 34 92 NA 15 38 49 10 28 36 35 NA 59 NA  2 89 19 11 27 34 61 65 18 83 95 21 98 NA  5 NA 78 30 38 NA NA  6 27
 [77] NA 57 18 18 49 57 NA NA 98 18  4 NA NA 33 43 66 69 74 97 13 18 28 66 NA
> mean(y)
[1] NA
> mean(y,na.rm=TRUE)
[1] 46.4375
> grades<-c(95,72,87,66)
> weights<-c(1/2,1/4,1/8,1/8)
> mean(grades)
[1] 80
> mean(weights)
[1] 0.25
> weighted.mean(x=grades,w=weights)
[1] 84.625
> var(x)
[1] 807.3935
> sum((x-mean(x))^2)/(length(x)-1)
[1] 807.3935
> sqrt(var(x))
[1] 28.41467
> sd(x)
[1] 28.41467
> sd(y)
[1] NA
> sd(y.na.rm=TRUE)
```



```
> sd(y,na.rm=TRUE)
[1] 27.356
> min(x)
[1] 2
> max(x)
[1] 99
> mean(x)
[1] 47.98
> min(y)
[1] NA
> max(y)
[1] NA
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   2.00   26.00   48.50   47.98   69.00   99.00 
> summary(y)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   2.00   25.00   43.00   46.44   66.00   98.00      20 
> quantile(x,probs=c(0.25,0.75))
25% 75% 
 26  69 
> quantile(y,probs=c(0.25,0.75))
Error in quantile.default(y, probs = c(0.25, 0.75)) : 
  missing values and NaN's not allowed if 'na.rm' is FALSE
> quantile(y, probs=c(0.25,0.75),na.rm=TRUE)
25% 75% 
 25  66 
```

RESULT:

HENCE THE CODE IS EXECUTED SUCCESSFULLY.

R LAB-11

TASK-01:

Implement statistical distribution concept in r studio

PROGRAM:

```
dnorm (x=0, mean=0, sd=1)
dnorm(x=0)
dnorm (x=10, mean=20, sd=5)
x<-seq (-4,4, length=100)
x
y<-dnorm(x)
y
require(ggplot2)
plot (x, y, type = "l", lwd = 2, axes = FALSE, xlab = "", ylab = "")
axis (1, at = -3:3, labels = c ("-3s", "-2s", "-1s", "mean", "1s", "2s", "3s"))
pnorm (74, mean=70, sd=2, lower. tail=FALSE)
pnorm (22, mean=30, sd=5)
pnorm (14, mean=13, sd=2)-pnorm (10, mean=13, sd=2)
qnorm (.99, mean=0, sd=1)
qnorm (.95)
qnorm (.10)
five <- rnorm (5, mean = 10, sd = 2)
five
narrowDistribution <- rnorm (1000, mean = 50, sd = 15)
wideDistribution <- rnorm (1000, mean = 50, sd = 25)
par (mfrow=c (1, 2))
hist (narrowDistribution, breaks=50, xlim=c (-50, 150))
hist (wideDistribution, breaks=50, xlim=c (-50, 150))
sample. range <- 50:150
iq. mean <- 100
iq.sd <- 15
iq. dist <- dnorm (sample. range, mean = iq. mean, sd = iq.sd)
iq.df <- data. frame ("IQ" = sample. range, "Density" = iq. dist)
```

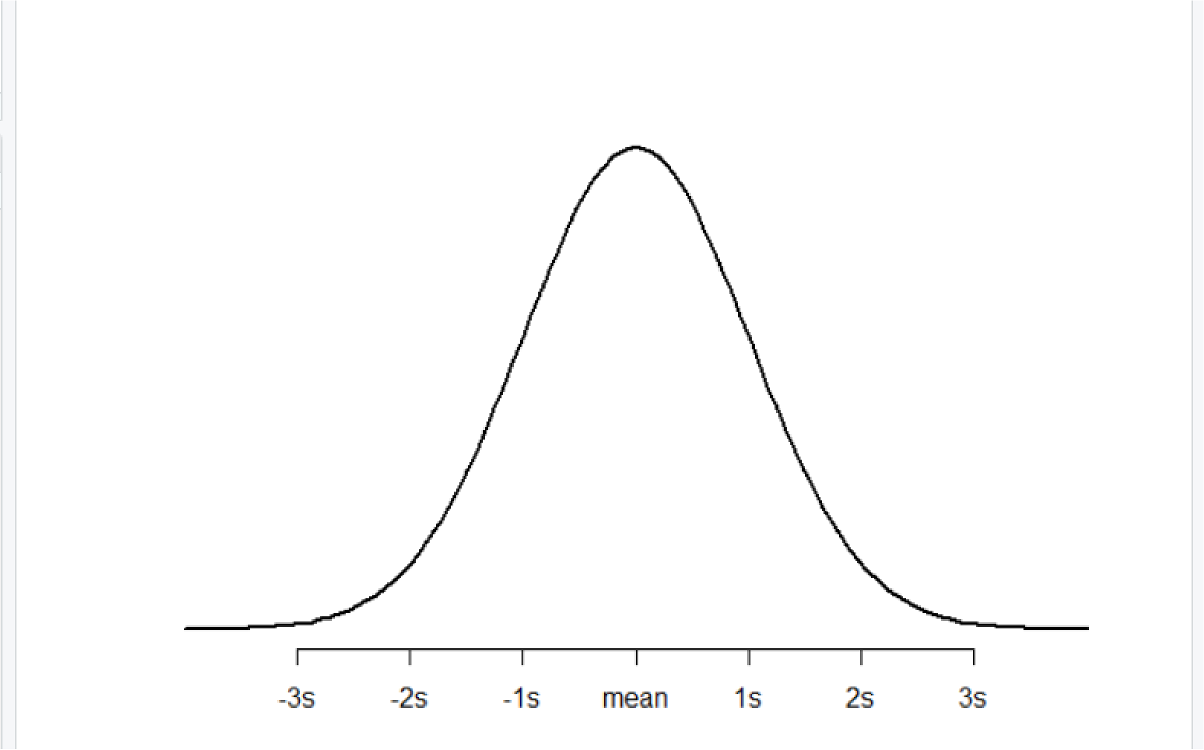
```
ggplot (iq.df, aes (x = IQ, y = Density)) + geom_point ()
pp <- function(x) {
  print (paste0(round (x * 100, 3), "%"))
}
pp(iq.df$Density[iq.df$IQ == 140])
pp(sum(iq.df$Density[iq.df$IQ >= 140]))
pp(sum(iq.df$Density[iq.df$IQ <= 90]))
cdf <- pnorm(sample.range, iq.mean, iq.sd)
iq.df <- cbind(iq.df, "CDF_LowerTail" = cdf)
ggplot(iq.df, aes(x = IQ, y = CDF_LowerTail)) + geom_point()
prob.range <- seq(0, 1, 0.001)
icdf.df <- data.frame("Probability" = prob.range, "IQ" = qnorm(prob.range, iq.mean, iq.sd))
ggplot(icdf.df, aes(x = Probability, y = IQ)) + geom_point()
set.seed(1)
n.samples <- c(100, 1000, 10000)
my.df <- do.call(rbind, lapply(n.samples, function(x) data.frame("SampleSize" = x, "IQ" = rnorm(x, iq.mean, iq.sd))))
ggplot() + geom_histogram(data = my.df, aes(x = IQ)) + facet_wrap(.~SampleSize, scales = "free_y")
my.sample <- sample(iq.df$IQ, 100, prob = iq.df$Density, replace = TRUE)
my.sample.df <- data.frame("IQ" = my.sample)
ggplot(my.sample.df, aes(x = IQ)) + geom_histogram()
```

OUTPUT:

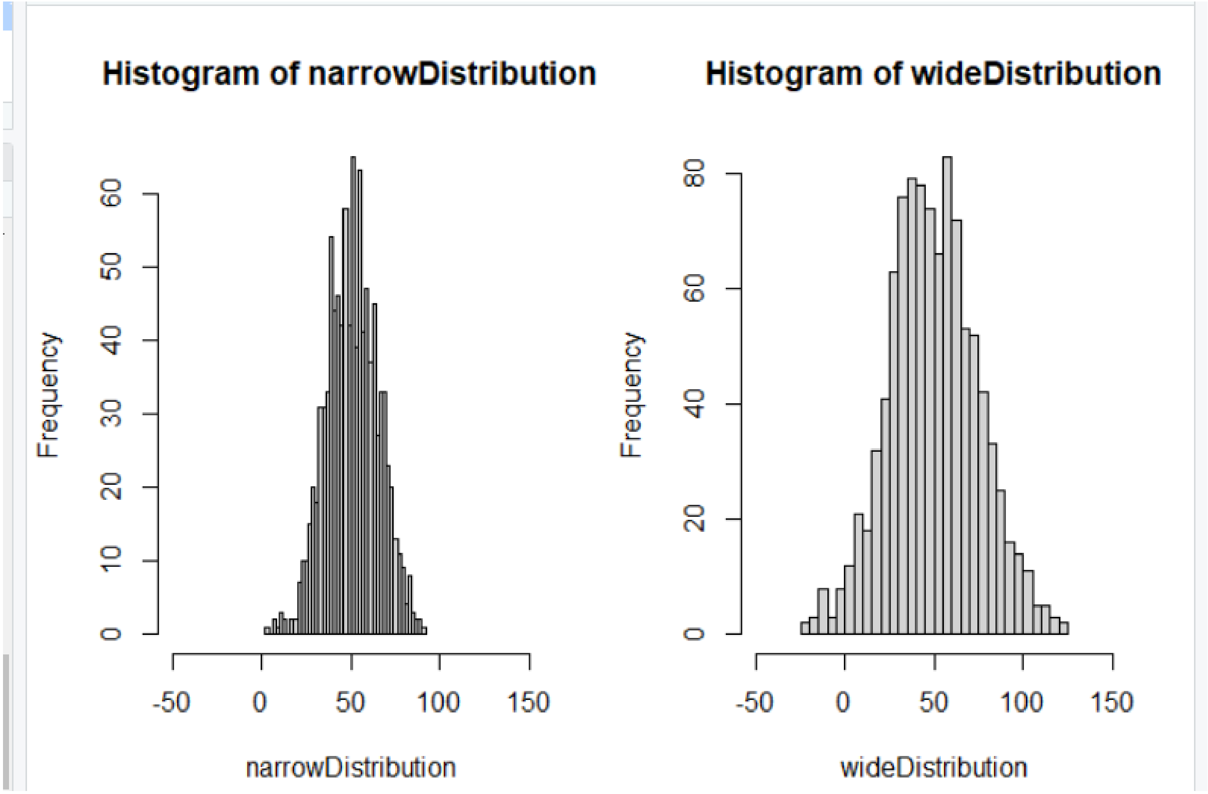
```
> dnorm(x=0,mean=0,sd=1)
[1] 0.3989423
> dnorm(x=0)
[1] 0.3989423
> dnorm(x=10,mean=20,sd=5)
[1] 0.01079819
> x<-seq(-4,4,length=100)
> x
  [1] -4.00000000 -3.91919192 -3.83838384 -3.75757576 -3.67676768 -3.59595960 -3.51515152 -3.43434343 -3.35353535
```

```
 [10]  -3.27272727  -3.19191919  -3.11111111  -3.03030303  -2.94949495  -2.86868687  -
2.78787879 -2.70707071 -2.62626263
 [19]  -2.54545455  -2.46464646  -2.38383838  -2.30303030  -2.22222222  -2.14141414  -
2.06060606 -1.97979798 -1.89898990
 [28]  -1.81818182  -1.73737374  -1.65656566  -1.57575758  -1.49494949  -1.41414141  -
1.33333333 -1.25252525 -1.17171717
 [37]  -1.09090909  -1.01010101  -0.92929293  -0.84848485  -0.76767677  -0.68686869  -
0.60606061 -0.52525253 -0.44444444
 [46]  -0.36363636  -0.28282828  -0.20202020  -0.12121212  -0.04040404   0.04040404
0.12121212  0.20202020  0.28282828
 [55]   0.36363636   0.44444444   0.52525253   0.60606061   0.68686869   0.76767677
0.84848485  0.92929293  1.01010101
 [64]   1.09090909   1.17171717   1.25252525   1.33333333   1.41414141   1.49494949
1.57575758  1.65656566  1.73737374
 [73]   1.81818182   1.89898990   1.97979798   2.06060606   2.14141414   2.22222222
2.30303030  2.38383838  2.46464646
 [82]   2.54545455   2.62626263   2.70707071   2.78787879   2.86868687   2.94949495
3.03030303  3.11111111  3.19191919
 [91]   3.27272727   3.35353535   3.43434343   3.51515152   3.59595960   3.67676768
3.75757576  3.83838384  3.91919192
[100]  4.00000000
> y<-dnorm(x)
> y
  [1] 0.0001338302 0.0001842953 0.0002521381 0.0003427099 0.0004627846 0.0006208623
0.0008275148 0.0010957722
  [9] 0.0014415473 0.0018840898 0.0024464615 0.0031560163 0.0040448664 0.0051503080
0.0065151783 0.0081881065
 [17] 0.0102236211 0.0126820683 0.0156292995 0.0191360817 0.0232771927 0.0281301641
0.0337736510 0.0402854146
 [25] 0.0477399263 0.0562056185 0.0657418315 0.0763955298 0.0881978860 0.1011608535
0.1152738702 0.1305008512
 [33] 0.1467776382 0.1640100747 0.1820728700 0.2008093962 0.2200325354 0.2395266587
0.2590507715 0.2783428081
```

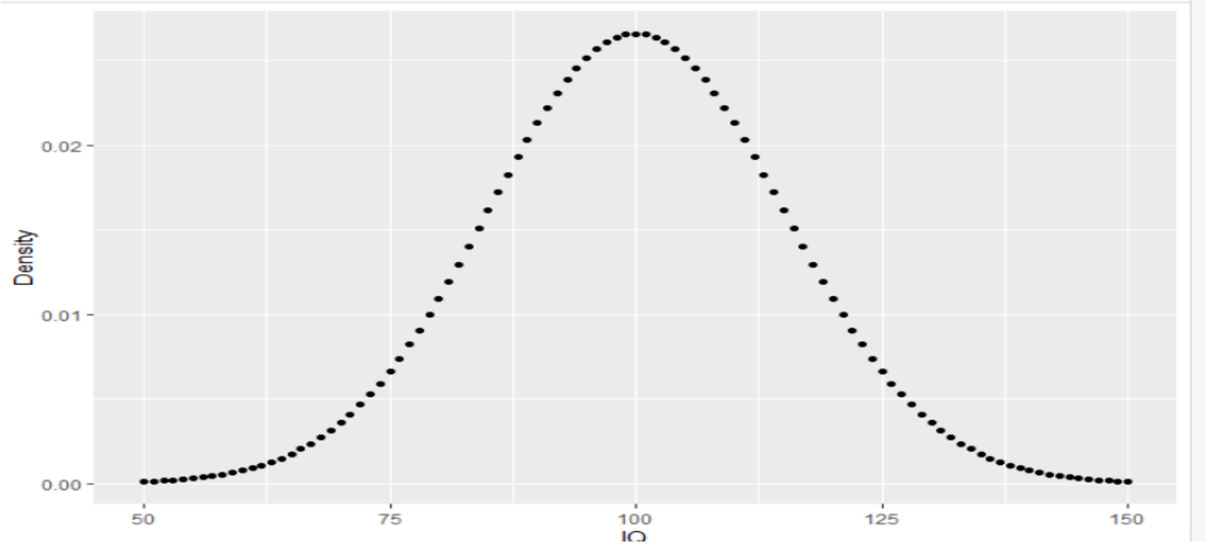
```
 [41] 0.2971250031 0.3151102096 0.3320089800 0.3475371752 0.3614238299 0.3734189738
0.3833010942 0.3908839312
 [49] 0.3960223134 0.3986167793 0.3986167793 0.3960223134 0.3908839312 0.3833010942
0.3734189738 0.3614238299
 [57] 0.3475371752 0.3320089800 0.3151102096 0.2971250031 0.2783428081 0.2590507715
0.2395266587 0.2200325354
 [65] 0.2008093962 0.1820728700 0.1640100747 0.1467776382 0.1305008512 0.1152738702
0.1011608535 0.0881978860
 [73] 0.0763955298 0.0657418315 0.0562056185 0.0477399263 0.0402854146 0.0337736510
0.0281301641 0.0232771927
 [81] 0.0191360817 0.0156292995 0.0126820683 0.0102236211 0.0081881065 0.0065151783
0.0051503080 0.0040448664
 [89] 0.0031560163 0.0024464615 0.0018840898 0.0014415473 0.0010957722 0.0008275148
0.0006208623 0.0004627846
 [97] 0.0003427099 0.0002521381 0.0001842953 0.0001338302
> require(ggplot2)
Loading required package: ggplot2

Attaching package: ‘ggplot2’

The following object is masked _by_ ‘.GlobalEnv’:

    economics

> plot(x,y, type = "l", lwd = 2, axes = FALSE, xlab = "", ylab = "")
> axis(1, at = -3:3, labels = c("-3s", "-2s", "-1s", "mean", "1s", "2s", "3s"))
```



```
> pnorm(74,mean=70,sd=2,lower.tail=FALSE)
[1] 0.02275013
> pnorm(22,mean=30,sd=5)
[1] 0.05479929
> pnorm(14,mean=13,sd=2)-pnorm(10,mean=13,sd=2)
[1] 0.6246553
> qnorm(.99,mean=0,sd=1)
[1] 2.326348
> qnorm(.95)
[1] 1.644854
> qnorm(.10)
[1] -1.281552
> five <- rnorm(5, mean = 10, sd = 2)
> five
[1] 10.472830 12.021374  7.595072  9.509793  8.212973
> narrowDistribution <- rnorm(1000, mean = 50, sd = 15)
> wideDistribution <- rnorm(1000, mean = 50, sd = 25)
> par(mfrow=c(1, 2))
> hist(narrowDistribution, breaks=50, xlim=c(-50, 150))
> hist(wideDistribution, breaks=50, xlim=c(-50, 150))
```

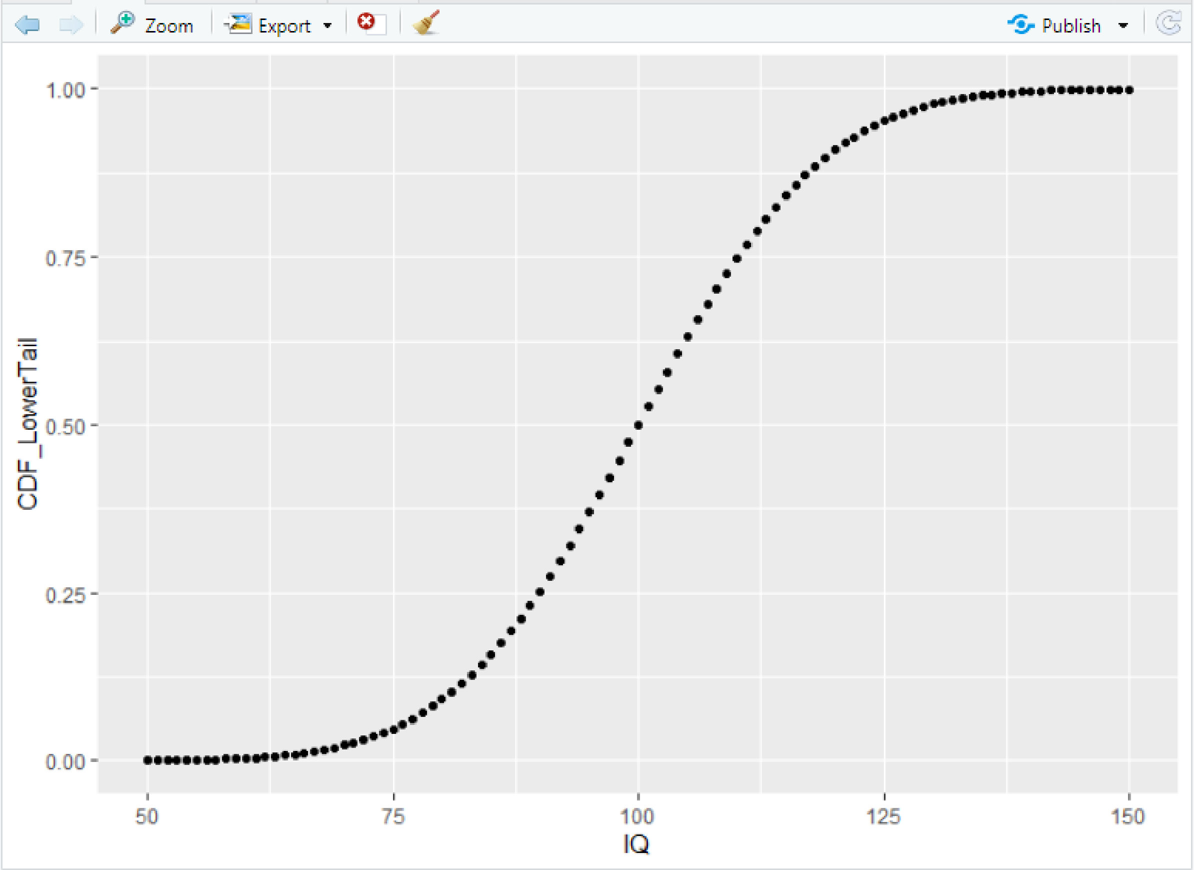


```
> sample.range <- 50:150
> iq.mean <- 100
> iq.sd <- 15
> iq.dist <- dnorm(sample.range, mean = iq.mean, sd = iq.sd)
> iq.df <- data.frame("IQ" = sample.range, "Density" = iq.dist)
> ggplot(iq.df, aes(x = IQ, y = Density)) + geom_point()
```

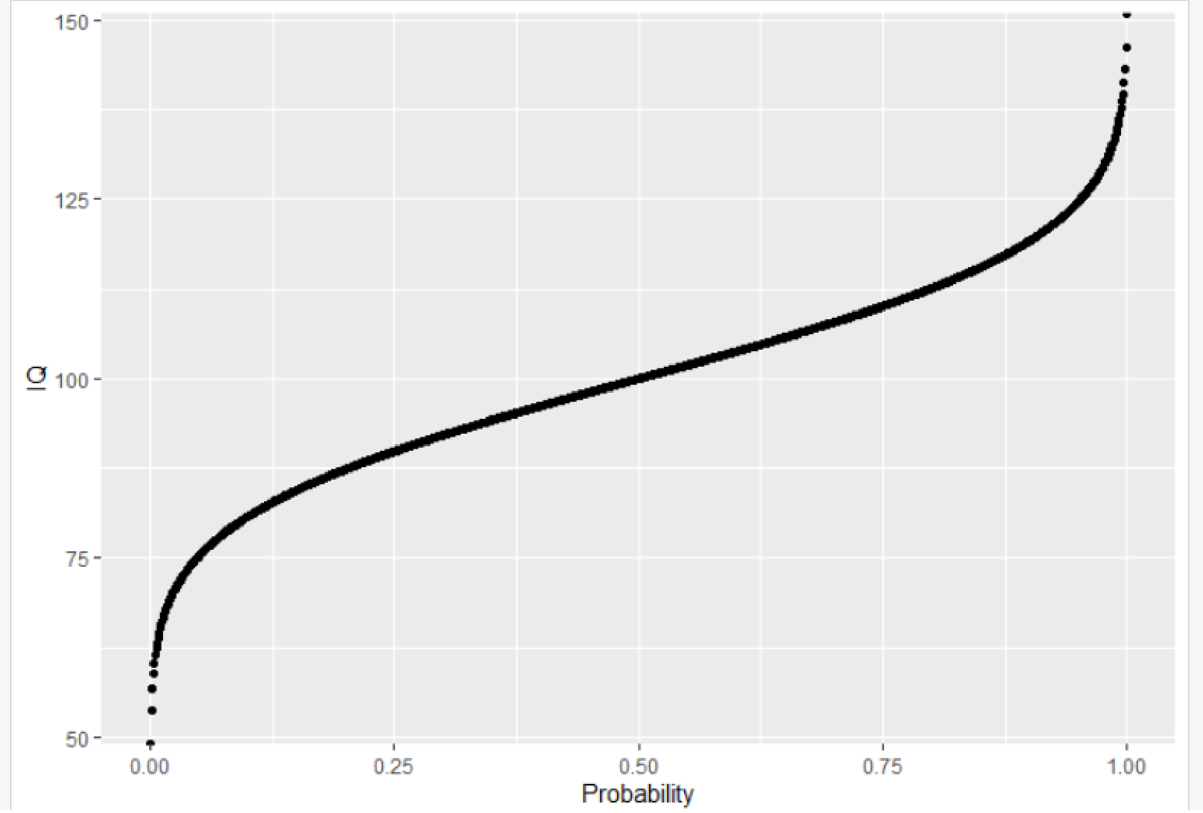


```
> pp <- function(x) {
+   print(paste0(round(x * 100, 3), "%"))
+ }
```

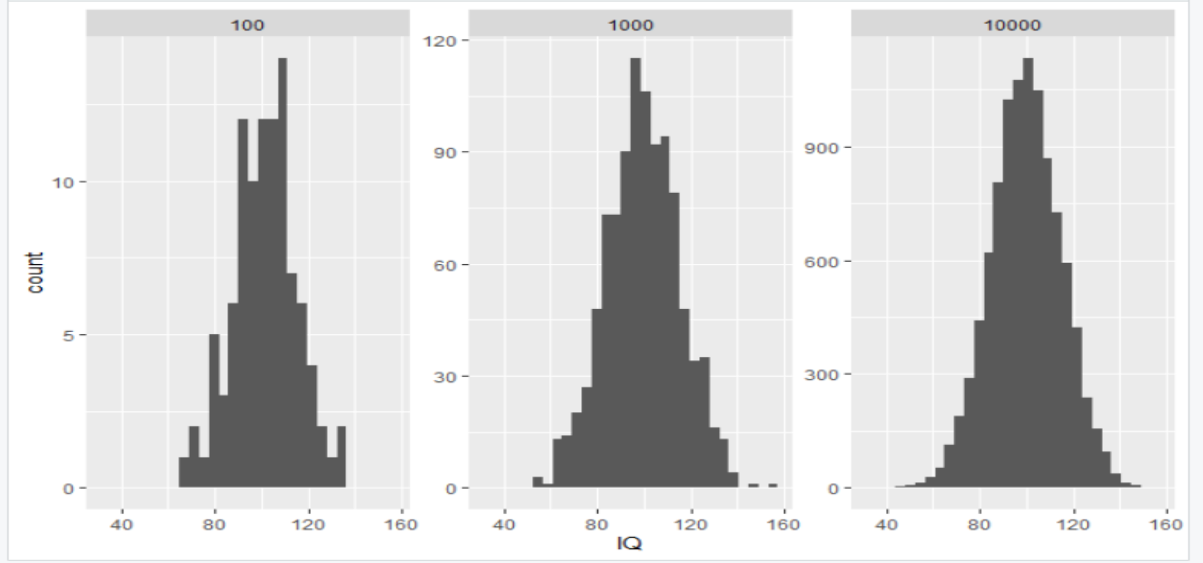
```
> pp(iq.df$Density[iq.df$IQ == 140])
[1] "0.076%"
> pp(sum(iq.df$Density[iq.df$IQ >= 140]))
[1] "0.384%"
> pp(sum(iq.df$Density[iq.df$IQ <= 90]))
[1] "26.284%"
> cdf <- pnorm(sample.range, iq.mean, iq.sd)
> iq.df <- cbind(iq.df, "CDF_LowerTail" = cdf)
> ggplot(iq.df, aes(x = IQ, y = CDF_LowerTail)) + geom_point()
```



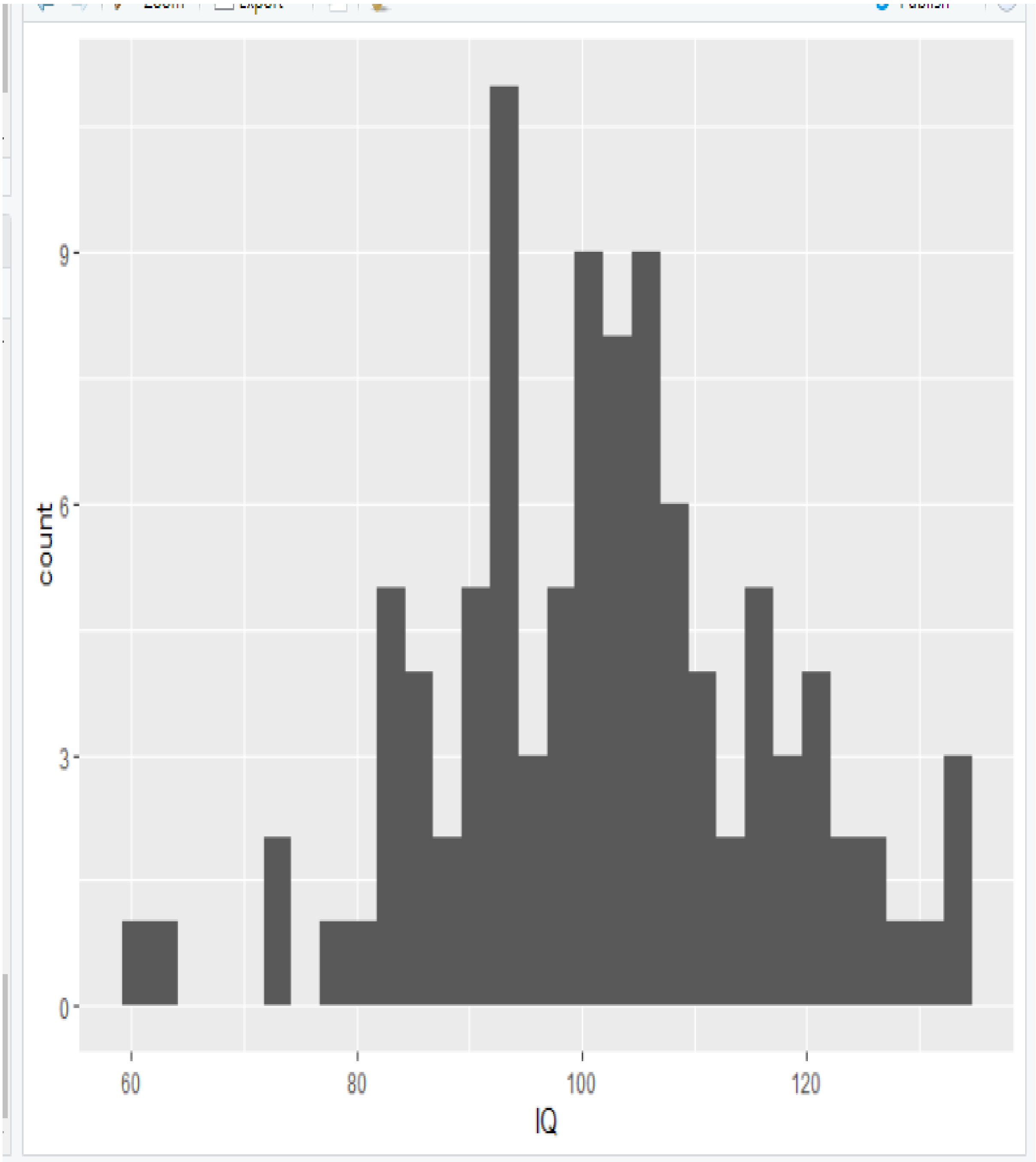
```
> prob.range <- seq(0, 1, 0.001)
> icdf.df <- data.frame("Probability" = prob.range, "IQ" = qnorm(prob.range, iq.mean, iq.sd))
> ggplot(icdf.df, aes(x = Probability, y = IQ)) + geom_point()
```



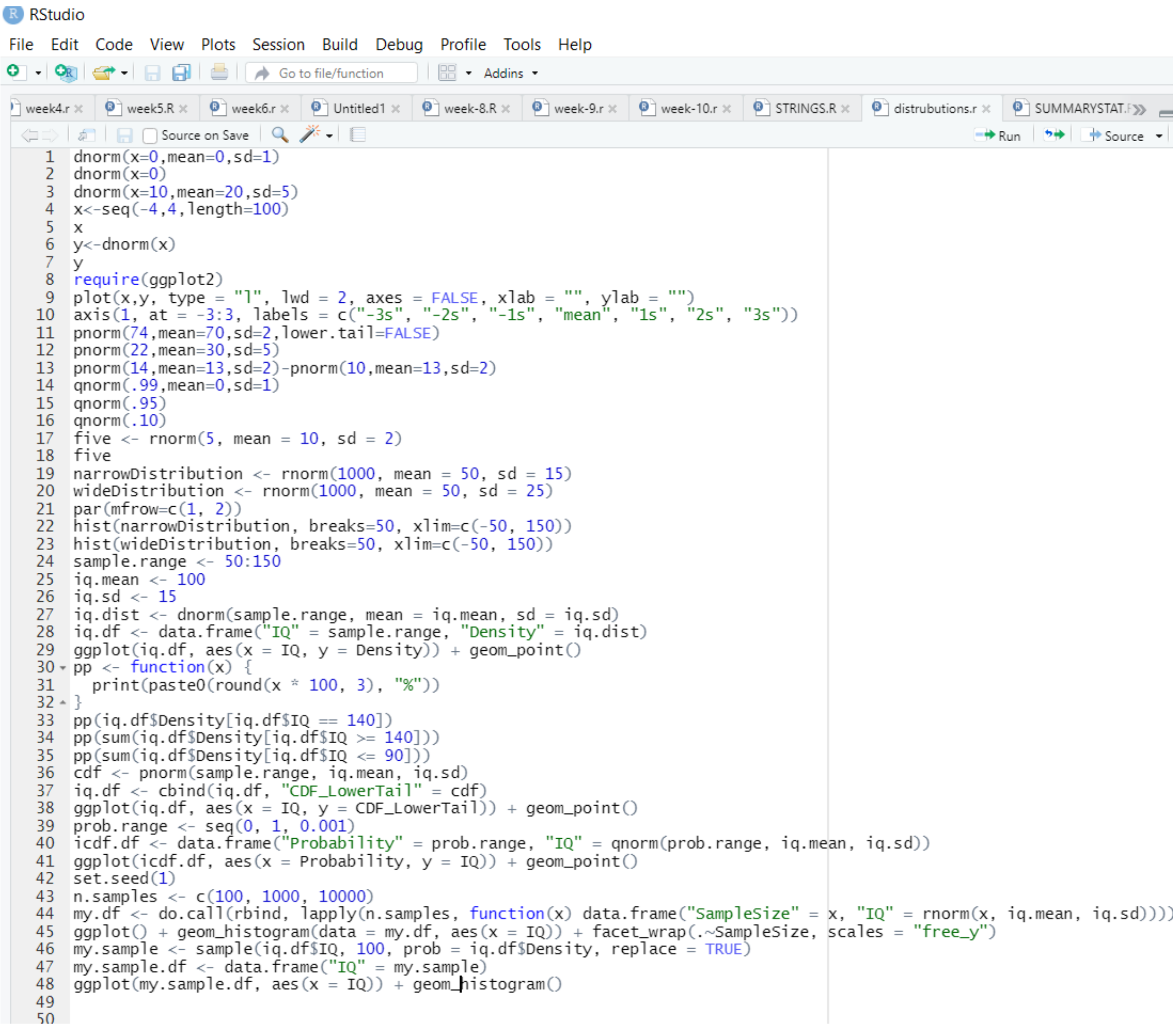
```
> set.seed(1)
> n.samples <- c(100, 1000, 10000)
> my.df <- do.call(rbind, lapply(n.samples, function(x) data.frame("SampleSize" = x, "IQ" = rnorm(x, iq.mean, iq.sd))))
> ggplot() + geom_histogram(data = my.df, aes(x = IQ)) + facet_wrap(.~SampleSize, scales = "free_y")
```



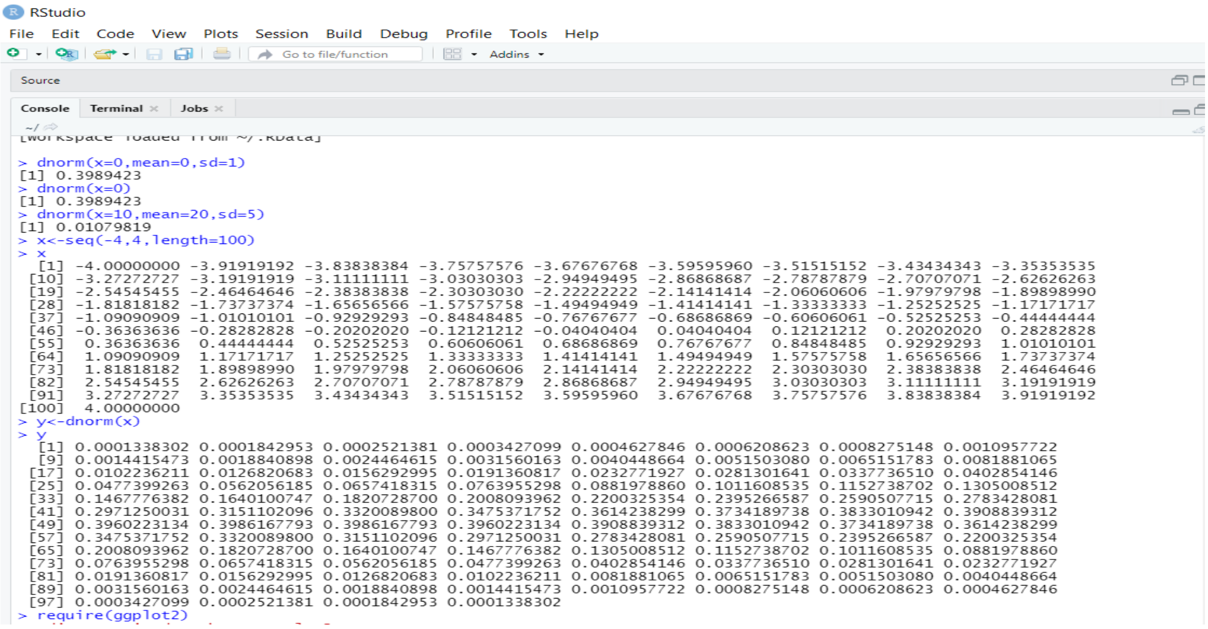
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> my.sample <- sample(iq.df$IQ, 100, prob = iq.df$Density, replace = TRUE)
> my.sample.df <- data.frame("IQ" = my.sample)
> ggplot(my.sample.df, aes(x = IQ)) + geom_histogram()
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



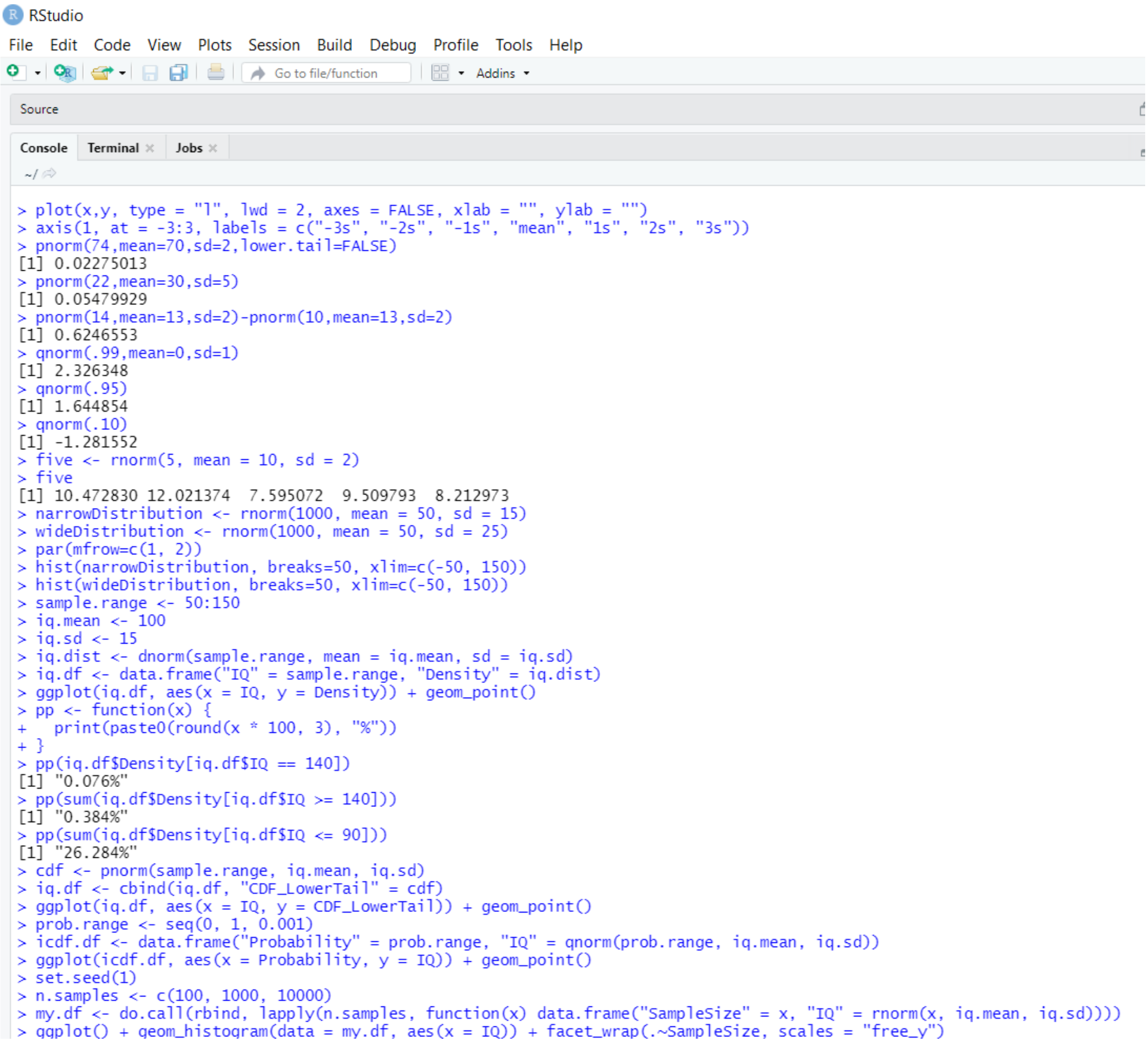
EDITOR CODE:



```r
dnorm(x=0,mean=0,sd=1)
dnorm(x=0)
dnorm(x=10,mean=20,sd=5)
x<-seq(-4,4,length=100)
x
y<-dnorm(x)
y
require(ggplot2)
plot(x,y, type = "l", lwd = 2, axes = FALSE, xlab = "", ylab = "")
axis(1, at = -3:3, labels = c("-3s", "-2s", "-1s", "mean", "1s", "2s", "3s"))
pnorm(74,mean=70,sd=2,lower.tail=FALSE)
pnorm(22,mean=30,sd=5)
pnorm(14,mean=13,sd=2)-pnorm(10,mean=13,sd=2)
qnorm(.99,mean=0,sd=1)
qnorm(.95)
qnorm(.10)
five <- rnorm(5, mean = 10, sd = 2)
five
narrowDistribution <- rnorm(1000, mean = 50, sd = 15)
wideDistribution <- rnorm(1000, mean = 50, sd = 25)
par(mfrow=c(1, 2))
hist(narrowDistribution, breaks=50, xlim=c(-50, 150))
hist(wideDistribution, breaks=50, xlim=c(-50, 150))
sample.range <- 50:150
iq.mean <- 100
iq.sd <- 15
iq.dist <- dnorm(sample.range, mean = iq.mean, sd = iq.sd)
iq.df <- data.frame("IQ" = sample.range, "Density" = iq.dist)
ggplot(iq.df, aes(x = IQ, y = Density)) + geom_point()
pp <- function(x) {
  print(paste0(round(x * 100, 3), "%"))
}
pp(iq.df$Density[iq.df$IQ == 140])
pp(sum(iq.df$Density[iq.df$IQ >= 140]))
pp(sum(iq.df$Density[iq.df$IQ <= 90]))
cdf <- pnorm(sample.range, iq.mean, iq.sd)
iq.df <- cbind(iq.df, "CDF_LowerTail" = cdf)
ggplot(iq.df, aes(x = IQ, y = CDF_LowerTail)) + geom_point()
prob.range <- seq(0, 1, 0.001)
icdf.df <- data.frame("Probability" = prob.range, "IQ" = qnorm(prob.range, iq.mean, iq.sd))
ggplot(icdf.df, aes(x = Probability, y = IQ)) + geom_point()
set.seed(1)
n.samples <- c(100, 1000, 10000)
my.df <- do.call(rbind, lapply(n.samples, function(x) data.frame("SampleSize" = x, "IQ" = rnorm(x, iq.mean, iq.sd))))
ggplot() + geom_histogram(data = my.df, aes(x = IQ)) + facet_wrap(.~SampleSize, scales = "free_y")
my.sample <- sample(iq.df$IQ, 100, prob = iq.df$Density, replace = TRUE)
my.sample.df <- data.frame("IQ" = my.sample)
ggplot(my.sample.df, aes(x = IQ)) + geom_histogram()
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
~/

```
[Workspace loaded from ~/.RData]

> dnorm(x=0,mean=0,sd=1)
[1] 0.3989423
> dnorm(x=0)
[1] 0.3989423
> dnorm(x=10,mean=20,sd=5)
[1] 0.01079819
> x<-seq(-4,4,length=100)
> x
  [1] -4.00000000 -3.91919192 -3.83838384 -3.75757576 -3.67676768 -3.59595960 -3.51515152 -3.43434343 -3.35353535
 [10] -3.27272727 -3.19191919 -3.11111111 -3.03030303 -2.94949495 -2.86868687 -2.78787879 -2.70707071 -2.62626263
 [19] -2.54545455 -2.46464646 -2.38383838 -2.30303030 -2.22222222 -2.14141414 -2.06060606 -1.97979798 -1.89898990
 [28] -1.81818182 -1.73737374 -1.65656566 -1.57575758 -1.49494949 -1.41414141 -1.33333333 -1.25252525 -1.17171717
 [37] -1.09090909 -1.01010101 -0.92929293 -0.84848485 -0.76767677 -0.68686869 -0.60606061 -0.52525253 -0.44444444
 [46] -0.36363636 -0.28282828 -0.20202020 -0.12121212 -0.04040404  0.04040404  0.12121212  0.20202020  0.28282828
 [55]  0.36363636  0.44444444  0.52525253  0.60606061  0.68686869  0.76767677  0.84848485  0.92929293  1.01010101
 [64]  1.09090909  1.17171717  1.25252525  1.33333333  1.41414141  1.49494949  1.57575758  1.65656566  1.73737374
 [73]  1.81818182  1.89898990  1.97979798  2.06060606  2.14141414  2.22222222  2.30303030  2.38383838  2.46464646
 [82]  2.54545455  2.62626263  2.70707071  2.78787879  2.86868687  2.94949495  3.03030303  3.11111111  3.19191919
 [91]  3.27272727  3.35353535  3.43434343  3.51515152  3.59595960  3.67676768  3.75757576  3.83838384  3.91919192
[100]  4.00000000
> y<-dnorm(x)
> y
  [1] 0.0001338302 0.0001842953 0.0002521381 0.0003427099 0.0004627846 0.0006208623 0.0008275148 0.0010957722
  [9] 0.0014415473 0.0018840898 0.0024464615 0.0031560163 0.0040448664 0.0051503080 0.0065151783 0.0081881065
 [17] 0.0102236211 0.0126820683 0.0156292995 0.0191360817 0.0232771927 0.0281301641 0.0337736510 0.0402854146
 [25] 0.0477399263 0.0562056185 0.0657418315 0.0763955298 0.0881978860 0.1011608535 0.1152738702 0.1305008512
 [33] 0.1467776382 0.1640100747 0.1820728700 0.2008093962 0.2200325354 0.2395266587 0.2590507715 0.2783428081
 [41] 0.2971250031 0.3151102096 0.3320089800 0.3475371752 0.3614238299 0.3734189738 0.3833010942 0.3908839312
 [49] 0.3960223134 0.3986167793 0.3986167793 0.3960223134 0.3908839312 0.3833010942 0.3734189738 0.3614238299
 [57] 0.3475371752 0.3320089800 0.3151102096 0.2971250031 0.2783428081 0.2590507715 0.2395266587 0.2200325354
 [65] 0.2008093962 0.1820728700 0.1640100747 0.1467776382 0.1305008512 0.1152738702 0.1011608535 0.0881978860
 [73] 0.0763955298 0.0657418315 0.0562056185 0.0477399263 0.0402854146 0.0337736510 0.0281301641 0.0232771927
 [81] 0.0191360817 0.0156292995 0.0126820683 0.0102236211 0.0081881065 0.0065151783 0.0051503080 0.0040448664
 [89] 0.0031560163 0.0024464615 0.0018840898 0.0014415473 0.0010957722 0.0008275148 0.0006208623 0.0004627846
 [97] 0.0003427099 0.0002521381 0.0001842953 0.0001338302
> require(ggplot2)
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
~/

```
> plot(x,y, type = "l", lwd = 2, axes = FALSE, xlab = "", ylab = "")
> axis(1, at = -3:3, labels = c("-3s", "-2s", "-1s", "mean", "1s", "2s", "3s"))
> pnorm(74,mean=70,sd=2,lower.tail=FALSE)
[1] 0.02275013
> pnorm(22,mean=30,sd=5)
[1] 0.05479929
> pnorm(14,mean=13,sd=2)-pnorm(10,mean=13,sd=2)
[1] 0.6246553
> qnorm(.99,mean=0,sd=1)
[1] 2.326348
> qnorm(.95)
[1] 1.644854
> qnorm(.10)
[1] -1.281552
> five <- rnorm(5, mean = 10, sd = 2)
> five
[1] 10.472830 12.021374  7.595072  9.509793  8.212973
> narrowDistribution <- rnorm(1000, mean = 50, sd = 15)
> wideDistribution <- rnorm(1000, mean = 50, sd = 25)
> par(mfrow=c(1, 2))
> hist(narrowDistribution, breaks=50, xlim=c(-50, 150))
> hist(wideDistribution, breaks=50, xlim=c(-50, 150))
> sample.range <- 50:150
> iq.mean <- 100
> iq.sd <- 15
> iq.dist <- dnorm(sample.range, mean = iq.mean, sd = iq.sd)
> iq.df <- data.frame("IQ" = sample.range, "Density" = iq.dist)
> ggplot(iq.df, aes(x = IQ, y = Density)) + geom_point()
> pp <- function(x) {
+   print(paste0(round(x * 100, 3), "%"))
+ }
> pp(iq.df$Density[iq.df$IQ == 140])
[1] "0.076%"
> pp(sum(iq.df$Density[iq.df$IQ >= 140]))
[1] "0.384%"
> pp(sum(iq.df$Density[iq.df$IQ <= 90]))
[1] "26.284%"
> cdf <- pnorm(sample.range, iq.mean, iq.sd)
> iq.df <- cbind(iq.df, "CDF_LowerTail" = cdf)
> ggplot(iq.df, aes(x = IQ, y = CDF_LowerTail)) + geom_point()
> prob.range <- seq(0, 1, 0.001)
> icdf.df <- data.frame("Probability" = prob.range, "IQ" = qnorm(prob.range, iq.mean, iq.sd))
> ggplot(icdf.df, aes(x = Probability, y = IQ)) + geom_point()
> set.seed(1)
> n.samples <- c(100, 1000, 10000)
> my.df <- do.call(rbind, lapply(n.samples, function(x) data.frame("SampleSize" = x, "IQ" = rnorm(x, iq.mean, iq.sd))))
> ggplot() + geom_histogram(data = my.df, aes(x = IQ)) + facet_wrap(.~SampleSize, scales = "free_y")
```

RESULT:

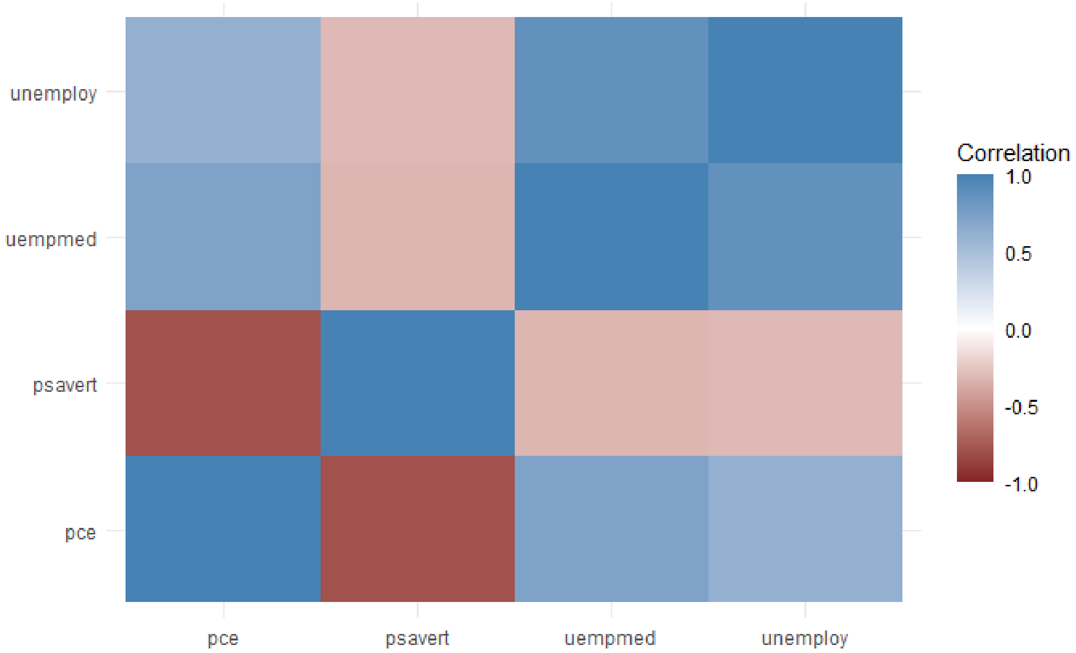
HENCE THE CODE IS EXECUTED SUCCESSSFULLY.

R LAB-12

TASK-01:

Correlation and Covariance:

```
> install.packages("ggplot2")
> require(ggplot2)
> head(economics)
# A tibble: 6 x 6
  date        pce    pop psavert uempmed unemploy
  <date>     <dbl>  <dbl>   <dbl>   <dbl>    <dbl>
1 1967-07-01  507. 198712    12.6     4.5     2944
2 1967-08-01  510. 198911    12.6     4.7     2945
3 1967-09-01  516. 199113    11.9     4.6     2958
4 1967-10-01  512. 199311    12.9     4.9     3143
5 1967-11-01  517. 199498    12.8     4.7     3066
6 1967-12-01  525. 199657    11.8     4.8     3018
> cor(economics$pce,economics$psavert)
[1] -0.7928546
> x<-economics$pce-mean(economics$pce)
> y<-economics$psavert-mean(economics$psavert)
> nMinusone<-(nrow(economics)-1)
> xSD<-sd(economics$pce)
> ySD<-sd(economics$psavert)
> sum(x*y)/(nMinusone*xSD*ySD)
[1] -0.7928546
> cor(economics[,c(2,4:6)])
                pce    psavert    uempmed   unemploy
pce       1.0000000 -0.7928546  0.7269616  0.6145176
psavert  -0.7928546  1.0000000 -0.3251377 -0.3093769
uempmed   0.7269616 -0.3251377  1.0000000  0.8693097
unemploy  0.6145176 -0.3093769  0.8693097  1.0000000
> install.packages("reshape2")
> require(reshape2)
> require(scales)
> econCor<-cor(economics[,c(2,4:6)])
> econMelt<-melt(econCor,varnames=c("x","y"),value.name="Correlation")
> econMelt<-econMelt[order(econMelt$Correlation),]
> econMelt
         x        y Correlation
2   psavert      pce  -0.7928546
5       pce  psavert  -0.7928546
7   uempmed  psavert  -0.3251377
10  psavert  uempmed  -0.3251377
8  unemploy  psavert  -0.3093769
14  psavert unemploy  -0.3093769
4  unemploy      pce   0.6145176
13      pce unemploy   0.6145176
3   uempmed      pce   0.7269616
9       pce  uempmed   0.7269616
12 unemploy  uempmed   0.8693097
15  uempmed unemploy   0.8693097
1       pce      pce   1.0000000
6   psavert  psavert   1.0000000
11  uempmed  uempmed   1.0000000
16 unemploy unemploy   1.0000000

>ggplot(econMelt,aes(x=x,y=y))+geom_tile(aes(fill=Correlation))+scale_fill_gradient2(low=
muted("red"),mid="white",high="steelblue",guide=guide_colorbar(ticks=FALSE,barheight=
10),limits=c(-1,1))+theme_minimal()+labs(x=NULL,y=NULL)
```

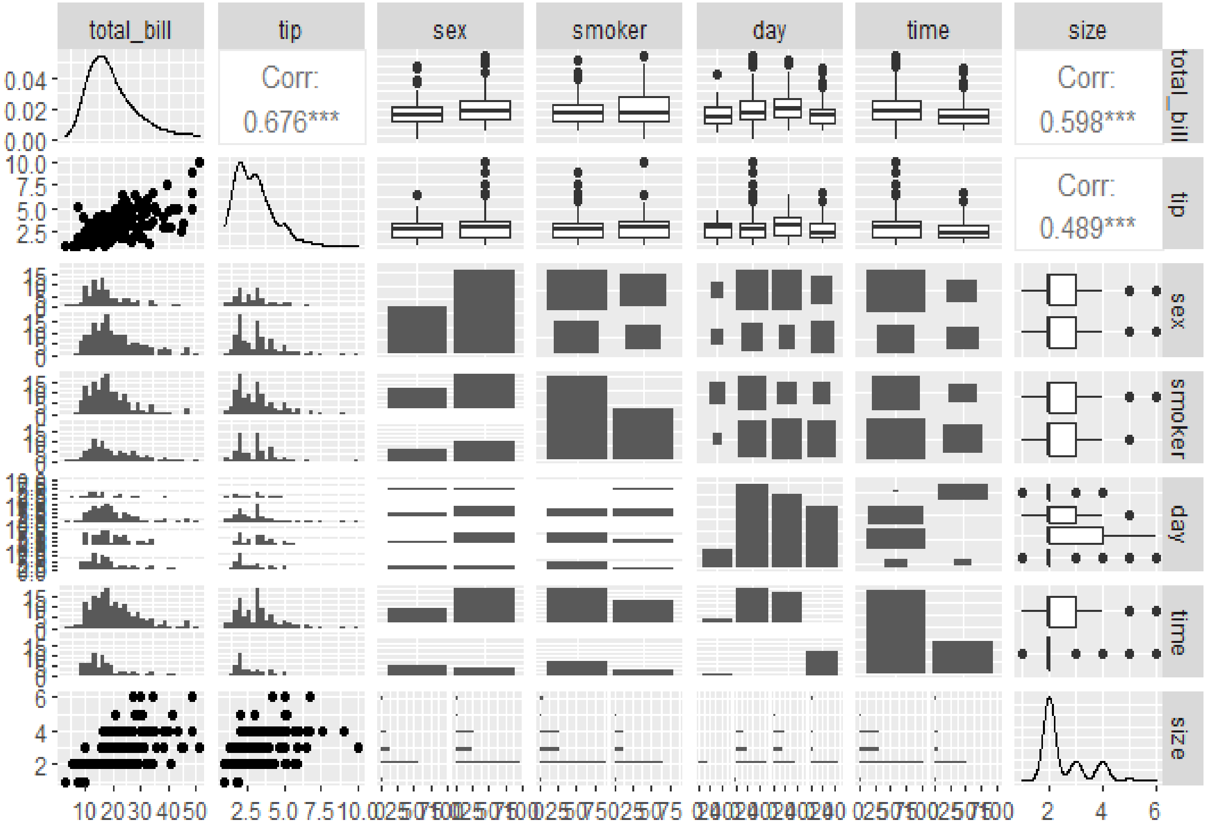


```
> m<-c(9,9,NA,3,NA,5,8,1,10,4)
> n<-c(2,NA,1,6,6,4,1,1,6,7)
> p<-c(8,4,3,9,10,NA,3,NA,9,9)
> q<-c(10,10,7,8,4,2,8,5,5,2)
> r<-c(1,9,7,6,5,6,2,7,9,10)
> theMat<-cbind(m,n,p,q,r)
> cor(theMat,use="everything")
   m  n  p          q          r
m  1 NA NA         NA         NA
n NA  1 NA         NA         NA
p NA NA  1         NA         NA
q NA NA NA  1.0000000 -0.4242958
r NA NA NA -0.4242958  1.0000000
> cor(theMat,use="all.obs")
Error in cor(theMat, use = "all.obs") : missing observations in cov/cor
> cor(theMat,use="complete.obs")
           m          n          p          q          r
m  1.0000000 -0.5228840 -0.2893527  0.2974398 -0.3459470
n -0.5228840  1.0000000  0.8090195 -0.7448453  0.9350718
p -0.2893527  0.8090195  1.0000000 -0.3613720  0.6221470
q  0.2974398 -0.7448453 -0.3613720  1.0000000 -0.9059384
```

```
r -0.3459470  0.9350718  0.6221470 -0.9059384  1.0000000
> cor(theMat,use="na.or.complete")
           m          n          p          q          r
m  1.0000000 -0.5228840 -0.2893527  0.2974398 -0.3459470
n -0.5228840  1.0000000  0.8090195 -0.7448453  0.9350718
p -0.2893527  0.8090195  1.0000000 -0.3613720  0.6221470
q  0.2974398 -0.7448453 -0.3613720  1.0000000 -0.9059384
r -0.3459470  0.9350718  0.6221470 -0.9059384  1.0000000
> cor(theMat[c(1,4,7,9,10),])
           m          n          p          q          r
m  1.0000000 -0.5228840 -0.2893527  0.2974398 -0.3459470
n -0.5228840  1.0000000  0.8090195 -0.7448453  0.9350718
p -0.2893527  0.8090195  1.0000000 -0.3613720  0.6221470
q  0.2974398 -0.7448453 -0.3613720  1.0000000 -0.9059384
r -0.3459470  0.9350718  0.6221470 -0.9059384  1.0000000
> identical(cor(theMat,use="complete.obs"),cor(theMat[c(1,4,7,9,10),]))
[1] TRUE
> cor(theMat,use="pairwise.complete.obs")
            m           n          p          q          r
m  1.00000000 -0.02511812 -0.3965859  0.4622943 -0.2001722
n -0.02511812  1.00000000  0.8717389 -0.5070416  0.5332259
p -0.39658588  0.87173889  1.0000000 -0.5197292  0.1312506
q  0.46229434 -0.50704163 -0.5197292  1.0000000 -0.4242958
r -0.20017222  0.53322585  0.1312506 -0.4242958  1.0000000
> cor(theMat[,c("m","n")],use="complete.obs")
            m           n
m  1.00000000 -0.02511812
n -0.02511812  1.00000000
> cor(theMat[,c("m","p")],use="complete.obs")
           m          p
m  1.0000000 -0.3965859
p -0.3965859  1.0000000
> data(tips,package="reshape2")
> head(tips)
```

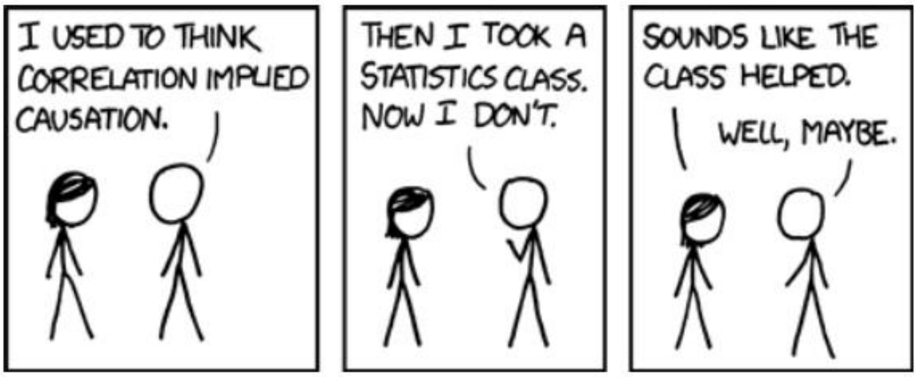
```
  total_bill  tip    sex smoker day   time size
1      16.99 1.01 Female     No Sun Dinner    2
2      10.34 1.66   Male     No Sun Dinner    3
3      21.01 3.50   Male     No Sun Dinner    3
4      23.68 3.31   Male     No Sun Dinner    2
5      24.59 3.61 Female     No Sun Dinner    4
6      25.29 4.71   Male     No Sun Dinner    4
> GGally::ggpairs(tips)
 plot: [3,1] [===================>---------------------------------------------] 31% est: 3s
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [3,2] [=====================>-------------------------------------------] 33% est: 4s
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [4,1] [=============================>-----------------------------------] 45% est:
3s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [4,2] [==============================>----------------------------------] 47%
est: 3s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [5,1] [=====================================>---------------------------]
59% est: 2s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [5,2] [======================================>--------------------------]
61% est: 2s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [6,1] [==============================================>-----------------
-] 73% est: 2s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [6,2] [================================================>--------------
--] 76% est: 2s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [7,3]
[===========================================================>-----]
92% est: 1s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [7,4]
[============================================================>----]
94% est: 0s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [7,5]
[=============================================================>---]
96% est: 0s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
plot:                                                                    [7,6]
[===============================================================>-
] 98% est: 0s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
> install.packages("RXKCD")
> require(RXKCD)
> getXKCD(which="552")
$month
[1] "3"

$num
[1] 552

$link
[1] ""

$year
[1] "2009"
```

```
$news
[1] ""

$safe_title
[1] "Correlation"

$transcript
[1] "[[A man is talking to a woman]]\nMan: I used to think correlation implied
causation.\nMan: Then I took a statistics class.  Now I don't.\nWoman: Sounds like the class
helped.\nMan: Well, maybe.\n{{Title text: Correlation doesn't imply causation, but it does
waggle its eyebrows suggestively and gesture furtively while mouthing 'look over there'.}}"

$alt
[1] "Correlation doesn't imply causation, but it does waggle its eyebrows suggestively and
gesture furtively while mouthing 'look over there'."

$img
[1] "https://imgs.xkcd.com/comics/correlation.png"

$title
[1] "Correlation"

$day
[1] "6"

attr(,"class")
[1] "rxkcd"
```


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

```
> cov(economics$pce,economics$psavert)
[1] -8359.069
> cov(economics[,c(2,4:6)])
                 pce      psavert      uempmed    unemploy
pce      12650851.944 -8359.069071 10618.386190 5774578.978
psavert     -8359.069     8.786360    -3.957847   -2422.805
uempmed     10618.386    -3.957847    16.864531    9431.652
unemploy  5774578.978 -2422.805358  9431.652268 6979948.309
>identical(cov(economics$pce,economics$psavert),cor(economics$pce,economics$psavert)*
sd(economics$pce)*sd(economics$psavert))
[1] TRUE
```