

R Textbook Companion for
Introduction to Linear Algebra
by Gilbert Strang¹

Created by
Malapati Venkata Dharun Raghava
B.Tech.
Computer Science and Engineering
Sri Venkateswara University College of Engineering
Cross-Checked by
R TBC Team

May 26, 2020

¹Funded 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: Introduction to Linear Algebra

Author: Gilbert Strang

Publisher: Wellesley - Cambridge Press, Wellesley MA USA

Edition: 4

Year: 2009

ISBN: 978-0-9802327-1-4

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 Introduction to Vectors	5
2 Solving Linear Equations	9
3 Vector Spaces and Subspaces	25
4 Orthogonality	50
5 Determinants	65
6 Eigen Values and Eigen Vectors	72
7 Linear Transformations	91
8 Applications	95
9 Numerical Linear Algebra	100
10 Complex numbers and Matrices	102

List of R Codes

Exa 1.1	Sum of two vectors	5
Exa 1.2.a	Schwartz and Traingular Inequality	5
Exa 1.2.b	Unit Vector and checking orthogonaity	6
Exa 1.2.c	Solution of Linear System	6
Exa 1.2.1	Dot Product of Vectors	7
Exa 1.2.2	Dot Product of Vectors	7
Exa 1.2.5	Angle between two vectors	8
Exa 1.3.a	Inverse of the matrix	8
Exa 2.1.a	Solving the equations by Column picture	9
Exa 2.1.1	Multiplication of vector as dotproducts of rows and cols	9
Exa 2.2.a	Pivots and Multipliers in converting matrix to upper traingular system	10
Exa 2.3.b	Multiplication with Elimination and Permutation matrices	11
Exa 2.3.c	Multiplying Matrices in two different ways	12
Exa 2.3.1	Matrix multiplication as dotproduct of row and column	12
Exa 2.3.2	Purpose of Elimination matrices	13
Exa 2.4.a	Square of Pascal Matrix is HyperCube Matrix	13
Exa 2.4.b	Commutative property does not hold for matrix multiplication	13
Exa 2.4.c	3step paths for the given directed graph	14
Exa 2.4.1	Multiplication of Square matrices	14
Exa 2.4.2	Column times Row	15
Exa 2.5.a	Inverse of difference matrix is Sum matrix	15
Exa 2.5.b	Inverse of the given matrices	15
Exa 2.5.c	Inverse of the Pascal matrix by gauss jordan elimination method	16
Exa 2.5.2	Inverse of an Elimination Matrix	16

Exa 2.5.3	Inverse of product of matrices	17
Exa 2.5.4	Inverse of matrix by guass jordan elimination matrix .	17
Exa 2.5.5	Inverse of Lower traingular matrix is also a lower train- gular matrix	18
Exa 2.6.a	LU Factorisation	19
Exa 2.6.b	Forward Elimination	19
Exa 2.6.1	LU Factorisation	20
Exa 2.6.2	LU Factorisation	21
Exa 2.6.3	Forward Elimination	21
Exa 2.7.a	Multiplication by Permutation matrices	22
Exa 2.7.b	Symmetric Factorisation	22
Exa 2.7.1	Inverses and Transposes	23
Exa 2.7.4	Product of matrix and its trnspose gives symmetric ma- trix	24
Exa 3.2.a	Find the Matrix having the given special solutions . .	25
Exa 3.2.b	Special solution pivot columns free columns and reduced row echelon form of the given matrix	25
Exa 3.2.2	Nullspace of given Singular Matrix	28
Exa 3.2.3	Nullspaces of given three matrices	30
Exa 3.2.4	Nullspace of given upper traingular Matrix	32
Exa 3.3.a	Row reduced echelon form rank and special solutions of given matrix	34
Exa 3.3.c	Special solutions and row reduced echelon forms . . .	36
Exa 3.4.a	Complete solution of the given system	38
Exa 3.4.c	Complete solution of the given system	41
Exa 3.4.2	particular solution and special solution	43
Exa 3.5.1	Columns of given matrix are linearly dependent	45
Exa 3.6.a	Four Fundamental Spaces of given matrix	46
Exa 3.6.1	Dimensions and rank of matrix	48
Exa 3.6.2	Dimensions and rank of matrix	49
Exa 4.1.a	Dimensions of the subspaces in the given space	50
Exa 4.1.b	Null space Basis of a plane subspace	50
Exa 4.1.3	Rows of matrix are perpendicualr to the vectors in nullspace	53
Exa 4.2.a	Projection onto the line and the plane	53
Exa 4.2.b	Best possible solution	54
Exa 4.2.1	Projection of vector onto line	55
Exa 4.2.2	Projection Matrix of the line	56

Exa 4.2.3	Best possible solution projection vector and Projection Matrix	56
Exa 4.3.a	Fit a straight line	57
Exa 4.3.b	Fit a Parabola	58
Exa 4.3.1	Fit a straight line	59
Exa 4.3.2	Fit a straight line	60
Exa 4.3.3	Fit a Parabola	61
Exa 4.4.4	Projections of vector onto the line plane if the basis are given as orthonormal vectors	62
Exa 4.4.5	Gram Schmidt method to convert matrix to its orthogonal form	63
Exa 5.2.1	Determinant of matrix by Product of pivots	65
Exa 5.2.5	Determinants of matrices	65
Exa 5.2.7	Determinants of matrices	66
Exa 5.3.a	Null space of matrix is the transpose of cofactor matrix	66
Exa 5.3.b	Solve by Crammers rule and inverse of matrix	67
Exa 5.3.1	Crammers rule for solving system of equations	68
Exa 5.3.3	The inverse of triangular matrix is triangular matrix	69
Exa 5.3.7	Cross product of vectors	69
Exa 5.3.8	Cross product of vectors	70
Exa 5.3.9	Right hand rule	70
Exa 6.1.a	Eigen values and eigen vectors	72
Exa 6.1.b	Eigen values and eigen vectors	73
Exa 6.1.1	Eigen values and eigen vectors	74
Exa 6.1.2	Eigen values and eigen vectors of Projection matrix	75
Exa 6.1.3	Eigen values and eigen vectors of Reflection matrix	75
Exa 6.1.4	Eigen values and eigen vectors of Singular matrix	76
Exa 6.2.b	Inverse Eigen values Determinant and eigen vector matrix	78
Exa 6.2.1	Diagonalizing a Matrix and Power of matrix computed from power of its diagonal matrix	80
Exa 6.2.2	The diagonal matrix of any matrix contains the eigen values in its main diagonal	80
Exa 6.3.b	Eigen values and eigen vectors	81
Exa 6.3.1	Solve Differential equation	82
Exa 6.3.2	Solve Differential equation	83
Exa 6.3.6	Solve Differential equation	84
Exa 6.4.b	Eigen values and eigen vectors	84

Exa 6.4.1	Eigen values and eigen vectors	85
Exa 6.4.4	pivots and eigen values have same signs for symmetric matrices	86
Exa 6.5.1	Tests for positive definiteness	87
Exa 6.6.a	Pascals Matrix and its inverse are similar	88
Exa 6.6.1	Similar matrices are matrices with same eigen values	88
Exa 6.6.2	Similar matrices with repeated eigen values	89
Exa 6.6.3	Jordans Theorem and jordan Matrix	89
Exa 6.7.3	Singular Value Decomposition	90
Exa 6.7.4	Singular Value Decomposition	90
Exa 7.3.a	leftinverse rightinverse Pseduoinverse of given matrices	91
Exa 7.3.1	Diagonalization of matrix	92
Exa 7.3.2	Similar Projection Matrices	92
Exa 7.3.3	Polar Decomposition	93
Exa 7.3.4	Pseduo Inverse of a matrix	93
Exa 8.1.1	Movements tensions elongations of spring	95
Exa 8.1.2	Movements tensions elongations of spring	95
Exa 8.2.1	Find the currents	96
Exa 8.3.1	Positive Markov matrix application	97
Exa 8.3.3	Markov matrix application	97
Exa 8.3.4	Linear Algebra in economy	97
Exa 8.3.5	Linear Algebra in economy	98
Exa 8.5.2	length of function $\sin x$ and $\sin x$ and $\cos x$ are orthogonal in function space	98
Exa 8.6.2	Singular Value Decomposition	99
Exa 9.2.2	Norm of a diagonal matrix	100
Exa 9.2.3	Condition number of Positive definite Matrix	100
Exa 9.3.1	Powers of some matrices can be calculated easily with max eigen value	101
Exa 10.1.1	r for complex numbers	102
Exa 10.2.1	Orthogonal Complex vectors	102

Chapter 1

Introduction to Vectors

R code Exa 1.1 Sum of two vectors

```
1 #Example : 1 Chapter : 1 pageno : 1
2 v<-matrix(c(1,1),2,1,TRUE)
3 w<-matrix(c(2,3),2,1,TRUE)
4 z<-v+w
5 z
```

R code Exa 1.2.a Schwartz and Traingular Inequality

```
1 #Example : 1.2A Chapter : 1.2 Pageno : 17
2 #Finds whether Schwartz and traingular inequality
  between the given vectors are satisfied
3 #Finds the Cosine of the angle between the given
  vectors
4 v<-c(3,4)
5 w<-c(4,3)
6 dp=sum(v*w)
7 magn_v=sqrt(sum(v*v))
8 magn_w=sqrt(sum(w*w))
```

```

9  if(dp<=magn_v*magn_w){
10    print("Schwartz inequality is satisfied for the
        given vectors")
11  }
12  z=v+w
13  magn_sum=sqrt(sum(z*z))
14  if(magn_sum<=magn_v+magn_w){
15    print("Traingular inequality is satisfied for the
        given vectors")
16  }
17  print(paste("Cosine of the angle between the given
        vectors is ",dp/(magn_v*magn_w)))

```

R code Exa 1.2.b Unit Vector and checking orthogonaity

```

1  #Example : 1.2B    Chapter : 1.2    Pageno:-18
2  v<-c(3,4)
3  V<-v/sqrt(sum(v*v))
4  print(paste("The Unit vector in the direction of the
        given vector is ", V[1]," ",V[2]))
5  u<-c(-4,3)
6  if(sum(u*v)==0){
7    print("The vectors u and v are perpendicular to
        each other")
8  }
9  U = u/sqrt(sum(u*u))
10 print(paste("The unit vector in the direction of u
        is",U[1]," ",U[2]))

```

R code Exa 1.2.c Solution of Linear System

```

1  #Example : 1.2C    Chapter : 1.2    Pageno : 18
2  A<-matrix(c(2,-1,-1,2),ncol=2)

```

```

3 b<-c(1,0)
4 x<-solve(A,b)
5 print("The solution of system is :")
6 print(x)

```

R code Exa 1.2.1 Dot Product of Vectors

```

1 #Example : 1      Chapter : 1.2      pgno:-11
2
3 #Computing the dotproduct of two vectors
4 dotproduct<-function(x,y){
5   res<-0
6   for(i in 1:length(x)){
7     res<-res+x[i]*y[i]
8   }
9   res
10 }
11 v<-matrix(c(4,2),nrow=2,ncol=1,byrow = T)
12 w<-matrix(c(-1,2),nrow=2,ncol=1,byrow=T)
13 r<-dotproduct(v,w)
14 print(paste("Dot product of given vectors is",r))

```

R code Exa 1.2.2 Dot Product of Vectors

```

1 #Example : 2      Chapter : 1.2      pgno:-11
2
3 #Computing the dotproduct of two vectors
4 dotproduct<-function(x,y){
5   dp<-sum(x*y)
6   return(dp)
7 }
8 weight<-matrix(c(4,2),nrow=2,ncol=1,byrow = T)
9 distance<-matrix(c(-1,2),nrow=2,ncol=1,byrow=T)

```

```
10 center_point<-dotproduct(weight,distance)
11 center_point
```

R code Exa 1.2.5 Angle between two vectors

```
1 #Example : 5      Chapter : 1.2      pageno : 16
2
3 v<-c(2,1)
4 w<-c(1,2)
5 dotproduct<-sum(v*w)
6 magn_v<-sqrt(sum(v*v))
7 magn_w<-sqrt(sum(w*w))
8 cos_angle<-dotproduct/(magn_v*magn_w)
9 print(paste("Cosine of the angle between the given
      vectors is",cos_angle))
```

R code Exa 1.3.a Inverse of the matrix

```
1 #Example : 1.3A    Chapter : 1.3    Pageno : 27
2 A=matrix(c(1,0,0,-1,1,0,1,-1,1),ncol=3,byrow=T)
3 A1=solve(A) # to find the inverse of the matrix
4 A1
```

Chapter 2

Solving Linear Equations

R code Exa 2.1.a Solving the equations by Column picture

```
1 # Example : 2.1A Chapter : 2.1 Pageno : 39
2 #Solving the equations by Column picture
3 #Column Picture : solution is the linear combination
  of columns of A that makes b
4 A=matrix(c(1,2,3,3,2,5,2,2,6),ncol=3)
5 b=c(-3,-2,-5)
6 x<-solve(A,b)
7 x
8 #if b=(4,4,8)
9 b=c(4,4,8)
10 x<-solve(A,b)
11 x
```

R code Exa 2.1.1 Multiplication of vector as dotproducts of rows and cols

```
1 #Example : 1 Chapter : 2.1 Pageno : 37
2 #Multiplication by rows and cols dotproduct.
3 multiply<-function(A,x){
```

```

4   b<-c()
5   for(i in 1:3){
6     b<-c(b,sum(A[i,]*x[,1]))
7   }
8   b<-matrix(b,ncol=1)
9   print(paste(" Multiplying matrices by dotproduct of
      rows and cols"))
10  print(A)
11  print("*")
12  print(x)
13  print("=")
14  print(b)
15 }
16
17
18 I<-matrix(c(1,0,0,0,1,0,0,0,1),ncol=3,byrow=T)
19 x<-matrix(c(4,5,6),ncol=1)
20 A=matrix(c(1,1,1,0,0,0,0,0,0),ncol=3)
21 multiply(A,x)
22 A<-matrix(c(1,0,0,0,1,0,0,0,1),nrow=3,ncol=3,byrow=T
  )
23 multiply(A,x)

```

R code Exa 2.2.a Pivots and Multipliers in converting matrix to upper traingular system

```

1 #Example : 2.2A Chapter : 2.2 page no :
   50
2 #Pivots and Multipliers in converting matrix to
   upper traingular system
3 matrix(c(1,-1,0,-1,2,-1,0,-1,2),ncol=3)->A
4 A
5 print(paste(" First pivot is",A[1,1]))
6 l21<-A[2,1]/A[1,1]
7 print(paste(" Multiplier L21 to convert the second

```

```

    row first element to 0 is",l21))
8  A[2,]<-A[2,]-l21*A[1,]
9  A
10 print(paste("The second pivot is ",A[2,2]))
11 l32<-A[3,2]/A[2,2]
12 A[3,]<-A[3,]-l32*A[2,]
13 print("The equivalent Upper traingular system for
    the matrix A is ")
14 A

```

R code Exa 2.3.b Multiplication with Elimination and Permutation matrices

```

1 #Example : 2.3b      Chapter : 2.3      Pageno : 61
2 #Multiplication with Elimination and Permutation
  matrices
3 Ab<-matrix(c(1,4,0,2,8,3,2,9,2,1,3,1),ncol=4)
4 E21<-matrix(c(1,0,0,-4,1,0,0,0,1),ncol=3,byrow=T)
5 P32<-matrix(c(1,0,0,0,0,1,0,1,0),ncol=3,byrow=T)
6 E21Ab<-E21%*%Ab
7 print(E21Ab)
8 P32E21Ab<-P32%*%E21Ab
9 print(P32E21Ab)
10 P32E21<-P32%*%E21
11 print(P32E21)
12 P32E21Ab<-P32E21%*%Ab
13 print(P32E21Ab)
14 #Solution for this system is
15 b<-P32E21Ab[,4]
16 P32E21Ab<-P32E21Ab[, -4]
17 x<-solve(P32E21Ab,b)
18 print(x)

```

R code Exa 2.3.c Multiplying Matrices in two different ways

```
1 # Example : 2.3C Chapter : 2.3 Pageno : 62
2 # Multiplying Matrices in two different ways
3 A<-matrix(c(3,1,2,4,5,0),ncol=2)
4 B<-matrix(c(2,1,4,1),ncol=2)
5 AB<-A%%B
6 print(AB)
7 #Multiplying matrices A and B as Rows of A times
  columns of B as dot product
8 for(r in 1:dim(A)[1]){
9   for(c in 1:dim(B)[2]){
10     AB[r,c]<-sum(A[r,]*B[,c])
11   }
12 }
13 print(AB)
14 #Multiplying Matrices A and B as Columns of A times
  rows of B
15 AB1<-matrix(A[,1],ncol=1)%%matrix(B[1,],nrow=1)
16 AB2<-matrix(A[,2],ncol=1)%%matrix(B[2,],nrow=1)
17 AB<-AB1+AB2
18 print(AB)
```

R code Exa 2.3.1 Matrix multiplication as dotproduct of row and column

```
1 # Example : 1 Chapter : 2.3 page no : 57
2 #Matrix multiplication as dotproduct of row and
  column
3 A<-matrix(c(3,4,5,6),nrow=2,ncol=2,byrow=T)
4 x<-c(2,1)
5 B<-A%%x
6 B
7 Ax<-c()
8 for(i in 1:2){
9   Ax<-c(Ax,sum(A[i,]*x))
```



```

10 }
11 Ax<-matrix(Ax,ncol=1)
12 Ax

```

R code Exa 2.3.2 Purpose of Elimination matrices

```

1 #Example : 2 Chapter : 2.3 pageno : 58
2 #Purpose of Elimination matrices
3 I<-matrix(c(1,0,0,0,1,0,0,0,1),ncol=3)
4 E31<-matrix(c(1,0,-4,0,1,0,0,0,1),ncol=3)
5 b<-matrix(c(1,3,9),ncol=1)
6 Ib<-I%%b
7 print(Ib)
8 Eb<-E31%%b
9 print(Eb)

```

R code Exa 2.4.a Square of Pascal Matrix is HyperCube Matrix

```

1 #Example : 2.4A Chapter : 2.4 Page no
: 72
2 #Square of Pascal Matrix is HyperCube Matrix
3 L<-matrix(c(1,1,1,1,0,1,2,3,0,0,1,3,0,0,0,1),ncol=4)
4 H<-L%%L
5 print("Square of Pascal Matrices is:")
6 print(H)
7 H1<-H%%matrix(c(1,1,1,1),ncol=1)
8 print(H1)

```

R code Exa 2.4.b Commutative property does not hold for matrix multiplication

```

1 #Example : 2.4B          Chapter:2.4          Pageno :
   74
2
3 B<-matrix(c(1,0,1,1),ncol=2)
4 C<-matrix(c(0,0,1,0),ncol=2)
5 BC<-B%%C
6 CB<-C%%B
7 print(BC)
8 print(CB)
9 print("Commutative property does not hold for matrix
      multiplication but by chance here it happened BC
      =CB")

```

R code Exa 2.4.c 3step paths for the given directed graph

```

1 #Example : 2.4C          Chapter : 2.4          Pageno :
   74
2 #3-step paths for the given directed graph
3 A<-matrix(c(1,1,1,0),ncol=2)
4 A3<-A%%A%%A
5 print("3 step paths between each pair of nodes in
      the given directed graph is :")
6 print(A3)

```

R code Exa 2.4.1 Multiplication of Square matrices

```

1 # Example : 1          Chapter : 2.4
   Pageno : 68
2 #Multiplication of Square matrices
3 A<-matrix(c(1,2,1,-1),ncol=2)
4 B<-matrix(c(2,3,2,4),ncol=2)
5 AB<-A%%B
6 print(AB)

```

R code Exa 2.4.2 Column times Row

```
1 #Example : 2          Chapter : 2.4          Page no :  
    68  
2 #Column times Row  
3 A<-matrix(c(0,1,2),ncol=1)  
4 B<-matrix(c(1,2,3),nrow=1)  
5 AB<-A%*%B  
6 print(AB)
```

R code Exa 2.5.a Inverse of difference matrix is Sum matrix

```
1 # Example : 2.5A    Chapter : 2.5    Pageno : 87  
2 # Inverse of difference matrix is Sum matrix  
3 A<-matrix(c(1,-1,0,0,1,-1,0,0,1),ncol=3)  
4 A1<-solve(A)  
5 print("Inverse of the difference matrix is singular  
    ")  
6 print(A1)
```

R code Exa 2.5.b Inverse of the given matrices

```
1 # Example : 2.5B    Chapter : 2.5    Pageno : 88  
2 # Inverse of the given matrices  
3 B<-matrix(c(4,8,3,7),ncol=2)  
4 C<-matrix(c(6,6,6,0),ncol=2)  
5 S<-matrix(c(1,1,1,0,1,1,0,0,1),ncol=3)  
6 B1<-solve(B)  
7 C1<-solve(C)
```

```

8 S1<-solve(S)
9 print("Inverses of given matrices ")
10 print(B1)
11 print(C1)
12 print(S1)

```

R code Exa 2.5.c Inverse of the Pascal matrix by gauss jordan elimination method

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
5 # for more information about pracma visit https://
  cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 2.5C Chapter : 2.5 Pageno : 88
8 # Inverse of the Pascal matrix by gauss jordan
  elimination method
9 library(pracma)
10 L<-matrix(c(1,1,1,1,0,1,2,3,0,0,1,3,0,0,0,1),ncol=4)
11 I<-eye(4)
12 LI<-cbind(L,I)
13 IL1<-rref(LI)
14 L1<-IL1[,c(5:8)]
15 print("Inverse of given Pascal matrix is")
16 print(L1)

```

R code Exa 2.5.2 Inverse of an Elimination Matrix

```

1 # Example : 2 Chapter : 2.5 Pageno : 82

```

```

2 # Inverse of an Elimination Matrix
3 E<-matrix(c(1,-5,0,0,1,0,0,0,1),ncol=3)
4 E1<-solve(E)
5 print("The inverse of the given elimination matrix
      is")
6 print(E1)

```

R code Exa 2.5.3 Inverse of product of matrices

```

1 # Example : 3    Chapter : 2.5    Pageno : 83
2 # Inverse of product of matrices
3 E<-matrix(c(1,-5,0,0,1,0,0,0,1),ncol=3)
4 E1<-solve(E)
5 F<-matrix(c(1,0,0,0,1,-4,0,0,1),ncol=3)
6 F1<-solve(F)
7 print(F1)
8 FE<-F%*%E
9 print(FE)
10 print("Inverse of FE ")
11 FE1<-solve(FE)
12 print(FE1)
13 print("Inverse of FE can also be E-1*F-1 ")
14 print(E1%*%F1)

```

R code Exa 2.5.4 Inverse of matrix by guass jordan elimination matrix

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "

```

```

5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 4    Chapter : 2.5    Pageno : 85
8 # Inverse of matrix by guass jordan elimination
   matrix
9 library(pracma)
10 A<-matrix(c(2,4,3,7),ncol=2)
11 I<-eye(2)
12 AI<-cbind(A,I)
13 R<-rref(AI)
14 X<-R[,c(3:4)]
15 print("Inverse of A is")
16 print(X)

```

R code Exa 2.5.5 Inverse of Lower traingular matrix is also a lower traingular matrix

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 5    Chapter : 2.5    Pageno : 86
8 # Inverse of Lower traingular matrix is also a lower
   traingular matrix
9 library(pracma)
10 L<-matrix(c(1,3,4,0,1,5,0,0,1),ncol=3)
11 I<-eye(3)
12 LI<-cbind(L,I)
13 R<-rref(LI)

```

```
14 L1<-R[,c(4:6)]
15 print("Inverse of L is")
16 print(L1)
```

R code Exa 2.6.a LU Factorisation

```
1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
5 # for more information about pracma visit https://
  cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 2.6A Chapter : 2.6 Pageno : 101
8 # LU factorisation
9 library(pracma)
10 P<-matrix(c(1,1,1,1,1,2,3,4,1,3,6,10,1,4,10,20),ncol
  =4)
11 print("LU factorisation of P")
12 print(lu(P))
```

R code Exa 2.6.b Forward Elimination

```
1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
5 # for more information about pracma visit https://
  cran.r-project.org/web/packages/pracma/index.html
```

```

6
7 # Example : 2.6B    Chapter : 2.6    Pageno : 102
8 # Forward Elimination
9 library(pracma)
10 P<-matrix(c(1,1,1,1,1,2,3,4,1,3,6,10,1,4,10,20),ncol
    =4)
11 b<-c(1,0,0,0)
12 L<-lu(P)$L
13 U<-lu(P)$U
14 c<-solve(L,b)
15 x<-solve(U,c)
16 print("The solution is ")
17 print(x)

```

R code Exa 2.6.1 LU Factorisation

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
5 # for more information about pracma visit https://
  cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 1    Chapter : 2.6    Pageno : 96
8 # LU Factorisation
9 library(pracma)
10 A<-matrix(c(2,1,0,1,2,1,0,1,2),ncol=3)
11 print("LU factorisation of A is")
12 print(lu(A))
13 # The answers may vary due to rounding off values

```

R code Exa 2.6.2 LU Factorisation

```
1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
5 # for more information about pracma visit https://
  cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 2 Chapter : 2.6 Pageno : 96
8 # LU Factorisation
9 library(pracma)
10 A<-matrix(c(1,1,0,0,1,2,1,0,0,1,2,1,0,0,1,2),ncol=4)
11 print("LU factorisation of A is")
12 print(lu(A))
13 # The answers may vary due to rounding off values
```

R code Exa 2.6.3 Forward Elimination

```
1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
5 # for more information about pracma visit https://
  cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 3 Chapter : 2.6 Pageno : 98
8 # Forward Elimination
9 library(pracma)
10 A<-matrix(c(1,4,2,9),ncol=2)
11 b<-c(5,21)
```

```

12 L<-lu(A)$L
13 U<-lu(A)$U
14 c<-solve(L,b)
15 x<-solve(U,c)
16 print("The solution is ")
17 print(x)
18
19 # The answers may vary due to rounding off values

```

R code Exa 2.7.a Multiplication by Permutation matrices

```

1 # Example : 2.7A Chapter : 2.7 Pageno : 114
2 # Multiplication by Permutation matrices
3 P<-matrix(c(0,0,1,1,0,0,0,1,0),ncol=3)
4 A<-matrix(c(1,4,5,4,2,6,5,6,3),ncol=3)
5 Q<-t(P)
6 PA<-P%*%A
7 PAQ<-PA%*%Q
8 print("P*A")
9 print(PA)
10 print("P*A*Q")
11 print(PAQ)

```

R code Exa 2.7.b Symmetric Factorisation

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html

```

```

6
7 # Example : 2.7B    Chapter : 2.7    Pageno : 115
8 # Symmetric Factorisation
9 library(pracma)
10 A<-matrix(c(1,4,5,4,2,6,5,6,3),ncol=3)
11 L<-lu(A)$L
12 D<-lu(A)$U
13 for(i in 1:3){
14   j<-i+1
15   while(j<=3){
16     D[i,j]<-0
17     j<-j+1
18   }
19 }
20 LT<-t(L)
21 print("Symmetric factorisation of A=LDLT")
22 print(L)
23 print(D)
24 print(LT)
25 print(L%%D%%LT)

```

R code Exa 2.7.1 Inverses and Transposes

```

1 # Example : 1    Chapter : 2.7    Pageno : 108
2 # Inverses and Transposes
3 A<-matrix(c(1,6,0,1),ncol=2)
4 AT<-t(A)
5 A1<-solve(A)
6 print("The inverse and transpose of the matrix are ")
7   )
8 print(A1)
9 print(AT)
10 print("Transpose of A-1 is ")
11 print(t(A1))
12 print("Invere of A transpose is")

```

```
12 print(solve(AT))
```

R code Exa 2.7.4 Product of matrix and its transpose gives symmetric matrix

```
1 # Example : 4    Chapter : 2.7    Pageno : 110
2 # Product of matrix and its transpose gives symmetric
  matrix
3 R<-matrix(c(-1,0,1,-1,0,1),ncol=3)
4 RT<-t(R)
5 print("R * t(R)")
6 print(R%*%RT)
7 print("t(R) * R")
8 print(RT%*%R)
```

Chapter 3

Vector Spaces and Subspaces

R code Exa 3.2.a Find the Matrix having the given special solutions

```
1 # Example : 3.2A    Chapter : 3.2    Page No: 139
2 # Find the Matrix having the given special solutions
3 s1<-c(-3,1,0,0)
4 s2<-c(-2,0,-6,1)
5 R<-diag(4)
6 R<-R[-4,]
7 #As 1st column is pivot column it is not needed to
  change
8 #As 3rd column is next pivot column , row reduced
  echolon form has 1 in second row of third column
9 R[,3]<-R[,2]
10 #Two free columns are modified acording to special
    solutions
11 R[,2]<-c(-1*s1[1],0,0)
12 R[,4]<-c(-1*s2[1],-1*s2[3],0)
13 print("matrix having given special solutions")
14 print(R)
```

R code Exa 3.2.b Special solution pivot columns free columns and reduced row echelon form of the given matrix

```
1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
5 # for more information about pracma visit https://
  cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 3.2B Chapter : 3.2 Page No: 140
8 # Find the Specialsolution , Pivotcolumns ,
  freecolumns and Reduced row echelon form for each
  given matrix
9
10 library(pracma)
11 solution <- function(A){
12   R<-rref(A)
13   m<-nrow(A)
14   n<-ncol(A)
15   pivotcol<-c() #vector to store the column numbers
    of pivot columns
16   freecol<-c() #vector to store the column numbers
    of free columns
17   i<-1
18   j<-1
19
20   # to find which columns are pivot and which are
    free
21   while(i<=m & j<=n){
22     if(R[i,j]==1){
23       pivotcol<-c(pivotcol,j)
24       i<-i+1
25       j<-j+1
26     }
27     else{
```

```

28     j<-j+1
29   }
30 }
31 y<-length(pivotcol)
32 freecol<-c(1:n)
33 freecol<-freecol[!freecol%in%pivotcol]
34 x<-length(freecol)
35 N<-c()
36 #find the basis for null space based on Row
    reduced echelon form of given matrix
37 if(y==n){
38   N<-c()
39 }
40 else{
41   for(i in 1:x){
42     temp<-c(1:n)
43     for(j in 1:x){
44       temp[freecol[j]]<-0
45     }
46     temp[freecol[i]]<-1
47     temp[freecol[i]]
48     for(j in 1:y){
49       temp[pivotcol[j]]<-R[j,freecol[i]]*-1
50     }
51     N<-c(N,temp)
52   }
53   N<-matrix(N,nrow=n,ncol=x)
54 }
55 #Basis for the nullspace of given matrix
56 print("Special solutions are given by the basis
    vectors of null space")
57 print(N)
58 print("Pivot columns are")
59 print(pivotcol)
60 print("Free columns or free variables are")
61 print(freecol)
62 print("Row reduced echelon form is")
63 print(R)

```

```

64 }
65 A1<-matrix(c(0,0,0,0,0,0,0,0),nrow=2)
66 print("For the matrix A1")
67 solution(A1)
68 A2<-matrix(c(3,1,6,2),nrow=2)
69 print("For the matrix A2")
70 solution(A2)
71 A3<-cbind(A2,A2)
72 print("For the matrix A3")
73 solution(A3)

```

R code Exa 3.2.2 Nullspace of given Singular Matrix

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 2 Chapter : 3.2 Page No: 132
8 # Find the Nullspace of given singular matrix
9
10 library(pracma)
11 nullspacebasis <- function(A){
12   R<-rref(A)
13   m<-nrow(A)
14   n<-ncol(A)
15   pivotcol<-c() #vector to store the column numbers
     of pivot columns
16   freecol<-c() #vector to store the column numbers
     of free columns
17   i<-1

```



```

18     j<-1
19
20     # to find which columns are pivot and which are
        free
21     while(i<=m & j<=n){
22         if(R[i,j]==1){
23             pivotcol<-c(pivotcol,j)
24             i<-i+1
25             j<-j+1
26         }
27         else{
28             j<-j+1
29         }
30     }
31     y<-length(pivotcol)
32     freecol<-c(1:n)
33     freecol<-freecol[!freecol%in%pivotcol]
34     x<-length(freecol)
35     N<-c()
36     #find the basis for null space based on Row
        reduced echelon form of given matrix
37     if(y==n){
38         return(N)
39     }
40     for(i in 1:x){
41         temp<-c(1:n)
42         for(j in 1:x){
43             temp[freecol[j]]<-0
44         }
45         temp[freecol[i]]<-1
46         temp[freecol[i]]
47         for(j in 1:y){
48             temp[pivotcol[j]]<-R[j,freecol[i]]*-1
49         }
50         N<-c(N,temp)
51     }
52     N<-matrix(N,nrow=n,ncol=x)
53     #Basis for the nullspace of given matrix

```

```

54   return(N)
55 }
56 A<-matrix(c(1,3,2,6),nrow=2,ncol=2)
57 N<-nullspacebasis(A)
58 print("Basis vectors for nullspace of given matrix
      is")
59 N

```

R code Exa 3.2.3 Nullspaces of given three matrices

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 3 Chapter : 3.2 Page No: 133
8 # Find the Nullspace of given matrices A,B,C
9
10 library(pracma)
11 nullspacebasis <- function(A){
12   R<-rref(A)
13   m<-nrow(A)
14   n<-ncol(A)
15   pivotcol<-c() #vector to store the column numbers
     of pivot columns
16   freecol<-c() #vector to store the column numbers
     of free columns
17   i<-1
18   j<-1
19
20   # to find which columns are pivot and which are

```

```

    free
21  while(i<=m & j<=n){
22      if(R[i,j]==1){
23          pivotcol<-c(pivotcol,j)
24          i<-i+1
25          j<-j+1
26      }
27      else{
28          j<-j+1
29      }
30  }
31  y<-length(pivotcol)
32  freecol<-c(1:n)
33  freecol<-freecol[!freecol%in%pivotcol]
34  x<-length(freecol)
35  N<-c()
36  #find the basis for null space based on Row
    reduced echelon form of given matrix
37  if(y==n){
38      return(N)
39  }
40  for(i in 1:x){
41      temp<-c(1:n)
42      for(j in 1:x){
43          temp[freecol[j]]<-0
44      }
45      temp[freecol[i]]<-1
46      temp[freecol[i]]
47      for(j in 1:y){
48          temp[pivotcol[j]]<-R[j,freecol[i]]*-1
49      }
50      N<-c(N,temp)
51  }
52  N<-matrix(N,nrow=n,ncol=x)
53  #Basis for the nullspace of given matrix
54  return(N)
55 }
56

```

```

57 A<-matrix(c(1,3,2,8),ncol=2,nrow=2)
58 N<-nullspacebasis(A)
59 print("Basis vectors of the nullspace of matrix A is
      ")
60 N
61 B<-rbind(A,2*A)
62 N1<-nullspacebasis(B)
63 print("Basis vectors of the nullspace of matrix B is
      ")
64 N1
65 C<-cbind(A,2*A)
66 C
67 N2<-nullspacebasis(C)
68 print("Basis vectors of the nullspace of matrix C is
      ")
69 N2

```

R code Exa 3.2.4 Nullspace of given upper traingular Matrix

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 4 Chapter : 3.2 Page No: 136
8 # Find the Nullspace of given Upper traingular
   matrix
9
10 library(pracma)
11 nullspacebasis <- function(A){
12   R<-rref(A)

```

```

13  m<-nrow(A)
14  n<-ncol(A)
15  pivotcol<-c() #vector to store the column numbers
      of pivot columns
16  freecol<-c() #vector to store the column numbers
      of free columns
17  i<-1
18  j<-1
19
20  # to find which columns are pivot and which are
      free
21  while(i<=m & j<=n){
22      if(R[i,j]==1){
23          pivotcol<-c(pivotcol,j)
24          i<-i+1
25          j<-j+1
26      }
27      else{
28          j<-j+1
29      }
30  }
31  y<-length(pivotcol)
32  freecol<-c(1:n)
33  freecol<-freecol[!freecol%in%pivotcol]
34  x<-length(freecol)
35  N<-c()
36  #find the basis for null space based on Row
      reduced echelon form of given matrix
37  if(y==n){
38      return(N)
39  }
40  for(i in 1:x){
41      temp<-c(1:n)
42      for(j in 1:x){
43          temp[freecol[j]]<-0
44      }
45      temp[freecol[i]]<-1
46      temp[freecol[i]]

```

```

47     for(j in 1:y){
48         temp[pivotcol[j]]<-R[j,freecol[i]]*-1
49     }
50     N<-c(N,temp)
51 }
52 N<-matrix(N,nrow=n,ncol=x)
53 #Basis for the nullspace of given matrix
54 return(N)
55 }
56
57 U<-matrix(c(1,0,5,0,7,9),nrow=2)
58 N<-nullspacebasis(U)
59 print("Basis vectors for the Nullspace of given
        upper traingular matrix is ")
60 N

```

R code Exa 3.3.a Row reduced echelon form rank and special solutions of given matrix

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 3.3A Chapter : 3.3 Page No: 149
8 # Find the row reduced echelon form, rank and
   special solution for given matrix
9
10 library(pracma)
11 solution <- function(A){
12     R<-rref(A)

```

```

13  m<-nrow(A)
14  n<-ncol(A)
15  pivotcol<-c() #vector to store the column numbers
      of pivot columns
16  freecol<-c() #vector to store the column numbers
      of free columns
17  i<-1
18  j<-1
19
20  # to find which columns are pivot and which are
      free
21  while(i<=m & j<=n){
22      if(R[i,j]==1){
23          pivotcol<-c(pivotcol,j)
24          i<-i+1
25          j<-j+1
26      }
27      else{
28          j<-j+1
29      }
30  }
31  y<-length(pivotcol)
32  freecol<-c(1:n)
33  freecol<-freecol[!freecol%in%pivotcol]
34  x<-length(freecol)
35  N<-c()
36  #find the basis for null space based on Row
      reduced echelon form of given matrix
37  if(y==n){
38      N<-c()
39  }
40  else{
41  for(i in 1:x){
42      temp<-c(1:n)
43      for(j in 1:x){
44          temp[freecol[j]]<-0
45      }
46      temp[freecol[i]]<-1

```

```

47     temp[freecol[i]]
48     for(j in 1:y){
49         temp[pivotcol[j]] <- R[j, freecol[i]] * -1
50     }
51     N <- c(N, temp)
52 }
53 N <- matrix(N, nrow=n, ncol=x)
54 }
55 print("Row reduced echelon form is")
56 print(R)
57 print("Rank of the Matrix is")
58 print(y)
59 print("Special solutions for given matrix are
        given by")
60 print(N)
61 }
62
63 A <- matrix(c(1, -1, 0, 0, -1, 2, -1, 0, 0, -1, 2, -1, 0, 0, -1, 1),
             nrow=4)
64 solution(A)

```

R code Exa 3.3.c Special solutions and row reduced echelon forms

```

1 # Packages used : pracma
2 # To install pracma, type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 3.3C Chapter : 3.3 Page No: 151
8 # Find the special solutions and row reduced echelon
   forms

```



```

9
10 library(pracma)
11 solution <- function(A){
12     R<-rref(A)
13     m<-nrow(A)
14     n<-ncol(A)
15     pivotcol<-c() #vector to store the column numbers
16                   of pivot columns
17     freecol<-c()  #vector to store the column numbers
18                   of free columns
19
20     i<-1
21     j<-1
22
23     # to find which columns are pivot and which are
24     free
25     while(i<=m & j<=n){
26         if(R[i,j]==1){
27             pivotcol<-c(pivotcol,j)
28             i<-i+1
29             j<-j+1
30         }
31         else{
32             j<-j+1
33         }
34     }
35     y<-length(pivotcol)
36     freecol<-c(1:n)
37     freecol<-freecol[!freecol%in%pivotcol]
38     x<-length(freecol)
39     N<-c()
40     #find the basis for null space based on Row
41     reduced echelon form of given matrix
42     if(y==n){
43         N<-c()
44     }
45     for(i in 1:x){
46         temp<-c(1:n)
47         for(j in 1:x){

```

```

43     temp[freecol[j]] <- 0
44   }
45   temp[freecol[i]] <- -1
46   temp[freecol[i]]
47   for(j in 1:y){
48     temp[pivotcol[j]] <- R[j, freecol[i]] * -1
49   }
50   N <- c(N, temp)
51 }
52 N <- matrix(N, nrow=n, ncol=x)
53 print("row reduced echelon form is")
54 print(R)
55 print("Special solution is ")
56 print(N)
57 }
58 #c is not equal to 4
59 A <- matrix(c(1,3,4,2,6,8,1,3,2), ncol=3)
60 solution(A)
61 #c=4
62 A4 <- matrix(c(1,3,4,2,6,8,1,3,4), ncol=3, nrow=3)
63 solution(A4)
64 #c is not equal to 0
65 B <- matrix(c(1,1,1,1), nrow=2)
66 solution(B)
67 #c = 0
68 B0 <- matrix(c(0,0,0,0), nrow=2)
69 solution(B0)

```

R code Exa 3.4.a Complete solution of the given system

```

1 # Packages used : pracma
2 # To install pracma, type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(

```

```

    pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 3.4A    Chapter : 3.4    Page No: 160
8 # Find the Complete solution of the given system
9
10 library(pracma)
11 completesolution <- function(A,b){
12     R<-rref(A)
13     m<-nrow(A)
14     n<-ncol(A)
15     pivotcol<-c() #vector to store the column numbers
        of pivot columns
16     freecol<-c() #vector to store the column numbers
        of free columns
17     i<-1
18     j<-1
19
20     # to find which columns are pivot and which are
        free
21     while(i<=m & j<=n){
22         if(R[i,j]==1){
23             pivotcol<-c(pivotcol,j)
24             i<-i+1
25             j<-j+1
26         }
27         else{
28             j<-j+1
29         }
30     }
31     y<-length(pivotcol)
32     freecol<-c(1:n)
33     freecol<-freecol[!freecol%in%pivotcol]
34     x<-length(freecol)
35     N<-c()
36     #find the basis for null space based on Row
        reduced echelon form of given matrix

```

```

37   if(y==n){
38       N<-c()
39   }
40   for(i in 1:x){
41       temp<-c(1:n)
42       for(j in 1:x){
43           temp[freecol[j]]<-0
44       }
45       temp[freecol[i]]<-1
46       temp[freecol[i]]
47       for(j in 1:y){
48           temp[pivotcol[j]]<-R[j,freecol[i]]*-1
49       }
50       N<-c(N,temp)
51   }
52   N<-matrix(N,nrow=n,ncol=x)
53   s<-N
54   Ab<-cbind(A,b)
55   Rd<-rref(Ab)
56   temp<-Rd[,n+1]
57   p<-c(1:n)
58   for(i in 1:length(freecol)){
59       p[freecol[i]]<-0
60   }
61   for(i in 1:length(pivotcol)){
62       p[pivotcol[i]]<-temp[i]
63   }
64   print("The special solution is")
65   print(s)
66   print("The particular solution is")
67   print(p)
68   print("Complete solution = Particular solution +
        Special solution")
69 }
70 A<-matrix(c(1,2,3,2,4,6,3,8,7,5,12,13),nrow=3)
71 b<-c(0,6,-6)
72 completesolution(A,b)

```

R code Exa 3.4.c Complete solution of the given system

```
1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 3.4C Chapter : 3.4 Page No: 162
8 # Find the Complete solution of the given system
9
10 library(pracma)
11 completesolution <- function(A,b){
12   R<-rref(A)
13   m<-nrow(A)
14   n<-ncol(A)
15   pivotcol<-c() #vector to store the column numbers
     of pivot columns
16   freecol<-c() #vector to store the column numbers
     of free columns
17   i<-1
18   j<-1
19
20   # to find which columns are pivot and which are
     free
21   while(i<=m & j<=n){
22     if(R[i,j]==1){
23       pivotcol<-c(pivotcol,j)
24       i<-i+1
25       j<-j+1
26     }
```

```

27     else{
28         j<-j+1
29     }
30 }
31 y<-length(pivotcol)
32 freecol<-c(1:n)
33 freecol<-freecol[!freecol%in%pivotcol]
34 x<-length(freecol)
35 N<-c()
36 #find the basis for null space based on Row
    reduced echelon form of given matrix
37 if(y==n){
38     N<-c()
39 }
40 for(i in 1:x){
41     temp<-c(1:n)
42     for(j in 1:x){
43         temp[freecol[j]]<-0
44     }
45     temp[freecol[i]]<-1
46     temp[freecol[i]]
47     for(j in 1:y){
48         temp[pivotcol[j]]<-R[j,freecol[i]]*-1
49     }
50     N<-c(N,temp)
51 }
52 N<-matrix(N,nrow=n,ncol=x)
53 s<-N
54 Ab<-cbind(A,b)
55 Rd<-rref(Ab)
56 temp<-Rd[,n+1]
57 p<-c(1:n)
58 for(i in 1:length(freecol)){
59     p[freecol[i]]<-0
60 }
61 for(i in 1:length(pivotcol)){
62     p[pivotcol[i]]<-temp[i]
63 }

```

```

64  print("The special solution is")
65  print(s)
66  print("The particular solution is")
67  print(p)
68  print("Complete solution = particular solution +
      Special solution")
69  }
70  A<-matrix(c(1,2,4,2,4,8,1,4,6,0,8,8),nrow=3)
71  b<-c(4,2,10)
72  completesolution(A,b)

```

R code Exa 3.4.2 particular solution and special solution

```

1  # Packages used : pracma
2  # To install pracma,type following in command line
   while connected to internet
3  # install.packages("pracma")
4  # package can be included by command " library(
   pracma) "
5  # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7  # Example : 2 Chapter : 3.4 Page No: 158
8  # Find the particular solution and special solution
   of the given system
9
10 library(pracma)
11 nullspacebasis <- function(A){
12   R<-rref(A)
13   m<-nrow(A)
14   n<-ncol(A)
15   pivotcol<-c() #vector to store the column numbers
   of pivot columns
16   freecol<-c() #vector to store the column numbers
   of free columns

```

```

17     i<-1
18     j<-1
19
20     # to find which columns are pivot and which are
        free
21     while(i<=m & j<=n){
22         if(R[i,j]==1){
23             pivotcol<-c(pivotcol,j)
24             i<-i+1
25             j<-j+1
26         }
27         else{
28             j<-j+1
29         }
30     }
31     y<-length(pivotcol)
32     freecol<-c(1:n)
33     freecol<-freecol[!freecol%in%pivotcol]
34     x<-length(freecol)
35     N<-c()
36     #find the basis for null space based on Row
        reduced echelon form of given matrix
37     if(y==n){
38         return(N)
39     }
40     for(i in 1:x){
41         temp<-c(1:n)
42         for(j in 1:x){
43             temp[freecol[j]]<-0
44         }
45         temp[freecol[i]]<-1
46         temp[freecol[i]]
47         for(j in 1:y){
48             temp[pivotcol[j]]<-R[j,freecol[i]]*-1
49         }
50         N<-c(N,temp)
51     }
52     N<-matrix(N,nrow=n,ncol=x)

```



```

53   #Basis for the nullspace of given matrix
54   return(N)
55 }
56 A<-matrix(c(1,1,1,2,1,-1),nrow=2)
57 b<-c(3,4)
58 s<-nullspacebasis(A)
59 Ab<-cbind(A,b)
60 Rd<-rref(Ab)
61 p<-Rd[,4]
62 p<-c(p,0)
63 print("Particular solution is")
64 print(p)
65 print("special solution is ")
66 print(s)
67 print("The complete solution = particular solution +
      special solution")

```

R code Exa 3.5.1 Columns of given matrix are linearly dependent

```

1  # Packages used : pracma
2  # To install pracma,type following in command line
   while connected to internet
3  # install.packages("pracma")
4  # package can be included by command " library(
   pracma) "
5  # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7  # Example : 1 Chapter : 3.5 Page No: 170
8  # Columns of given matrix are dependent
9  library(pracma)
10 A<-matrix(c(1,2,1,0,1,0,3,5,3),nrow=3)
11 if(Rank(A)!=ncol(A)){
12   print("The columns of A are dependent")
13 }

```

```

14 print("As the columns of A are dependent ,there is a
      nonzero solution to  $Ax=0$ ")
15 x<-matrix(c(-3,1,1),ncol=1)
16 print(A%*%x)

```

R code Exa 3.6.a Four Fundamental Spaces of given matrix

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
5 # for more information about pracma visit https://
  cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 3.6A Chapter : 3.6 Page No: 190
8 # Four Fundamental Spaces of given matrix
9
10 library(pracma)
11 nullspacebasis <- function(A){
12   R<-rref(A)
13   m<-nrow(A)
14   n<-ncol(A)
15   pivotcol<-c() #vector to store the column numbers
    of pivot columns
16   freecol<-c() #vector to store the column numbers
    of free columns
17   i<-1
18   j<-1
19
20   # to find which columns are pivot and which are
    free
21   while(i<=m & j<=n){
22     if(R[i,j]==1){

```

```

23     pivotcol<-c(pivotcol,j)
24     i<-i+1
25     j<-j+1
26 }
27 else{
28     j<-j+1
29 }
30 }
31 y<-length(pivotcol)
32 freecol<-c(1:n)
33 freecol<-freecol[!freecol%in%pivotcol]
34 x<-length(freecol)
35 N<-c()
36 #find the basis for null space based on Row
    reduced echelon form of given matrix
37 if(y==n){
38     return(N)
39 }
40 for(i in 1:x){
41     temp<-c(1:n)
42     for(j in 1:x){
43         temp[freecol[j]]<-0
44     }
45     temp[freecol[i]]<-1
46     temp[freecol[i]]
47     for(j in 1:y){
48         temp[pivotcol[j]]<-R[j,freecol[i]]*-1
49     }
50     N<-c(N,temp)
51 }
52 N<-matrix(N,nrow=n,ncol=x)
53 #Basis for the nullspace of given matrix
54 return(N)
55 }
56 l<-matrix(c(1,2,5,0,1,0,0,0,1),ncol=3)
57 u<-matrix(c(1,0,0,3,0,0,0,1,0,5,6,0),ncol=4)
58 A<-l%*%u
59 print("Row space Basis of A")

```

```

60 print(u[1,])
61 print(u[2,])
62 print("Column space Basis of A ")
63 print(l[,1])
64 print(l[,2])
65 print("Null space Basis of A")
66 N<-nullspacebasis(A)
67 print(N)
68 print("Null space Basis of A transpose")
69 NT<-nullspacebasis(t(A))
70 print(NT)

```

R code Exa 3.6.1 Dimensions and rank of matrix

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 1 Chapter : 3.6 Page No: 188
8 # Dimensions and rank of matrix
9 library(pracma)
10 solution<-function(A){
11   print(paste("Number of rows , m=",nrow(A)))
12   print(paste("Number of columns ,n=",ncol(A)))
13   print(paste("Rank of the given matrix",Rank(A)))
14 }
15 A<-matrix(c(1,2,3),nrow=1)
16 solution(A)

```

R code Exa 3.6.2 Dimensions and rank of matrix

```
1 # Packages used : pracma
2 # To install pracma, type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 2 Chapter : 3.6 Page No: 188
8 # Dimensions and rank of matrix
9 library(pracma)
10 solution<-function(A){
11   print(paste("Number of rows , m=",nrow(A)))
12   print(paste("Number of columns ,n=",ncol(A)))
13   print(paste("Rank of the given matrix",Rank(A)))
14 }
15 A<-matrix(c(1,2,2,4,3,6),nrow=2)
16 solution(A)
```

Chapter 4

Orthogonality

R code Exa 4.1.a Dimensions of the subspaces in the given space

```
1 # Example : 4.1A    Chapter : 4.1    Page No: 201
2 # Dimensions of the subspaces in the given space
3 dim_R<-9
4 dim_S<-6
5 print("Possible dimensions of the subspaces
      orthogonal to S")
6 x<-dim_R-dim_S
7 orthogonal_dimensions<-c(0:x)
8 print(orthogonal_dimensions)
9 print(paste("possible dimensions of orthogonal
      complement subspaces to S",dim_R-dim_S))
10 print(paste("The smallest matrix A in S is ",dim_S,"
      by ",dim_R))
11 print(paste("The Null space matrix N is ",dim_R," by
      ",dim_R-dim_S))
```

R code Exa 4.1.b Null space Basis of a plane subspace

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 4.1B Chapter : 4.1 Page No: 201
8 # Null space Basis of a plane subspace
9 library(pracma)
10 nullspacebasis <- function(A){
11   R<-rref(A)
12   m<-nrow(A)
13   n<-ncol(A)
14   pivotcol<-c() #vector to store the column numbers
      of pivot columns
15   freecol<-c() #vector to store the column numbers
      of free columns
16   i<-1
17   j<-1
18
19   # to find which columns are pivot and which are
      free
20   while(i<=m & j<=n){
21     if(R[i,j]==1){
22       pivotcol<-c(pivotcol,j)
23       i<-i+1
24       j<-j+1
25     }
26     else{
27       j<-j+1
28     }
29   }
30   y<-length(pivotcol)
31   freecol<-c(1:n)
32   freecol<-freecol[!freecol%in%pivotcol]

```

```

33  x<-length(freecol)
34  N<-c()
35  #find the basis for null space based on Row
    reduced echelon form of given matrix
36  if(y==n){
37    return(N)
38  }
39  for(i in 1:x){
40    temp<-c(1:n)
41    for(j in 1:x){
42      temp[freecol[j]]<-0
43    }
44    temp[freecol[i]]<-1
45    temp[freecol[i]]
46    for(j in 1:y){
47      temp[pivotcol[j]]<-R[j,freecol[i]]*-1
48    }
49    N<-c(N,temp)
50  }
51  N<-matrix(N,nrow=n,ncol=x)
52  #Basis for the nullspace of given matrix
53  return(N)
54 }
55
56 A<-matrix(c(1,-3,-4),ncol=3)
57 print("Given Plane  $x-3y-4z=0$  is a null space of
    following 1*3 matrix")
58 print(A)
59 #to make matrix copatible for our function
60 A<-rbind(A,c(0,0,0),c(0,0,0))
61 N<-nullspacebasis(A)
62 print("SPecial solutions or nullspace basis of given
    plane subspace is")
63 print(N)
64 A<-A[-c(2,3),]
65 print("Row space is ")
66 print(A)
67 temp<-cbind(N,c(1,-3,-4))

```



```

68 x<-c(1,1,-1)
69 print("vector 6,4,5 is split into vn + vs as 1 of
      each vector in nullspace basis and -1 of row space
      basis")
70 v<-temp%*%x
71 print(v)

```

R code Exa 4.1.3 Rows of matrix are perpendicular to the vectors in nullspace

```

1 # Example : 3    Chapter : 4.1    Page No: 197
2 # Rows of given matrix are perpendicular to the
  vector in nullspace
3 A<-matrix(c(1,5,3,2,4,7),nrow=2)
4 x<-c(1,1,-1)
5 for(i in 1:nrow(A)){
6   dot_product<-sum(A[i,]*x)
7   if(dot_product==0){
8     print(paste("Row ",i," is perpendicular to x"))
9   }
10 }

```

R code Exa 4.2.a Projection onto the line and the plane

```

1 # Example : 4.2A    Chapter : 4.2    Page No: 213
2 # Projection onto the line and onto the plane
3
4 projection_line<-function(a,b){
5   p<-((sum(a*b))/(sum(a*a)))*a
6   e<-b-p
7   print("The projection vector is ")
8   print(p)
9   print("The error vector is ")
10  print(e)

```

```

11 }
12
13 projection_plane<-function(A,b){
14   b<-matrix(c(b),ncol=1)
15   ATA<-t(A)%*%A
16   ATA1<-solve(ATA)
17   P<-A%*%ATA1
18   P<-P%*%t(A)
19   p<-P%*%b
20   e<-b-p
21   print("The projection vector is ")
22   print(p)
23   print("The error vector is ")
24   print(e)
25 }
26
27 b<-c(3,4,4)
28 a<-c(2,2,1)
29 projection_line(a,b)
30 A<-matrix(c(a,1,0,0),ncol=2)
31 projection_plane(A,b)
32 #The answer may slightly vary due to rounding off
    values

```

R code Exa 4.2.b Best possible solution

```

1 # Example : 4.2B Chapter : 4.2 Page No: 213
2 # Find best Possible solution
3
4 solution<-function(A,b){
5   ATA<-t(A)%*%A
6   ATb<-t(A)%*%b
7   x<-solve(ATA,ATb)
8   return(x)
9 }

```

```

10
11 A<-matrix(c(1,1,1),ncol=1)
12 b<-matrix(c(70,80,120),ncol=1)
13 x<-solution(A,b)
14 print("The best possible solution is ")
15 print(x)

```

R code Exa 4.2.1 Projection of vector onto line

```

1 # Example : 1 Chapter : 4.2 Page No: 208
2 # Projection of the vector onto line
3
4 #Answers to this problem are displayed in the form
  of x/y in textbook
5 #Here the same answers are in decimal formats
6
7 projection<-function(b,a){
8   xhat<-(sum(a*b))/(sum(a*a))
9   p<-xhat*a
10  return(p)
11 }
12 b<-c(1,1,1)
13 a<-c(1,2,2)
14 p<-projection(b,a)
15 print("The projection vector p i.e., b on a is ")
16 print(p)
17 e<-b-p
18 print("The error vector is ")
19 print(e)
20 if(sum(e*a)==0){
21   print("Vector e is perpendicular to a")
22 }
23 #The answer may slightly vary due to rounding off
  values

```

R code Exa 4.2.2 Projection Matrix of the line

```
1 # Example : 2    Chapter : 4.2    Page No: 209
2 # Find the projection matrix onto the line
3
4 projection_matrix<-function(a){
5   a<-matrix(c(a),ncol=1)
6   P<-a%*%t(a)
7   temp<-t(a)%*%a
8   temp<-1/temp
9   t<-temp[1,1]
10  P<-t*P
11  return(P)
12 }
13 a<-c(1,2,2)
14 P<-projection_matrix(a)
15 print("The projection matrix is ")
16 print(P)
17 #The answer may slightly vary due to rounding off
    values
```

R code Exa 4.2.3 Best possible solution projection vector and Projection Matrix

```
1 # Example : 3    Chapter : 4.2    Page No: 211
2 # Find the best possible solution , projection vector
    and projection matrix
3
4 solution<-function(A,b){
5   ATA<-t(A)%*%A
6   b<-matrix(c(b),ncol=1)
7   ATb<-t(A)%*%b
```

```

8   xhat<-solve(ATA,ATb)
9   p<-A%%xhat
10  e<-b-p
11  ATA1<-solve(ATA)
12  P<-A%%ATA1
13  P<-P%%t(A)
14  print("The best possible solution is ")
15  print(xhat)
16  print("The projection vector is ")
17  print(p)
18  print("The error vector e is")
19  print(e)
20  print("The projection matrix is ")
21  print(P)
22 }
23
24 A<-matrix(c(1,1,1,0,1,2),ncol=2)
25 b<-c(6,0,0)
26 solution(A,b)
27 #The answer may slightly vary due to rounding off
    values

```

R code Exa 4.3.a Fit a straight line

```

1  # Example : 4.3A   Chapter : 4.3   Page No: 225
2  # Fit a straight line
3
4  #1,2,3 solutions can be done without any need of
    computation
5  solution<-function(A,b){
6    ATA<-t(A)%%A
7    ATb<-t(A)%%b
8    xhat<-solve(ATA,ATb)
9    return(xhat)
10 }

```

```

11 fit_line<-function(D){
12   num_of_points<-nrow(D)
13   t<-c()
14   for(i in 1:num_of_points){
15     t<-c(t,1)
16   }
17   t<-c(t,D[,1])
18   A<-matrix(c(t),ncol=2)
19   b<-D[,2]
20   b<-matrix(c(b),ncol=1)
21   x<-solution(A,b)# The system has no solution , we
                     need to find the best solution
22   return(x)
23 }
24 t<-c(1:10)
25 b<-c(0,0,0,0,0,0,0,0,0,40)
26 Data<-matrix(c(t,b),ncol=2)
27 x<-fit_line(Data)
28 C<-x[1]
29 D<-x[2]
30 print(paste("The best straight line is b= ",C," + ",
              D," t"))
31 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 4.3.b Fit a Parabola

```

1 # Example : 4.3B    Chapter : 4.3    Page No: 226
2 # Fit a Parabola
3 solution<-function(A,b){
4   ATA<-t(A)%*%A
5   ATb<-t(A)%*%b
6   xhat<-solve(ATA,ATb)
7   return(xhat)
8 }

```

```

9  fit_parabola<-function(D){
10    num_of_points<-nrow(D)
11    t<-c()
12    for(i in 1:num_of_points){
13      t<-c(t,1)
14    }
15    t<-c(t,D[,1])
16    t<-c(t,D[,1]*D[,1])
17    A<-matrix(c(t),ncol=3)
18    b<-D[,2]
19    b<-matrix(c(b),ncol=1)
20    x<-solution(A,b)# The system has no solution , we
                     need to find the best solution
21    return(x)
22  }
23  t<-c(-2,-1,0,1,2)
24  b<-c(0,0,1,0,0)
25  Data<-matrix(c(t,b),ncol=2)
26  x<-fit_parabola(Data)
27  C<-x[1]
28  D<-x[2]
29  E<-x[3]
30  print(paste("The best Parabola that fitt in is b= ",
               C,"+",D,"t+",E,"t2"))
31  #The answer may slightly vary due to rounding off
    values.

```

R code Exa 4.3.1 Fit a straight line

```

1  # Example : 1    Chapter : 4.3    Page No: 218
2  # Fit a straight line
3  solution<-function(A,b){
4    ATA<-t(A)%*%A
5    ATb<-t(A)%*%b
6    xhat<-solve(ATA,ATb)

```

```

7   return(xhat)
8 }
9 fit_line<-function(D){
10  num_of_points<-nrow(D)
11  t<-c()
12  for(i in 1:num_of_points){
13    t<-c(t,1)
14  }
15  t<-c(t,D[,1])
16  A<-matrix(c(t),ncol=2)
17  b<-D[,2]
18  b<-matrix(c(b),ncol=1)
19  x<-solution(A,b)# The system has no solution , we
    need to find the best solution
20  return(x)
21 }
22 Data<-matrix(c(0,6,1,0,2,0),ncol=2,byrow=T)
23 x<-fit_line(Data)
24 C<-x[1]
25 D<-x[2]
26 print(paste("The best straight line is b= ",C," + ",
    D," t"))

```

R code Exa 4.3.2 Fit a straight line

```

1 # Example : 2      Chapter : 4.3      Page No: 222
2 # Fit a straight line
3 solution<-function(A,b){
4   ATA<-t(A)%*%A
5   ATb<-t(A)%*%b
6   xhat<-solve(ATA,ATb)
7   return(xhat)
8 }
9 fit_line<-function(D){
10  num_of_points<-nrow(D)

```



```

11   t<-c()
12   for(i in 1:num_of_points){
13     t<-c(t,1)
14   }
15   t<-c(t,D[,1])
16   A<-matrix(c(t),ncol=2)
17   b<-D[,2]
18   b<-matrix(c(b),ncol=1)
19   x<-solution(A,b)# The system has no solution , we
                     need to find the best solution
20   return(x)
21 }
22 Data<-matrix(c(-2,0,2,1,2,4),ncol=2)
23 x<-fit_line(Data)
24 C<-x[1]
25 D<-x[2]
26 print(paste("The best straight line is b= ",C," + ",
              D,"t"))
27 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 4.3.3 Fit a Parabola

```

1 # Example : 3    Chapter : 4.3    Page No: 224
2 # Fit a Parabola
3 solution<-function(A,b){
4   ATA<-t(A)%*%A
5   ATb<-t(A)%*%b
6   xhat<-solve(ATA,ATb)
7   return(xhat)
8 }
9 fit_parabola<-function(D){
10   num_of_points<-nrow(D)
11   t<-c()
12   for(i in 1:num_of_points){

```

```

13     t<-c(t,1)
14 }
15 t<-c(t,D[,1])
16 t<-c(t,D[,1]*D[,1])
17 A<-matrix(c(t),ncol=3)
18 b<-D[,2]
19 b<-matrix(c(b),ncol=1)
20 x<-solution(A,b)# The system has no solution , we
    need to find the best solution
21 return(x)
22 }
23 Data<-matrix(c(0,1,2,6,0,0),ncol=2)
24 x<-fit_parabola(Data)
25 C<-x[1]
26 D<-x[2]
27 E<-x[3]
28 print(paste("The best Parabola that fitt in is b= ",
    C,"+",D,"t+",E,"t2"))

```

R code Exa 4.4.4 Projections of vector onto the line plane if the basis are given as orthonormal vectors

```

1 # Example : 4    Chapter : 4.4    Page No: 233
2 # Projections of the vector onto line ,plane if basis
    are given as orthonormal vectors
3 Q<-matrix(c(-1,2,2,2,-1,2,2,2,-1),ncol=3)
4 Q<-(1/3)*Q
5 q1<-Q[,1]
6 q2<-Q[,2]
7 q3<-Q[,3]
8 b<-c(0,0,1)
9 p1<-sum(q1*b)*q1
10 p2<-sum(q2*b)*q2
11 p3<-sum(q3*b)*q3
12 print("Projection of b onto q1")

```

```

13 print(p1)
14 print("Projection of b onto q2")
15 print(p2)
16 print("Projection of b onto q3")
17 print(p3)
18 print("Projection of b onto plane of q1 and q2")
19 print(p1+p2)
20 print("Projection of b onto space of q1,q2, and q3")
21 print(p1+p2+p3) # same as vector b

```

R code Exa 4.4.5 Gram Schmidt method to convert matrix to its orthogonal form

```

1 # Example : 5    Chapter : 4.4    Page No: 233
2 # Gram-Schmidt method to convert matrix into its
  orthogonal form
3 magnitude<-function(a){
4   x<-0
5   for(i in 1:length(a)){
6     x<-x+a[i]*a[i]
7   }
8   x<-sqrt(x)
9   return(x)
10 }
11 a<-c(1,-1,0)
12 b<-c(2,0,-2)
13 c<-c(3,-3,3)
14 A<-a
15 B<-b-(sum(A*b)/sum(A*A))*A
16 C<-c-(sum(A*c)/sum(A*A))*A-(sum(B*c)/sum(B*B))*B
17 print("Orthogonal vectors corresponding to a,b,c are
    ")
18 print(A)
19 print(B)
20 print(C)

```

```
21 q1<-(1/magnitude(A))*A
22 q2<-(1/magnitude(B))*B
23 q3<-(1/magnitude(C))*C
24 Q<-matrix(c(q1,q2,q3),ncol=3)
25 print("Orthogonal matrix with orthonormal vectors of
      a,b,c ")
26 print(Q)
27 #The answer may slightly vary due to rounding off
    values
```

Chapter 5

Determinants

R code Exa 5.2.1 Determinant of matrix by Product of pivots

```
1 # Example : 1    Chapter : 5.2    Page No: 255
2 # Determinant of matrices by multiplying pivots
3 A<-matrix(c(0,0,4,0,2,5,1,3,6),ncol=3)
4 P<-matrix(c(0,0,1,0,1,0,1,0,0),ncol=3)
5 PA<-P%*%A
6 print(PA)
7 detA<-1*PA[1,1]*PA[2,2]*PA[3,3]
8 print(detA)
9 det(A)
10 print("detA is the determinant of the given matrix")
```

R code Exa 5.2.5 Determinants of matrices

```
1 # Example : 5    Chapter : 5.2    Page No: 259
2 # Determinants of matrices
3 A4<-matrix(c(0,1,0,0,1,0,1,0,0,1,0,0,1,0),ncol
  =4)
4 P4<-matrix(c(0,1,0,0,1,0,0,0,0,0,0,0,1,0),ncol
  =4)
```

```

5 print(det(A4))
6 print(det(P4))

```

R code Exa 5.2.7 Determinants of matrices

```

1 # Example : 7    Chapter : 5.2    Page No: 261
2 # Determinants of matrices
3 B4<-matrix(c(1,-1,0,0,-1,2,-1,0,0,-1,2,-1,0,0,-1,2),
4           ncol=4)
5 print(det(B4))

```

R code Exa 5.3.a Null space of matrix is the transpose of cofactor matrix

```

1 # Example : 5.3A    Chapter : 5.3    Page No: 277
2 # Nullspace of matrix as transpose of Cofactor
  matrix
3 nullspacebasis<-function(A){
4   C<-matrix(c(1:9),ncol=3)
5   for(i in 1:3){
6     for(j in 1:3){
7       if((i+j)%2==0){
8         x<-1
9       }
10      else{
11        x<--1
12      }
13      C[i,j]<-x*det(A[-i,-j])
14    }
15  }
16  C<-t(C)
17  return(C)
18 }
19

```

```

20 A1<-matrix(c(1,2,2,4,3,2,7,9,8),ncol=3)
21 N1<-nullspacebasis(A1)
22 print("The null space basis are given by columns of
      transpose of cofactor matrix")
23 print("Null space of A1 is")
24 print(N1)
25 A2<-matrix(c(1,1,1,1,1,1,2,1,1),ncol=3)
26 N2<-nullspacebasis(A2)
27 print("Null space of A2 is ")
28 print(N2)

```

R code Exa 5.3.b Solve by Crammers rule and inverse of matrix

```

1 # Example : 5.3B Chapter : 5.3 Page No: 278
2 # Solve by crammers rule and inverse of the matrix
3
4 solve_by_crammersrule<-function(A,b){
5   B1<-A
6   B2<-A
7   B3<-A
8   B1[,1]<-b
9   B2[,2]<-b
10  B3[,3]<-b
11  x1<-det(B1)/det(A)
12  x2<-det(B2)/det(A)
13  x3<-det(B3)/det(A)
14  x<-c(x1,x2,x3)
15  return(x)
16 }
17 inverse<-function(A){
18   C<-matrix(c(1:9),ncol=3)
19   for(i in 1:3){
20     for(j in 1:3){
21       if((i+j)%2==0){
22         x<-1

```

```

23     }
24     else{
25         x<--1
26     }
27     C[i,j]<-x*det(A[-i,-j])
28 }
29 }
30 CT<-t(C)
31 A1<-(1/det(A))*CT
32 return(A1)
33 }
34
35 A<-matrix(c(2,1,5,6,4,9,2,2,0),ncol=3)
36 b<-c(0,0,1)
37 x<-solve_by_crammersrule(A,b)
38 print("x is ")
39 print(x)
40 A1<-inverse(A)
41 print("Inverse of A is ")
42 print(A1)
43 print("A * inverse of A is Identity matrix")
44 I<-A%%A1
45 print(I)
46 #The answer may slightly vary due to rounding off
    values

```

R code Exa 5.3.1 Crammers rule for solving system of equations

```

1 # Example : 1    Chapter : 5.3    Page No: 269
2 # Cramers rule for solving system of equations
3 A<-matrix(c(3,5,4,6),ncol=2)
4 b<-c(2,4)
5 B1<-A
6 B1[,1]<-b
7 B2<-A

```



```

8 B2[,2] <-b
9 x1<-det(B1)/det(A)
10 x2<-det(B2)/det(A)
11 print("SOlution is ")
12 print(x1)
13 print(x2)

```

R code Exa 5.3.3 The inverse of triangular matrix is triangular matrix

```

1 # Example : 3    Chapter : 5.3    Page No: 271
2 # The inverse of the traingular matrix is traingular
3 A<-matrix(c(1,1,1,1,0,1,1,1,0,0,1,1,0,0,0,1),ncol=4)
4 A1<-solve(A)
5 print("The inverse of A is")
6 print(A1)

```

R code Exa 5.3.7 Cross product of vectors

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
   while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
   pracma) "
5 # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 7    Chapter : 5.3    Page No: 275
8 # CrossProduct of Vectors
9 library(pracma)
10 u<-c(3,2,0)
11 v<-c(1,4,0)
12 cp<-cross(u,v)

```

```
13 print("u * v is ")
14 print(cp)
```

R code Exa 5.3.8 Cross product of vectors

```
1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
5 # for more information about pracma visit https://
  cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 8 Chapter : 5.3 Page No: 276
8 # CrossProduct of Vectors
9
10 library(pracma)
11 u<-c(1,1,1)
12 v<-c(1,1,2)
13 cp<-cross(u,v)
14 print("u * v is ")
15 print(cp)
```

R code Exa 5.3.9 Right hand rule

```
1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
```

```
5 # for more information about pracma visit https://  
   cran.r-project.org/web/packages/pracma/index.html  
6  
7 # Example : 9    Chapter : 5.3    Page No: 276  
8 # The right hand rule – cross product of x-axis and  
   y-axis is z-axis  
9  
10 library(pracma)  
11 u<-c(1,0,0)  
12 v<-c(0,1,0)  
13 cp<-cross(u,v)  
14 print("u * v is ")  
15 print(cp)
```

Chapter 6

Eigen Values and Eigen Vectors

R code Exa 6.1.a Eigen values and eigen vectors

```
1 # Example : 6.1A Chapter : 6.1 Page No: 291
2 # Eigen values and eigen vectors
3 solution<-function(A){
4   sol<-eigen(A)
5   lambda<-sol$values
6   x<-sol$vectors #these are normalised eigen vectors
7   #to get the eigen vectors as in texxt book
8   multiply these normalised vectors with scalars
9   x[,1]<-x[,1]*(1/x[1,1])
10  x[,2]<-x[,2]*(1/x[1,2])
11  print("The eigen values of the matrix are")
12  print(lambda)
13  print("The eigen vecotrs of the matrix respective
14  to above eigen values are")
15  print(x)
16 }
17
18 A<-matrix(c(2,-1,-1,2),ncol=2)
19 print("For A")
20 solution(A)
21 A2<-A%*%A
```

```

20 print("For square of A")
21 solution(A2)
22 A1<-solve(A)
23 print("For inverse of A")
24 solution(A1)
25 I<-matrix(c(1,0,0,1),ncol=2)
26 A4I<-A+4*I
27 print("For A+4I")
28 solution(A4I)
29 #The answer may slightly vary due to rounding off
    values
30 #The answers provided in the text book may vary
    because of the computation process
31 #Both answers are correct , here it is taken -Ax+b=0
    , In the text book it is considered as Ax-b=0

```

R code Exa 6.1.b Eigen values and eigen vectors

```

1 # Example : 6.1B Chapter : 6.1 Page No: 292
2 # Eigen values and eigen vectors
3 solution<-function(A){
4   sol<-eigen(A)
5   lambda<-round(sol$values)
6   x<-sol$vectors #these are normalised eigen vectors
7   #to get eigen vectors in text book multiply
    normalised eigen vectors with scalars
8   x[,1]<-x[,1]*(1/x[1,1])
9   x[,2]<-round(x[,2]*(1/x[1,2]))
10  x[,3]<-x[,3]*(1/x[1,3])
11  print("The eigen values of the matrix are")
12  print(lambda)
13  print("The eigen vectors of the matrix respective
    to above eigen values are")
14  print(x)
15 }

```

```

16
17 A<-matrix(c(1,-1,0,-1,2,-1,0,-1,1),ncol=3)
18 solution(A)
19 #The answer may slightly vary due to rounding off
    values

```

R code Exa 6.1.1 Eigen values and eigen vectors

```

1 # Example : 1    Chapter : 6.1    Page No: 284
2 # Eigen values and eigen vectors
3 A<-matrix(c(0.8,0.2,0.3,0.7),ncol=2)
4 sol<-eigen(A)
5 lambda<-sol$values
6 x<-sol$vectors
7 print("The eigen values of the matrix are")
8 print(lambda)
9 print("The eigen vectors of the matrix in normalised
    form are")
10 print(x)
11 #to get eigen vectors in the textbook multiply
    normalised vectors by scalars
12 x[,1]<-x[,1]*(0.6/x[1,1])
13 x[,2]<-x[,2]*(1/x[1,2])
14 print("Eigen vectors with respect to the above eigen
    values respectively are")
15 print(x)
16 print(A%*%x)
17 #The answer may slightly vary due to rounding off
    values
18 #The answers provided in the text book may vary
    because of the computation process
19 #Both answers are correct , here it is taken -Ax+b=0
    , In the text book it is considered as Ax-b=0

```

R code Exa 6.1.2 Eigen values and eigen vectors of Projection matrix

```
1 # Example : 2    Chapter : 6.1    Page No: 285
2 # Eigen values and eigen vectors of Projection
  matrix
3 A<-matrix(c(0.5,0.5,0.5,0.5),ncol=2)
4 sol<-eigen(A)
5 lambda<-sol$values
6 x<-sol$vectors #These are normalised eigen vectors
7 #to get eigen vectors in text book multiply them
  with scalars
8 x[,1]<-x[,1]*(1/x[1,1])
9 x[,2]<-x[,2]*(1/x[1,2])
10 print("The eigen values of the matrix are")
11 print(lambda)
12 print("The eigen vectors of the matrix are")
13 print(x)
```

R code Exa 6.1.3 Eigen values and eigen vectors of Reflection matrix

```
1 # Example : 3    Chapter : 6.1    Page No: 286
2 # Eigen values and eigen vectors of Reflection
  matrix
3 A<-matrix(c(0,1,1,0),ncol=2)
4 sol<-eigen(A)
5 lambda<-sol$values
6 print("The eigen values of the matrix are")
7 print(lambda)
8 #The answer may slightly vary due to rounding off
  values
```

R code Exa 6.1.4 Eigen values and eigen vectors of Singular matrix

```
1 # Packages used : pracma
2 # To install pracma,type following in command line
  while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
  pracma) "
5 # for more information about pracma visit https://
  cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 4 Chapter : 6.1 Page No: 287
8 # Eigen values and eigen vectors of Singular matrix
9
10 library(pracma)
11 nullspacebasis <- function(A){
12   R<-rref(A)
13   m<-nrow(A)
14   n<-ncol(A)
15   pivotcol<-c() #vector to store the column numbers
    of pivot columns
16   freecol<-c() #vector to store the column numbers
    of free columns
17   i<-1
18   j<-1
19
20   # to find which columns are pivot and which are
    free
21   while(i<=m & j<=n){
22     if(R[i,j]==1){
23       pivotcol<-c(pivotcol,j)
24       i<-i+1
25       j<-j+1
26     }
```



```

27     else{
28         j<-j+1
29     }
30 }
31 y<-length(pivotcol)
32 freecol<-c(1:n)
33 freecol<-freecol[!freecol%in%pivotcol]
34 x<-length(freecol)
35 N<-c()
36 #find the basis for null space based on Row
    reduced echelon form of given matrix
37 if(y==n){
38     return(N)
39 }
40 for(i in 1:x){
41     temp<-c(1:n)
42     for(j in 1:x){
43         temp[freecol[j]]<-0
44     }
45     temp[freecol[i]]<-1
46     temp[freecol[i]]
47     for(j in 1:y){
48         temp[pivotcol[j]]<-R[j,freecol[i]]*-1
49     }
50     N<-c(N,temp)
51 }
52 N<-matrix(N,nrow=n,ncol=x)
53 #Basis for the nullspace of given matrix
54 return(N)
55 }
56 A<-matrix(c(1,2,2,4),ncol=2)
57 sol<-eigen(A)
58 print(sol)
59 lambda<-sol$values
60 print("The eigen values of the matrix are")
61 print(lambda)
62 I<-matrix(c(1,0,0,1),ncol=2)
63 E1<-A-lambda[1]*I

```

```

64 E1
65 rref(E1)
66 x1<-nullspacebasis(E1)
67 E2<-A-lambda[2]*I
68 rref(E2)
69 x2<-nullspacebasis(E2)
70 print("The eigen vectors of the matrix in normalized
      form are")
71 print(x1)
72 print(x2)
73 #to get eigen vectors in the textbook multiply
      normalised vectors by scalars
74 x1<-2*x1
75 x2<-1*x2
76 print("The eigen vectors of the matrix are ")
77 print(x2)
78 print(x1)
79 #The answer may slightly vary due to rounding off
      values
80 #The answers provided in the text book may vary
      because of the computation process
81 #Both answers are correct , here it is taken -Ax+b=0
      , In the text book it is considered as Ax-b=0

```

R code Exa 6.2.b Inverse Eigen values Determinant and eigen vector matrix

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
      while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
      pracma) "
5 # for more information about pracma visit https://
      cran.r-project.org/web/packages/pracma/index.html

```

```

6
7 # Example : 6.2B Chapter : 6.2 Page No: 306
8 # Inverse Eigen values Determinant and eigen vector
  matrix
9 library(pracma)
10 A<-5*eye(4)-ones(4)
11 eigenvalues<-eigen(A)$values
12 print("Eigen Values of A ")
13 print(eigenvalues)
14 A1<-solve(A)
15 print("Inverse of A ")
16 print(A1)
17 eigenvalues1<-eigen(A1)$value
18 print("Eigen Values of A inverse ")
19 print(eigenvalues1)
20 detA<-det(A)
21 print("Determinant of A")
22 print(detA)
23 x<-eigen(A)$vectors[,4] #normalized eigen vector of
  eigen value=1
24 #to get eigen vector in text book
25 x<-x*(1/x[1])
26 print("Eigen vector of A for eigen value 1")
27 print(x)
28 print("The other eigen vectors are perpendicular to
  x since A is Symmetric so eigen vector matrix
  contains x with different signs as follows")
29 S<-matrix(c(1,-1,1,-1,1,1,-1,-1,1,-1,-1,1,x),ncol=4)
30 print(S)
31 print("To get normalized matrix multiply by
  magnitude of vectors which is same for all and is
  0.5")
32 S<-0.5*S
33 print(S) # The eigen vectors are with respect to
  5,5,5,1

```

R code Exa 6.2.1 Diagonalizing a Matrix and Power of matrix computed from power of its diagonal matrix

```

1 # Example : 1    Chapter : 6.2    Page No: 299
2 # Diagonalizing a Matrix and Power of matrix
  computed from power of its diagonal matrix
3 A<-matrix(c(1,0,5,6),ncol=2)
4 lambda<-eigen(A)$values
5 S<-eigen(A)$vectors
6 S<-round(S)
7 S1<-solve(S)
8 Diag_matrix<-round(S1%*%A%*%S)
9 print("Diagonal Matrix is ")
10 print(Diag_matrix)
11 A2<-A%*%A
12 print("The square of matrix")
13 print(A2)
14 print("The Power of matrix can also be computed from
  its diagonal matrix as follows")
15 A2_diag<-S%*%Diag_matrix%*%Diag_matrix%*%S1
16 print("The square of matrix computed from its
  diagonal matrix")
17 print(A2_diag)
18 #The answer may slightly vary due to rounding off
  values
19 #The answers provided in the text book may vary
  because of the computation process
20 #Both answers are correct , here it is taken -Ax+b=0
  , In the text book it is considered as Ax-b=0

```

R code Exa 6.2.2 The diagonal matrix of any matrix contains the eigen values in its main diagonal

```

1 # Example : 2    Chapter : 6.2    Page No: 300
2 # The diagonal matrix of any matrix contains the
   eigen values in its main diagonal
3 A<-matrix(c(0.8,0.2,0.3,0.7),ncol=2)
4 lambda<-eigen(A)$values
5 print(lambda)
6 S<-eigen(A)$vectors #Normlised eigen vectors , Answer
   can also be validated with normalised eigen
   vectors
7 #to get eigen vector matrix in text book
8 S[,1]<-S[,1]*(0.6/S[1,1])
9 S[,2]<-S[,2]*(1/S[1,2])
10 S1<-solve(S)
11 Diag_matrix<-diag(2)*lambda
12 print("S*diag_matrix(A)*S-1 is A")
13 print(S%*%Diag_matrix%*%S1)
14 #The answer may slightly vary due to rounding off
   values
15 #The answers provided in the text book may vary
   because of the computation process
16 #Both answers are correct , here it is taken -Ax+b=0
   , In the text book it is considered as Ax-b=0

```

R code Exa 6.3.b Eigen values and eigen vectors

```

1 # Example : 6.3B    Chapter : 6.3    Page No: 322
2 # Eigen Values and Eigen vectors of A
3 A<-matrix(c(-2,1,0,1,-2,1,0,1,-2),ncol=3)
4 eigenvalues<-eigen(A)$values
5 x<-eigen(A)$vectors
6 print("Eigen values of A are ")
7 print(eigenvalues)
8 print("Eigen vectors of A in normalised form")
9 print(x)
10 #to get eigen vectors in the textbook multiply

```

```

        normalised vectors by scalars
11 x[,1]<-x[,1]*(1/x[1,1])
12 x[,2]<-x[,2]*(1/x[1,2])
13 x[,3]<-x[,3]*(1/x[1,3])
14 print("Eigen vectors with respect to the above eigen
        values respectively are")
15 print(x)
16 #The answer may slightly vary due to rounding off
        values
17 #The answers provided in the text book may vary
        because of the computation process
18 #Both answers are correct , here it is taken -Ax+b=0
        , In the text book it is considered as Ax-b=0

```

R code Exa 6.3.1 Solve Differential equation

```

1 # Example : 1      Chapter : 6.3      Page No: 313
2 # Solve Differential equation
3
4 # lambda1,lamda2,x1,x2,c,d are computed here.. for
        remaining details look textbook
5 A<-matrix(c(0,1,1,0),ncol=2)
6 lambda<-eigen(A)$values
7 x<-eigen(A)$vectors #These are normalised eigen
        vectors
8 #to get eigen vectors in textbook .. Multiply them
        with the scalars
9 x[,1]<-x[,1]*(1/x[1,1])
10 x[,2]<-x[,2]*(1/x[1,2])
11 print(x)
12 u<-c(4,2)
13 cd<-solve(x,u)
14 C<-cd[1]
15 D<-cd[2]
16 print("Lambda 1 and Lambda 2")

```

```

17 print(lambda)
18 print("x1 and x2")
19 print(x)
20 print("C and D are")
21 print(C)
22 print(D)

```

R code Exa 6.3.2 Solve Differential equation

```

1 # Example : 2 Chapter : 6.3 Page No: 314
2 # Solve Differential equation
3
4 # lambda1, lamda2, lambda3, x1, x2, x3, c1, c2, c3 are
   computed here.. for remaining details look
   textbook
5 A<-matrix(c(1,0,0,1,2,0,1,1,3),ncol=3)
6 lambda<-eigen(A)$values
7 x<-round(eigen(A)$vectors)
8 u<-c(9,7,4)
9 c<-solve(x,u)
10 print("Lambda 3 and Lambda 2 and Lambda 1 are")
11 print(lambda)
12 print("x3 and x2 and x1")
13 print(x)
14 print("c3, c2 and c1 are")
15 print(c)
16 #The answer may slightly vary due to rounding off
   values
17 #The answers provided in the text book may vary
   because of the computation process
18 #Both answers are correct , here it is taken -Ax+b=0
   , In the text book it is considered as Ax-b=0

```

R code Exa 6.3.6 Solve Differential equation

```
1 # Example : 6 Chapter : 6.3 Page No: 321
2 # Solve Differential equation
3
4 # lambda1, lamda2, x1, x2, c, d are computed here.. for
   remaining details look textbook
5 A<-matrix(c(1,0,1,2),ncol=2)
6 lambda<-eigen(A)$values
7 x<-round(eigen(A)$vectors)
8 u<-c(2,1)
9 c<-solve(x,u)
10 print("Lambda 1 and Lambda 2")
11 print(lambda)
12 print("x1 and x2")
13 print(x)
14 print("c1 and c2 are")
15 print(c)
16 #The answer may slightly vary due to rounding off
   values
17 #The answers provided in the text book may vary
   because of the computation process
18 #Both answers are correct , here it is taken -Ax+b=0
   , In the text book it is considered as Ax-b=0
```

R code Exa 6.4.b Eigen values and eigen vectors

```
1 # Example : 6.4B Chapter : 6.4 Page No: 336
2 # Eigen Values and Eigen vectors of given matrices
3 A3<-matrix(c(2,-1,0,-1,2,-1,0,-1,2),ncol=3)
4 eigenvalues<-eigen(A3)$values
5 x<-eigen(A3)$vectors
6 print("Eigen values of A3 are ")
7 print(eigenvalues)
8 print("Eigen vectors of A3 in normalised form are ")
```



```

9  print(x)
10 #to get eigen vectors in the textbook multiply
    normalised vectors by scalars
11 x[,1]<-x[,1]*(1/x[1,1])
12 x[,2]<-x[,2]*(sqrt(2)/x[1,2])
13 x[,3]<-x[,3]*(1/x[1,3])
14 print("Eigen vectors with respect to the above eigen
    values respectively are")
15 print(x)
16 B4<-matrix(c(1,-1,0,0,-1,2,-1,0,0,-1,2,-1,0,0,-1,1),
    ncol=4)
17 B4eigenvalues<-eigen(B4)$values
18 B4x<-eigen(B4)$vectors
19 print("Eigen values of B4 are ")
20 print(B4eigenvalues)
21 print("Eigen vectors of B4 in normalised form are")
22 print(B4x)
23 #to get eigen vectors in the textbook multiply
    normalised vectors by scalars
24 B4x[,1]<-B4x[,1]*(1/B4x[1,1])
25 B4x[,2]<-B4x[,2]*(1/B4x[1,2])
26 B4x[,3]<-B4x[,3]*(1/B4x[1,3])
27 B4x[,4]<-B4x[,4]*(1/B4x[1,4])
28 print("Eigen vectors with respect to the above eigen
    values respectively are")
29 print(B4x)
30 #The answer may slightly vary due to rounding off
    values
31 #The answers provided in the text book may vary
    because of the computation process
32 #Both answers are correct , here it is taken -Ax+b=0
    , In the text book it is considered as Ax-b=0

```

R code Exa 6.4.1 Eigen values and eigen vectors

```

1 # Example : 1    Chapter : 6.4    Page No: 331
2 # Eigen Values and Eigen vectors of A
3 A<-matrix(c(1,2,2,4),ncol=2)
4 lambda<-eigen(A)$values
5 x<-eigen(A)$vectors #These are Normalised eigen
    vectors
6 #to get eigen vectors in textbook
7 x[,1]<-x[,1]*(1/x[1,1])
8 x[,2]<-x[,2]*(2/x[1,2])
9 print("Eigen values and eigen vectors of respective
    eigen values of A")
10 print(lambda)
11 print(x)
12 #The answer may slightly vary due to rounding off
    values
13 #The answers provided in the text book may vary
    because of the computation process
14 #Both answers are correct , here it is taken -Ax+b=0
    , In the text book it is considered as Ax-b=0

```

R code Exa 6.4.4 pivots and eigen values have same signs for symmetric matrices

```

1 # Packages used : pracma
2 # To install pracma,type following in command line
    while connected to internet
3 # install.packages("pracma")
4 # package can be included by command " library(
    pracma) "
5 # for more information about pracma visit https://
    cran.r-project.org/web/packages/pracma/index.html
6
7 # Example : 4    Chapter : 6.4    Page No: 334
8 # Pivots and Eigen values have same signs for the
    Symmetric matrices

```

```

9  library(pracma)
10 A<-matrix(c(1,3,3,1),ncol=2)
11 u<-lu(A)$U
12 pivots<-c(u[1,1],u[2,2])
13 eigenvalues<-eigen(A)$values
14 print("Eigen values and pivots have same signs for
      symmetric matrix")
15 print(pivots)
16 print(eigenvalues)
17 print("This is not true if the matrix is not
      symmetric")
18 B<-matrix(c(1,-1,6,-4),ncol=2)
19 Bu<-lu(B)$U
20 Bpivots<-c(Bu[1,1],Bu[2,2])
21 Beigenvalues<-eigen(B)$values
22 print(Bpivots)
23 print(Beigenvalues)

```

R code Exa 6.5.1 Tests for positive definiteness

```

1  # Packages used : pracma
2  # To install pracma,type following in command line
   while connected to internet
3  # install.packages("pracma")
4  # package can be included by command " library(
   pracma) "
5  # for more information about pracma visit https://
   cran.r-project.org/web/packages/pracma/index.html
6
7  # Example : 1 Chapter : 6.5 Page No: 344
8  # Tests for positive definitenes
9  library(pracma)
10 A<-matrix(c(2,-1,0,-1,2,-1,0,-1,2),ncol=3)
11 u<-lu(A)$U
12 pivots<-c(u[1,1],u[2,2],u[3,3])

```

```

13 upper_left_deteminants<-c(A[1,1],det(A[-3,-3]),det(A
    ))
14 eigenvalues<-eigen(A)$values
15 print("All pivots , upper left determinants and eigen
    values are positive")
16 print(pivots)
17 print(upper_left_deteminants)
18 print(eigenvalues)

```

R code Exa 6.6.a Pascals Matrix and its inverse are similar

```

1 # Example : 3    Chapter : 6.6    Page No: 357
2 # Pascals matrix and its inverse are similar
3 A<-matrix(c(1,1,1,1,0,1,2,3,0,0,1,3,0,0,0,1),ncol=4)
4 A1<-solve(A)
5 Aev<-eigen(A)$values
6 A1ev<-eigen(A1)$values
7 print("Both pascal matrix and its inverse have same
    eigen values")
8 print(Aev)
9 print(A1ev)

```

R code Exa 6.6.1 Similar matrices are matrices with same eigen values

```

1 # Example : 1    Chapter : 6.6    Page No: 356
2 # Similar matrices are matrices with same eigen
    values
3 A<-matrix(c(0.5,0.5,0.5,0.5),ncol=2)
4 Aev<-eigen(A)$values
5 print("Eigen values of A ")
6 print(Aev)
7 Lambda<-matrix(c(1,0,0,0),ncol=2)
8 Lev<-eigen(Lambda)$values

```

```

9 print("eigen values of lambda matrix")
10 print(Lev)
11 M<-matrix(c(1,1,0,2),ncol=2)
12 M1<-solve(M)
13 M1AM<-M1%*%A%*%M
14 M1AMev<-eigen(M1AM)$values
15 print("Eigen values of M-1*A*M")
16 print(M1AMev)
17 print("A and M-1*A*M are similar matrices")

```

R code Exa 6.6.2 Similar matrices with repeated eigen values

```

1 # Example : 2 Chapter : 6.6 Page No: 356
2 # Similar matrices with repeated eigen values
3 A<-matrix(c(0,0,1,0),ncol=2)
4 Aev<-eigen(A)$values
5 A1<-matrix(c(1,1,-1,-1),ncol=2)
6 A1ev<-round(eigen(A1)$values)
7 print("Both the given matrices are similar because
      their eigen values are same")
8 print(Aev)
9 print(A1ev)
10 #The answer may slightly vary due to rounding off
    values

```

R code Exa 6.6.3 Jordans Theorem and jordan Matrix

```

1 # Example : 3 Chapter : 6.6 Page No: 357
2 # Jordans theorem and Jordan Matrix
3 J<-matrix(c(5,0,0,1,5,0,0,1,5),ncol=3)
4 Jev<-eigen(J)$values
5 JT<-t(J)
6 JTev<-eigen(JT)$values

```

```
7 print("Jordans theorem says that both J and
   transpose of J are similar")
8 print(Jev)
9 print(JTev)
```

R code Exa 6.7.3 Singular Value Decomposition

```
1 # Example : 3    Chapter : 6.7    Page No: 366
2 # Singular Value decomposition
3
4 A<-matrix(c(2,-1,2,1),ncol=2)
5 print("Singular value decomposition is given by")
6 print(svd(A))
7 #The answer may slightly vary due to rounding off
  values
8 #The answers provided in the text book may vary
  because of the computation method followed.
```

R code Exa 6.7.4 Singular Value Decomposition

```
1 # Example : 3    Chapter : 6.7    Page No: 366
2 # Singular Value decomposition
3
4 A<-matrix(c(2,1,2,1),ncol=2)
5 print("Singular value decomposition is given by")
6 print(svd(A))
7 #The answer may slightly vary due to rounding off
  values
8 #The answers provided in the text book may vary
  because of the computation method followed.
```

Chapter 7

Linear Transformations

R code Exa 7.3.a leftinverse rightinverse Pseduoinverse of given matrices

```
1 # Example : 7.3A    Chapter : 7.3    Page No: 405
2 # leftinverse ,rightinverse ,Pseduoinverse of given
   matrices
3 A1<-matrix(c(2,2),ncol=1)
4 A2<-matrix(c(2,2),ncol=2)
5 A3<-matrix(c(2,2,2,2),ncol=2)
6 A1T<-t(A1)
7 A2T<-t(A2)
8 A1inv<-solve(A1T*%A1)*%A1T
9 print("Left inverse of A1")
10 print(A1inv)
11 print(A1inv*%A1)
12 A2inv<-A2T*%solve(A2*%A2T)
13 print("right inverse of A2")
14 print(A2inv)
15 print(A2*%A2inv)
16
17 #The answers given in the text book is wrong it is 1
   /8 .. not 1/sqrt(8)
18 V1<-svd(A3)$v
19 U1T<-t(svd(A3)$u)
```

```

20 d<-svd(A3)$d
21 sigma1<-matrix(c(1/d[1],0,0,0),ncol=2)
22 A3inv<-2*(V1%*%sigma1%*%U1T)
23 print("The Pseduo inverse of the given matrix")
24 print(A3inv)

```

R code Exa 7.3.1 Diagonalization of matrix

```

1 # Example : 1 Chapter : 7.3 Page No: 400
2 # Diagonaliazation of matrix
3 A<-matrix(c(0.5,-0.5,-0.5,0.5),ncol=2)
4 ev<-eigen(A)$values
5 D<-matrix(c(ev[1],0,0,ev[2]),ncol=2)
6 print("The diagonalized matrix")
7 print(D)

```

R code Exa 7.3.2 Similar Projection Matrices

```

1 # Example : 2 Chapter : 7.3 Page No: 401
2 # Similar Projection Matrices
3 A<-matrix(c(0.5,-0.5,-0.5,0.5),ncol=2)
4 Aev<-eigen(A)$values
5 W<-matrix(c(2,0,1,1),ncol=2)
6 W1<-solve(W)
7 B<-W1%*%A%*%W
8 print("Matrix B = W-1 * A * W")
9 print(B)
10 Bev<-eigen(B)$values
11 print("A and B are similar matrices")
12 print(Aev)
13 print(Bev)

```

R code Exa 7.3.3 Polar Decomposition

```
1 # Example : 3    Chapter : 7.3    Page No: 402
2 # Polar Decomposition
3 A<-matrix(c(2,-1,2,1),ncol=2)
4 Q<-round(svd(A)$u)%*%t(svd(A)$v)
5 H<-t(Q)%*%A
6 print("Polar Decomposition A=QH")
7 print("Q is ")
8 print(Q)
9 print("H is ")
10 print(H)
11 #The answer may slightly vary due to rounding off
    values
12 #The answers provided in the text book may vary
    because of the computation method followed.
```

R code Exa 7.3.4 Pseduo Inverse of a matrix

```
1 # Example : 4    Chapter : 7.3    Page No: 404
2 # Pseduoinverse of matrix
3 A<-matrix(c(2,1,2,1),ncol=2)
4 V<-svd(A)$v
5 UT<-t(svd(A)$u)
6 d<-svd(A)$d
7 sigma1<-matrix(c(1/d[1],0,0,0),ncol=2)
8 A1<-V%*%sigma1%*%UT
9 print("The Pseduo inverse of the given matrix")
10 print(A1)
11 #The answer may slightly vary due to rounding off
    values
```

- 12 #The answers provided in the text book may vary
because of the computation method followed.
-

Chapter 8

Applications

R code Exa 8.1.1 Movements tensions elongations of spring

```
1 # Example : 1    Chapter : 8.1    Page No: 413
2 # Find movements , tensions , elongations of spring
3 K<-matrix(c(2,-1,0,-1,2,-1,0,-1,2),ncol=3)
4 K1<-solve(K)
5 f<-c(1,1,1)#since all mi=m
6 u<-K1%*%f
7 print("Movements are given by mg/c * u and u is")
8 print(u)
9 A<-matrix(c(1,-1,0,0,0,1,-1,0,0,0,1,-1),ncol=3)
10 e<-A%*%u
11 print("Elongations are given by mg/c * e and e is")
12 print(e)
13 print("Tensions are given by mg * e and e is")
14 print(e)
```

R code Exa 8.1.2 Movements tensions elongations of spring

```
1 # Example : 2    Chapter : 8.1    Page No: 414
```

```

2 # Find movements , tensions , elongations of spring
3 K1<-matrix(c(2,-1,0,-1,2,-1,0,-1,1),ncol=3)
4 K11<-solve(K1)
5 f<-c(1,1,1)
6 u<-K11%*%f
7 print("Movements are given by mg/c * u and u is")
8 print(u)
9 A<-matrix(c(1,-1,0,0,1,-1,0,0,1),ncol=3)
10 e<-A%*%u
11 print("Elongations are given by mg/c * e and e is")
12 print(e)
13 y<-solve(t(A))%*%f
14 print("Tensions are given by mg * y and y is")
15 print(y)

```

R code Exa 8.2.1 Find the currents

```

1 # Example : 1 Chapter : 8.2 Page No: 427
2 # Find the currents
3 A<-matrix(c
      (-1,-1,0,-1,0,0,1,0,-1,0,-1,0,0,1,1,0,0,-1,0,0,0,1,1,1)
      ,ncol=4)
4 AT<-t(A)
5 Laplacian_matrix<-AT%*%A
6 Laplacian_matrix<-Laplacian_matrix[-4,-4]
7 b<-c(1,0,0)
8 x<-solve(Laplacian_matrix,b)
9 x<-c(x,0)
10 print(" Voltages are given by S * ")
11 print(x)
12 print("S is source")
13 y<-1*A%*%x
14 print("Currents are given by S * ")
15 print(y)

```

R code Exa 8.3.1 Positive Markov matrix application

```
1 # Example : 1    Chapter : 8.3    Page No: 432
2 # Positive Markov matrix Application.
3 A<-matrix(c(0.8,0.2,0.05,0.95),ncol=2)
4 u0<-c(0.02,0.98)
5 u1<-A%*%u0
6 print(u1)
7 u2<-A%*%u1
8 print(u2)
```

R code Exa 8.3.3 Markov matrix application

```
1 # Example : 3    Chapter : 8.3    Page No: 433
2 # Markov matrix Application.
3 A<-matrix(c(0,0.5,0.5,0.5,0,0.5,0.5,0.5,0),ncol=3)
4 u0<-c(8,16,32)
5 u1<-A%*%u0
6 u2<-A%*%u1
7 u3<-A%*%u2
8 print("Populations in three groups in subsequent
      months")
9 print(u1)
10 print(u2)
11 print(u3)
```

R code Exa 8.3.4 Linear Algebra in economy

```
1 # Example : 4    Chapter : 8.3    Page No: 437
```

```

2 # Linear Algebra in Economy
3 A<-matrix(c(0.2,0.4,0.5,0.3,0.4,0.1,0.4,0.1,0.3),
           ncol=3)
4 lambda<-eigen(A)$values
5 print("Lambda max is ")
6 print(lambda[1])
7 I<-matrix(c(1,0,0,0,1,0,0,0,1),ncol=3)
8 A1<-solve(I-A)
9 print(A1)
10 #The answers may vary due to rounding off values

```

R code Exa 8.3.5 Linear Algebra in economy

```

1 # Example : 5    Chapter : 8.3    Page No: 437
2 # Linear Algebra in Economy
3 A<-matrix(c(0,1,4,0),ncol=2)
4 lambda<-eigen(A)$values
5 print("Lambda max is ")
6 print(lambda[1])
7 I<-matrix(c(1,0,0,1),ncol=2)
8 A1<-solve(I-A)
9 print(A1)
10 #The answers may vary due to rounding off values

```

R code Exa 8.5.2 length of function sinx and sinx and cosx are orthogonal in function space

```

1 #integrate is a function from package stats which is
   included in R by default
2 #if not install package stats
3 # Example : 2    Chapter : 8.5    Page No: 448
4 # length of function sinx and sinx and cosx are
   orthogonal in function space

```

```

5 f1<-function(x){
6   return (sin(x)*sin(x))
7 }
8 f2<-function(x){
9   return (sin(x)*cos(x))
10 }
11 print("Inner product of sinx and sinx is ")
12 lsin2x<-integrate(f1,0,2*pi)
13 print(lsin2x)
14 print("Length of sinx is square root of above inner
    product")
15 print(sqrt(lsin2x$value))
16 print("Cos x and sinx are orthogonal in functional
    space because their inner product equals zero")
17 lsincosx<-integrate(f2,0,2*pi)
18 print(lsincosx)
19 #The answer may slightly vary due to rounding off
    values

```

R code Exa 8.6.2 Singular Value Decomposition

```

1 # Example : 6    Chapter : 8.6    Page No: 457
2 # Singular Value Decomposition
3 gradematrixA<-matrix(c
    (-6,0,4,2,2,4,0,-6,0,-6,2,4,4,2,-6,0),ncol=4)
4 print("SVD of Grade matrix is")
5 print(svd(gradematrixA))
6 #The answers may vary due to rounding off values

```

Chapter 9

Numerical Linear Algebra

R code Exa 9.2.2 Norm of a diagonal matrix

```
1 # Example : 2    Chapter : 9.2    Page No: 475
2 # Norm of diagonal matrix
3 A<-matrix(c(2,0,0,3),ncol=2)
4 print("The norm of the diagonal matrix is its
      largest entry")
5 print(A[2,2])
6 print("Largest eigen value and norm are equal for
      this matrix")
7 lambdamax<-eigen(A)$values[1]
8 print(lambdamax)
9 print("Eigen vectors of this matrix are")
10 ev<-round(eigen(A)$vectors)
11 print(ev)
```

R code Exa 9.2.3 Condition number of Positive definite Matrix

```
1 # Example : 3    Chapter : 9.2    Page No: 478
2 # Condition number for positive definite matrix
```



```

3 A<-matrix(c(6,0,0,2),ncol=2)
4 print("Condition number for positive definite matrix
      is max.eigen value/min.eign value")
5 condition_number=eigen(A)$values[1]/eigen(A)$values
      [2]
6 print("Condition number for A")
7 print(condition_number)

```

R code Exa 9.3.1 Powers of some matrices can be calculated easily with max eigen value

```

1 # Example : 1      Chapter : 9.3      Page No: 482
2 # Powers of some matrices can be calculated easily
      with max eigen value
3 B<-matrix(c(0.6,0.6,0.5,0.5),ncol=2)
4 B1<-matrix(c(0.6,0,1.1,0.5),ncol=2)
5 lambda<-eigen(B)$values
6 lambda1<-eigen(B1)$values
7 lmax<-lambda[1]
8 l1max<-lambda1[1]
9 print("Lambda max of B is")
10 print(lmax)
11 print("Lambda max of B1 is ")
12 print(l1max)
13 B2<-B%*%B
14 print("B square is")
15 print(B2)
16 B2<-lmax*B
17 print(B2)
18 print("B1 square has 0.6*0.6 and 0.5*0.5 on its
      diagonal")
19 B12<-B1%*%B1
20 print(B12)

```

Chapter 10

Complex numbers and Matrices

R code Exa 10.1.1 r for complex numbers

```
1 # Example : 1    Chapter : 10.1    Page No: 496
2 # r of complex numbers
3 rad2deg <- function(rad) {
4   (rad * 180) / (pi)
5 }
6 z<-complex(real=1,imaginary=1)
7 z1<-Conj(z)
8 print("r of z and its conjugate are")
9 print(Mod(z))
10 print(Mod(z1))
11 print("The argument of z and its conjugate in
    degrees are")
12 print(paste(rad2deg(Arg(z)),"degrees")) # in radians
    which is equal to 45 degree
13 print(paste(rad2deg(Arg(z1)),"degrees"))
```

R code Exa 10.2.1 Orthogonal Complex vectors

```

1 # Example : 1    Chapter : 10.2    Page No: 502
2 # Orthogonal Complex vectors
3 i<-complex(real=0,imaginary=1)
4 u<-matrix(c(1,i),ncol=1)
5 ut<-t(u)
6 #take conjugate of ut
7 ut[1,2]<-Conj(ut[1,2])
8 v<-matrix(c(i,1),ncol=1)
9 print("inner product of u and v is conj(u transpose )
      * v is ")
10 innerproduct<-ut%*%v
11 print(innerproduct)
12 print("As innerproduct is zero , they are orthogonal
      complex vectors")

```
