Guru Gobind Singh Indraprastha University

University School of Automation & Robotics

Probability and Statistics for Engineers Lab (BS110P)

Practical File

Name	Sujal Singh
Enrollment Number	04119051723
Batch	IIOT–B1

Index

This page intentionally left blank.

Introduction to R Software.

Sujal Singh – 04119051723, IIOT–B1

1. Perform operations like assignment, addition and multiplication on variables.

Program:

```
a <- 42
b <- 24
a + b
a * c
```

Output:

```
[1] 66
[1] 1008
```

2. Make a vector and assign values to it and access 4^{th}, 3^{rd} and 6^{th} element and access all values from 3^{rd} element upto the length of vector.

Program:

```
myvector <- c(0, 10, 20, 30, 40 , 50, 60, 70, 80)

myvector[4]
myvector[3]
myvector[6]
myvector[0:length(myvector)]
```

Output:

```
[1] 30
[1] 20
[1] 50
[1]  0 10 20 30 40 50 60 70 80
```

3. Perform arithmetic operations on two vectors of different length.

Program:

```
a <- c(0, 1, 1, 2, 3, 5, 8, 13, 21)
b <- c(21, 34, 55)

a + b
a * b
```

Output:

```
[1] 21 35 56 23 37 60 29 47 76
[1]    0   34   55   42  102  275  168  442 1155
```

4. Create a sequence of numbers from 1 to 20 with a difference of 2 between each number using the `seq()` function.

Program:

```
seq(from = 1, to = 20, by = 2)
```

Output:

```
[1]  1  3  5  7  9 11 13 15 17 19
```

5. Create a matrix and access it's elements.

Program:

```
source <- c(
  1, 2, 3, 4,
  5, 6, 7, 8,
  9, 10, 11, 1
)
M <- matrix(source, nrow = 3, ncol = 4)
M
M[2,3]
```

Output:

```
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9    1
[1] 8
```

6. Create two matrices using vectors and perform element multiplication, matrix operations on matrices.

Program:

```
a <- matrix(seq(from = 1, to = 25), nrow = 5, ncol = 5)
b <- matrix(seq(from = 26, to = 50), nrow = 5, ncol = 5)
a * b
```

Output:

```
     [,1] [,2] [,3] [,4] [,5]
[1,]   26  186  396  656  966
[2,]   54  224  444  714 1034
[3,]   84  264  494  774 1104
[4,]  116  306  546  836 1176
[5,]  150  350  600  900 1250
```

7. Using the two matrices perform transpose operations on matrices.

Program:

```
a <- matrix(seq(from = 1, to = 20), nrow = 4, ncol = 5)
b <- matrix(seq(from = 21, to = 40), nrow = 4, ncol = 5)
t(a)
t(b)
```

Output:

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
[5,]   17   18   19   20
     [,1] [,2] [,3] [,4]
[1,]   21   22   23   24
[2,]   25   26   27   28
[3,]   29   30   31   32
[4,]   33   34   35   36
[5,]   37   38   39   40
```

8. Find the range of values, max value, min value, mean, median, variance, sum of a vector.

Program:

```
a <- seq(from = 1, to = 100, by = 3)
range(a)
max(a)
min(a)
mean(a)
median(a)
var(a)
sum(a)
```

Output:

```
[1]   1 100
[1] 100
[1] 1
[1] 50.5
[1] 50.5
[1] 892.5
[1] 1717
```

9. Create a data frame from vectors.

Program:

```
attended <- c(2, 4, 5, 6)
total <- rep(10, 4)

attendance <- data.frame(
  'Classes Attended' = attended, 'Total Classes' = total,
  row.names = c('BS106', 'A201', 'BS158', 'A104')
)

attendance
```

Output:

```
      Classes.Attended Total.Classes
BS106                2            10
A201                 4            10
BS158                5            10
A104                 6            10
```

10. Perform the operation of adding a new row to a data frame.

Program:

```
attended <- c(2, 4, 5, 6)
total <- rep(10, 4)

attendance <- data.frame(
  'Classes Attended' = attended, 'Total Classes' = total,
  row.names = c('BS106', 'A201', 'BS158', 'A104')
)

attendance <- rbind(attendance, 'ICT114'=c(2, 10))

attendance
```

Output:

```
BS106                2            10
A201                 4            10
BS158                5            10
A104                 6            10
ICT114               2            10
```

Graphs & Diagrams

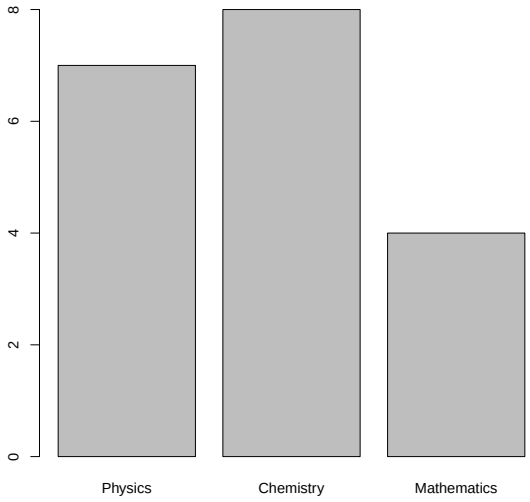
Sujal Singh – 04119051723, IIOT–B1

1. Perform operations like assignment, addition and multiplication on variables.

Program:

```
a <- 42
b <- 24
a + b
a * c
```

Output:



2. Using R make a grouped bar plot.

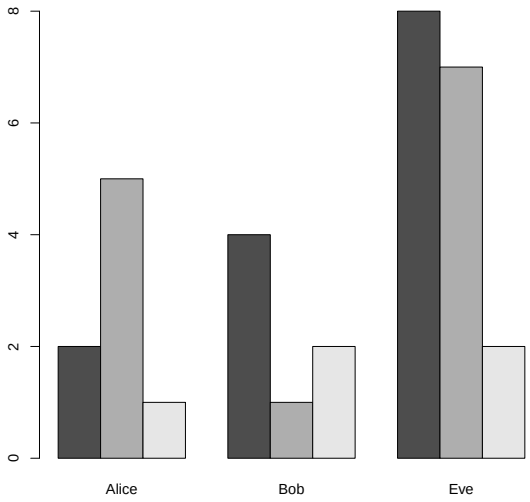
Program:

```
students <- c(
  "Alice", "Bob", "Eve"
)
subjects <- c(
  "Physics", "Chemistry", "Mathematics"
)
marks <- c(
  2, 5, 1,
  4, 1, 2,
  8, 7, 2
)

results <- matrix(marks, ncol = 3, nrow = 3)
rownames(results) <- subjects
colnames(results) <- students

barplot(results, beside = TRUE)
```

Output:

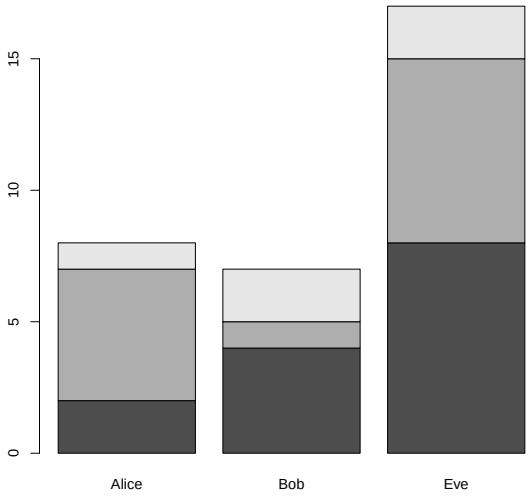


3. Using R make a grouped bar plot.

Program:

```
1  students <- c("Alice", "Bob", "Eve")
2  subjects <- c("Physics", "Chemistry", "Mathematics")
3  marks <- c(2, 5, 1, 4, 1, 2, 8, 7, 2)
4
5  results <- matrix(marks, ncol = 3, nrow = 3)
6  rownames(results) <- subjects
7  colnames(results) <- students
8
9  barplot(results)
```

Output:



4. Make a horizontal bar plot.

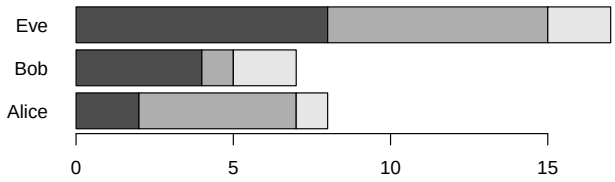
Program:

```r
students <- c("Alice", "Bob", "Eve")
subjects <- c("Physics", "Chemistry", "Mathematics")
marks <- c(2, 5, 1, 4, 1, 2, 8, 7, 2)

results <- matrix(marks, ncol = 3, nrow = 3)
rownames(results) <- subjects
colnames(results) <- students

barplot(results, horiz = TRUE, las = 1)
```

Output:

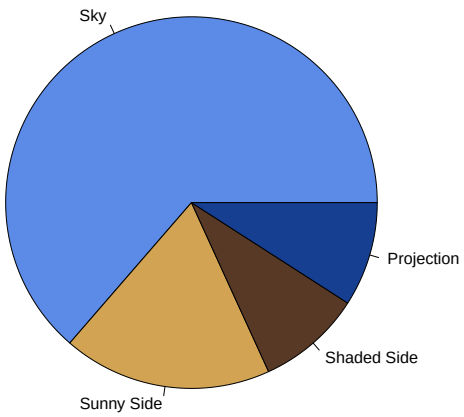


5. Make a pie chart.

Program:

```r
pie(
  c(70, 20, 10, 10),
  labels = c("Sky", "Sunny Side", "Shaded Side", "Projection"),
  col = c("#5B8AE7", "#D2A353", "#583925", "#163F92")
)
```

Output:

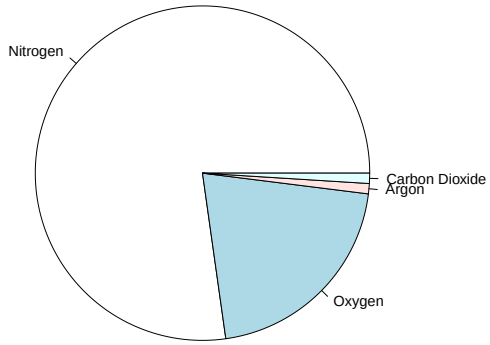


6. Make a pie chart showing atmospheric composition.

Program:

```
pie(
  c(78, 21, 1, 1),
  labels = c("Nitrogen", "Oxygen", "Argon", "Carbon Dioxide"),
)
```

Output:

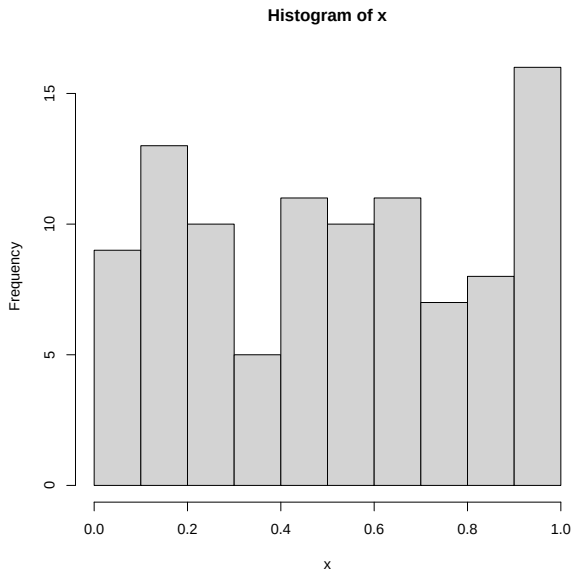


7. Make a histogram showing frequency of random numbers between 0 and 100.

Program:

```
x <- runif(100)
hist(x)
```

Output:

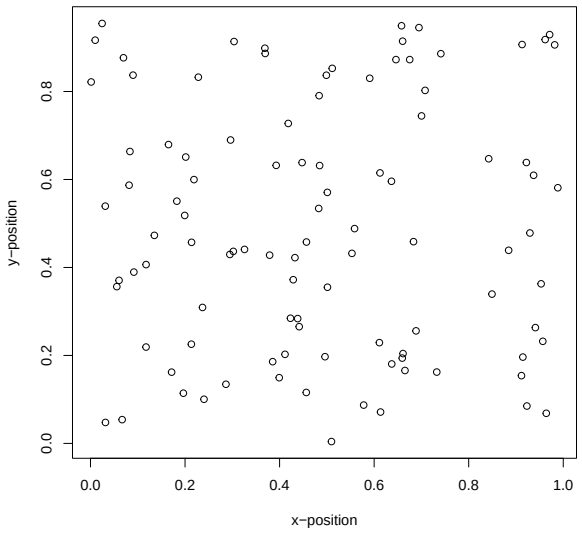


8. Generate a plot using random points.

Program:

```
plot(
  runif(100),
  runif(100),
  xlab = "x-position",
  ylab = "y-position"
)
```

Output:

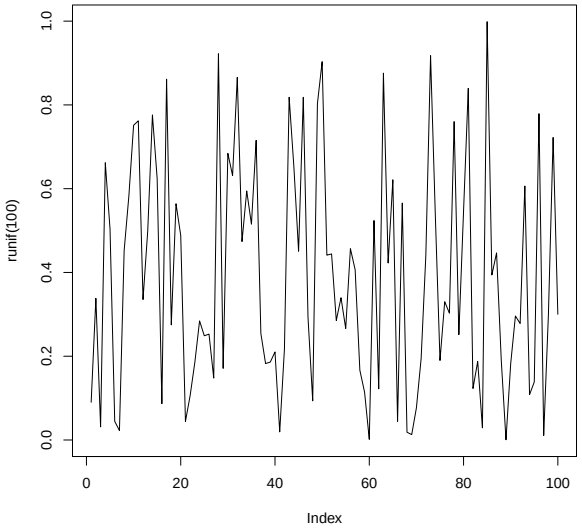


9. Make a line type plot.

Program:

```
plot(runif(100), type = "l")
```

Output:

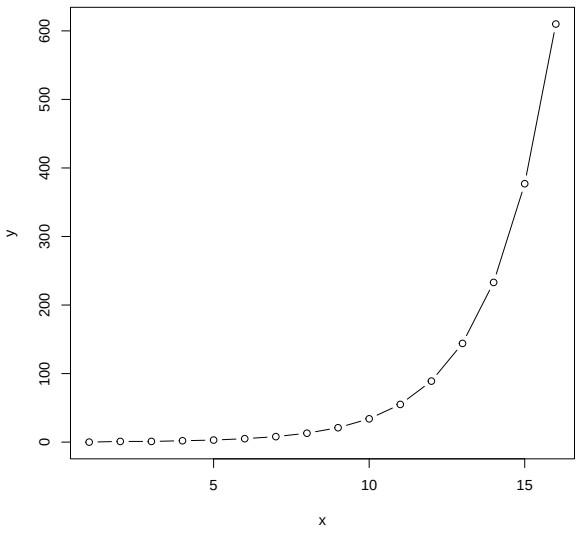


10. Plot the Fibonacci sequence.

Program:

```
1 data <- c(0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610)
2 plot(data, data, type = "b", xlab = "x", ylab = "y")
```

Output:



Measures of Central Tendency

Sujal Singh – 04119051723, IIOT–B1

1. Caculate the mean of a vector.

Program:

```
a <- c(28, 47, 29, 74, 47, 67, 83, 58, 93, 10)
mean(a)
```

Output:

```
[1] 53.6
```

2. Caculate the mean and median of a vector x.

Program:

```
x <- c(28, 47, 29, 74, 47, 67, 83, 58, 93, 10)
mean(x)
median(x)
```

Output:

```
[1] 53.6
[1] 52.5
```

3. Caculate the geometric mean of a vector.

Program:

```
x <- c(4, 12, 36, 108, 324, 972, 2916)
sqrt(x[1] * x[length(x)])
# OR
exp(mean(log(x)))
```

Output:

```
[1] 108
[1] 108
```

4. Caculate the harmonic mean of a vector.

Program:

```
library("psych")
x <- c(4, 12, 36, 108, 324, 972, 2916)
harmonic.mean(x)
```

Output:

```
[1] 18.67521
```

5. Caculate the arithmetic, geometric and harmonic mean of a vector.

Program:

```
library("psych")
x <- c(4, 12, 36, 108, 324, 972, 2916)
mean(x)
geometric.mean(x)
harmonic.mean(x)
harmonic.mean(x)
```

Output:

```
[1] 624.5714
[1] 108
[1] 18.67521
```

6. Caculate the first, second and third quartile of a vector.

Program:

```
x <- c(4, 12, 36, 108, 324, 972, 2916, 8748)
quantile(x, 1 / 4)
quantile(x, 2 / 4)
quantile(x, 3 / 4)
```

Output:

```
25%
 30
50%
216
 75%
1458
```

7. Caculate the third, fifth and sixth decile of a vector.

Program:

```
x <- c(4, 12, 36, 108, 324, 972, 2916, 8748)
quantile(x, 3 / 10)
quantile(x, 5 / 10)
quantile(x, 6 / 10)
```

Output:

```
 30%
43.2
50%
216
  60%
453.6
```

8. Caculate the mean and median of a vector and ignore NA values.

Program:

```
x <- c(4, 12, NA, 108, 324, NA, 2916, 8748)
mean(x, na.rm = TRUE)
median(x, na.rm = TRUE)
```

Output:

```
[1] 2018.667
[1] 216
```

9. Caculate the arithmetic, geometric and harmonic mean with NA removal.

Program:

```
library("psych")
x <- c(4, 12, 36, 108, 324, 972, 2916)
mean(x, na.rm = TRUE)
geometric.mean(x, na.rm = TRUE)
harmonic.mean(x, na.rm = TRUE)
```

Output:

```
[1] 624.5714
[1] 108
[1] 18.67521
```

10. Find the first and third quartile of a vector with NA removal.

Program:

```
x <- c(4, 12, NA, 108, 324, NA, 2916, 8748)
quantile(x, 1 / 4, na.rm = TRUE)
quantile(x, 3 / 4, na.rm = TRUE)
```

Output:

```
25%
 36
 75%
2268
```

Measures of Dispersion

Sujal Singh – 04119051723, IIOT–B1

1. Find the range and coefficent of range.

Program:

```r
x <- floor(runif(10, min = 0, max = 100))
x
range(x)
# Coefficent of range
(max(x) - min(x)) / (max(x) + min(x))
```

Output:

```
 [1] 38 31 39 44 68 78 47 12 20 27
[1] 12 78
[1] 0.7333333
```

2. Find the quartile deviation and its coefficients.

Program:

```r
x <- floor(runif(20, min = 0, max = 100))
x
QD <- (quantile(x, 3 / 4) - quantile(x, 1 / 4)) / 2
coeffQD <- (quantile(x, 3 / 4) - quantile(x, 1 / 4)) / (quantile(x, 3 / 4) +
↪  quantile(x, 1 / 4))
QD
coeffQD
```

Output:

```
 [1] 94 26 60 42 34 66 66 12 38 56 67 62 66 40 21 72 97 70 25 40
   75%
14.625
     75%
0.283293
```

3. Find the variance and standard deviation.

Program:

```r
x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
var(x)
sd(x)
```

Output:

```
[1] 9.166667
[1] 3.02765
```

4. Find the standard deviation and coefficient of variance.

Program:

```r
x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
mean(x)
sd(x)
# Coefficient of variance
(sd(x) / mean(x)) * 100
```

Output:

```
[1] 5.5
[1] 3.02765
[1] 55.04819
```

5. Find the mean deviation from median, variance, standard deviation, population standard deviation and their coefficients.

Program:

```r
x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
mad(x)
mad(x) / median(x)
var(x)
psd <- (sd(x) / mean(x)) * 100
psd
(psd / mean(x)) * 100
```

Output:

```
[1] 3.7065
[1] 0.6739091
[1] 9.166667
[1] 55.04819
[1] 1000.876
```

6. Find the range, coefficient of range, quartile deviation, coefficient of quartile deviation for ungrouped data.

Program:

```r
grp <- seq(0, 5)
f <- c(5, 35, 2, 24, 25, 35)
x <- rep(grp, f)
crange <- (max(x) - min(x)) / (max(x) + min(x))
QD <- (quantile(x, 3 / 4) - quantile(x, 1 / 4)) / 2
CQD <- (quantile(x, 3 / 4) - quantile(x, 1 / 4)) / (quantile(x, 3 / 4) +
 ↪  quantile(x, 1 / 4))
x
diff(range(x))
crange
QD
CQD
```

Output:

```
  [1] 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
  ↪  1 1
 [38] 1 1 1 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4
  ↪  4 4
 [75] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
  ↪  5 5
[112] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[1] 5
[1] 1
75%
  2
     75%
0.6666667
```

7. Find the mean deviation from median and coefficient of mean deviation from median for ungrouped data.

Program:

```r
grp <- seq(0, 5)
f <- c(5, 35, 2, 24, 25, 35)
x <- rep(grp, f)
mad(x)
mad(x) / median(x)
```

Output:

```
[1] 2.9652
[1] 0.9884
```

8. Find the variance, coefficient of variance, sample standard deviation, population standard deviation and standard error

Program:

```r
grp <- seq(0, 5)
f <- c(5, 35, 2, 24, 25, 35)
x <- rep(grp, f)
psd <- (sd(x) * length(x) - 1) / sqrt(length(x))
cv <- (psd / mean(x)) * 100
var(x)
sd(x)
cv
```

Output:

```
[1] 2.811937
[1] 1.676883
[1] 611.5204
```

9. Find the skewness.

Program:

```
library(moments)

x <- c(25, 29, 30, 17, 19, 30, 18, 28, 31, 33, 26, 28)
psd <- (sd(x) * sqrt(length(x) - 1)) / sqrt(length(x))
skp <- (3 * (mean(x) - median(x))) / psd
a <- quantile(x, 3 / 4)
b <- quantile(x, 1 / 4)
c <- 2 * quantile(x, 1 / 2)
skb <- (a + b - c) / (a - b)

skewness(x)
```

Output:

```
[1] -0.6760079
```

10. Find the kurtosis.

Program:

```
library(moments)

x <- c(25, 29, 30, 17, 19, 30, 18, 28, 31, 33, 26, 28)
kurtosis(x)
kurtosis(x) - 3
```

Output:

```
[1] 2.068371
[1] -0.9316292
```

Probability and Probability Distributions

Sujal Singh – 04119051723, IIOT–B1

1. A box contains 4 red, 3 white and 2 blue balls. Three balls are drawn at random. Find out the number of ways of selecting the balls of different colours.

Program:

```
#    RED           WHITE          BLUE
choose(4, 1) * choose(3, 1) * choose(2, 1)
```

Output:

```
[1] 24
```

2. Find the probability of picking two Ace cards from a well shuffled pack of 52 playing cards.

Program:

```
choose(4, 2) / choose(52, 2)
```

Output:

```
[1] 0.004524887
```

3. If $X \sim Bino(10, 0.6)$. Find:

(a) $P(X = 0)$

(b) $P(X = 2)$

(c) $P(X \leq 3)$

Program:

```
al <- dbinom(0, 10, 0.6)
al
bl <- dbinom(2, 10, 0.6)
bl
cl <- dbinom(3, 10, 0.6)
cl
dl <- dbinom(5, 10, 0.6)
dl
```

Output:

```
[1] 0.0001048576
[1] 0.01061683
[1] 0.04246733
[1] 0.2006581
```

4. If $X \sim P(3.2)$. Find:

(a) $P(X = 0)$

(b) $P(X = 3)$

(c) $P(X = 5)$

(d) $P(X \leq 1)$

(e) $P(X > 3)$

(f) $P(X \geq 5)$

Program:

```
al <- dpois(0, 3.2)
al
bl <- dpois(3, 3.2)
bl
cl <- dpois(5, 3.2)
cl
dl <- ppois(10, 3.2)
dl
el <- 1 - ppois(3, 3.2)
el
fl <- 1 - ppois(5, 3.2)
fl
```

Output:

```
[1] 0.0407622
[1] 0.222616
[1] 0.1139794
[1] 0.9995028
[1] 0.3974803
[1] 0.1054081
```

5. Fit the Poisson distribution to the following data with respect to the number of red blood corpuscles (x) per cell - x:

0	1	2	3	4	5
142	156	69	27	5	1

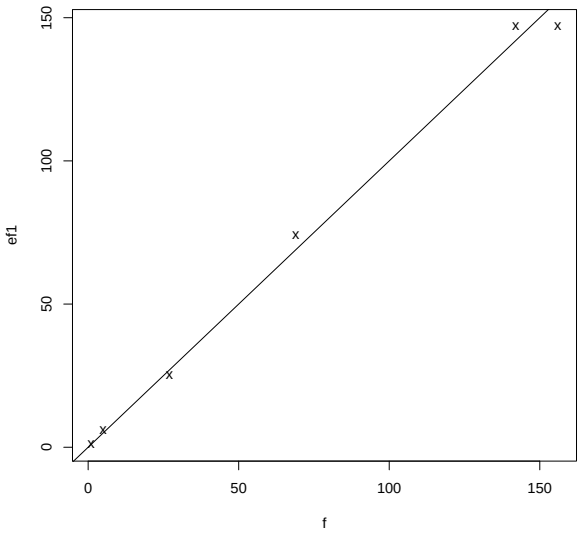
Program:

```
x <- 0:5; f <- c(142, 156, 69, 27, 5, 1)
m <- sum(x * f) / sum(f)
px <- dpois(x, m); px <- round(px, 4)
ef <- sum(f) * px
ef1 <- round(ef, 0)
d <- data.frame(x, f, "expected frequency" = ef)
d
plot(f, ef1, pch = "x"); abline(0, 1)
```

Output:

```
  x   f expected.frequency
1 0 142             147.16
2 1 156             147.16
3 2  69              73.56
4 3  27              24.52
5 4   5               6.12
6 5   1               1.24
```

Figure:



6. Plot the probability mass function (PMF) and distribution function for the following random variables $X \sim P(2.6)$.

Program:

```
m <- 2.6
x <- 0:10
p <- dpois(x, m)
d <- data.frame(x, p)
d
plot(x, p, "h")

cp <- ppois(x, m)
cp1 <- round(cp, 4)
d1 <- data.frame(x, cp1)
plot(x, cp1, "s")
```

Output:

```
    x            p
1   0 0.0742735782
2   1 0.1931113034
3   2 0.2510446944
4   3 0.2175720684
5   4 0.1414218445
6   5 0.0735393591
7   6 0.0318670556
8   7 0.0118363349
9   8 0.0038468089
10  9 0.0011113003
11 10 0.0002889381
```

Figure:

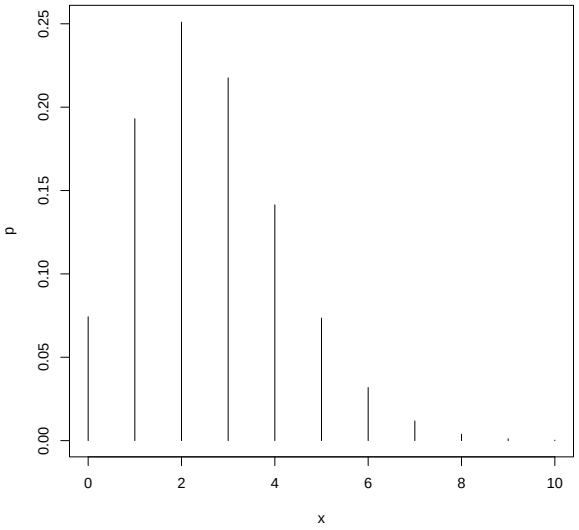
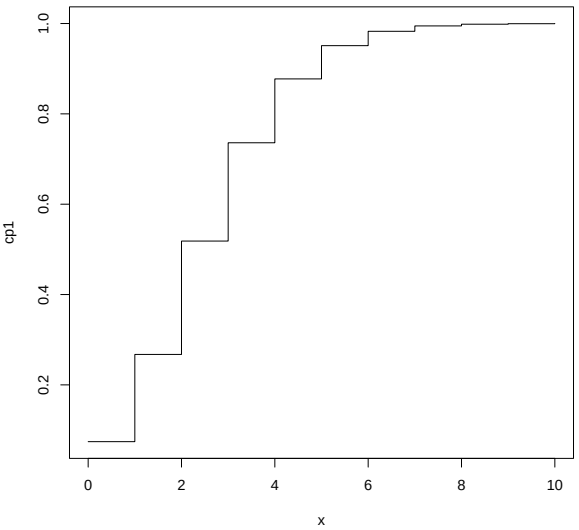


Figure:



7. Plot the probability mass function (PMF) and distribution function for the following random variables $X \sim Bino(8, 0.65)$.

Program:

```
n <- 8
p <- 0.65
x <- 0:n
bp <- dbinom(x, n, p)
d <- data.frame(x, "probabilities" = bp)
d
plot(x, bp, "h")
```

Output:

```
  x probabilities
1 0  0.0002251875
2 1  0.0033456434
3 2  0.0217466823
4 3  0.0807733916
5 4  0.1875096590
6 5  0.2785857791
7 6  0.2586867948
8 7  0.1372623809
9 8  0.0318644813
```

Figure:

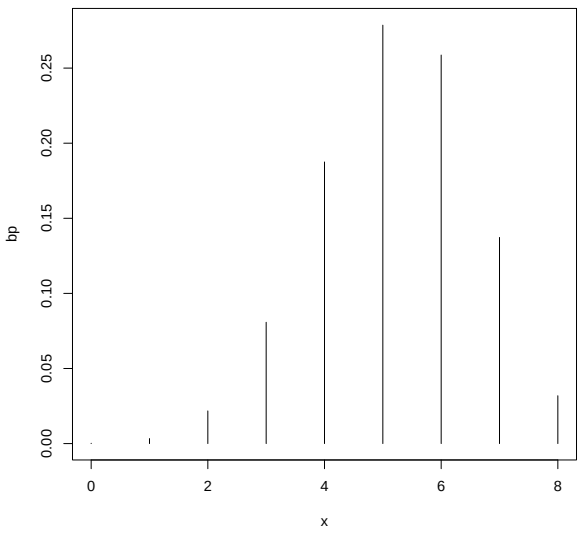


Figure:

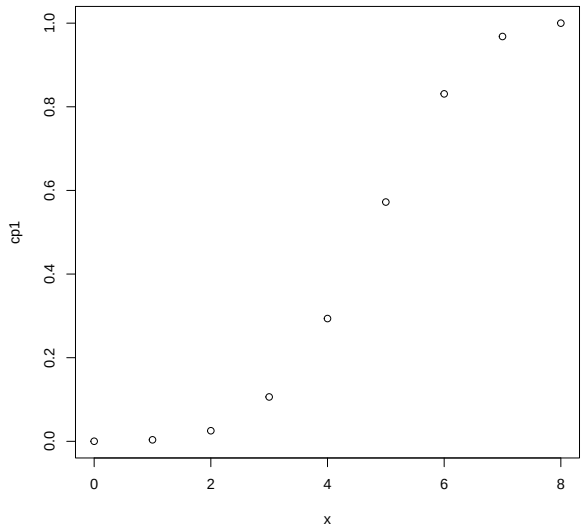
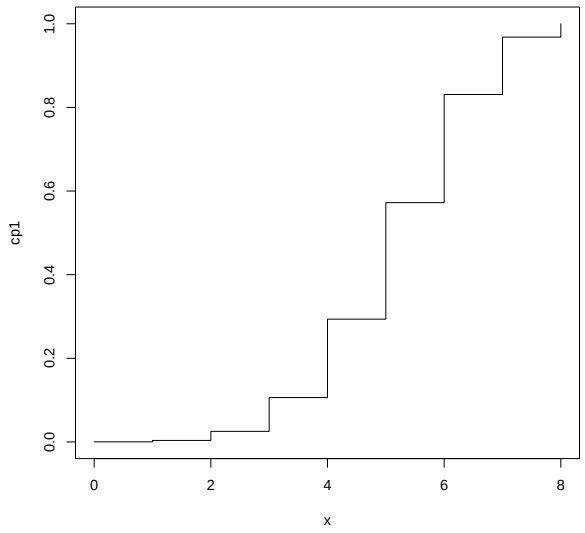


Figure:



8. Plot the probability mass function (PMF) and distribution function for the following random variables $X \sim \text{HyperGeo}(N = 50,\ M = 10,\ n = 7)$.

Program:

```
1  N <- 50; M <- 10; n <- 7; x <- 0:n
2  hp <- dhyper(x, M, N - M, n)
3  d <- data.frame(x, hp); d
4  plot(x, hp, "h")
5
6  cp <- phyper(x, M, N - M, n)
7  cp1 <- round(cp, 4)
8  di <- data.frame(x, cp1); di
9  plot(x, cp1, "s")
```

Output:

```
  x           hp
1 0 1.866514e-01
2 1 3.842822e-01
3 2 2.964463e-01
4 3 1.097949e-01
5 4 2.077201e-02
6 5 1.967875e-03
7 6 8.409722e-05
8 7 1.201389e-06
  x    cp1
1 0 0.1867
2 1 0.5709
3 2 0.8674
4 3 0.9772
5 4 0.9979
6 5 0.9999
7 6 1.0000
8 7 1.000
```

Figure:

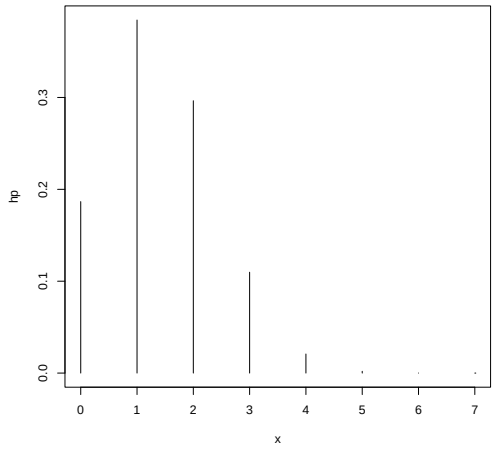
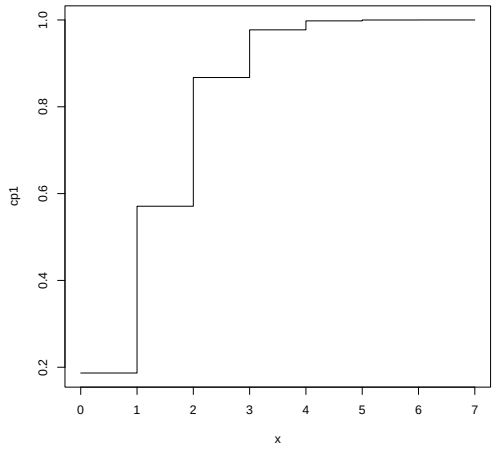


Figure:



9. Fit a normal distribution to the following data of height (in cms) of 200 Indian adult males:

Height (cm)	144–150	150–156	156–162	162–168	168–174	174–180	180–186
No. of Adults	3	12	23	52	61	39	10

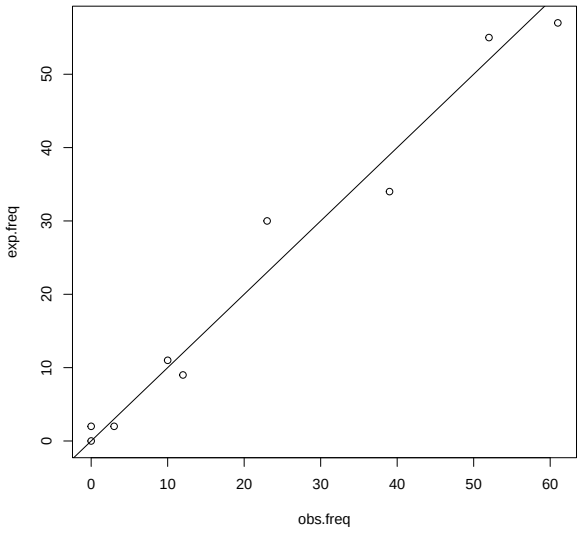
Program:

```
li <- seq(144, 180, 6); ul <- seq(150, 186, 6)
f <- c(3, 12, 23, 52, 61, 39, 10)
x <- (li + ul) / 2; n <- sum(f); k <- length(f)
m <- sum(f * x) / n; v <- sum(f * (x - m)^2) / n; sd <- sqrt(v)
l1 <- c(-9999, li, 186)
cp <- pnorm(l1, m, sd)
p <- diff(cp)
p <- c(p, 1 - cp[k + 2])
ul <- c(144, ul, 9999); f <- c(0, f, 0)
ef <- round(n * p, 0)
d <- data.frame(
  "Lower Limit" = l1, "Upper Limit" = ul, "Obs. freq" = f,
  "prob" = p, "cumprob" = cp, "expfreq" = ef
)
d
plot(f, ef, xlab = "obs.freq", ylab = "exp.freq", "p"); abline(0, 1)
```

Output:

```
 Lower.Limit Upper.Limit Obs..freq         prob      cumprob expfreq
1       -9999         144         0 0.0009277682 0.0000000000       0
2         144         150         3 0.0085408285 0.0009277682       2
3         150         156        12 0.0474590553 0.0094685967       9
4         156         162        23 0.1504843558 0.0569276520      30
5         162         168        52 0.2727415211 0.2074120077      55
6         168         174        61 0.2828190953 0.4801535289      57
7         174         180        39 0.1677990586 0.7629726242      34
8         180         186        10 0.0569156032 0.9307716828      11
9         186        9999         0 0.0123127140 0.9876872860       2
```

Figure:



10. Let $X \sim N(5, 40)$. Find $P(X \leq 60), P(X \geq 100), P(10 \leq x \leq 20)$ and $P(X \leq k) = 0.293$.

Program:

```
mu <- 50; sd <- sqrt(40)
pl <- pnorm(60, mu, sd)
pl
p2 <- 1 - pnorm(100, mu, sd)
p2
p3 <- pnorm(20, mu, sd)
p3
p4 <- qnorm(0.293, mu, sd)
p4
```

Output:

```
[1] 0.9430769
[1] 1.332268e-15
[1] 1.050718e-06
[1] 46.55538
```