



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	ILOT-B1

Index

No.	Chapter Name	Remarks
1	Introduction to R Software.	
2	Graphs & Diagrams.	
3	Measures of Central Tendency.	
4	Measures of Dispersion.	
5	Probability and Probability Distributions.	
6	Sampling Distribution & Central Limit Theorem.	
7	Statistical Tests using R.	
8	Analysis of Variance (ANOVA) using R.	

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:	
1	a <- 42
2	b <- 24
3	a + b
4	a * c

Output:	
1	[1] 66
2	[1] 1008

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

Program:	
1	myvector <- c(0, 10, 20, 30, 40 , 50, 60, 70, 80)
2	
3	myvector[4]
4	myvector[3]
5	myvector[6]
6	myvector[0:length(myvector)]

Output:	
1	[1] 30
2	[1] 20
3	[1] 50
4	[1] 0 10 20 30 40 50 60 70 80

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

Program:	
1	a <- c(0, 1, 1, 2, 3, 5, 8, 13, 21)
2	b <- c(21, 34, 55)
3	
4	a + b
5	a * b

Output:	
1	[1] 21 35 56 23 37 60 29 47 76
2	[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:	
1	<code>seq(from = 1, to = 20, by = 2)</code>
Output:	
1	<code>[1] 1 3 5 7 9 11 13 15 17 19</code>

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

Program:	
1	<code>source <- c(</code>
2	<code> 1, 2, 3, 4,</code>
3	<code> 5, 6, 7, 8,</code>
4	<code> 9, 10, 11, 1</code>
5	<code>)</code>
6	<code>M <- matrix(source, nrow = 3, ncol = 4)</code>
7	<code>M</code>
8	<code>M[2,3]</code>
Output:	
1	<code> [,1] [,2] [,3] [,4]</code>
2	<code>[1,] 1 4 7 10</code>
3	<code>[2,] 2 5 8 11</code>
4	<code>[3,] 3 6 9 1</code>
5	<code>[1] 8</code>

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

Program:	
1	<code>a <- matrix(seq(from = 1, to = 25), nrow = 5, ncol = 5)</code>
2	<code>b <- matrix(seq(from = 26, to = 50), nrow = 5, ncol = 5)</code>
3	<code>a * b</code>
Output:	
1	<code> [,1] [,2] [,3] [,4] [,5]</code>
2	<code>[1,] 26 186 396 656 966</code>
3	<code>[2,] 54 224 444 714 1034</code>
4	<code>[3,] 84 264 494 774 1104</code>
5	<code>[4,] 116 306 546 836 1176</code>
6	<code>[5,] 150 350 600 900 1250</code>

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

Program:	
1	a <- matrix(seq(from = 1, to = 20), nrow = 4, ncol = 5)
2	b <- matrix(seq(from = 21, to = 40), nrow = 4, ncol = 5)
3	t(a)
4	t(b)
Output:	
1	[,1] [,2] [,3] [,4]
2	[1,] 1 2 3 4
3	[2,] 5 6 7 8
4	[3,] 9 10 11 12
5	[4,] 13 14 15 16
6	[5,] 17 18 19 20
7	[,1] [,2] [,3] [,4]
8	[1,] 21 22 23 24
9	[2,] 25 26 27 28
10	[3,] 29 30 31 32
11	[4,] 33 34 35 36
12	[5,] 37 38 39 40

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

Program:	
1	a <- seq(from = 1, to = 100, by = 3)
2	range(a)
3	max(a)
4	min(a)
5	mean(a)
6	median(a)
7	var(a)
8	sum(a)
Output:	
1	[1] 1 100
2	[1] 100
3	[1] 1
4	[1] 50.5
5	[1] 50.5
6	[1] 892.5
7	[1] 1717

9. Create a data frame from vectors.

Program:	
1	<code>attended <- c(2, 4, 5, 6)</code>
2	<code>total <- rep(10, 4)</code>
3	
4	<code>attendance <- data.frame(</code>
5	<code> 'Classes Attended' = attended, 'Total Classes' = total,</code>
6	<code> row.names = c('BS106', 'A201', 'BS158', 'A104')</code>
7	<code>)</code>
8	
9	<code>attendance</code>
Output:	
1	<pre>Classes.Attended Total.Classes</pre>
2	<pre>BS106 2 10</pre>
3	<pre>A201 4 10</pre>
4	<pre>BS158 5 10</pre>
5	<pre>A104 6 10</pre>

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

Program:	
1	<code>attended <- c(2, 4, 5, 6)</code>
2	<code>total <- rep(10, 4)</code>
3	
4	<code>attendance <- data.frame(</code>
5	<code> 'Classes Attended' = attended, 'Total Classes' = total,</code>
6	<code> row.names = c('BS106', 'A201', 'BS158', 'A104')</code>
7	<code>)</code>
8	
9	<code>attendance <- rbind(attendance, 'ICT114'=c(2, 10))</code>
10	
11	<code>attendance</code>
Output:	
1	<pre>BS106 2 10</pre>
2	<pre>A201 4 10</pre>
3	<pre>BS158 5 10</pre>
4	<pre>A104 6 10</pre>
5	<pre>ICT114 2 10</pre>

Graphs & Diagrams

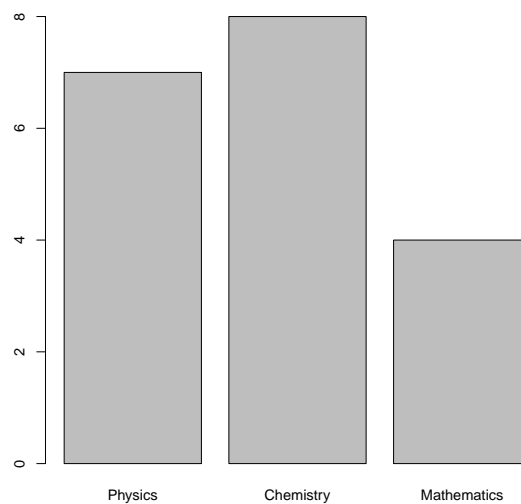
Sujal Singh – 04119051723, IIOT–B1

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

Program:

```
1 a <- 42
2 b <- 24
3 a + b
4 a * c
```

Output:

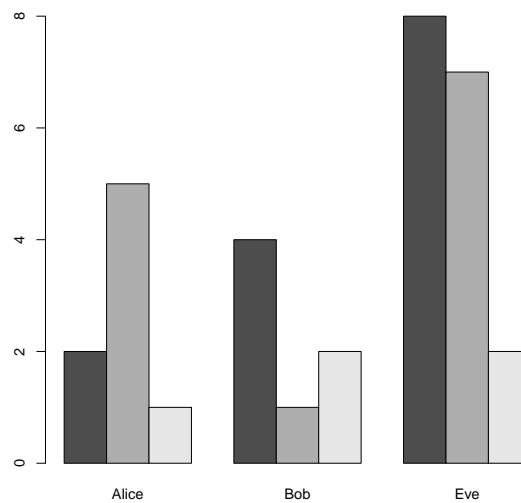


2. Using R make a grouped bar plot.

Program:

```
1 students <- c(
2   "Alice", "Bob", "Eve"
3 )
4 subjects <- c(
5   "Physics", "Chemistry", "Mathematics"
6 )
7 marks <- c(
8   2, 5, 1,
9   4, 1, 2,
10  8, 7, 2
11 )
12
13 results <- matrix(marks, ncol = 3, nrow = 3)
14 rownames(results) <- subjects
15 colnames(results) <- students
16
17 barplot(results, beside = TRUE)
```


Output:

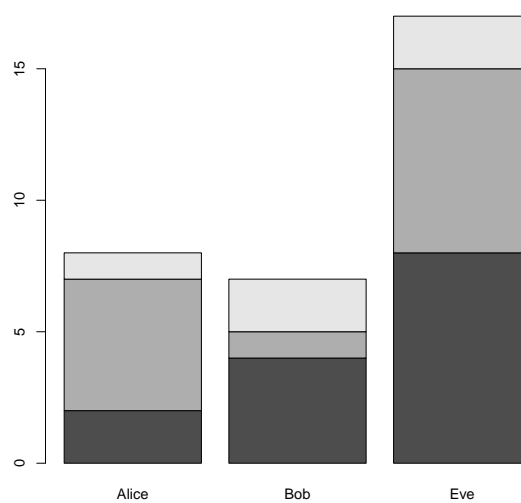


3. Using R make a grouped bar plot.

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)
```

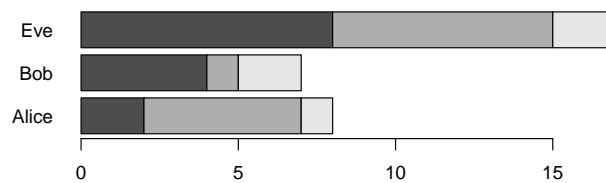
Output:



4. Make a horizontal bar plot.

Program:

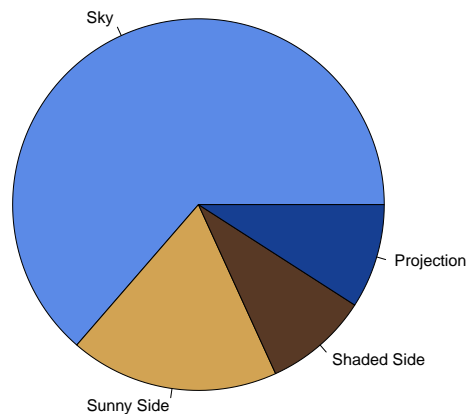
```
1 students <- c("Alice", "Bob", "Eve")
2 subjects <- c("Physics", "Chemistry", "Mathematics")
3 marks <- c(2, 5, 1, 4, 1, 2, 8, 7, 2)
4
5 results <- matrix(marks, ncol = 3, nrow = 3)
6 rownames(results) <- subjects
7 colnames(results) <- students
8
9 barplot(results, horiz = TRUE, las = 1)
```

Output:

5. Make a pie chart.

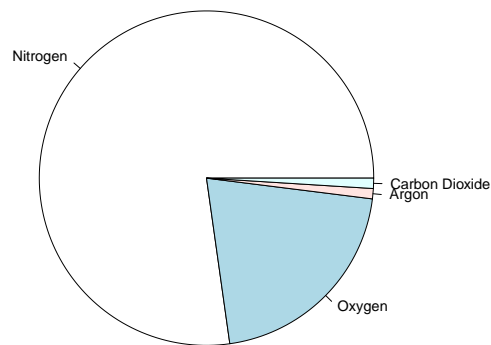
Program:

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

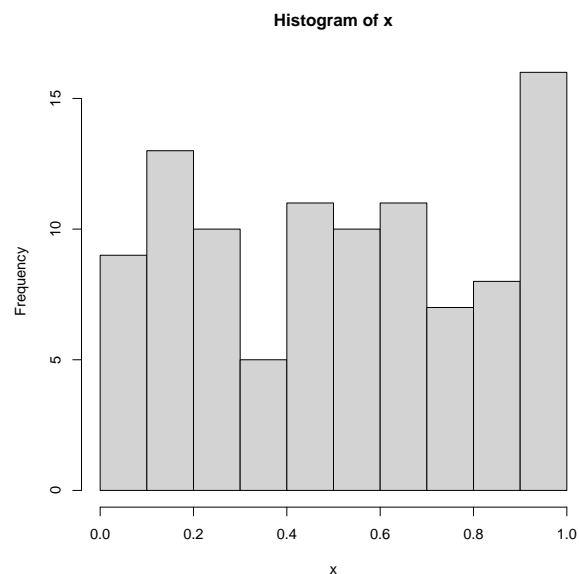
Output:

6. Make a pie chart showing atmospheric composition.**Program:**

```
1 pie(  
2   c(78, 21, 1, 1),  
3   labels = c("Nitrogen", "Oxygen", "Argon", "Carbon Dioxide"),  
4 )
```

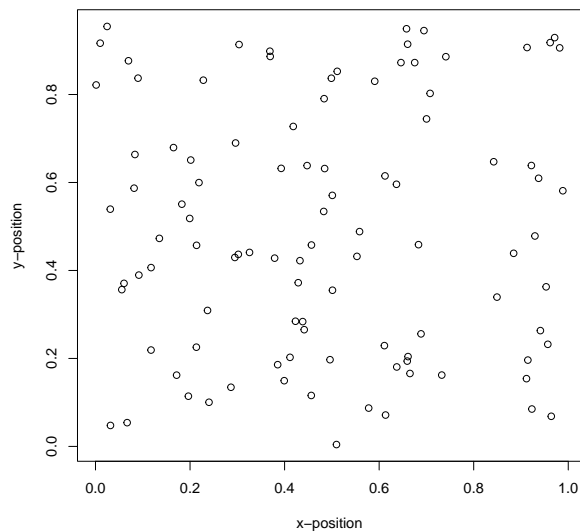
Output:**7. Make a histogram showing frequency of random numbers between 0 and 100.****Program:**

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

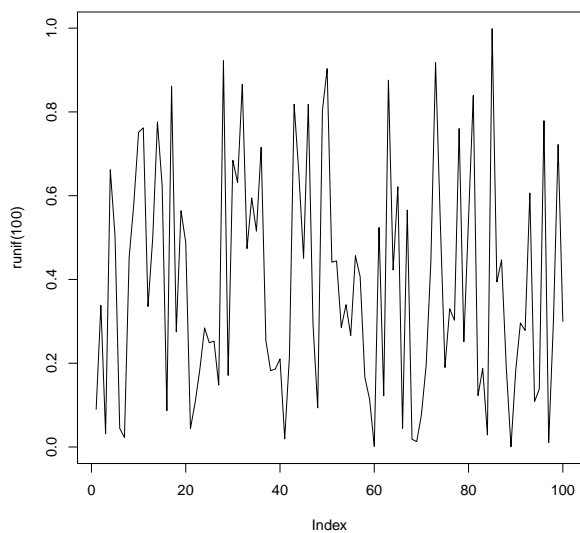
Output:

8. Generate a plot using random points.**Program:**

```
1 plot(  
2   runif(100),  
3   runif(100),  
4   xlab = "x-position",  
5   ylab = "y-position"  
6 )
```

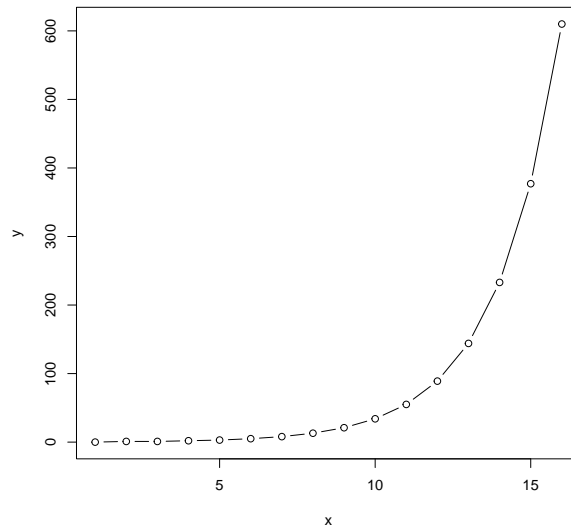
Output:**9. Make a line type plot.****Program:**

```
1 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:	
1	a <- c(28, 47, 29, 74, 47, 67, 83, 58, 93, 10)
2	mean(a)
Output:	
1	[1] 53.6

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

Program:	
1	x <- c(28, 47, 29, 74, 47, 67, 83, 58, 93, 10)
2	mean(x)
3	median(x)
Output:	
1	[1] 53.6
2	[1] 52.5

3. Caculate the geometric mean of a vector.

Program:	
1	x <- c(4, 12, 36, 108, 324, 972, 2916)
2	sqrt(x[1] * x[length(x)])
3	# OR
4	exp(mean(log(x)))
Output:	
1	[1] 108
2	[1] 108

4. Caculate the harmonic mean of a vector.

Program:	
1	library("psych")
2	x <- c(4, 12, 36, 108, 324, 972, 2916)
3	harmonic.mean(x)
Output:	
1	[1] 18.67521

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

Program:	
1	library("psych")
2	x <- c(4, 12, 36, 108, 324, 972, 2916)
3	mean(x)
4	geometric.mean(x)
5	harmonic.mean(x)
6	harmonic.mean(x)
Output:	
1	[1] 624.5714
2	[1] 108
3	[1] 18.67521

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

Program:	
1	x <- c(4, 12, 36, 108, 324, 972, 2916, 8748)
2	quantile(x, 1 / 4)
3	quantile(x, 2 / 4)
4	quantile(x, 3 / 4)
Output:	
1	25%
2	30
3	50%
4	216
5	75%
6	1458

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

Program:	
1	x <- c(4, 12, 36, 108, 324, 972, 2916, 8748)
2	quantile(x, 3 / 10)
3	quantile(x, 5 / 10)
4	quantile(x, 6 / 10)
Output:	
1	30%
2	43.2
3	50%
4	216
5	60%
6	453.6

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

Program:	
1	<code>x <- c(4, 12, NA, 108, 324, NA, 2916, 8748)</code>
2	<code>mean(x, na.rm = TRUE)</code>
3	<code>median(x, na.rm = TRUE)</code>
Output:	
1	<code>[1] 2018.667</code>
2	<code>[1] 216</code>

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

Program:	
1	<code>library("psych")</code>
2	<code>x <- c(4, 12, 36, 108, 324, 972, 2916)</code>
3	<code>mean(x, na.rm = TRUE)</code>
4	<code>geometric.mean(x, na.rm = TRUE)</code>
5	<code>harmonic.mean(x, na.rm = TRUE)</code>
Output:	
1	<code>[1] 624.5714</code>
2	<code>[1] 108</code>
3	<code>[1] 18.67521</code>

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

Program:	
1	<code>x <- c(4, 12, NA, 108, 324, NA, 2916, 8748)</code>
2	<code>quantile(x, 1 / 4, na.rm = TRUE)</code>
3	<code>quantile(x, 3 / 4, na.rm = TRUE)</code>
Output:	
1	<code>25%</code>
2	<code>36</code>
3	<code>75%</code>
4	<code>2268</code>

Measures of Dispersion

Sujal Singh – 04119051723, IIOT–B1

1. Find the range and coefficient of range.

Program:	
1	x <- floor(runif(10, min = 0, max = 100))
2	x
3	range(x)
4	<i># Coefficient of range</i>
5	(max(x) - min(x)) / (max(x) + min(x))
Output:	
1	[1] 38 31 39 44 68 78 47 12 20 27
2	[1] 12 78
3	[1] 0.7333333

2. Find the quartile deviation and its coefficients.

Program:	
1	x <- floor(runif(20, min = 0, max = 100))
2	x
3	QD <- (quantile(x, 3 / 4) - quantile(x, 1 / 4)) / 2
4	coeffQD <- (quantile(x, 3 / 4) - quantile(x, 1 / 4)) / (quantile(x, 3 / 4) + ↪ quantile(x, 1 / 4))
5	QD
6	coeffQD
Output:	
1	[1] 94 26 60 42 34 66 66 12 38 56 67 62 66 40 21 72 97 70 25 40
2	75%
3	14.625
4	75%
5	0.283293

3. Find the variance and standard deviation.

Program:	
1	x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
2	var(x)
3	sd(x)
Output:	
1	[1] 9.166667
2	[1] 3.02765

4. Find the standard deviation and coefficient of variance.

Program:	
1	x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
2	mean(x)
3	sd(x)
4	# Coefficient of variance
5	(sd(x) / mean(x)) * 100
Output:	
1	[1] 5.5
2	[1] 3.02765
3	[1] 55.04819

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

Program:	
1	x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
2	mad(x)
3	mad(x) / median(x)
4	var(x)
5	psd <- (sd(x) / mean(x)) * 100
6	psd
7	(psd / mean(x)) * 100
Output:	
1	[1] 3.7065
2	[1] 0.6739091
3	[1] 9.166667
4	[1] 55.04819
5	[1] 1000.876

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

Program:	
1	grp <- seq(0, 5)
2	f <- c(5, 35, 2, 24, 25, 35)
3	x <- rep(grp, f)
4	crange <- (max(x) - min(x)) / (max(x) + min(x))
5	QD <- (quantile(x, 3 / 4) - quantile(x, 1 / 4)) / 2
6	CQD <- (quantile(x, 3 / 4) - quantile(x, 1 / 4)) / (quantile(x, 3 / 4) + → quantile(x, 1 / 4))
7	x
8	diff(range(x))
9	crange
10	QD
11	CQD

9. Find the skewness.

Program:	
1	library(moments)
2	
3	x <- c(25, 29, 30, 17, 19, 30, 18, 28, 31, 33, 26, 28)
4	psd <- (sd(x) * sqrt(length(x) - 1)) / sqrt(length(x))
5	skp <- (3 * (mean(x) - median(x))) / psd
6	a <- quantile(x, 3 / 4)
7	b <- quantile(x, 1 / 4)
8	c <- 2 * quantile(x, 1 / 2)
9	skb <- (a + b - c) / (a - b)
10	
11	skewness(x)
Output:	
1	[1] -0.6760079

10. Find the kurtosis.

Program:	
1	library(moments)
2	
3	x <- c(25, 29, 30, 17, 19, 30, 18, 28, 31, 33, 26, 28)
4	kurtosis(x)
5	kurtosis(x) - 3
Output:	
1	[1] 2.068371
2	[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:	
1	# <i>RED</i> <i>WHITE</i> <i>BLUE</i>
2	choose(4, 1) * choose(3, 1) * choose(2, 1)
Output:	
1	[1] 24

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

Program:	
1	choose(4, 2) / choose(52, 2)
Output:	
1	[1] 0.004524887

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

(a) $P(X = 0)$

(b) $P(X = 2)$

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

Program:	
1	a1 <- dbinom(0, 10, 0.6)
2	a1
3	b1 <- dbinom(2, 10, 0.6)
4	b1
5	c1 <- dbinom(3, 10, 0.6)
6	c1
7	d1 <- dbinom(5, 10, 0.6)
8	d1
Output:	
1	[1] 0.0001048576
2	[1] 0.01061683
3	[1] 0.04246733
4	[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:

```

1 a1 <- dpois(0, 3.2)
2 a1
3 b1 <- dpois(3, 3.2)
4 b1
5 c1 <- dpois(5, 3.2)
6 c1
7 d1 <- ppois(10, 3.2)
8 d1
9 e1 <- 1 - ppois(3, 3.2)
10 e1
11 f1 <- 1 - ppois(5, 3.2)
12 f1

```

Output:

```

1 [1] 0.0407622
2 [1] 0.222616
3 [1] 0.1139794
4 [1] 0.9995028
5 [1] 0.3974803
6 [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:

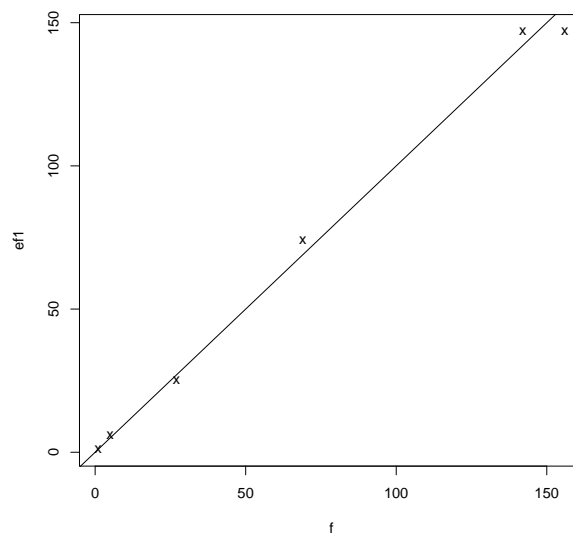
```

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

```

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

Figure:

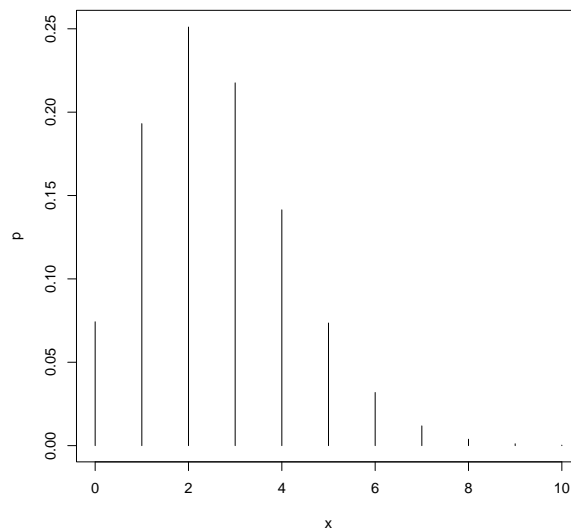
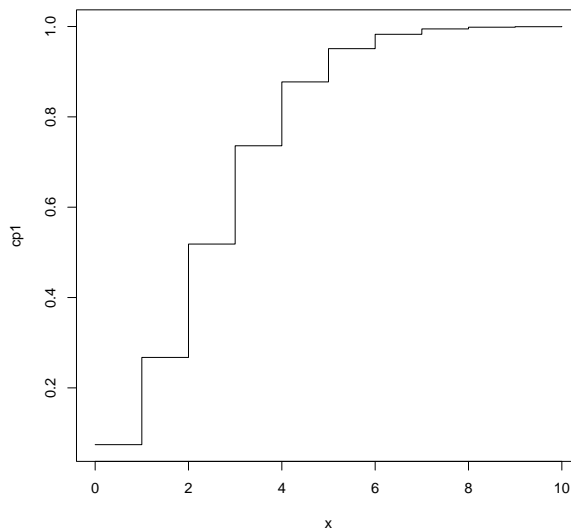


6. Plot the probability mass function (PMF) and distribution function for the following random variables $X \sim P(2.6)$.

Program:	
1	<code>m <- 2.6</code>
2	<code>x <- 0:10</code>
3	<code>p <- dpois(x, m)</code>
4	<code>d <- data.frame(x, p)</code>
5	<code>d</code>
6	<code>plot(x, p, "h")</code>
7	
8	<code>cp <- ppois(x, m)</code>
9	<code>cp1 <- round(cp, 4)</code>
10	<code>d1 <- data.frame(x, cp1)</code>
11	<code>plot(x, cp1, "s")</code>

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
12	11	0.0002889381

Figure:**Figure:**

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

Program:	
1	<code>n <- 8</code>
2	<code>p <- 0.65</code>
3	<code>x <- 0:n</code>
4	<code>bp <- dbinom(x, n, p)</code>
5	<code>d <- data.frame(x, "probabilities" = bp)</code>
6	<code>d</code>
7	<code>plot(x, bp, "h")</code>

Output:	
1	x probabilities
2	1 0 0.0002251875
3	2 1 0.0033456434
4	3 2 0.0217466823
5	4 3 0.0807733916
6	5 4 0.1875096590
7	6 5 0.2785857791
8	7 6 0.2586867948
9	8 7 0.1372623809
10	9 8 0.0318644813

Figure:

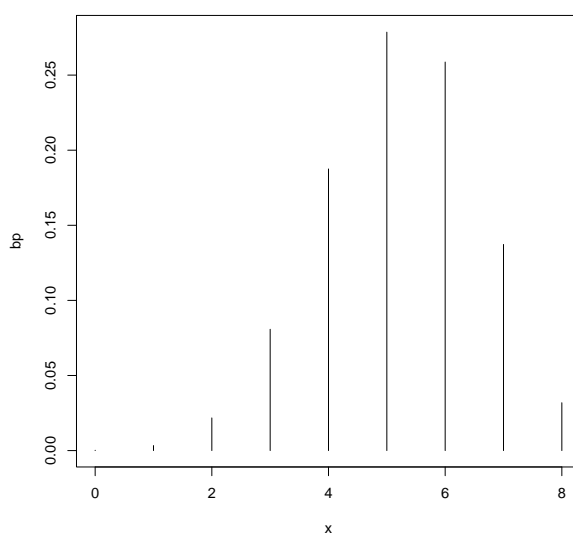


Figure:

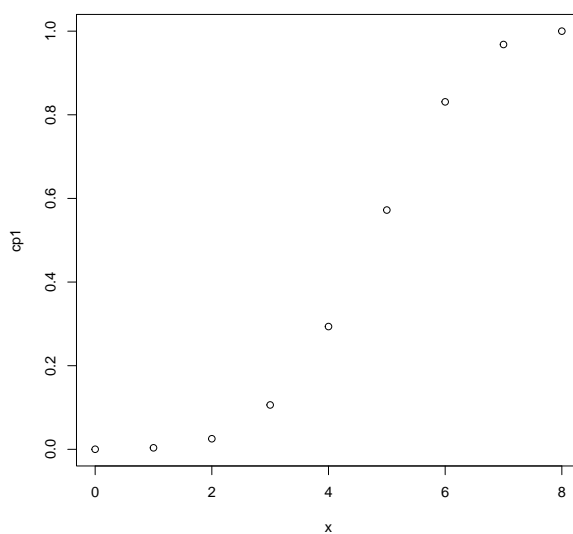
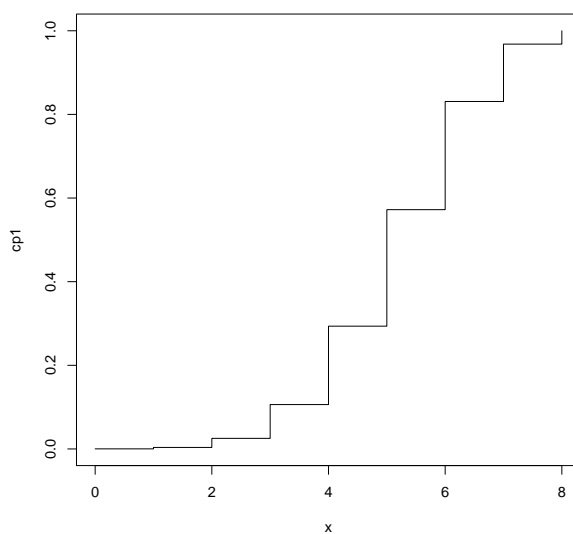


Figure:



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")

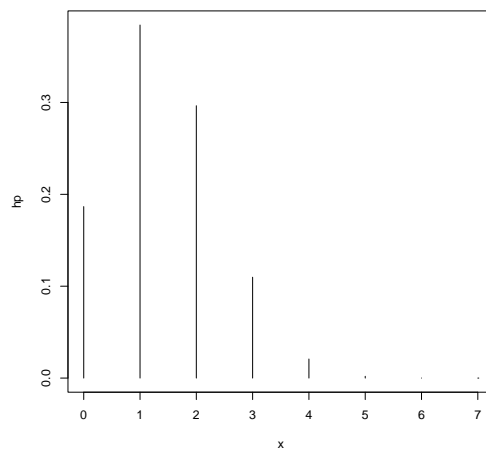
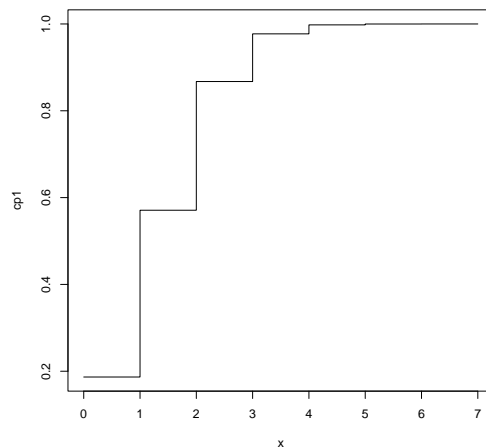
```

Output:

```

1      x      hp
2  1 0 1.866514e-01
3  2 1 3.842822e-01
4  3 2 2.964463e-01
5  4 3 1.097949e-01
6  5 4 2.077201e-02
7  6 5 1.967875e-03
8  7 6 8.409722e-05
9  8 7 1.201389e-06
10     x      cp1
11  1 0 0.1867
12  2 1 0.5709
13  3 2 0.8674
14  4 3 0.9772
15  5 4 0.9979
16  6 5 0.9999
17  7 6 1.0000
18  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:

```

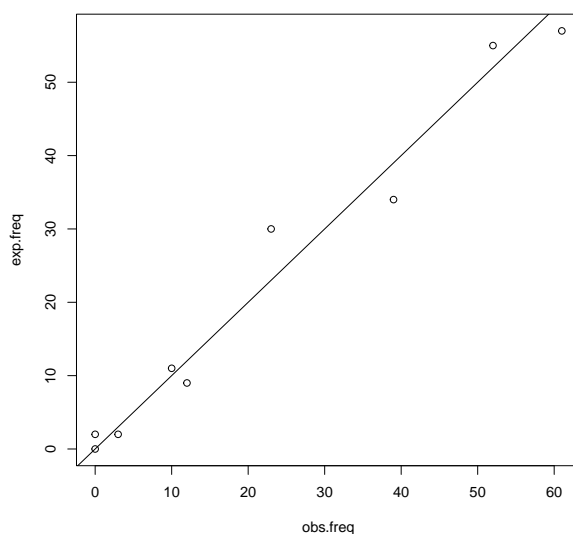
1 li <- seq(144, 180, 6); ul <- seq(150, 186, 6)
2 f <- c(3, 12, 23, 52, 61, 39, 10)
3 x <- (li + ul) / 2; n <- sum(f); k <- length(f)
4 m <- sum(f * x) / n; v <- sum(f * (x - m)^2) / n; sd <- sqrt(v)
5 l1 <- c(-9999, li, 186)
6 cp <- pnorm(l1, m, sd)
7 p <- diff(cp)
8 p <- c(p, 1 - cp[k + 2])
9 ul <- c(144, ul, 9999); f <- c(0, f, 0)
10 ef <- round(n * p, 0)
11 d <- data.frame(
12   "Lower Limit" = l1, "Upper Limit" = ul, "Obs. freq" = f,
13   "prob" = p, "cumprob" = cp, "expfreq" = ef
14 )
15 d
16 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 $X \sim N(50, 40)$. Find $P(X \leq 60)$, $P(X \geq 100)$, $P(10 \leq x \leq 20)$ and $P(X \leq k) = 0.293$.

Program:

```
1 mu <- 50; sd <- sqrt(40)
2 p1 <- pnorm(60, mu, sd)
3 p1
4 p2 <- 1 - pnorm(100, mu, sd)
5 p2
6 p3 <- pnorm(20, mu, sd)
7 p3
8 p4 <- qnorm(0.293, mu, sd)
9 p4
```

Output:

```
1 [1] 0.9430769
2 [1] 1.332268e-15
3 [1] 1.050718e-06
4 [1] 46.55538
```