

R Textbook Companion for  
Linear Algebra and Its Applications  
by David C. Lay<sup>1</sup>

Created by  
Sai Sugun L  
B.Tech.  
Electronics Engineering  
Sreenidhi Institute of Science & Technology  
Cross-Checked by  
R TBC Team

May 26, 2020

<sup>1</sup>Funded by a grant from the National Mission on Education through ICT  
- <http://spoken-tutorial.org/NMEICT-Intro>. This Textbook Companion and R  
codes written in it can be downloaded from the "Textbook Companion Project"  
section at the website - <https://r.fossee.in>.

# Book Description

**Title:** Linear Algebra and Its Applications

**Author:** David C. Lay

**Publisher:** Pearson Addison-Wesley

**Edition:** 3

**Year:** 2006

**ISBN:** 0-321-28066-0

R numbering policy used in this document and the relation to the above book.

**Exa** Example (Solved example)

**Eqn** Equation (Particular equation of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means an R code whose theory is explained in Section 2.3 of the book.

# Contents

List of R Codes	4
1 Linear Equations in Linear Algebra	5
2 Matrix Algebra	45
3 Determinants	70
4 Vector Spaces	83
5 Eigenvalues And Eigenvectors	105
6 Orthogonality And Least Squares	130
7 Symmetric Matrices	155

# List of R Codes

Exa 1.1	1 . . . . .	5
Exa 1.2	2 . . . . .	6
Exa 1.3	3 . . . . .	7
Exa 1.4	4 . . . . .	8
Exa 1.5	5 . . . . .	9
Exa 1.6	6 . . . . .	10
Exa 1.7	7 . . . . .	11
Exa 1.8	8 . . . . .	12
Exa 1.9	9 . . . . .	13
Exa 1.10	10 . . . . .	14
Exa 1.11	11 . . . . .	15
Exa 1.12	12 . . . . .	17
Exa 1.13	13 . . . . .	18
Exa 1.14	14 . . . . .	20
Exa 1.15	15 . . . . .	21
Exa 1.16	16 . . . . .	22
Exa 1.17	17 . . . . .	23
Exa 1.18	18 . . . . .	24
Exa 1.19	19 . . . . .	25
Exa 1.20	20 . . . . .	27
Exa 1.21	21 . . . . .	28
Exa 1.22	22 . . . . .	30
Exa 1.23	23 . . . . .	31
Exa 1.24	24 . . . . .	32
Exa 1.25	25 . . . . .	33
Exa 1.26	26 . . . . .	34
Exa 1.27	27 . . . . .	35
Exa 1.28	28 . . . . .	36

Exa 1.29	29	37
Exa 1.30	30	39
Exa 1.31	31	40
Exa 1.32	32	43
Exa 2.1	1	45
Exa 2.2	2	46
Exa 2.3	3	46
Exa 2.4	4	48
Exa 2.5	5	49
Exa 2.6	6	49
Exa 2.7	7	50
Exa 2.8	8	51
Exa 2.9	9	53
Exa 2.10	10	54
Exa 2.11	11	55
Exa 2.12	12	57
Exa 2.13	13	57
Exa 2.14	14	58
Exa 2.15	15	59
Exa 2.16	16	60
Exa 2.17	17	61
Exa 2.18	18	62
Exa 2.19	19	64
Exa 2.20	20	65
Exa 2.21	21	66
Exa 2.22	22	67
Exa 2.23	23	68
Exa 3.1	1	70
Exa 3.2	2	71
Exa 3.3	3	71
Exa 3.4	4	73
Exa 3.5	5	73
Exa 3.6	6	74
Exa 3.7	7	75
Exa 3.8	8	76
Exa 3.9	9	77
Exa 3.10	10	78
Exa 3.11	11	80

Exa 3.12	12	81
Exa 4.1	1	83
Exa 4.2	2	84
Exa 4.3	3	85
Exa 4.4	4	86
Exa 4.5	5	87
Exa 4.6	6	88
Exa 4.7	7	89
Exa 4.8	8	91
Exa 4.9	9	92
Exa 4.10	10	93
Exa 4.11	11	94
Exa 4.12	12	95
Exa 4.13	13	96
Exa 4.14	14	97
Exa 4.15	15	98
Exa 4.16	16	99
Exa 4.17	17	99
Exa 4.18	18	100
Exa 4.19	19	101
Exa 4.20	20	102
Exa 4.21	21	103
Exa 5.1	1	105
Exa 5.2	2	106
Exa 5.3	3	107
Exa 5.4	4	109
Exa 5.5	5	109
Exa 5.6	6	110
Exa 5.7	7	111
Exa 5.8	8	111
Exa 5.9	9	113
Exa 5.10	10	114
Exa 5.11	11	115
Exa 5.12	12	117
Exa 5.13	13	119
Exa 5.14	14	120
Exa 5.15	15	121
Exa 5.16	16	122

Exa 5.17	17	122
Exa 5.18	18	123
Exa 5.19	19	124
Exa 5.20	20	125
Exa 5.21	21	126
Exa 5.22	22	127
Exa 5.23	23	128
Exa 5.24	24	128
Exa 6.1	1	130
Exa 6.2	2	131
Exa 6.3	3	131
Exa 6.4	4	132
Exa 6.5	5	133
Exa 6.6	6	134
Exa 6.7	7	135
Exa 6.8	8	136
Exa 6.9	9	137
Exa 6.10	10	139
Exa 6.11	11	140
Exa 6.12	12	141
Exa 6.13	13	142
Exa 6.14	14	144
Exa 6.15	15	144
Exa 6.16	16	145
Exa 6.17	17	147
Exa 6.18	18	148
Exa 6.19	19	149
Exa 6.20	20	150
Exa 6.21	21	151
Exa 6.22	22	153
Exa 6.23	23	153
Exa 7.1	1	155
Exa 7.2	2	156
Exa 7.3	3	157
Exa 7.4	4	160
Exa 7.5	5	163
Exa 7.6	6	165
Exa 7.7	7	166



Exa 7.8	8 . . . . .	167
Exa 7.9	9 . . . . .	168
Exa 7.10	10 . . . . .	169
Exa 7.11	11 . . . . .	171
Exa 7.12	12 . . . . .	172
Exa 7.13	13 . . . . .	174
Exa 7.14	14 . . . . .	175

# Chapter 1

## Linear Equations in Linear Algebra

R code Exa 1.1 1

```
1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Gaussian Elimination
3 #Page No.15 / 1–3
4 #Prob 1
5 #1.1.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('performing Gaussian elimination')
14 av<-c(1,5,-2,-7)
15 a=matrix(av,
16           nrow=2,
17           ncol=2,
18           byrow=TRUE)
19 print('the co-efficient matrix is:')
```

```

20 print(a)
21 bv<-c(7,-5)
22 b=matrix(
23     bv,
24     nrow=2,
25     ncol=1,
26     byrow=TRUE)
27 cm=cbind(a,b)
28 print('the augmented matrix is:')
29 print(cm)
30 print('R2=R2+2*R1')
31 cm[2,]=cm[2,]+2*cm[1,]
32 print(cm)
33 print('R2=(1/3)*R2')
34 cm[2,]=(1/3)*cm[2,]
35 print(cm)
36 print('R1=R1-5*R2')
37 cm[1,]=cm[1,]-5*cm[2,]
38 print(cm)
39 x1=cm[1,3]/cm[1,1]
40 x2=cm[2,3]/cm[2,2]
41 cat('the solution is: x1 =',x1,' x2 =',x2)

```

---

## R code Exa 1.2 2

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Gaussian Elimination Singular case
3 #Page No.15 / 1-3
4 #Prob 7
5 #1.1.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables

```

```

11 rm(list=ls(all=TRUE))
12
13 print('the augmented matrix is:')
14 av<-c(1,7,3,-4,0,1,-1,3,0,0,0,1,0,0,1,-2)
15 a=matrix(
16     av,
17     nrow=4,
18     ncol=4,
19     byrow=TRUE)
20 print(a)
21 print('interchange R3 and R4')
22 a[c(3,4),]=a[c(4,3),]
23 print(a)
24 print('from R4 we get 0=1')
25 print('hence, no solution')

```

---

### R code Exa 1.3 3

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Gaussian Elimination with row exchange
3 #Page No.15 / 1–3
4 #Prob 13
5 #1.1.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the augmented matrix is:')
14 av<-c(1,0,-3,8,2,2,9,7,0,1,5,-2)
15 a=matrix(
16     av,
17     nrow=3,

```

```

18   ncol=4,
19   byrow=TRUE)
20   print(a)
21   print('R2=R2-2*R1')
22   a[2,]=a[2,]-2*a[1,]
23   print(a)
24   print('interchange R2 and R3')
25   a[c(2,3),]=a[c(3,2),]
26   print(a)
27   print('R3=R3-2*R2')
28   a[3,]=a[3,]-2*a[2,]
29   print(a)
30   print('R3=(1/5)*R3')
31   a[3,]=(1/5)*a[3,]
32   print(a)
33   print('R2=R2-5*R3 and R1=R1+3*R3')
34   a[2,]=a[2,]-5*a[3,]
35   a[1,]=a[1,]+3*a[3,]
36   print(a)
37   s=matrix(
38     c(a[1,4],a[2,4],a[3,4]),
39     nrow=3,
40     ncol=1,
41     byrow=TRUE)
42   print('solution is')
43   print(s)

```

---

#### R code Exa 1.4 4

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Condition for a solution to exist
3 #Page No.16 / 1-4
4 #Prob 19
5 #1.1.19
6

```

```

7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the augmented matrix for h=2')
14 av<-c(1,2,4,3,6,8)
15 a=matrix(av,
16          nrow=2,
17          ncol=3,
18          byrow=TRUE)
19 print(a)
20 print('R2-2*R1')
21 a[2,]=a[2,]-3*a[1,]
22 print(a)
23 print('from R3 we get 0=-4')
24 print('hence, if h=2 no solution, else solution
      exists')

```

---

### R code Exa 1.5 5

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Consistent System
3 #Page No.16 / 1-4
4 #Prob 25
5 #1.1.25
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the co-efficient matrix is:')

```

```

14 av<-c(1,-4,7,0,3,-5,-2,5,-9)
15 a=matrix(av,
16           nrow = 3,
17           ncol = 3,
18           byrow = TRUE)
19 print(a)
20 print('let g,h,k be the constants on RHS')
21 print('R3=R3+2*R1')
22 a[3,]=a[3,]+2*a[1,]
23 print(a)
24 print('the constants on RHS are:g,h,k+2g')
25 print('R3=R3+R2')
26 a[3,]=a[3,]+a[2,]
27 print(a)
28 print('the constants on RHS are:g,h,k+2g+h')
29 print('for solution to exist')
30 print('from R3:k+2g+h=0')

```

---

### R code Exa 1.6 6

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #General solution of the system
3 #Page No.20 / 1–8
4 #Prob 7
5 #1.2.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the augmented matrix is')
14 av<-c(1,3,4,7,3,9,7,6)
15 a=matrix(av,

```

```

16         nrow = 2,
17         ncol=4,
18         byrow=TRUE)
19 print(a)
20 print('R2=R2-3*R1')
21 a[2,]=a[2,]-3*a[1,]
22 print(a)
23 print('(-1/5)*R2')
24 a[2,]=(-1/5)*a[2,]
25 print(a)
26 print('R1=R1-4*R2')
27 a[1,]=a[1,]-4*a[2,]
28 print('the row reduced form is:')
29 print(a)
30 print('corresponding equations are')
31 print('x1+3*x2=-5 and x3=3')
32 print('free variables:x2')
33 print('general solution is:')
34 print('x1=-5-3*x2, x2, x3=3')

```

---

### R code Exa 1.7 7

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #General solution of the system
3 #Page No.20 / 1-8
4 #Prob 13
5 #1.2.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the augmented matrix is')

```



```

14 av<-c
    (1,-3,0,-1,0,-2,0,1,0,0,-4,1,0,0,0,1,9,4,0,0,0,0,0,0)

15 a=matrix(av,
16          nrow=4,
17          ncol=6,
18          byrow=TRUE)
19 print(a)
20 print('R1=R1+R3')
21 a[1,]=a[1,]+a[3,]
22 print(a)
23 print('R1=R1+3*R2')
24 a[1,]=a[1,]+3*a[2,]
25 print(a)
26 print('corresponding equations are:')
27 print('x1-3*x5=5, x2-4*x5=1, x4+9*x5=4, and 0=0')
28 print('free variables:x3, x5')
29 print('general solution is:')
30 print('x1=5+3*x5, x2=1+4*x5, x3, x4=4-9*x5, x5')

```

---

### R code Exa 1.8 8

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Linear Combination of two vectors
3 #Page No.23 / 1-11
4 #Prob 1
5 #1.3.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 uv<-c(-1,2)

```

```

14 u=matrix(uv,
15           nrow=2,
16           ncol=1,
17           byrow=TRUE)
18 print('u=')
19 print(u)
20 vv<-c(-3,-1)
21 v=matrix(vv,
22           nrow=2,
23           ncol=1,
24           byrow=TRUE)
25 print('v=')
26 print(v)
27 a=u+v
28 print('u+v=')
29 print(a)
30 s=u-2*v
31 print('u-2v=')
32 print(s)

```

---

### R code Exa 1.9 9

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Determination of Linear Combination
3 #Page No.25 / 1-13
4 #Prob 13
5 #1.3.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(1,-4,2,3,0,3,5,-7,-2,8,-4,-3)

```

```

14
15 a=matrix(
16     av,
17     nrow=3,
18     ncol=4,
19     byrow=TRUE
20 )
21
22 print('the augmented matrix is:')
23 print(a)
24
25 #Row reduce the augmented matrix to reach echelon
    form
26 print('R3=R3+2*R1')
27 a[3,]=a[3,]+2*a[1,]
28 print(a)
29 print('The system for the augmented matrix is
    inconsistent, so b is not a linear combination of
    the columns of A')

```

---

#### R code Exa 1.10 10

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Determine if b is in W
3 #Page No.25 / 1–13
4 #Prob 25
5 #1.3.25
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(1,0,-4,4,0,3,-2,1,-2,6,3,-4)

```

```

14
15 a=matrix(
16     av,
17     nrow=3,
18     ncol=4,
19     byrow=TRUE
20 )
21
22 print('the augmented matrix is:')
23 print(a)
24
25 print('R3=R3+2*R1')
26 a[3,]=a[3,]+2*a[1,]
27 print(a)
28 print('R3=R3-2*R2')
29 a[3,]=a[3,]-2*a[2,]
30 print(a)

```

---

### R code Exa 1.11 11

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Applications of Gaussian Elimination
3 #Page No.26 / 1–14
4 #Prob 31
5 #1.3.31
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('1 gram at (0,1), 1 gram at (8,1) and 1 gram
      at (2,4)')
14 m1v<-c(0,1)

```

```

15 m2v<-c(8,1)
16 m3v<-c(2,4)
17 m1=matrix(m1v,
18           nrow=2,
19           ncol=1,
20           byrow=TRUE)
21 m2=matrix(m2v,
22           nrow=2,
23           ncol=1,
24           byrow=TRUE)
25 m3=matrix(m3v,
26           nrow=2,
27           ncol=1,
28           byrow=TRUE)
29 cm=(1/3)*(1*m1+1*m2+1*m3)
30 print('centre of mass is at')
31 print(cm)
32 print('the new weight of the system=9 grams')
33 print('new centre of mass is at')
34 sv<-c(2,2)
35 s=matrix(
36   sv,
37   nrow=2,
38   ncol=1,
39   byrow=TRUE
40 )
41 print(s)
42 print('let w1,w2 and w3 be the weights added at
      (0,1),(8,1) and (2,4) respectively')
43 print('hence, w1+w2+w3=6')
44 print('using the formula for the centre of mass, we
      get')
45 print('8*w2+2*w3=8 and w1+w2+4*w3=12')
46 av<-c(1,1,1,6,0,8,2,8,1,1,4,12)
47 a=matrix(
48   av,
49   nrow=3,
50   ncol=4,

```

```

51   byrow=TRUE
52 )
53 print('the augmented matrix is:')
54 print(a)
55 print('R3=R3-R1')
56 a[3,]=a[3,]-a[1,]
57 print(a)
58 print('R3=(1/3)*R3')
59 a[3,]=(1/3)*a[3,]
60 print(a)
61 print('R2=R2-2*R3 and R1=R1-R3')
62 a[2,]=a[2,]-2*a[3,]
63 a[1,]=a[1,]-a[3,]
64 print(a)
65 print('R1=R1-(1/8)*R2')
66 a[1,]=a[1,]-(1/8)*a[2,]
67 print(a)
68 print('R2=(1/8)*R2')
69 a[2,]=(1/8)*a[2,]
70 print(a)
71 cat(sprintf('Add %.1f grams at (0,1), %.1f grams at
    (8,1) and %d grams at (2,4)', a[1,4], a[2,4], a
    [3,4])) )

```

---

## R code Exa 1.12 12

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Vectors as columns of a matrix
3 #Page No.28 / 1-16
4 #Prob 7
5 #1.4.7
6
7 #clear console
8 cat("\014")
9

```

```

10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the three vectors are:')
14 uv<-c(4,-1,7,-4)
15 vv<-c(-5,3,-5,1)
16 ww<-c(7,-8,0,2)
17 u<-matrix(uv,
18           nrow=4,
19           ncol=1,
20           byrow=TRUE)
21 v<-matrix(vv,
22           nrow=4,
23           ncol=1,
24           byrow=TRUE)
25 w<-matrix(ww,
26           nrow=4,
27           ncol=1,
28           byrow=TRUE)
29
30 print(u)
31 print(v)
32 print(w)
33 print('u v and w form the columns of A')
34 A=cbind(u,v,w)
35 print(A)
36 print('the augmented matrix is:')
37 nmv<-c(6,-8,0,-7)
38 nm<-matrix(nmv,
39            nrow=4,
40            ncol=1,
41            byrow=TRUE)
42 c=cbind(A,nmv)
43 print(c)

```

---

### R code Exa 1.13 13

```
1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Span of vectors
3 #Page No.29 / 1–17
4 #Prob 13
5 #1.4.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the augmented matrix is:')
14 av<-c(3,-5,0,-2,6,4,1,1,4)
15 a=matrix(
16     av,
17     nrow=3,
18     ncol=3,
19     byrow=TRUE
20 )
21
22 print(a)
23 print('interchange R1 and R3')
24 a[c(1,3),]=a[c(3,1),]
25 print(a)
26 print('R2=R2+2*R1 and R3=R3-3*R1')
27 a[2,]=a[2,]+2*a[1,]
28 a[3,]=a[3,]-3*a[1,]
29 print(a)
30 print('R3=R3+R2')
31 a[3,]=a[3,]+a[2,]
32 print(a)
33 print('from the entries of last row, the system is
      consistent')
34 print('hence, u is in the plane spanned by the
      columns of a')
```



---

**R code Exa 1.14 14**

```
1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Free and pivot variables
3 #Page No.33 / 1–21
4 #Prob 1
5 #1.5.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the augmented matrix is:')
14 av<-c(2,-5,8,0,-2,-7,1,0,4,2,7,0)
15 a=matrix(av,
16          nrow=3,
17          ncol=4,
18          byrow=TRUE)
19 print(a)
20 print('R2=R2+2*R1 and R3=R3-2*R1')
21 a[2,]=a[2,]+a[1,]
22 a[3,]=a[3,]-2*a[1,]
23 print(a)
24 print('R3=R3+R2')
25 a[3,]=a[3,]+a[2,]
26 print(a)
27 print('only two columns have non zero pivots')
28 print('hence, one column is a free column and
        therefore there exists a non trivial solution')
```

---

## R code Exa 1.15 15

```
1 #Chapter 1 – Linear Equations In Linear Algebra
2 #General solution of the system
3 #Page No.34 / 1–22
4 #Prob 7
5 #1.5.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the augmented matrix is:')
14 av<-c(1,3,-3,7,0,0,1,-4,5,0)
15 a=matrix(av,
16          nrow=2,
17          ncol=5,
18          byrow=TRUE)
19 print(a)
20 print('R1=R1-3*R2')
21 a[1,]=a[1,]-3*a[2,]
22 print(a)
23 print('basic variables:x1 x2')
24 print('free variables:x3 x4')
25 print('x1=-9*x3+8*x4')
26 print('x2=4*x3-5*x4')
27 print('hence, solution is')
28 print('[-9*x3+8*x4 4*x3-5*x4 x3 x4]')
29 uv<-c(-9,4,1,0)
30 vv<-c(8,-5,0,1)
31 u=matrix(uv,
32          nrow=4,
33          ncol=1,
34          byrow=TRUE
35          )
36 v=matrix(vv,
```

```

37         nrow=4,
38         ncol=1,
39         byrow=TRUE
40     )
41     print('The solution set is the same as Span{u,v}')
42     print('u=')
43     print(u)
44     print('v=')
45     print(v)

```

---

#### R code Exa 1.16 16

```

1  #Chapter 1 – Linear Equations In Linear Algebra
2  #General solution of the system
3  #Page No.34 / 1–22
4  #Prob 11
5  #1.5.11
6
7  #clear console
8  cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the augmented matrix is')
14 av<-c
      (1,-4,-2,0,3,-5,0,0,0,1,0,0,-1,0,0,0,0,0,-1,4,0,0,0,0,0,0,0,0)
15 a=matrix(av,
16         nrow=4,
17         ncol=7,
18         byrow=TRUE)
19 print(a)
20 print('R1=R1-3*R3')
21 a[1,]=a[1,]-3*a[3,]

```

```

22 print(a)
23 print('R1=R1+2*R2')
24 a[1,]=a[1,]+2*a[2,]
25 print(a)
26 print('the free variables are:x2, x4 and x6')
27 print('the basic variables are:x1, x3 and x5')
28 print('the solution is:')
29 print(' [4*x2-5*x6  x2  x6  x4  4*x6  x6] ')

```

---

### R code Exa 1.17 17

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #General solution of the system
3 #Page No.35 / 1-23
4 #Prob 13
5 #1.5.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 xv<-c(5,-2,0)
14 x=matrix(xv,
15          nrow=3,
16          ncol=1,
17          byrow=TRUE)
18 x1v<-c(4,-7,1)
19 x1=matrix(x1v,
20           nrow=3,
21           ncol=1,
22           byrow=TRUE)
23 print(x)
24 print(x1)

```

```

25 print('=p+x3*q')
26 cat('geometrically the solution set is the line
      through [', x, ','] parallel to [',x1,']')

```

---

### R code Exa 1.18 18

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Economic Equilibrium
3 #Page No.38 / 1–26
4 #Prob 1
5 #1.6.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(0.8,-0.7,0,-0.8,0.7,0)
14 a=matrix(av,
15          nrow=2,
16          ncol=3,
17          byrow=TRUE)
18 print('the augmented matrix is')
19 print(a)
20
21 #row reduce augmented matrix
22 a[2,]=a[2,]+a[1,]
23 print(a)
24 a=a*1.25
25 print(a)
26
27 print('ratio of prices is pg=0.875ps')
28 print('one equilibrium solution is ps=1000 and pg
      =875')

```

```
29 print('economic equilibrium is unaffected by a  
    proportional change in prices')
```

---

### R code Exa 1.19 19

```
1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Balanced Equation
3 #Page No.39 / 1–27
4 #Prob 7
5 #1.6.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 nahcov<-c(1,1,1,3)
14 hchov<-c(0,8,6,7)
15 nachov<-c(3,5,6,7)
16 hov<-c(0,2,0,1)
17 cov<-c(0,0,1,2)
18
19 nahco=matrix(nahcov,
20              nrow=4,
21              ncol=1,
22              byrow=TRUE)
23 hcho=matrix(hchov,
24             nrow=4,
25             ncol=1,
26             byrow=TRUE)
27 nacho=matrix(nachov,
28              nrow=4,
29              ncol=1,
30              byrow=TRUE)
```

```

31 ho=matrix(hov,
32           nrow=4,
33           ncol=1,
34           byrow=TRUE)
35 co=matrix(cov,
36           nrow=4,
37           ncol=1,
38           byrow=TRUE)
39
40 print('nahco3 ')
41 print(nahco)
42 print('h3c6h5o7 ')
43 print(hcho)
44 print('na3c6h5o7 ')
45 print(nacho)
46 print('h2o ')
47 print(ho)
48 print('co2 ')
49 print(co)
50 zm= matrix(0, nrow = 4, ncol = 1)
51 av=cbind(nahco,hcho,nacho,ho,co,zm)
52
53 print('augmented matrix')
54 av[,3]=av[,3]*-1
55 av[,4]=av[,4]*-1
56 av[,5]=av[,5]*-1
57 print(av)
58
59 amrv<-c(1,0,0,0,-1,0,0,1,0,0,-1/3,0,0,0,1,0,-1/
        3,0,0,0,0,1,-1,0)
60 amr=matrix(amrv,
61           nrow=4,
62           ncol=6,
63           byrow=TRUE)
64 print('reduced augmented matrix')
65 print(amr)
66 print('the general solution is :')
67 print('x1=x5')

```

```

68 print('x2=(1/3)*x5')
69 print('x3=(1/3)*x5')
70 print('x4=x5')
71 print('x5 is free')
72 print('if x5=3, then')
73 print('x1=x4=3')
74 print('x2=x3=1')
75 print('the balanced equation is :')
76 print('3NaHCO3+H3C6H5O7 -> Na3C6H5O7+3H2O+3CO2')

```

---

### R code Exa 1.20 20

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Intersection Equations
3 #Page No.40 / 1–28
4 #Prob 13
5 #1.6.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av1<-c
      (1,-1,0,0,0,0,-50,0,1,-1,1,-1,0,0,0,0,0,0,1,-1,60,0,0,0,1,0,-1,50
14 a1=matrix(av1,
15           nrow=5,
16           ncol=7,
17           byrow=TRUE)
18 print(a1)
19 av2<-c
      (1,-1,0,0,0,0,-50,0,1,-1,1,-1,0,0,0,0,0,1,0,-1,50,0,0,0,0,1,-1,60

```



```

20 a2=matrix(av2,
21           nrow=5,
22           ncol=7,
23           byrow=TRUE)
24 print('..by reduction')
25 print(a2)
26 av3<-c
    (1,0,-1,0,0,0,-40,0,1,-1,0,0,0,10,0,0,0,1,0,-1,50,0,0,0,0,1,-1,60

27 a3=matrix(av3,
28           nrow=5,
29           ncol=7,
30           byrow=TRUE)
31 print('..by reduction')
32 print(a3)
33 print('part-a')
34 print('the general solution is:')
35 print('x1=x3-40')
36 print('x2=x3+10')
37 print('x3 is free')
38 print('x4=x6+50')
39 print('x5=x6+60')
40 print('x6 is free')
41 print('part-b')
42 print('x3>=40')
43 print('x2>=50')
44 print('x4>=50')
45 print('x5>=60')
46 print('the minimum flows are:')
47 print('x2=50')
48 print('x3=40')
49 print('x4=50')
50 print('x5=60')

```

---

R code Exa 1.21 21

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Linear independence of vectors
3 #Page No.42 / 1–30
4 #Prob 1
5 #1.7.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('given vectors u, v and w are')
14 uv<-c(5,0,0)
15 um=matrix(uv,
16           nrow=1,
17           ncol=3,
18           byrow=TRUE)
19 u=t(um)
20 print(u)
21 vv<-c(7,2,-6)
22 vm=matrix(vv,
23           nrow=1,
24           ncol=3,
25           byrow=TRUE)
26 v=t(vm)
27 print(v)
28 ww<-c(9,4,-8)
29 wm=matrix(ww,
30           nrow=1,
31           ncol=3,
32           byrow=TRUE)
33 w=t(wm)
34 print(w)
35 print('the augmented matrix is')
36 av<-c(5,7,9,0,0,2,4,0,0,-6,-8,0)
37 a=matrix(av,
38           nrow=3,

```

```

39         ncol=4,
40         byrow=TRUE)
41 print(a)
42 print('R3=R3+3*R2')
43 a[3,]=a[3,]+3*a[2,]
44 print(a)
45 print('there are no free variables')
46 print('hence, the homogeneous equation has only
      trivial solution and the vectors are linearly
      independent')

```

---

#### R code Exa 1.22 22

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Linear dependence of vectors
3 #Page No.43 / 1–31
4 #Prob 7
5 #1.7.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the augmented matrix is')
14 Av1<-c(1,4,-3,0,0,-2,7,5,1,0,-4,-5,7,-5,0)
15 A1=matrix(Av1,
16           nrow=3,
17           ncol=5,
18           byrow=TRUE)
19 print(A1)
20 Av2<-c(1,4,-3,0,0,0,1,-1,1,0,0,11,-5,5,0)
21 A2=matrix(Av2,
22           nrow=3,

```

```

23             ncol=5,
24             byrow=TRUE)
25 print(A2)
26 Av3<-c(1,4,-3,0,0,0,1,-1,1,0,0,0,6,-6,0)
27 A3=matrix(Av3,
28           nrow=3,
29           ncol=5,
30           byrow=TRUE)
31 print(A3)
32 print('since there are three rows, the maximum
        number of pivots can be 3')
33 print('hence, at least one of the four variable must
        be free')
34 print('so the equations have non trivial solution
        and the columns of A are linearly independent')

```

---

### R code Exa 1.23 23

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Introduction to Linear Transformations
3 #Page No.46 / 1–34
4 #Prob 1
5 #1.8.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 Av<-c(2,0,0,2)
14 uv<-c(1,-3)
15 A=matrix(Av,
16         nrow=2,
17         ncol=2,

```

```

18             byrow=TRUE)
19 u=matrix(uv,
20          nrow=2,
21          ncol=1,
22          byrow=TRUE)
23 print(A)
24 print(u)
25 T=A %*% u
26 print(T)

```

---

#### R code Exa 1.24 24

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Introduction to Linear Transformations
3 #Page No.46 / 1–34
4 #Prob 5
5 #1.8.5
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av1<-c(1,-5,-7,-2,-3,7,5,-2)
14 av2<-c(1,-5,-7,-2,0,1,2,1)
15 av3<-c(1,0,3,3,0,1,2,1)
16 am1=matrix(av1,
17            nrow=2,
18            ncol=4,
19            byrow=TRUE)
20 am2=matrix(av2,
21            nrow=2,
22            ncol=4,
23            byrow=TRUE)

```

```

24 am3=matrix(av3,
25             nrow=2,
26             ncol=4,
27             byrow=TRUE)
28 print(am1)
29 print('~')
30 print(am2)
31 print('~')
32 print(am3)
33 print('x1=3-3*x3')
34 print('x2=1-2*x3')
35 print('x3 is free')
36 print('General Solution')
37 m1v<-c(3,1,0)
38 m1=matrix(m1v,
39           nrow=3,
40           ncol=1,
41           byrow=TRUE)
42 m2v<-c(-3,-2,1)
43 m2=matrix(m2v,
44           nrow=3,
45           ncol=1,
46           byrow=TRUE)
47 print(m1)
48 print('+ x3*')
49 print(m2)
50 print('when x3=0')
51 mv<-c(3,1,0)
52 m=matrix(mv,
53          nrow=3,
54          ncol=1,
55          byrow=TRUE)
56 print(m)

```

---

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Introduction to Linear Transformations
3 #Page No.47 / 1–35
4 #Prob 19
5 #1.8.19
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 m1v<-c(2,5)
14 m2v<-c(-1,6)
15 m1=matrix(m1v,
16           nrow=2,
17           ncol=1,
18           byrow=TRUE)
19 print(m1)
20 m2=matrix(m2v,
21           nrow=2,
22           ncol=1,
23           byrow=TRUE)
24 print(m2)
25 print('5y1-3y2=')
26 tm=5*m1-3*m2
27 print(tm)

```

---

#### R code Exa 1.26 26

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Matrix of a linear transformation
3 #Page No.50 / 1–38
4 #Prob 1
5 #1.9.1

```

```

6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 te1v<-c(3,1,3,1)
14 te2v<-c(-5,2,0,0)
15 te1=matrix(te1v,
16             nrow=4,
17             ncol=1,
18             byrow=TRUE)
19 print(te1)
20 te2=matrix(te2v,
21             nrow=4,
22             ncol=1,
23             byrow=TRUE)
24 print(te2)
25 A=cbind(te1,te2)
26 print(A)

```

---

#### R code Exa 1.27 27

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Matrix of a linear transformation
3 #Page No.50 / 1–38
4 #Prob 7
5 #1.9.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))

```



```

12
13 e1v<-c(-1/sqrt(2),-1/sqrt(2))
14 e1=matrix(e1v,
15           nrow=2,
16           ncol=1,
17           byrow=TRUE)
18 print('e1 ')
19 print(e1)
20 e1[2]=e1[2]*-1
21 print('→')
22 print(e1)
23
24 e2v<-c(1/sqrt(2),-1/sqrt(2))
25 e2=matrix(e2v,
26           nrow=2,
27           ncol=1,
28           byrow=TRUE)
29 print('e2 ')
30 print(e2)
31 e2[2]=e2[2]*-1
32 print('→')
33 print(e2)
34 A=cbind(e1,e2)
35 print('A')
36 print(A)

```

---

#### R code Exa 1.28 28

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Matrix of a linear transformation
3 #Page No.51 / 1–39
4 #Prob 21
5 #1.9.21
6
7 #clear console

```

```

8  cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(1,1,4,5)
14 bv<-c(3,8)
15 a=matrix(av,
16           nrow=2,
17           ncol=2,
18           byrow=TRUE)
19 print('a')
20 print(a)
21 b=matrix(bv,
22           nrow=2,
23           ncol=1,
24           byrow=TRUE)
25 print('b')
26 print(b)
27 A=cbind(a,b)
28 print('A')
29 print(A)
30 print('R2=R2-4R1')
31 A[2,]=A[2,]-4*A[1,]
32 print(A)
33 print('R1=R1-4R2')
34 A[1,]=A[1,]-A[2,]
35 print(A)
36 xv=c(7,-4)
37 x=matrix(xv,nrow=2,ncol=1,byrow = TRUE)
38 print('x=')
39 print(x)

```

---

**R code Exa 1.29** 29

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Linear Models in Business, Science and Engineering
3 #Page No.55 / 1-43
4 #Prob 1
5 #1.10.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(110,130,295,4,3,9,20,18,48,2,5,8)
14 a=matrix(av,
15          nrow=4,
16          ncol=3,
17          byrow=TRUE)
18 print(a)
19 print('~')
20 print('interchange R1 and R4')
21 a[c(1,4),]=a[c(4,1),]
22 print(a)
23 print('~')
24 a[1,]=a[1,]/2
25 a[3,]=a[3,]/2
26 print(a)
27 a1v<-c(1,2.5,4,0,-7,-7,0,-16,-16,0,-145,-145)
28 a2v<-c(1,2.5,4,0,1,1,0,0,0,0,0,0)
29 a3v<-c(1,0,1.5,0,1,1,0,0,0,0,0,0)
30 a1=matrix(a1v,
31           nrow=4,
32           ncol=3,
33           byrow=TRUE)
34 a2=matrix(a2v,
35           nrow=4,
36           ncol=3,
37           byrow=TRUE)
38 a3=matrix(a3v,

```

```

39             nrow=4,
40             ncol=3,
41             byrow=TRUE)
42 print(a1)
43 print('~')
44 print(a2)
45 print('~')
46 print(a3)

```

---

### R code Exa 1.30 30

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Linear Models in Business, Science and Engineering
3 #Page No.56 / 1-44
4 #Prob 7
5 #1.10.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 r1v<-c(12,-7,0,-4)
14 r2v<-c(-7,15,-6,0)
15 r3v<-c(0,-6,14,-5)
16 r4v<-c(-4,0,-5,13)
17 r1=matrix(r1v,
18           nrow=4,
19           ncol=1,
20           byrow=TRUE)
21 r2=matrix(r2v,
22           nrow=4,
23           ncol=1,
24           byrow=TRUE)

```

```

25 r3=matrix(r3v,
26           nrow=4,
27           ncol=1,
28           byrow=TRUE)
29 r4=matrix(r4v,
30           nrow=4,
31           ncol=1,
32           byrow=TRUE)
33 print('r1 ')
34 print(r1)
35 print('r2 ')
36 print(r2)
37 print('r3 ')
38 print(r3)
39 print('r4 ')
40 print(r4)
41 R=cbind(r1,r2,r3,r4)
42 print(R)
43 vv<-c(40,30,20,-10)
44 v=matrix(vv,
45           nrow=4,
46           ncol=1,
47           byrow=TRUE)
48 print('v ')
49 print(v)
50 x<-solve(R,v)
51 print('the solution is ')
52 print(x)

```

---

### R code Exa 1.31 31

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Linear Models in Business, Science and Engineering
3 #Page No.57 / 1–45
4 #Prob 13

```

```

5 #1.10.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 mv<-c(0.95,0.03,0.05,0.97)
14 x0v<-c(600000,400000)
15 m=matrix(mv,
16           nrow=2,
17           ncol=2,
18           byrow=TRUE)
19 print('M')
20 print(m)
21 x0=matrix(x0v,
22           nrow=2,
23           ncol=1,
24           byrow=TRUE)
25 print('when X0=')
26 print(x0)
27 x5v<-c(523293,476707)
28 x5=matrix(x5v,
29           nrow=2,
30           ncol=1,
31           byrow=TRUE)
32 print('x5 ')
33 print(x5)
34 x10v<-c(472737,527263)
35 x10=matrix(x10v,
36            nrow=2,
37            ncol=1,
38            byrow=TRUE)
39 print('x10 ')
40 print(x10)
41 x15v<-c(439417,560583)
42 x15=matrix(x15v,

```

```

43             nrow=2,
44             ncol=1,
45             byrow=TRUE)
46 print('x15 ')
47 print(x15)
48 x20v<-c(417456,582544)
49 x20=matrix(x20v,
50           nrow=2,
51           ncol=1,
52           byrow=TRUE)
53 print('x20 ')
54 print(x20)
55 x0=matrix(x0v,
56          nrow=2,
57          ncol=1,
58          byrow=TRUE)
59 x0v<-c(350000,650000)
60 print('when X0=')
61 x0=matrix(x0v,
62          nrow=2,
63          ncol=1,
64          byrow=TRUE)
65 print(x0)
66 x5v<-c(358523,641477)
67 x5=matrix(x5v,
68          nrow=2,
69          ncol=1,
70          byrow=TRUE)
71 print('x5 ')
72 print(x5)
73 x10v<-c(364140,635860)
74 x10=matrix(x10v,
75           nrow=2,
76           ncol=1,
77           byrow=TRUE)
78 print('x10 ')
79 print(x10)
80 x15v<-c(367843,632157)

```

```

81 x15=matrix(x15v,
82             nrow=2,
83             ncol=1,
84             byrow=TRUE)
85 print('x15')
86 print(x15)
87 x20v<-c(370283,629717)
88 x20=matrix(x20v,
89             nrow=2,
90             ncol=1,
91             byrow=TRUE)
92 print('x20')
93 print(x20)

```

---

#### R code Exa 1.32 32

```

1 #Chapter 1 – Linear Equations In Linear Algebra
2 #Supplementary Exercises
3 #Page No.58 / 1–46
4 #Prob 7a
5 #7a
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 v1v<-c(2,-5,7)
14 v2v<-c(-4,1,-5)
15 v3v<-c(-2,1,-3)
16
17 v1=matrix(v1v,3,1,TRUE)
18 v2=matrix(v2v,3,1,TRUE)
19 v3=matrix(v3v,3,1,TRUE)

```



```
20 print('v1')
21 print(v1)
22 print('v2')
23 print(v2)
24 print('v3')
25 print(v3)
26 R=cbind(v1,v2,v3)
27 print(R)
28 R[1,]=R[1,]/2
29 print('~')
30 print(R)
31 Rnv<-c(1,-2,-1,0,-9,-4,0,9,4)
32 Rn=matrix(Rnv,3,3,TRUE)
33 print('~')
34 print(Rn)
35 Rnv<-c(1,-2,-1,0,-9,-4,0,0,0)
36 Rn=matrix(Rnv,3,3,TRUE)
37 print('~')
38 print(Rn)
```

---

# Chapter 2

## Matrix Algebra

R code Exa 2.1 1

```
1 #Chapter 2- Matrix Algebra
2 #Matrix operations
3 #Page No.64 / 2-2
4 #Prob 1
5 #2.1.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(2,0,-1,4,-5,2)
14 a=matrix(av,2,3,TRUE)
15 print('A=')
16 print(a)
17 print('-2A=')
18 print(-2*a)
19 print('matrix B')
20 bv<-c(7,-5,1,1,-4,-3)
21 b=matrix(bv,2,3,TRUE)
```

```

22 print(b)
23 print('B-2A')
24 print(b-2*a)
25 c=matrix(c(1,2,-2,1),2,2,TRUE)
26 d=matrix(c(3,5,-1,4),2,2,TRUE)
27 cd=c %*% d
28 print('CD')
29 print(cd)

```

---

### R code Exa 2.2 2

```

1 #Chapter 2- Matrix Algebra
2 #Inverse of a matrix
3 #Page No.67 / 2-5
4 #Prob 1
5 #2.2.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('given matrix:')
14 av<-c(8,6,5,4)
15 a=matrix(av,2,2,TRUE)
16 print(a)
17 print('inverse of the matrix is:')
18 print(solve(a))

```

---

### R code Exa 2.3 3

```

1 #Chapter 2- Matrix Algebra

```

```

2 #Inverse of a matrix
3 #Page No.67 / 2-5
4 #Prob 7
5 #2.2.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the co-efficient matrix is:')
14 av<-c(1,2,5,12)
15 a=matrix(av,2,2,TRUE)
16 print(a)
17 print('inverse of the matrix is:')
18 print(solve(a))
19 print('solution is:')
20
21 b1v<-c(-1,3)
22 b1=matrix(b1v,2,1,TRUE)
23 c=solve(a)
24 res=c %*% b1
25 print(res)
26
27 b2v<-c(1,-5)
28 b2=matrix(b2v,2,1,TRUE)
29 res1=c %*% b2
30 print(res1)
31
32 b3v<-c(2,6)
33 b3=matrix(b3v,2,1,TRUE)
34 res2=c %*% b3
35 print(res2)
36
37 b4v<-c(3,5)
38 b4=matrix(b4v,2,1,TRUE)
39 res3=c %*% b4

```

```

40 print(res3)
41
42 cm=cbind(a,b1,b2,b3,b4)
43 print(cm)
44 cm[2,]=cm[2,]-5*cm[1,]
45 print(cm)
46 cm[2,]=cm[2,]/2
47 print(cm)
48 cm[1,]=cm[1,]-2*cm[2,]
49 print(cm)

```

---

#### R code Exa 2.4 4

```

1 #Chapter 2- Matrix Algebra
2 #Matrix
3 #Page No.70 / 2-8
4 #Prob 31
5 #2.2.31
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(1,0,-2,-3,1,4,2,-3,4)
14 a=matrix(av,3,3,TRUE)
15 i=diag(3)
16 ai=cbind(a,i)
17 print(ai)
18 ai[2,]=ai[2,]+3*ai[1,]
19 print(ai)
20 ai[3,]=ai[3,]-2*ai[1,]
21 print(ai)
22 ai[3,]=ai[3,]+3*ai[2,]

```

```

23 print(ai)
24 ai[1,]=ai[1,]+ai[3,]
25 print(ai)
26 ai[2,]=ai[2,]+ai[3,]
27 print(ai)
28 ai[3,]=ai[3,]/2
29 print(ai)
30 print('A-1')
31 print(ai[,c(4,5,6)])

```

---

#### R code Exa 2.5 5

```

1 #Chapter 2– Matrix Algebra
2 #Invertibility of a matrix
3 #Page No.73 / 2–11
4 #Prob 1
5 #2.3.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 av<-c(5,7,-3,-6)
15 a=matrix(av,2,2,TRUE)
16 print(a)
17 print('the columns are linearly independent')
18 print('hence, by invertible matrix theorem')
19 print('the matrix A is invertible')

```

---

#### R code Exa 2.6 6

```

1 #Chapter 2– Matrix Algebra
2 #Invertible matrix
3 #Page No.73 / 2–11
4 #Prob 7
5 #2.3.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(-1,-3,0,1,3,5,8,-3,-2,-6,3,2,0,-1,2,1)
14 a=matrix(av,4,4,TRUE)
15 print(a)
16 a[1,]=a[1,]*-1
17 a[2,]=a[2,]-3*a[1,]
18 a[3,]=a[3,]+2*a[1,]
19 a[1,]=a[1,]*-1
20 print(a)
21 a[4,]=4*a[4,]-a[2,]
22 a[4,]=a[4,]/4
23 print(a)

```

---

#### R code Exa 2.7 7

```

1 #Chapter 2– Matrix Algebra
2 #Invertible matrix theorem
3 #Page No.74 / 2–12
4 #Prob 33
5 #2.3.33
6
7 #clear console
8 cat("\014")
9

```

```

10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('matrix A corresponding to transformation T is
      :')
14 Av<-c(-5,9,4,-7)
15 A=matrix(Av,2,2,TRUE)
16 print(A)
17 print('determinant of A is:')
18 print(det(A))
19 print('since det(A) is not equal to zero')
20 print('by IMT, A is invertible')
21 print('hence, the inverse of A exists')
22 print('inverse of A is:')
23 print(solve(A))

```

---

#### R code Exa 2.8 8

```

1 #Chapter 2- Matrix Algebra
2 #Inverse using matrix partition
3 #Page No.81 / 2-19
4 #Prob 25
5 #2.4.25
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('given matrix is:')
14 av<-c
      (1,2,0,0,0,3,5,0,0,0,0,0,2,0,0,0,0,0,7,8,0,0,0,5,6)
15 a=matrix(av,5,5,TRUE)

```



```

16 print(a)
17 print('partitioning the matrix into 4 submatrices')
18 A11=rbind(a[1,1:2],a[2,1:2])
19 print('A11=')
20 print(A11)
21 A22=rbind(a[3,3:5],a[4,3:5],a[5,3:5])
22 print('A22=')
23 print(A22)
24 A12=matrix(0,2,3,TRUE)
25 print('A12=')
26 print(A12)
27 A21=matrix(0,3,2,TRUE)
28 print('A21=')
29 print(A21)
30 print('partitioning A22 into 4 submatrices')
31 A221=matrix(2,1,1,TRUE)
32 print(A221)
33 B=rbind(A22[2,2:3],A22[3,2:3])
34 print('B=')
35 print(B)
36 print(matrix(0,1,2,TRUE))
37 print(matrix(0,2,1,TRUE))
38 print('determinant of B=')
39 print(det(B))
40 print('Hence, B is invertible')
41 print('inverse of B is')
42 print(solve(B))
43 print('determinant of inverse of B is:')
44 print(det(solve(B)))
45 print('inverse of A11')
46 print(solve(A11))
47 print('hence the inverse of A22 is:')
48 v1=det(solve(B))
49 v2=matrix(0,1,2,TRUE)
50 v3=matrix(c(0,3,-4),1,3,TRUE)
51 v4=matrix(c(0,-2.5,3.5),1,3,TRUE)
52 r1=cbind(v1,v2)
53 c=rbind(r1,v3,v4)

```

54 `print(c)`

---

### R code Exa 2.9 9

```
1 #Chapter 2- Matrix Algebra
2 #Application of LU decomposition
3 #Page No.83 / 2-21
4 #Prob 1
5 #2.5.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the lower triangular matrix is:')
14 Lv<-c(1,0,0,-1,1,0,2,-5,1)
15 L=matrix(Lv,3,3,TRUE)
16 print(L)
17 print('the upper triangular matrix is:')
18 Uv<-c(3,-7,-2,0,-2,-1,0,0,-1)
19 U=matrix(Uv,3,3,TRUE)
20 print(U)
21 print('the RHS of the equations are')
22 bv<-c(-7,5,2)
23 b=matrix(bv,3,1,TRUE)
24 print(b)
25 print('combining matrices L and b')
26 cm=cbind(L,b)
27 print(cm)
28 print('performing row operations')
29 print('R2=R2+R1')
30 cm[2,]=cm[2,]+cm[1,]
31 print(cm)
```

```

32 print('R3=R3-2*R1')
33 cm[3,]=cm[3,]-2*cm[1,]
34 print(cm)
35 print('R3=R3+5*R2')
36 cm[3,]=cm[3,]+5*cm[2,]
37 print(cm)
38 y=cm[,4]
39 print('y=')
40 print(y)
41 print('combining U and y')
42 d=cbind(U,y)
43 print(d)
44 print('performing row operations')
45 print('R3=R3/-6')
46 d[3,]=d[3,]/(-1)
47 print(d)
48 print('R2=R2+R3 and R1=R1+2*R3')
49 d[2,]=d[2,]+d[3,]
50 d[1,]=d[1,]+2*d[3,]
51 print(d)
52 print('R1=R1-3.5*R2')
53 d[1,]=d[1,]-3.5*d[2,]
54 print(d)
55 print('R1=R1/3 and R2=R2/-2')
56 d[1,]=d[1,]/3
57 d[2,]=d[2,]/(-2)
58 print(d)
59 print('the solution is:')
60 x=d[,4]
61 print('x=')
62 print(x)

```

---

**R code Exa 2.10** 10

1 #Chapter 2– Matrix Algebra

```

2 #LU decomposition of a matrix
3 #Page No.84 / 2-22
4 #Prob 7
5 #2.5.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('given matrix is:')
14 av<-c(2,5,-3,-4)
15 a=matrix(av,2,2,TRUE)
16 d=a;
17 print(a)
18 print('performing row operations')
19 a[2,]=a[2,]-(a[2,1]/a[1,1])*a[1,]
20 print(a)
21 print(a)
22 print('thus, the upper triangular matrix is')
23 U=a;
24 print('U=')
25 print(U)
26 print('the lower triangular matrix is:')
27 c2=d[2,1]/d[1,1]
28 r1=matrix(c(1,c2),2,1,TRUE)
29 r2=matrix(c(0,1),2,1,TRUE)
30 L=cbind(r1,r2)
31 print('L=')
32 print(L)

```

---

**R code Exa 2.11 11**

```

1 #Chapter 2- Matrix Algebra

```

```

2 #LU decomposition of a matrix
3 #Page No.84 / 2-22
4 #Prob 13
5 #2.5.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('given matrix is:')
14 av<-c(1,3,-5,-3,-1,-5,8,4,4,2,-5,-7,-2,-4,7,5)
15 a=matrix(av,4,4,TRUE)
16 d=a
17 print(a)
18 print('performing row operations')
19 p21=a[2,1]/a[1,1]
20 p31=a[3,1]/a[1,1]
21 p41=a[4,1]/a[1,1]
22 a[2,]=a[2,]-p21*a[1,]
23 a[3,]=a[3,]-p31*a[1,]
24 a[4,]=a[4,]-p41*a[1,]
25 print(a)
26 p32=a[3,2]/a[2,2]
27 p42=a[4,2]/a[2,2]
28 a[3,]=a[3,]-p32*a[2,]
29 a[4,]=a[4,]-p42*a[2,]
30 print(a)
31 print('thus, lower triangular matrix is:')
32 v1=matrix(c(1,0,0,0),1,4,TRUE)
33 v2=matrix(c(p21,1,0,0),1,4,TRUE)
34 v3=matrix(c(p31,p32,1,0),1,4,TRUE)
35 v4=matrix(c(p41,p42,0,1),1,4,TRUE)
36 L=rbind(v1,v2,v3,v4)
37 print('L=')
38 print(L)
39 print('Upper triangular matrix is:')

```

```
40 print('U=')
41 print(a)
```

---

#### R code Exa 2.12 12

```
1 #Chapter 2– Matrix Algebra
2 #Application of matrix algebra
3 #Page No.91 / 2–29
4 #Prob 1
5 #2.6.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the consumption matrix is:')
14 cv<-c(.1,.6,.6,.3,.2,0,.3,.1,.1)
15 cm=matrix(cv,3,3,TRUE)
16 print(cm)
17 print('Assuming that agriculture plans to produce
      100 units and other units produce nothing')
18 print('the production vector is given by')
19 x=matrix(c(0,100,0),3,1,TRUE)
20 print('x=')
21 print(x)
22 print('thus the intermediate demand is:')
23 print(cm %*% x)
```

---

#### R code Exa 2.13 13

```
1 #Chapter 2– Matrix Algebra
```

```

2 #Application of matrix algebra
3 #Page No.91 / 2-29
4 #Prob 7
5 #2.6.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the consumption matrix is:')
14 Cv<-c(0,.5,.6,.2)
15 Cm=matrix(Cv,2,2,TRUE)
16 print(Cm)
17 print('the demand for 1 unit of output sector 1')
18 d1=matrix(c(1,0),2,1,TRUE)
19 print(d1)
20 print('the production required to satisfy demand d1
      is:')
21 x1=solve(diag(2)-Cm) %*% d1
22 print('x1=')
23 print(x1)
24 print('the final demand is:')
25 d2=matrix(c(51,30),2,1,TRUE)
26 print('d2=')
27 print(d2)
28 print('the production required to satisfy demand d2
      is:')
29 x2=solve(diag(2)-Cm) %*% d2
30 print('x2=')
31 print(x2)

```

---

**R code Exa 2.14** 14

```

1 #Chapter 2– Matrix Algebra
2 #Application of matrix algebra
3 #Page No.93 / 2–31
4 #Prob 7
5 #2.7.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the matrix in R2 to rotate a vector by 60
      degrees is:')
14 av<-c(cos(pi/3),-sin(pi/3),sin(pi/3),cos(pi/3))
15 a=matrix(av,2,2,TRUE)
16 print(a)
17 x=matrix(c(6,8),2,1,TRUE)
18 print('x=')
19 print(x)
20 print('so the 3X3 matrix for rotation about x is:')
21 yv<-c(1,0,6,0,1,8,0,0,1)
22 y=matrix(yv,3,3,TRUE)
23 zv<-c(1,0,-6,0,1,-8,0,0,1)
24 z=matrix(zv,3,3,TRUE)
25 av<-c(cos(pi/3),-sin(pi/3),0,sin(pi/3),cos(pi/3)
      ,0,0,0,1)
26 a=matrix(av,3,3,TRUE)
27 R=y %*% (a %*% z)
28 print(R)

```

---

#### R code Exa 2.15 15

```

1 #Chapter 2– Matrix Algebra
2 #R3 coordinates of matrix

```



```

3 #Page No.94 / 2-32
4 #Prob 19
5 #2.7.19
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 Pv<-c(1,0,0,0,0,1,0,0,0,0,0,0,0,0,-0.1,1)
14 P=matrix(Pv,4,4,TRUE)
15 print(P)
16 Dv<-c(4.2,6,2,1.2,4,2,4,2,6,1,1,1)
17 D=matrix(Dv,4,3,TRUE)
18 print(D)
19 PD=P %*% D
20 print('PD')
21 print(PD)
22 R3=matrix(0,3,3,TRUE)
23 R31=PD[1:3,1]/PD[4,1]
24 R32=PD[1:3,2]/PD[4,2]
25 R33=PD[1:3,3]/PD[4,3]
26 R3=cbind(R31,R32,R33)
27 print(R3)

```

---

#### R code Exa 2.16 16

```

1 #Chapter 2- Matrix Algebra
2 #Column space of a matrix
3 #Page No.95 / 2-33
4 #Prob 7c
5 #2.8.7c
6
7 #clear console

```

```

8  cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 Av<-c(2,-3,-4,-8,8,6,6,-7,-7)
15 A=matrix(Av,3,3,TRUE)
16 print('A=')
17 print(A)
18 print('the given vector is:')
19 p=matrix(c(6,-10,11),3,1,TRUE)
20 print('p=')
21 print(p)
22 print('combining A and p')
23 b=cbind(A,p)
24 print(b)
25 print('performing row operations')
26 b[2,]=b[2,]-(b[2,1]/b[1,1])*b[1,]
27 b[3,]=b[3,]-(b[3,1]/b[1,1])*b[1,]
28 print(b)
29 b[3,]=b[3,]-(b[3,2]/b[2,2])*b[2,]
30 print(b)
31 if(b[3,3]==0 & b[3,4]==0){
32   print('p lies in column space of A')
33 }else{
34   print('p does not lie in column space of A')
35 }

```

---

## R code Exa 2.17 17

```

1 #Chapter 2– Matrix Algebra
2 #Matrix operation
3 #Page No.96 / 2–34
4 #Prob 13

```

```

5 #2.8.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(3,2,1,-5,0,-9,-4,1,7,0,9,2,-5,1,0)
14 a=matrix(av,3,5,TRUE)
15 print(a)
16 a[2,]=a[2,]+a[3,]
17 a[2,]=a[2,]/(-1)
18 a[3,]=a[3,]-3*a[1,]
19 print(a)
20 a[3,]=a[3,]+2*a[2,]
21 print(a)
22 a[2,]=a[2,]/2
23 print(a)
24 a[1,]=a[1,]-2*a[2,]
25 a[1,]=a[1,]/3
26 print(a)
27 print('the generalsolution is  $x_1=x_3-x_4$ ')
28 print('and  $x_2=-2x_3+4x_4$ ')
29 print('x3 and x4 are free')
30 print('if  $x_3=1$  and  $x_4=0$ ')
31 print('vector is  $(1, -2, 1, 0)$ ')
32 print('if  $x_3=0$  and  $x_4=1$ ')
33 print('vector is  $(-1, 4, 0, 1)$ ')

```

---

#### R code Exa 2.18 18

```

1 #Chapter 2- Matrix Algebra
2 #Pivot columns
3 #Page No.96 / 2-34

```

```

4 #Prob 23
5 #2.8.23
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 av<-c(4,5,9,-2,6,5,1,12,3,4,8,-3)
15 a=matrix(av,3,4,TRUE)
16 print(a)
17 print('performing row operations')
18 a[1,]=a[1,]*(1/4)
19 a[2,]=a[2,]-6*a[1,]
20 a[3,]=a[3,]-3*a[1,]
21 a[2,]=a[2,]*(-2/5)
22 a[3,]=a[3,]+(-1/4)*a[2,]
23 a[1,]=a[1,]+(-5/4)*a[2,]
24 a[1,]=a[1,]+2*a[2,]
25 print(a)
26 i1<-c(1:3)
27 j1<-c(1:4)
28 for(i in i1){
29   for(j in j1){
30     if(a[i,j]!=0){
31       cat('column ',j,' is a pivot column\n')
32       break
33     }
34   }
35 }
36 a[1,]=a[1,]-2*a[2,]
37 zm=matrix(0,3,1,TRUE)
38 c=cbind(a,zm)
39 print(c)
40 print('x1-4x3+7x4=0')
41 print('x2+5x3-6x4=0')

```

```
42 print('0=0')
```

---

### R code Exa 2.19 19

```
1 #Chapter 2- Matrix Algebra
2 #Pivot columns
3 #Page No.97 / 2-35
4 #Prob 25
5 #2.8.25
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 av<-c(1,4,8,-3,-7,-1,2,7,3,4,-2,2,9,5,5,3,6,9,-5,-2)
15 a=matrix(av,4,5,TRUE)
16 print(a)
17 print('performing row operations')
18 a[2,]=a[2,]+a[1,]
19 a[3,]=a[3,]+2*a[1,]
20 a[4,]=a[4,]-3*a[1,]
21 a[2,]=a[2,]*(1/6)
22 a[3,]=a[3,]-10*a[2,]
23 a[4,]=a[4,]+6*a[2,]
24 a[3,]=a[3,]*-1
25 a[4,]=a[4,]-4*a[3,]
26 a[1,]=a[1,]+3*a[3,]
27 a[2,]=a[2,]*2
28 print(a)
29 il<-c(1:4)
30 jl<-c(1:5)
31 for(i in il){
```

```

32     for(j in j1){
33         if(a[i,j]!=0){
34             cat('column ',j,' is a pivot column\n')
35             break
36         }
37     }
38 }
39 a[2,]=a[2,]/2
40 a[1,]=a[1,]-4*a[2,]
41 print(a)
42 print('x1-2x3+7x5=0')
43 print('x2+2.5x3-0.5x5=0')
44 print('x4+4x5=0')
45 print('0=0')
46 print('x1=2x3-7x5')
47 print('x2=-2.5x3+0.5x5')
48 print('x3=x3')
49 print('x4=-4x5')
50 print('x5=x5')

```

---

## R code Exa 2.20 20

```

1 #Chapter 2- Matrix Algebra
2 #Pivot columns
3 #Page No.98 / 2-36
4 #Prob 37
5 #2.8.37
6
7 #clear console
8 cat("\014")
9 #clear variables
10 rm(list=ls(all=TRUE))
11 av<-c
    (3,-5,0,-1,3,-7,9,-4,9,-11,-5,7,-2,5,-7,3,-7,-3,4,0)

```

```

12 a=matrix(av,4,5,TRUE)
13 print(a)
14 a[1,]=a[1,]*(1/3)
15 a[2,]=a[2,]+7*a[1,]
16 a[3,]=a[3,]+5*a[1,]
17 a[4,]=a[4,]+(-3)*a[1,]
18 a[2,]=a[2,]*(-3/8)
19 a[3,]=a[3,]+(4/3)*a[2,]
20 a[4,]=a[4,]+2*a[2,]
21 a[1,]=a[1,]+(5/3)*a[2,]
22 a[4,]=ceiling(a[4,])#the numbers are very small
23 a[3,]=floor(a[3,])#the numbers are very small
24 a[1,2]=ceiling(a[1,2])
25 print(a)
26 il<-c(1:4)
27 j1<-c(1:5)
28 for(i in il){
29   for(j in j1){
30     if(a[i,j]!=0){
31       cat('column ',j,' is a pivot column\n')
32       break
33     }
34   }
35 }
36 print('x1+2.5x3-4.5x4+3.5x5=0')
37 print('x2+1.5x3-2.5x4+1.5x5=0')
38 print('x3,x4,x5 are free')

```

---

## R code Exa 2.21 21

```

1 #Chapter 2- Matrix Algebra
2 #Pivot columns
3 #Page No.101 / 2-39
4 #Prob 1
5 #2.9.1

```

```

6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 xs=matrix(c(3,2),2,1,TRUE)
14 print(xs)
15 b1=matrix(c(1,1),2,1,TRUE)
16 b2=matrix(c(2,-1),2,1,TRUE)
17 print(b1)
18 print(b2)
19 x=3*b1+2*b2
20 print(x)

```

---

## R code Exa 2.22 22

```

1 #Chapter 2– Matrix Algebra
2 #Pivot columns
3 #Page No.102 / 2–40
4 #Prob 7
5 #2.9.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 xs=matrix(c(2,-1),2,1,TRUE)
14 print(xs)
15 xs1=matrix(c(1.5,0.5),2,1,TRUE)
16 print(xs1)
17 b1=matrix(c(3,0),2,1,TRUE)

```



```

18 b2=matrix(c(-1,2),2,1,TRUE)
19 print(b1)
20 print(b2)
21 x=1.5*b1+0.5*b2
22 print(x)

```

---

### R code Exa 2.23 23

```

1 #Chapter 2- Matrix Algebra
2 #Dimension of a matrix
3 #Page No.102 / 2-40
4 #Prob 13
5 #2.9.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 av<-c(1,-3,2,-4,-3,9,-1,5,2,-6,4,-3,-4,12,2,7)
15 a=matrix(av,4,4,TRUE)
16 print(a)
17 print('performing row operations')
18 a[2,]=a[2,]-(a[2,1]/a[1,1])*a[1,]
19 a[3,]=a[3,]-(a[3,1]/a[1,1])*a[1,]
20 a[4,]=a[4,]-(a[4,1]/a[1,1])*a[1,]
21 print(a)
22 a[4,]=a[4,]-2*a[2,]
23 print(a)
24 a[4,]=a[4,]-a[3,]
25 print(a)
26 k=0
27 i1=c(1:4)

```

```
28 j1=c(1:4)
29 for(i in il)
30 {for (j in j1){
31   if(a[i,j]!=0){
32     k=k+1
33     break
34   }
35 }
36 }
37 print('dimension of the matrix=')
38 print(k)
```

---

# Chapter 3

## Determinants

R code Exa 3.1 1

```
1 #Chapter 3– Determinants
2 #Determinant of a matrix
3 #Page No.108 / 3–2
4 #Prob 1
5 #3.1.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 Av<-c(3,0,4,2,3,2,0,5,-1)
15 A=matrix(Av,3,3,TRUE)
16 print(A)
17 print('calculating det(A) using cofactor expression
      along first row')
18 print('det(A)=3 X (-1 X 3-5 X 2)+4 X (2 X 5-3 X 0)')
19 print('det(A) =')
20 print(det(A))
```

---

### R code Exa 3.2 2

```
1 #Chapter 3– Determinants
2 #Determinant of a matrix
3 #Page No.108 / 3–2
4 #Prob 7
5 #3.1.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('given matrix is:')
14 Av<-c(4,3,0,6,5,2,9,7,3)
15 A=matrix(Av,3,3,TRUE)
16 print(A)
17 print('calculating det(A) using cofactor expression
      along first row')
18 print('det(A)=4 X (5 X 3–7 X 2)–3 X (6 X 3–9 X 2)')
19 print('det(A) =')
20 print(det(A))
```

---

### R code Exa 3.3 3

```
1 #Chapter 3– Determinants
2 #Determinant of a matrix
3 #Page No.108 / 3–2
4 #Prob 13
5 #3.1.13
```

```

6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 Av<-c
    (4,0,-7,3,-5,0,0,2,0,0,7,3,-6,4,-8,5,0,5,2,-3,0,0,9,-1,2)

15 A=matrix(Av,5,5,TRUE)
16 print('A=')
17 print(A)
18 P=A
19 print('since row 2 has maximum zeros , using row 2
    for cofactor expression')
20 A<-A[-(2),]
21 A<-A[,-(3)]
22 print('deleting second row and third column from A,
    we get')
23 print(A)
24 print('det(A)=-2 X')
25 print('det')
26 print(A)
27 print('for the 4X4 matrix obtained , using column 2
    for cofactor exansion')
28 print('deleting second column and row from the 4X4
    matrix')
29 A<-A[-(2),]
30 A<-A[,-(2)]
31 print(A)
32 print('det(A)=-2 X 3 X')
33 print('det')
34 print(A)
35 print('-6 X [4 X (4-3)-5 X (6-5)]')
36 print('=')
37 print(-6*det(A))

```

---

#### R code Exa 3.4 4

```
1 #Chapter 3– Determinants
2 #Property of determinants
3 #Page No.110 / 3–4
4 #Prob 37
5 #3.1.37
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 Av<-c(3,1,4,2)
14 A=matrix(Av,2,2,TRUE)
15 print('the given matrix is:')
16 print(A)
17 print('det(A)=')
18 print(det(A))
19 print('5 X A = ')
20 print(5*A)
21 print('det(5*A)=')
22 print(det(5*A))
23 print('thus, det(5A) is not equal to 5Xdet(A)')
24 print('infact, the relation between det(rA) and det(
    A) for a nxn matrix is:')
25 print('det(rA)=(r^n)*det(A)')
```

---

#### R code Exa 3.5 5

```

1 #Chapter 3– Determinants
2 #Determinant of a matrix
3 #Page No.110 / 3–4
4 #Prob 41
5 #3.1.41
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(3,1,0,2)
14 a=matrix(av,2,2,TRUE)
15 print(a)
16 print('det=')
17 print(det(a))

```

---

### R code Exa 3.6 6

```

1 #Chapter 3– Determinants
2 #Determinant of a matrix
3 #Page No.111 / 3–5
4 #Prob 7
5 #3.2.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 Av<-c(1,3,0,2,-2,-5,7,4,3,5,2,1,1,-1,2,-3)
15 A=matrix(Av,4,4,TRUE)

```

```

16 print('A=')
17 print(A)
18 print('performing row operations')
19 A[2,]=A[2,]-(A[2,1]/A[1,1])*A[1,]
20 A[3,]=A[3,]-(A[3,1]/A[1,1])*A[1,]
21 A[4,]=A[4,]-(A[4,1]/A[1,1])*A[1,]
22 print(A)
23 A[3,]=A[3,]-(A[3,2]/A[2,2])*A[2,]
24 A[4,]=A[4,]-(A[4,2]/A[2,2])*A[2,]
25 print(A)
26 A[4,]=A[4,]-(A[4,3]/A[3,3])*A[3,]
27 print(A)
28 print('det(A) is the product of diagonal entries')
29 print('det(A)=')
30 print(det(A))

```

---

### R code Exa 3.7 7

```

1 #Chapter 3- Determinants
2 #Determinant of a matrix
3 #Page No.112 / 3-6
4 #Prob 13
5 #3.2.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 av<-c(2,5,4,1,4,7,6,2,6,-2,-4,0,-6,7,7,0)
15 a=matrix(av,4,4,TRUE)
16 print('A=')
17 print(a)

```



```

18 print('performing row operations')
19 a[2,]=a[2,]-2*a[1,]
20 print(a)
21 print('using cofactor expansion about fourth column'
      )
22 a<-a[-(1),]
23 a<-a[,-(4)]
24 print('det(A)= -1 X')
25 print('det')
26 print(a)
27 print('performing row operations')
28 a[3,]=a[3,]+a[2,]
29 print(a)
30 print('using cofactor expansion about first column')
31 a<-a[-(2),]
32 a<-a[,-(1)]
33 print('det(A)= -1 X -6 X')
34 print('det')
35 print(a)
36 print('=')
37 print(6*det(a))

```

---

### R code Exa 3.8 8

```

1 #Chapter 3– Determinants
2 #Determinant of a matrix
3 #Page No.112 / 3–6
4 #Prob 19
5 #3.2.19
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))

```

```

12
13 print('the given matrix is:')
14 print('A=')
15 print(' a      b      c ')
16 print('2d+a    2e+b    2f+c ')
17 print(' g      h      i ')
18 print('B=')
19 print('a  b  c ')
20 print('d  e  f ')
21 print('g  h  i ')
22 print('given , det(B)=7')
23 print('performing row operations on A')
24 print('R2=R2-R1')
25 print('A=')
26 print('a  b  c ')
27 print('2d  2e  2f ')
28 print('g  h  i ')
29 print('factoring 2 out of row 2')
30 print('A=')
31 print('2 X')
32 print('a  b  c ')
33 print('d  e  f ')
34 print('g  h  i ')
35 print('therefore , det(A)=2 X det(B)')
36 print('=2 X 7')
37 print('= 14')

```

---

### R code Exa 3.9 9

```

1 #Chapter 3– Determinants
2 #Linear independency using determinants
3 #Page No.112 / 3–6
4 #Prob 25
5 #3.2.25
6

```

```

7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given vectors are:')
14 v1v<-c(7,-4,-6)
15 v2v<-c(-8,5,7)
16 v3v<-c(7,0,-5)
17 v1=matrix(v1v,3,1,TRUE)
18 v2=matrix(v2v,3,1,TRUE)
19 v3=matrix(v3v,3,1,TRUE)
20 print('v1=')
21 print(v1)
22 print('v2=')
23 print(v2)
24 print('v3=')
25 print(v3)
26 print('combining them as a matrix')
27 a=cbind(v1,v2,v3)
28 print('A=')
29 print(a)
30 print('if det(A) is not equal to zero , then v1 v2
      and v3 are linearly independent')
31 print('expanding about third column')
32 print('det(A)=7 X (-28+30) - 5 X (35-32)')
33 print('=')
34 print(det(a))
35 print('hence , v1 v2 and v3 are linearly independent'
      )

```

---

**R code Exa 3.10 10**

1 #Chapter 3– Determinants

```

2 #Cramers Rule
3 #Page No.115 / 3-9
4 #Prob 1
5 #3.3.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the co-efficient matrix is:')
14 av<-c(5,7,2,4)
15 a=matrix(av,2,2,TRUE)
16 print('A=')
17 print(a)
18 print('the RHS is:')
19 b=matrix(c(3,1),2,1,TRUE)
20 print(b)
21 print('applying cramers rule')
22 print('replacing first column of matrix A by b')
23 A1v<-c(3,7,1,4)
24 A1=matrix(A1v,2,2,TRUE)
25 print('A1=')
26 print(A1)
27 print('replacing second column of matrix A by b')
28 A2v<-c(5,3,2,1)
29 A2=matrix(A2v,2,2,TRUE)
30 print('A2=')
31 print(A2)
32 print('x1=det(A1)/det(A)')
33 print('=')
34 print((det(A1)/det(a)))
35 print('x2=det(A2)/det(A)')
36 print('=')
37 print((det(A2)/det(a)))

```

---

### R code Exa 3.11 11

```
1 #Chapter 3– Determinants
2 #Inverse of a matrix
3 #Page No.116 / 3–10
4 #Prob 13
5 #3.3.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 av<-c(3,5,4,1,0,1,2,1,1)
15 a=matrix(av,3,3,TRUE)
16 print('A=')
17 print(a)
18 print('the cofactors are:')
19 m1=matrix(c(0,1,1,1),2,2,TRUE)
20 C11=det(m1)
21 print('C11=')
22 print(C11)
23 m2=matrix(c(1,1,2,1),2,2,TRUE)
24 C12=-det(m2)
25 print('C12=')
26 print(C12)
27 m3=matrix(c(1,0,2,1),2,2,TRUE)
28 C13=det(m3)
29 print('C13=')
30 print(C13)
31 m3=matrix(c(5,4,1,1),2,2,TRUE)
32 C21=-det(m3)
```

```

33 print('C21=')
34 print(C21)
35 m4=matrix(c(3,4,2,1),2,2,TRUE)
36 C22=det(m4)
37 print('C22=')
38 print(C22)
39 m4=matrix(c(3,5,2,1),2,2,TRUE)
40 C23=-det(m4)
41 print('C23=')
42 print(C23)
43 m5=matrix(c(5,4,0,1),2,2,TRUE)
44 C31=det(m5)
45 print('C31=')
46 print(C31)
47 m6=matrix(c(3,4,1,1),2,2,TRUE)
48 C32=-det(m6)
49 print('C32=')
50 print(C32)
51 m7=matrix(c(3,5,1,0),2,2,TRUE)
52 C33=det(m7)
53 print('C33=')
54 print(C33)
55 c1m=rbind(C11,C12,C13)
56 c2m=rbind(C21,C22,C23)
57 c3m=rbind(C31,C32,C33)
58 B=cbind(c1m,c2m,c3m)
59 print('adj(A)=')
60 print(B)
61 print('A.adj(A)=')
62 adjA=ceiling(a %*% B)
63 print(adjA)
64 C=B/(det(a))
65 print('inv(A)=')
66 print(C)

```

---

### R code Exa 3.12 12

```
1 #Chapter 3– Determinants
2 #Applicant of determinant
3 #Page No.117 / 3–11
4 #Prob 19
5 #3.3.19
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the points forming the parrallelogram are')
14 print('(0,0),(5,2),(6,4),(11,6)')
15 print('using the vertices adjacent to origin to form
      a matrix')
16 Av<-c(5,6,2,4)
17 A=matrix(Av,2,2,TRUE)
18 print('A=')
19 print(A)
20 print('Area of parallelogram = det(A)')
21 print('=')
22 print(det(A))
```

---

# Chapter 4

## Vector Spaces

R code Exa 4.1 1

```
1 #Chapter 4– Vector Spaces
2 #Subspace of vectors
3 #Page No.123 / 4–3
4 #Prob 13c
5 #4.1.13c
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given vectors are:')
14 v1=matrix(c(1,0,-1),3,1,TRUE)
15 print('v1=')
16 print(v1)
17 v2=matrix(c(2,1,3),3,1,TRUE)
18 print('v2=')
19 print(v2)
20 v3=matrix(c(4,2,6),3,1,TRUE)
21 print('v3=')
```



```

22 print(v3)
23 w=matrix(c(3,1,2),3,1,TRUE)
24 print('w=')
25 print(w)
26 print('It is clear that w is not one of the three
      vectors in v1,v2 and v3')
27 print('The span of v1,v2 and v3 contains infinitely
      many vectors.')
28 print('To check if w is in the subspace of v1,v2 and
      v3,')
29 print('we form an augmented matrix.')
30 av<-c(1,2,4,3,0,1,2,1,-1,3,6,2)
31 a=matrix(av,3,4,TRUE)
32 print(a)
33 print('performing row operations')
34 print('R3=R3+R1')
35 a[3,]=a[3,]+a[1,]
36 print(a)
37 print('R3=R3-5xR2')
38 a[3,]=a[3,]-5*a[2,]
39 print(a)
40 print('there is no pivot in the augmented column,')
41 print('hence the vector equation is consistent and w
      is in span{v1 v2 v3}.')

```

---

## R code Exa 4.2 2

```

1 #Chapter 4- Vector Spaces
2 #Null space of a matrix
3 #Page No.125 / 4-5
4 #Prob 1
5 #4.2.1
6
7 #clear console
8 cat("\014")

```

```

9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 av<-c(3,-5,-3,6,-2,0,-8,4,1)
15 a=matrix(av,3,3,TRUE)
16 print('A=')
17 print(a)
18 print('the vector x is:')
19 x=matrix(c(1,3,-4),3,1,TRUE)
20 print('x=')
21 print(x)
22 print('To check if x is in nullspace of A')
23 print('Ax=')
24 zm=matrix(0,3,1,TRUE)
25 print('=')
26 print(zm)
27 print('hence, x is in the null space of A')

```

---

### R code Exa 4.3 3

```

1 #Chapter 4- Vector Spaces
2 #Column space of a matrix
3 #Page No.128 / 4-8
4 #Prob 13
5 #4.3.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')

```

```

14 av<-c(1,0,6,5,0,0,2,5,3,0,0,0,0,0,0)
15 a=matrix(av,3,5,TRUE)
16 p=a
17 print('A=')
18 print(a)
19 print('Reducing A to echelon form')
20 print('R2=R2/2')
21 a[2,]=a[2,]/2
22 print(a)
23 print('the pivot columns are column 1 and 2 of A')
24 print('x1=-6x3-5x4')
25 print('x2=-(5/2)x3-(3/2)x4')
26 print('x3 and x4 are free')
27 print('x3[-6 -5/2 1 0] + ')
28 print('x4[-5 -3/2 0 1] ')
29 print('the standard choice is (-6,-5/2,1,0)')
30 print('and (-5,-3/2,0,1) ')
31 print('Another choice is ')
32 print('(-12,-5,2,0) ')
33 print('and (-10,-3,0,2) ')
34 print('which avoids fractions')

```

---

#### R code Exa 4.4 4

```

1 #Chapter 4- Vector Spaces
2 #Matrix operation
3 #Page No.132 / 4-12
4 #Prob 1
5 #4.4.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))

```

```

12
13 xb=matrix(c(5,3),2,1,TRUE)
14 print(xb)
15 b1=matrix(c(3,-5),2,1,TRUE)
16 b2=matrix(c(-4,6),2,1,TRUE)
17 x=5*b1+3*b2
18 print(x)

```

---

### R code Exa 4.5 5

```

1 #Chapter 4– Vector Spaces
2 #Gaussian Elimination
3 #Page No.132 / 4–12
4 #Prob 7
5 #4.4.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('vector x=')
14 x=matrix(c(8,-9,6),3,1,TRUE)
15 print(x)
16 print('the given basis is:')
17 b1=matrix(c(1,-1,-3),3,1,TRUE)
18 b2=matrix(c(-3,4,9),3,1,TRUE)
19 b3=matrix(c(2,-2,4),3,1,TRUE)
20 print('b1=')
21 print(b1)
22 print('b2=')
23 print(b2)
24 print('b3=')
25 print(b3)

```

```

26 print('to solve the vector equation')
27 print('an augmented matrix is formed')
28 av<-c(1,-3,2,8,-1,4,-2,-9,-3,9,4,6)
29 a=matrix(av,3,4,TRUE)
30 print('A=')
31 print(a)
32 print('performing row operations')
33 a[2,]=a[2,]-(a[2,1]/a[1,1])*a[1,]
34 a[3,]=a[3,]-(a[3,1]/a[1,1])*a[1,]
35 print(a)
36 a[3,]=a[3,]/a[3,3]
37 a[1,]=a[1,]-2*a[3,]
38 print(a)
39 a[1,]=a[1,]+3*a[2,]
40 print(a)
41 print('Xb=')
42 print(a[,4])

```

---

#### R code Exa 4.6 6

```

1 #Chapter 4- Vector Spaces
2 #Matix Solution
3 #Page No.133 / 4-13
4 #Prob 13
5 #4.4.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('c1+c3=1')
14 print('c2+2c3=4')
15 print('c1+c2+c3=7')

```

```

16
17 av<-c(1,0,1,1,0,1,2,4,1,1,1,7)
18 a=matrix(av,3,4,TRUE)
19 print(a)
20 a[3,]=a[3,]-1*a[1,]
21 a[3,]=a[3,]-1*a[2,]
22 print(a)
23 a[3,]=a[3,]/-2
24 a[2,]=a[2,]-2*a[3,]
25 a[1,]=a[1,]-1*a[3,]
26 print(a)
27 ps=a[,4]
28 print('ps=')
29 print(ps)
30 m1=matrix(c(1,0,1),3,1,TRUE)
31 m2=matrix(c(0,1,1),3,1,TRUE)
32 m3=matrix(c(1,2,1),3,1,TRUE)
33 mm=matrix(c(1,4,7),3,1,TRUE)
34 print('c1 ')
35 print(m1)
36 print('c2 ')
37 print(m2)
38 print('c3 ')
39 print(m3)
40 print('=')
41 print(mm)

```

---

#### R code Exa 4.7 7

```

1 #Chapter 4– Vector Spaces
2 #Linear independence of vectors
3 #Page No.135 / 4–15
4 #Prob 27
5 #4.4.27
6

```

```

7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('to check if vectors v1 v2 and v3 are linearly
      independent')
14 v1=matrix(c(1,0,0,1),4,1,TRUE)
15 v2=matrix(c(3,1,-2,1),4,1,TRUE)
16 v3=matrix(c(0,-1,3,-1),4,1,TRUE)
17 print('v1=')
18 print(v1)
19 print('v2=')
20 print(v2)
21 print('v3=')
22 print(v3)
23 print('forming an augmented matrix')
24 av<-c(1,3,0,0,0,1,-1,0,0,-2,3,0,1,0,-1,0)
25 a=matrix(av,4,4,TRUE)
26 print('A=')
27 print(a)
28 print('performing row operations')
29 a[4,]=a[4,]+a[1,]
30 a[1,]=a[1,]-a[4,]
31 a[1,]=a[1,]/(-1)
32 print(a)
33 a[4,]=a[4,]-a[1,]
34 a[3,]=a[3,]+2*a[2,]
35 a[4,]=a[4,]+3*a[2,]
36 a[4,]=a[4,]+4*a[3,]
37 a[4,]=a[4,]-6*a[2,]
38 a[4,]=a[4,]-a[1,]
39 a[4,]=a[4,]/8
40 a[4,]=a[4,]-a[3,]
41 print(a)
42 print('since the vector equation has only the
      trivial solution')

```

```
43 print('vectors v1 v2 and v3 are linearly independent
      ')
```

---

#### R code Exa 4.8 8

```
1 #Chapter 4– Vector Spaces
2 #Span of vectors
3 #Page No.135 / 4–15
4 #Prob 31a
5 #4.4.31a
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('to check if the polynomials span R3')
14 print('placing the coordinate vectors of the
      polynomial into the columns of a matrix')
15 av<-c(1,-3,-4,1,-3,5,5,0,5,-7,-6,-1)
16 a=matrix(av,3,4,TRUE)
17 print('A=')
18 print(a)
19 print('performing row operations')
20 a[2,]=a[2,]+3*a[1,]
21 a[3,]=a[3,]-5*a[1,]
22 print(a)
23 a[3,]=a[3,]+2*a[2,]
24 print(a)
25 print('the four vectors DO NOT span R3 as there is
      no pivot in row 3')
26
27 #4.4.31b
28
```



```

29 print('to check if the polynomials span R3')
30 print('placing the coordinate vectors of the
    polynomial into the columns of a matrix')
31 av1<-c(0,1,-3,2,5,-8,4,-3,1,-2,2,0)
32 a1=matrix(av1,3,4,TRUE)
33 print('A=')
34 print(a1)
35 print('performing row operations')
36 a1[c(1,3),]=a1[c(3,1),]
37 print(a1)
38 a1[2,]=a1[2,]-5*a1[1,]
39 print(a1)
40 a1[3,]=a1[3,]-0.5*a1[2,]
41 print(a1)
42 print('the four vectors DO NOT span R3 as there is
    no pivot in row 3')

```

---

#### R code Exa 4.9 9

```

1 #Chapter 4– Vector Spaces
2 #Dimension of a vector space
3 #Page No.137 / 4–17
4 #Prob 3
5 #4.5.3
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('to find the dimension of subspace H, which is
    the set of linear combination of vectors v1 v2
    and v3')
14 v1=matrix(c(0,1,0,1),4,1,TRUE)

```

```

15 v2=matrix(c(0,-1,1,2),4,1,TRUE)
16 v3=matrix(c(2,0,-3,0),4,1,TRUE)
17 print('v1=')
18 print(v1)
19 print('v2=')
20 print(v2)
21 print('v3=')
22 print(v3)
23 print('Clearly, v1 is not equal to zero')
24 print('and v2 is not a multiple of v1 as third
      element of v1 is zero whereas that of v2 is 1.')
25 print('Also, v3 is not a linear combination of v1
      and v2 as the first element of v1 and v2 is zero
      but that of v3 is 2')
26 print('Hence, v1 v2 and v3 are linearly independent
      and dim(H)=3')

```

---

#### R code Exa 4.10 10

```

1 #Chapter 4– Vector Spaces
2 #Rank of a matrix
3 #Page No.141 / 4–21
4 #Prob 1
5 #4.6.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('to find the rank of matrix A')
14 av<-c(1,-4,9,-7,-1,2,-4,1,5,-6,10,7)
15 a=matrix(av,3,4,TRUE)
16 p=a

```

```

17 print('A=')
18 print(a)
19 print('performing row operations')
20 a[2,]=a[2,]+a[1,]
21 a[3,]=a[3,]-5*a[1,]
22 a[3,]=a[3,]+7*a[2,]
23 a[1,]=a[1,]-2*a[2,]
24 print(a)
25 print('It is clear that matrix A has 2 pivot columns
      ')
26 print('Hence, rank(A)=2')
27 print('Columns 1 and 2 are pivot columns')
28 print('Hence, basis for C(A) is:')
29 cat(p[,1], 'and', p[,2])
30 print('Basis for row space of A is:')
31 cat(a[1,], 'and', a[2,])
32 print('To find the basis of N(A), solve Ax=0')
33 print('on solving, we get the basis of N(A) as:')
34 u=matrix(c(1,2.5,1,0),4,1,TRUE)
35 v=matrix(c(-5,-3,0,1),4,1,TRUE)
36 cat(v, 'and', u)

```

---

#### R code Exa 4.11 11

```

1 #Chapter 4– Vector Spaces
2 #Change of Basis
3 #Page No.145 / 4–25
4 #Prob 1a
5 #4.7.1a
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))

```

```

12
13 b1=matrix(c(6,-2),2,1,TRUE)
14 print(b1)
15 b2=matrix(c(9,-4),2,1,TRUE)
16 print(b2)
17 pcb=cbind(b1,b2)
18 print(pcb)
19
20 #4.7.1b
21 xb=matrix(c(-3,2),2,1,TRUE)
22 print(xb)
23 xc=pcb %*% xb
24 print(xc)

```

---

#### R code Exa 4.12 12

```

1 #Chapter 4– Vector Spaces
2 #Change of Basis
3 #Page No.145 / 4–25
4 #Prob 7
5 #4.7.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 c1=matrix(c(1,-5),2,1,TRUE)
14 c2=matrix(c(-2,2),2,1,TRUE)
15 b1=matrix(c(7,5),2,1,TRUE)
16 b2=matrix(c(-3,-1),2,1,TRUE)
17 m=cbind(c1,c2,b1,b2)
18 print(m)
19 m[2,]=m[2,]+5*m[1,]

```

```

20 print(m)
21 m[2,]=m[2,]/8
22 m[2,]=m[2,]/-1
23 print(m)
24 m[1,]=m[1,]+2*m[2,]
25 print(m)
26 pcb=m[,3:4]
27 print(pcb)
28 print('inv(pcb)')
29 print(solve(pcb))

```

---

#### R code Exa 4.13 13

```

1 #Chapter 4- Vector Spaces
2 #Change of Basis
3 #Page No.146 / 4-26
4 #Prob 13
5 #4.7.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 b1c=matrix(c(1,-2,1),3,1,TRUE)
14 b2c=matrix(c(3,-5,4),3,1,TRUE)
15 b3c=matrix(c(0,2,3),3,1,TRUE)
16 print('b1c')
17 print(b1c)
18 print('b2c')
19 print(b2c)
20 print('b3c')
21 print(b3c)
22 pcb=cbind(b1c,b2c,b3c)

```

```

23 print(pcb)
24 print('coordinate vector  $[-1+2t]B$  satisfies')
25 print('pcb  $[-1+2t]B = [-1+2t]C$ ')
26 print('= $[-1 \ 2 \ 0]$ ')
27 nm=matrix(c(-1,2,0),3,1,TRUE)
28 cvm=cbind(pcb,nm)
29 print(cvm)
30 cvm[2,]=cvm[2,]+2*cvm[1,]
31 cvm[3,]=cvm[3,]-1*cvm[1,]
32 cvm[3,]=cvm[3,]-1*cvm[2,]
33 cvm[2,]=cvm[2,]-2*cvm[3,]
34 cvm[1,]=cvm[1,]-3*cvm[2,]
35 print(cvm)
36 print('= $[-1+2t]B =$  ')
37 print(cvm[,4])

```

---

#### R code Exa 4.14 14

```

1 #Chapter 4- Vector Spaces
2 #Change of Basis
3 #Page No.146 / 4-26
4 #Prob 19a
5 #4.7.19a
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 vv<-c(-2,-8,-7,2,5,2,3,2,6)
14 pv<-c(1,2,-1,-3,-5,0,4,6,1)
15 v=matrix(vv,3,3,TRUE)
16 p=matrix(pv,3,3,TRUE)
17 print(v)

```

```

18 print(p)
19 vp=v %*% p
20 print(vp)
21 u1=vp[,1]
22 u2=vp[,2]
23 u3=vp[,3]
24 print('u1')
25 print(u1)
26 print('u2')
27 print(u2)
28 print('u3')
29 print(u3)

```

---

#### R code Exa 4.15 15

```

1 #Chapter 4– Vector Spaces
2 #Linear Independence
3 #Page No.148 / 4–28
4 #Prob 7
5 #4.8.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(1^0,2^0,(-2)^0,1^1,2^1,(-2)^1,1^2,2^2,(-2)^2)
14 a=matrix(av,3,3,TRUE)
15 print(a)
16 a[2,]=a[2,]-a[1,]
17 print(a)
18 a[3,]=a[3,]-a[1,]
19 print(a)
20 a[3,]=a[3,]-3*a[2,]

```

```
21 print(a)
22 print('Hence the set of signals is linearly
    independent in S')
```

---

#### R code Exa 4.16 16

```
1 #Chapter 4– Vector Spaces
2 #Change of Basis
3 #Page No.149 / 4–29
4 #Prob 13
5 #4.8.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 r1=(1+(1/3))/2
14 r2=(1-(1/3))/2
15 print('r=')
16 print(r1)
17 print('or ')
18 print(r2)
19 cat('Two solutions of the difference equation are',
    r1, '^k and ', r2, '^k')
```

---

#### R code Exa 4.17 17

```
1 #Chapter 4– Vector Spaces
2 #Change of Basis
3 #Page No.149 / 4–29
4 #Prob 19
```



```

5 #4.8.19
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 r1=(-4+sqrt(16-4))/2
14 r2=(-4-sqrt(16-4))/2
15 print('r=')
16 print(r1)
17 print('or ')
18 print(r2)
19 cat('Two solutions of the difference equation are',
      r1, '^k and ', r2, '^k')
20 print('')
21 cat('The general solution has the form c1 ', r1, '^k+c2
      ', r2, '^k')

```

---

#### R code Exa 4.18 18

```

1 #Chapter 4– Vector Spaces
2 #Change of Basis
3 #Page No.150 / 4–30
4 #Prob 19b
5 #4.8.19b
6 #(this solution is for Section 4.7)
7
8 #clear console
9 cat("\014")
10
11 #clear variables
12 rm(list=ls(all=TRUE))
13

```

```

14 vv<-c(-2,-8,-7,2,5,2,3,2,6)
15 v=matrix(vv,3,3,TRUE)
16 print(v)
17 pv<-c(5,8,5,-3,-5,-3,-2,-2,-1)
18 pinv=matrix(pv,3,3,TRUE)
19 print(pinv)
20 w= v %*% pinv
21 print(w)
22 w1=w[,1]
23 w2=w[,2]
24 w3=w[,3]
25 print(w1)
26 print(w2)
27 print(w3)

```

---

#### R code Exa 4.19 19

```

1 #Chapter 4– Vector Spaces
2 #Change of Basis
3 #Page No.152 / 4–32
4 #Prob 1
5 #4.9.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 p=matrix(c(.7,.6,.3,.4),2,2,TRUE)
14 x1m=matrix(c(1,0),2,1,TRUE)
15 x2m=matrix(c(.7,.3),2,1,TRUE)
16 print('x1m')
17 print(x1m)
18 print('x2m')

```

```

19 print(x2m)
20 x1=p %*% x1m
21 x2=p %*% x2m
22 print('x1')
23 print(x1)
24 print('x2')
25 print(x2)

```

---

#### R code Exa 4.20 20

```

1 #Chapter 4– Vector Spaces
2 #Change of Basis
3 #Page No.153 / 4–33
4 #Prob 7
5 #4.9.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 pv<-c(.7,.1,.1,.2,.8,.2,.1,.1,.7)
14 p=matrix(pv,3,3,TRUE)
15 print(p)
16 PI=p-diag(3)
17 print(PI)
18 zm=matrix(0,3,1,TRUE)
19 PIx=cbind(PI,zm)
20 print(PIx)
21 PIx[,c(1,3)]=PIx[,c(3,1)]
22 print(PIx)
23 PIx=PIx*10
24 print(PIx)
25 PIx[2,]=PIx[2,]-2*PIx[1,]

```

```

26 PIx[3,]=PIx[3,]+3*PIx[1,]
27 PIx[2,]=PIx[2,]/(-4)
28 PIx[3,]=PIx[3,]-4*PIx[2,]
29 PIx[1,]=PIx[1,]-1*PIx[2,]
30 print(round(PIx))
31 print('x1=x3')
32 print('x2=2x3')
33 print('x3 is free')
34 print('[x1 x2 x3]=x3[1 2 1]')
35 print('the entries in [1 2 1] sum to 4')
36 q=(1/4)*matrix(c(1,2,1),3,1,TRUE)
37 print(q)

```

---

#### R code Exa 4.21 21

```

1 #Chapter 4– Vector Spaces
2 #Change of Basis
3 #Page No.154 / 4–34
4 #Prob 13a
5 #4.9.13a
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 P=matrix(c(.95,.45,.05,.55),2,2,TRUE)
14 print(P)
15 PI=P-diag(2)
16 print(PI)
17 zm=matrix(0,2,1,TRUE)
18 PIm=cbind(PI,zm)
19 print(PIm)
20 PIm[2,]=round(PIm[2,]+PIm[1,])

```

```
21 PIm[1,]=PIm[1,]*20
22 print(PIm)
23 print('x1=9x2')
24 print('x2 is free')
25 print('A basis for Nul(P-I) is [9 1]')
26 print('the steady-state vector is q=[.9 .1]')
```

---

# Chapter 5

## Eigenvalues And Eigenvectors

R code Exa 5.1 1

```
1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Eigenvalue of a matrix
3 #Page No.157 / 5–1
4 #Prob 1
5 #5.1.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('to check if 2 is an eigenvalue of matrix A')
14 a=matrix(c(3,2,3,8),2,2,TRUE)
15 print('A=')
16 print(a)
17 print('A-2I=')
18 b=a-2*diag(2)
19 print(b)
20 print('The columns of A are clearly independent,')
21 print('hence (A-2I)x=0 has a non trivial solution')
```

and 2 is an eigenvalue of matrix A')

---

### R code Exa 5.2 2

```
1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Eigenvalue of a matrix
3 #Page No.158 / 5–2
4 #Prob 7
5 #5.1.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('To check if 4 is an eigenvalue of matrix A')
14 av<-c(3,0,-1,2,3,1,-3,4,5)
15 a=matrix(av,3,3,TRUE)
16 print('A=')
17 print(a)
18 print('Therefore ')
19 print('A-4I=')
20 print(a-4*diag(3))
21 b=a-4*diag(3)
22 print('to check the invertibility of A-4I, form an
      augmented matrix')
23 zm=matrix(0,3,1,TRUE)
24 c=cbind(b,zm)
25 print(c)
26 print('performing row operations')
27 c[2,]=c[2,]+2*c[1,]
28 c[3,]=c[3,]-3*c[1,]
29 print(c)
30 c[3,]=c[3,]+4*c[2,]
```

```

31 print(c)
32 print('We can see that there exists a non trivial
      solution.')
33 print('Hence, 4 is an eigenvalue of A.')
34 print('For the eigenvector,  $-x_1-x_3=0$  and  $-x_2-x_3=0$ ')
35 print('If  $x_3=1$ ,')
36 x=matrix(c(-1,-1,1),3,1,TRUE)
37 print('x=')
38 print(x)

```

---

### R code Exa 5.3 3

```

1 #Chapter 5- Eigenvalues and Eigenvectors
2 #Eigenvectors
3 #Page No.158 / 5-2
4 #Prob 13
5 #5.1.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('To find a basis for the eigenspace')
14 print('Matrix A=')
15 av<-c(4,0,1,-2,1,0,-2,0,1)
16 a=matrix(av,3,3,TRUE)
17 print(a)
18 print('for lambda=1')
19 print('A-I=')
20 b=a-diag(3)
21 print(b)
22 print('solving  $(A-I)x=0$ , we get')
23 print('  $-2*x_1=0$  and  $3*x_1+x_3=0$  ')

```



```

24 print('therefore , x1=x3=0')
25 print('which leaves x2 as a free variable')
26 print('Hence a basis for the eigen space is:')
27 es=matrix(c(0,1,0),3,1,TRUE)
28 print(es)
29 print('for lambda=2')
30 print('A-2I=')
31 b=a-2*diag(3)
32 print(b)
33 print('performing row operations on the augmented
      matrix')
34 zm=matrix(0,3,1,TRUE)
35 c=cbind(b,zm)
36 print(c)
37 c[2,]=c[2,]+c[1,]
38 c[3,]=c[3,]+c[1,]
39 print(c)
40 c[1,]=c[1,]/c[2,2]
41 print(c)
42 print('We can see that x3 is a free variable')
43 print('x2=x3 and x1=-.05*x3')
44 print('Hence, a basis for the eigenspace is:')
45 es1=matrix(c(1,2,2),3,1,TRUE)
46 print(es1)
47 print('for lambda=3')
48 print('A-3I=')
49 b=a-3*diag(3)
50 print(b)
51 print('performing row operations on the augmented
      matrix')
52 c=cbind(b,zm)
53 print(c)
54 c[2,]=c[2,]+2*c[1,]
55 c[3,]=c[3,]+2*c[1,]
56 print(c)
57 c[2,]=c[2,]/2
58 c[2,]=c[2,]/(-1)
59 print(c)

```

```

60 print('Again x3 is a free variable')
61 print('x1=-x3 and x2=x3')
62 print('Hence, a basis for the eigenspace is:')
63 es2=matrix(c(-1,1,1),3,1,TRUE)
64 print(es2)

```

---

#### R code Exa 5.4 4

```

1 #Chapter 5- Eigenvalues and Eigenvectors
2 #Property of non-invertible matrices
3 #Page No.159 / 5-3
4 #Prob 19
5 #5.1.19
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('The given matrix is:')
14 av<-c(1,2,3,1,2,3,1,2,3)
15 a=matrix(av,3,3,TRUE)
16 print('A=')
17 print(a)
18 print('A is not invertible because its columns are
      linearly dependent.')
19 print('Hence, 0 is an eigenvalue of matrix A.')

```

---

#### R code Exa 5.5 5

```

1 #Chapter 5- Eigenvalues and Eigenvectors
2 #Eigenvalue of a matrices

```

```

3 #Page No.162 / 5-6
4 #Prob 1
5 #5.2.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('To find the eigenvalue of matrix A')
14 print('A=')
15 a=matrix(c(2,7,7,2),2,2,TRUE)
16 print(a)
17 print('Eigen values of A are:')
18 eig<-eigen(a)
19 eigv<-eig$values
20 print(eigv)

```

---

#### R code Exa 5.6 6

```

1 #Chapter 5- Eigenvalues and Eigenvectors
2 #Complex eigenvalues
3 #Page No.162 / 5-6
4 #Prob 7
5 #5.2.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('To find the eigenvalues of matrix A.')
14 print('A=')

```

```

15 a=matrix(c(5,3,-4,4),2,2,TRUE)
16 print(a)
17 print('Eigen values of A are:')
18 eig<-eigen(a)
19 eigv<-eig$values
20 print(eigv)
21 print('Hence, A has no real eigenvalues.')
```

---

#### R code Exa 5.7 7

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Eigenvalues of a matrix
3 #Page No.162 / 5–6
4 #Prob 13
5 #5.2.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('To find the eigenvalues of the matrix A')
14 print('A=')
15 av<-c(6,-2,0,-2,9,0,5,8,3)
16 a=matrix(av,3,3,TRUE)
17 print(a)
18 print('Eigenvalues of A are:')
19 eig<-eigen(a)
20 eigv<-eig$values
21 print(eigv)
```

---

#### R code Exa 5.8 8

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Eigenvectors
3 #Page No.163 / 5–7
4 #Prob 25a
5 #5.2.25 a
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('Matrix A=')
14 a=matrix(c(.6,.3,.4,.7),2,2,TRUE)
15 print(a)
16 print('Eigenvector v1=')
17 v1=matrix(c(3/7,4/7),2,1,TRUE)
18 print(v1)
19 print('vector Xo=')
20 Xo=matrix(c(.5,.5),2,1,TRUE)
21 print(Xo)
22 print('Eigenvalues of A are:')
23 eig<-eigen(a)
24 c<-eig$values
25 print(c)
26 print('To verify if v1 is an eigenvector of A:')
27 print('A*v1=')
28 print(a %*% v1)
29 print('=')
30 print('1*v1')
31 print('Hence v1 is an eigenvector of A corresponding
      to eigenvalue 1.')
32 print('for lambda=.3')
33 print('A-.3I=')
34 b=a-.3*diag(2)
35 print(b)
36 print('performing row operations on the augmented
      matrix')

```

```

37 zm=matrix(0,2,1,TRUE)
38 c=cbind(b,zm)
39 print(c)
40 c[2,]=c[2,]-(c[2,1]/c[1,1])*c[1,]
41 c[2,2]=round(c[2,2])
42 print(c)
43 print('hence , x1+x2=0')
44 print('Eigenvector corresponding to eigenvalue .3 is
      :')
45 ev=matrix(c(-1,1),2,1,TRUE)
46 print(ev)
47
48 #5.2.25b
49 x0=matrix(c((1/2),(1/2)),2,1,TRUE)
50 v1=matrix(c((3/7),(4/7)),2,1,TRUE)
51 md=x0-v1
52 print(md)
53 print('=')
54 print('(1/14)v2')

```

---

### R code Exa 5.9 9

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Diagonalization of a matrix
3 #Page No.166 / 5–10
4 #Prob 1
5 #5.3.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('The given eigenvector matrix is:')

```

```

14 p=matrix(c(5,7,2,3),2,2,TRUE)
15 print('P=')
16 print(p)
17 print('The diagonal matrix is:')
18 d=matrix(c(2,0,0,1),2,2,TRUE)
19 print('D=')
20 print(d)
21 print('Therefore , matrix A=PD(p^-1)')
22 s=solve(p)
23 print(p %*% d %*% s)
24 print('Hence , A^4=P(D^4)(P^-1)')
25 print(p%*%(d^4)%*%s)

```

---

#### R code Exa 5.10 10

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Diagonalization of a matrix
3 #Page No.166 / 5–10
4 #Prob 7
5 #5.3.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('the given matrix is:')
14 a=matrix(c(1,0,6,-1),2,2,TRUE)
15 print('A=')
16 print(a)
17 print('Since A is triangular , eigenvalues are the
      diagonal entries.')
18 cat('Eigenvalues are: ',a[2,2],a[1,1])
19 print('for lambda=1')

```

```

20 print('A-I=')
21 b=a-diag(2)
22 print(b)
23 print('Hence, x1=(1/3)x2 with x2 as free variable.')
24 print('Eigenvector corresponding to lambda=1 is:')
25 u1=matrix(c(1,3),2,1,TRUE)
26 print(u1)
27 print('for lambda=-1')
28 print('A-(-1)I=')
29 b=a+diag(2)
30 print(b)
31 print('Hence, x1=0 with x2 as free variable.')
32 print('Eigenvector corresponding to lambda=-1 is:')
33 u2=matrix(c(0,1),2,1,TRUE)
34 print(u2)
35 print('Thus, matrix P=')
36 um=cbind(u1,u2)
37 print(um)
38 print('and matrix D=')
39 Dm=matrix(c(1,0,0,-1),2,2,TRUE)
40 print(Dm)

```

---

#### R code Exa 5.11 11

```

1 #Chapter 5- Eigenvalues and Eigenvectors
2 #Diagonalization of a matrix
3 #Page No.167 / 5-11
4 #Prob 13
5 #5.3.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))

```



```

12
13 print('Given matrix A=')
14 av<-c(2,2,-1,1,3,-1,-1,-2,2)
15 a=matrix(av,3,3,TRUE)
16 print(a)
17 print('Given its eigen values are 5 and 1')
18 print('for lambda=5')
19 print('A-5I=')
20 b=a-5*diag(3)
21 print(b)
22 print('performing row operations')
23 zm=matrix(0,3,1,TRUE)
24 cm=cbind(b,zm)
25 print(cm)
26 cm[c(1,2),]=cm[c(2,1),]
27 print(cm)
28 cm[2,]=cm[2,]+3*cm[1,]
29 cm[3,]=cm[3,]+cm[1,]
30 print(cm)
31 cm[3,]=cm[3,]-cm[2,]
32 print(c)
33 cm[2,]=cm[2,]/cm[2,2]
34 cm[1,]=cm[1,]+2*cm[2,]
35 print(cm)
36 print('With x3 as free variable , x1=-x3 and x2=-x3')
37 print('Hence, for lambda=5 eigenvector is:')
38 u1=matrix(c(-1,-1,1),3,1,TRUE)
39 print(u1)
40 print('for lambda=1')
41 print('A-I=')
42 b=a-diag(3)
43 print(b)
44 print('performing row operations')
45 cn=cbind(b,zm)
46 print(cn)
47 cn[2,]=cn[2,]-cn[1,]
48 cn[3,]=cn[3,]+cn[1,]
49 print(cn)

```

```

50 print('With x2 and x3 as free variables , eigen
    vectors corresponding to lambda=1 are')
51 u2=matrix(c(-2,1,0),3,1,TRUE)
52 u3=matrix(c(1,0,1),3,1,TRUE)
53 print(u2)
54 print(u3)
55 print('Hence , matrix P=')
56 pm=cbind(u1,u2,u3)
57 print(pm)
58 print('and matrix D=')
59 Dm=matrix(c(5,0,0,0,1,0,0,0,1),3,3,TRUE)
60 print(Dm)

```

---

#### R code Exa 5.12 12

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Eigen values of matrix
3 #Page No.168 / 5–12
4 #Prob 19
5 #5.3.19
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(5,-3,0,9,0,3,1,-2,0,0,2,0,0,0,0,2)
14 a=matrix(av,4,4,TRUE)
15 print(a)
16 eig=eigen(a)
17 eigv<-eig$values
18 print('eigen values are:')
19 print(eigv)
20 print('for lambda=2')

```

```

21 print(' (A-2I)x=0')
22 zm=matrix(0,4,1,TRUE)
23 c1=a-2*diag(4)
24 m1=cbind(c1,zm)
25 print(m1)
26 m1[1,]=m1[1,]+3*m1[2,]
27 print(m1)
28 m1[1,]=m1[1,]/3
29 print(m1)
30 print('x1=x3-x4')
31 print('x2=-x3+2x4')
32 print('x3 and x4 are free')
33 v1=matrix(c(-1,-1,1,0),4,1,TRUE)
34 v2=matrix(c(-1,2,0,1),4,1,TRUE)
35 print('v1')
36 print(v1)
37 print('v2')
38 print(v2)
39 print('for lambda=3')
40 print(' (A-3I)x=0')
41 c2=a-3*diag(4)
42 m2=cbind(c2,zm)
43 print(m2)
44 m2[1,]=m2[1,]/2
45 m2[3,]=m2[3,]+m2[2,]
46 m2[3,]=m2[3,]/(-2)
47 m2[4,]=m2[4,]+m2[3,]
48 m2[2,]=m2[2,]+2*m2[3,]
49 m2[1,]=m2[1,]+(-9/2)*m2[3,]
50 print(m2)
51 print('x1=(3/2)x2')
52 print('x2 is free')
53 print('x3 and x4 = 0')
54 print('choosing x2=2 produces eigen vector v3
      =(3,2,0,0)')
55 print('for lambda=5')
56 print(' (A-5I)x=0')
57 c3=a-5*diag(4)

```

```

58 m3=cbind(c3,zm)
59 print(m3)
60 m3[1,]=m3[1,]/(-3)
61 m3[2,]=m3[2,]+2*m3[1,]
62 m3[3,]=m3[3,]+3*m3[2,]
63 m3[3,]=ceiling(m3[2,]/(-24))
64 m3[4,]=m3[4,]+3*m3[3,]
65 m3[2,]=m3[2,]+8*m3[3,]
66 m3[1,]=m3[1,]+3*m3[3,]
67 print(m3)
68 print('x1 is free')
69 print('x2=0')
70 print('x3=0')
71 print('x4=0')
72 print('a basis vector for the eigenspace is v4
      =(1,0,0,0)')
73 Pmv<-c(-1,-1,3,1,-1,2,2,0,1,0,0,0,0,1,0,0)
74 Pm=matrix(Pmv,4,4,TRUE)
75 print(Pm)
76 Dmv<-c(2,0,0,0,0,2,0,0,0,0,3,0,0,0,0,5)
77 Dm=matrix(Dmv,4,4,TRUE)
78 print(Dm)

```

---

### R code Exa 5.13 13

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Page No.170 / 5–14
3 #Prob 29
4 #5.3.29
5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))

```

```

11
12 p1=matrix(c(1,1,-2,-1),2,2,TRUE)
13 d1=matrix(c(3,0,0,5),2,2,TRUE)
14 print('p1=')
15 print(p1)
16 print('d1=')
17 print(d1)
18 p2=matrix(c(-3,1,6,-1),2,2,TRUE)
19 print('p2=')
20 print(p2)
21 A=p1*d1*solve(p1)
22 A1=p2*d1*solve(p2)
23 print('P1D1P1-1')
24 print(A)
25 print('P2D1P2-1')
26 print(A1)
27 print(A==A1)
28 print('Hence , A=P1D1P1-1=P2D1P2-1')

```

---

#### R code Exa 5.14 14

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Transform of matrix
3 #Page No.172 / 5–16
4 #Prob 1
5 #5.4.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 b1=matrix(c(3,-5),1,2,TRUE)
14 b2=matrix(c(-1,6),1,2,TRUE)

```

```

15 b3=matrix(c(0,4),1,2,TRUE)
16 b1t=t(b1)
17 b2t=t(b2)
18 b3t=t(b3)
19 cm=cbind(b1t,b2t,b3t)
20 print(cm)

```

---

### R code Exa 5.15 15

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Eigenvalues
3 #Page No.173 / 5–17
4 #Prob 13
5 #5.4.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 a=matrix(c(0,1,-3,4),2,2,TRUE)
14 eig=eigen(a)
15 eigv<-eig$values
16 print('the eigen values are')
17 print(eigv)
18 print('for lambda=1')
19 m1=a-diag(2)
20 print(m1)
21 print('as a basis vector, take u1=[1 1]')
22 print('for lambda=3')
23 m2=a-3*diag(2)
24 print(m2)
25 print('as a basis vector, take u1=[1 3]')
26 #the matrix for lambda=3 has a calculation error in

```

**R code Exa 5.16 16**

```
1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Eigenvalues
3 #Page No.175 / 5–19
4 #Prob 31
5 #5.4.31
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('Given matrix A=')
14 av<-c(-7,-48,-16,1,14,6,-3,-45,-19)
15 a=matrix(av,3,3,TRUE)
16 print(a)
17 print('and matrix P=')
18 pv<-c(-3,-2,3,1,1,-1,-3,-3,0)
19 p=matrix(pv,3,3,TRUE)
20 print(p)
21 print('Hence, matrix D=')
22 s=solve(p)
23 Dv=round(s %*% a %*% p)
24 print(Dv)
```

---

**R code Exa 5.17 17**

```
1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Complex Eigenvectors
```

```

3 #Page No.176 / 5-20
4 #Prob 1
5 #5.5.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('Matrix A=')
14 a=matrix(c(1,-2,1,3),2,2,TRUE)
15 print(a)
16 print('Eigen values of A are')
17 eig=eigen(a)
18 eigv<-eig$values
19 print(eigv)
20 print('for lambda=2+i')
21 i=complex(real=2,imaginary=1)
22 print('A-(2+i)I=')
23 b=a-(i)*diag(2)
24 print(b)
25 print('With x2 as free variable , x1=-(1-i)x2')
26 print('Hence, eigenvector corresponding to lambda=2+
      i is:')
27 i1=complex(real=-1,imaginary=1)
28 m1=matrix(c(i1,1),2,1,TRUE)
29 print(m1)
30 i2=complex(real=-1,imaginary=-1)
31 print('for lambda=2-i, eigenvector is:')
32 m2=matrix(c(i2,1),2,1,TRUE)
33 print(m2)

```

---

**R code Exa 5.18** 18



```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Scale factor of transformation
3 #Page No.176 / 5–20
4 #Prob 7
5 #5.5.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('Matrix A=')
14 a=matrix(c(sqrt(3),-1,1,sqrt(3)),2,2,TRUE)
15 print(a)
16 print('Eigenvalues of A are:')
17 eig=eigen(a)
18 eigv<-eig$values
19 print(eigv)
20 print('The scale factor associated with the
      transformation x to Ax is:')
21 print(abs(eigv[1]))

```

---

#### R code Exa 5.19 19

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Page No.177 / 5–21
3 #Prob 13
4 #5.5.13
5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))

```

```

11
12 i=complex(real=-1,imaginary=-1)
13 v=matrix(c(i,1),2,1,TRUE)
14 a=matrix(c(1,-2,1,3),2,2,TRUE)
15 print(v)
16 rev=Re(v)
17 print(rev)
18 imv=Im(v)
19 print(imv)
20 pm=cbind(rev,imv)
21 print(pm)
22 cm=solve(pm) %*% a %*% pm
23 print(cm)

```

---

#### R code Exa 5.20 20

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Page No.177 / 5–21
3 #Prob 19
4 #5.5.19
5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))
11
12 a=matrix(c(1.52,-0.7,0.56,0.4),2,2,TRUE)
13 print(a)
14 lambdap1=(1.92)/2
15 lambdap2=sqrt(abs(-0.3136))/2
16 i=complex(real=0,imaginary=lambdap2)
17 print('lambda = ')
18 cat(lambdap1+i)
19 print('or ')

```

```

20 cat(lambdap1-i)
21 print('if x2=2')
22 iv=complex(real=2,imaginary=-1)
23 vm=matrix(c(iv,2),2,1,TRUE)
24 print('v=')
25 print(vm)
26 revm=Re(vm)
27 imvm=Im(vm)
28 print(revm)
29 print(imvm)
30 p=cbind(revm,imvm)
31 print(p)
32 fm=solve(p) %*% a %*% p
33 print(fm)

```

---

#### R code Exa 5.21 21

```

1 #Chapter 5- Eigenvalues and Eigenvectors
2 #Page No.179 / 5-23
3 #Prob 1
4 #5.6.1
5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))
11
12 v1=matrix(c(1,1),2,1,TRUE)
13 v2=matrix(c(-1,1),2,1,TRUE)
14 cat('v1=',v1)
15 print(' ')
16 cat('v2=',v2)
17 x0=matrix(c(9,1),2,1,TRUE)
18 nm=cbind(v1,v2,x0)

```

```

19 print(nm)
20 nm[1,]=nm[1,]+nm[2,]
21 nm[1,]=nm[1,]/2
22 nm[2,]=nm[2,]-nm[1,]
23 print(nm)
24 nv1=matrix(c(15,15),2,1,TRUE)
25 nv2=matrix(c((-4/3),(4/3)),2,1,TRUE)
26 nvv=nv1-nv2
27 print(nvv)

```

---

## R code Exa 5.22 22

```

1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Page No.181 / 5–25
3 #Prob 13
4 #5.6.13
5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))
11
12 a=matrix(c(.8,.3,-.4,1.5),2,2,TRUE)
13 print(a)
14 print('eigen values are')
15 eig=eigen(a)
16 print(eig$values)
17 zm=matrix(0,2,1,TRUE)
18 c1=a-1.2*diag(2)
19 m1=cbind(c1,zm)
20 print(m1)
21 m1[2,]=m1[2,]-m1[1,]
22 m1[2,]=round(m1[2,])
23 m1[1,]=m1[1,]/(-4)

```

24 `print(m1)`

---

#### R code Exa 5.23 23

```
1 #Chapter 5– Eigenvalues and Eigenvectors
2 #Page No.183 / 5–27
3 #Prob 1
4 #5.7.1
5
6 #clear console
7 cat("\014")
8
9 av<-c(-3,-1,-6,1,1,1)
10 a=matrix(av,2,3,TRUE)
11 p=a
12 print(a)
13 a[c(1,2),]<-a[c(2,1),]
14 print(a)
15 a[2,]=a[2,]+3*a[1,]
16 a[2,]=a[2,]/2
17 a[1,]=a[1,]-a[2,]
18 print(a)
19 print('c1=')
20 print(a[1,3])
21 print('c2=')
22 print(a[2,3])
23 print('x(t)=')
24 cat('[',a[1,3]*p[,1],']*e^4t')
25 cat(' - [',a[2,3]*p[,2],']*e^2t')
```

---

#### R code Exa 5.24 24

```
1 #Chapter 5– Eigenvalues and Eigenvectors
```

```

2 #Page No.184 / 5-28
3 #Prob 19
4 #5.7.19
5
6 #clear console
7 cat("\014")
8
9 a=matrix(c(-2,(3/4),1,-1),2,2,TRUE)
10 print(a)
11 eig=eigen(a)
12 print('eigen values are')
13 print(eig$values)
14 v1=matrix(c(1,2),2,1,TRUE)
15 v2=matrix(c(-3,2),2,1,TRUE)
16 print('v1=')
17 print(v1)
18 print('v2=')
19 print(v2)
20 print('general solution x(t) =')
21 cat('c1*[',v1,']e^-.5t+c2*[',v2,']e^-2.5t')
22 print('')
23 ma1=cbind(v1,v2)
24 print(ma1)
25 mb=matrix(c(4,4),2,1,TRUE)
26 mc=solve(ma1) %*% mb
27 print(mc)
28 print('c1=')
29 print(mc[1])
30 print('c2=')
31 print(mc[2])
32 x1=mc[1]*v1
33 x2=mc[2]*v2
34 print('x(t)=')
35 cat('['',x1,']*e^-.5t')
36 cat(' -['',x2,']*e^-2.5t')

```

---

## Chapter 6

# Orthogonality And Least Squares

R code Exa 6.1 1

```
1 #Chapter 6– Orthogonality and Least Squares
2 #Dot product of vectors
3 #Page No.193 / 6–1
4 #Prob 1
5 #6.1.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('Vectors u and v are:')
14 u=matrix(c(-1,2),2,1,TRUE)
15 v=matrix(c(4,6),2,1,TRUE)
16 print(u)
17 print(v)
18 u2=u * u
19 vu=v * u
```

```

20 print('u . u')
21 print(u2[1]+u2[2])
22 print('v . u')
23 print(vu[1]+vu[2])
24 print('Projection of v on u=(u.v)/(v.v)')
25 a=t(u) %*% v
26 b=t(u) %*% u
27 p=a/b
28 print('=')
29 print(p)

```

---

### R code Exa 6.2 2

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Norm of a vector
3 #Page No.193 / 6–1
4 #Prob 7
5 #6.1.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('w=')
14 w=matrix(c(3,-1,-5),3,1,TRUE)
15 print(w)
16 wv=w[1]^2+w[2]^2+w[3]^2
17 print('||w||=sqrt(9+1+25)')
18 print(sqrt(wv))

```

---

### R code Exa 6.3 3



```

1 #Chapter 6– Orthogonality and Least Squares
2 #Distance between two points
3 #Page No.193 / 6–1
4 #Prob 13
5 #6.1.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('Vector x and y are:')
14 x=matrix(c(10,-3),2,1,TRUE)
15 y=matrix(c(-1,-5),2,1,TRUE)
16 print(x)
17 print(y)
18 xyv=((x[1]-y[1])^2+(x[2]-y[2])^2)
19 print('||x-y||=sqrt(125)')
20 print('=')
21 print(sqrt(xyv))

```

---

#### R code Exa 6.4 4

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Orthogonality of vectors
3 #Page No.195 / 6–3
4 #Prob 1
5 #6.2.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))

```

```

12
13 print('To verify if u v and w are orthogonal')
14 u=matrix(c(-1,4,-3),3,1,TRUE)
15 v=matrix(c(5,2,1),3,1,TRUE)
16 w=matrix(c(3,-4,-7),3,1,TRUE)
17 print('u')
18 print(u)
19 print('v')
20 print(v)
21 print('w')
22 print(w)
23 print('u.v=')
24 print(t(v) %*% u)
25 print('u.w=')
26 print(t(u) %*% w)
27 print('Since u.w is not equal to zero , the set {u v
      w} is not orthogonal.')

```

---

### R code Exa 6.5 5

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Orthogonal basis
3 #Page No.195 / 6–3
4 #Prob 7
5 #6.2.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('vectors u1 u2 and x are:')
14 u1=matrix(c(2,-3),2,1,TRUE)
15 u2=matrix(c(6,4),2,1,TRUE)

```

```

16 x=matrix(c(9,-7),2,1,TRUE)
17 print('x')
18 print(x)
19 print('u1')
20 print(u1)
21 print('u2')
22 print(u2)
23 print('u1.u2=')
24 print(t(u1) %*% u2)
25 print('u1.u2=0, {u1 u2} is an orthogonal set')
26 print('Hence {u1 u2} forms a basis of R2')
27 print('x can be written as: x=a*u1+b*u2')
28 print('where a=(x.u1)/(u1.u1)')
29 u1p1=(x * u1)
30 u1p2=(u1 * u1)
31 u1m=(u1p1[1]+u1p1[2])/(u1p2[1]+u1p2[2])
32 u2p1=(x * u2)
33 u2p2=(u2 * u2)
34 u2m=(u2p1[1]+u2p1[2])/(u2p2[1]+u2p2[2])
35 x1=(u1m * u1)
36 x2=(u2m * u2)
37 x=x1+x2
38 print('x1=')
39 print(x1)
40 print('x2=')
41 print(x2)
42 print('x=')
43 print(x)

```

---

## R code Exa 6.6 6

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Projection of vectors
3 #Page No.195 / 6–3
4 #Prob 13

```

```

5 #6.2.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('Vectors y and u are:')
14 y=matrix(c(2,3),2,1,TRUE)
15 u=matrix(c(4,-7),2,1,TRUE)
16 print('u')
17 print(u)
18 print('y')
19 print(y)
20 print('The orthogonal projection of y on u=((y.u)/(u
    .u))*u')
21 a=t(y) %*% u
22 b=t(u) %*% u
23 c1=(a/b)
24 c= c1[1] * u
25 print('=')
26 print(c)
27 print('The component of y orthogonal to u is:')
28 print(y-c)
29 print('Thus y=')
30 print(c+(y-c))

```

---

## R code Exa 6.7 7

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Orthonormal vectors
3 #Page No.196 / 6–4
4 #Prob 19
5 #6.2.19

```

```

6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('given vectors u and v are:')
14 u=matrix(c(-.6,.8),2,1,TRUE)
15 v=matrix(c(.8,.6),2,1,TRUE)
16 print('u')
17 print(u)
18 print('v')
19 print(v)
20 uvv=u*v
21 uvv=uvv[1]+uvv[2]
22 print('u.v=')
23 print(uvv)
24 uuv=u*u
25 uuv=uuv[1]+uuv[2]
26 print('u.u=')
27 print(uuv)
28 vvv=v*v
29 vvv=vvv[1]+vvv[2]
30 print('v.v=')
31 print(vvv)
32 print('Hence, {u v} is an orthogonal set.')
33 print('||u||=1 and ||v||=1')
34 print('Thus, {u v} is an orthonormal set')

```

---

### R code Exa 6.8 8

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Orthogonal projection
3 #Page No.198 / 6–6

```

```

4 #Prob 1
5 #6.3.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('Given vectors are:')
14 u1=matrix(c(0,1,-4,-1),4,1,TRUE)
15 print('u1')
16 print(u1)
17 u2=matrix(c(3,5,1,1),4,1,TRUE)
18 print('u2')
19 print(u2)
20 u3=matrix(c(1,0,1,-4),4,1,TRUE)
21 print('u3')
22 print(u3)
23 u4=matrix(c(5,-3,-1,1),4,1,TRUE)
24 print('u4')
25 print(u4)
26 x=matrix(c(10,-8,2,0),4,1,TRUE)
27 print('x')
28 print(x)
29 print('The vector in span{u4}=((x.u4)/(u4.u4))*u4')
30 a1=t(x) %*% u4
31 a2=t(u4) %*% u4
32 av=(a1/a2)
33 print(av[1]*u4)
34 print('Therefore, the vector in span{u1 u2 u3}=x-2*
      u4')
35 print(x-2*u4)

```

---

R code Exa 6.9 9

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Orthogonal projection
3 #Page No.198 / 6–6
4 #Prob 7
5 #6.3.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('Vectors u1 u2 and y are')
14 u1=matrix(c(1,3,-2),3,1,TRUE)
15 u2=matrix(c(5,1,4),3,1,TRUE)
16 y=matrix(c(1,3,5),3,1,TRUE)
17 print('y=')
18 print(y)
19 print('u2=')
20 print(u2)
21 print('u1=')
22 print(u1)
23 print('u1.u2=')
24 a=t(u1) %*% u2
25 print('=')
26 print(a)
27 print('Hence, {u1 u2} form an orthogonal basis.')
28 print('Let W=span{u1 u2}')
29 print('Therefore, projection of y on W is:')
30 print('((y.u1)/(u1.u1))*u1+((y.u2)/(u2.u2))*u2')
31 a1=t(y) %*% u1
32 a2=t(u1) %*% u1
33 b1=t(y) %*% u2
34 b2=t(u2) %*% u2
35 print('=')
36 bv=(b1/b2)
37 av=(a1/a2)
38 print(bv[1]*u2)

```

```

39 print('Therefore , projection of y on W is:')
40 ypy=y-bv[1]*u2
41 print(ypy)
42 ypy=ypy/ypy[2]
43 print(ypy)

```

---

#### R code Exa 6.10 10

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Orthogonal projection
3 #Page No.199 / 6–7
4 #Prob 13
5 #6.3.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('Given vectors are:')
14 v1=matrix(c(2,-1,-3,1),4,1,TRUE)
15 v2=matrix(c(1,1,0,-1),4,1,TRUE)
16 z=matrix(c(3,-7,2,3),4,1,TRUE)
17 print('v1=')
18 print(v1)
19 print('v2=')
20 print(v2)
21 print('z=')
22 print(z)
23 a=t(v1) %*% v2
24 print('v1.v2=')
25 print(a)
26 if(a==0){
27 print('v1 and v2 are orthogonal')

```



```

28 }
29 print('By best approximation theorem, closest point
      in span{v1 v2} to z is the orthogonal projection'
      )
30 print('=((z.v1)/(v1.v1))*v1+((z.v2)/(v2.v2))*v2')
31 a1=t(z) %*% v1
32 a2=t(v1) %*% v1
33 b1=t(z) %*% v2
34 b2=t(v2) %*% v2
35 av=(a1/a2)
36 bv=(b1/b2)
37 print('=')
38 print(av[1]*v1)
39 print('+')
40 print(bv[1]*v2)
41 print('=')
42 zb=(av[1]*v1)+(bv[1]*v2)
43 print(zb)
44 zn=z-zb
45 print(zn)
46 zn=zn/4
47 print(zn)

```

---

#### R code Exa 6.11 11

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Orthogonal decomposition theorem
3 #Page No.200 / 6–8
4 #Prob 19
5 #6.3.19
6
7 #clear console
8 cat("\014")
9
10 #clear variables

```

```

11 rm(list=ls(all=TRUE))
12
13 print('By orthogonal decomposition theorem,')
14 print('u3 is the sum of a vector in W=span{u1 u2}
    and a vector v orthogonal to W')
15 print('To find v, given u1 and u2')
16 u1=matrix(c(1,1,-2),3,1,TRUE)
17 u2=matrix(c(5,-1,2),3,1,TRUE)
18 print('u1=')
19 print(u1)
20 print('u2=')
21 print(u2)
22 print('Projection of u3 on W')
23 print('= (-1/3)*u1+(1/15)*u2')
24 print('=')
25 print((-1/3)*u1+(1/15)*u2)
26 print('v= u3-(projection of u3 on W)')
27 print('=')
28 m1=matrix(c(0,0,1),3,1,TRUE)
29 print(m1)
30 print('-')
31 print((-1/3)*u1+(1/15)*u2)
32 print(m1-((-1/3)*u1+(1/15)*u2))

```

---

## R code Exa 6.12 12

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Gram Schimdt Orthogonalisation
3 #Page No.202 / 6–10
4 #Prob 1
5 #6.4.1
6
7 #clear console
8 cat("\014")
9

```

```

10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('to orthogonalise the given vectors using Gram
    -Schmidt orthogonalisation')
14 x1=matrix(c(3,0,-1),3,1,TRUE)
15 x2=matrix(c(8,5,-6),3,1,TRUE)
16 print('x1=')
17 print(x1)
18 print('x2=')
19 print(x2)
20 print('Let v1=x1')
21 v1=x1
22 print('v2=x2-((x2.v1)/(v1.v1))*v1')
23 a1=t(x2) %*% v1
24 a2=t(v1) %*% v1
25 av=(a1/a2)
26 p=av[1]*v1
27 v2=x2-p
28 print('=')
29 print(x2)
30 print('-')
31 print(p)
32 print('=')
33 print(v2)
34 print('Thus, an orthogonal basis is:')
35 print(v1)
36 print(v2)

```

---

### R code Exa 6.13 13

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Gram Schmidt Orthogonalisation
3 #Page No.202 / 6–10
4 #Prob 7

```

```

5 #6.4.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('to orthogonalise the given vectors using Gram
    -Schmidt orthogonalisation')
14 x1=matrix(c(2,-5,1),3,1,TRUE)
15 x2=matrix(c(4,-1,2),3,1,TRUE)
16 print('x1=')
17 print(x1)
18 print('x2=')
19 print(x2)
20 print('Let v1=x1')
21 v1=x1
22 print('v2=x2-((x2.v1)/(v1.v1))*v1')
23 a1=t(x2) %*% v1
24 a2=t(v1) %*% v1
25 av=(a1/a2)
26 p=av[1]*v1
27 v2=x2-p
28 print('=')
29 print(x2)
30 print('-')
31 print(p)
32 print('=')
33 print(v2)
34 print('Thus, an orthogonal basis is:')
35 print(v1)
36 print(v2)
37 print('Normalizing v1 and v2, we get')
38 s1=sqrt(v1[1,1]^2+v1[2,1]^2+v1[3,1]^2)
39 s2=sqrt(v2[1,1]^2+v2[2,1]^2+v2[3,1]^2)
40 print(v1/s1)
41 print(v2/s2)

```

---

**R code Exa 6.14 14**

```
1 #Chapter 6– Orthogonality and Least Squares
2 #QR decomposition of a matrix
3 #Page No.202 / 6–10
4 #Prob 13
5 #6.4.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('QR decomposition of a matrix')
14 print('given matrix A=')
15 a=matrix(c(5,9,1,7,-3,-5,1,5),4,2,TRUE)
16 print(a)
17 print('given matrix Q=')
18 m1=matrix(c(5,-1,1,5,-3,1,1,3),4,2,TRUE)
19 q=(1/6)*m1
20 print(q)
21 print('Therefore , R=')
22 s=t(q) %*% a
23 print(s)
24 qr=q %*% s
25 print(qr)
```

---

**R code Exa 6.15 15**

```
1 #Chapter 6– Orthogonality and Least Squares
```

```

2 #Least square solution
3 #Page No.205 / 6-13
4 #Prob 1
5 #6.5.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('The co-efficient matrix is:')
14 a=matrix(c(-1,2,2,-3,-1,3),3,2,TRUE)
15 print('A=')
16 print(a)
17 print('The RHS is:')
18 b=matrix(c(4,1,2),3,1,TRUE)
19 print(b)
20 print('Product of transpose of A and A=')
21 p1=t(a) %*% a
22 print(p1)
23 print('Product of transpose of A and b=')
24 p2=t(a) %*% b
25 print(p2)
26 print('Hence, the solution is:')
27 p=solve(p1) %*% p2
28 print(p)

```

---

#### R code Exa 6.16 16

```

1 #Chapter 6- Orthogonality and Least Squares
2 #Least square solution
3 #Page No.206 / 6-14
4 #Prob 7
5 #6.5.7

```

```

6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('The co-efficient matrix is:')
14 a=matrix(c(1,-2,-1,2,0,3,2,5),4,2,TRUE)
15 print('A=')
16 print(a)
17 print('The RHS is:')
18 b=matrix(c(3,1,-4,2),4,1,TRUE)
19 print('b=')
20 print(b)
21 print('Product of transpose of A and A=')
22 p1=t(a) %*% a
23 print(p1)
24 print('Product of transpose of A and b=')
25 p2=t(a) %*% b
26 print(p2)
27 print('Forming an augmented matrix to solve the
      normal equations')
28 p=cbind(p1,p2)
29 print(p)
30 print('performing row operations')
31 print('R2=R2-R1')
32 p[2,]=p[2,]-p[1,]
33 print(p)
34 print('R1=R1/6 and R2=R2/36')
35 p[1,]=p[1,]/6
36 p[2,]=p[2,]/36
37 print(p)
38 print('R1=R1-R2')
39 p[1,]=p[1,]-p[2,]
40 print(p)
41 print('Hence, the solution is:')
42 print(p[,3])

```

```

43 x=p[,3]
44 print('The least square error is =||Ax-b||')
45 print('Ax-b=')
46 print(a %*% x-b)
47 c=a %*% x-b
48 s=0
49 il<-c(1:4)
50 for (i in il)
51 {
52 s=s+c[i,1]^2
53 }
54 print('||Ax-b||=')
55 print(sqrt(s))

```

---

#### R code Exa 6.17 17

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Least square solution
3 #Page No.207 / 6–15
4 #Prob 13
5 #6.5.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('To determine if u is the least square
      solution to Ax=b')
14 print('Given')
15 a=matrix(c(3,4,-2,1,3,4),3,2,TRUE)
16 print('A=')
17 print(a)
18 b=matrix(c(11,-9,5),3,1,TRUE)

```



```

19 print('b=')
20 print(b)
21 u=matrix(c(5,-1),2,1,TRUE)
22 v=matrix(c(5,-2),2,1,TRUE)
23 print('u=')
24 print(u)
25 print('v=')
26 print(v)
27 print('Au=')
28 print(a %*% u)
29 c=b-a %*% u
30 print('b-Au=')
31 print(c)
32 print('||b-Au||=')
33 print(sqrt(c[1,1]^2+c[2,1]^2+c[3,1]^2))
34 print('Av=')
35 print(a %*% v)
36 d=b-a %*% v
37 print('b-Av=')
38 print(d)
39 print('||b-Av||=')
40 print(sqrt(d[1,1]^2+d[2,1]^2+d[3,1]^2))
41 print('Since Av is more closer to A than Au, u is
      not the least square solution.')

```

---

#### R code Exa 6.18 18

```

1 #Chapter 6- Orthogonality and Least Squares
2 #Least squares line
3 #Page No.209 / 6-17
4 #Prob 1
5 #6.6.1
6
7 #clear console
8 cat("\014")

```

```

9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('To obtain a least square line from the given
      data')
14 print('Placing the x coordinates of the data in
      second column of matrix X we get:')
15 x=matrix(c(1,0,1,1,1,2,1,3),4,2,TRUE)
16 print('X=')
17 print(x)
18 print('Placing the y coordinates in y vector')
19 y=matrix(c(1,1,2,2),4,1,TRUE)
20 print('y=')
21 print(y)
22 print('Product of transpose of X and X=')
23 p1=t(x) %*% x
24 print(p1)
25 print('Product of transpose of X and y=')
26 p2=t(x) %*% y
27 print(p2)
28 print('The least square solution =')
29 print(solve(p1) %*% p2)
30 p=solve(p1) %*% p2
31 print('Hence, the least square line is:')
32 print('y')
33 print('=')
34 print(p[1,1])
35 print('+')
36 print(p[2,1])
37 print('x')

```

---

**R code Exa 6.19** 19

1 #Chapter 6– Orthogonality and Least Squares

```

2 #Page No.212 / 6-20
3 #Prob 1
4 #6.7.1
5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))
11
12 ipx=matrix(c(4,1,1),3,1,TRUE)
13 ipy=matrix(c(5,1,1),3,1,TRUE)
14 x=matrix(c(1,1),2,1,TRUE)
15 y=matrix(c(5,-1),2,1,TRUE)
16 print('x')
17 print(x)
18 print('y')
19 print(y)
20 print('x^2=')
21 xs=sqrt((ipx[1]*x[1]*x[1])+(ipy[1]*x[2]*x[2]))
22 ys=sqrt((ipx[1]*y[1]*y[1])+(ipy[1]*y[2]*y[2]))
23 print('||x||^2')
24 print(xs)
25 print('||y||^2')
26 print(ys)
27 print('||<x,y>||^2')
28 xys=((ipx[1]*x[1]*y[1])+(ipy[1]*x[2]*y[2]))^2
29 print(xys)

```

---

#### R code Exa 6.20 20

```

1 #Chapter 6- Orthogonality and Least Squares
2 #Page No.214 / 6-22
3 #Prob 25
4 #6.7.25

```

```

5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))
11
12 integrand <-function(t) {1*t}
13 print('<t,1>=')
14 v1=integrate(integrand,lower=-1,upper=1)
15 print(v1)
16 integrand1 <-function(t) {1*(t*t)}
17 print('<t^2,1>=')
18 v2=integrate(integrand1,lower=-1,upper=1)
19 print(v2)
20 integrand2 <-function(x) {1*1}
21 print('<1,1>=')
22 v3=integrate(Vectorize(integrand2),lower=-1,upper=1)
23 print(v3)
24 integrand3 <-function(t) {(t*t)*t}
25 print('<t^2,t>=')
26 v4=integrate(integrand3,lower=-1,upper=1)
27 print(v4)
28 integrand4 <-function(t) {(t*t)}
29 v5=integrate(integrand4,lower=-1,upper=1)
30 print('Thus projwt^2=')
31 projt=((v2$value/v3$value)*1)+((v4$value/v5$value))
32 print(projt)
33 print('A polynomial orthogonal to W is t^2-projt^2='
34 )
35 print('t^2-(1/3)')
36 print('thus the polynomials 1,t^2and 3t^2-1 form an
orthogonal basis for Span(1,t,t^2)')

```

---

## R code Exa 6.21 21

```
1 #Chapter 6– Orthogonality and Least Squares
2 #Applications of Inner Product Spaces
3 #Page No.215 / 6–23
4 #Prob 1
5 #6.8.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 xv<-c(1,-2,1,-1,1,0,1,1,1,2)
14 x=matrix(xv,5,2,TRUE)
15 print('X=')
16 print(x)
17 yv<-c(0,0,2,4,4)
18 y=matrix(yv,5,1,TRUE)
19 print(y)
20 Wv<-c
    (1,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,1)

21 W=matrix(Wv,5,5,TRUE)
22 WX=W %*% x
23 print('WX=')
24 print(WX)
25 Wy=W %*% y
26 print('Wy=')
27 print(Wy)
28 wxtwx=t(WX) %*% WX
29 print(' (WX) ^T*(WX)=')
30 print(wxtwx)
31 wxtwy=t(WX) %*% Wy
32 print(' (WX) ^T*(Wy)=')
33 print(wxtwy)
34 betav=solve(wxtwx) %*% (wxtwy)
```

```

35 print(' [B0 B1]= ')
36 print(betav)
37 cat('the equation of the least squares line is y=',
      betav[1], '+', betav[2], 'x')

```

---

#### R code Exa 6.22 22

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Applications of Inner Product Spaces
3 #Page No.216 / 6–24
4 #Prob 7
5 #6.8.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 integrand <-function(kt) {cos(kt)*cos(kt)}
14 print(' || coskt ||^2= ')
15 v1=integrate(integrand, lower=0, upper=360)
16 print(round(v1$value))
17
18 integrand <-function(kt) {sin(kt)*sin(kt)}
19 print(' || sinkt ||^2= ')
20 v1=integrate(integrand, lower=0, upper=360)
21 print(round(v1$value))

```

---

#### R code Exa 6.23 23

```

1 #Chapter 6– Orthogonality and Least Squares
2 #Applications of Inner Product Spaces

```

```

3 #Page No.217 / 6-25
4 #Prob 9
5 #6.8.9
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 integrand <-function(t) {(2*180)-t}
14 print('The fourier coefficients of f are:')
15 v1=integrate(integrand,lower=0,upper=360)
16 v1v=(1/2)*(1/180)*v1$value
17 print('a0/2=')
18 print(v1v)
19 print('for k=1')
20 integrand1 <-function(t) {((2*180)-t)*cos(t)}
21 v2=integrate(integrand1,lower=0,upper=360)
22 v2v=(1/180)*v2$value
23 print('ak=')
24 print(round(v2v))
25 integrand2 <-function(t) {((2*180)-t)*sin(t)}
26 v3=integrate(integrand2,lower=0,upper=360)
27 v3v=(1/180)*v3$value
28 print('bk=')
29 print(round(v3v))
30 print('The third-order Fourier approximation to f is
      :')
31 print('180 + 2sint + sin2t + (2/3)sint3t')

```

---

# Chapter 7

## Symmetric Matrices

R code Exa 7.1 1

```
1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #Symmetric Matrices
3 #Page No.222 / 7-2
4 #Prob 1
5 #7.1.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('To check if the given 2X2 matrix is symmetric
14      ')
15 #Declaration of matrix
16 a = matrix(
17   c(3,5,5,-7),
18   nrow = 2,
19   ncol = 2,
20   byrow = TRUE)
```



```

21
22 print('A=')
23 print(a)
24
25 #Check for symmetry
26 if(a[1,2]==a[2,1]){
27     print('A is a symmetric matrix because the (1,2)
        and(2,1) entries match.')
28 }else{
29     print('A is not a symmetric matrix')
30 }

```

---

### R code Exa 7.2 2

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #Orthogonal Matrix
3 #Page No.222 / 7-2
4 #Prob 7
5 #7.1.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('To show that the given matrix P is orthogonal
    .')
14 p = matrix(
15     c(0.6,0.8,0.8,-0.6),
16     nrow = 2,
17     ncol = 2,
18     byrow = TRUE)
19
20 print('P=')

```

```

21 print(p)
22 print('P is composed of two vectors.')
23 p1 = matrix(
24     c(0.6,0.8),
25     nrow = 2,
26     ncol = 1,
27     byrow = TRUE)
28
29 p2 = matrix(
30     c(0.8,-0.6),
31     nrow = 2,
32     ncol = 1,
33     byrow = TRUE)
34
35 print('p1=')
36 print(p1)
37 print('p2=')
38 print(p2)
39 print('To show that the columns are orthonormal')
40 print('p1.p2=')
41 s1=p1 * p2
42 s=s1[1]+s1[2]
43 r=t(p1)
44 print(s)
45 if(s==0)
46     print('The columns of P are orthonormal')
47 end
48 print('||p1||^2=')
49 print(sqrt(p[1,1]^2+p[2,1]^2))
50 print('||p2||^2=')
51 print(sqrt(p[1,2]^2+p[2,2]^2))
52 print('Hence, ||p1||=||p2||=1. Thus P is an
      orthogonal matrix')

```

---

**R code Exa 7.3 3**

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #PD decomposition of a matrix
3 #Page No.222 / 7-2
4 #Prob 13
5 #7.1.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('To diagonalize the given matrix A')
14
15 #Declaration of matrix
16 a = matrix(
17   c(3,1,1,3),
18   nrow = 2,
19   ncol = 2,
20   byrow = TRUE)
21
22 print('A=')
23 print(a)
24 eig=eigen(a)
25 values <- eig$values
26 print('Eigen values of A are:')
27 print(values)
28 print('for lambda=4')
29 print('A-4I=')
30 print(a-4*diag(2))
31 b=a-4*diag(2)
32 print('To find the eigenvector, form an augmented
      matrix.')
```

```

33 zm=matrix(
34   c(0,0),
35   nrow=2,
36   ncol=1,
37   byrow=TRUE)
```

```

38
39 c=cbind(b,zm)
40 print('performing row operations')
41 print(c)
42
43 c[2,]=c[2,]+c[1,]
44 print(c)
45 print('With x2 as free variable , x1=x2')
46 print('Hence a basis for the eigenspace is:')
47 d=matrix(
48     c(1,1),
49     nrow=2,
50     ncol=1,
51     byrow=TRUE)
52
53 print(d)
54 print('Upon normalizing')
55 print(d/(sqrt(2)))
56 u1=d/(sqrt(2))
57 print('for lambda=2')
58 print('A-2I=')
59 b=a-2*diag(2)
60 print(b)
61 print('To find the eigenvector , form an augmented
        matrix.')
62 c=cbind(b,zm)
63 print('performing row operations')
64 print(c)
65 c[2,]=c[2,]-c[1,]
66 print(c)
67 print('With x2 as free variable , x1=-x2')
68 print('Hence a basis for the eigenspace is:')
69 d=matrix(
70     c(-1,1),
71     nrow=2,
72     ncol=1,
73     byrow=TRUE)
74

```

```

75 print(d)
76 print('Upon normalizing ')
77 print(d/(sqrt(2)))
78 u2=d/(sqrt(2))
79 print('Matrix P=')
80 p=cbind(u1,u2)
81 print(p)
82 print('The corresponding matrix D=')
83
84 D1=matrix(
85   c(values[1],0),
86   nrow=1,
87   ncol=2,
88   byrow=TRUE)
89
90 D2=matrix(
91   c(0,values[2]),
92   nrow=1,
93   ncol=2,
94   byrow=TRUE)
95
96
97 D = matrix(
98   c(D1,D2),
99   nrow=2,
100  ncol=2,
101  byrow = TRUE)
102 print(D)

```

---

#### R code Exa 7.4 4

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #PD decomposition of a matrix
3 #Page No.223 / 7-3
4 #Prob 19

```

```

5 #7.1.19
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 print('PD decomposition of a matrix A')
14 av<-c(3,-2,4,-2,6,2,4,2,3)
15 a=matrix(av,3,3,TRUE)
16 print('A=')
17 print(a)
18 print('Eigenvalues of A are')
19 eig=eigen(a)
20 evv=eig$values
21 print(evv)
22 cat('for lambda =',evv[1])
23 print('A-(lambda)I=')
24 b=a-evv[1]*diag(3)
25 print(b)
26 print('To find eigenvector, form an augmented matrix
      ')
27 zm=matrix(0,3,1,TRUE)
28 cm=cbind(b,zm)
29 print(cm)
30 print('performing row operations')
31 cm[2,]=cm[2,]-(cm[2,1]/cm[1,1])*cm[1,]
32 cm[3,]=cm[3,]-(cm[3,1]/cm[1,1])*cm[1,]
33 cm[1,]=cm[1,]/-4
34 print(cm)
35 print('With x2 and x3 as free variables, we get two
      vectors.')
36 print('x1=-.5x2+x3')
37 print('Thus, the two vectors are')
38 v1=matrix(c(-1,2,0),3,1,TRUE)
39 v2=matrix(c(1,0,1),3,1,TRUE)
40 print('v1=')

```

```

41 print(v1)
42 print('v2=')
43 print(v2)
44 print('Orthogonalizing v1 and v2')
45 print('Let x1=v1')
46 print('x2=v2-((v2.v1)/(v1.v1))*v1')
47 x1=v1
48 a1=t(v2) %*% v1
49 a2=t(v1) %*% v1
50 av=(a1/a2)
51 x2=v2-av[1]*v1
52 x1=x1/(sqrt(x1[1,1]^2+x1[2,1]^2+x1[3,1]^2))
53 u1=x1
54 x1=x2/(sqrt(x2[1,1]^2+x2[2,1]^2+x2[3,1]^2))
55 print(x2)
56 x2=x2*5
57 print(x2)
58 print('An orthonormal basis is:')
59 print('u1=')
60 print(u1)
61 print('u2=')
62 print(x1)
63 cat('for lambda=',evv[3])
64 print('A-(lambda)I=')
65 b1=a-evv[3]*diag(3)
66 print(b1)
67 print('To find eigenvector, form an augmented matrix
    ')
68 cm1=cbind(b1,zm)
69 print(cm1)
70 print('performing row operations')
71 cm1[2,]=cm1[2,]/2
72 cm1[c(1,2),]=cm1[c(2,1),]
73 cm1[1,]=cm1[1,]/-1
74 print(cm1)
75 cm1[2,]=round(cm1[2,]-5*cm1[1,])
76 cm1[3,]=cm1[3,]-4*cm1[1,]
77 print(cm1)

```

```

78 cm1[3,]=round(cm1[3,]-cm1[2,])
79 cm1[2,]=cm1[2,]/18
80 print(cm1)
81 cm1[1,]=round(cm1[1,]+4*cm1[2,])
82 print(cm1)
83 print('With x3 as free variable')
84 print('x1=x3 and x2=-.5x3')
85 print('Thus a basis for the eigenspace is:')
86 v3=matrix(c(-2,-1,2),3,1,TRUE)
87 print(v3)
88 print('upon normalizing')
89 v3=v3/(sqrt(v3[1,1]^2+v3[2,1]^2+v3[3,1]^2))
90 print(v3)
91 print('Thus, matrix P=')
92 print(cbind(u1,x1,v3))
93 print('Corresponding matrix D=')
94 D1=cbind(evv[1],0,0)
95 D2=cbind(0,evv[2],0)
96 D3=cbind(0,0,evv[3])
97 D=rbind(D1,D2,D3)
98 print(D)

```

---

## R code Exa 7.5 5

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #Quadratic Form
3 #Page No.226 / 7–6
4 #Prob 1
5 #7.2.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))

```



```

12
13 print('given matrix A and vector x')
14 a = matrix(
15     c(5,(1/3),(1/3),1),
16     nrow = 2,
17     ncol = 2,
18     byrow = TRUE)
19
20 print('A=')
21 print(a)
22
23 x = matrix(
24     c(6,1),
25     nrow = 2,
26     ncol = 1,
27     byrow = TRUE)
28 print('x=')
29 print(x)
30
31 print('Product of transpose of x and A and x=')
32 tx=t(x)
33 p=tx %*% a %*% x
34 print(p)
35 print('New value of vector x=')
36
37 x = matrix(
38     c(1,3),
39     nrow = 2,
40     ncol = 1,
41     byrow = TRUE)
42
43 print(x)
44 print('Product of transpose of x and A and x=')
45 tx1=t(x)
46 p=tx1 %*% a %*% x
47 print(p)

```

---

## R code Exa 7.6 6

```
1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #Page No.226 / 7–6
3 #Prob 7
4 #7.2.7
5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))
11
12 a=matrix(c(1,5,5,1),2,2,TRUE)
13 print('A=')
14 print(a)
15 eig=eigen(a)
16 eigv<-eig$values
17 print('Eigen values are:')
18 print(eigv)
19 ev=eig$vectors
20 ev1=(ev[,1]/(ev[1,1]))
21 print('For lambda=6')
22 print('an eigen vector is')
23 print(ev1)
24 print('normalized u1=')
25 print(eig$vectors[,1])
26 ev2=(ev[,2]/(ev[1,1]))
27 print('For lambda=-4')
28 print('an eigen vector is')
29 print(ev2)
30 print('normalized u2=')
31 print(eig$vectors[,2])
32 pm=eig$vectors
```

```

33 print('P=')
34 print(pm)
35 Dm=t(pm) %*% a %*% pm
36 print('D=')
37 print(round(Dm))

```

---

### R code Exa 7.7 7

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #Page No.227 / 7-7
3 #Prob 13
4 #7.2.13
5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))
11
12 a=matrix(c(1,-3,-3,9),2,2,TRUE)
13 print('A=')
14 print(a)
15 eig=eigen(a)
16 eigv<-round(eig$values)
17 print('Eigen values are:')
18 print(eigv)
19 ev=eig$vectors
20 ev1=(ev[,1]/(ev[1,1]))
21 print('For lambda=10')
22 print('an eigen vector is')
23 print(ev1)
24 print('normalized u1=')
25 print(-eig$vectors[,1])
26 ev2=(ev[,2]/(ev[1,1]))
27 print('For lambda=0')

```

```

28 print('an eigen vector is')
29 print(ev2)
30 print('normalized u2=')
31 print(-eig$vectors[,2])
32 pm=-eig$vectors
33 print('P=')
34 print(pm)
35 Dm=t(pm) %*% a %*% pm
36 print('D=')
37 print(round(Dm))

```

---

#### R code Exa 7.8 8

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #Page No.229 / 7-9
3 #Prob 1
4 #7.3.1
5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))
11
12 av<-c(5,2,0,2,6,-2,0,-2,7)
13 a=matrix(av,3,3,TRUE)
14 print('A=')
15 print(a)
16 eig=eigen(a)
17 eigv<-eig$values
18 print('eigen values are:')
19 print(eigv)
20 print('for lambda=9')
21 print('normalized eigenvector u1=')
22 print(-eig$vectors[,1])

```

```

23 print('for lambda=6')
24 print('normalized eigenvector u2=')
25 print(eig$eigenvectors[,2])
26 print('for lambda=3')
27 print('normalized eigenvector u3=')
28 print(-eig$eigenvectors[,3])
29 print('x=Py')
30 print('P=')
31 p1=-eig$eigenvectors[,1]
32 p2=eig$eigenvectors[,2]
33 p3=-eig$eigenvectors[,3]
34 p=cbind(p1,p2,p3)
35 print(p)

```

---

#### R code Exa 7.9 9

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #Page No.229 / 7–9
3 #Prob 7
4 #7.3.7
5
6 #clear console
7 cat("\014")
8
9 #clear variables
10 rm(list=ls(all=TRUE))
11
12 av<-c(-2,2,0,2,-1,2,0,2,0)
13 a=matrix(av,3,3,TRUE)
14 print('A=')
15 print(a)
16 eig=eigen(a)
17 eigv<-eig$values
18 print('eigen values are:')
19 print(eigv)

```

```

20 print('2 is the greatest eigen value')
21 eigve<-eig$vectors[,1]
22 v1=eigve/eigve[3]
23 print('v1=')
24 print(v1)
25 v1=v1*2
26 print('scaled v1=')
27 print(v1)
28 print('u1=')
29 print(eigve)

```

---

#### R code Exa 7.10 10

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #The Singular Value Decomposition
3 #Page No.232 / 7–12
4 #Prob 1
5 #7.4.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 a=matrix(c(1,0,0,-3),2,2,TRUE)
14 print('A=')
15 print(a)
16 print('ATA')
17 ata=t(a) %*% a
18 print(ata)
19 eig=eigen(ata)
20 eigv<-eig$values
21 print('eigen values are:')
22 print(eigv)

```

```

23 print('eigen vectors are:')
24 print('for lambda=9')
25 print(eig$vectors[,1])
26 print('for lambda=1')
27 print(-eig$vectors[,2])
28 r1=eig$vectors[,1]
29 r2=-eig$vectors[,2]
30 V=cbind(r1,r2)
31 print('V=')
32 print(V)
33 sv1=sqrt(eigv[1])
34 sv2=sqrt(eigv[2])
35 print('Singular values are:')
36 print(sv1)
37 print(sv2)
38 sigmam=matrix(c(sv1,0,0,sv2),2,2,TRUE)
39 print('The matrix is the same shape as A=')
40 print(sigmam)
41 Av1=a %*% r1
42 Av2=a %*% r2
43 print('Av1=')
44 print(Av1)
45 print('Av2=')
46 print(Av2)
47 u1=Av1/sv1
48 u2=Av2/sv2
49 print('normalize:')
50 print('u1=')
51 print(u1)
52 print('u2=')
53 print(u2)
54 um=cbind(u1,u2)
55 print(um)
56 print('A=')
57 print(um)
58 print('*')
59 print(sigmam)
60 print('*')

```

```
61 print(t(V))
```

---

### R code Exa 7.11 11

```
1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #The Singular Value Decomposition
3 #Page No.232 / 7–12
4 #Prob 7
5 #7.4.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 a=matrix(c(2,-1,2,2),2,2,TRUE)
14 print('A=')
15 print(a)
16 ata=t(a) %*% a
17 print('ATA=')
18 print(ata)
19 eig=eigen(ata)
20 print('eigen values are:')
21 eigv<-eig$values
22 print(eigv)
23 eigve1<-(-eig$vectors[,1])
24 eigve2<-(-eig$vectors[,2])
25 print('for lambda=9')
26 print(eigve1)
27 print('for lambda=4')
28 print(eigve2)
29 V=cbind(eigve1,eigve2)
30 print(V)
31 sv1=sqrt(eigv[1])
```



```

32 sv2=sqrt(eigv[2])
33 print('Singular values are')
34 print(sv1)
35 print(sv2)
36 c1<-c(sv1,0)
37 c2<-c(0,sv2)
38 sigmam=cbind(c1,c2)
39 av1=a %*% eigve1
40 av2=a %*% eigve2
41 print('Av1')
42 print(av1)
43 print('Av2')
44 print(av2)
45 print('Check orthogonality')
46 co=av1*av2
47 co=co[1]+co[2]
48 print('Av1.Av2=')
49 print(co)
50 u1=av1/sv1
51 u2=av2/sv2
52 print('u1=')
53 print(u1)
54 print('u2=')
55 print(u2)
56 um=cbind(u1,u2)
57 print(um)
58 print('A=')
59 print(um)
60 print('*')
61 print(sigmam)
62 print('*')
63 print(t(V))

```

---

**R code Exa 7.12 12**

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #The Singular Value Decomposition
3 #Page No.233 / 7–13
4 #Prob 13
5 #7.4.13
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 av<-c(3,2,2,2,3,-2)
14 a=matrix(av,2,3,TRUE)
15 print(a)
16 aat=a %*% t(a)
17 print(aat)
18 eig=eigen(aat)
19 eigv<-eig$values
20 print('eigen values are')
21 print(eigv)
22 v1=eig$vectors[,1]
23 v2=eig$vectors[,2]
24 print('eigen vectors are')
25 print('for lambda=25')
26 print(v1)
27 print('for lambda=9')
28 print(v2)
29 V=cbind(v1,v2)
30 print(V)
31 sv1=sqrt(eigv[1])
32 sv2=sqrt(eigv[2])
33 print('singular values are')
34 print(sv1)
35 print(sv2)
36 zm=matrix(0,1,2,TRUE)
37 c1=matrix(c(sv1,0),1,2,TRUE)
38 c2=matrix(c(0,sv2),1,2,TRUE)

```

```

39 sigmam=rbind(c1,c2,zm)
40 print(sigmam)
41 print(t(a) %*% V)
42 u1=(t(a) %*% v1)/sv1
43 u2=(t(a) %*% v2)/sv2
44 print('u1=')
45 print(u1)
46 print('u2=')
47 print(u2)
48 u3=matrix(c((-2/3),(2/3),(1/3)),3,1,TRUE)
49 print('u3=')
50 print(u3)
51 u=cbind(u1,u2,u3)
52 print('A=')
53 print(V)
54 print('*')
55 print(t(sigmam))
56 print('*')
57 print(t(u))

```

---

### R code Exa 7.13 13

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #Applications to Image Processing and Statistics
3 #Page No.237 / 7–17
4 #Prob 1
5 #7.5.1
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 xv<-c(19,22,6,3,2,20,22,6,9,15,13,5)

```

```

14 x=matrix(xv,2,6,TRUE)
15 B=x
16 print(x)
17 M=matrix(c(12,10),2,1,TRUE)
18 print(M)
19 B[1,]=x[1,] - M[1]
20 B[2,]=x[2,] - M[2]
21 print(B)
22 #There is a calculation error in the textbook
23 S1=B %*% t(B)
24 S=S1/(dim(B)[2]-1)
25 print(S)

```

---

#### R code Exa 7.14 14

```

1 #Chapter 7 – Symmetric Matrices And Quadratic Forms
2 #Applications to Image Processing and Statistics
3 #Page No.237 / 7–17
4 #Prob 7
5 #7.5.7
6
7 #clear console
8 cat("\014")
9
10 #clear variables
11 rm(list=ls(all=TRUE))
12
13 # Constructing Quadratic Formula
14 result <- function(a,b,c){
15   if(delta(a,b,c) > 0){ # first case D>0
16     x_1 = (-b+sqrt(delta(a,b,c)))/(2*a)
17     x_2 = (-b-sqrt(delta(a,b,c)))/(2*a)
18     result = c(x_1,x_2)
19   }
20   else if(delta(a,b,c) == 0){ # second case D=0

```

```

21     x = -b/(2*a)
22 }
23 else {"There are no real roots."} # third case D<0
24 }
25
26 # Constructing delta
27 delta<-function(a,b,c){
28     b^2-4*a*c
29 }
30
31 a <- result(1,-102,647)
32 print(a)
33 print('Two possible choices for new variable are:')
34 print('y1=-.95x1+.32x2 and')
35 print('y1=.95x1-.32x2')
36 print('variance of y1 is')
37 print(a[1])
38 print('total variance is')
39 print(a[1]+a[2])
40 print('the new variable y1 explains about')
41 print(a[1]/(a[1]+a[2]))
42 print('% of the variance in the data')

```

---