Practical No. 1

BASICS OF R COMMANDS

SOURCE CODE & OUTPUT:

1. R – Datatypes:

i. **Numeric**
ii. **Integer**
iii. **Complex**
iv. **Logical**
v. **Character**

```
> # R - Datatypes
> # Numeric
> x<-5
> y<-7
> z<-x+y
> z
[1] 12
> class(z)
[1] "numeric"
>
> # Integer
> x<-5L
> y<-7L
> z<-x+y
> z
[1] 12
> class(z)
[1] "integer"
>
> # Complex
> x<-2+3i
> y<-3-2i
> z<-x+y
> z
[1] 5+1i
> class(z)
[1] "complex"
```

```
> # Logical
> x<-TRUE
> y<-FALSE
> z<-x&y
> z
[1] FALSE
> class(z)
[1] "logical"
> x<-T
> y<-F
> z<-x|y
> z
[1] TRUE
> class(z)
[1] "logical"
>
> # Character
> course<-"Bioinformatics"
> course
[1] "Bioinformatics"
> class(course)
[1] "character"
> x<-"TRUE"
> x
[1] "TRUE"
> class(x)
[1] "character"
```

2. R – Vectors:

i. Vector Creation Using Colon Operator:

```
> # R - Vectors
> # Vector creation using colon operator
> x<-2:9
> x
[1] 2 3 4 5 6 7 8 9
> class(x)
[1] "integer"
> y<-3.2:8.2
> y
[1] 3.2 4.2 5.2 6.2 7.2 8.2
> class(y)
[1] "numeric"
> z<-1.6:5
> z
[1] 1.6 2.6 3.6 4.6
> class(z)
[1] "numeric"
```

ii. Vector Creation Using seq() Function:

```
> # Vector creation using seq() Function
> x<-seq(from=2,to=5,by=0.5)
> x
[1] 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> y<-seq(7,2)
> y
[1] 7 6 5 4 3 2
> z<-seq(7,4,-0.5)
> z
[1] 7.0 6.5 6.0 5.5 5.0 4.5 4.0
> v<-seq(1,20,4)
> v
[1]  1  5  9 13 17
```

```
> u<-seq(from=3,length=7,by=6)
> u
[1]  3  9 15 21 27 33 39
```

iii. Vector Creation Using c() Function:

```
> # Vector creation using c() function
> x<-c(2,3,6)
> x
[1] 2 3 6
> y<-c('T','C','G','A')
> y
[1] "T" "C" "G" "A"
> z<-c(2+3i,3-1i,-2+4i)
> z
[1]  2+3i  3-1i -2+4i
> v<-c(3,3-1i,7.2)
> v
[1] 3.0+0i 3.0-1i 7.2+0i
> u<-c(4,'C',3+4i)
> u
[1] "4"    "C"    "3+4i"
> class(x)
[1] "numeric"
> class(y)
[1] "character"
> class(z)
[1] "complex"
> class(v)
[1] "complex"
> class(u)
[1] "character"
```

iv. Vector Creation Using scan() Function:

```
> # Vector creation using scan() function
> x=scan()
1: 23
2: 4
3: 7
4: 12
5: -6
6: 19
7:
Read 6 items
> x
[1] 23  4  7 12 -6 19
> y=scan(what="character")
1: "Khalsa"
2: "Matunga" "Bioinformatics"
4:
Read 3 items
> y
[1] "Khalsa"          "Matunga"          "Bioinformatics"
> z=scan(nmax=4)
1: 5 7 9 11 13 15
Read 4 items
> z
[1]  5  7  9 11
```

- **Accessing Vector Elements:**

```
> # Accessing vector elements using position
> WeekDays<-c('Mon','Tue','Wed','Thur','Fri')
> WeekDays
[1] "Mon"  "Tue"  "Wed"  "Thur" "Fri"
> WeekDays[3]
[1] "Wed"
> WeekDays[c(1,3,5)]
[1] "Mon" "Wed" "Fri"
>
> # Accessing vector elements using logical indexing
> WeekDays[c(F,F,T,F,F)]
[1] "Wed"
> WeekDays[c(F,T)]
[1] "Tue"  "Thur"
>
> # Accessing vector elements using negative indexing
> WeekDays[-2]
[1] "Mon"  "Wed"  "Thur" "Fri"
> WeekDays[c(-2,-4)]
[1] "Mon" "Wed" "Fri"
```

- **Manipulation with Vectors:**

```
> # Manipulation with vectors
> x<-c(2,3,6)
> y<-c(4,5,2)
> x+5
[1]  7  8 11
> y-2
[1] 2 3 0
> 2*x
[1]  4  6 12
> y/4
[1] 1.00 1.25 0.50
> x%%2
[1] 0 1 0
> x%/%2
[1] 1 1 3
> x^2
[1]  4  9 36
> x+y
[1] 6 8 8
> x-y
[1] -2 -2  4
> x*y
[1]  8 15 12
> x/y
[1] 0.5 0.6 3.0
> x^y
[1]  16 243  36
```

```
> # rep() function
> x<-c(13,17,20,21)
> rep(x,times=3)
 [1] 13 17 20 21 13 17 20 21 13 17 20 21
> rep(x,each=2)
[1] 13 13 17 17 20 20 21 21
> rep(x,each=2,times=3)
 [1] 13 13 17 17 20 20 21 21 13 13 17 17 20 20 21 21 13 13 17 17 20 20 21 21
```

3. R – Lists:

```
> # Creating a R-List
> firstList<-list('Nucleotides',c(1,2,3,4),list('T','C','G','A'),sin)
> firstList
[[1]]
[1] "Nucleotides"

[[2]]
[1] 1 2 3 4

[[3]]
[[3]][[1]]
[1] "T"

[[3]][[2]]
[1] "C"

[[3]][[3]]
[1] "G"

[[3]][[4]]
[1] "A"


[[4]]
function (x)  .Primitive("sin")
```

```
> # Accessing List Elements
> firstList[[2]]
[1] 1 2 3 4
> firstList[[2]][3]
[1] 3
> firstList[[3]][4]
[[1]]
[1] "A"

> firstList[[4]]
function (x)  .Primitive("sin")
```

4. R – Factors

```
> # Creating factors
> age<-c(19,19,20,21,19,26,27,19,18,18,20,20,21,22,22,19,18,26,21)
> factor(age)
 [1] 19 19 20 21 19 26 27 19 18 18 20 20 21 22 22 19 18 26 21
Levels: 18 19 20 21 22 26 27
> nlevels(factor(age))
[1] 7
```

5. R – Data Frames:

```
> # Creating Data Frame
> Name<-c('Manish','Danish','David','Sifa')
> Age<-c(32,21,28,25)
> Salary<-c(41000,20000,32000,28000)
> Info<-data.frame(Name,Age,Salary)
> Info
    Name Age Salary
1 Manish  32  41000
2 Danish  21  20000
3  David  28  32000
4   Sifa  25  28000
> Info$Name
[1] Manish Danish David  Sifa
Levels: Danish David Manish Sifa
> class(Info$Name)
[1] "factor"
> Info$Name<-as.character(Name)
> class(Info$Name)
[1] "character"
> Info$Salary
[1] 41000 20000 32000 28000
> Info$Name[3]
[1] "David"
> Info$Salary[3]
[1] 32000
> # Adding a column to existing data frame
> Emp_ID<-c(1001,1002,1003,1004)
> InfoNew<-cbind(Emp_ID,Info)
> InfoNew
  Emp_ID   Name Age Salary
1   1001 Manish  32  41000
2   1002 Danish  21  20000
3   1003  David  28  32000
4   1004   Sifa  25  28000
> # Adding a row to existing data frame
> NewRow<-c(1005,'Ashok',20,19000)
> InfoNew2<-rbind(InfoNew,NewRow)
> InfoNew2
  Emp_ID   Name Age Salary
1   1001 Manish  32  41000
2   1002 Danish  21  20000
3   1003  David  28  32000
4   1004   Sifa  25  28000
5   1005  Ashok  20  19000
```

PRACTICAL No. 2

MATRIX COMPUTATIONS

SOURCE CODE & OUTPUT:

```
> # Creating a matrix
> A<-matrix(data=c(1,0,-1,3,2,4,5,1,-2),nrow=3,ncol=3)
> A
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    0    2    1
[3,]   -1    4   -2
> B<-matrix(data=c(1,0,-1,3,2,4,5,1,-2),nrow=3,ncol=3,byrow=TRUE)
> B
     [,1] [,2] [,3]
[1,]    1    0   -1
[2,]    3    2    4
[3,]    5    1   -2
> X<-matrix(1:9,3,3,T)
> X
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
> Y<-matrix(1:9,3)
> Y
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> Z<-matrix(data=c(1,2,-1,3,2,6),nrow=3,byrow=T)
> Z
     [,1] [,2]
[1,]    1    2
[2,]   -1    3
[3,]    2    6
```

```
> # Accessing elements of matrix
> A<-matrix(1:9,3)
> A[2,3]
[1] 8
> A[2,]
[1] 2 5 8
> A[,3]
[1] 7 8 9
```

```
> # Matrix Computations
> A<-matrix(c(3,2,-1,0,2,6,1,2,1),nrow=3)
> B<-matrix(c(1,0,-1,3,2,6,0,-2,-1),nrow=3)
> A
     [,1] [,2] [,3]
[1,]    3    0    1
[2,]    2    2    2
[3,]   -1    6    1
> B
     [,1] [,2] [,3]
[1,]    1    3    0
[2,]    0    2   -2
[3,]   -1    6   -1
>
> # Matrix Addition
> A+B
     [,1] [,2] [,3]
[1,]    4    3    1
[2,]    2    4    0
[3,]   -2   12    0
>
> # Matrix Subtraction
> A-B
     [,1] [,2] [,3]
[1,]    2   -3    1
[2,]    2    0    4
[3,]    0    0    2


> # Matrix Multiplication
> A%*%B
     [,1] [,2] [,3]
[1,]    2   15   -1
[2,]    0   22   -6
[3,]   -2   15  -13
>
> # Matrix Transpose
> t(A)
     [,1] [,2] [,3]
[1,]    3    2   -1
[2,]    0    2    6
[3,]    1    2    1
>
> # Matrix Inverse
> solve(A)
       [,1]   [,2]   [,3]
[1,]  0.625 -0.375  0.125
[2,]  0.250 -0.250  0.250
[3,] -0.875  1.125 -0.375
```

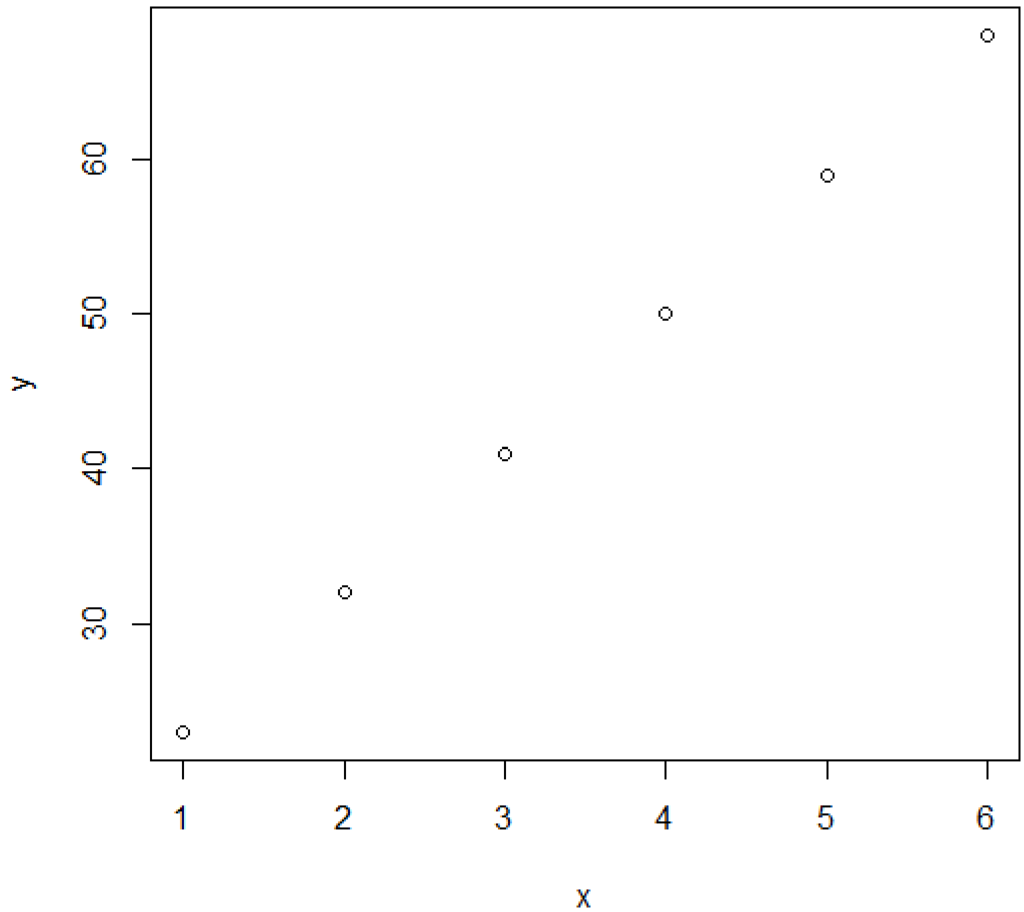
```
> # Determinant
> det(A)
[1] -16
>
> # Trace
> sum(diag(A))
[1] 6
>
> # Diagonal matrix
> D<-diag(c(2,7,1),nrow=3)
> D
     [,1] [,2] [,3]
[1,]    2    0    0
[2,]    0    7    0
[3,]    0    0    1
>
> S<-diag(5,nrow=3)
> S
     [,1] [,2] [,3]
[1,]    5    0    0
[2,]    0    5    0
[3,]    0    0    5
```

PRACTICAL No. 3

GRAPHS

SOURCE CODE & OUTPUT:

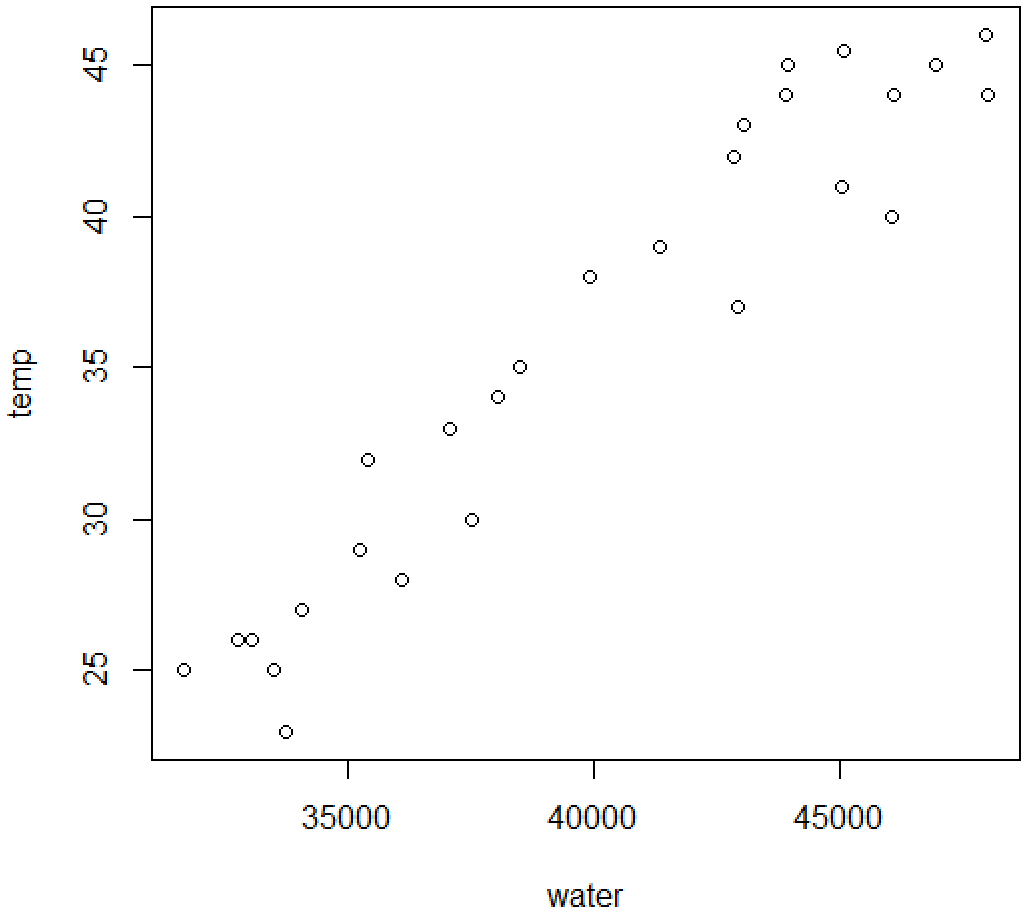
```
> x<-c(1,2,3,4,5,6)
> y<-c(23,32,41,50,59,68)
> plot(x,y)
```



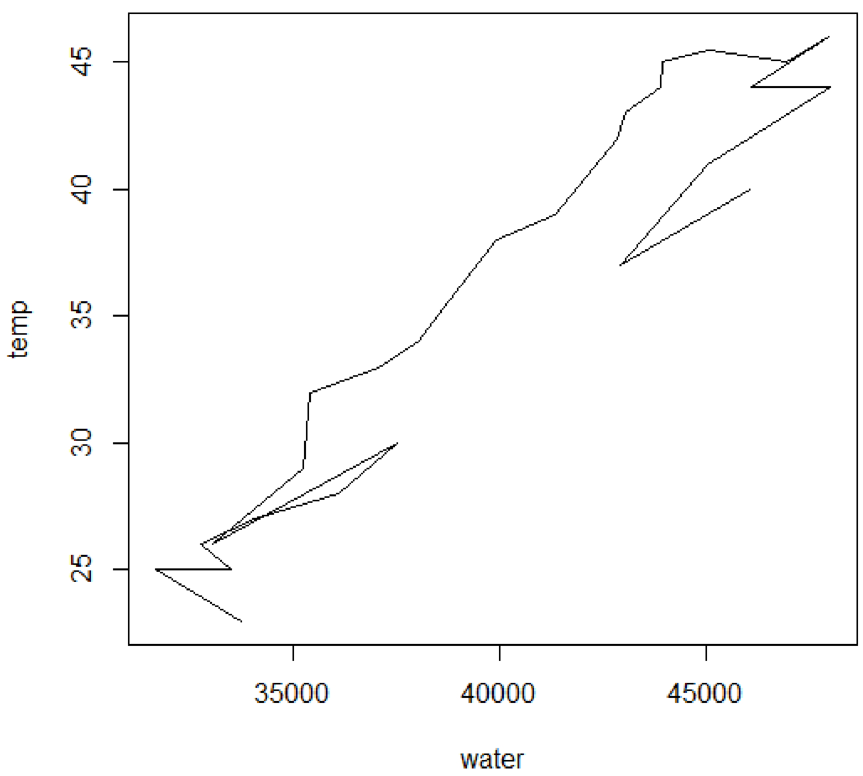
`plot(x, y, type)`

type	
"p" for **p**oints	"l" for **l**ines
"b" for **b**oth	"c" for the lines part alone of "b"
"o" for both '**o**verplotted'	"s" for stair **s**teps.
"h" for '**h**istogram' like (or 'high-density') vertical lines	

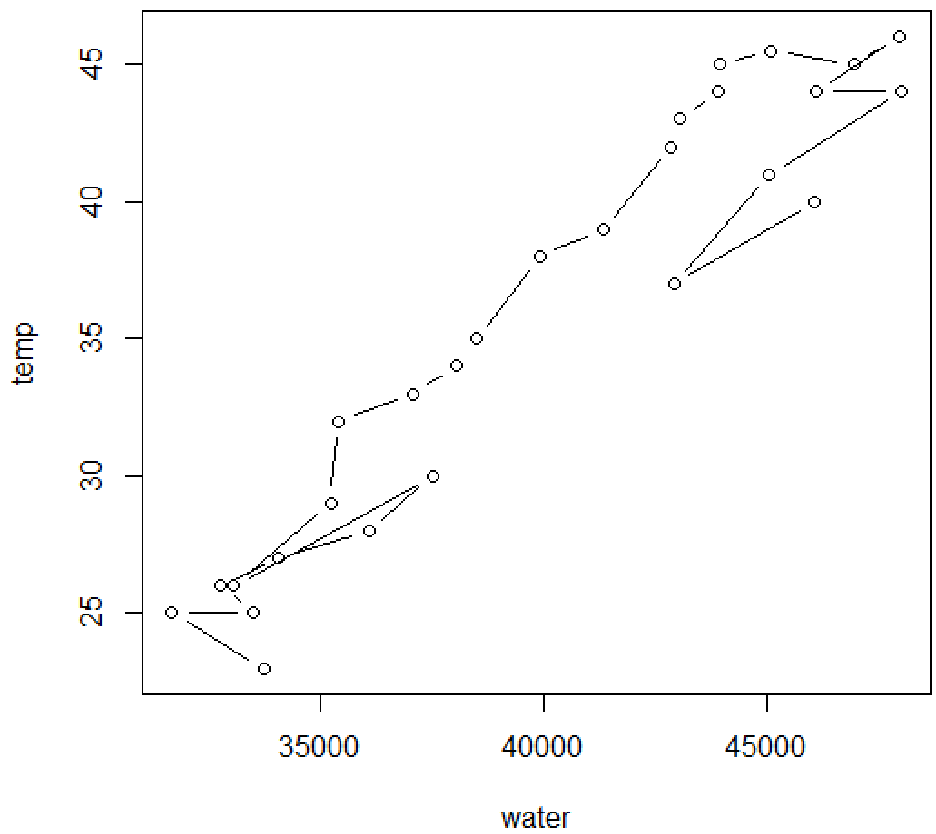
```
> water <- c(33710,31666,33495,32758,34067,36069, 37497,33044,35216, 35383,
+ 37066,38037,38495, 39895,41311,42849,43038,43873,43923, 45078, 46935,
+ 47951,46085,48003,45050,42924,46061)
> temp <- c(23,25,25,26,27,28,30,26,29,32,33,34, 35,38,39,42,43,44, 45,
+ 45.5,45,46,44,44,41,37,40)
> plot(water,temp)
> plot(water,temp,type='p')
```



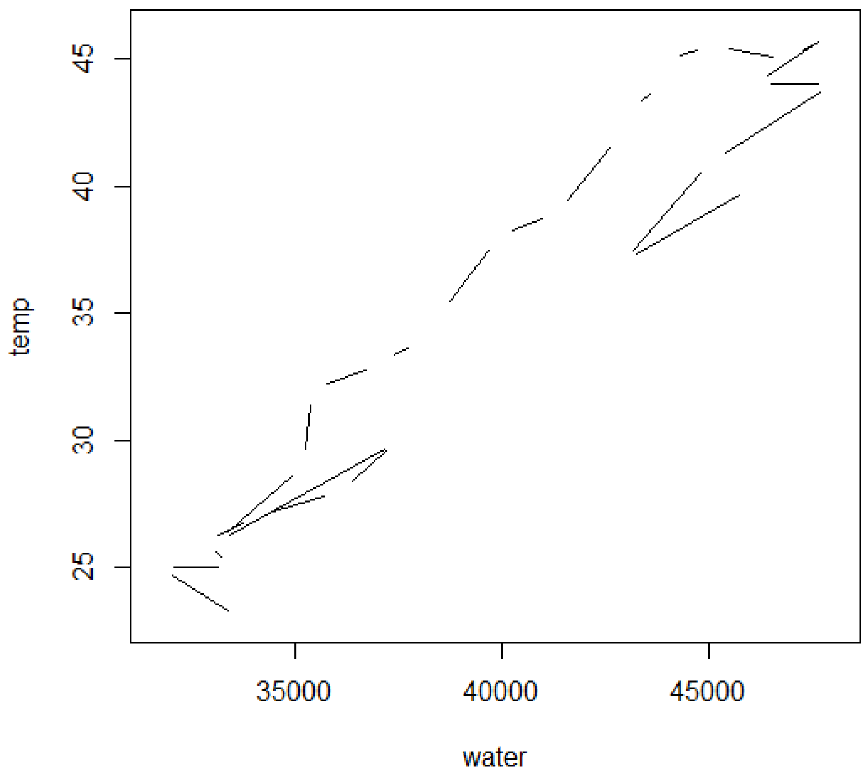
```
> plot(water,temp,type='l')
```



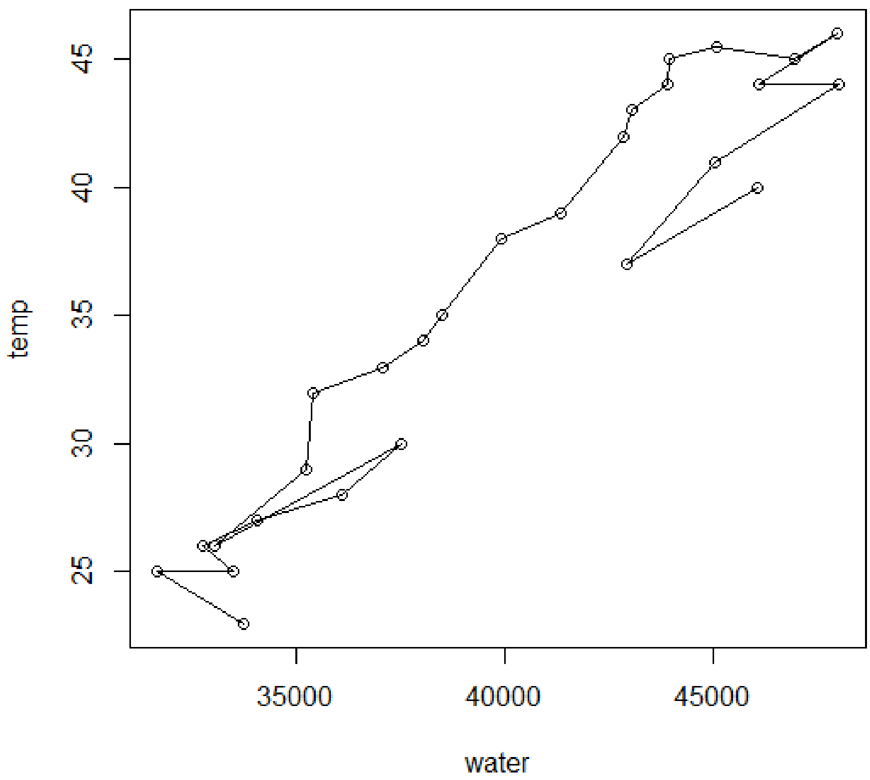

```
> plot(water,temp,type='b')
```



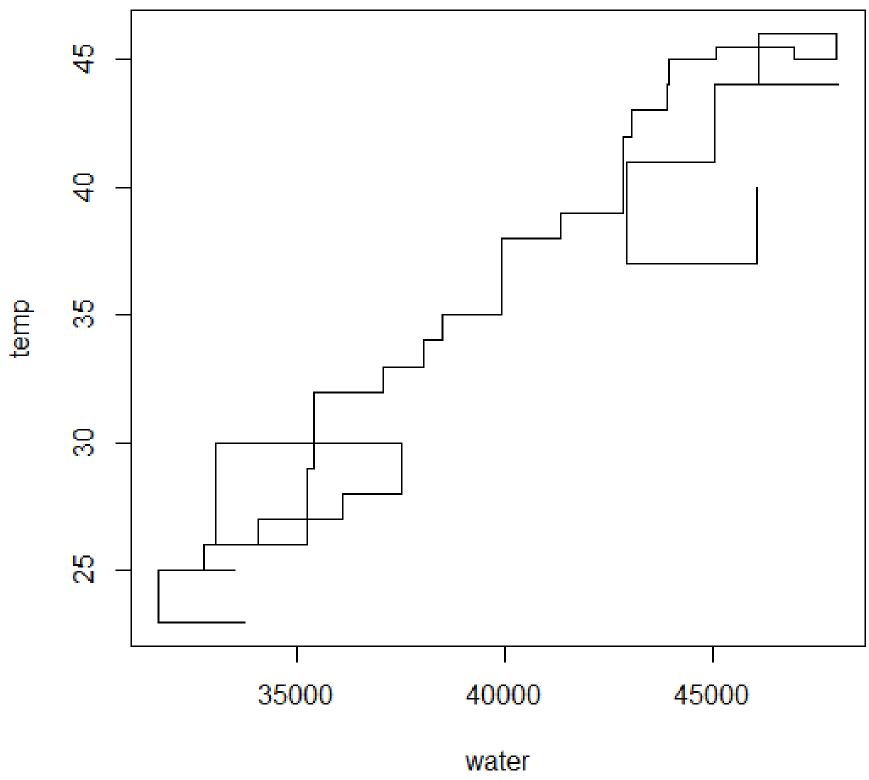
```
> plot(water,temp,type='c')
```



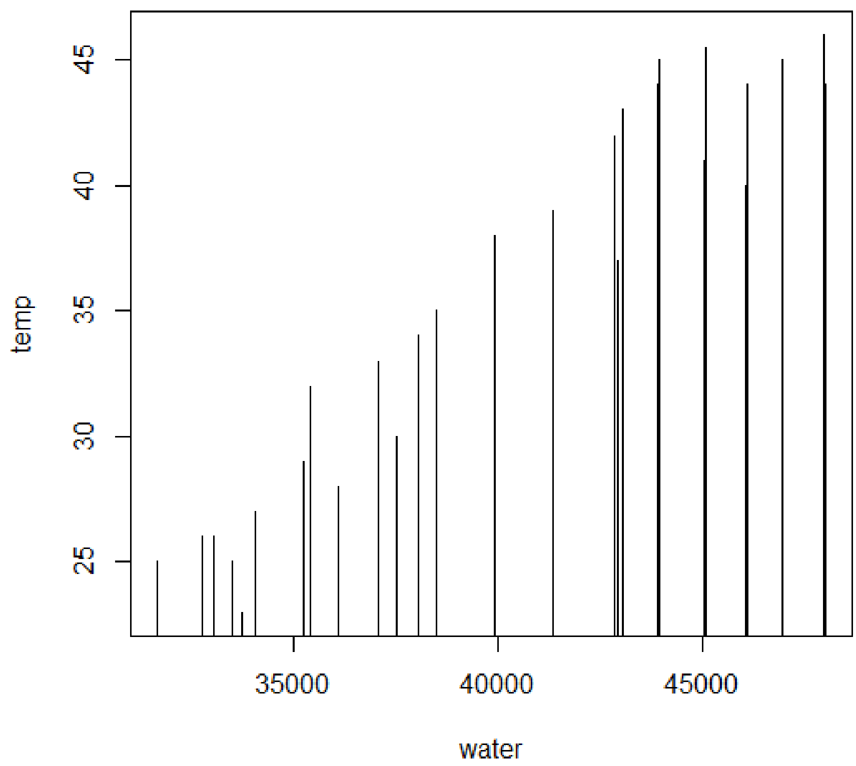

```
> plot(water,temp,type='o')
```



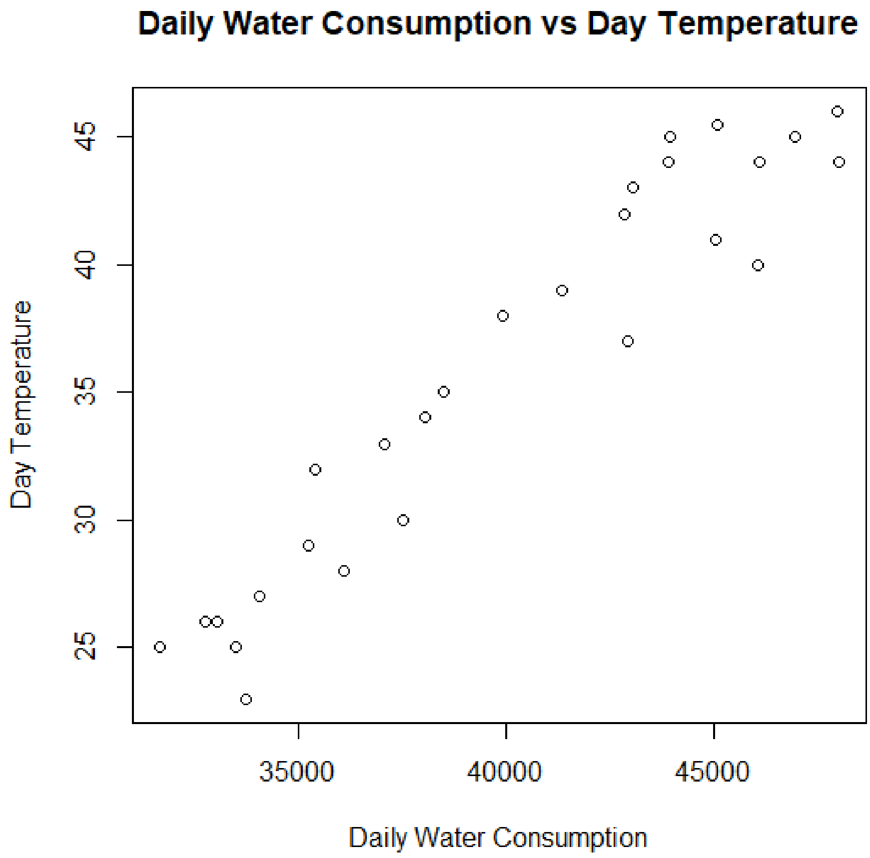
```
> plot(water,temp,type='s')
```




```
> plot(water,temp,type='h')
```



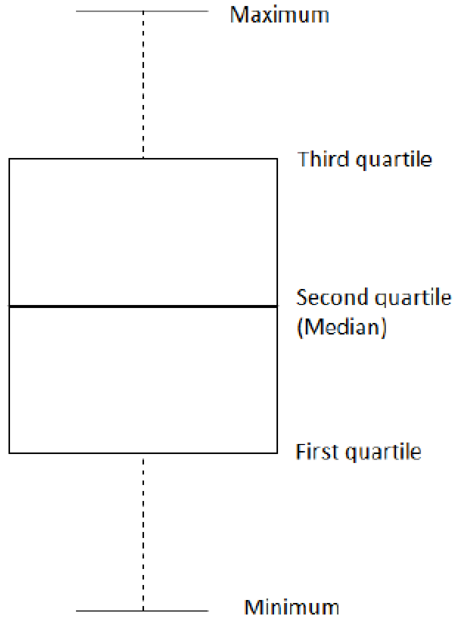
```
> plot(water,temp,xlab='Daily Water Consumption', ylab='Day Temperature',
+ main='Daily Water Consumption vs Day Temperature')
```



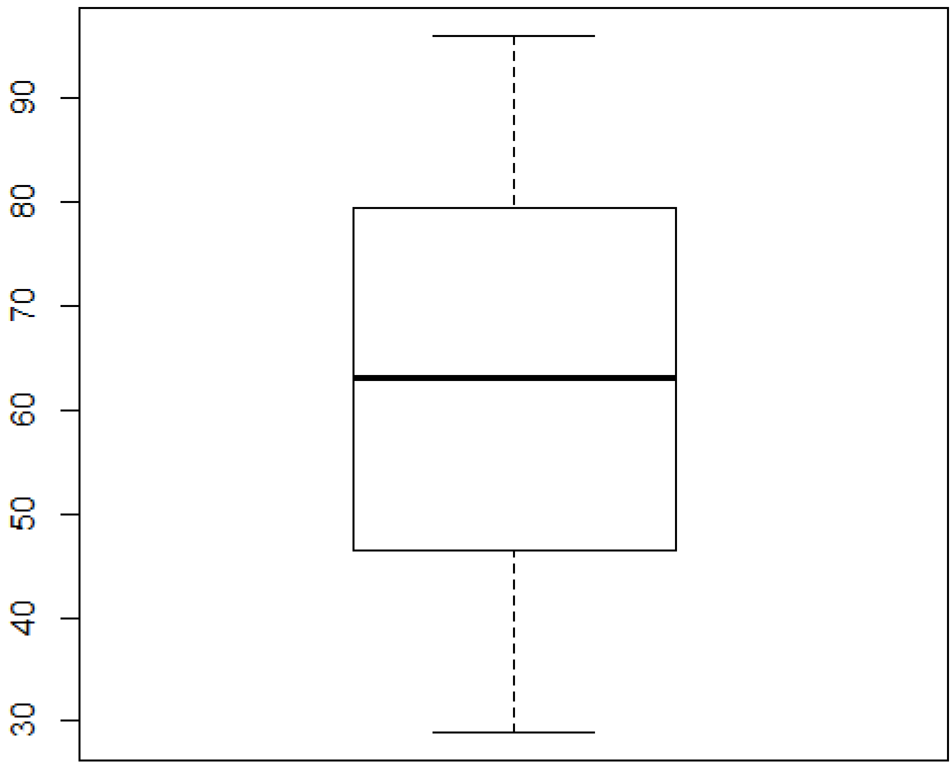
High-Level Plotting Functions

Function name	**Plot produced**
boxplot(x)	"Box and whiskers" plot
pie(x)	Circular pie chart
hist(x)	Histogram of the frequencies of x
barplot(x)	Histogram of the values of x
stripchart(x)	Plots values of x along a line
dotchart(x)	Cleveland dot plot
pairs(x)	For a matrix x, plots all bivariate pairs
plot.ts(x)	Plot of x with respect to time (index values of the vector unless specified)
contour(x,y,z)	Contour plot of vectors x and y, z must be a matrix of dimension rows=x and columns=y
image(x,y,z)	Same as contour plot but uses colors instead of lines
persp(x,y,z)	3-d contour plot

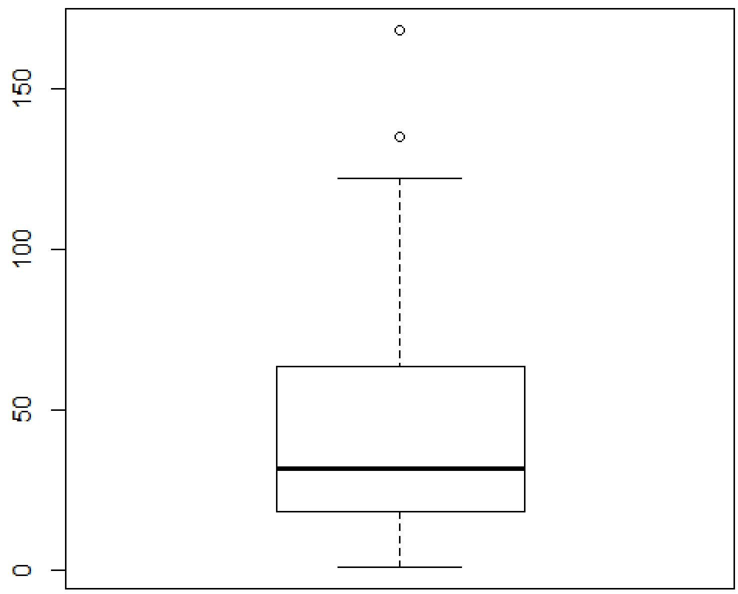
BOXPLOT:




```
> marks <- c(68, 82, 63, 86, 34, 96, 41, 89, 29, 51, 75, 77, 56, 59, 42)
> boxplot(marks)
```

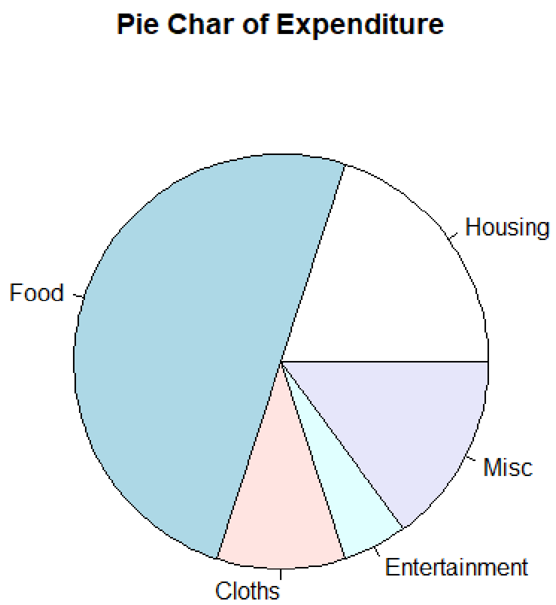


```
> str(airquality)
'data.frame':   153 obs. of  6 variables:
 $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
 $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
 $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
 $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
> boxplot(airquality$Ozone)
```



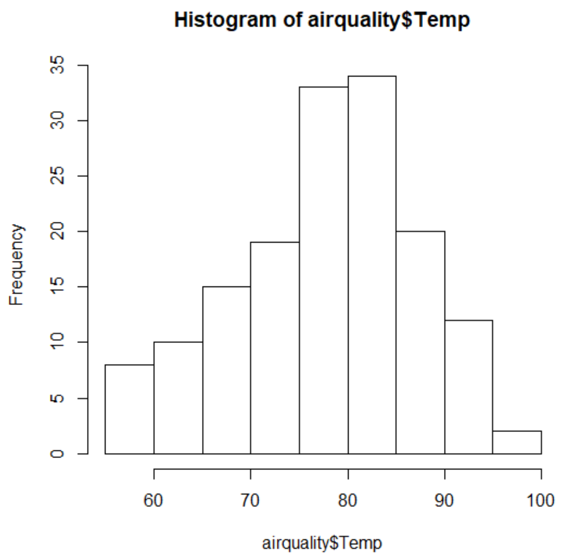
PIE CHART:

```
> exp<-c(6000,15000,3000,1500,4500)
> lbl<-c('Housing','Food','Cloths','Entertainment','Misc')
> pie(exp,labels=lbl,main='Pie Char of Expenditure')
```

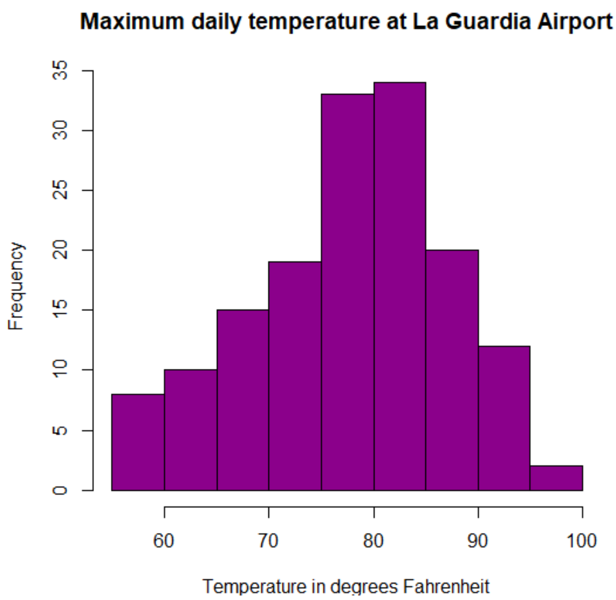


HISTOGRAM:

```
> str(airquality)
'data.frame':   153 obs. of  6 variables:
 $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
 $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
 $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
 $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
> hist(airquality$Temp)
```

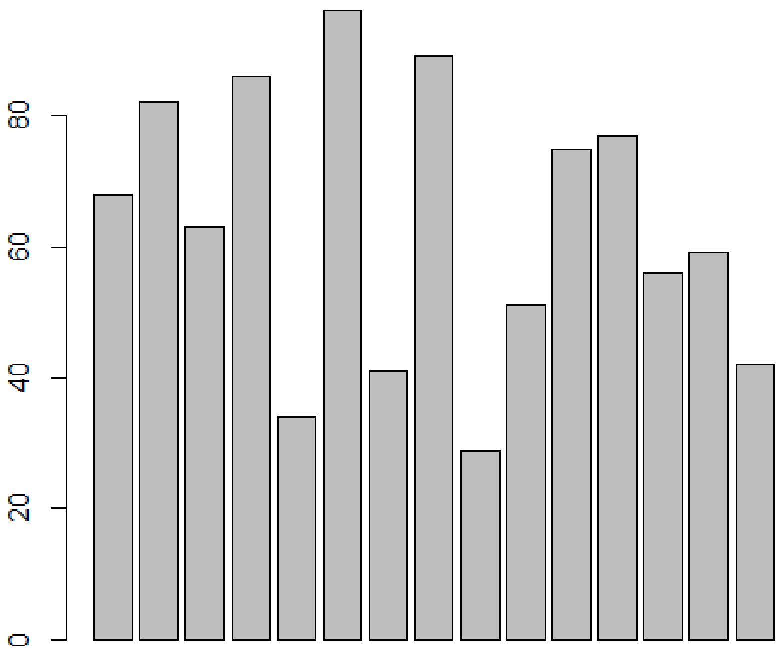



```
> # histogram with added parameters
> hist(airquality$Temp,
+ main="Maximum daily temperature at La Guardia Airport",
+ xlab="Temperature in degrees Fahrenheit",
+ col="darkmagenta")
```



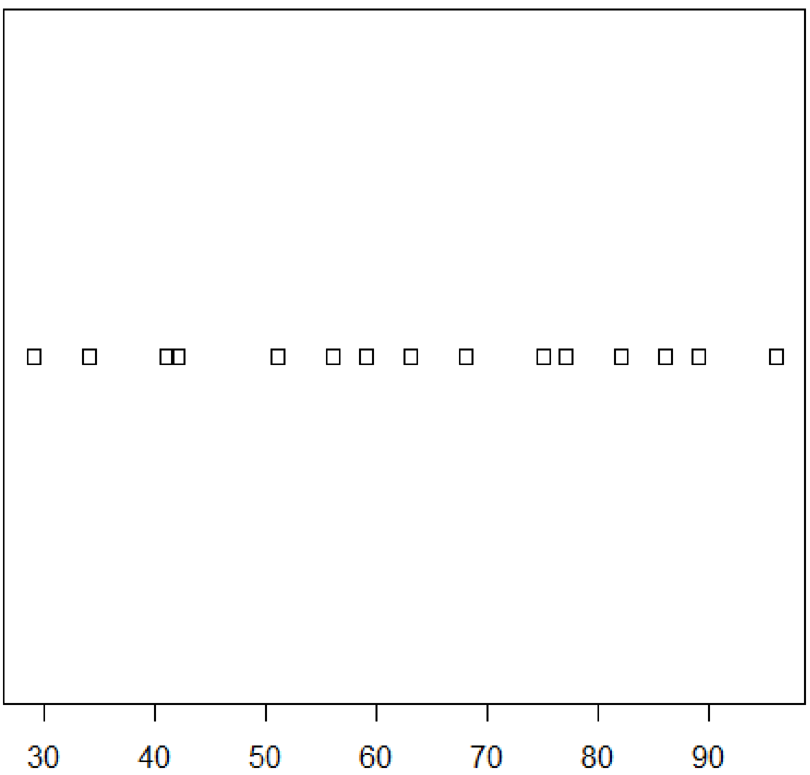
BARPLOT:

```
> marks <- c(68, 82, 63, 86, 34, 96, 41, 89, 29, 51, 75, 77, 56, 59, 42)
> barplot(marks)
```



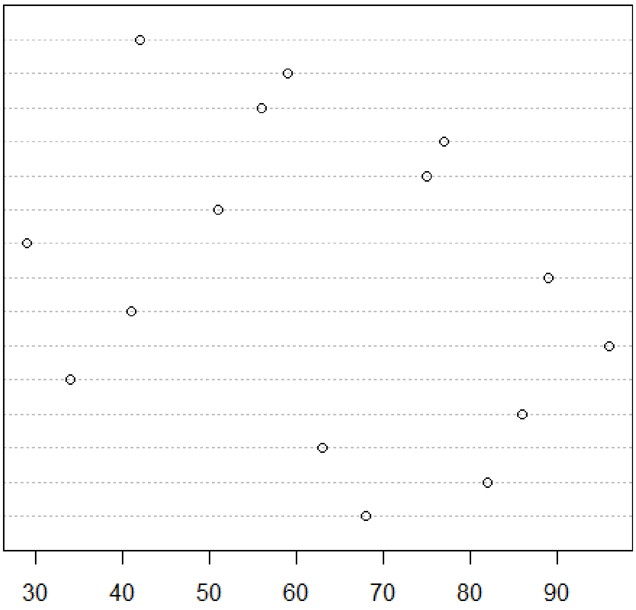
STRIPCHART:

```
> stripchart(marks)
```



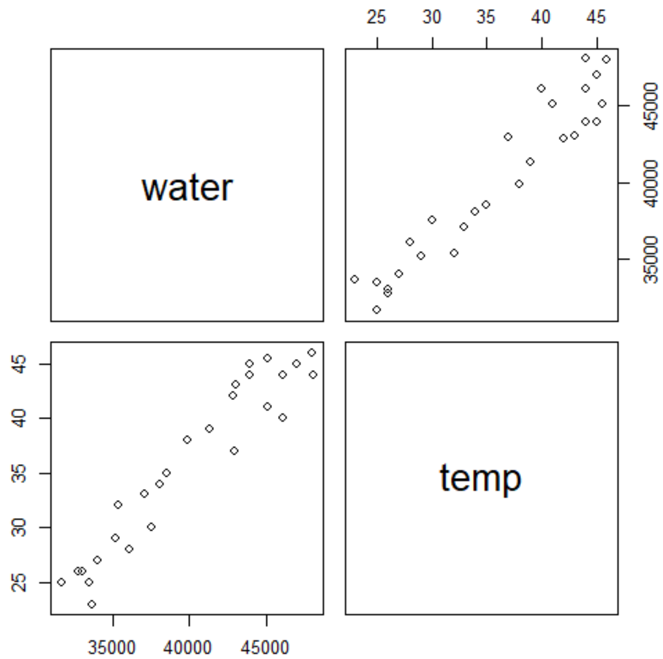
DOTCHART:

```
> dotchart(marks)
```



PAIRS:

```
> water <- c(33710,31666,33495,32758,34067,36069, 37497,33044,35216, 35383,
+ 37066,38037,38495, 39895,41311,42849,43038,43873,43923, 45078, 46935,
+ 47951,46085,48003,45050,42924,46061)
> temp <- c(23,25,25,26,27,28,30,26,29,32,33,34, 35,38,39,42,43,44, 45,
+ 45.5,45,46,44,44,41,37,40)
> pairs(cbind(water,temp))
```



PRACTICAL No. 4

COUNTING & PROBABILITY

SOURCE CODE & OUTPUT:

- **COUNTING**

```
> # Counting
>
> # Factorial Computation
> # gamma(n+1)=n!
> # Compute 5!
> gamma(6)
[1] 120
>
> # nCr Computation
> # Compute 10C2
> choose(10,2)
[1] 45
>
> # nPr Computation
> # nPr=n!/(n-r)!
> # Compute 5P2
> # 5P2=5!/3!=gamma(6)/gamma(4)
> gamma(6)/gamma(4)
[1] 20

> # choosing 3 nucleotides from 4
> choose(4,3)
[1] 4
>
> # calculate the numbers of unique 8-mer peptide
> # arrangements taken from the 20 amino acids
> # Ans=20P8=20!/12!=gamma(21)/gamma(13)
> Ans<-gamma(21)/gamma(13)
> Ans
[1] 5079110400
```

- **PROBABILITY**

Consider the RNA sequence below:

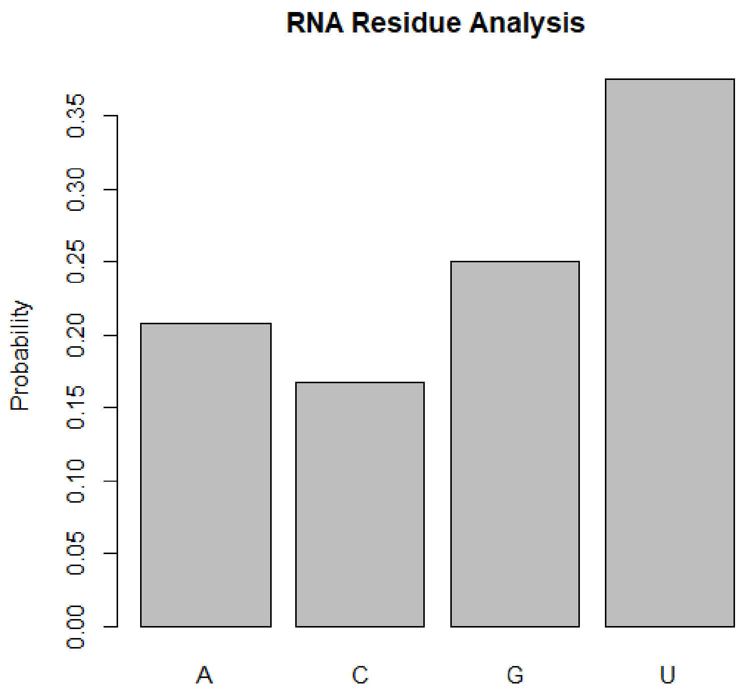
AUGCUUCGAAUGCUGUAUGAUGUC

In this sequence there are 5 A's, 9 U's, 6 G's, and 4 C's with a total of 24 residues. To model this sequence, the random variable X can be used where X represents the nucleotide residues. let's assign the random variable values representing A as 0, C as 1, G as 2 and U as 3.

Residue	Value of X (=x)	P (X=x)
A	0	5/24=0.208
C	1	4/24=0.167
G	2	6/24=0.25
U	3	9/24=0.375

In R a simple histogram can be used to model the probability distribution function for this example.

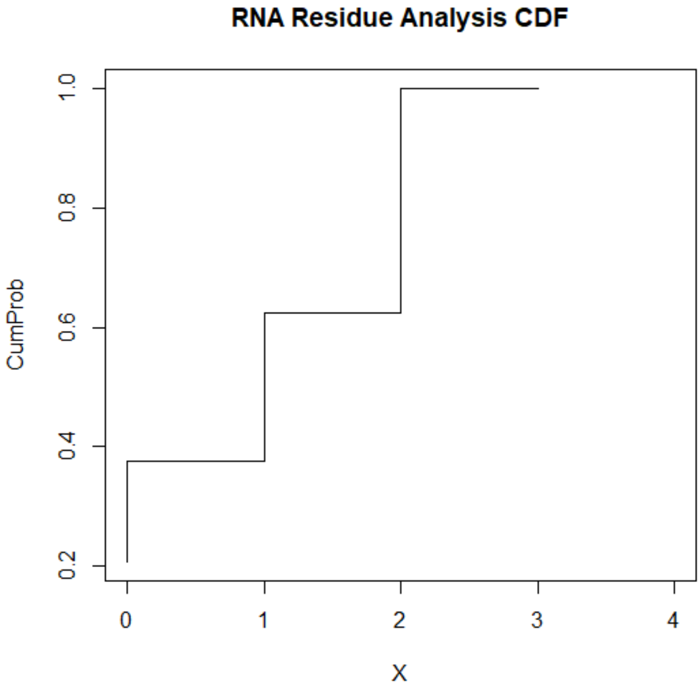
```
> X<-c(0,1,2,3)
> Probability<-c(0.208,0.167,0.25,0.375)
> N<-c('A','C','G','U')
> barplot(Probability,names=N,ylab="Probability",main="RNA Residue Analysis")
```



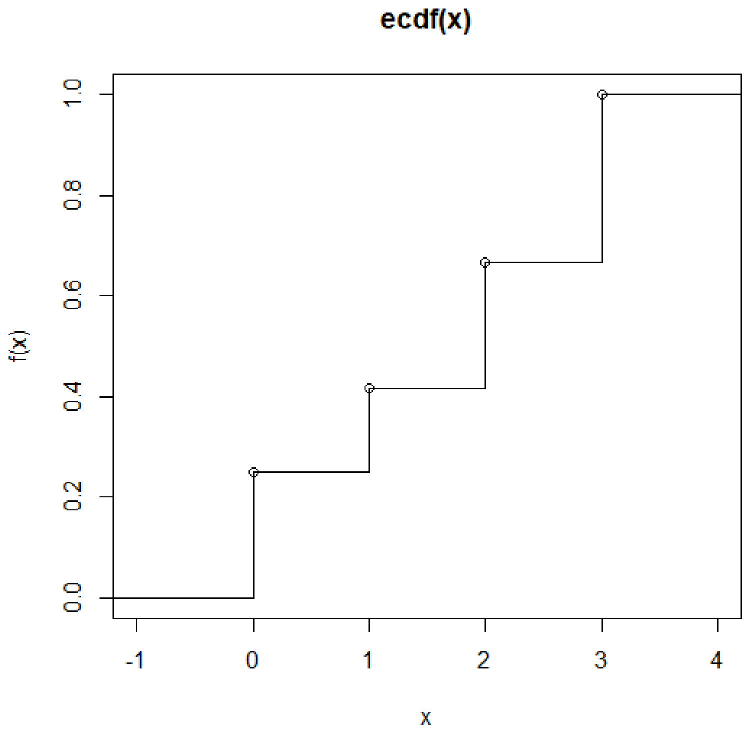
Residue	Value of X (=x)	P (X=x)	F(x)= P(X$\leq$ x)
A	0	5/24=0.208	0.208
C	1	4/24=0.167	0.375
G	2	9/24=0.375	0.625
U	3	6/24=0.25	1

In R a simple step graph can be used to model the cumulative probability distribution function for this example.

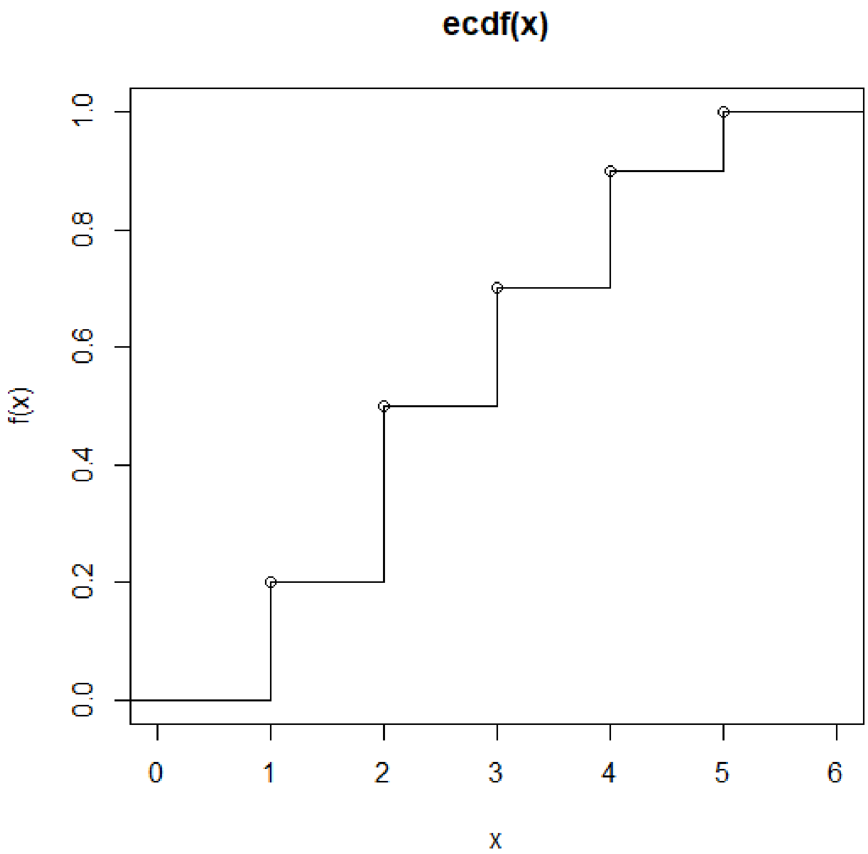
```
> X<-c(0,1,2,3)
> CumProb<-c(0.208,0.375,0.625,1)
> plot(X,CumProb,xlim=range(0,1,2,3,4),
+ main="RNA Residue Analysis CDF",type="S")
```



```
> x<-c(0,3,2,1,3,3,1,2,0,0,3,2,1,3,2,3,0,3,2,0,3,2,0,1)
> plot.stepfun(x)
```



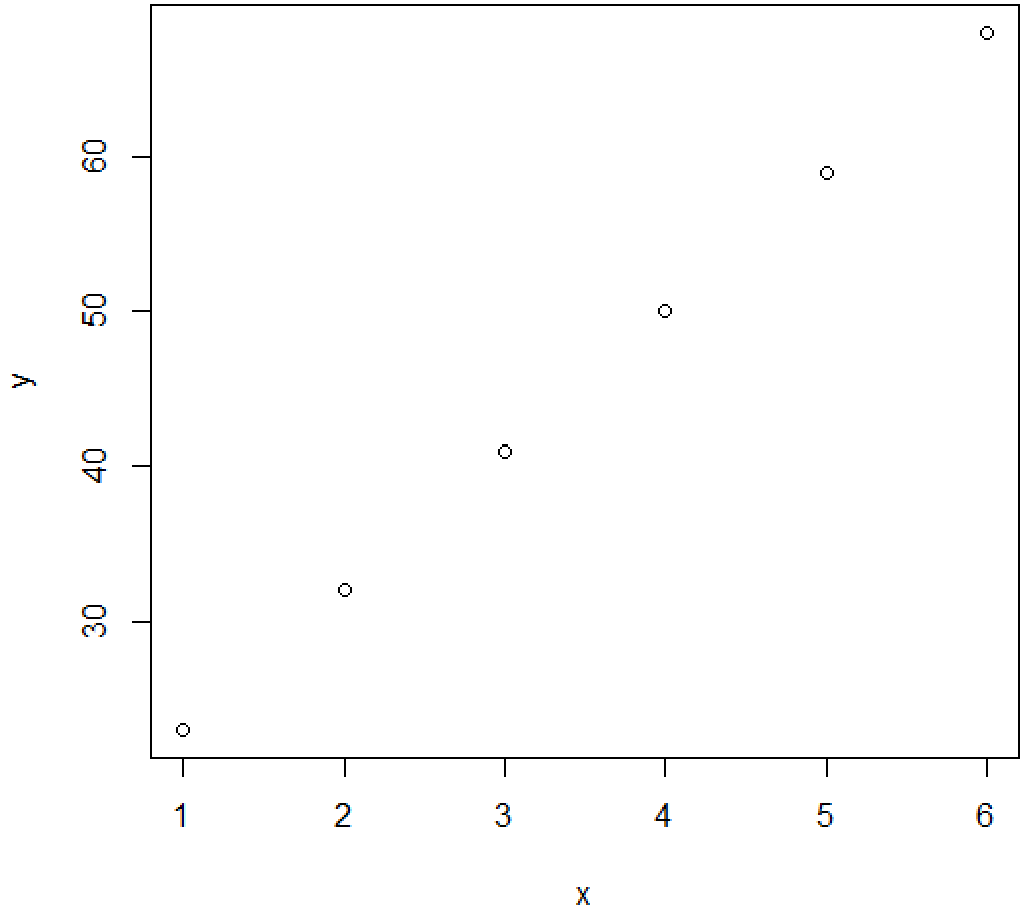

```
> x<-c(2,4,2,1,3,4,2,1,3,5)
> plot.stepfun(x)
```



GRAPHS

Scatter Plot

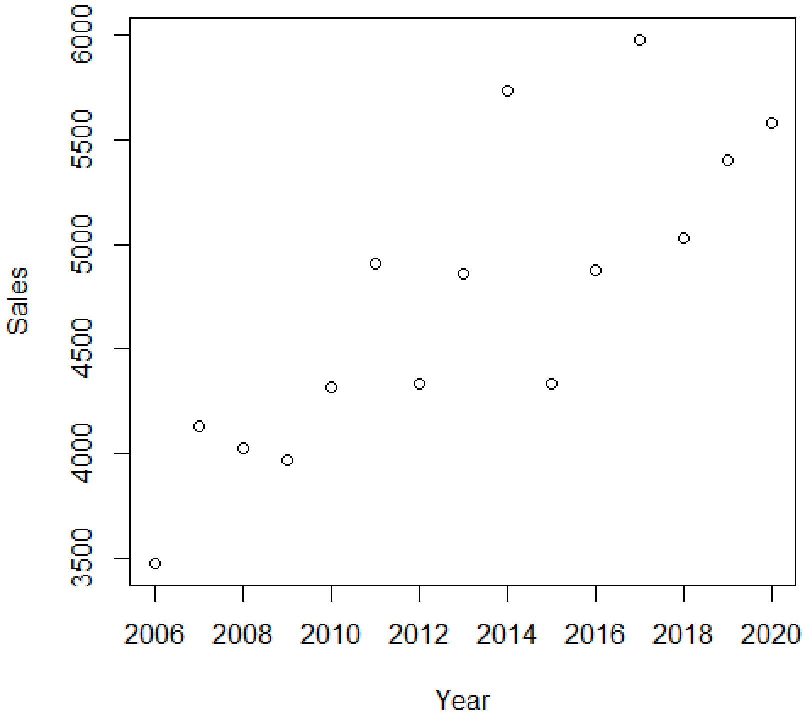
```
> x<-c(1,2,3,4,5,6)
> y<-c(23,32,41,50,59,68)
> plot(x,y)
```



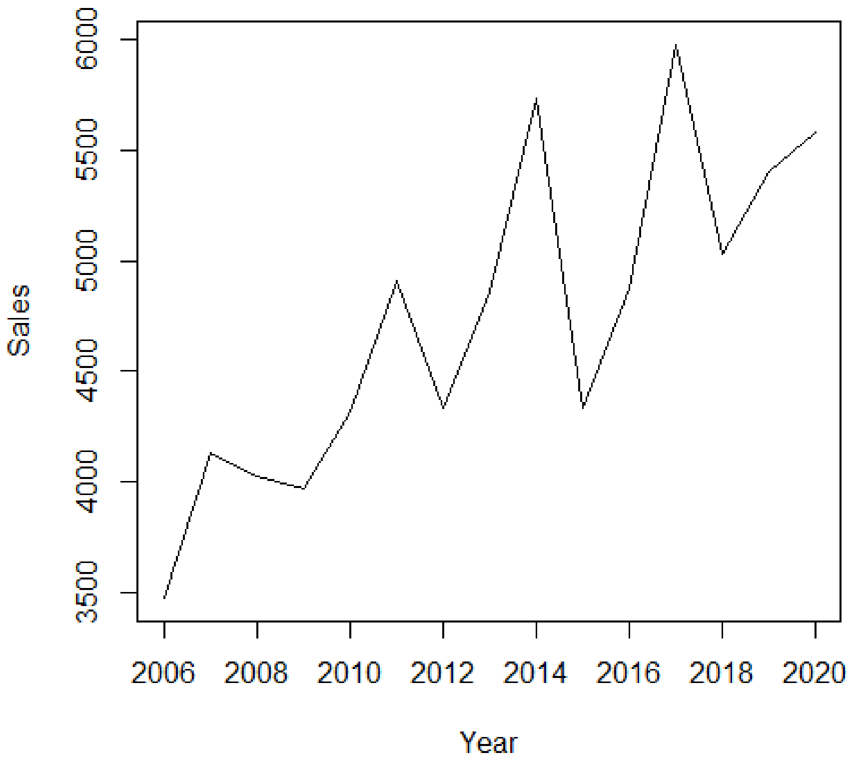

```
plot(x, y, type)
```

type	
"p" for **p**oints	"l" for **l**ines
"b" for **b**oth	"c" for the lines part alone of "b"
"o" for both '**o**verplotted'	"s" for stair **s**teps.
"h" for '**h**istogram' like (or 'high-density') vertical lines	

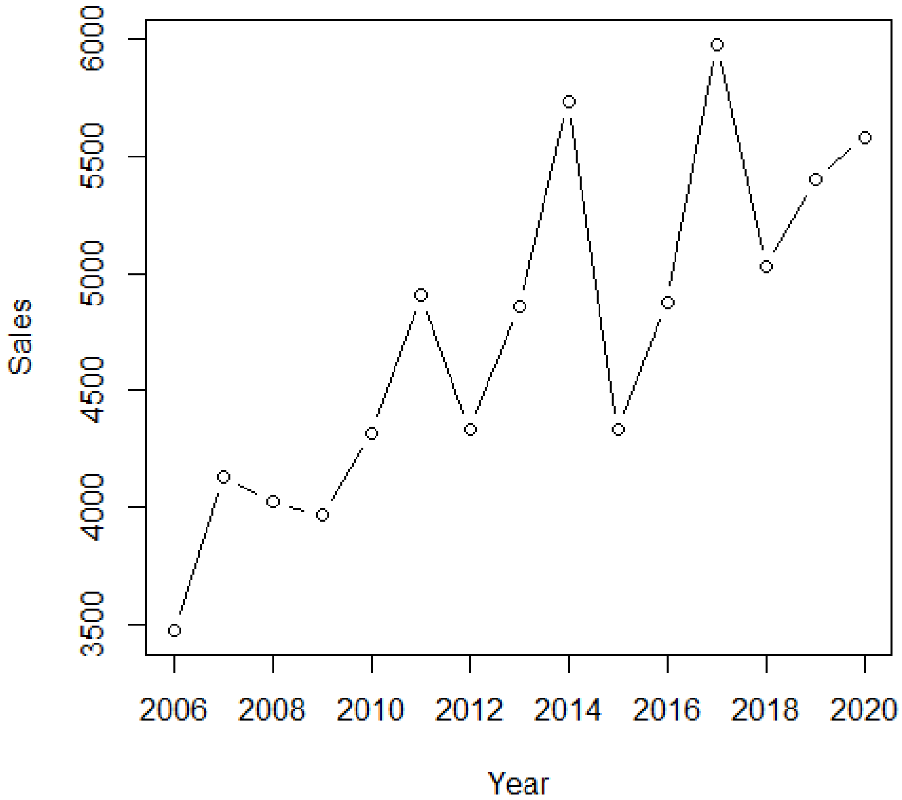
```
> Year<-c(2006,2007,2008,2009,2010,2011,2012,2013,
+ 2014,2015,2016,2017,2018,2019,2020)
> Sales<-c(3472,4134,4021,3964,4315,4907,4331,4856,
+ 5738,4330,4873,5981,5030,5400,5580)
> plot(Year,Sales,type='p')
```



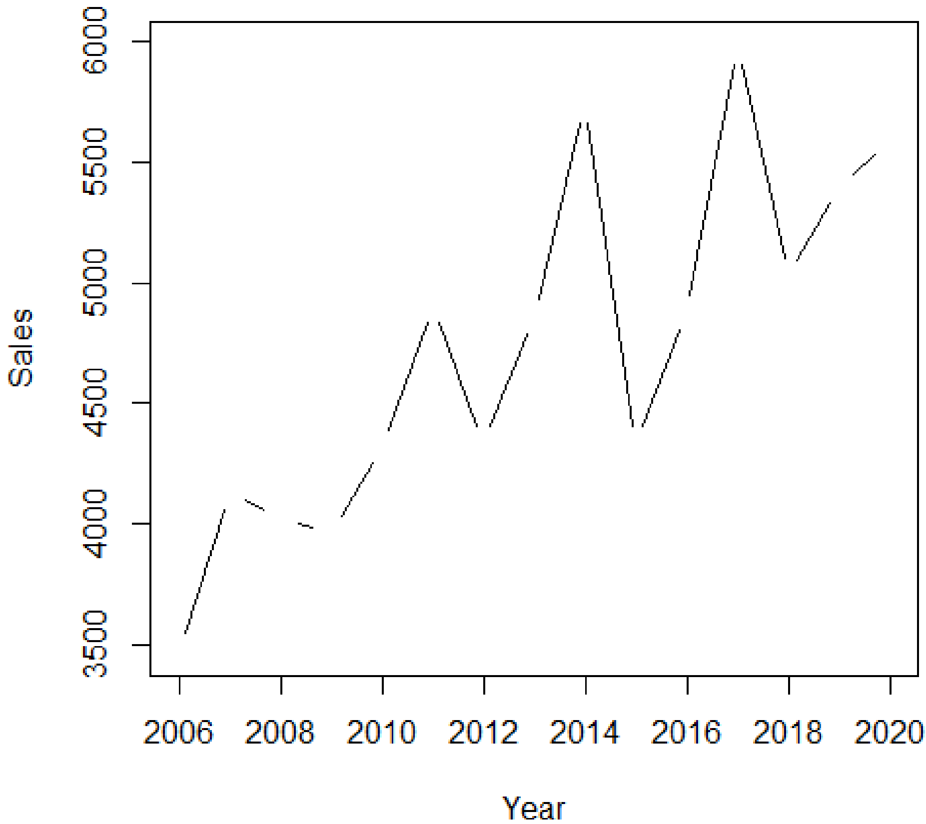

```
> plot(Year,Sales,type='l')
```



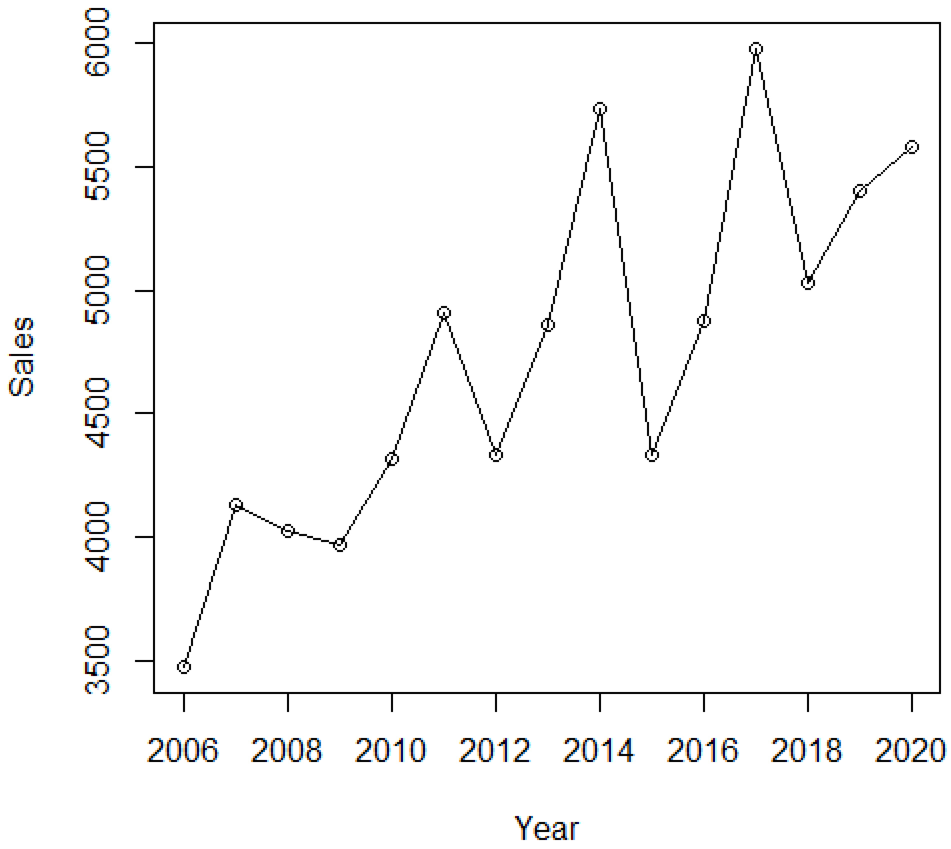
```
> plot(Year,Sales,type='b')
```



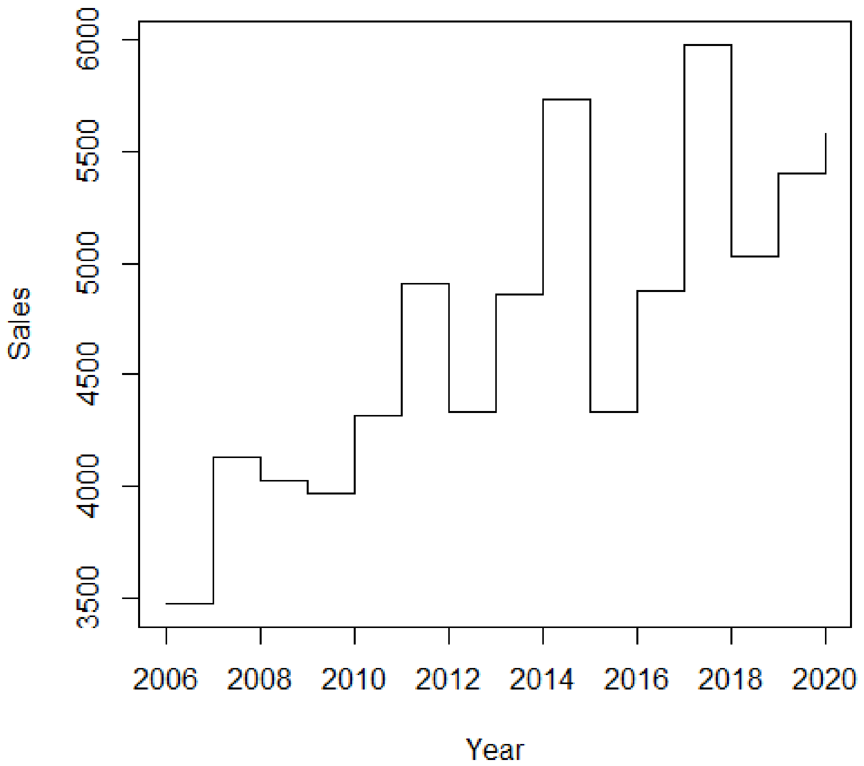

```
> plot(Year,Sales,type='c')
```



```
> plot(Year,Sales,type='o')
```

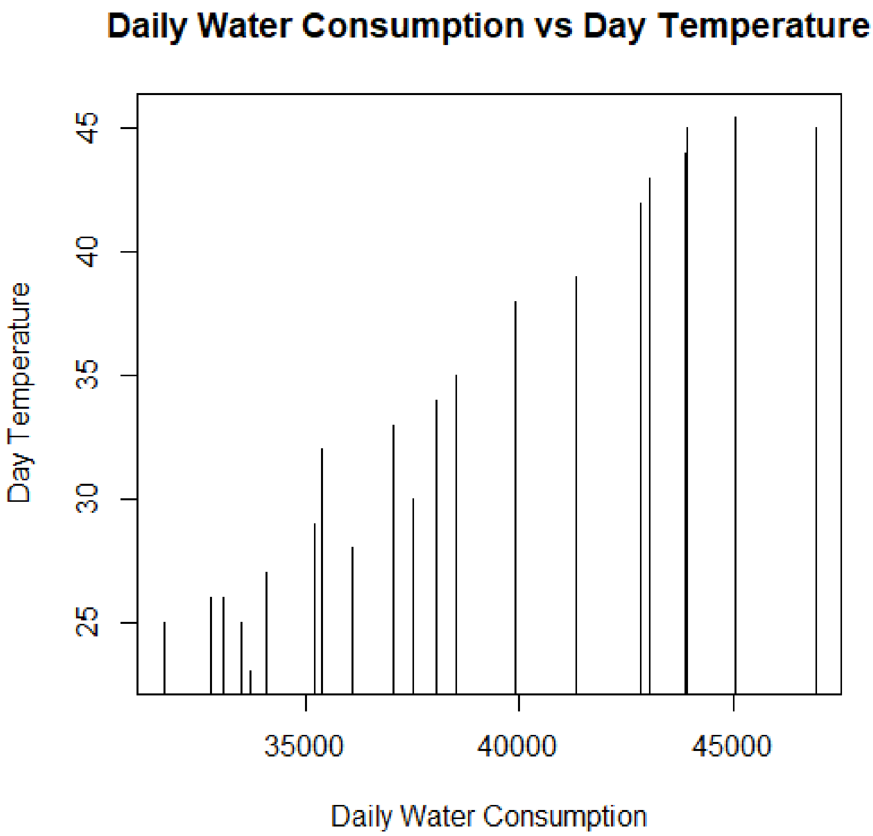



```
> plot(Year,Sales,type='s')
```



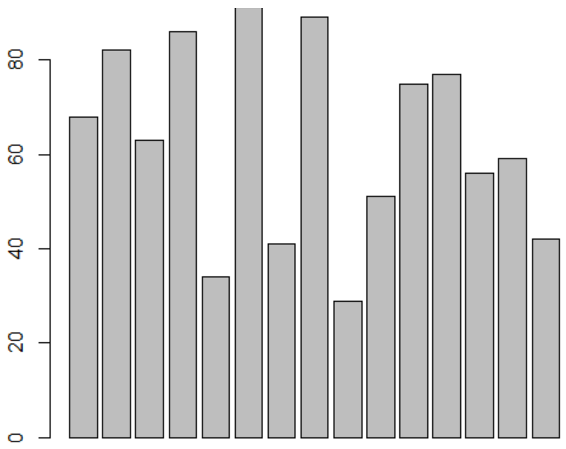
```
> plot(Year,Sales,type='h')
```


```
> water<-c(33710,31666,33495,32758,34067,36069,37497,33044,35216,35383,
+ 37066,38037,38495,39895,41311,42849,43038,43873,43923,45078,46935)
> temp<-c(23,25,25,26,27,28,30,26,29,32,33,34,35,38,39,42,43,44,45,45.5,45)
> plot(water,temp,type='h',xlab='Daily Water Consumption',ylab='Day Temperature',
+ main='Daily Water Consumption vs Day Temperature')
```



Barplot

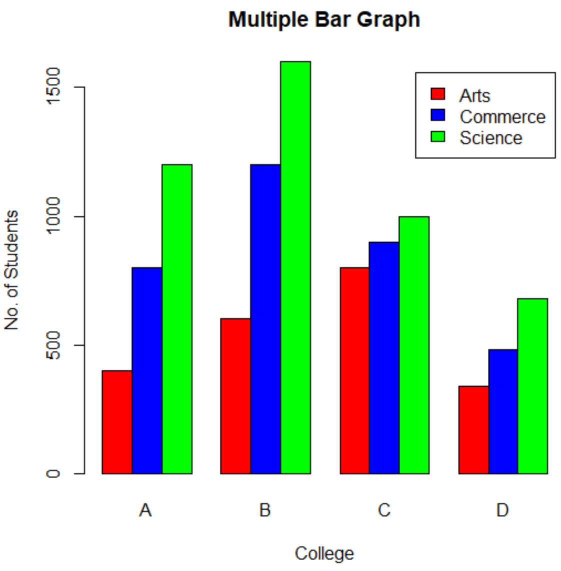
```
> marks <- c(68, 82, 63, 86, 34, 96, 41, 89, 29, 51, 75, 77, 56, 59, 42)
> barplot(marks)
```



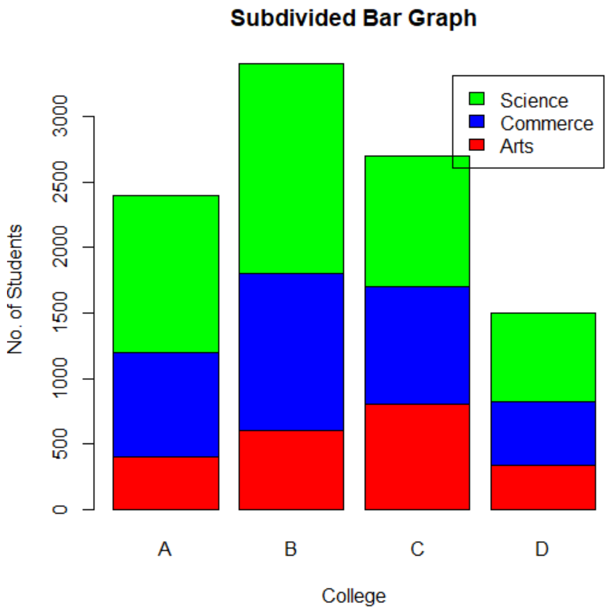
Represent the following data by multiple bar diagram and subdivided bar diagram.

College	Arts	Commerce	Science
A	400	800	1200
B	600	1200	1600
C	800	900	1000
D	340	480	680

```
> x<-c(400,800,1200,600,1200,1600,800,900,1000,340,480,680)
> data<-matrix(x,3)
> barplot(data,beside=T,names.arg=c('A','B','C','D'),
+ col=c('Red','Blue','Green'),legend=c('Arts','Commerce','Science'),
+ xlab='College',ylab='No. of Students',main='Multiple Bar Graph')
```



```
> x<-c(400,800,1200,600,1200,1600,800,900,1000,340,480,680)
> data<-matrix(x,3)
> barplot(data,beside=F,names.arg=c('A','B','C','D'),
+ col=c('Red','Blue','Green'),legend=c('Arts','Commerce','Science'),
+ xlab='College',ylab='No. of Students',main='Subdivided Bar Graph')
```

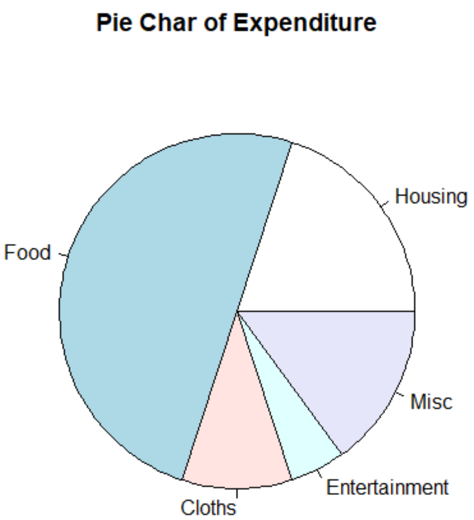


Pie Chart

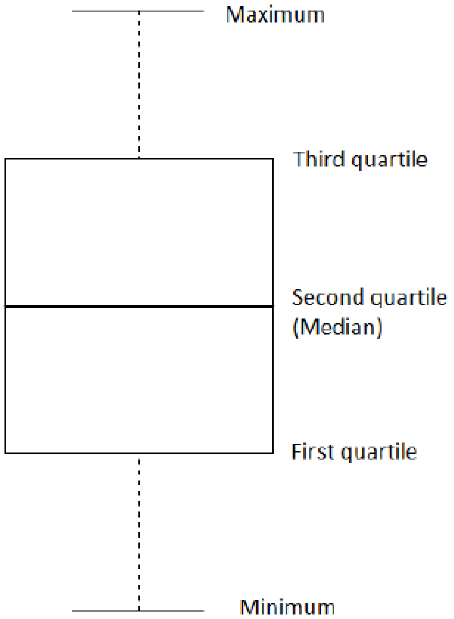
Represent the following data by pie chart:

Category	Housing	Food	Cloths	Entertainment	Misc
Expense	6000	15000	3000	1500	4500

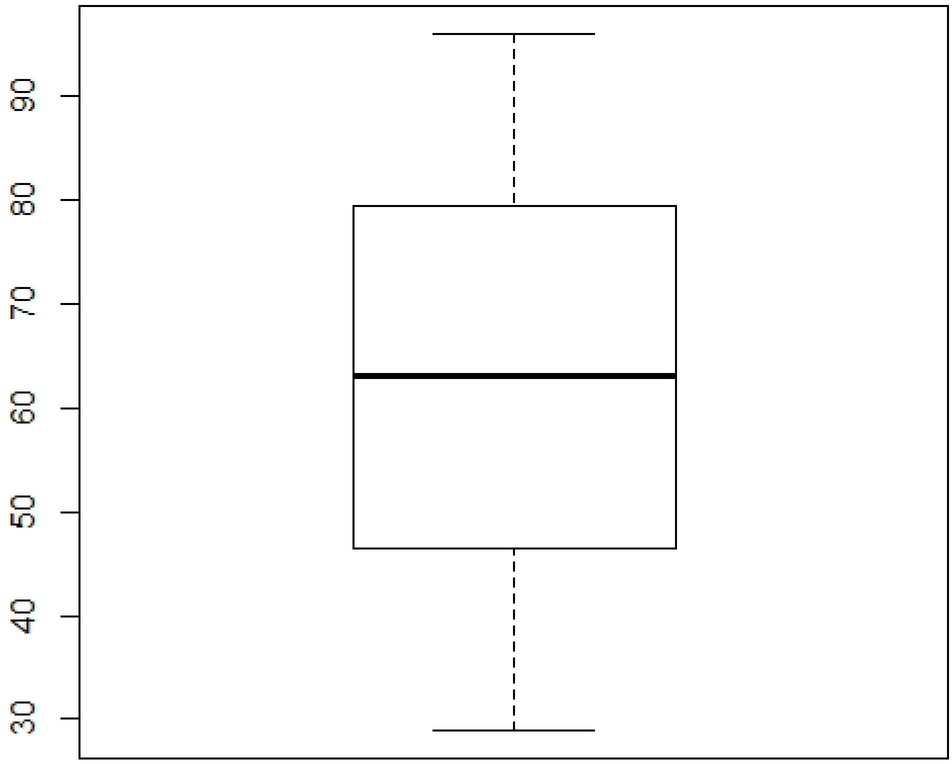
```
> exp<-c(6000,15000,3000,1500,4500)
> lbl<-c('Housing','Food','Cloths','Entertainment','Misc')
> pie(exp,labels=lbl,main='Pie Char of Expenditure')
```



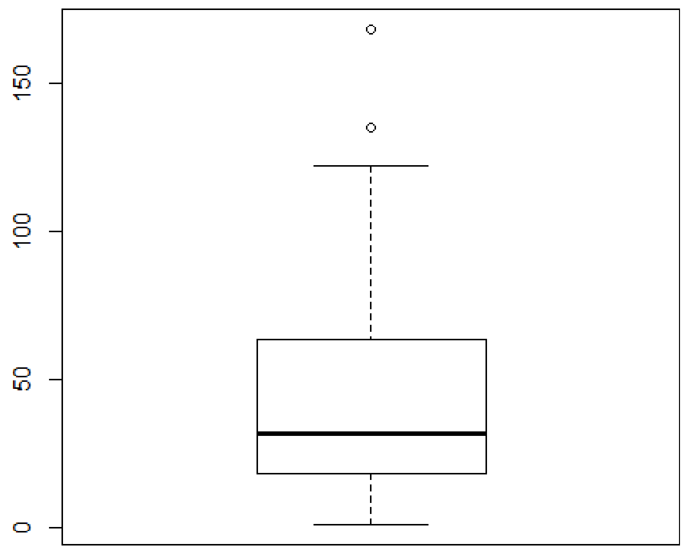
Boxplot




```
> marks <- c(68, 82, 63, 86, 34, 96, 41, 89, 29, 51, 75, 77, 56, 59, 42)
> boxplot(marks)
```

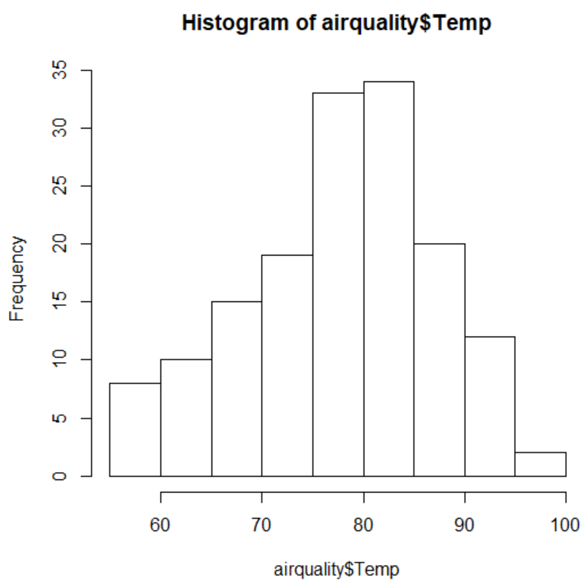


```
> str(airquality)
'data.frame':   153 obs. of  6 variables:
 $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
 $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
 $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
 $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
> boxplot(airquality$Ozone)
```

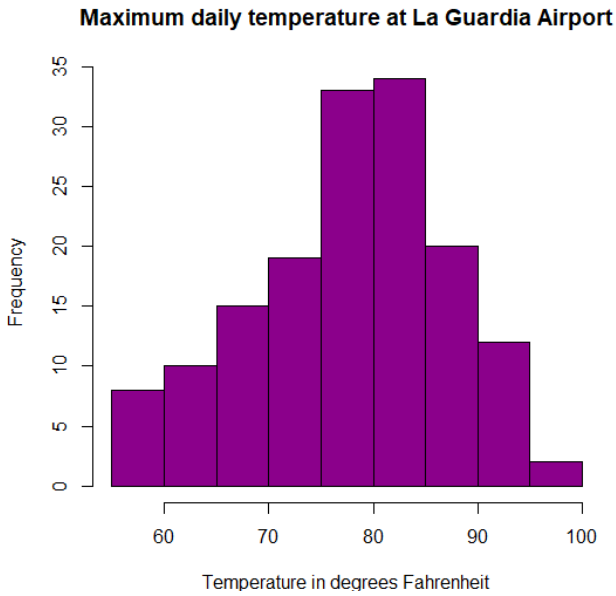


Histogram

```
> str(airquality)
'data.frame':   153 obs. of  6 variables:
 $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
 $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
 $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
 $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
> hist(airquality$Temp)
```

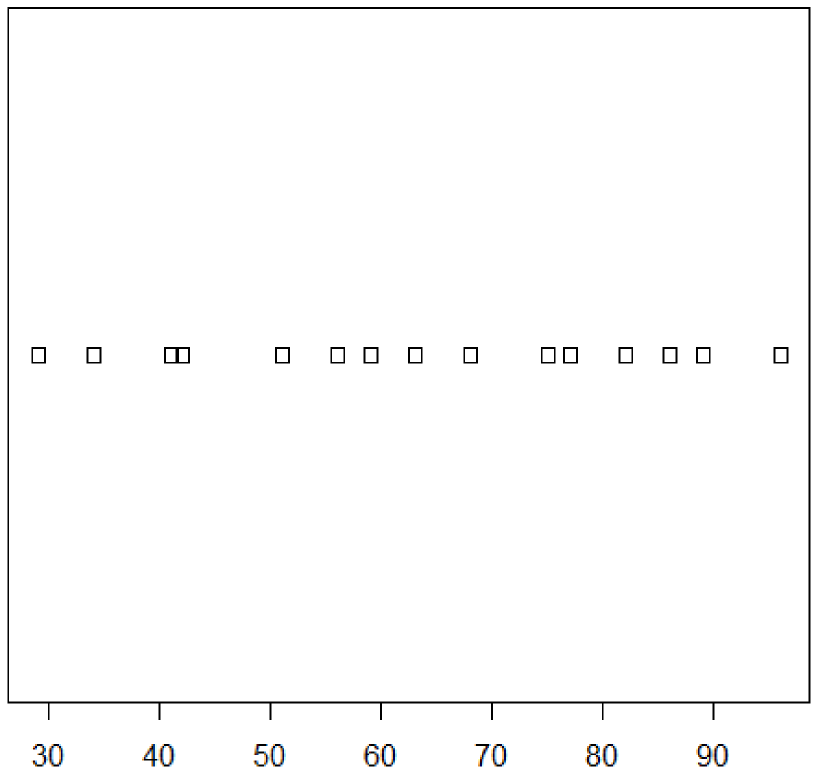


```
> # histogram with added parameters
> hist(airquality$Temp,
+ main="Maximum daily temperature at La Guardia Airport",
+ xlab="Temperature in degrees Fahrenheit",
+ col="darkmagenta")
```



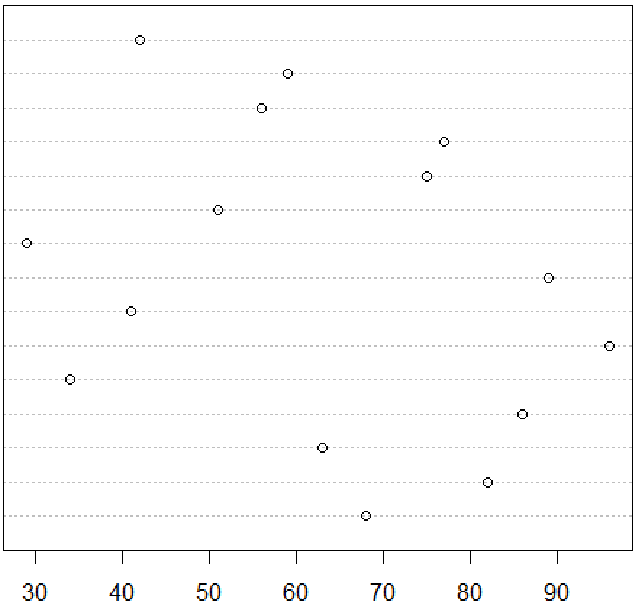
Stripchart

```
> stripchart(marks)
```



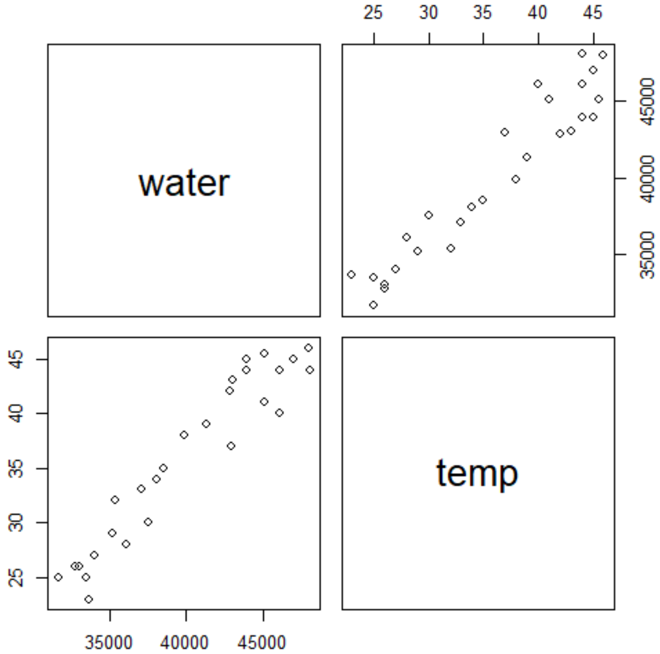
Dotchart

```
> dotchart(marks)
```



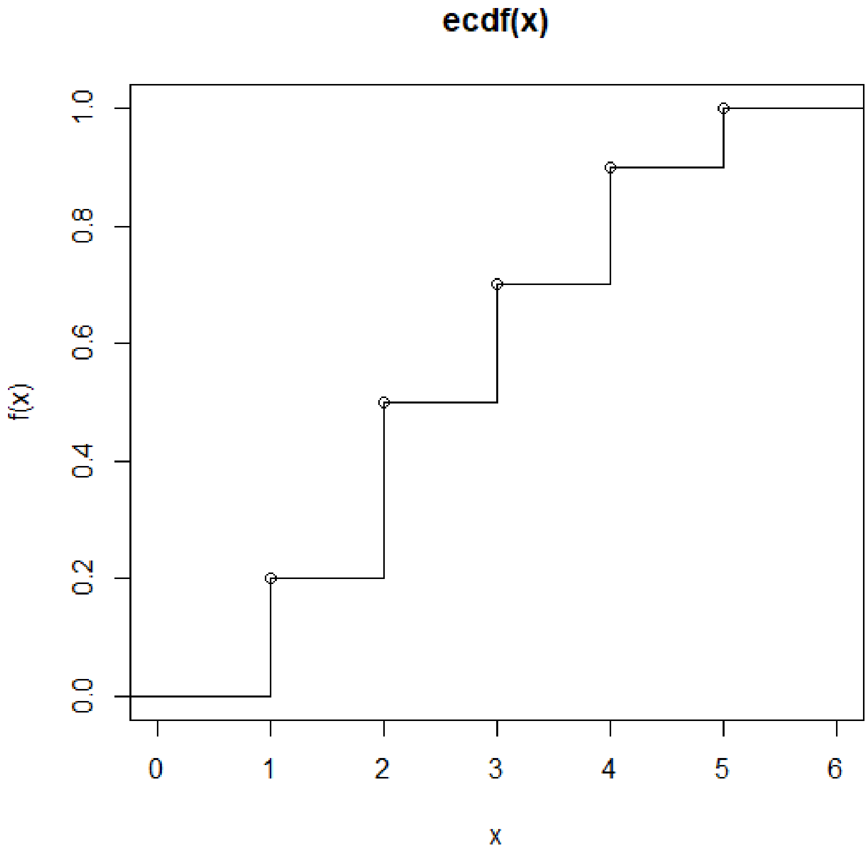
PAIRS:

```
> water <- c(33710,31666,33495,32758,34067,36069, 37497,33044,35216, 35383,
+ 37066,38037,38495, 39895,41311,42849,43038,43873,43923, 45078, 46935,
+ 47951,46085,48003,45050,42924,46061)
> temp <- c(23,25,25,26,27,28,30,26,29,32,33,34, 35,38,39,42,43,44, 45,
+ 45.5,45,46,44,44,41,37,40)
> pairs(cbind(water,temp))
```

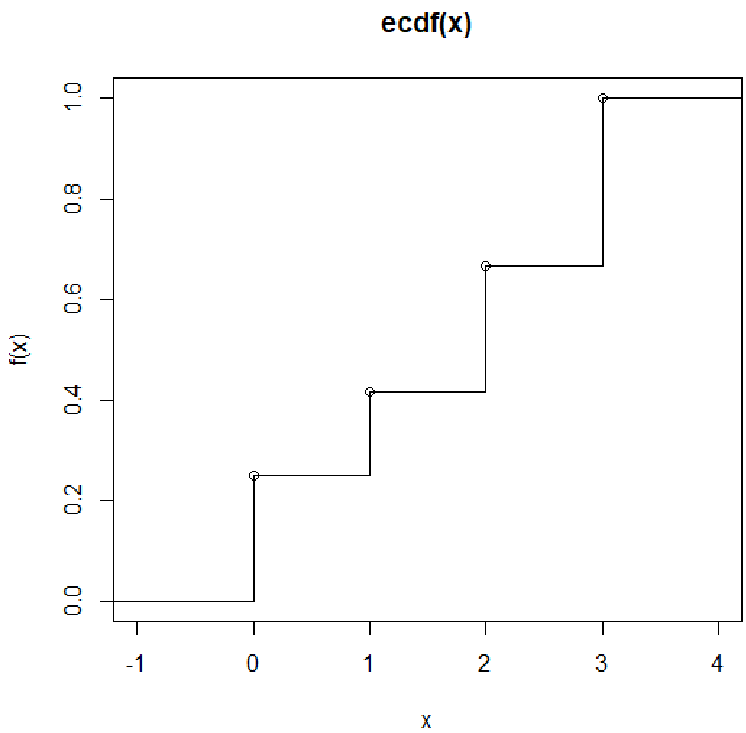


Step Function Plot

```
> x<-c(2,4,2,1,3,4,2,1,3,5)
> plot.stepfun(x)
```

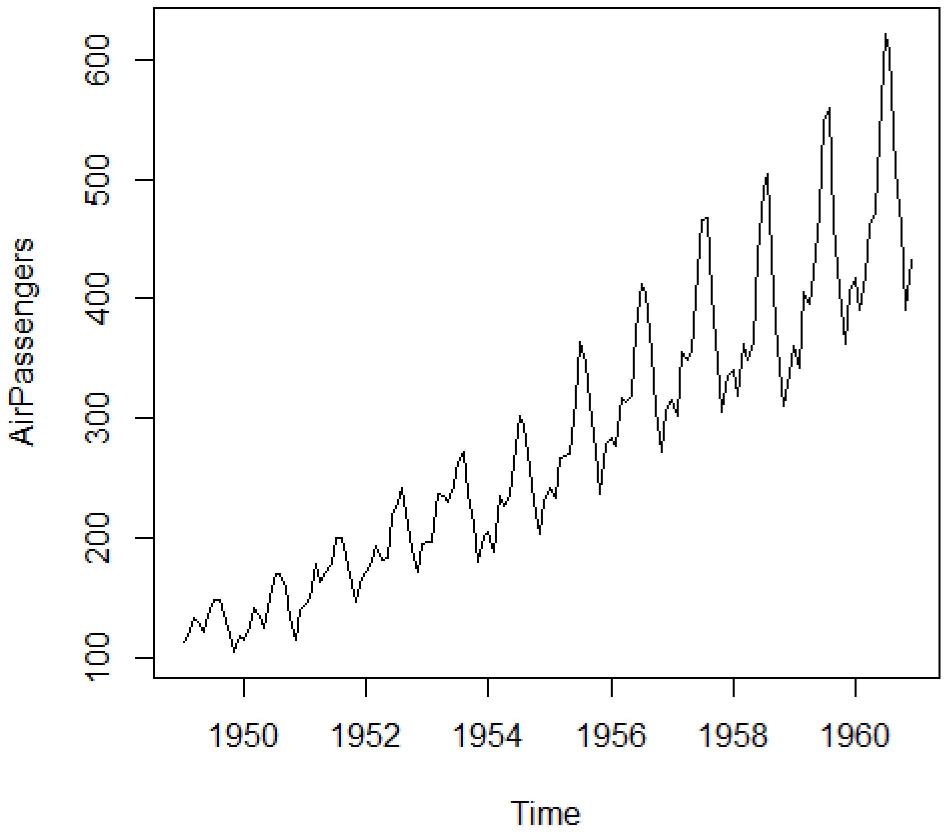



```
> x<-c(0,3,2,1,3,3,1,2,0,0,3,2,1,3,2,3,0,3,2,0,3,2,0,1)
> plot.stepfun(x)
```



Time Series Plot

```
> str(AirPassengers)
 Time-Series [1:144] from 1949 to 1961: 112 118 132 129 121 135 148 148
> plot.ts(AirPassengers)
```




```
> Aff_Seg<-c(173,153,195,147,120,144,148,109,174,130,172,131)
> Lux_Seg<-c(189,189,105,112,173,109,151,197,174,145,177,161)
> SLux_Seg<-c(185,185,126,134,196,153,112,133,200,145,167,110)
> data<-matrix(c(Aff_Seg,Lux_Seg,SLux_Seg),ncol=3)
> colnames<-c("Aff_Seg","Lux_Seg","SLux_Seg")
> rownames<-c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")
> data<-matrix(c(Aff_Seg,Lux_Seg,SLux_Seg),ncol=3,dimnames=list(rownames,colnames))
> t<-ts(data,frequency=12,start=2020)
> t
         Aff_Seg Lux_Seg SLux_Seg
Jan 2020     173     189      185
Feb 2020     153     189      185
Mar 2020     195     105      126
Apr 2020     147     112      134
May 2020     120     173      196
Jun 2020     144     109      153
Jul 2020     148     151      112
Aug 2020     109     197      133
Sep 2020     174     174      200
Oct 2020     130     145      145
Nov 2020     172     177      167
Dec 2020     131     161      110
> plot.ts(t)
```

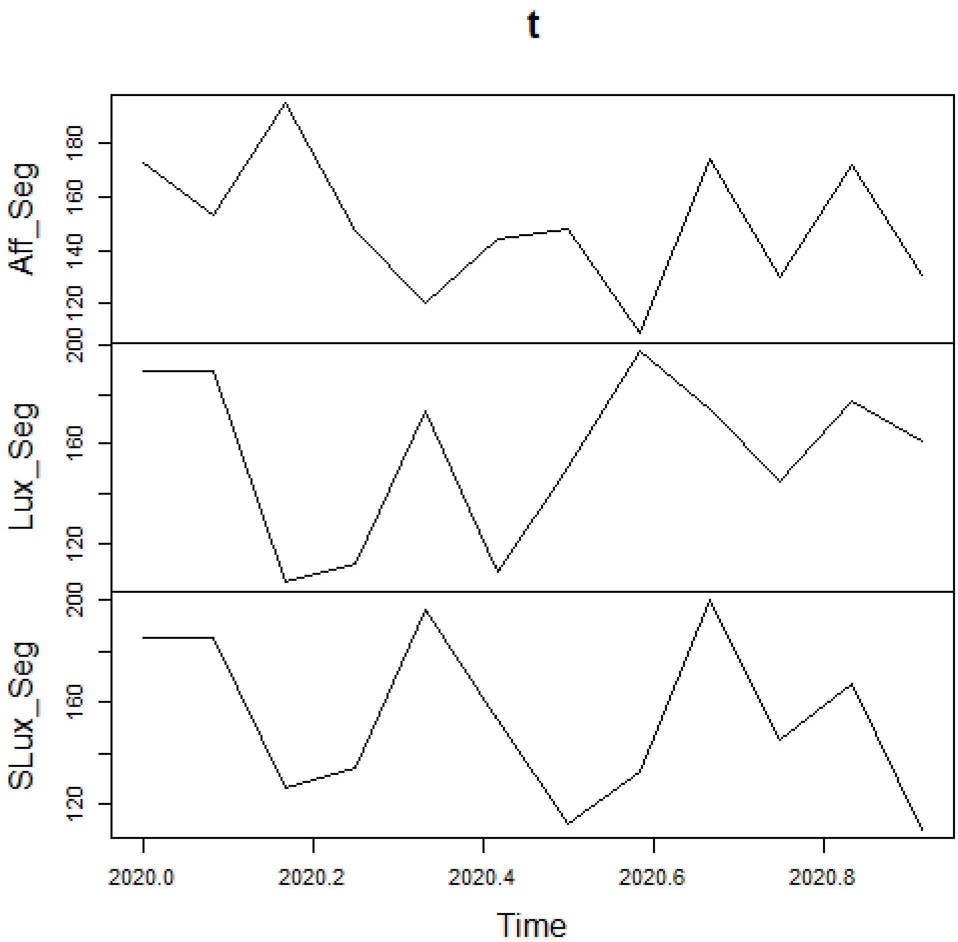
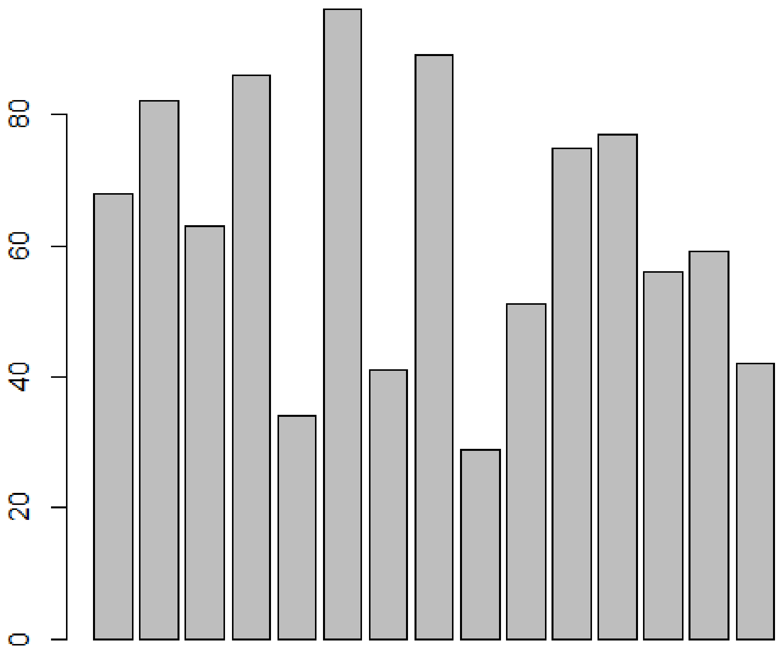


Table 5-2: Selected High-Level Plotting Functions

Function name	Plot produced
boxplot(x)	"Box and whiskers" plot
pie(x)	Circular pie chart
hist(x)	Histogram of the frequencies of x
barplot(x)	Histogram of the values of x
stripchart(x)	Plots values of x along a line
dotchart(x)	Cleveland dot plot
pairs(x)	For a matrix x, plots all bivariate pairs
plot.ts(x)	Plot of x with respect to time (index values of the vector unless specified)
contour(x,y,z)	Contour plot of vectors x and y, z must be a matrix of dimension rows=x and columns=y
image(x,y,z)	Same as contour plot but uses colors instead of lines
persp(x,y,z)	3-d contour plot

BARPLOT:

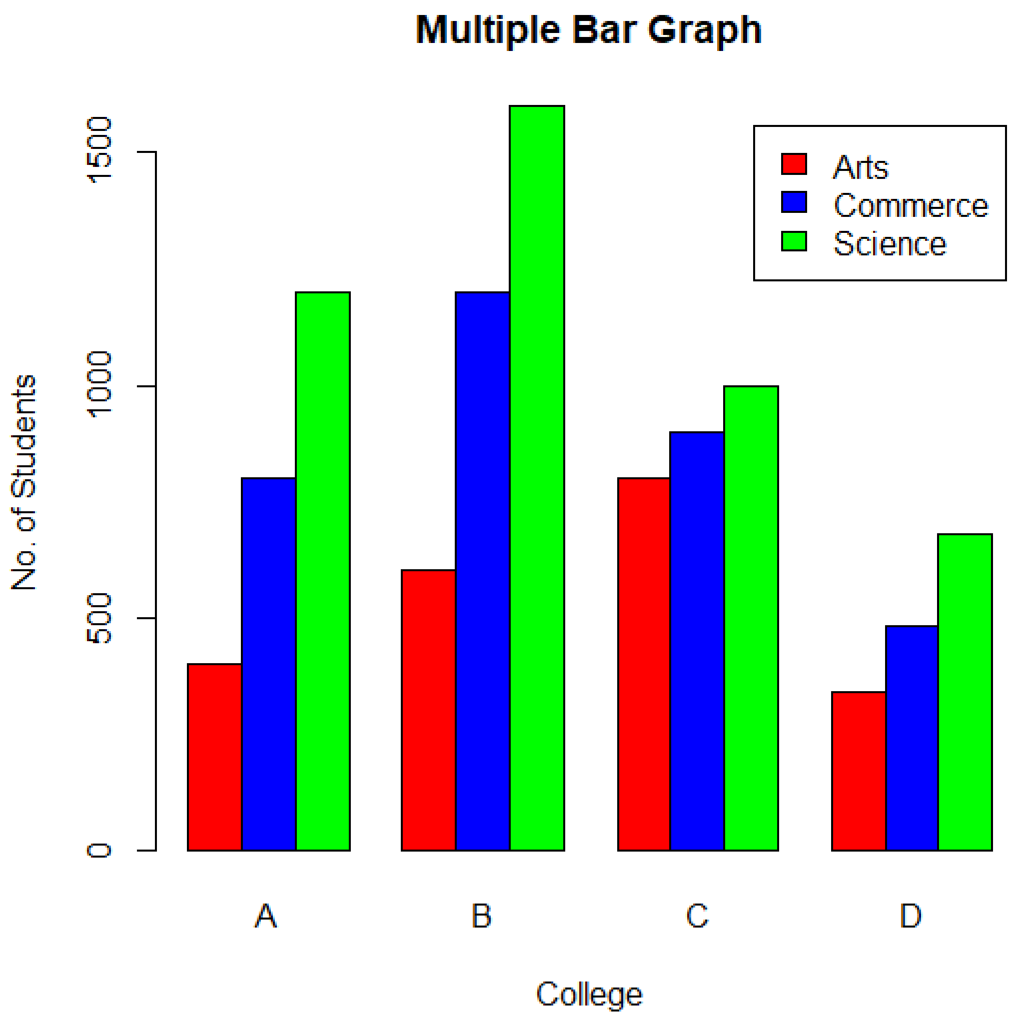
```
> marks <- c(68, 82, 63, 86, 34, 96, 41, 89, 29, 51, 75, 77, 56, 59, 42)
> barplot(marks)
```



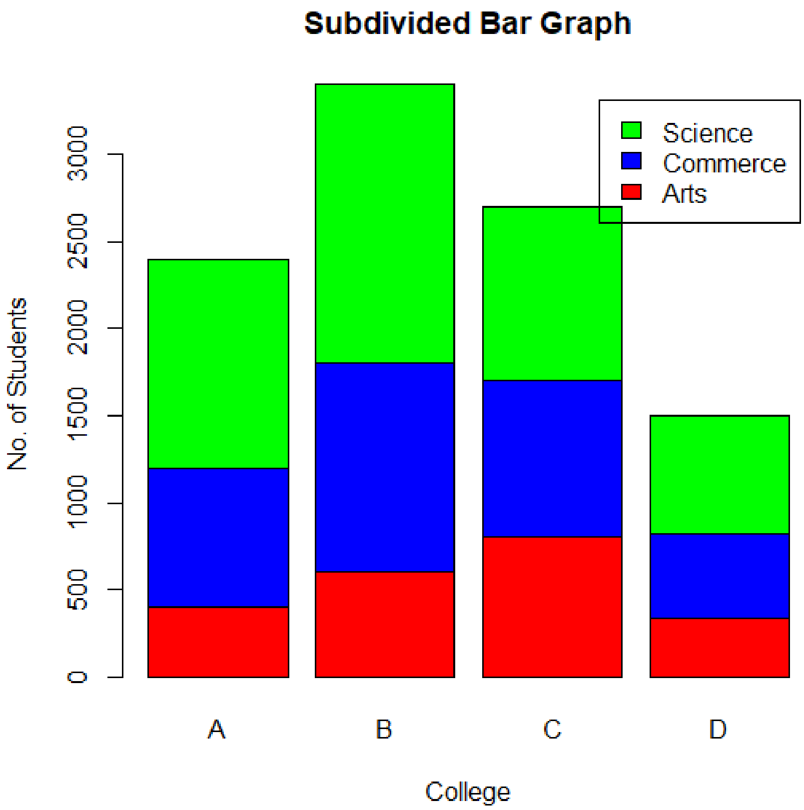
- Represent the following data by multiple bar diagram and subdivided bar diagram.

College	Arts	Commerce	Science
A	400	800	1200
B	600	1200	1600
C	800	900	1000
D	340	480	680

```
> x<-c(400,800,1200,600,1200,1600,800,900,1000,340,480,680)
> data<-matrix(x,3)
> barplot(data,beside=T,names.arg=c('A','B','C','D'),
+ col=c('Red','Blue','Green'),legend=c('Arts','Commerce','Science'),
+ xlab='College',ylab='No. of Students',main='Multiple Bar Graph')
```



```
> x<-c(400,800,1200,600,1200,1600,800,900,1000,340,480,680)
> data<-matrix(x,3)
> barplot(data,beside=F,names.arg=c('A','B','C','D'),
+ col=c('Red','Blue','Green'),legend=c('Arts','Commerce','Science'),
+ xlab='College',ylab='No. of Students',main='Subdivided Bar Graph')
```



Practical No. 5

MEASURES OF CENTRAL TENDENCY

SOURCE CODE & OUTPUT:

1. **Mean**

```
> # Finding Mean
> x<-c(84,91,72,68,87,78)
> mean(x)
[1] 80
>
> y<-c(2,3,4,11,14,17,23,25,27,28,80,84,88)
> mean(y)
[1] 31.23077
>
> mean(y,trim=0.3)
[1] 20.71429
>
> z<-c(11,12,36,17,19,25,34,47,9,22,NA)
> mean(z)
[1] NA
> mean(z,na.rm=TRUE)
[1] 23.2
```

2. **Median**

```
> # Finding Median
> x<-c(84,91,72,68,87,78)
> median(x)
[1] 81
>
> y<-c(2,3,4,11,14,17,23,25,27,28,80,84,88)
> median(y)
[1] 23
>
> z<-c(11,12,36,17,19,25,34,47,9,22,NA)
> median(z)
[1] NA
> median(z,na.rm=TRUE)
[1] 20.5
```

3. Mode

```
> # Creating getMode function
> getMode<-function(x)
+ {
+ u<-unique(x)
+ u[which.max(tabulate(match(x,u)))]
+ }
>
> # Finding Mode of Vector with Numeric Values
> x<-c(11,14,17,16,16,16,17,17,13,13,13,13)
> getMode(x)
[1] 13
>
> # Finding Mode of Vector with Character Values
> y<-c('IT','IT','CS','PM','CS','OS','IT','PM')
> getMode(y)
[1] "IT"
```

4. Quartiles

```
> # Finding First Quartile
> v<-c(11,12,36,17,19,25,34,47,9,22)
> Q1<-quantile(v,prob=0.25)
> cat('First quartile is:',Q1,'\n')
First quartile is: 13.25
>
> # Finding Second Quartile
> Q2<-quantile(v,prob=0.5)
> cat('Second quartile is:',Q2,'\n')
Second quartile is: 20.5
>
> # Finding Third Quartile
> Q3<-quantile(v,prob=0.75)
> cat('Third quartile is:',Q3,'\n')
Third quartile is: 31.75
>
> # Finding Quantiles
> quantile(v)
   0%   25%   50%   75%  100% 
 9.00 13.25 20.50 31.75 47.00 
```

5. Deciles and Percentiles

```
> # Finding Fourth Decile
> D4<-quantile(v,prob=0.4)
> cat('Fourth decile is:',D4,'\n')
Fourth decile is: 18.2
>
> # Finding Sixty Fifth Percentile
> P65<-quantile(v,prob=0.65)
> cat('Sixty Fifth Percentile is:',P65,'\n')
Sixty Fifth Percentile is: 24.55
```

6. Performing Above Statistical Functions on 'faithful' Dataset

```
> # Finding Mean of eruptions column of faithful Dataset
> mean(faithful$eruptions)
[1] 3.487783
>
> # Finding Mean of waiting column of faithful Dataset
> mean(faithful$waiting)
[1] 70.89706
>
> # Finding Median of eruptions column of faithful Dataset
> median(faithful$eruptions)
[1] 4
>
> # Finding Median of waiting column of faithful Dataset
> median(faithful$waiting)
[1] 76
```

```
> # Creating getMode function
> getMode<-function(x)
+ {
+ u<-unique(x)
+ u[which.max(tabulate(match(x,u)))]
+ }
>
> # Finding Mode of eruption column of faithful Dataset
> getMode(faithful$eruption)
[1] 1.867
>
> # Finding Mode of waiting column of faithful Dataset
> getMode(faithful$waiting)
[1] 78
```

```
> # Finding First and Third Quartile of eruptions column of faithful Dataset
> quantile(faithful$eruptions,prob=0.25)
    25%
2.16275
> quantile(faithful$eruptions,prob=0.75)
    75%
4.45425
>
> # Finding First and Third Quartile of waiting column of faithful Dataset
> quantile(faithful$waiting,prob=0.25)
25%
 58
> quantile(faithful$waiting,prob=0.75)
75%
 82
> # Finding Fourth Decile and Sixty Fifth Percentile of eruptions column of faithful Dataset
> quantile(faithful$eruptions,prob=0.4)
40%
3.6
> quantile(faithful$eruptions,prob=0.65)
    65%
4.28555
>
> # Finding Fourth Decile and Sixty Fifth Percentile of waiting column of faithful Dataset
> quantile(faithful$waiting,prob=0.4)
40%
 71
> quantile(faithful$waiting,prob=0.65)
65%
 79
```

7. Performing Above Statistical Functions on CSV Data

- Find working directory by using ***getwd()***.
- Copy and paste .csv file in working directory.
- Import data from .csv file by using ***read.csv()***.

```
> getwd()
[1] "C:/Users/Ajay/Desktop/R Programming/New folder"
> emp<-read.csv("employee.csv")
> emp
   Emp_Id Emp_Name        DOJ Salary Department
1    1001    Sadik 07-06-2012  47000    Finance
2    1002    Pinky 15-11-2012  45000         HR
3    1003    Manoj 03-03-2013  43000 Operations
4    1004     Aman 27-08-2013  38000         IT
5    1005    Sonam 15-12-2013  29000      Admin
6    1006   Rajesh 04-10-2014  23000      Admin
7    1007   Ramesh 04-10-2014  41000         HR
8    1008  Radhika 07-10-2014  40000 Operations
9    1009   Manish 17-10-2014  25000         IT
10   1010   Ritika 17-10-2014  33000         HR
11   1011    Aryan 28-10-2014  28000 Operations
12   1012     Ayan 28-10-2014  39000    Finance
13   1013   Suyash 07-11-2014  23000         IT
14   1014   Naresh 07-11-2014  27000      Admin
15   1015    Jyoti 09-11-2014  25000      Admin
```

```
> # Finding Mean
> cat('Mean Salary =',mean(emp$Salary),'\n')
Mean Salary = 33733.33
>
> # Finding Meadian
> cat('Median Salary =',median(emp$Salary),'\n')
Median Salary = 33000
>
> # Finding Quartile
> cat('First Quartile =',quantile(emp$Salary,prob=0.25),'\n')
First Quartile = 26000
> cat('Third Quartile =',quantile(emp$Salary,prob=0.75),'\n')
Third Quartile = 40500
>
> # Finding Decile and Percentile
> cat('Fourth Decile =',quantile(emp$Salary,prob=0.4),'\n')
Fourth Decile = 28600
> cat('Sixty Fifth Percentile =',quantile(emp$Salary,prob=0.65),'\n')
Sixty Fifth Percentile = 39100
```

```
> # Creating getMode function
> getMode<-function(x)
+ {
+ u<-unique(x)
+ u[which.max(tabulate(match(x,u)))]
+ }
>
> # Finding Mode
> cat('Mode of Salary =',getMode(emp$Salary),'\n')
Mode of Salary = 23000
```

8. Performing Above Statistical Functions on Excel Data

- Copy and paste .xlsx file in working directory.
- Install xlsx package by using ***install.packages("xlsx")***.
- Add xlsx library by using ***library("xlsx")***.
- Import data from .csv file by using ***read.xlsx()***.

```
> install.packages("xlsx")
Installing package into 'C:/Users/Ajay/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
trying URL 'https://cran.ma.imperial.ac.uk/bin/windows/contrib/3.4/xlsx_0.6.1.z$
Content type 'application/zip' length 405318 bytes (395 KB)
downloaded 395 KB

package 'xlsx' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Ajay\AppData\Local\Temp\RtmpU5exRJ\downloaded_packages
> library("xlsx")
Warning message:
package 'xlsx' was built under R version 3.4.4
```

```
> emp2<-read.xlsx('employee.xlsx',sheetIndex=1)
> emp2
   Emp_Id Emp_Name        DOJ Salary Department
1    1001    Sadik 2012-06-07  47000    Finance
2    1002    Pinky 2012-11-15  45000         HR
3    1003    Manoj 2013-03-03  43000 Operations
4    1004     Aman 2013-08-27  38000         IT
5    1005    Sonam 2013-12-15  29000      Admin
6    1006   Rajesh 2014-10-04  23000      Admin
7    1007   Ramesh 2014-10-04  41000         HR
8    1008  Radhika 2014-10-07  40000 Operations
9    1009   Manish 2014-10-17  25000         IT
10   1010   Ritika 2014-10-17  33000         HR
11   1011    Aryan 2014-10-28  28000 Operations
12   1012     Ayan 2014-10-28  39000    Finance
13   1013   Suyash 2014-11-07  23000         IT
14   1014   Naresh 2014-11-07  27000      Admin
15   1015    Jyoti 2014-11-09  25000      Admin
```

```
> # Finding Mean
> cat('Mean Salary =',mean(emp2$Salary),'\n')
Mean Salary = 33733.33
>
> # Finding Meadian
> cat('Median Salary =',median(emp2$Salary),'\n')
Median Salary = 33000
>
> # Finding Quartile
> cat('First Quartile =',quantile(emp2$Salary,prob=0.25),'\n')
First Quartile = 26000
> cat('Third Quartile =',quantile(emp2$Salary,prob=0.75),'\n')
Third Quartile = 40500
>
> # Finding Decile and Percentile
> cat('Fourth Decile =',quantile(emp2$Salary,prob=0.4),'\n')
Fourth Decile = 28600
> cat('Sixty Fifth Percentile =',quantile(emp2$Salary,prob=0.65),'\n')
Sixty Fifth Percentile = 39100
```

```
> # Creating getMode function
> getMode<-function(x)
+ {
+ u<-unique(x)
+ u[which.max(tabulate(match(x,u)))]
+ }
>
> # Finding Mode
> cat('Mode of Salary =',getMode(emp2$Salary),'\n')
Mode of Salary = 23000
```

Practical No. 6

MEASURES OF DISPERSION

SOURCE CODE & OUTPUT:

1. **Range**

```
> # Finding Range
> v<-c(11,12,36,17,19,25,34,47,9,22)
> range<-max(v)-min(v)
> cat('Range is:',range,'\n')
Range is: 38
```

2. **Inter-Quartile Range and Semi-Interquartile Range**

```
> v<-c(11,12,36,17,19,25,34,47,9,22)
> quantile(v)
   0%   25%   50%   75%  100%
 9.00 13.25 20.50 31.75 47.00
>
> iqr<-IQR(v)
> cat('Inter-Quartile Range is:',iqr,'\n')
Inter-Quartile Range is: 18.5
>
> siqr<-IQR(v)/2
> cat('Semi-Interquartile Range is:',siqr,'\n')
Semi-Interquartile Range is: 9.25
```

3. **Mean Deviation**

```
> v<-c(11,12,36,17,19,25,34,47,9,22)
> # Finding Mean
> xbar<-mean(v)
> xbar
[1] 23.2
>
> n<-length(v)
>
> # Finding Mean Deviation from Mean
> MDfromMean<-sum(abs(v-xbar))/n
> cat('Mean Deviation from mean is:',MDfromMean,'\n')
Mean Deviation from mean is: 9.84
```

```
> # Finding Median
> M<-median(v)
> M
[1] 20.5
>
> # Finding Mean Deviation from Median
> MDfromMedian<-sum(abs(v-M))/n
> cat('Mean Deviation from median is:',MDfromMedian,'\n')
Mean Deviation from median is: 9.6
```

4. Variance, Standard Deviation and Coefficient of Variaion

```
> v<-c(11,12,36,17,19,25,34,47,9,22)
>
> # Finding Variance
> variance<-var(v)
> cat('Variance is:',variance,'\n')
Variance is: 153.7333
>
> # Finding Standard Deviation
> stdDev<-sd(v)
> cat('Standard Deviation is:',stdDev,'\n')
Standard Deviation is: 12.39892
>
> # Finding Coefficient of Variation
> cv<-sd(v)/mean(v)*100
> cat('Coefficient of Variation is:',cv,'\n')
Coefficient of Variation is: 53.44364
```

```
> # Finding Consistency of Data
> StudentA<-c(74,75,80,78,72,77)
> StudentB<-c(86,84,80,88,87,85)
>
> # Finding Means
> mean(StudentA)
[1] 76
> mean(StudentB)
[1] 85
>
> # Finding Standard Deviation
> sd(StudentA)
[1] 2.898275
> sd(StudentB)
[1] 2.828427
>
> # Finding Coefficient of Variation
> CVA<-sd(StudentA)/mean(StudentA)*100
> CVA
[1] 3.81352
> CVB<-sd(StudentB)/mean(StudentB)*100
> CVB
[1] 3.327561
>
> # Comparing
> if(CVA<CVB){
+ cat('Performance of student A is more consistant.\n')
+ }else{cat('Performance of student B is more consistant.\n')}
Performance of student B is more consistant.
```

5. Performing Above Statistical Functions on ‘faithful’ Dataset

```
> # Finding Range of eruptions column of faithful dataset
> max(faithful$eruptions)-min(faithful$eruptions)
[1] 3.5
>
> # Finding Range of waiting column of faithful dataset
> max(faithful$waiting)-min(faithful$waiting)
[1] 53
>
> # Finding Inter-Quartile Range of eruptions column of faithful dataset
> IQR(faithful$eruptions)
[1] 2.2915
>
> # Finding Inter-Quartile Range of waiting column of faithful dataset
> IQR(faithful$waiting)
[1] 24
>
> # Finding Semi-Interquartile Range of eruptions column of faithful dataset
> IQR(faithful$eruptions)/2
[1] 1.14575
>
> # Finding Semi-Interquartile Range of waiting column of faithful dataset
> IQR(faithful$waiting)/2
[1] 12
```

```
> # Mean Deviation from Mean
> xbar<-mean(faithful$eruptions)
> xbar
[1] 3.487783
> n<-length(faithful$eruptions)
> MDfromMean<-sum(abs(faithful$eruptions-xbar))/n
> MDfromMean
[1] 1.042253
>
> # Mean Deviation from Median
> M<-median(faithful$eruptions)
> M
[1] 4
> MDfromMedian<-sum(abs(faithful$eruptions-M))/n
> MDfromMedian
[1] 0.9724669
```

```
> # Variance
> var(faithful$eruptions)
[1] 1.302728
>
> # Standard Deviation
> sd(faithful$eruptions)
[1] 1.141371
>
> # Coefficient of Variation
> sd(faithful$eruptions)/mean(faithful$eruptions)*100
[1] 32.72483
```

6. Performing Above Statistical Functions on CSV/Excel Data

Left as exercise

Practical No. 7

HYPOTHESIS TESTING

Q.1 The mean lifetime of electric light bulbs produced by a company has in the past been 1120h with a standard deviation of 125h. A sample of 8 electric bulbs recently chosen from supply of newly produced bulb showed a mean lifetime of 1030h. Test the hypothesis that the mean lifetime of the bulb has not changed at 0.05 significance level.

```
> # Two Tailed Problem
> # H0: mu=1120 vs H1: mu!=1120
> mu<-1120
> sigma<-125
> n<-8
> xbar<-1030
> tCal<-(xbar-mu)/(sigma/sqrt(n-1))
> tCal
[1] -1.904941
> alpha<-0.05
> df<-n-1
> tTab<-qt(1-alpha/2,df)
> tTab
[1] 2.364624
> if(abs(tCal)<tTab)
+ {
+ print('Accept H0.')
+ print('Mean lifetime of bulb has not changed.')
+ }else
+ {
+ print('Reject H0.')
+ print('Mean lifetime of bulb has changed.')
+ }
[1] "Accept H0."
[1] "Mean lifetime of bulb has not changed."
```

Q.2 The mean lifetime of electric light bulbs produced by a company has in the past been 1120h with a standard deviation of 125h. A sample of 8 electric bulbs recently chosen from supply of newly produced bulb showed a mean lifetime of 1030h. Test the hypothesis that the mean lifetime of the bulb has decreased at 0.05 significance level.

```
> # Left Tailed Problem
> # H0: mu=1120 vs H1: mu<1120
> mu<-1120
> sigma<-125
> n<-8
> xbar<-1030
> tCal<-(xbar-mu)/(sigma/sqrt(n-1))
> tCal
[1] -1.904941
> alpha<-0.05
> df<-n-1
> tTab<-qt(1-alpha,df)
> tTab
[1] 1.894579
> if(abs(tCal)<tTab)
+ {
+ print('Accept H0.')
+ print('Mean lifetime of bulb has not decreased.')
+ }else
+ {
+ print('Reject H0.')
+ print('Mean lifetime of bulb has decreased.')
+ }
[1] "Reject H0."
[1] "Mean lifetime of bulb has decreased."
```

Q.3 The mean lifetime of electric light bulbs produced by a company has in the past been 1120h with a standard deviation of 125h. A sample of 8 electric bulbs recently chosen from supply of newly produced bulb showed a mean lifetime of 1200h. Test the hypothesis that the mean lifetime of the bulb has increased at 0.05 significance level.

```
> # Right Tailed Problem
> # H0: mu=1120 vs H1: mu>1120
> mu<-1120
> sigma<-125
> n<-8
> xbar<-1200
> tCal<-(xbar-mu)/(sigma/sqrt(n-1))
> tCal
[1] 1.693281
> alpha<-0.05
> df<-n-1
> tTab<-qt(1-alpha,df)
> tTab
[1] 1.894579
> if(abs(tCal)<tTab)
+ {
+ print('Accept H0.')
+ print('Mean lifetime of bulb has not increased.')
+ }else
+ {
+ print('Reject H0.')
+ print('Mean lifetime of bulb has increased.')
+ }
[1] "Accept H0."
[1] "Mean lifetime of bulb has not increased."
```

Q.4 Test the hypothesis $H_0: \mu = 3400\ vs\ H_1: \mu < 3400$ for the following data at 5% LOS.

3366, 3337, 3361, 3410, 3316, 3357, 3348, 3356, 3376, 3382, 3377, 3355, 3408, 3401, 3390, 3424, 3383, 3374, 3384, 3390.

t.test(x,y,mu,alt,conf.level,paired)

where,

x:	**vector of observations**
y:	**vector of observations** **set NULL if not applicable**
mu:	**specified value of true mean**
alt:	**alternative hypothesis (can take value 'less', 'greater', 'two.sided')**
conf.level:	**confidence level (1-alpha)** **default value is 0.95**
paired:	**can take value TRUE or FALSE** **default value is FALSE**

```
> # Student t-Test
> # Left Tailed Problem
> # H0: mu=3400 vs H1: mu<3400
> x<- c(3366,3337,3361,3410,3316,3357,3348,3356,3376,
+ 3382,3377,3355,3408,3401,3390,3424,3383,3374,3484,3390)
> y<-NULL
> mu<-3400
>
> tTest<-t.test(x,y,mu,alt="less")
> tTest

        One Sample t-test

data:  x
t = -2.5268, df = 19, p-value = 0.01027
alternative hypothesis: true mean is less than 3400
95 percent confidence interval:
     -Inf 3393.607
sample estimates:
mean of x
  3379.75

> names(tTest)
[1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
[6] "null.value"  "alternative" "method"      "data.name"
```

```
> tTest$statistic
        t
-2.526799
> tTest$parameter
df
19
> tTest$p.value
[1] 0.01027214
> tTest$conf.int
[1]      -Inf 3393.607
attr(,"conf.level")
[1] 0.95
> tTest$estimate
mean of x
  3379.75
> tTest$null.value
mean
3400
> tTest$alternative
[1] "less"
> tTest$method
[1] "One Sample t-test"
> tTest$data.name
[1] "x"

> if(tTest$p.value<0.05)
+ {
+ print('Reject H0 i.e. population mean is less than 3400.')
+ }else
+ {
+ print('Accept H0 i.e. population mean is 3400.')
+ }
[1] "Reject H0 i.e. population mean is less than 3400."
```

Q.5 The following data refer to the amount of coffee (in ounces) filled by a machine in six randomly picked jars: 15.7, 15.9, 16.3, 16.2, 15.7 and 15.9. Is the true mean amount of coffee in a jar is 16 ounces? Use LOS 5%.

```
> # Student t-Test
> # Two Tailed Problem
> # H0: mu=16 vs H1: mu!=16
> x<-c(15.7,15.9,16.3,16.2,15.7,15.9)
> y<-NULL
> mu<-16
>
> tTest<-t.test(x,y,mu,alt="two.sided")
> tTest

        One Sample t-test

data:  x
t = -0.48795, df = 5, p-value = 0.6462
alternative hypothesis: true mean is not equal to 16
95 percent confidence interval:
 15.68659 16.21341
sample estimates:
mean of x
    15.95
> if(tTest$p.value<0.05)
+ {
+ print('Reject H0 i.e. mean amount of coffee in a jar is not 16.')
+ }else
+ {
+ print('Accept H0 i.e. mean amount of coffee in a jar is 16.')
+ }
[1] "Accept H0 i.e. mean amount of coffee in a jar is 16."
```

Q.6 Below are given the gain in weights (in lbs) of pigs fed on two diets A and B.
Diet A: 25,32,30,43,24,14,32,24,31,31,35,25
Diet B: 44,34,22,10,47,31,40,30,32,35,18,21,35,29,22
Test, if the two diets differ significantly as regards their effect on increase in weight. Use LOS 5%.

```
> # Student t-Test for double mean
> # Two Tailed Problem
> # H0: No significant difference between means of x and y vs
> # H1: Significant difference between means of x and y
> x<-c(25,32,30,34,24,14,32,24,30,31,35,25)
> y<-c(44,34,22,10,47,31,40,30,32,35,18,21,35,29,22)
>
> tTest<-t.test(x,y,var.equal=T)
> tTest

        Two Sample t-test

data:  x and y
t = -0.61028, df = 25, p-value = 0.5472
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -8.749507  4.749507
sample estimates:
mean of x mean of y
       28        30
> if(tTest$p.value<0.05)
+ {
+ print('Reject H0 i.e. there is significant difference b/n means.')
+ }else
+ {
+ print('Accept H0 i.e. there is no significant difference b/n means.')
+ }
[1] "Accept H0 i.e. there is no significant difference b/n means."
```

Q.7 Eleven school boys were given a test in mathematics. They were given a month's tuition and a second test was held at the end of it. Do the marks give evidence that the student's have benefited by the extra coaching? Use LOS 5%.

Marks in test-1: 23, 20, 19, 21, 18, 20, 18, 17, 23, 16, 19

Marks in test-2: 24, 19, 22, 18, 20, 22, 20, 20, 23, 20, 17

```
> # Paired t-Test
> # Two Tailed Problem
> # H0: No significant difference between x and y vs
> # H1: Significant difference between x and y
> x<-c(23,20,19,21,18,20,18,17,23,16,19)
> y<-c(24,19,22,18,20,22,20,20,23,20,17)
>
> tTest<-t.test(x,y,paired=T)
> tTest

        Paired t-test

data:  x and y
t = -1.4832, df = 10, p-value = 0.1688
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.5022109  0.5022109
sample estimates:
mean of the differences
                     -1
> if(tTest$p.value<0.05)
+ {
+ print('Reject H0 i.e. there is significant difference b/n x & y.')
+ }else
+ {
+ print('Accept H0 i.e. there is no significant difference b/n x & y.')
+ }
[1] "Accept H0 i.e. there is no significant difference b/n x & y."
```

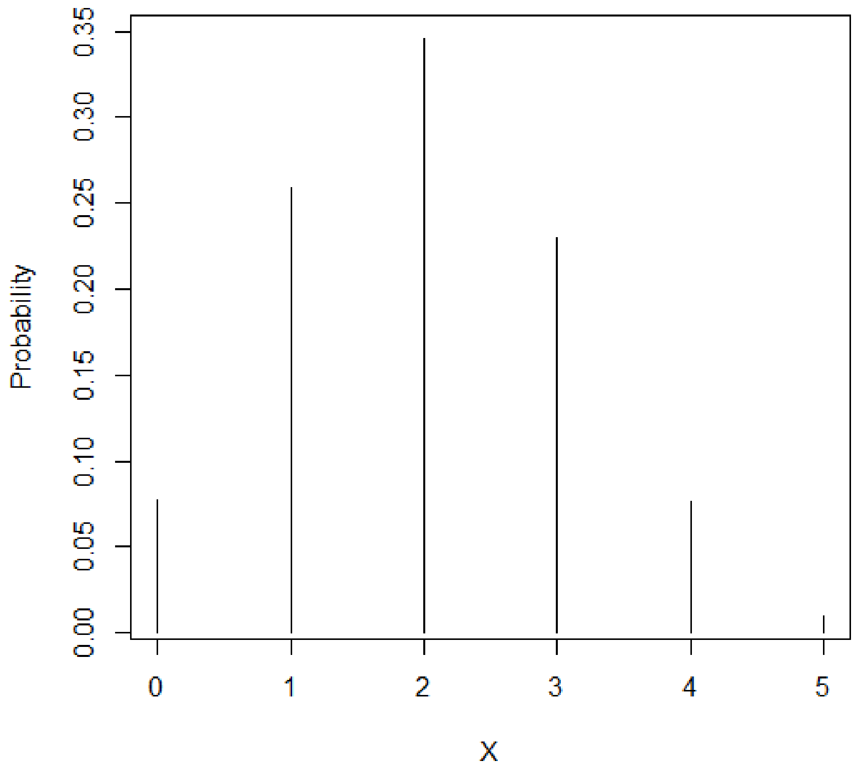
Practical No. 8

BINOMIAL DISTRIBUTION

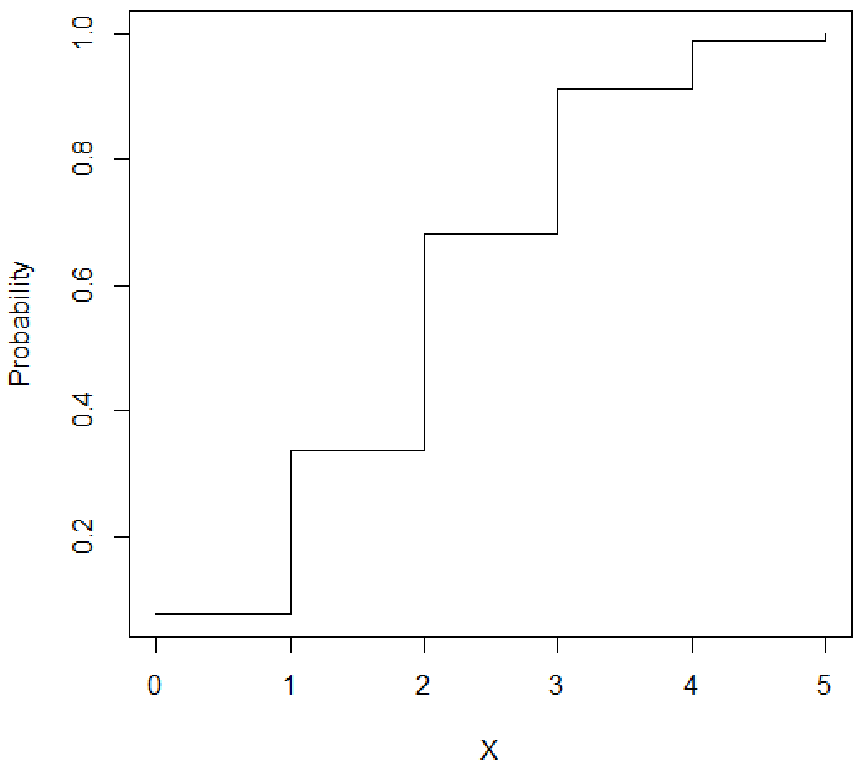
SOURCE CODE & OUTPUT:

```
> # Binomial Distribution
> # X~B(n=5,p=0.4)
> x<-0:5
> n<-5
> p<-0.4
>
> # Probability Distribution
> fx<-dbinom(x,n,p)
>
> # Cumulative Probability Distribution
> Fx<-pbinom(x,n,p)
>
> data.frame("x"=x,"Probability"=round(fx,5),
+ "Cumulative Probability"=round(Fx,5))
  x Probability Cumulative.Probability
1 0     0.07776                0.07776
2 1     0.25920                0.33696
3 2     0.34560                0.68256
4 3     0.23040                0.91296
5 4     0.07680                0.98976
6 5     0.01024                1.00000
> # P[X=2]
> dbinom(2,n,p)
[1] 0.3456
>
> # P[X<=3]
> pbinom(3,n,p)
[1] 0.91296
>
> # P[X<2]
> dbinom(0,n,p)+dbinom(1,n,p)
[1] 0.33696
>
> # P[X>=4]
> dbinom(4,n,p)+dbinom(5,n,p)
[1] 0.08704
```

```
> # High Density Vertical Lines Graph of P(x)
> plot(x,fx,type='h',xlab='X',ylab='Probability')
```

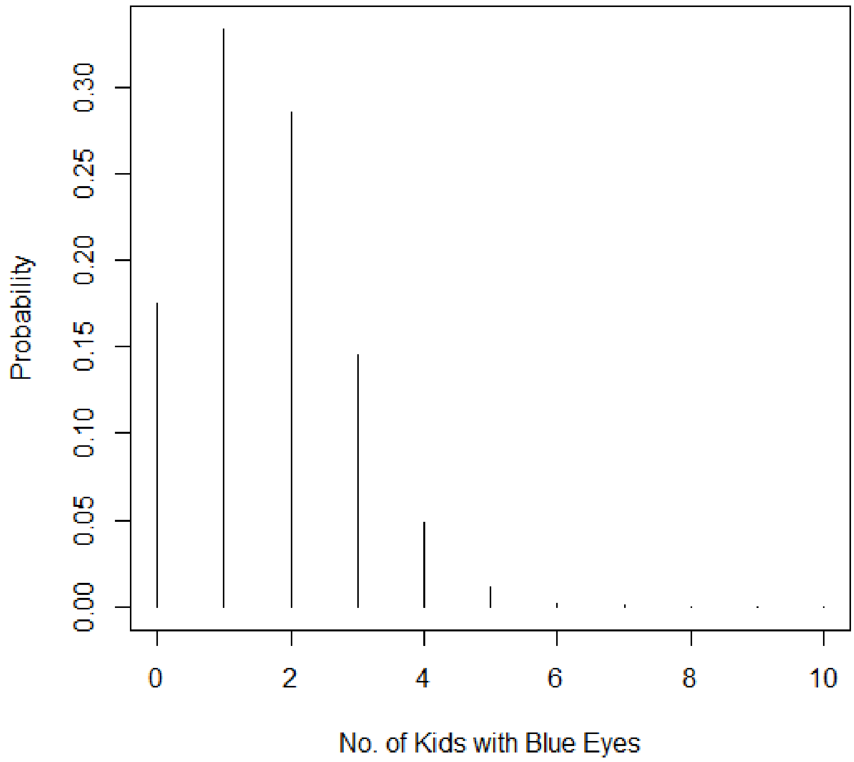



```
> # Stair steps Graph of F(x)
> plot(x,Fx,type='s',xlab='X',ylab='Probability')
```

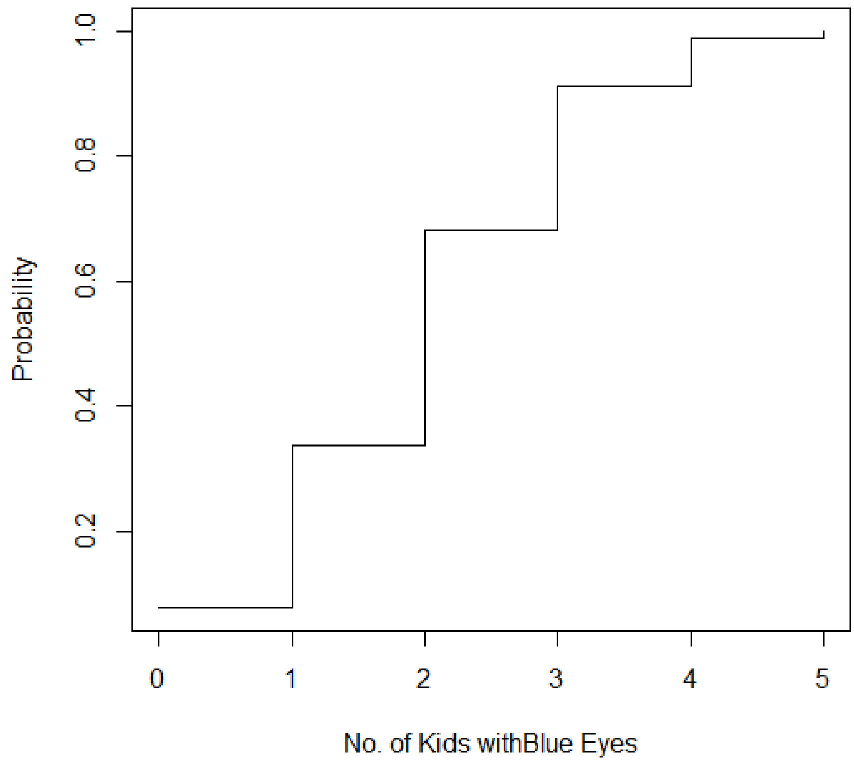


```
> # probability of child having blue eyes is 0.16
> # sample of 10 children taken
> x<-0:10
> n<-10
> p<-0.16
>
> # Probability Distribution
> fx<-dbinom(x,n,p)
>
> # Cumulative Probability Distribution
> Fx<-pbinom(x,n,p)
>
> data.frame("x"=x,"Probability"=round(fx,5),
+ "Cumulative Probability"=round(Fx,5))
    x Probability Cumulative.Probability
1   0     0.17490                0.17490
2   1     0.33315                0.50805
3   2     0.28555                0.79360
4   3     0.14504                0.93864
5   4     0.04835                0.98699
6   5     0.01105                0.99804
7   6     0.00175                0.99979
8   7     0.00019                0.99999
9   8     0.00001                1.00000
10  9     0.00000                1.00000
11 10     0.00000                1.00000
> # Probability that 2 children have blue eyes
> dbinom(2,n,p)
[1] 0.285553
>
> # Probability that at most 3 children have blue eyes
> pbinom(3,n,p)
[1] 0.9386423
>
> # Probability that less than 2 children have blue eyes
> dbinom(0,n,p)+dbinom(1,n,p)
[1] 0.5080464
>
> # Probability that at least 4 children have blue eyes
> 1-pbinom(3,n,p)
[1] 0.06135774
```

```
> # High Density Vertical Lines Graph of P(x)
> plot(x,fx,type='h',xlab='No. of Kids withBlue Eyes',ylab='Probability')
```




```
> # Stair steps Graph of F(x)
> plot(x,Fx,type='s',xlab='No. of Kids withBlue Eyes',ylab='Probability')
```



Practical No. 9

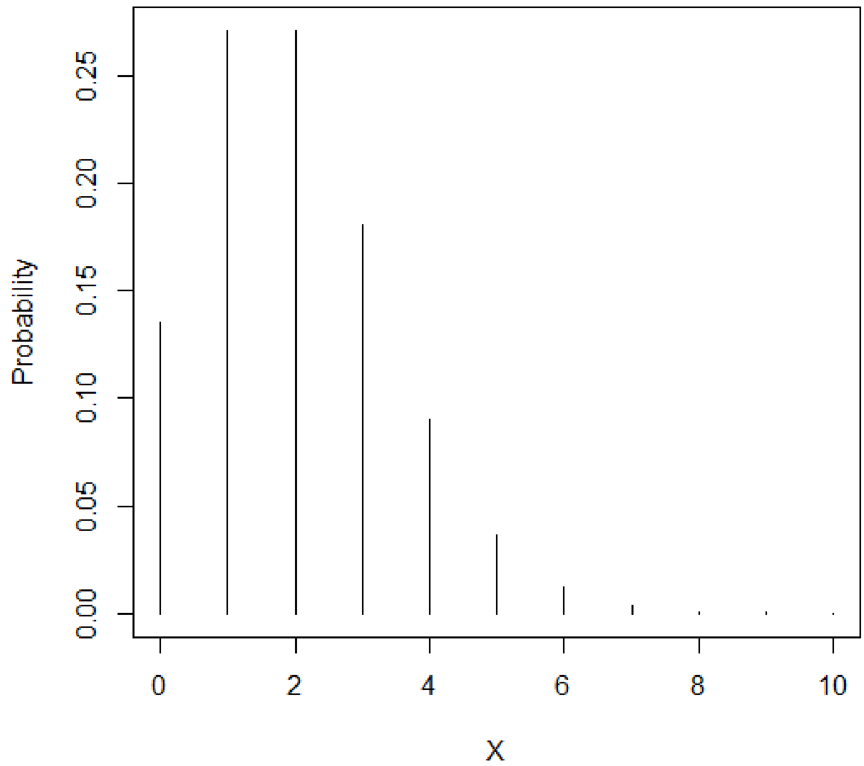
POISSON DISTRIBUTION

SOURCE CODE & OUTPUT:

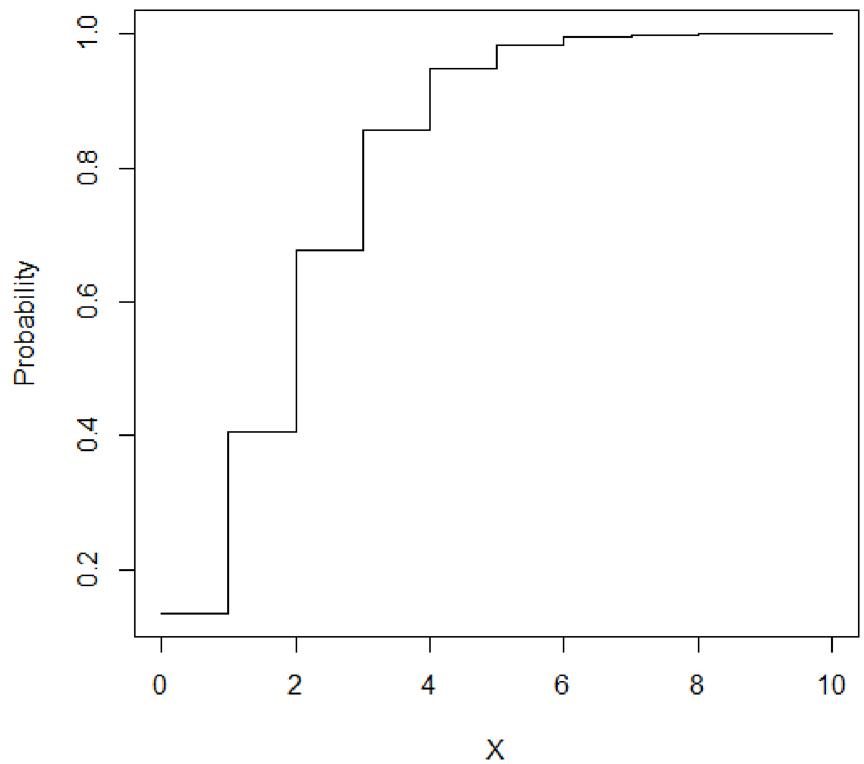
```
> # Poisson Distribution
> # X~P(lambda=2)
> x<-0:10
> lambda<-2
>
> # Probability Distribution
> fx<-dpois(x,lambda)
>
> # Cumulative Probability Distribution
> Fx<-ppois(x,lambda)
>
> data.frame("x"=x,"Probability"=round(fx,5),
+ "Cumulative Probability"=round(Fx,5))
    x Probability Cumulative.Probability
1   0     0.13534                0.13534
2   1     0.27067                0.40601
3   2     0.27067                0.67668
4   3     0.18045                0.85712
5   4     0.09022                0.94735
6   5     0.03609                0.98344
7   6     0.01203                0.99547
8   7     0.00344                0.99890
9   8     0.00086                0.99976
10  9     0.00019                0.99995
11 10     0.00004                0.99999

> # P[X=1]
> dpois(1,lambda)
[1] 0.2706706
>
> # P[X<=6]
> ppois(6,lambda)
[1] 0.9954662
>
> # P[X<3]
> dpois(0,lambda)+dpois(1,lambda)+dpois(2,lambda)
[1] 0.6766764
>
> # P[X>=4]
> 1-ppois(3,lambda)
[1] 0.1428765
```

```
> # High Density Vertical Lines Graph of P(x)
> plot(x,fx,type='h',xlab='X',ylab='Probability')
```




```
> # Stair steps Graph of F(x)
> plot(x,Fx,type='s',xlab='X',ylab='Probability')
```

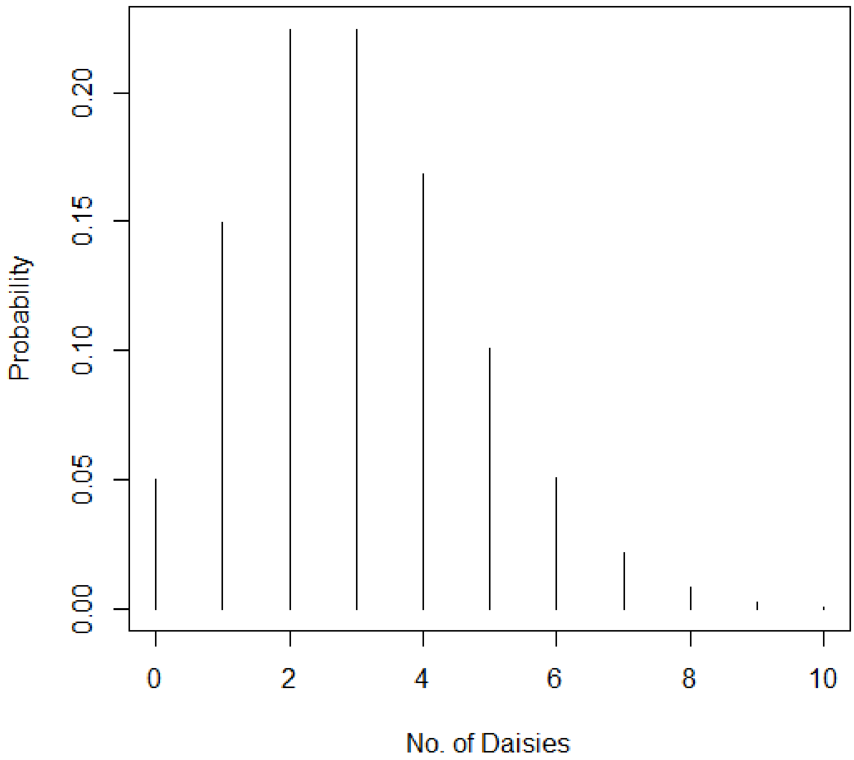


- A botanist is studying the distribution of daisies in a field. The field is divided into a number of equal sized squares. The mean number of daisies per square is assumed to be 3. The daisies are distributed randomly throughout the field.
  Find the probability that, in a randomly chosen square there will be
  i. more than 2 daisies, ii. either 5 or 6 daisies.

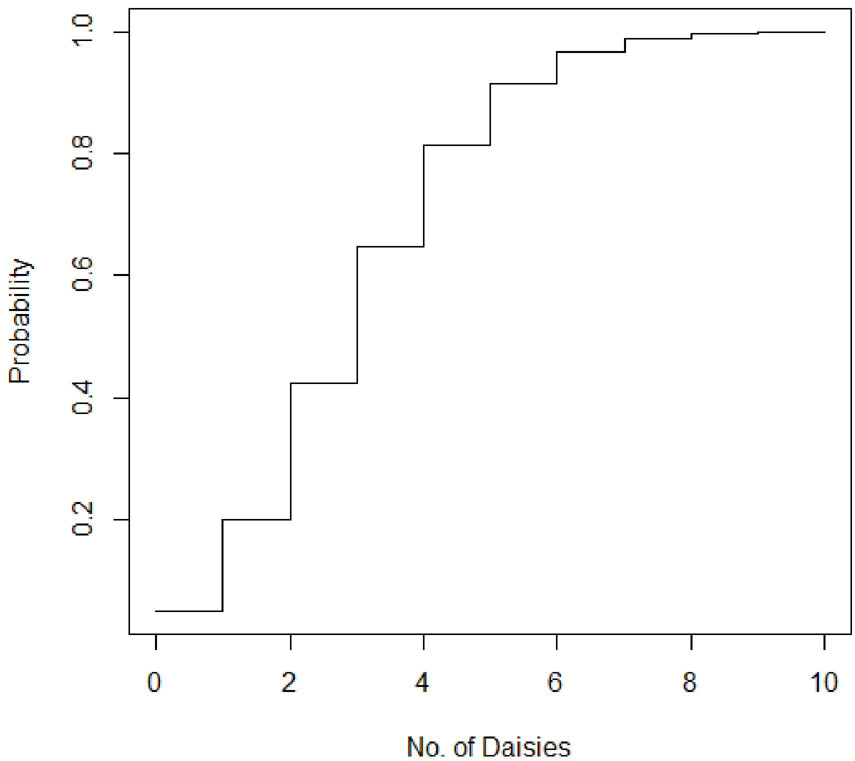
```
> # Poisson Distribution
> # X~P(lambda=3)
>
> lambda<-3
>
> # P(X>2)=1-P(X<=2)
> p1<-1-ppois(2,lambda)
> cat('Probability that the randomly selected
+ square has more than 2 daisies is ',p1,'.\n')
Probability that the randomly selected
square has more than 2 daisies is  0.5768099 .
>
> # P(X=5 OR 6)=P(X=5)+P(X=6)
> p2<-dpois(5,lambda)+dpois(6,lambda)
> cat('Probability that the randomly selected
+ square has 5 or 6 daisies is ',p2,'.\n')
Probability that the randomly selected
square has 5 or 6 daisies is  0.1512282 .

> # Probability Distribution
> x<-0:10
> fx<-dpois(x,lambda)
>
> # Cumulative Probability Distribution
> Fx<-ppois(x,lambda)
>
> data.frame("x"=x,"Probability"=round(fx,5),
+ "Cumulative Probability"=round(Fx,5))
    x Probability Cumulative.Probability
1   0     0.04979                0.04979
2   1     0.14936                0.19915
3   2     0.22404                0.42319
4   3     0.22404                0.64723
5   4     0.16803                0.81526
6   5     0.10082                0.91608
7   6     0.05041                0.96649
8   7     0.02160                0.98810
9   8     0.00810                0.99620
10  9     0.00270                0.99890
11 10     0.00081                0.99971
```

```
> # High Density Vertical Lines Graph of P(x)
> plot(x,fx,type='h',xlab='No. of Daisies',ylab='Probability')
```




```
> # Stair steps Graph of F(x)
> plot(x,Fx,type='s',xlab='No. of Daisies',ylab='Probability')
```

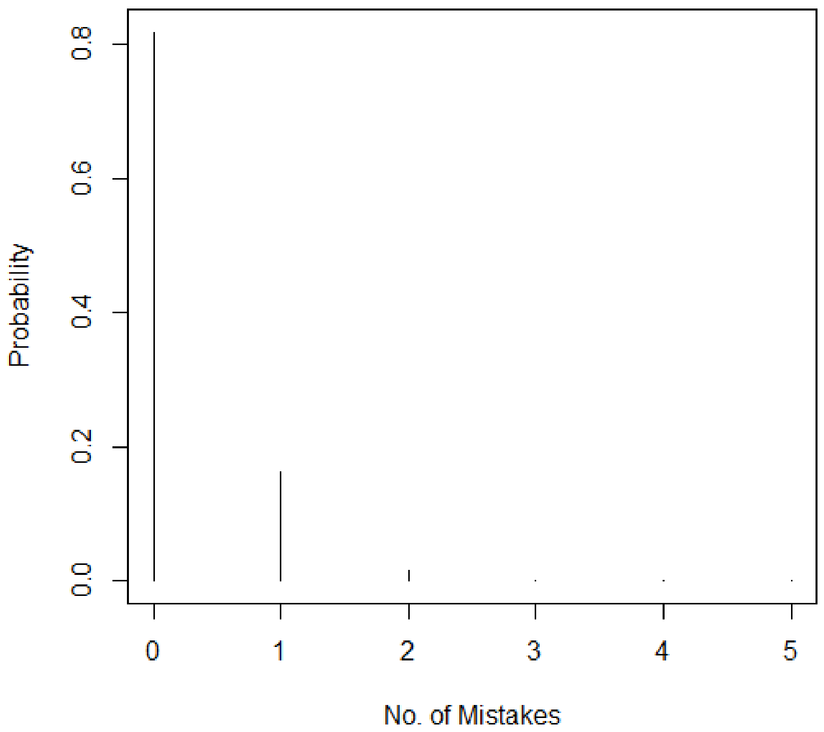


- Suppose we are using a new sequencing technique and the error rate is one mistake per 10,000 base pairs. Suppose we are sequencing 2000 base pair regions at a time. What is the probability of 0 mistakes using this technique? Of 1 mistake, 4 mistakes?

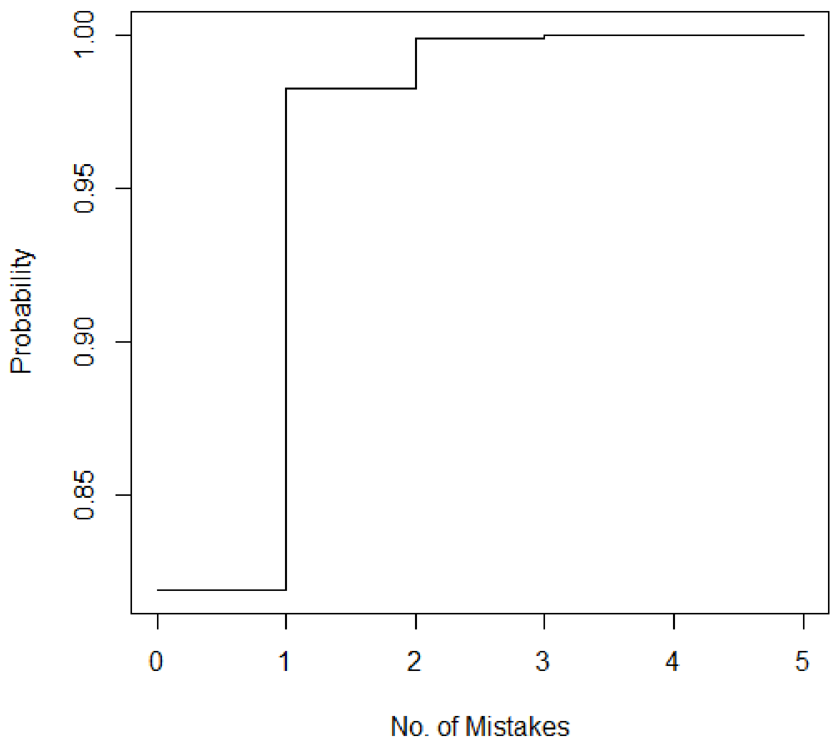
```
> # Poisson Distribution
> p<-1/10000
> n<-2000
> lambda<-n*p
> lambda
[1] 0.2
>
> # To find P[X=0], P[X=1] and P[X=4]
> p0<-dpois(0,lambda)
> cat('Probability of 0 mistakes is ',p0,'\n')
Probability of 0 mistakes is  0.8187308
> p1<-dpois(1,lambda)
> cat('Probability of 1 mistake is ',p1,'\n')
Probability of 1 mistake is  0.1637462
> p4<-dpois(4,lambda)
> cat('Probability of 4 mistakes is ',p4,'\n')
Probability of 4 mistakes is  5.458205e-05
```

```
> # Probability Distribution
> x<-0:5
> fx<-dpois(x,lambda)
>
> # Cumulative Probability Distribution
> Fx<-ppois(x,lambda)
>
> data.frame("x"=x,"Probability"=round(fx,5),
+ "Cumulative Probability"=round(Fx,5))
  x Probability Cumulative.Probability
1 0     0.81873                0.81873
2 1     0.16375                0.98248
3 2     0.01637                0.99885
4 3     0.00109                0.99994
5 4     0.00005                1.00000
6 5     0.00000                1.00000
```

```
> # High Density Vertical Lines Graph of P(x)
> plot(x,fx,type='h',xlab='No. of Mistakes',ylab='Probability')
```




```
> # Stair steps Graph of F(x)
> plot(x,Fx,type='s',xlab='No. of Mistakes',ylab='Probability')
```

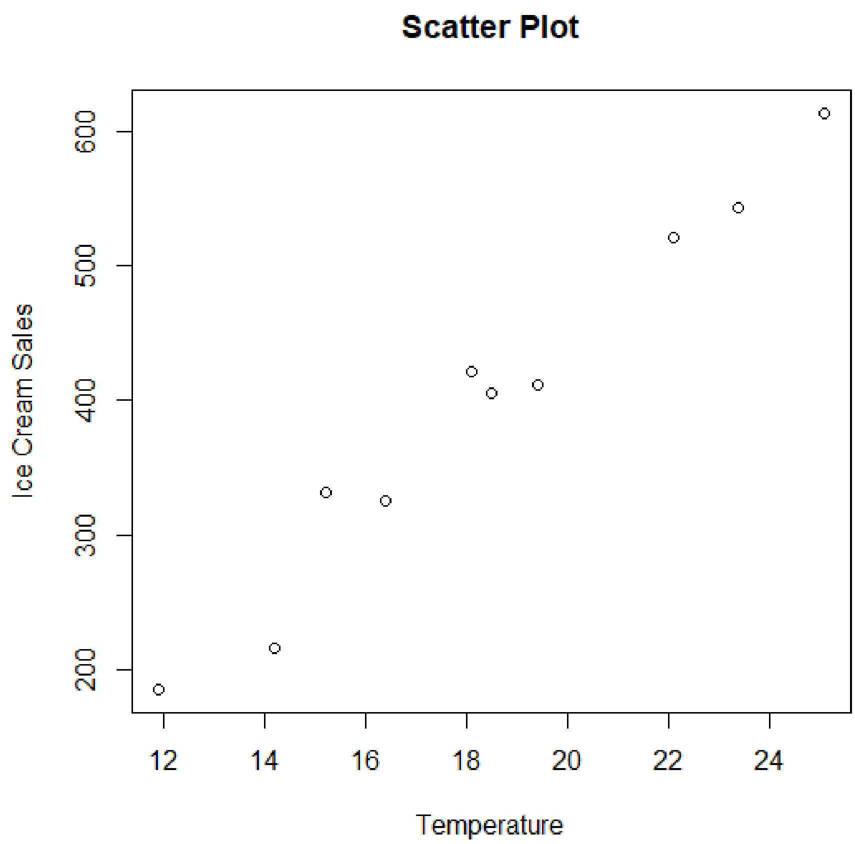


Practical No. 10

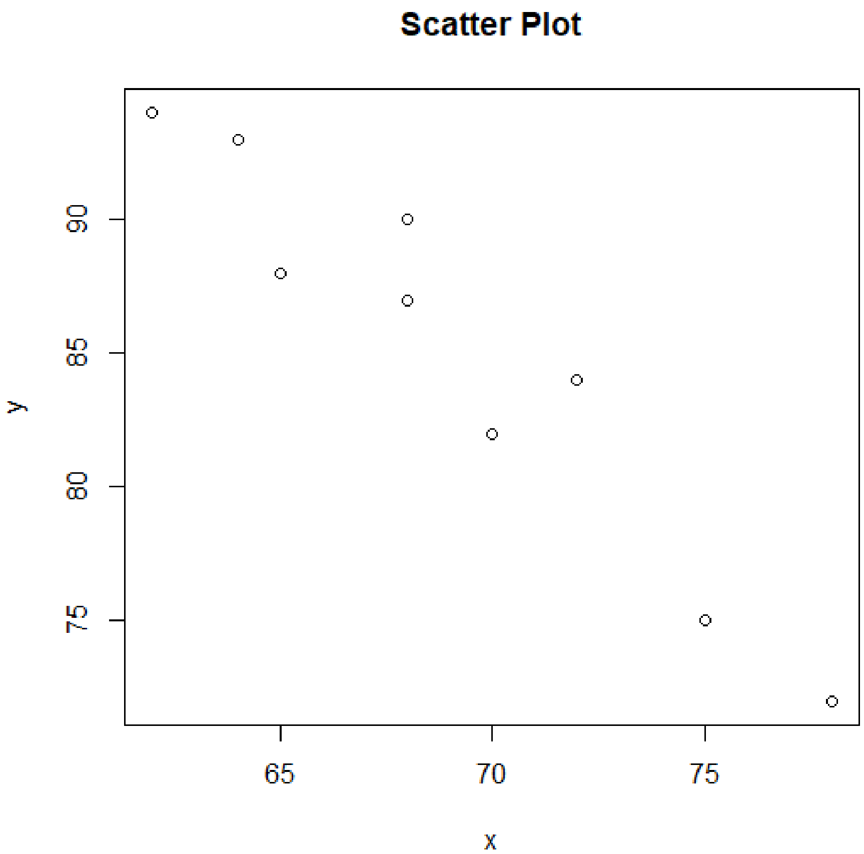
CORRELATION

SOURCE CODE & OUTPUT:

```
> # Positive Correlation Example
> temp<-c(14.2,16.4,11.9,15.2,18.5,22.1,19.4,25.1,23.4,18.1)
> sales<-c(215,325,185,332,406,522,412,614,544,421)
> r<-cor(temp,sales,method="pearson")
> cat('Coefficient of Correlation between temperature
+ and ice cream sales is ',r,'.\n')
Coefficient of Correlation between temperature
and ice cream sales is  0.9842961 .
>
> # Scatter diagram
> plot(temp,sales,xlab="Temperature",ylab="Ice Cream Sales",
+ main="Scatter Plot")
```



```
> # Negative Correlation Example
> x<-c(68,72,65,70,62,75,78,64,68)
> y<-c(90,84,88,82,94,75,72,93,87)
> r<-cor(x,y,method="pearson")
> cat('Coefficient of Correlation between x
+ and y is ',r,'.\n')
Coefficient of Correlation between x
and y is  -0.9591069 .
>
> # Scatter diagram
> plot(x,y,xlab="x",ylab="y",main="Scatter Plot")
```




```
> # Spearman's Rank Correlation
> R1<-c(1,2,3,4,5,6,7,8,9,10)
> R2<-c(4,8,1,3,2,5,10,7,6,9)
> R<-cor(R1,R2,method="spearman")
> R
[1] 0.5151515
> cat('Spearman rank correlation coefficient is ',R,'.\n')
Spearman rank correlation coefficient is  0.5151515 .
```

Practical No. 11

REGRESSION

SOURCE CODE & OUTPUT:

```
> # Regression
> temp<-c(14.2,16.4,11.9,15.2,18.5,22.1,19.4,25.1,23.4,18.1)
> sales<-c(215,325,185,332,406,522,412,614,544,421)
> reg<-lm(sales~temp)
> reg

Call:
lm(formula = sales ~ temp)

Coefficients:
(Intercept)         temp
    -200.60        32.46

> reg$coefficients[1]
(Intercept)
   -200.596
> reg$coefficients[2]
    temp
32.45773
> cat('Regression equation is
+ y=',reg$coefficients[1],'+(',reg$coefficients[2],')x.\n')
Regression equation is
y= -200.596 +( 32.45773 )x.

> fitted(reg)
       1        2        3        4        5        6        7        8
260.3038 331.7108 185.6510 292.7615 399.8720 516.7199 429.0840 614.0931
       9       10
558.9149 386.8889
```

```
> # Regression
> IT<-c(25,28,35,32,31,36,29,38,34,32)
> STATS<-c(43,46,49,41,36,32,31,30,33,39)
> reg<-lm(STATS~IT)
> reg

Call:
lm(formula = STATS ~ IT)

Coefficients:
(Intercept)           IT
    59.2571      -0.6643

> reg$coefficients[1]
(Intercept)
   59.25714
> reg$coefficients[2]
        IT
-0.6642857
> cat('Regression equation is
+ y=',reg$coefficients[1],'+(',reg$coefficients[2],')x.\n')
Regression equation is
y= 59.25714 +( -0.6642857 )x.

> fitted(reg)
       1        2        3        4        5        6        7        8
42.65000 40.65714 36.00714 38.00000 38.66429 35.34286 39.99286 34.01429
       9       10
36.67143 38.00000
```

```
> # Multiple Linear Regression
> x1<-c(3.33,3.96,4.58,5.33,3.13,3.67,4.58,3.00,4.50,3.50)
> x2<-c(3.92,2.58,3.92,3.08,3.54,4.17,4.17,3.67,4.67,4.25)
> y<-c(3.38,3.61,3.83,3.92,3.92,3.96,4.00,4.00,4.04,4.04)
> reg<-lm(y~x1+x2)
> reg

Call:
lm(formula = y ~ x1 + x2)

Coefficients:
(Intercept)           x1           x2
    3.07253      0.05609      0.15156

> cat('Regression equation is
+ y=',reg$coefficients[1],'+(',reg$coefficients[2],')x1','+(',reg$coefficients[3],')x2.\n')
Regression equation is
y= 3.072531 +( 0.05609011 )x1 +( 0.1515576 )x2.
> fitted(reg)
       1        2        3        4        5        6        7        8        9       10
3.853417 3.685667 3.923530 3.838289 3.784607 3.910377 3.961419 3.797018 4.032711 3.912966
```