R Textbook Companion for Linear Algebra and Its Applications by David C. Lay[1]

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

May 26, 2020

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

Book Description

Title: Linear Algebra and Its Applications

Author: David C. Lay

Publisher: Pearson Addison-Wesley

Edition: 3

Year: 2006

ISBN: 0-321-28066-0

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

Contents

List of R Codes

Chapter 1

Linear Equations in Linear Algebra

R code Exa 1.1 1

```
#Chapter 1 - Linear Equations In Linear Algebra
#Gaussian Elimination
#Page No.15 / 1-3
#Prob 1
#1.1.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('performing Gaussian elimination')
av<-c(1,5,-2,-7)
a=matrix(av,
         nrow=2,
         ncol=2,
         byrow=TRUE)
print('the co-efficient matrix is:')
print(a)
bv<-c(7,-5)
b=matrix(
  bv,
  nrow=2,
  ncol=1,
  byrow=TRUE)
cm=cbind(a,b)
print('the augmented matrix is:')
print(cm)
print('R2=R2+2*R1')
cm[2,]=cm[2,]+2*cm[1,]
print(cm)
print('R2=(1/3)*R2')
cm[2,]=(1/3)*cm[2,]
print(cm)
print('R1=R1-5*R2')
cm[1,]=cm[1,]-5*cm[2,]
print(cm)
x1=cm[1,3]/cm[1,1]
x2=cm[2,3]/cm[2,2]
cat('the solution is: x1 =',x1,' x2 =',x2)
```

R code Exa 1.2 2

```
#Chapter 1 - Linear Equations In Linear Algebra
#Gaussian Elimination Singular case
#Page No.15 / 1-3
#Prob 7
#1.1.7

#clear console
cat("\014")

#clear variables
```

```
rm(list=ls(all=TRUE))

print('the augmented matrix is:')
av<-c(1,7,3,-4,0,1,-1,3,0,0,0,1,0,0,1,-2)
a=matrix(
  av,
  nrow=4,
  ncol=4,
  byrow=TRUE)
print(a)
print('interchange R3 and R4')
a[c(3,4),]=a[c(4,3),]
print(a)
print('from R4 we get 0=1')
print('hence, no solution')
```

R code Exa 1.3 3

```
#Chapter 1 - Linear Equations In Linear Algebra
#Gaussian Elimination with row exchange
#Page No.15 / 1-3
#Prob 13
#1.1.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the augmented matrix is')
av<-c(1,0,-3,8,2,2,9,7,0,1,5,-2)
a=matrix(
  av,
  nrow=3,
  ncol=4,
  byrow=TRUE)
print(a)
print('R2=R2-2*R1')
a[2,]=a[2,]-2*a[1,]
print(a)
print('interchange R2 and R3')
a[c(2,3),]=a[c(3,2),]
print(a)
print('R3=R3-2*R2')
a[3,]=a[3,]-2*a[2,]
print(a)
print('R3=(1/5)*R3')
a[3,]=(1/5)*a[3,]
print(a)
print('R2=R2-5*R3 and R1=R1+3*R3')
a[2,]=a[2,]-5*a[3,]
a[1,]=a[1,]+3*a[3,]
print(a)
s=matrix(
  c(a[1,4],a[2,4],a[3,4]),
  nrow=3,
  ncol=1,
  byrow=TRUE)
print('solution is')
print(s)
```

R code Exa 1.4 4

```
#Chapter 1 - Linear Equations In Linear Algebra
#Condition for a solution to exist
#Page No.16 / 1-4
#Prob 19
#1.1.19

```

```
#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the augmented matrix for h=2’)
av<-c(1,2,4,3,6,8)
a=matrix(av,
         nrow=2,
         ncol=3,
         byrow=TRUE)
print(a)
print(’R2−2*R1’)
a[2,]=a[2,]-3*a[1,]
print(a)
print(’from R3 we get 0=−4’)
print(’hence, if h=2 no solution, else solution exists’)
```

R code Exa 1.5 5

```
#Chapter 1 − Linear Equations In Linear Algebra
#Consistent System
#Page No.16 / 1−4
#Prob 25
#1.1.25

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the co−efficient matrix is:’)
```

```
av<-c(1,-4,7,0,3,-5,-2,5,-9)
a=matrix(av,
         nrow = 3,
         ncol = 3,
         byrow = TRUE)
print(a)
print('let g,h,k be the constants on RHS')
print('R3=R3+2*R1')
a[3,]=a[3,]+2*a[1,]
print(a)
print('the constants on RHS are:g,h,k+2g')
print('R3=R3+R2')
a[3,]=a[3,]+a[2,]
print(a)
print('the constants on RHS are:g,h,k+2g+h')
print('for solution to exist')
print('from R3:k+2g+h=0')
```

R code Exa 1.6 6

```
#Chapter 1 - Linear Equations In Linear Algebra
#General solution of the system
#Page No.20 / 1-8
#Prob 7
#1.2.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the augmented matrix is')
av<-c(1,3,4,7,3,9,7,6)
a=matrix(av,
         nrow = 2,
         ncol=4,
         byrow=TRUE)
print(a)
print('R2=R2-3*R1')
a[2,]=a[2,]-3*a[1,]
print(a)
print('(-1/5)*R2')
a[2,]=(-1/5)*a[2,]
print(a)
print('R1=R1-4*R2')
a[1,]=a[1,]-4*a[2,]
print('the row reduced form is:')
print(a)
print('corresponding equations are')
print('x1+3*x2=-5 and x3=3')
print('free variables:x2')
print('general solution is:')
print('x1=-5-3*x2, x2, x3=3')
```

R code Exa 1.7 7

```
#Chapter 1 - Linear Equations In Linear Algebra
#General solution of the system
#Page No.20 / 1-8
#Prob 13
#1.2.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the augmented matrix is')
```

```
av<-c(1,-3,0,-1,0,-2,0,1,0,0,-4,1,0,0,0,1,9,4,0,0,0,0,0,0)
a=matrix(av,
        nrow=4,
        ncol=6,
        byrow=TRUE)
print(a)
print('R1=R1+R3')
a[1,]=a[1,]+a[3,]
print(a)
print('R1=R1+3*R2')
a[1,]=a[1,]+3*a[2,]
print(a)
print('corresponding equations are:')
print('x1-3*x5=5, x2-4*x5=1, x4+9*x5=4, and 0=0')
print('free variables:x3, x5')
print('general solution is:')
print('x1=5+3*x5, x2=1+4*x5, x3, x4=4-9*x5, x5')
```

R code Exa 1.8 8

```
#Chapter 1 - Linear Equations In Linear Algebra
#Linear Combination of two vectors
#Page No.23 / 1-11
#Prob 1
#1.3.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

uv<-c(-1,2)
```

```
u=matrix(uv,
         nrow=2,
         ncol=1,
         byrow=TRUE)
print(’u=’)
print(u)
vv<-c(-3,-1)
v=matrix(vv,
         nrow=2,
         ncol=1,
         byrow=TRUE)
print(’v=’)
print(v)
a=u+v
print(’u+v=’)
print(a)
s=u-2*v
print(’u−2v=’)
print(s)
```

R code Exa 1.9 9

```
#Chapter 1 − Linear Equations In Linear Algebra
#Determination of Linear Combination
#Page No.25 / 1−13
#Prob 13
#1.3.13

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

av<-c(1,-4,2,3,0,3,5,-7,-2,8,-4,-3)
```

```

a=matrix(
  av,
  nrow=3,
  ncol=4,
  byrow=TRUE
)

print('the augmented matrix is:')
print(a)

#Row reduce the augmented matrix to reach echelon form
print('R3=R3+2*R1')
a[3,]=a[3,]+2*a[1,]
print(a)
print('The system for the augmented matrix is inconsistent, so b is not a linear combination of the columns of A')
```

R code Exa 1.10 10

```
#Chapter 1 - Linear Equations In Linear Algebra
#Determine if b is in W
#Page No.25 / 1-13
#Prob 25
#1.3.25

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

av<-c(1,0,-4,4,0,3,-2,1,-2,6,3,-4)
```

```
a=matrix(
  av,
  nrow=3,
  ncol=4,
  byrow=TRUE
)

print('the augmented matrix is:')
print(a)

print('R3=R3+2*R1')
a[3,]=a[3,]+2*a[1,]
print(a)
print('R3=R3-2*R2')
a[3,]=a[3,]-2*a[2,]
print(a)
```

R code Exa 1.11 11

```
#Chapter 1 - Linear Equations In Linear Algebra
#Applications of Gaussian Elimination
#Page No.26 / 1-14
#Prob 31
#1.3.31

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('1 gram at (0,1), 1 gram at (8,1) and 1 gram at (2,4)')
m1v<-c(0,1)
```

```
m2v<-c(8,1)
m3v<-c(2,4)
m1=matrix(m1v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
m2=matrix(m2v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
m3=matrix(m3v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
cm=(1/3)*(1*m1+1*m2+1*m3)
print('centre of mass is at')
print(cm)
print('the new weight of the system=9 grams')
print('new centre of mass is at')
sv<-c(2,2)
s=matrix(
  sv,
  nrow=2,
  ncol=1,
  byrow=TRUE
)
print(s)
print('let w1,w2 and w3 be the weights added at (0,1),(8,1) and (2,4) respectively')
print('hence, w1+w2+w3=6')
print('using the formula for the centre of mass, we get')
print('8*w2+2*w3=8 and w1+w2+4*w3=12')
av<-c(1,1,1,6,0,8,2,8,1,1,4,12)
a=matrix(
  av,
  nrow=3,
  ncol=4,
  byrow=TRUE
)
print('the augmented matrix is:')
print(a)
print('R3=R3-R1')
a[3,]=a[3,]-a[1,]
print(a)
print('R3=(1/3)*R3')
a[3,]=(1/3)*a[3,]
print(a)
print('R2=R2-2*R3 and R1=R1-R3')
a[2,]=a[2,]-2*a[3,]
a[1,]=a[1,]-a[3,]
print(a)
print('R1=R1-(1/8)*R2')
a[1,]=a[1,]-(1/8)*a[2,]
print(a)
print('R2=(1/8)*R2')
a[2,]=(1/8)*a[2,]
print(a)
cat(sprintf('Add %.1f grams at (0,1), %.1f grams at (8,1) and %d grams at (2,4)',a[1,4],a[2,4],a[3,4]) )
```

R code Exa 1.12 12

```
#Chapter 1 - Linear Equations In Linear Algebra
#Vectors as columns of a matrix
#Page No.28 / 1-16
#Prob 7
#1.4.7

#clear console
cat("\014")

```

```
#clear variables
rm(list=ls(all=TRUE))

print('the three vectors are:')
uv<-c(4,-1,7,-4)
vv<-c(-5,3,-5,1)
wv<-c(7,-8,0,2)
u<-matrix(uv,
          nrow=4,
          ncol=1,
          byrow=TRUE)
v<-matrix(vv,
          nrow=4,
          ncol=1,
          byrow=TRUE)
w<-matrix(wv,
          nrow=4,
          ncol=1,
          byrow=TRUE)

print(u)
print(v)
print(w)
print('u v and w form the columns of A')
A=cbind(u,v,w)
print(A)
print('the augmented matrix is:')
nmv<-c(6,-8,0,-7)
nm<-matrix(nmv,
           nrow=4,
           ncol=1,
           byrow=TRUE)
c=cbind(A,nmv)
print(c)
```

R code Exa 1.13 13

```
#Chapter 1 - Linear Equations In Linear Algebra
#Span of vectors
#Page No.29 / 1-17
#Prob 13
#1.4.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the augmented matrix is:')
av<-c(3,-5,0,-2,6,4,1,1,4)
a=matrix(
  av,
  nrow=3,
  ncol=3,
  byrow=TRUE
)

print(a)
print('interchange R1 and R3')
a[c(1,3),]=a[c(3,1),]
print(a)
print('R2=R2+2*R1 and R3=R3-3*R1')
a[2,]=a[2,]+2*a[1,]
a[3,]=a[3,]-3*a[1,]
print(a)
print('R3=R3+R2')
a[3,]=a[3,]+a[2,]
print(a)
print('from the entries of last row, the system is consistent')
print('hence, u is in the plane spanned by the columns of a')
```

R code Exa 1.14 14

```r
#Chapter 1 - Linear Equations In Linear Algebra
#Free and pivot variables
#Page No.33 / 1-21
#Prob 1
#1.5.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the augmented matrix is:')
av<-c(2,-5,8,0,-2,-7,1,0,4,2,7,0)
a=matrix(av,
         nrow=3,
         ncol=4,
         byrow=TRUE)
print(a)
print('R2=R2+2*R1 and R3=R3-2*R1')
a[2,]=a[2,]+a[1,]
a[3,]=a[3,]-2*a[1,]
print(a)
print('R3=R3+R2')
a[3,]=a[3,]+a[2,]
print(a)
print('only two columns have non zero pivots')
print('hence, one column is a free column and therefore there exists a non trivial solution')
```

R code Exa 1.15 15

```
#Chapter 1 - Linear Equations In Linear Algebra
#General solution of the system
#Page No.34 / 1-22
#Prob 7
#1.5.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the augmented matrix is:')
av<-c(1,3,-3,7,0,0,1,-4,5,0)
a=matrix(av,
         nrow=2,
         ncol=5,
         byrow=TRUE)
print(a)
print('R1=R1-3*R2')
a[1,]=a[1,]-3*a[2,]
print(a)
print('basic variables:x1 x2')
print('free variables:x3 x4')
print('x1=-9*x3+8*x4')
print('x2=4*x3-5*x4')
print('hence, solution is')
print('[-9*x3+8*x4 4*x3-5*x4 x3 x4]')
uv<-c(-9,4,1,0)
vv<-c(8,-5,0,1)
u=matrix(uv,
  nrow=4,
  ncol=1,
  byrow=TRUE
)
v=matrix(vv,
          nrow=4,
          ncol=1,
          byrow=TRUE
)
print('The solution set is the same as Span{u,v}')
print('u=')
print(u)
print('v=')
print(v)
```

R code Exa 1.16 16

```
#Chapter 1 - Linear Equations In Linear Algebra
#General solution of the system
#Page No.34 / 1-22
#Prob 11
#1.5.11

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the augmented matrix is')
av<-c(1,-4,-2,0,3,-5,0,0,0,1,0,0,-1,0,0,0,0,0,-1,4,0,0,0,0,0,0,0,0)
a=matrix(av,
          nrow=4,
          ncol=7,
          byrow=TRUE)
print(a)
print('R1=R1-3*R3')
a[1,]=a[1,]-3*a[3,]
```

```
print(a)
print(’R1=R1+2*R2’)
a[1,]=a[1,]+2*a[2,]
print(a)
print(’the free variables are:x2, x4 and x6’)
print(’the basic variables are:x1, x3 and x5’)
print(’the solution is:’)
print(’[4*x2−5*x6  x2  x6  x4  4*x6  x6]’)
```

R code Exa 1.17 17

```
#Chapter 1 − Linear Equations In Linear Algebra
#General solution of the system
#Page No.35 / 1−23
#Prob 13
#1.5.13

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

xv<-c(5,-2,0)
x=matrix(xv,
         nrow=3,
         ncol=1,
         byrow=TRUE)
x1v<-c(4,-7,1)
x1=matrix(x1v,
         nrow=3,
         ncol=1,
         byrow=TRUE)
print(x)
print(x1)
print('=p+x3*q')
cat('geometrically the solution set is the line through [', x ,'] parallel to [',x1,']')
```

R code Exa 1.18 18

```
#Chapter 1 - Linear Equations In Linear Algebra
#Economic Equilibrium
#Page No.38 / 1-26
#Prob 1
#1.6.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

av<-c(0.8,-0.7,0,-0.8,0.7,0)
a=matrix(av,
         nrow=2,
         ncol=3,
         byrow=TRUE)
print('the augmented matrix is')
print(a)

#row reduce augmented matrix
a[2,]=a[2,]+a[1,]
print(a)
a=a*1.25
print(a)

print('ratio of prices is pg=0.875ps')
print('one equilibrium solution is ps=1000 and pg=875')
print('economic equilibrium is unaffected by a proportional change in prices')
```

R code Exa 1.19 19

```
#Chapter 1 - Linear Equations In Linear Algebra
#Balanced Equation
#Page No.39 / 1-27
#Prob 7
#1.6.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

nahcov<-c(1,1,1,3)
hchov<-c(0,8,6,7)
nachov<-c(3,5,6,7)
hov<-c(0,2,0,1)
cov<-c(0,0,1,2)

nahco=matrix(nahcov,
             nrow=4,
             ncol=1,
             byrow=TRUE)
hcho=matrix(hchov,
             nrow=4,
             ncol=1,
             byrow=TRUE)
nacho=matrix(nachov,
             nrow=4,
             ncol=1,
             byrow=TRUE)
```

```r
ho=matrix(hov,
          nrow=4,
          ncol=1,
          byrow=TRUE)
co=matrix(cov,
          nrow=4,
          ncol=1,
          byrow=TRUE)

print('nahco3')
print(nahco)
print('h3c6h5o7')
print(hcho)
print('na3c6h5o7')
print(nacho)
print('h2o')
print(ho)
print('co2')
print(co)
zm= matrix(0, nrow = 4, ncol = 1)
av=cbind(nahco,hcho,nacho,ho,co,zm)

print('augmented matrix')
av[,3]=av[,3]*-1
av[,4]=av[,4]*-1
av[,5]=av[,5]*-1
print(av)

amrv<-c(1,0,0,0,-1,0,0,1,0,0,-1/3,0,0,0,1,0,-1/3,0,0,0,0,1,-1,0)
amr=matrix(amrv,
           nrow=4,
           ncol=6,
           byrow=TRUE)
print('reduced augmented matrix')
print(amr)
print('the general solution is :')
print('x1=x5')
print('x2=(1/3)*x5')
print('x3=(1/3)*x5')
print('x4=x5')
print('x5 is free')
print('if x5=3, then')
print('x1=x4=3')
print('x2=x3=1')
print('the balanced equation is :')
print('3NaHCO3+H3C6H5O7 -> Na3C6H5O7+3H20+3C02')
```

R code Exa 1.20 20

```
#Chapter 1 - Linear Equations In Linear Algebra
#Intersection Equations
#Page No.40 / 1-28
#Prob 13
#1.6.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

av1<-c
   (1,-1,0,0,0,0,-50,0,1,-1,1,-1,0,0,0,0,0,0,1,-1,60,0,0,0,1,0,-1,50

a1=matrix(av1,
          nrow=5,
          ncol=7,
          byrow=TRUE)
print(a1)
av2<-c
   (1,-1,0,0,0,0,-50,0,1,-1,1,-1,0,0,0,0,0,1,0,-1,50,0,0,0,0,1,-1,60
a2=matrix(av2,
          nrow=5,
          ncol=7,
          byrow=TRUE)
print('..by reduction')
print(a2)
av3<-c
   (1,0,-1,0,0,0,-40,0,1,-1,0,0,0,10,0,0,0,1,0,-1,50,0,0,0,0,1,-1,60

a3=matrix(av3,
          nrow=5,
          ncol=7,
          byrow=TRUE)
print('..by reduction')
print(a3)
print('part−a')
print('the general solution is:')
print('x1=x3−40')
print('x2=x3+10')
print('x3 is free')
print('x4=x6+50')
print('x5=x6+60')
print('x6 is free')
print('part−b')
print('x3>=40')
print('x2>=50')
print('x4>=50')
print('x5>=60')
print('the minimum flows are:')
print('x2=50')
print('x3=40')
print('x4=50')
print('x5=60')
```

R code Exa 1.21 21

```
#Chapter 1 – Linear Equations In Linear Algebra
#Linear independence of vectors
#Page No.42 / 1–30
#Prob 1
#1.7.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’given vectors u, v and w are’)
uv<-c(5,0,0)
um=matrix(uv,
         nrow=1,
         ncol=3,
         byrow=TRUE)
u=t(um)
print(u)
vv<-c(7,2,-6)
vm=matrix(vv,
          nrow=1,
          ncol=3,
          byrow=TRUE)
v=t(vm)
print(v)
wv<-c(9,4,-8)
wm=matrix(wv,
         nrow=1,
         ncol=3,
         byrow=TRUE)
w=t(wm)
print(w)
print(’the augmented matrix is’)
av<-c(5,7,9,0,0,2,4,0,0,-6,-8,0)
a=matrix(av,
         nrow=3,
          ncol=4,
          byrow=TRUE)
print(a)
print('R3=R3+3*R2')
a[3,]=a[3,]+3*a[2,]
print(a)
print('there are no free variables')
print('hence, the homogeneous equation has only trivial solution and the vectors are linearly independent')
```

R code Exa 1.22 22

```
#Chapter 1 - Linear Equations In Linear Algebra
#Linear dependence of vectors
#Page No.43 / 1-31
#Prob 7
#1.7.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the augmented matrix is')
Av1<-c(1,4,-3,0,0,-2,7,5,1,0,-4,-5,7,-5,0)
A1=matrix(Av1,
          nrow=3,
          ncol=5,
          byrow=TRUE)
print(A1)
Av2<-c(1,4,-3,0,0,0,1,-1,1,0,0,11,-5,5,0)
A2=matrix(Av2,
          nrow=3,
         ncol=5,
         byrow=TRUE)
print(A2)
Av3<-c(1,4,-3,0,0,0,1,-1,1,0,0,0,6,-6,0)
A3=matrix(Av3,
         nrow=3,
         ncol=5,
         byrow=TRUE)
print(A3)
print('since there are three rows, the maximum number of pivots can be 3')
print('hence, at least one of the four variable must be free')
print('so the equations have non trivial solution and the columns of A are linearly independent')
```

R code Exa 1.23 23

```
#Chapter 1 − Linear Equations In Linear Algebra
#Introduction to Linear Transformations
#Page No.46 / 1−34
#Prob 1
#1.8.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

Av<-c(2,0,0,2)
uv<-c(1,-3)
A=matrix(Av,
         nrow=2,
         ncol=2,
         byrow=TRUE)
u=matrix(uv,
         nrow=2,
         ncol=1,
         byrow=TRUE)
print(A)
print(u)
T=A %*% u
print(T)
```

R code Exa 1.24 24

```
#Chapter 1 – Linear Equations In Linear Algebra
#Introduction to Linear Transformations
#Page No.46 / 1–34
#Prob 5
#1.8.5

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

av1<-c(1,-5,-7,-2,-3,7,5,-2)
av2<-c(1,-5,-7,-2,0,1,2,1)
av3<-c(1,0,3,3,0,1,2,1)
am1=matrix(av1,
           nrow=2,
           ncol=4,
           byrow=TRUE)
am2=matrix(av2,
           nrow=2,
           ncol=4,
           byrow=TRUE)
```

```
am3=matrix(av3,
           nrow=2,
           ncol=4,
           byrow=TRUE)
print(am1)
print('~')
print(am2)
print('~')
print(am3)
print('x1=3-3*x3')
print('x2=1-2*x3')
print('x3 is free')
print('General Solution')
m1v<-c(3,1,0)
m1=matrix(m1v,
          nrow=3,
          ncol=1,
          byrow=TRUE)
m2v<-c(-3,-2,1)
m2=matrix(m2v,
          nrow=3,
          ncol=1,
          byrow=TRUE)
print(m1)
print('+ x3*')
print(m2)
print('when x3=0')
mv<-c(3,1,0)
m=matrix(mv,
         nrow=3,
         ncol=1,
         byrow=TRUE)
print(m)
```

R code Exa 1.25 25

```
#Chapter 1 – Linear Equations In Linear Algebra
#Introduction to Linear Transformations
#Page No.47 / 1–35
#Prob 19
#1.8.19

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

m1v<-c(2,5)
m2v<-c(-1,6)
m1=matrix(m1v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
print(m1)
m2=matrix(m2v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
print(m2)
print(’5y1–3y2=’)
tm=5*m1-3*m2
print(tm)
```

R code Exa 1.26 26

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

```

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

te1v<-c(3,1,3,1)
te2v<-c(-5,2,0,0)
te1=matrix(te1v,
           nrow=4,
           ncol=1,
           byrow=TRUE)
print(te1)
te2=matrix(te2v,
           nrow=4,
           ncol=1,
           byrow=TRUE)
print(te2)
A=cbind(te1,te2)
print(A)
```

R code Exa 1.27 27

```
#Chapter 1 – Linear Equations In Linear Algebra
#Matrix of a linear transformation
#Page No.50 / 1−38
#Prob 7
#1.9.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))
```

```

e1v<-c(-1/sqrt(2),-1/sqrt(2))
e1=matrix(e1v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
print('e1')
print(e1)
e1[2]=e1[2]*-1
print('->')
print(e1)

e2v<-c(1/sqrt(2),-1/sqrt(2))
e2=matrix(e2v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
print('e2')
print(e2)
e2[2]=e2[2]*-1
print('->')
print(e2)
A=cbind(e1,e2)
print('A')
print(A)
```

R code Exa 1.28 28

```
#Chapter 1 - Linear Equations In Linear Algebra
#Matrix of a linear transformation
#Page No.51 / 1-39
#Prob 21
#1.9.21

#clear console
```

```
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

av<-c(1,1,4,5)
bv<-c(3,8)
a=matrix(av,
         nrow=2,
         ncol=2,
         byrow=TRUE)
print(’a’)
print(a)
b=matrix(bv,
         nrow=2,
         ncol=1,
         byrow=TRUE)
print(’b’)
print(b)
A=cbind(a,b)
print(’A’)
print(A)
print(’R2=R2−4R1’)
A[2,]=A[2,]-4*A[1,]
print(A)
print(’R1=R1−4R2’)
A[1,]=A[1,]-A[2,]
print(A)
xv=c(7,-4)
x=matrix(xv,nrow=2,ncol=1,byrow = TRUE)
print(’x=’)
print(x)
```

R code Exa 1.29 29

```
#Chapter 1 – Linear Equations In Linear Algebra
#Linear Models in Business , Science and Engineering
#Page No.55 / 1−43
#Prob 1
#1.10.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

av<-c(110,130,295,4,3,9,20,18,48,2,5,8)
a=matrix(av,
         nrow=4,
         ncol=3,
         byrow=TRUE)
print(a)
print(’~’)
print(’interchange R1 and R4’)
a[c(1,4),]=a[c(4,1),]
print(a)
print(’~’)
a[1,]=a[1,]/2
a[3,]=a[3,]/2
print(a)
a1v<-c(1,2.5,4,0,-7,-7,0,-16,-16,0,-145,-145)
a2v<-c(1,2.5,4,0,1,1,0,0,0,0,0,0)
a3v<-c(1,0,1.5,0,1,1,0,0,0,0,0,0)
a1=matrix(a1v,
          nrow=4,
          ncol=3,
          byrow=TRUE)
a2=matrix(a2v,
          nrow=4,
          ncol=3,
          byrow=TRUE)
a3=matrix(a3v,
          nrow=4,
          ncol=3,
          byrow=TRUE)
print(a1)
print('~')
print(a2)
print('~')
print(a3)
```

R code Exa 1.30 30

```
#Chapter 1 - Linear Equations In Linear Algebra
#Linear Models in Business , Science and Engineering
#Page No.56 / 1-44
#Prob 7
#1.10.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

r1v<-c(12,-7,0,-4)
r2v<-c(-7,15,-6,0)
r3v<-c(0,-6,14,-5)
r4v<-c(-4,0,-5,13)
r1=matrix(r1v,
          nrow=4,
          ncol=1,
          byrow=TRUE)
r2=matrix(r2v,
          nrow=4,
          ncol=1,
          byrow=TRUE)
```

```
r3=matrix(r3v,
          nrow=4,
          ncol=1,
          byrow=TRUE)
r4=matrix(r4v,
          nrow=4,
          ncol=1,
          byrow=TRUE)
print('r1')
print(r1)
print('r2')
print(r2)
print('r3')
print(r3)
print('r4')
print(r4)
R=cbind(r1,r2,r3,r4)
print(R)
vv<-c(40,30,20,-10)
v=matrix(vv,
         nrow=4,
         ncol=1,
         byrow=TRUE)
print('v')
print(v)
x<-solve(R,v)
print('the solution is')
print(x)
```

R code Exa 1.31 31

```
#Chapter 1 - Linear Equations In Linear Algebra
#Linear Models in Business, Science and Engineering
#Page No.57 / 1-45
#Prob 13
```

```
#1.10.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

mv<-c(0.95,0.03,0.05,0.97)
x0v<-c(600000,400000)
m=matrix(mv,
         nrow=2,
         ncol=2,
         byrow=TRUE)
print('M')
print(m)
x0=matrix(x0v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
print('when X0=')
print(x0)
x5v<-c(523293,476707)
x5=matrix(x5v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
print('x5')
print(x5)
x10v<-c(472737,527263)
x10=matrix(x10v,
           nrow=2,
           ncol=1,
           byrow=TRUE)
print('x10')
print(x10)
x15v<-c(439417,560583)
x15=matrix(x15v,
           nrow=2,
           ncol=1,
           byrow=TRUE)
print('x15')
print(x15)
x20v<-c(417456,582544)
x20=matrix(x20v,
           nrow=2,
           ncol=1,
           byrow=TRUE)
print('x20')
print(x20)
x0=matrix(x0v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
x0v<-c(350000,650000)
print('when X0=')
x0=matrix(x0v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
print(x0)
x5v<-c(358523,641477)
x5=matrix(x5v,
          nrow=2,
          ncol=1,
          byrow=TRUE)
print('x5')
print(x5)
x10v<-c(364140,635860)
x10=matrix(x10v,
           nrow=2,
           ncol=1,
           byrow=TRUE)
print('x10')
print(x10)
x15v<-c(367843,632157)
```

```
x15=matrix(x15v,
           nrow=2,
           ncol=1,
           byrow=TRUE)
print('x15')
print(x15)
x20v<-c(370283,629717)
x20=matrix(x20v,
           nrow=2,
           ncol=1,
           byrow=TRUE)
print('x20')
print(x20)
```

R code Exa 1.32 32

```
#Chapter 1 − Linear Equations In Linear Algebra
#Supplementary Exercises
#Page No.58 / 1−46
#Prob 7a
#7a

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

v1v<-c(2,-5,7)
v2v<-c(-4,1,-5)
v3v<-c(-2,1,-3)

v1=matrix(v1v,3,1,TRUE)
v2=matrix(v2v,3,1,TRUE)
v3=matrix(v3v,3,1,TRUE)
```

```
print('v1')
print(v1)
print('v2')
print(v2)
print('v3')
print(v3)
R=cbind(v1,v2,v3)
print(R)
R[1,]=R[1,]/2
print('~')
print(R)
Rnv<-c(1,-2,-1,0,-9,-4,0,9,4)
Rn=matrix(Rnv,3,3,TRUE)
print('~')
print(Rn)
Rnv<-c(1,-2,-1,0,-9,-4,0,0,0)
Rn=matrix(Rnv,3,3,TRUE)
print('~')
print(Rn)
```

Chapter 2

Matrix Algebra

R code Exa 2.1 1

```
#Chapter 2− Matrix Algebra
#Matrix operations
#Page No.64 / 2−2
#Prob 1
#2.1.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

av<-c(2,0,-1,4,-5,2)
a=matrix(av,2,3,TRUE)
print(’A=’)
print(a)
print(’−2A=’)
print(-2*a)
print(’matrix B’)
bv<-c(7,-5,1,1,-4,-3)
b=matrix(bv,2,3,TRUE)
```

```
print(b)
print('B-2A')
print(b-2*a)
c=matrix(c(1,2,-2,1),2,2,TRUE)
d=matrix(c(3,5,-1,4),2,2,TRUE)
cd=c %*% d
print('CD')
print(cd)
```

R code Exa 2.2 2

```
#Chapter 2- Matrix Algebra
#Inverse of a matrix
#Page No.67 / 2-5
#Prob 1
#2.2.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('given matrix:')
av<-c(8,6,5,4)
a=matrix(av,2,2,TRUE)
print(a)
print('inverse of the matrix is:')
print(solve(a))
```

R code Exa 2.3 3

```
#Chapter 2- Matrix Algebra
```

```
#Inverse of a matrix
#Page No.67 / 2-5
#Prob 7
#2.2.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the co-efficient matrix is:')
av<-c(1,2,5,12)
a=matrix(av,2,2,TRUE)
print(a)
print('inverse of the matrix is:')
print(solve(a))
print('solution is:')

b1v<-c(-1,3)
b1=matrix(b1v,2,1,TRUE)
c=solve(a)
res=c %*% b1
print(res)

b2v<-c(1,-5)
b2=matrix(b2v,2,1,TRUE)
res1=c %*% b2
print(res1)

b3v<-c(2,6)
b3=matrix(b3v,2,1,TRUE)
res2=c %*% b3
print(res2)

b4v<-c(3,5)
b4=matrix(b4v,2,1,TRUE)
res3=c %*% b4
```

```r
print(res3)

cm=cbind(a,b1,b2,b3,b4)
print(cm)
cm[2,]=cm[2,]-5*cm[1,]
print(cm)
cm[2,]=cm[2,]/2
print(cm)
cm[1,]=cm[1,]-2*cm[2,]
print(cm)
```

R code Exa 2.4 4

```r
#Chapter 2- Matrix Algebra
#Matrix
#Page No.70 / 2-8
#Prob 31
#2.2.31

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

av<-c(1,0,-2,-3,1,4,2,-3,4)
a=matrix(av,3,3,TRUE)
i=diag(3)
ai=cbind(a,i)
print(ai)
ai[2,]=ai[2,]+3*ai[1,]
print(ai)
ai[3,]=ai[3,]-2*ai[1,]
print(ai)
ai[3,]=ai[3,]+3*ai[2,]
```

```
print(ai)
ai[1,]=ai[1,]+ai[3,]
print(ai)
ai[2,]=ai[2,]+ai[3,]
print(ai)
ai[3,]=ai[3,]/2
print(ai)
print('A^-1')
print(ai[,c(4,5,6)])
```

R code Exa 2.5 5

```
#Chapter 2- Matrix Algebra
#Invertibility of a matrix
#Page No.73 / 2-11
#Prob 1
#2.3.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the given matrix is:')
av<-c(5,7,-3,-6)
a=matrix(av,2,2,TRUE)
print(a)
print('the columns are lineraly independent')
print('hence, by invertible matrix theorem')
print('the matrix A is invertible')
```

R code Exa 2.6 6

```
#Chapter 2– Matrix Algebra
#Invertible matrix
#Page No.73 / 2–11
#Prob 7
#2.3.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

av<-c(-1,-3,0,1,3,5,8,-3,-2,-6,3,2,0,-1,2,1)
a=matrix(av,4,4,TRUE)
print(a)
a[1,]=a[1,]*-1
a[2,]=a[2,]-3*a[1,]
a[3,]=a[3,]+2*a[1,]
a[1,]=a[1,]*-1
print(a)
a[4,]=4*a[4,]-a[2,]
a[4,]=a[4,]/4
print(a)
```

R code Exa 2.7 7

```
#Chapter 2– Matrix Algebra
#Invertible matrix theorem
#Page No.74 / 2–12
#Prob 33
#2.3.33

#clear console
cat(”\014”)

```

```r
#clear variables
rm(list=ls(all=TRUE))

print('matrix A corresponding to transformation T is :')
Av<-c(-5,9,4,-7)
A=matrix(Av,2,2,TRUE)
print(A)
print('determinant of A is:')
print(det(A))
print('since det(A) is not equal to zero')
print('by IMT, A is invertible')
print('hence, the inverse of A exists')
print('inverse of A is:')
print(solve(A))
```

R code Exa 2.8 8

```r
#Chapter 2- Matrix Algebra
#Inverse using matrix partition
#Page No.81 / 2-19
#Prob 25
#2.4.25

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('given matrix is:')
av<-c(1,2,0,0,0,3,5,0,0,0,0,0,2,0,0,0,0,0,7,8,0,0,0,5,6)
a=matrix(av,5,5,TRUE)
```

```r
print(a)
print('partitioning the matrix into 4 submatrices')
A11=rbind(a[1,1:2],a[2,1:2])
print('A11=')
print(A11)
A22=rbind(a[3,3:5],a[4,3:5],a[5,3:5])
print('A22=')
print(A22)
A12=matrix(0,2,3,TRUE)
print('A12=')
print(A12)
A21=matrix(0,3,2,TRUE)
print('A21=')
print(A21)
print('partitioning A22 into 4 submatrices')
A221=matrix(2,1,1,TRUE)
print(A221)
B=rbind(A22[2,2:3],A22[3,2:3])
print('B=')
print(B)
print(matrix(0,1,2,TRUE))
print(matrix(0,2,1,TRUE))
print('determinant of B=')
print(det(B))
print('Hence, B is invertible')
print('inverse of B is')
print(solve(B))
print('determinant of inverse of  B is:')
print(det(solve(B)))
print('inverse of A11')
print(solve(A11))
print('hence the inverse of A22 is:')
v1=det(solve(B))
v2=matrix(0,1,2,TRUE)
v3=matrix(c(0,3,-4),1,3,TRUE)
v4=matrix(c(0,-2.5,3.5),1,3,TRUE)
r1=cbind(v1,v2)
c=rbind(r1,v3,v4)
```

```
print(c)
```

R code Exa 2.9 9

```
#Chapter 2− Matrix Algebra
#Application of LU decomposition
#Page No.83 / 2−21
#Prob 1
#2.5.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the lower triangular matrix is:’)
Lv<-c(1,0,0,-1,1,0,2,-5,1)
L=matrix(Lv,3,3,TRUE)
print(L)
print(’the upper triangular matrix is:’)
Uv<-c(3,-7,-2,0,-2,-1,0,0,-1)
U=matrix(Uv,3,3,TRUE)
print(U)
print(’the RHS of the equations are’)
bv<-c(-7,5,2)
b=matrix(bv,3,1,TRUE)
print(b)
print(’combining matrices L and b’)
cm=cbind(L,b)
print(cm)
print(’performing row operations’)
print(’R2=R2+R1’)
cm[2,]=cm[2,]+cm[1,]
print(cm)
print('R3=R3-2*R1')
cm[3,]=cm[3,]-2*cm[1,]
print(cm)
print('R3=R3+5*R2')
cm[3,]=cm[3,]+5*cm[2,]
print(cm)
y=cm[,4]
print('y=')
print(y)
print('combining U and y')
d=cbind(U,y)
print(d)
print('performing row operations')
print('R3=R3/-6')
d[3,]=d[3,]/(-1)
print(d)
print('R2=R2+R3 and R1=R1+2*R3')
d[2,]=d[2,]+d[3,]
d[1,]=d[1,]+2*d[3,]
print(d)
print('R1=R1-3.5*R2')
d[1,]=d[1,]-3.5*d[2,]
print(d)
print('R1=R1/3 and R2=R2/-2')
d[1,]=d[1,]/3
d[2,]=d[2,]/(-2)
print(d)
print('the solution is:')
x=d[,4]
print('x=')
print(x)
```

R code Exa 2.10 10

```
#Chapter 2- Matrix Algebra
```

```
#LU decomposition of a matrix
#Page No.84 / 2−22
#Prob 7
#2.5.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('given matrix is:')
av<-c(2,5,-3,-4)
a=matrix(av,2,2,TRUE)
d=a;
print(a)
print('performing row operations')
a[2,]=a[2,]-(a[2,1]/a[1,1])*a[1,]
print(a)
print(a)
print('thus, the upper triangular matrix is')
U=a;
print('U=')
print(U)
print('the lower triangular matrix is:')
c2=d[2,1]/d[1,1]
r1=matrix(c(1,c2),2,1,TRUE)
r2=matrix(c(0,1),2,1,TRUE)
L=cbind(r1,r2)
print('L=')
print(L)
```

R code Exa 2.11 11

```
#Chapter 2− Matrix Algebra
```

```
#LU decomposition of a matrix
#Page No.84 / 2−22
#Prob 13
#2.5.13

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print('given matrix is:')
av<-c(1,3,-5,-3,-1,-5,8,4,4,2,-5,-7,-2,-4,7,5)
a=matrix(av,4,4,TRUE)
d=a
print(a)
print('performing row operations')
p21=a[2,1]/a[1,1]
p31=a[3,1]/a[1,1]
p41=a[4,1]/a[1,1]
a[2,]=a[2,]-p21*a[1,]
a[3,]=a[3,]-p31*a[1,]
a[4,]=a[4,]-p41*a[1,]
print(a)
p32=a[3,2]/a[2,2]
p42=a[4,2]/a[2,2]
a[3,]=a[3,]-p32*a[2,]
a[4,]=a[4,]-p42*a[2,]
print(a)
print('thus, lower triangular matrix is:')
v1=matrix(c(1,0,0,0),1,4,TRUE)
v2=matrix(c(p21,1,0,0),1,4,TRUE)
v3=matrix(c(p31,p32,1,0),1,4,TRUE)
v4=matrix(c(p41,p42,0,1),1,4,TRUE)
L=rbind(v1,v2,v3,v4)
print('L=')
print(L)
print('Upper triangular matrix is:')
print('U=')
print(a)
```

R code Exa 2.12 12

```
#Chapter 2- Matrix Algebra
#Application of matrix algebra
#Page No.91 / 2-29
#Prob 1
#2.6.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the consumption matrix is:')
cv<-c(.1,.6,.6,.3,.2,0,.3,.1,.1)
cm=matrix(cv,3,3,TRUE)
print(cm)
print('Assuming that agriculture plans to produce 100 units and other units produce nothing')
print('the production vector is given by')
x=matrix(c(0,100,0),3,1,TRUE)
print('x=')
print(x)
print('thus the intermediate demand is:')
print(cm %*% x)
```

R code Exa 2.13 13

```
#Chapter 2- Matrix Algebra
```

```
#Application of matrix algebra
#Page No.91 / 2−29
#Prob 7
#2.6.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the consumption matrix is:’)
Cv<-c(0,.5,.6,.2)
Cm=matrix(Cv,2,2,TRUE)
print(Cm)
print(’the demand for 1 unit of output sector 1’)
d1=matrix(c(1,0),2,1,TRUE)
print(d1)
print(’the production required to satisfy demand d1 is:’)
x1=solve(diag(2)-Cm) %*% d1
print(’x1=’)
print(x1)
print(’the final demand is:’)
d2=matrix(c(51,30),2,1,TRUE)
print(’d2=’)
print(d2)
print(’the production required to satisfy demand d2 is:’)
x2=solve(diag(2)-Cm) %*% d2
print(’x2=’)
print(x2)
```

R code Exa 2.14 14

```r
#Chapter 2– Matrix Algebra
#Application of matrix algebra
#Page No.93 / 2–31
#Prob 7
#2.7.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the matrix in R2 to rotate a vector by 60 degrees is:')
av<-c(cos(pi/3),-sin(pi/3),sin(pi/3),cos(pi/3))
a=matrix(av,2,2,TRUE)
print(a)
x=matrix(c(6,8),2,1,TRUE)
print('x=')
print(x)
print('so the 3X3 matrix for rotation about x is:')
yv<-c(1,0,6,0,1,8,0,0,1)
y=matrix(yv,3,3,TRUE)
zv<-c(1,0,-6,0,1,-8,0,0,1)
z=matrix(zv,3,3,TRUE)
av<-c(cos(pi/3),-sin(pi/3),0,sin(pi/3),cos(pi/3),0,0,0,1)
a=matrix(av,3,3,TRUE)
R=y %*% (a %*% z)
print(R)
```

R code Exa 2.15 15

```r
#Chapter 2– Matrix Algebra
#R3 coordinates of matrix
```

```
#Page No.94 / 2-32
#Prob 19
#2.7.19

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

Pv<-c(1,0,0,0,0,1,0,0,0,0,0,0,0,0,-0.1,1)
P=matrix(Pv,4,4,TRUE)
print(P)
Dv<-c(4.2,6,2,1.2,4,2,4,2,6,1,1,1)
D=matrix(Dv,4,3,TRUE)
print(D)
PD=P %*% D
print('PD')
print(PD)
R3=matrix(0,3,3,TRUE)
R31=PD[1:3,1]/PD[4,1]
R32=PD[1:3,2]/PD[4,2]
R33=PD[1:3,3]/PD[4,3]
R3=cbind(R31,R32,R33)
print(R3)
```

R code Exa 2.16 16

```
#Chapter 2- Matrix Algebra
#Column space of a matrix
#Page No.95 / 2-33
#Prob 7c
#2.8.7c

#clear console
```

```
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the given matrix is:’)
Av<-c(2,-3,-4,-8,8,6,6,-7,-7)
A=matrix(Av,3,3,TRUE)
print(’A=’)
print(A)
print(’the given vector is:’)
p=matrix(c(6,-10,11),3,1,TRUE)
print(’p=’)
print(p)
print(’combining A and p’)
b=cbind(A,p)
print(b)
print(’performing row operations’)
b[2,]=b[2,]-(b[2,1]/b[1,1])*b[1,]
b[3,]=b[3,]-(b[3,1]/b[1,1])*b[1,]
print(b)
b[3,]=b[3,]-(b[3,2]/b[2,2])*b[2,]
print(b)
if(b[3,3]==0 & b[3,4]==0){
  print(’p lies in column space of A’)
}else{
  print(’p does not lie in column space of A’)
}
```

R code Exa 2.17 17

```
#Chapter 2– Matrix Algebra
#Matrix operation
#Page No.96 / 2–34
#Prob 13
```

```
#2.8.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

av<-c(3,2,1,-5,0,-9,-4,1,7,0,9,2,-5,1,0)
a=matrix(av,3,5,TRUE)
print(a)
a[2,]=a[2,]+a[3,]
a[2,]=a[2,]/(-1)
a[3,]=a[3,]-3*a[1,]
print(a)
a[3,]=a[3,]+2*a[2,]
print(a)
a[2,]=a[2,]/2
print(a)
a[1,]=a[1,]-2*a[2,]
a[1,]=a[1,]/3
print(a)
print('the generalsolution is x1=x3-x4')
print('and x2=-2x3+4x4')
print('x3 and x4 are free')
print('if x3=1 and x4=0')
print('vector is (1,-2,1,0)')
print('if x3=0 and x4=1')
print('vector is (-1,4,0,1)')
```

R code Exa 2.18 18

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

```
#Prob 23
#2.8.23

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the given matrix is:’)
av<-c(4,5,9,-2,6,5,1,12,3,4,8,-3)
a=matrix(av,3,4,TRUE)
print(a)
print(’performing row operaions’)
a[1,]=a[1,]*(1/4)
a[2,]=a[2,]-6*a[1,]
a[3,]=a[3,]-3*a[1,]
a[2,]=a[2,]*(-2/5)
a[3,]=a[3,]+(-1/4)*a[2,]
a[1,]=a[1,]+(-5/4)*a[2,]
a[1,]=a[1,]+2*a[2,]
print(a)
il<-c(1:3)
jl<-c(1:4)
for(i in il){
  for(j in jl){
    if(a[i,j]!=0){
      cat(’column ’,j,’ is a pivot column\n’)
      break
    }
  }
}
a[1,]=a[1,]-2*a[2,]
zm=matrix(0,3,1,TRUE)
c=cbind(a,zm)
print(c)
print(’x1−4x3+7x4=0’)
print(’x2+5x3−6x4=0’)
print(’0=0’)
```

R code Exa 2.19 19

```
#Chapter 2− Matrix Algebra
#Pivot columns
#Page No.97 / 2−35
#Prob 25
#2.8.25

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the given matrix is:’)
av<-c(1,4,8,-3,-7,-1,2,7,3,4,-2,2,9,5,5,3,6,9,-5,-2)
a=matrix(av,4,5,TRUE)
print(a)
print(’performing row operations’)
a[2,]=a[2,]+a[1,]
a[3,]=a[3,]+2*a[1,]
a[4,]=a[4,]-3*a[1,]
a[2,]=a[2,]*(1/6)
a[3,]=a[3,]-10*a[2,]
a[4,]=a[4,]+6*a[2,]
a[3,]=a[3,]*-1
a[4,]=a[4,]-4*a[3,]
a[1,]=a[1,]+3*a[3,]
a[2,]=a[2,]*2
print(a)
il<-c(1:4)
jl<-c(1:5)
for(i in il){
  for(j in jl){
    if(a[i,j]!=0){
      cat('column ',j,' is a pivot column\n')
      break
    }
  }
}
a[2,]=a[2,]/2
a[1,]=a[1,]-4*a[2,]
print(a)
print('x1-2x3+7x5=0')
print('x2+2.5x3-0.5x5=0')
print('x4+4x5=0')
print('0=0')
print('x1=2x3-7x5')
print('x2=-2.5x3+0.5x5')
print('x3=x3')
print('x4=-4x5')
print('x5=x5')
```

R code Exa 2.20 20

```
#Chapter 2- Matrix Algebra
#Pivot columns
#Page No.98 / 2-36
#Prob 37
#2.8.37

#clear console
cat("\014")
#clear variables
rm(list=ls(all=TRUE))
av<-c(3,-5,0,-1,3,-7,9,-4,9,-11,-5,7,-2,5,-7,3,-7,-3,4,0)
```

```
a=matrix(av,4,5,TRUE)
print(a)
a[1,]=a[1,]*(1/3)
a[2,]=a[2,]+7*a[1,]
a[3,]=a[3,]+5*a[1,]
a[4,]=a[4,]+(-3)*a[1,]
a[2,]=a[2,]*(-3/8)
a[3,]=a[3,]+(4/3)*a[2,]
a[4,]=a[4,]+2*a[2,]
a[1,]=a[1,]+(5/3)*a[2,]
a[4,]=ceiling(a[4,])#the numbers are very small
a[3,]=floor(a[3,])#the numbers are very small
a[1,2]=ceiling(a[1,2])
print(a)
il<-c(1:4)
jl<-c(1:5)
for(i in il){
  for(j in jl){
    if(a[i,j]!=0){
      cat('column ',j,' is a pivot column\n')
      break
    }
  }
}
print('x1+2.5x3-4.5x4+3.5x5=0')
print('x2+1.5x3-2.5x4+1.5x5=0')
print('x3,x4,x5 are free')
```

R code Exa 2.21 21

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

```

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

xs=matrix(c(3,2),2,1,TRUE)
print(xs)
b1=matrix(c(1,1),2,1,TRUE)
b2=matrix(c(2,-1),2,1,TRUE)
print(b1)
print(b2)
x=3*b1+2*b2
print(x)
```

R code Exa 2.22 22

```
#Chapter 2− Matrix Algebra
#Pivot columns
#Page No.102 / 2−40
#Prob 7
#2.9.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

xs=matrix(c(2,-1),2,1,TRUE)
print(xs)
xs1=matrix(c(1.5,0.5),2,1,TRUE)
print(xs1)
b1=matrix(c(3,0),2,1,TRUE)
```

```
b2=matrix(c(-1,2),2,1,TRUE)
print(b1)
print(b2)
x=1.5*b1+0.5*b2
print(x)
```

R code Exa 2.23 23

```
#Chapter 2- Matrix Algebra
#Dimension of a matrix
#Page No.102 / 2-40
#Prob 13
#2.9.13

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print('the given matrix is:')
av<-c(1,-3,2,-4,-3,9,-1,5,2,-6,4,-3,-4,12,2,7)
a=matrix(av,4,4,TRUE)
print(a)
print('performing row operations')
a[2,]=a[2,]-(a[2,1]/a[1,1])*a[1,]
a[3,]=a[3,]-(a[3,1]/a[1,1])*a[1,]
a[4,]=a[4,]-(a[4,1]/a[1,1])*a[1,]
print(a)
a[4,]=a[4,]-2*a[2,]
print(a)
a[4,]=a[4,]-a[3,]
print(a)
k=0
il=c(1:4)
```

```
jl=c(1:4)
for(i in il)
{for (j in jl){
if(a[i,j]!=0){
  k=k+1
break
}
}
}
print('dimension of the matrix=')
print(k)
```

Chapter 3

Determinants

R code Exa 3.1 1

```
#Chapter 3− Determinants
#Determinant of a matrix
#Page No.108 / 3−2
#Prob 1
#3.1.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the given matrix is:’)
Av<-c(3,0,4,2,3,2,0,5,-1)
A=matrix(Av,3,3,TRUE)
print(A)
print(’calculating det(A) using cofactor expression along first row’)
print(’det(A)=3 X (−1 X 3−5 X 2)+4 X (2 X 5−3 X 0)’)
print(’det(A) =’)
print(det(A))
```

R code Exa 3.2 2

```
#Chapter 3- Determinants
#Determinant of a matrix
#Page No.108 / 3-2
#Prob 7
#3.1.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('given matrix is:')
Av<-c(4,3,0,6,5,2,9,7,3)
A=matrix(Av,3,3,TRUE)
print(A)
print('calculating det(A) using cofactor expression
   along first row')
print('det(A)=4 X (5 X 3-7 X 2)-3 X (6 X 3-9 X 2)')
print('det(A) =')
print(det(A))
```

R code Exa 3.3 3

```
#Chapter 3- Determinants
#Determinant of a matrix
#Page No.108 / 3-2
#Prob 13
#3.1.13
```

```

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the given matrix is:')
Av<-c(4,0,-7,3,-5,0,0,2,0,0,7,3,-6,4,-8,5,0,5,2,-3,0,0,9,-1,2)
A=matrix(Av,5,5,TRUE)
print('A=')
print(A)
P=A
print('since row 2 has maximum zeros, using row 2 for cofactor expression')
A<-A[-(2),]
A<-A[,-(3)]
print('deleting second row and third column from A, we get')
print(A)
print('det(A)=-2 X')
print('det')
print(A)
print('for the 4X4 matrix obtained, using column 2 for cofactor exansion')
print('deleting second column and row from the 4X4 matrix')
A<-A[-(2),]
A<-A[,-(2)]
print(A)
print('det(A)=-2 X 3 X')
print('det')
print(A)
print('-6 X [4 X (4-3)-5 X (6-5)]')
print('=')
print(-6*det(A))
```

R code Exa 3.4 4

```r
#Chapter 3- Determinants
#Property of determinants
#Page No.110 / 3-4
#Prob 37
#3.1.37

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

Av<-c(3,1,4,2)
A=matrix(Av,2,2,TRUE)
print('the given matrix is:')
print(A)
print('det(A)=')
print(det(A))
print('5 X A = ')
print(5*A)
print('det(5*A)=')
print(det(5*A))
print('thus, det(5A) is not equal to 5Xdet(A)')
print('infact, the relation between det(rA) and det(A) for a nxn matrix is:')
print('det(rA)=(r^n)*det(A)')
```

R code Exa 3.5 5

```r
#Chapter 3- Determinants
#Determinant of a matrix
#Page No.110 / 3-4
#Prob 41
#3.1.41

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

av<-c(3,1,0,2)
a=matrix(av,2,2,TRUE)
print(a)
print('det=')
print(det(a))
```

R code Exa 3.6 6

```r
#Chapter 3- Determinants
#Determinant of a matrix
#Page No.111 / 3-5
#Prob 7
#3.2.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the given matrix is:')
Av<-c(1,3,0,2,-2,-5,7,4,3,5,2,1,1,-1,2,-3)
A=matrix(Av,4,4,TRUE)
```

```
print('A=')
print(A)
print('performing row operations')
A[2,]=A[2,]-(A[2,1]/A[1,1])*A[1,]
A[3,]=A[3,]-(A[3,1]/A[1,1])*A[1,]
A[4,]=A[4,]-(A[4,1]/A[1,1])*A[1,]
print(A)
A[3,]=A[3,]-(A[3,2]/A[2,2])*A[2,]
A[4,]=A[4,]-(A[4,2]/A[2,2])*A[2,]
print(A)
A[4,]=A[4,]-(A[4,3]/A[3,3])*A[3,]
print(A)
print('det(A) is the product of diagonal entries')
print('det(A)=')
print(det(A))
```

R code Exa 3.7 7

```
#Chapter 3- Determinants
#Determinant of a matrix
#Page No.112 / 3-6
#Prob 13
#3.2.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the given matrix is:')
av<-c(2,5,4,1,4,7,6,2,6,-2,-4,0,-6,7,7,0)
a=matrix(av,4,4,TRUE)
print('A=')
print(a)
print(’performing row operations’)
a[2,]=a[2,]-2*a[1,]
print(a)
print(’using cofactor expansion about fourth column’
   )
a<-a[-(1),]
a<-a[,-(4)]
print(’det(A)= −1 X’)
print(’det’)
print(a)
print(’performing row operations’)
a[3,]=a[3,]+a[2,]
print(a)
print(’using cofactor expansion about first column’)
a<-a[-(2),]
a<-a[,-(1)]
print(’det(A)= −1 X −6 X’)
print(’det’)
print(a)
print(’=’)
print(6*det(a))
```

R code Exa 3.8 8

```
#Chapter 3− Determinants
#Determinant of a matrix
#Page No.112 / 3−6
#Prob 19
#3.2.19

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))
```

```

print('the given matrix is:')
print('A=')
print(' a         b         c')
print('2d+a     2e+b     2f+c')
print(' g         h         i')
print('B=')
print('a   b   c')
print('d   e   f')
print('g   h   i')
print('given, det(B)=7')
print('performing row operations on A')
print('R2=R2-R1')
print('A=')
print('a    b    c')
print('2d   2e   2f')
print('g    h    i')
print('factoring 2 out of row 2')
print('A=')
print('2 X')
print('a   b   c')
print('d   e   f')
print('g   h   i')
print('therefore, det(A)=2 X det(B)')
print('=2 X 7')
print('= 14')
```

R code Exa 3.9 9

```
#Chapter 3- Determinants
#Linear independency using determinants
#Page No.112 / 3-6
#Prob 25
#3.2.25

```

```
#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the given vectors are:’)
v1v<-c(7,-4,-6)
v2v<-c(-8,5,7)
v3v<-c(7,0,-5)
v1=matrix(v1v,3,1,TRUE)
v2=matrix(v2v,3,1,TRUE)
v3=matrix(v3v,3,1,TRUE)
print(’v1=’)
print(v1)
print(’v2=’)
print(v2)
print(’v3=’)
print(v3)
print(’combining them as a matrix’)
a=cbind(v1,v2,v3)
print(’A=’)
print(a)
print(’if det(A) is not equal to zero, then v1 v2 and v3 are linearly independent’)
print(’expanding about third column’)
print(’det(A)=7 X (-28+30) - 5 X (35-32)’)
print(’=’)
print(det(a))
print(’hence, v1 v2 and v3 are linearly independent’)
```

R code Exa 3.10 10

```
#Chapter 3- Determinants
```

```
#Cramers Rule
#Page No.115 / 3-9
#Prob 1
#3.3.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the co-efficient matrix is:')
av<-c(5,7,2,4)
a=matrix(av,2,2,TRUE)
print('A=')
print(a)
print('the RHS is:')
b=matrix(c(3,1),2,1,TRUE)
print(b)
print('applying cramers rule')
print('replacing first column of matrix A by b')
A1v<-c(3,7,1,4)
A1=matrix(A1v,2,2,TRUE)
print('A1=')
print(A1)
print('replacing second column of matrix A by b')
A2v<-c(5,3,2,1)
A2=matrix(A2v,2,2,TRUE)
print('A2=')
print(A2)
print('x1=det(A1)/det(A)')
print('=')
print((det(A1)/det(a)))
print('x2=det(A2)/det(A)')
print('=')
print((det(A2)/det(a)))
```

R code Exa 3.11 11

```
#Chapter 3- Determinants
#Inverse of a matrix
#Page No.116 / 3-10
#Prob 13
#3.3.13

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the given matrix is:’)
av<-c(3,5,4,1,0,1,2,1,1)
a=matrix(av,3,3,TRUE)
print(’A=’)
print(a)
print(’the cofactors are:’)
m1=matrix(c(0,1,1,1),2,2,TRUE)
C11=det(m1)
print(’C11=’)
print(C11)
m2=matrix(c(1,1,2,1),2,2,TRUE)
C12=-det(m2)
print(’C12=’)
print(C12)
m3=matrix(c(1,0,2,1),2,2,TRUE)
C13=det(m3)
print(’C13=’)
print(C13)
m3=matrix(c(5,4,1,1),2,2,TRUE)
C21=-det(m3)
```

```
print('C21=')
print(C21)
m4=matrix(c(3,4,2,1),2,2,TRUE)
C22=det(m4)
print('C22=')
print(C22)
m4=matrix(c(3,5,2,1),2,2,TRUE)
C23=-det(m4)
print('C23=')
print(C23)
m5=matrix(c(5,4,0,1),2,2,TRUE)
C31=det(m5)
print('C31=')
print(C31)
m6=matrix(c(3,4,1,1),2,2,TRUE)
C32=-det(m6)
print('C32=')
print(C32)
m7=matrix(c(3,5,1,0),2,2,TRUE)
C33=det(m7)
print('C33=')
print(C33)
c1m=rbind(C11,C12,C13)
c2m=rbind(C21,C22,C23)
c3m=rbind(C31,C32,C33)
B=cbind(c1m,c2m,c3m)
print('adj(A)=')
print(B)
print('A.adj(A)=')
adja=ceiling(a %*% B)
print(adja)
C=B/(det(a))
print('inv(A)=')
print(C)
```

R code Exa 3.12 12

```
#Chapter 3- Determinants
#Applicant of determinant
#Page No.117 / 3-11
#Prob 19
#3.3.19

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’the points forming the parrallelogram are’)
print(’(0,0),(5,2),(6,4),(11,6)’)
print(’using the vertices adjacent to origin to form a matrix’)
Av<-c(5,6,2,4)
A=matrix(Av,2,2,TRUE)
print(’A=’)
print(A)
print(’Area of parallelogram = det(A)’)
print(’=’)
print(det(A))
```

Chapter 4

Vector Spaces

R code Exa 4.1 1

```
#Chapter 4- Vector Spaces
#Subspace of vectors
#Page No.123 / 4-3
#Prob 13c
#4.1.13c

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the given vectors are:')
v1=matrix(c(1,0,-1),3,1,TRUE)
print('v1=')
print(v1)
v2=matrix(c(2,1,3),3,1,TRUE)
print('v2=')
print(v2)
v3=matrix(c(4,2,6),3,1,TRUE)
print('v3=')
print(v3)
w=matrix(c(3,1,2),3,1,TRUE)
print('w=')
print(w)
print('It is clear that w is not one of the three vectors in v1,v2 and v3')
print('The span of v1,v2 and v3 contains infinitely many vectors.')
print('To check if w is in the subspace of v1,v2 and v3,')
print('we form an augmented matrix.')
av<-c(1,2,4,3,0,1,2,1,-1,3,6,2)
a=matrix(av,3,4,TRUE)
print(a)
print('performing row operations')
print('R3=R3+R1')
a[3,]=a[3,]+a[1,]
print(a)
print('R3=R3-5xR2')
a[3,]=a[3,]-5*a[2,]
print(a)
print('there is no pivot in the augmented column,')
print('hence the vector equation is consistent and w is in span{v1 v2 v3}.')
```

R code Exa 4.2 2

```
#Chapter 4- Vector Spaces
#Null space of a matrix
#Page No.125 / 4-5
#Prob 1
#4.2.1

#clear console
cat("\014")
```

```

#clear variables
rm(list=ls(all=TRUE))

print('the given matrix is:')
av<-c(3,-5,-3,6,-2,0,-8,4,1)
a=matrix(av,3,3,TRUE)
print('A=')
print(a)
print('the vector x is:')
x=matrix(c(1,3,-4),3,1,TRUE)
print('x=')
print(x)
print('To check if x is in nullspace of A')
print('Ax=')
zm=matrix(0,3,1,TRUE)
print('=')
print(zm)
print('hence, x is in the null space of A')
```

R code Exa 4.3 3

```
#Chapter 4- Vector Spaces
#Column space of a matrix
#Page No.128 / 4-8
#Prob 13
#4.3.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the given matrix is:')
```

```
av<-c(1,0,6,5,0,0,2,5,3,0,0,0,0,0,0)
a=matrix(av,3,5,TRUE)
p=a
print('A=')
print(a)
print('Reducing A to echelon form')
print('R2=R2/2')
a[2,]=a[2,]/2
print(a)
print('the pivot columns are column 1 and 2 of A')
print('x1=-6x3-5x4')
print('x2=-(5/2)x3-(3/2)x4')
print('x3 and x4 are free')
print('x3[-6 -5/2 1 0] + ')
print('x4[-5 -3/2 0 1]')
print('the standard choice is (-6,-5/2,1,0)')
print('and (-5,-3/2,0,1) ')
print('Another choice is ')
print('(-12,-5,2,0) ')
print('and (-10,-3,0,2) ')
print('which avoids fractions')
```

R code Exa 4.4 4

```
#Chapter 4- Vector Spaces
#Matrix operation
#Page No.132 / 4-12
#Prob 1
#4.4.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))
```

```

xb=matrix(c(5,3),2,1,TRUE)
print(xb)
b1=matrix(c(3,-5),2,1,TRUE)
b2=matrix(c(-4,6),2,1,TRUE)
x=5*b1+3*b2
print(x)
```

R code Exa 4.5 5

```
#Chapter 4- Vector Spaces
#Gaussian Elimination
#Page No.132 / 4-12
#Prob 7
#4.4.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('vector x=')
x=matrix(c(8,-9,6),3,1,TRUE)
print(x)
print('the given basis is:')
b1=matrix(c(1,-1,-3),3,1,TRUE)
b2=matrix(c(-3,4,9),3,1,TRUE)
b3=matrix(c(2,-2,4),3,1,TRUE)
print('b1=')
print(b1)
print('b2=')
print(b2)
print('b3=')
print(b3)
print('to solve the vector equation')
print('an augmented matrix is formed')
av<-c(1,-3,2,8,-1,4,-2,-9,-3,9,4,6)
a=matrix(av,3,4,TRUE)
print('A=')
print(a)
print('performing row operations')
a[2,]=a[2,]-(a[2,1]/a[1,1])*a[1,]
a[3,]=a[3,]-(a[3,1]/a[1,1])*a[1,]
print(a)
a[3,]=a[3,]/a[3,3]
a[1,]=a[1,]-2*a[3,]
print(a)
a[1,]=a[1,]+3*a[2,]
print(a)
print('Xb=')
print(a[,4])
```

R code Exa 4.6 6

```
#Chapter 4- Vector Spaces
#Matix Solution
#Page No.133 / 4-13
#Prob 13
#4.4.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('c1+c3=1')
print('c2+2c3=4')
print('c1+c2+c3=7')
```

```

av<-c(1,0,1,1,0,1,2,4,1,1,1,7)
a=matrix(av,3,4,TRUE)
print(a)
a[3,]=a[3,]-1*a[1,]
a[3,]=a[3,]-1*a[2,]
print(a)
a[3,]=a[3,]/-2
a[2,]=a[2,]-2*a[3,]
a[1,]=a[1,]-1*a[3,]
print(a)
ps=a[,4]
print('ps=')
print(ps)
m1=matrix(c(1,0,1),3,1,TRUE)
m2=matrix(c(0,1,1),3,1,TRUE)
m3=matrix(c(1,2,1),3,1,TRUE)
mm=matrix(c(1,4,7),3,1,TRUE)
print('c1')
print(m1)
print('c2')
print(m2)
print('c3')
print(m3)
print('=')
print(mm)
```

R code Exa 4.7 7

```
#Chapter 4- Vector Spaces
#Linear independence of vectors
#Page No.135 / 4-15
#Prob 27
#4.4.27

```

```
#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('to check if vectors v1 v2 and v3 are linearly independent')
v1=matrix(c(1,0,0,1),4,1,TRUE)
v2=matrix(c(3,1,-2,1),4,1,TRUE)
v3=matrix(c(0,-1,3,-1),4,1,TRUE)
print('v1=')
print(v1)
print('v2=')
print(v2)
print('v3=')
print(v3)
print('forming an augmented matrix')
av<-c(1,3,0,0,0,1,-1,0,0,-2,3,0,1,0,-1,0)
a=matrix(av,4,4,TRUE)
print('A=')
print(a)
print('performing row operations')
a[4,]=a[4,]+a[1,]
a[1,]=a[1,]-a[4,]
a[1,]=a[1,]/(-1)
print(a)
a[4,]=a[4,]-a[1,]
a[3,]=a[3,]+2*a[2,]
a[4,]=a[4,]+3*a[2,]
a[4,]=a[4,]+4*a[3,]
a[4,]=a[4,]-6*a[2,]
a[4,]=a[4,]-a[1,]
a[4,]=a[4,]/8
a[4,]=a[4,]-a[3,]
print(a)
print('since the vector equation has only the trivial solution')
print('vectors v1 v2 and v3 are linearly independent')
```

R code Exa 4.8 8

```
#Chapter 4- Vector Spaces
#Span of vectors
#Page No.135 / 4-15
#Prob 31a
#4.4.31a

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('to check if the polynomials span R3')
print('placing the coordinate vectors of the polynomial into the columns of a matrix')
av<-c(1,-3,-4,1,-3,5,5,0,5,-7,-6,-1)
a=matrix(av,3,4,TRUE)
print('A=')
print(a)
print('performing row operations')
a[2,]=a[2,]+3*a[1,]
a[3,]=a[3,]-5*a[1,]
print(a)
a[3,]=a[3,]+2*a[2,]
print(a)
print('the four vectors DO NOT span R3 as there is no pivot in row 3')

#4.4.31b

```

```
print('to check if the polynomials span R3')
print('placing the coordinate vectors of the polynomial into the columns of a matrix')
av1<-c(0,1,-3,2,5,-8,4,-3,1,-2,2,0)
a1=matrix(av1,3,4,TRUE)
print('A=')
print(a1)
print('performing row operations')
a1[c(1,3),]=a1[c(3,1),]
print(a1)
a1[2,]=a1[2,]-5*a1[1,]
print(a1)
a1[3,]=a1[3,]-0.5*a1[2,]
print(a1)
print('the four vectors DO NOT span R3 as there is no pivot in row 3')
```

R code Exa 4.9 9

```
#Chapter 4- Vector Spaces
#Dimension of a vector space
#Page No.137 / 4-17
#Prob 3
#4.5.3

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('to find the dimension of subspace H, which is the set of linear combination of vectors v1 v2 and v3')
v1=matrix(c(0,1,0,1),4,1,TRUE)
```

```
v2=matrix(c(0,-1,1,2),4,1,TRUE)
v3=matrix(c(2,0,-3,0),4,1,TRUE)
print('v1=')
print(v1)
print('v2=')
print(v2)
print('v3=')
print(v3)
print('Clearly, v1 is not equal to zero')
print('and v2 is not a multiple of v1 as third
   element of v1 is zero whereas that of v2 is 1.')
print('Also, v3 is not a linear combination of v1
   and v2 as the first element of v1 and v2 is zero
   but that of v3 is 2')
print('Hence, v1 v2 and v3 are linearly independent
   and dim(H)=3')
```

R code Exa 4.10 10

```
#Chapter 4- Vector Spaces
#Rank of a matrix
#Page No.141 / 4-21
#Prob 1
#4.6.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('to find the rank of matrix A')
av<-c(1,-4,9,-7,-1,2,-4,1,5,-6,10,7)
a=matrix(av,3,4,TRUE)
p=a
```

```
print('A=')
print(a)
print('performing row operations')
a[2,]=a[2,]+a[1,]
a[3,]=a[3,]-5*a[1,]
a[3,]=a[3,]+7*a[2,]
a[1,]=a[1,]-2*a[2,]
print(a)
print('It is clear that matrix A has 2 pivot columns
   ')
print('Hence, rank(A)=2')
print('Columns 1 and 2 are pivot columns')
print('Hence, basis for C(A) is:')
cat(p[,1],'and',p[,2])
print('Basis for row space of A is:')
cat(a[1,],'and',a[2,])
print('To find the basis of N(A), solve Ax=0')
print('on solving, we get the basis of N(A) as:')
u=matrix(c(1,2.5,1,0),4,1,TRUE)
v=matrix(c(-5,-3,0,1),4,1,TRUE)
cat(v,'and',u)
```

R code Exa 4.11 11

```
#Chapter 4- Vector Spaces
#Change of Basis
#Page No.145 / 4-25
#Prob 1a
#4.7.1a

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))
```

```

b1=matrix(c(6,-2),2,1,TRUE)
print(b1)
b2=matrix(c(9,-4),2,1,TRUE)
print(b2)
pcb=cbind(b1,b2)
print(pcb)

#4.7.1b
xb=matrix(c(-3,2),2,1,TRUE)
print(xb)
xc=pcb %*% xb
print(xc)
```

R code Exa 4.12 12

```
#Chapter 4− Vector Spaces
#Change of Basis
#Page No.145 / 4−25
#Prob 7
#4.7.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

c1=matrix(c(1,-5),2,1,TRUE)
c2=matrix(c(-2,2),2,1,TRUE)
b1=matrix(c(7,5),2,1,TRUE)
b2=matrix(c(-3,-1),2,1,TRUE)
m=cbind(c1,c2,b1,b2)
print(m)
m[2,]=m[2,]+5*m[1,]
```

```
print(m)
m[2,]=m[2,]/8
m[2,]=m[2,]/-1
print(m)
m[1,]=m[1,]+2*m[2,]
print(m)
pcb=m[,3:4]
print(pcb)
print('inv(pcb)')
print(solve(pcb))
```

R code Exa 4.13 13

```
#Chapter 4- Vector Spaces
#Change of Basis
#Page No.146 / 4-26
#Prob 13
#4.7.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

b1c=matrix(c(1,-2,1),3,1,TRUE)
b2c=matrix(c(3,-5,4),3,1,TRUE)
b3c=matrix(c(0,2,3),3,1,TRUE)
print('b1c')
print(b1c)
print('b2c')
print(b2c)
print('b3c')
print(b3c)
pcb=cbind(b1c,b2c,b3c)
```

```
print(pcb)
print('coordinate vector [-1+2t]B satisfies')
print('pcb [-1+2t]B = [-1+2t]C')
print('=[-1 2 0]')
nm=matrix(c(-1,2,0),3,1,TRUE)
cvm=cbind(pcb,nm)
print(cvm)
cvm[2,]=cvm[2,]+2*cvm[1,]
cvm[3,]=cvm[3,]-1*cvm[1,]
cvm[3,]=cvm[3,]-1*cvm[2,]
cvm[2,]=cvm[2,]-2*cvm[3,]
cvm[1,]=cvm[1,]-3*cvm[2,]
print(cvm)
print('[-1+2t]B = ')
print(cvm[,4])
```

R code Exa 4.14 14

```
#Chapter 4- Vector Spaces
#Change of Basis
#Page No.146 / 4-26
#Prob 19a
#4.7.19a

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

vv<-c(-2,-8,-7,2,5,2,3,2,6)
pv<-c(1,2,-1,-3,-5,0,4,6,1)
v=matrix(vv,3,3,TRUE)
p=matrix(pv,3,3,TRUE)
print(v)
print(p)
vp=v %*% p
print(vp)
u1=vp[,1]
u2=vp[,2]
u3=vp[,3]
print(’u1’)
print(u1)
print(’u2’)
print(u2)
print(’u3’)
print(u3)
```

R code Exa 4.15 15

```
#Chapter 4- Vector Spaces
#Linear Independence
#Page No.148 / 4-28
#Prob 7
#4.8.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

av<-c(1^0,2^0,(-2)^0,1^1,2^1,(-2)^1,1^2,2^2,(-2)^2)
a=matrix(av,3,3,TRUE)
print(a)
a[2,]=a[2,]-a[1,]
print(a)
a[3,]=a[3,]-a[1,]
print(a)
a[3,]=a[3,]-3*a[2,]
```

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

R code Exa 4.16 16

```
#Chapter 4- Vector Spaces
#Change of Basis
#Page No.149 / 4-29
#Prob 13
#4.8.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

r1=(1+(1/3))/2
r2=(1-(1/3))/2
print('r=')
print(r1)
print('or')
print(r2)
cat('Two solutions of the difference equation are',
   r1,'^k and',r2,'^k')
```

R code Exa 4.17 17

```
#Chapter 4- Vector Spaces
#Change of Basis
#Page No.149 / 4-29
#Prob 19
```

```r
#4.8.19

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

r1=(-4+sqrt(16-4))/2
r2=(-4-sqrt(16-4))/2
print('r=')
print(r1)
print('or')
print(r2)
cat('Two solutions of the difference equation are',r1,'^k and',r2,'^k')
print('')
cat('The general solution has the form c1',r1,'^k+c2',r2,'^k')
```

R code Exa 4.18 [18]

```r
#Chapter 4- Vector Spaces
#Change of Basis
#Page No.150 / 4-30
#Prob 19b
#4.8.19b
#(this solution is for Section 4.7)

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

```

```
vv<-c(-2,-8,-7,2,5,2,3,2,6)
v=matrix(vv,3,3,TRUE)
print(v)
pv<-c(5,8,5,-3,-5,-3,-2,-2,-1)
pinv=matrix(pv,3,3,TRUE)
print(pinv)
w= v %*% pinv
print(w)
w1=w[,1]
w2=w[,2]
w3=w[,3]
print(w1)
print(w2)
print(w3)
```

R code Exa 4.19 19

```
#Chapter 4- Vector Spaces
#Change of Basis
#Page No.152 / 4-32
#Prob 1
#4.9.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

p=matrix(c(.7,.6,.3,.4),2,2,TRUE)
x1m=matrix(c(1,0),2,1,TRUE)
x2m=matrix(c(.7,.3),2,1,TRUE)
print(’x1m’)
print(x1m)
print(’x2m’)
print(x2m)
x1=p %*% x1m
x2=p %*% x2m
print(’x1’)
print(x1)
print(’x2’)
print(x2)
```

R code Exa 4.20 20

```
#Chapter 4− Vector Spaces
#Change of Basis
#Page No.153 / 4−33
#Prob 7
#4.9.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

pv<-c(.7,.1,.1,.2,.8,.2,.1,.1,.7)
p=matrix(pv,3,3,TRUE)
print(p)
PI=p-diag(3)
print(PI)
zm=matrix(0,3,1,TRUE)
PIx=cbind(PI,zm)
print(PIx)
PIx[,c(1,3)]=PIx[,c(3,1)]
print(PIx)
PIx=PIx*10
print(PIx)
PIx[2,]=PIx[2,]-2*PIx[1,]
```

```
PIx[3,]=PIx[3,]+3*PIx[1,]
PIx[2,]=PIx[2,]/(-4)
PIx[3,]=PIx[3,]-4*PIx[2,]
PIx[1,]=PIx[1,]-1*PIx[2,]
print(round(PIx))
print('x1=x3')
print('x2=2x3')
print('x3 is free')
print('[x1 x2 x3]=x3[1 2 1]')
print('the entries in [1 2 1] sum to 4')
q=(1/4)*matrix(c(1,2,1),3,1,TRUE)
print(q)
```

R code Exa 4.21 21

```
#Chapter 4- Vector Spaces
#Change of Basis
#Page No.154 / 4-34
#Prob 13a
#4.9.13a

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

P=matrix(c(.95,.45,.05,.55),2,2,TRUE)
print(P)
PI=P-diag(2)
print(PI)
zm=matrix(0,2,1,TRUE)
PIm=cbind(PI,zm)
print(PIm)
PIm[2,]=round(PIm[2,]+PIm[1,])
```

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

Chapter 5

Eigenvalues And Eigenvectors

R code Exa 5.1 [1]

```
#Chapter 5- Eigenvalues and Eigenvectors
#Eigenvalue of a matrix
#Page No.157 / 5-1
#Prob 1
#5.1.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('to check if 2 is an eigenvalue of matrix A')
a=matrix(c(3,2,3,8),2,2,TRUE)
print('A=')
print(a)
print('A-2I=')
b=a-2*diag(2)
print(b)
print('The columns of A are clearly independent,')
print('hence (A-2I)x=0 has a non trivial solution
   and 2 is an eigenvalue of matrix A')
```

R code Exa 5.2 2

```
#Chapter 5- Eigenvalues and Eigenvectors
#Eigenvalue of a matrix
#Page No.158 / 5-2
#Prob 7
#5.1.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('To check if 4 is an eigenvalue of matrix A')
av<-c(3,0,-1,2,3,1,-3,4,5)
a=matrix(av,3,3,TRUE)
print('A=')
print(a)
print('Therefore')
print('A-4I=')
print(a-4*diag(3))
b=a-4*diag(3)
print('to check the invertibility of A-4I, form an augmented matrix')
zm=matrix(0,3,1,TRUE)
c=cbind(b,zm)
print(c)
print('performing row operations')
c[2,]=c[2,]+2*c[1,]
c[3,]=c[3,]-3*c[1,]
print(c)
c[3,]=c[3,]+4*c[2,]
```

```
print(c)
print('We can see that there exists a non trivial solution.')
print('Hence, 4 is an eigenvalue of A.')
print('For the eigenvector, -x1-x3=0 and -x2-x3=0')
print('If x3=1,')
x=matrix(c(-1,-1,1),3,1,TRUE)
print('x=')
print(x)
```

R code Exa 5.3 3

```
#Chapter 5- Eigenvalues and Eigenvectors
#Eigenvectors
#Page No.158 / 5-2
#Prob 13
#5.1.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('To find a basis for the eigenspace')
print('Matrix A=')
av<-c(4,0,1,-2,1,0,-2,0,1)
a=matrix(av,3,3,TRUE)
print(a)
print('for lambda=1')
print('A-1I=')
b=a-diag(3)
print(b)
print('solving (A-I)x=0, we get')
print('-2*x1=0 and 3*x1+x3=0')
print('therefore, x1=x3=0')
print('which leaves x2 as a free variable')
print('Hence a basis for the eigen space is:')
es=matrix(c(0,1,0),3,1,TRUE)
print(es)
print('for lambda=2')
print('A-2I=')
b=a-2*diag(3)
print(b)
print('performing row operations on the augmented matrix')
zm=matrix(0,3,1,TRUE)
c=cbind(b,zm)
print(c)
c[2,]=c[2,]+c[1,]
c[3,]=c[3,]+c[1,]
print(c)
c[1,]=c[1,]/c[2,2]
print(c)
print('We can see that x3 is a free variable')
print('x2=x3 and x1=-.05*x3')
print('Hence, a basis for the eigenspace is:')
es1=matrix(c(1,2,2),3,1,TRUE)
print(es1)
print('for lambda=3')
print('A-3I=')
b=a-3*diag(3)
print(b)
print('performing row operations on the augmented matrix')
c=cbind(b,zm)
print(c)
c[2,]=c[2,]+2*c[1,]
c[3,]=c[3,]+2*c[1,]
print(c)
c[2,]=c[2,]/2
c[2,]=c[2,]/(-1)
print(c)
print('Again x3 is a free variable')
print('x1=-x3 and x2=x3')
print('Hence, a basis for the eigenspace is:')
es2=matrix(c(-1,1,1),3,1,TRUE)
print(es2)
```

R code Exa 5.4 4

```
#Chapter 5- Eigenvalues and Eigenvectors
#Property of non-invertible matrices
#Page No.159 / 5-3
#Prob 19
#5.1.19

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('The given matrix is:')
av<-c(1,2,3,1,2,3,1,2,3)
a=matrix(av,3,3,TRUE)
print('A=')
print(a)
print('A is not invertible because its columns are linearly dependent.')
print('Hence, 0 is an eigenvalue of matrix A.')
```

R code Exa 5.5 5

```
#Chapter 5- Eigenvalues and Eigenvectors
#Eigenvalue of a matrices
```

```
#Page No.162 / 5-6
#Prob 1
#5.2.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('To find the eigenvalue of matrix A')
print('A=')
a=matrix(c(2,7,7,2),2,2,TRUE)
print(a)
print('Eigen values of A are:')
eig<-eigen(a)
eigv<-eig$values
print(eigv)
```

R code Exa 5.6 6

```
#Chapter 5- Eigenvalues and Eigenvectors
#Complex eigenvalues
#Page No.162 / 5-6
#Prob 7
#5.2.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('To find the eigenvalues of matrix A.')
print('A=')
```

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

R code Exa 5.7 7

```
#Chapter 5- Eigenvalues and Eigenvectors
#Eigenvalues of a matrix
#Page No.162 / 5-6
#Prob 13
#5.2.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('To find the eigenvalues of the matrix A')
print('A=')
av<-c(6,-2,0,-2,9,0,5,8,3)
a=matrix(av,3,3,TRUE)
print(a)
print('Eigenvalues of A are:')
eig<-eigen(a)
eigv<-eig$values
print(eigv)
```

R code Exa 5.8 8

```
#Chapter 5- Eigenvalues and Eigenvectors
#Eigenvectors
#Page No.163 / 5-7
#Prob 25a
#5.2.25a

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('Matrix A=')
a=matrix(c(.6,.3,.4,.7),2,2,TRUE)
print(a)
print('Eigenvector v1=')
v1=matrix(c(3/7,4/7),2,1,TRUE)
print(v1)
print('vector Xo=')
Xo=matrix(c(.5,.5),2,1,TRUE)
print(Xo)
print('Eigenvalues of A are:')
eig<-eigen(a)
c<-eig$values
print(c)
print('To verify if v1 is an eigenvector of A:')
print('A*v1=')
print(a %*% v1)
print('=')
print('1*v1')
print('Hence v1 is an eigenvector of A corresponding to eigenvalue 1.')
print('for lambda=.3')
print('A-.3I=')
b=a-.3*diag(2)
print(b)
print('performing row operations on the augmented matrix')
```

```
zm=matrix(0,2,1,TRUE)
c=cbind(b,zm)
print(c)
c[2,]=c[2,]-(c[2,1]/c[1,1])*c[1,]
c[2,2]=round(c[2,2])
print(c)
print('hence, x1+x2=0')
print('Eigenvector corresponding to eigenvalue .3 is :')
ev=matrix(c(-1,1),2,1,TRUE)
print(ev)

#5.2.25b
x0=matrix(c((1/2),(1/2)),2,1,TRUE)
v1=matrix(c((3/7),(4/7)),2,1,TRUE)
md=x0-v1
print(md)
print('=')
print('(1/14)v2')
```

R code Exa 5.9 9

```
#Chapter 5- Eigenvalues and Eigenvectors
#Diagonalization of a matrix
#Page No.166 / 5-10
#Prob 1
#5.3.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('The given eigenvector matrix is:')
```

```
p=matrix(c(5,7,2,3),2,2,TRUE)
print('P=')
print(p)
print('The diagonal matrix is:')
d=matrix(c(2,0,0,1),2,2,TRUE)
print('D=')
print(d)
print('Therefore, matrix A=PD(p^-1)')
s=solve(p)
print(p %*% d %*% s)
print('Hence, A^4=P(D^4)(P^-1)')
print(p%*%(d^4)%*%s)
```

R code Exa 5.10 10

```
#Chapter 5- Eigenvalues and Eigenvectors
#Diagonalization of a matrix
#Page No.166 / 5-10
#Prob 7
#5.3.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('the given matrix is:')
a=matrix(c(1,0,6,-1),2,2,TRUE)
print('A=')
print(a)
print('Since A is triangular, eigenvalues are the
   diagonal entries.')
cat('Eigenvalues are:',a[2,2],a[1,1])
print('for lambda=1')
print('A-1I=')
b=a-diag(2)
print(b)
print('Hence, x1=(1/3)x2 with x2 as free variable.')
print('Eigenvector corresponding to lambda=1 is:')
u1=matrix(c(1,3),2,1,TRUE)
print(u1)
print('for lambda=-1')
print('A-(-1)I=')
b=a+diag(2)
print(b)
print('Hence, x1=0 with x2 as free variable.')
print('Eigenvector corresponding to lambda=-1 is:')
u2=matrix(c(0,1),2,1,TRUE)
print(u2)
print('Thus, matrix P=')
um=cbind(u1,u2)
print(um)
print('and matrix D=')
Dm=matrix(c(1,0,0,-1),2,2,TRUE)
print(Dm)
```

R code Exa 5.11 11

```
#Chapter 5- Eigenvalues and Eigenvectors
#Diagonalization of a matrix
#Page No.167 / 5-11
#Prob 13
#5.3.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))
```

```

print('Given matrix A=')
av<-c(2,2,-1,1,3,-1,-1,-2,2)
a=matrix(av,3,3,TRUE)
print(a)
print('Given its eigen values are 5 and 1')
print('for lambda=5')
print('A-5I=')
b=a-5*diag(3)
print(b)
print('performing row operations')
zm=matrix(0,3,1,TRUE)
cm=cbind(b,zm)
print(cm)
cm[c(1,2),]=cm[c(2,1),]
print(cm)
cm[2,]=cm[2,]+3*cm[1,]
cm[3,]=cm[3,]+cm[1,]
print(cm)
cm[3,]=cm[3,]-cm[2,]
print(c)
cm[2,]=cm[2,]/cm[2,2]
cm[1,]=cm[1,]+2*cm[2,]
print(cm)
print('With x3 as free variable, x1=-x3 and x2=-x3')
print('Hence, for lambda=5 eigenvector is:')
u1=matrix(c(-1,-1,1),3,1,TRUE)
print(u1)
print('for lambda=1')
print('A-I=')
b=a-diag(3)
print(b)
print('performing row operations')
cn=cbind(b,zm)
print(cn)
cn[2,]=cn[2,]-cn[1,]
cn[3,]=cn[3,]+cn[1,]
print(cn)
print('With x2 and x3 as free variables, eigen vectors corresponding to lambda=1 are')
u2=matrix(c(-2,1,0),3,1,TRUE)
u3=matrix(c(1,0,1),3,1,TRUE)
print(u2)
print(u3)
print('Hence, matrix P=')
pm=cbind(u1,u2,u3)
print(pm)
print('and matrix D=')
Dm=matrix(c(5,0,0,0,1,0,0,0,1),3,3,TRUE)
print(Dm)
```

R code Exa 5.12 12

```
#Chapter 5- Eigenvalues and Eigenvectors
#Eigen values of matrix
#Page No.168 / 5-12
#Prob 19
#5.3.19

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

av<-c(5,-3,0,9,0,3,1,-2,0,0,2,0,0,0,0,2)
a=matrix(av,4,4,TRUE)
print(a)
eig=eigen(a)
eigv<-eig$values
print('eigen values are:')
print(eigv)
print('for lambda=2')
print('(A-2I)x=0')
zm=matrix(0,4,1,TRUE)
c1=a-2*diag(4)
m1=cbind(c1,zm)
print(m1)
m1[1,]=m1[1,]+3*m1[2,]
print(m1)
m1[1,]=m1[1,]/3
print(m1)
print('x1=x3-x4')
print('x2=-x3+2x4')
print('x3 and x4 are free')
v1=matrix(c(-1,-1,1,0),4,1,TRUE)
v2=matrix(c(-1,2,0,1),4,1,TRUE)
print('v1')
print(v1)
print('v2')
print(v2)
print('for lambda=3')
print('(A-3I)x=0')
c2=a-3*diag(4)
m2=cbind(c2,zm)
print(m2)
m2[1,]=m2[1,]/2
m2[3,]=m2[3,]+m2[2,]
m2[3,]=m2[3,]/(-2)
m2[4,]=m2[4,]+m2[3,]
m2[2,]=m2[2,]+2*m2[3,]
m2[1,]=m2[1,]+(-9/2)*m2[3,]
print(m2)
print('x1=(3/2)x2')
print('x2 is free')
print('x3 and x4 = 0')
print('choosing x2=2 produces eigen vector v3=(3,2,0,0)')
print('for lambda=5')
print('(A-5I)x=0')
c3=a-5*diag(4)
```

```
m3=cbind(c3,zm)
print(m3)
m3[1,]=m3[1,]/(-3)
m3[2,]=m3[2,]+2*m3[1,]
m3[3,]=m3[3,]+3*m3[2,]
m3[3,]=ceiling(m3[2,]/(-24))
m3[4,]=m3[4,]+3*m3[3,]
m3[2,]=m3[2,]+8*m3[3,]
m3[1,]=m3[1,]+3*m3[3,]
print(m3)
print('x1 is free')
print('x2=0')
print('x3=0')
print('x4=0')
print('a basis vector for the eigenspace is v4
   =(1,0,0,0)')
Pmv<-c(-1,-1,3,1,-1,2,2,0,1,0,0,0,0,1,0,0)
Pm=matrix(Pmv,4,4,TRUE)
print(Pm)
Dmv<-c(2,0,0,0,0,2,0,0,0,0,3,0,0,0,0,5)
Dm=matrix(Dmv,4,4,TRUE)
print(Dm)
```

R code Exa 5.13 13

```
#Chapter 5− Eigenvalues and Eigenvectors
#Page No.170 / 5−14
#Prob 29
#5.3.29

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))
```

```r

p1=matrix(c(1,1,-2,-1),2,2,TRUE)
d1=matrix(c(3,0,0,5),2,2,TRUE)
print('p1=')
print(p1)
print('d1=')
print(d1)
p2=matrix(c(-3,1,6,-1),2,2,TRUE)
print('p2=')
print(p2)
A=p1*d1*solve(p1)
A1=p2*d1*solve(p2)
print('P1D1P1^-1')
print(A)
print('P2D1P2^-1')
print(A1)
print(A==A1)
print('Hence, A=P1D1P1^-1=P2D1P2^-1')
```

R code Exa 5.14 14

```r
#Chapter 5- Eigenvalues and Eigenvectors
#Transform of matrix
#Page No.172 / 5-16
#Prob 1
#5.4.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

b1=matrix(c(3,-5),1,2,TRUE)
b2=matrix(c(-1,6),1,2,TRUE)
```

```
b3=matrix(c(0,4),1,2,TRUE)
b1t=t(b1)
b2t=t(b2)
b3t=t(b3)
cm=cbind(b1t,b2t,b3t)
print(cm)
```

R code Exa 5.15 15

```
#Chapter 5− Eigenvalues and Eigenvectors
#Eigenvalues
#Page No.173 / 5−17
#Prob 13
#5.4.13

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

a=matrix(c(0,1,-3,4),2,2,TRUE)
eig=eigen(a)
eigv<-eig$values
print(’the eigen values are’)
print(eigv)
print(’for lambda=1’)
m1=a-diag(2)
print(m1)
print(’as a basis vector, take u1=[1 1]’)
print(’for lambda=3’)
m2=a-3*diag(2)
print(m2)
print(’as a basis vector, take u1=[1 3]’)
#the matrix for lambda=3 has a calculation error in
```

```
   the textbook
```

R code Exa 5.16 16

```
#Chapter 5- Eigenvalues and Eigenvectors
#Eigenvalues
#Page No.175 / 5-19
#Prob 31
#5.4.31

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’Given matrix A=’)
av<-c(-7,-48,-16,1,14,6,-3,-45,-19)
a=matrix(av,3,3,TRUE)
print(a)
print(’and matrix P=’)
pv<-c(-3,-2,3,1,1,-1,-3,-3,0)
p=matrix(pv,3,3,TRUE)
print(p)
print(’Hence, marix D=’)
s=solve(p)
Dv=round(s %*% a %*% p)
print(Dv)
```

R code Exa 5.17 17

```
#Chapter 5- Eigenvalues and Eigenvectors
#Complex Eigenvectors
```

```
#Page No.176 / 5-20
#Prob 1
#5.5.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('Matrix A=')
a=matrix(c(1,-2,1,3),2,2,TRUE)
print(a)
print('Eigen values of A are')
eig=eigen(a)
eigv<-eig$values
print(eigv)
print('for lambda=2+i')
i=complex(real=2,imaginary=1)
print('A-(2+i)I=')
b=a-(i)*diag(2)
print(b)
print('With x2 as free variable, x1=-(1-i)x2')
print('Hence, eigenvector corresponding to lambda=2+i is:')
i1=complex(real=-1,imaginary=1)
m1=matrix(c(i1,1),2,1,TRUE)
print(m1)
i2=complex(real=-1,imaginary=-1)
print('for lambda=2-i, eigenvector is:')
m2=matrix(c(i2,1),2,1,TRUE)
print(m2)
```

R code Exa 5.18 18

```
#Chapter 5— Eigenvalues and Eigenvectors
#Scale factor of transformation
#Page No.176 / 5—20
#Prob 7
#5.5.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’Matrix A=’)
a=matrix(c(sqrt(3),-1,1,sqrt(3)),2,2,TRUE)
print(a)
print(’Eigenvalues of A are:’)
eig=eigen(a)
eigv<-eig$values
print(eigv)
print(’The scale factor associated with the transformation x to Ax is:’)
print(abs(eigv[1]))
```

R code Exa 5.19 19

```
#Chapter 5— Eigenvalues and Eigenvectors
#Page No.177 / 5—21
#Prob 13
#5.5.13

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))
```

```

i=complex(real=-1,imaginary=-1)
v=matrix(c(i,1),2,1,TRUE)
a=matrix(c(1,-2,1,3),2,2,TRUE)
print(v)
rev=Re(v)
print(rev)
imv=Im(v)
print(imv)
pm=cbind(rev,imv)
print(pm)
cm=solve(pm) %*% a %*% pm
print(cm)
```

R code Exa 5.20 20

```
#Chapter 5- Eigenvalues and Eigenvectors
#Page No.177 / 5-21
#Prob 19
#5.5.19

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

a=matrix(c(1.52,-0.7,0.56,0.4),2,2,TRUE)
print(a)
lambdap1=(1.92)/2
lambdap2=sqrt(abs(-0.3136))/2
i=complex(real=0,imaginary=lambdap2)
print(’lambda = ’)
cat(lambdap1+i)
print(’or’)
```

```
cat(lambdap1-i)
print('if x2=2')
iv=complex(real=2,imaginary=-1)
vm=matrix(c(iv,2),2,1,TRUE)
print('v=')
print(vm)
revm=Re(vm)
imvm=Im(vm)
print(revm)
print(imvm)
p=cbind(revm,imvm)
print(p)
fm=solve(p) %*% a %*% p
print(fm)
```

R code Exa 5.21 21

```
#Chapter 5- Eigenvalues and Eigenvectors
#Page No.179 / 5-23
#Prob 1
#5.6.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

v1=matrix(c(1,1),2,1,TRUE)
v2=matrix(c(-1,1),2,1,TRUE)
cat('v1=',v1)
print('')
cat('v2=',v2)
x0=matrix(c(9,1),2,1,TRUE)
nm=cbind(v1,v2,x0)
```

```
print(nm)
nm[1,]=nm[1,]+nm[2,]
nm[1,]=nm[1,]/2
nm[2,]=nm[2,]-nm[1,]
print(nm)
nv1=matrix(c(15,15),2,1,TRUE)
nv2=matrix(c((-4/3),(4/3)),2,1,TRUE)
nvv=nv1-nv2
print(nvv)
```

R code Exa 5.22 22

```
#Chapter 5– Eigenvalues and Eigenvectors
#Page No.181 / 5–25
#Prob 13
#5.6.13

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

a=matrix(c(.8,.3,-.4,1.5),2,2,TRUE)
print(a)
print(’eigen values are’)
eig=eigen(a)
print(eig$values)
zm=matrix(0,2,1,TRUE)
c1=a-1.2*diag(2)
m1=cbind(c1,zm)
print(m1)
m1[2,]=m1[2,]-m1[1,]
m1[2,]=round(m1[2,])
m1[1,]=m1[1,]/(-4)
```

```
print(m1)
```

R code Exa 5.23 23

```
#Chapter 5- Eigenvalues and Eigenvectors
#Page No.183 / 5-27
#Prob 1
#5.7.1

#clear console
cat("\014")

av<-c(-3,-1,-6,1,1,1)
a=matrix(av,2,3,TRUE)
p=a
print(a)
a[c(1,2),]<-a[c(2,1),]
print(a)
a[2,]=a[2,]+3*a[1,]
a[2,]=a[2,]/2
a[1,]=a[1,]-a[2,]
print(a)
print('c1=')
print(a[1,3])
print('c2=')
print(a[2,3])
print('x(t)=')
cat('[',a[1,3]*p[,1],']*e^4t')
cat('-[',a[2,3]*p[,2],']*e^2t')
```

R code Exa 5.24 24

```
#Chapter 5- Eigenvalues and Eigenvectors
```

```
#Page No.184 / 5-28
#Prob 19
#5.7.19

#clear console
cat("\014")

a=matrix(c(-2,(3/4),1,-1),2,2,TRUE)
print(a)
eig=eigen(a)
print('eigen values are')
print(eig$values)
v1=matrix(c(1,2),2,1,TRUE)
v2=matrix(c(-3,2),2,1,TRUE)
print('v1=')
print(v1)
print('v2=')
print(v2)
print('general solution x(t) =')
cat('c1*[',v1,']e^-.5t+c2*[',v2,']e^-2.5t')
print('')
ma1=cbind(v1,v2)
print(ma1)
mb=matrix(c(4,4),2,1,TRUE)
mc=solve(ma1) %*% mb
print(mc)
print('c1=')
print(mc[1])
print('c2=')
print(mc[2])
x1=mc[1]*v1
x2=mc[2]*v2
print('x(t)=')
cat('[',x1,']*e^-.5t')
cat('-[',x2,']*e^-2.5t')
```

Chapter 6

Orthogonality And Least Squares

R code Exa 6.1 1

```
#Chapter 6- Orthogonality and Least Squares
#Dot product of vectors
#Page No.193 / 6-1
#Prob 1
#6.1.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('Vectors u and v are:')
u=matrix(c(-1,2),2,1,TRUE)
v=matrix(c(4,6),2,1,TRUE)
print(u)
print(v)
u2=u * u
vu=v * u
```

```
print('u . u')
print(u2[1]+u2[2])
print('v . u')
print(vu[1]+vu[2])
print('Projection of v on u=(u.v)/(v.v)')
a=t(u) %*% v
b=t(u) %*% u
p=a/b
print('=')
print(p)
```

R code Exa 6.2 2

```
#Chapter 6- Orthogonality and Least Squares
#Norm of a vector
#Page No.193 / 6-1
#Prob 7
#6.1.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('w=')
w=matrix(c(3,-1,-5),3,1,TRUE)
print(w)
wv=w[1]^2+w[2]^2+w[3]^2
print('||w||=sqrt(9+1+25)')
print(sqrt(wv))
```

R code Exa 6.3 3

```
#Chapter 6– Orthogonality and Least Squares
#Distance between two points
#Page No.193 / 6–1
#Prob 13
#6.1.13

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’Vector x and y are:’)
x=matrix(c(10,-3),2,1,TRUE)
y=matrix(c(-1,-5),2,1,TRUE)
print(x)
print(y)
xyv=((x[1]-y[1])^2+(x[2]-y[2])^2)
print(’||x–y||=sqrt(125)’)
print(’=’)
print(sqrt(xyv))
```

R code Exa 6.4 4

```
#Chapter 6– Orthogonality and Least Squares
#Orthogonality of vectors
#Page No.195 / 6–3
#Prob 1
#6.2.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))
```

```
print('To verify if u v and w are orthogonal')
u=matrix(c(-1,4,-3),3,1,TRUE)
v=matrix(c(5,2,1),3,1,TRUE)
w=matrix(c(3,-4,-7),3,1,TRUE)
print('u')
print(u)
print('v')
print(v)
print('w')
print(w)
print('u.v=')
print(t(v) %*% u)
print('u.w=')
print(t(u) %*% w)
print('Since u.w is not equal to zero, the set {u v w} is not orthogonal.')
```

R code Exa 6.5 5

```
#Chapter 6- Orthogonality and Least Squares
#Orthogonal basis
#Page No.195 / 6-3
#Prob 7
#6.2.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('vectors u1 u2 and x are:')
u1=matrix(c(2,-3),2,1,TRUE)
u2=matrix(c(6,4),2,1,TRUE)
```

```
x=matrix(c(9,-7),2,1,TRUE)
print(’x’)
print(x)
print(’u1’)
print(u1)
print(’u2’)
print(u2)
print(’u1.u2=’)
print(t(u1) %*% u2)
print(’u1.u2=0, {u1 u2} is an orthogonal set’)
print(’Hence {u1 u2} forms a basis of R2’)
print(’x can be written as: x=a*u1+b*u2’)
print(’where a=(x.u1)/(u1.u1)’)
u1p1=(x * u1)
u1p2=(u1 * u1)
u1m=(u1p1[1]+u1p1[2])/(u1p2[1]+u1p2[2])
u2p1=(x * u2)
u2p2=(u2 * u2)
u2m=(u2p1[1]+u2p1[2])/(u2p2[1]+u2p2[2])
x1=(u1m * u1)
x2=(u2m * u2)
x=x1+x2
print(’x1=’)
print(x1)
print(’x2=’)
print(x2)
print(’x=’)
print(x)
```

R code Exa 6.6 6

```
#Chapter 6- Orthogonality and Least Squares
#Projection of vectors
#Page No.195 / 6-3
#Prob 13
```

```
#6.2.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('Vectors y and u are:')
y=matrix(c(2,3),2,1,TRUE)
u=matrix(c(4,-7),2,1,TRUE)
print('u')
print(u)
print('y')
print(y)
print('The orthogonal projection of y on u=((y.u)/(u.u))*u')
a=t(y) %*% u
b=t(u) %*% u
c1=(a/b)
c= c1[1] * u
print('=')
print(c)
print('The component of y orthogonal to u is:')
print(y-c)
print('Thus y=')
print(c+(y-c))
```

R code Exa 6.7 7

```
#Chapter 6— Orthogonality and Least Squares
#Orthonormal vectors
#Page No.196 / 6—4
#Prob 19
#6.2.19
```

```

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’given vectors u and v are:’)
u=matrix(c(-.6,.8),2,1,TRUE)
v=matrix(c(.8,.6),2,1,TRUE)
print(’u’)
print(u)
print(’v’)
print(v)
uvv=u*v
uvv=uvv[1]+uvv[2]
print(’u.v=’)
print(uvv)
uuv=u*u
uuv=uuv[1]+uuv[2]
print(’u.u=’)
print(uuv)
vvv=v*v
vvv=vvv[1]+vvv[2]
print(’v.v=’)
print(vvv)
print(’Hence, {u v} is an orthogonal set.’)
print(’||u||=1 and ||v||=1’)
print(’Thus, {u v} is an orthonormal set’)
```

R code Exa 6.8 8

```
#Chapter 6− Orthogonality and Least Squares
#Orthogonal projection
#Page No.198 / 6−6
```

```
#Prob 1
#6.3.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('Given vectors are:')
u1=matrix(c(0,1,-4,-1),4,1,TRUE)
print('u1')
print(u1)
u2=matrix(c(3,5,1,1),4,1,TRUE)
print('u2')
print(u2)
u3=matrix(c(1,0,1,-4),4,1,TRUE)
print('u3')
print(u3)
u4=matrix(c(5,-3,-1,1),4,1,TRUE)
print('u4')
print(u4)
x=matrix(c(10,-8,2,0),4,1,TRUE)
print('x')
print(x)
print('The vector in span{u4}=((x.u4)/(u4.u4))*u4')
a1=t(x) %*% u4
a2=t(u4) %*% u4
av=(a1/a2)
print(av[1]*u4)
print('Therefore, the vector in span{u1 u2 u3}=x-2*u4')
print(x-2*u4)
```

R code Exa 6.9 9

```
#Chapter 6− Orthogonality and Least Squares
#Orthogonal projection
#Page No.198 / 6−6
#Prob 7
#6.3.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print(’Vectors u1 u2 and y are’)
u1=matrix(c(1,3,-2),3,1,TRUE)
u2=matrix(c(5,1,4),3,1,TRUE)
y=matrix(c(1,3,5),3,1,TRUE)
print(’y=’)
print(y)
print(’u2=’)
print(u2)
print(’u1=’)
print(u1)
print(’u1.u2=’)
a=t(u1) %*% u2
print(’=’)
print(a)
print(’Hence, {u1 u2} form an orthogonal basis.’)
print(’Let W=span{u1 u2}’)
print(’Therefore, projection of y on W is:’)
print(’((y.u1)/(u1.u1))*u1+((y.u2)/(u2.u2))*u2’)
a1=t(y) %*% u1
a2=t(u1) %*% u1
b1=t(y) %*% u2
b2=t(u2) %*% u2
print(’=’)
bv=(b1/b2)
av=(a1/a2)
print(bv[1]*u2)
print('Therefore, projection of y on W is:')
ypy=y-bv[1]*u2
print(ypy)
ypy=ypy/ypy[2]
print(ypy)
```

R code Exa 6.10 10

```
#Chapter 6– Orthogonality and Least Squares
#Orthogonal projection
#Page No.199 / 6–7
#Prob 13
#6.3.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('Given vectors are:')
v1=matrix(c(2,-1,-3,1),4,1,TRUE)
v2=matrix(c(1,1,0,-1),4,1,TRUE)
z=matrix(c(3,-7,2,3),4,1,TRUE)
print('v1=')
print(v1)
print('v2=')
print(v2)
print('z=')
print(z)
a=t(v1) %*% v2
print('v1.v2=')
print(a)
if(a==0){
print('v1 and v2 are orthogonal')
}
print('By best approximation theorem, closest point in span{v1 v2} to z is the orthogonal projection')
print('=((z.v1)/(v1.v1))*v1+((z.v2)/(v2.v2))*v2')
a1=t(z) %*% v1
a2=t(v1) %*% v1
b1=t(z) %*% v2
b2=t(v2) %*% v2
av=(a1/a2)
bv=(b1/b2)
print('=')
print(av[1]*v1)
print('+')
print(bv[1]*v2)
print('=')
zb=(av[1]*v1)+(bv[1]*v2)
print(zb)
zn=z-zb
print(zn)
zn=zn/4
print(zn)
```

R code Exa 6.11 11

```
#Chapter 6- Orthogonality and Least Squares
#Orthogonal decomposition theorem
#Page No.200 / 6-8
#Prob 19
#6.3.19

#clear console
cat("\014")

#clear variables
```

```
rm(list=ls(all=TRUE))

print('By orthogonal decomposition theorem,')
print('u3 is the sum of a vector in W=span{u1 u2} and a vector v orthogonal to W')
print('To find v, given u1 and u2')
u1=matrix(c(1,1,-2),3,1,TRUE)
u2=matrix(c(5,-1,2),3,1,TRUE)
print('u1=')
print(u1)
print('u2=')
print(u2)
print('Projection of u3 on W')
print('= (-1/3)*u1+(1/15)*u2')
print('=')
print((-1/3)*u1+(1/15)*u2)
print('v= u3-(projection of u3 on W)')
print('=')
m1=matrix(c(0,0,1),3,1,TRUE)
print(m1)
print('-')
print((-1/3)*u1+(1/15)*u2)
print(m1-((-1/3)*u1+(1/15)*u2))
```

R code Exa 6.12 12

```
#Chapter 6- Orthogonality and Least Squares
#Gram Schimdt Orthogonalisation
#Page No.202 / 6-10
#Prob 1
#6.4.1

#clear console
cat("\014")

```

```
#clear variables
rm(list=ls(all=TRUE))

print('to orthogonalise the given vectors using Gram-Schimdt orthogonalisation')
x1=matrix(c(3,0,-1),3,1,TRUE)
x2=matrix(c(8,5,-6),3,1,TRUE)
print('x1=')
print(x1)
print('x2=')
print(x2)
print('Let v1=x1')
v1=x1
print('v2=x2-((x2.v1)/(v1.v1))*v1')
a1=t(x2) %*% v1
a2=t(v1) %*% v1
av=(a1/a2)
p=av[1]*v1
v2=x2-p
print('=')
print(x2)
print('-')
print(p)
print('=')
print(v2)
print('Thus, an orthogonal basis is:')
print(v1)
print(v2)
```

R code Exa 6.13 13

```
#Chapter 6- Orthogonality and Least Squares
#Gram Schimdt Orthogonalisation
#Page No.202 / 6-10
#Prob 7
```

```
#6.4.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('to orthogonalise the given vectors using Gram-Schimdt orthogonalisation')
x1=matrix(c(2,-5,1),3,1,TRUE)
x2=matrix(c(4,-1,2),3,1,TRUE)
print('x1=')
print(x1)
print('x2=')
print(x2)
print('Let v1=x1')
v1=x1
print('v2=x2-((x2.v1)/(v1.v1))*v1')
a1=t(x2) %*% v1
a2=t(v1) %*% v1
av=(a1/a2)
p=av[1]*v1
v2=x2-p
print('=')
print(x2)
print('-')
print(p)
print('=')
print(v2)
print('Thus, an orthogonal basis is:')
print(v1)
print(v2)
print('Normalizing v1 and v2, we get')
s1=sqrt(v1[1,1]^2+v1[2,1]^2+v1[3,1]^2)
s2=sqrt(v2[1,1]^2+v2[2,1]^2+v2[3,1]^2)
print(v1/s1)
print(v2/s2)
```

R code Exa 6.14 14

```r
#Chapter 6- Orthogonality and Least Squares
#QR decomposition of a matrix
#Page No.202 / 6-10
#Prob 13
#6.4.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('QR decomposition of a matrix')
print('given matrix A=')
a=matrix(c(5,9,1,7,-3,-5,1,5),4,2,TRUE)
print(a)
print('given matrix Q=')
m1=matrix(c(5,-1,1,5,-3,1,1,3),4,2,TRUE)
q=(1/6)*m1
print(q)
print('Therefore, R=')
s=t(q) %*% a
print(s)
qr=q %*% s
print(qr)
```

R code Exa 6.15 15

```r
#Chapter 6- Orthogonality and Least Squares
```

```
#Least square solution
#Page No.205 / 6-13
#Prob 1
#6.5.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('The co-efficient matrix is:')
a=matrix(c(-1,2,2,-3,-1,3),3,2,TRUE)
print('A=')
print(a)
print('The RHS is:')
b=matrix(c(4,1,2),3,1,TRUE)
print(b)
print('Product of transpose of A and A=')
p1=t(a) %*% a
print(p1)
print('Product of transpose of A and b=')
p2=t(a) %*% b
print(p2)
print('Hence, the solution is:')
p=solve(p1) %*% p2
print(p)
```

R code Exa 6.16 16

```
#Chapter 6- Orthogonality and Least Squares
#Least square solution
#Page No.206 / 6-14
#Prob 7
#6.5.7
```

```

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('The co-efficient matrix is:')
a=matrix(c(1,-2,-1,2,0,3,2,5),4,2,TRUE)
print('A=')
print(a)
print('The RHS is:')
b=matrix(c(3,1,-4,2),4,1,TRUE)
print('b=')
print(b)
print('Product of transpose of A and A=')
p1=t(a) %*% a
print(p1)
print('Product of transpose of A and b=')
p2=t(a) %*% b
print(p2)
print('Forming an augmented matrix to solve the normal equations')
p=cbind(p1,p2)
print(p)
print('performing row operations')
print('R2=R2-R1')
p[2,]=p[2,]-p[1,]
print(p)
print('R1=R1/6 and R2=R2/36')
p[1,]=p[1,]/6
p[2,]=p[2,]/36
print(p)
print('R1=R1-R2')
p[1,]=p[1,]-p[2,]
print(p)
print('Hence, the solution is:')
print(p[,3])
```

```r
x=p[,3]
print('The least square error is =||Ax-b||')
print('Ax-b=')
print(a %*% x-b)
c=a %*% x-b
s=0
il<-c(1:4)
for (i in il)
{
s=s+c[i,1]^2
}
print('||Ax-b||=')
print(sqrt(s))
```

R code Exa 6.17 17

```r
#Chapter 6- Orthogonality and Least Squares
#Least square solution
#Page No.207 / 6-15
#Prob 13
#6.5.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('To determine if u is the least square solution to Ax=b')
print('Given')
a=matrix(c(3,4,-2,1,3,4),3,2,TRUE)
print('A=')
print(a)
b=matrix(c(11,-9,5),3,1,TRUE)
```

```
print('b=')
print(b)
u=matrix(c(5,-1),2,1,TRUE)
v=matrix(c(5,-2),2,1,TRUE)
print('u=')
print(u)
print('v=')
print(v)
print('Au=')
print(a %*% u)
c=b-a %*% u
print('b–Au=')
print(c)
print('||b–Au||= ')
print(sqrt(c[1,1]^2+c[2,1]^2+c[3,1]^2))
print('Av=')
print(a %*% v)
d=b-a %*% v
print('b–Av=')
print(d)
print('||b–Av||= ')
print(sqrt(d[1,1]^2+d[2,1]^2+d[3,1]^2))
print('Since Av is more closer to A than Au, u is not the least square solution.')
```

R code Exa 6.18 18

```
#Chapter 6– Orthogonality and Least Squares
#Least squares line
#Page No.209 / 6–17
#Prob 1
#6.6.1

#clear console
cat("\014")
```

```
#clear variables
rm(list=ls(all=TRUE))

print('To obtain a least sqaure line from the given data')
print('Placing the x coordinates of the data in second column of matrix X we get:')
x=matrix(c(1,0,1,1,1,2,1,3),4,2,TRUE)
print('X=')
print(x)
print('Placing the y coordinates in y vector')
y=matrix(c(1,1,2,2),4,1,TRUE)
print('y=')
print(y)
print('Product of transpose of X and X=')
p1=t(x) %*% x
print(p1)
print('Product of transpose of X and y=')
p2=t(x) %*% y
print(p2)
print('The least square solution =')
print(solve(p1) %*% p2)
p=solve(p1) %*% p2
print('Hence, the least square line is:')
print('y')
print('=')
print(p[1,1])
print('+')
print(p[2,1])
print('x')
```

R code Exa 6.19 19

```
#Chapter 6- Orthogonality and Least Squares
```

```
#Page No.212 / 6-20
#Prob 1
#6.7.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

ipx=matrix(c(4,1,1),3,1,TRUE)
ipy=matrix(c(5,1,1),3,1,TRUE)
x=matrix(c(1,1),2,1,TRUE)
y=matrix(c(5,-1),2,1,TRUE)
print('x')
print(x)
print('y')
print(y)
print('x^2=')
xs=sqrt((ipx[1]*x[1]*x[1])+(ipy[1]*x[2]*x[2]))
ys=sqrt((ipx[1]*y[1]*y[1])+(ipy[1]*y[2]*y[2]))
print('||x||^2')
print(xs)
print('||y||^2')
print(ys)
print('||<x,y>||^2')
xys=((ipx[1]*x[1]*y[1])+(ipy[1]*x[2]*y[2]))^2
print(xys)
```

R code Exa 6.20 20

```
#Chapter 6- Orthogonality and Least Squares
#Page No.214 / 6-22
#Prob 25
#6.7.25
```

```

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

integrand <-function(t) {1*t}
print(’<t,1>=’)
v1=integrate(integrand,lower=-1,upper=1)
print(v1)
integrand1 <-function(t) {1*(t*t)}
print(’<t^2,1>=’)
v2=integrate(integrand1,lower=-1,upper=1)
print(v2)
integrand2 <-function(x) {1*1}
print(’<1,1>=’)
v3=integrate(Vectorize(integrand2),lower=-1,upper=1)
print(v3)
integrand3 <-function(t) {(t*t)*t}
print(’<t^2,t>=’)
v4=integrate(integrand3,lower=-1,upper=1)
print(v4)
integrand4 <-function(t) {(t*t)}
v5=integrate(integrand4,lower=-1,upper=1)
print(’Thus projwt^2=’)
projt=(((v2$value/v3$value)*1)+((v4$value/v5$value)))
print(projt)
print(’A polynomial orthogonal to W is t^2-projt^2=’)
print(’t^2-(1/3)’)
print(’thus the polynomials 1,t^2and 3t^2-1 form an orthogonal basis for Span(1,t,t^2)’)
```

R code Exa 6.21 21

```
#Chapter 6- Orthogonality and Least Squares
#Applications of Inner Product Spaces
#Page No.215 / 6-23
#Prob 1
#6.8.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

xv<-c(1,-2,1,-1,1,0,1,1,1,2)
x=matrix(xv,5,2,TRUE)
print('X=')
print(x)
yv<-c(0,0,2,4,4)
y=matrix(yv,5,1,TRUE)
print(y)
Wv<-c(1,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,1)
W=matrix(Wv,5,5,TRUE)
WX=W %*% x
print('WX=')
print(WX)
Wy=W %*% y
print('Wy=')
print(Wy)
wxtwx=t(WX) %*% WX
print('(WX)^T*(WX)=')
print(wxtwx)
wxtwy=t(WX) %*% Wy
print('(WX)^T*(Wy)=')
print(wxtwy)
betav=solve(wxtwx) %*% (wxtwy)
```

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

R code Exa 6.22 22

```
#Chapter 6- Orthogonality and Least Squares
#Applications of Inner Product Spaces
#Page No.216 / 6-24
#Prob 7
#6.8.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

integrand <-function(kt) {cos(kt)*cos(kt)}
print('||coskt||^2=')
v1=integrate(integrand,lower=0,upper=360)
print(round(v1$value))

integrand <-function(kt) {sin(kt)*sin(kt)}
print('||sinkt||^2=')
v1=integrate(integrand,lower=0,upper=360)
print(round(v1$value))
```

R code Exa 6.23 23

```
#Chapter 6- Orthogonality and Least Squares
#Applications of Inner Product Spaces
```

```
#Page No.217 / 6-25
#Prob 9
#6.8.9

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

integrand <-function(t) {(2*180)-t}
print('The fourier coefficients of f are:')
v1=integrate(integrand,lower=0,upper=360)
v1v=(1/2)*(1/180)*v1$value
print('a0/2=')
print(v1v)
print('for k=1')
integrand1 <-function(t) {((2*180)-t)*cos(t)}
v2=integrate(integrand1,lower=0,upper=360)
v2v=(1/180)*v2$value
print('ak=')
print(round(v2v))
integrand2 <-function(t) {((2*180)-t)*sin(t)}
v3=integrate(integrand2,lower=0,upper=360)
v3v=(1/180)*v3$value
print('bk=')
print(round(v3v))
print('The third-order Fourier approximation to f is : ')
print('180 + 2sint + sin2t + (2/3)sint3t')
```

Chapter 7

Symmetric Matrices

R code Exa 7.1 1

```
#Chapter 7 - Symmetric Matrices And Quadratic Forms
#Symmetric Matrices
#Page No.222 / 7-2
#Prob 1
#7.1.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('To check if the given 2X2 matrix is symmetric')

#Declaration of matrix
a = matrix(
  c(3,5,5,-7),
  nrow = 2,
  ncol = 2,
  byrow = TRUE)
```

```

print('A=')
print(a)

#Check for symmetry
if(a[1,2]==a[2,1]){
  print('A is a symmetric matrix because the (1,2) and(2,1) entries match.')
}else{
  print('A is not a symmetric matrix')
}
```

R code Exa 7.2 2

```
#Chapter 7 - Symmetric Matrices And Quadratic Forms
#Orthogonal Matrix
#Page No.222 / 7-2
#Prob 7
#7.1.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('To show that the given matrix P is orthogonal.')
p = matrix(
  c(0.6,0.8,0.8,-0.6),
  nrow = 2,
  ncol = 2,
  byrow = TRUE)

print('P=')
print(p)
print('P is composed of two vectors.')
p1 = matrix(
  c(0.6,0.8),
  nrow = 2,
  ncol = 1,
  byrow = TRUE)

p2 = matrix(
  c(0.8,-0.6),
  nrow = 2,
  ncol = 1,
  byrow = TRUE)

print('p1=')
print(p1)
print('p2=')
print(p2)
print('To show that the columns are orthonormal')
print('p1.p2=')
s1=p1 * p2
s=s1[1]+s1[2]
r=t(p1)
print(s)
if(s==0)
  print('The columns of P are orthonormal')
end
print('||p1||^2=')
print(sqrt(p[1,1]^2+p[2,1]^2))
print('||p2||^2=')
print(sqrt(p[1,2]^2+p[2,2]^2))
print('Hence, ||p1||=||p2||=1. Thus P is an orthogonal matrix')
```

R code Exa 7.3 3

```
#Chapter 7 – Symmetric Matrices And Quadratic Forms
#PD decomposition of a matrix
#Page No.222 / 7–2
#Prob 13
#7.1.13

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

print('To diagonalize the given matrix A')

#Declaration of matrix
a = matrix(
  c(3,1,1,3),
  nrow = 2,
  ncol = 2,
  byrow = TRUE)

print('A=')
print(a)
eig=eigen(a)
values <- eig$values
print('Eigen values of A are:')
print(values)
print('for lambda=4')
print('A–4I=')
print(a-4*diag(2))
b=a-4*diag(2)
print('To find the eigenvector, form an augmented matrix.')
zm=matrix(
  c(0,0),
  nrow=2,
  ncol=1,
  byrow=TRUE)
```

```

c=cbind(b,zm)
print('performing row operations')
print(c)

c[2,]=c[2,]+c[1,]
print(c)
print('With x2 as free variable, x1=x2')
print('Hence a basis for the eigenspace is:')
d=matrix(
  c(1,1),
  nrow=2,
  ncol=1,
  byrow=TRUE)

print(d)
print('Upon normalizing')
print(d/(sqrt(2)))
u1=d/(sqrt(2))
print('for lambda=2')
print('A-2I=')
b=a-2*diag(2)
print(b)
print('To find the eigenvector, form an augmented matrix.')
c=cbind(b,zm)
print('performing row operations')
print(c)
c[2,]=c[2,]-c[1,]
print(c)
print('With x2 as free variable, x1=-x2')
print('Hence a basis for the eigenspace is:')
d=matrix(
  c(-1,1),
  nrow=2,
  ncol=1,
  byrow=TRUE)

```

```
print(d)
print('Upon normalizing')
print(d/(sqrt(2)))
u2=d/(sqrt(2))
print('Matrix P=')
p=cbind(u1,u2)
print(p)
print('The corresponding matrix D=')

D1=matrix(
  c(values[1],0),
  nrow=1,
  ncol=2,
  byrow=TRUE)

D2=matrix(
  c(0,values[2]),
  nrow=1,
  ncol=2,
  byrow=TRUE)


D = matrix(
  c(D1,D2),
  nrow=2,
  ncol=2,
  byrow = TRUE)
print(D)
```

R code Exa 7.4 4

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

```
#7.1.19

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

print('PD decomposition of a matrix A')
av<-c(3,-2,4,-2,6,2,4,2,3)
a=matrix(av,3,3,TRUE)
print('A=')
print(a)
print('Eigenvalues of A are')
eig=eigen(a)
evv=eig$values
print(evv)
cat('for lambda =',evv[1])
print('A-(lambda)I=')
b=a-evv[1]*diag(3)
print(b)
print('To find eigenvector, form an augmented matrix')
zm=matrix(0,3,1,TRUE)
cm=cbind(b,zm)
print(cm)
print('performing row operations')
cm[2,]=cm[2,]-(cm[2,1]/cm[1,1])*cm[1,]
cm[3,]=cm[3,]-(cm[3,1]/cm[1,1])*cm[1,]
cm[1,]=cm[1,]/-4
print(cm)
print('With x2 and x3 as free variables, we get two vectors.')
print('x1=-.5x2+x3')
print('Thus, the two vectors are')
v1=matrix(c(-1,2,0),3,1,TRUE)
v2=matrix(c(1,0,1),3,1,TRUE)
print('v1=')
print(v1)
print('v2=')
print(v2)
print('Orthogonalizing v1 and v2')
print('Let x1=v1')
print('x2=v2-((v2.v1)/(v1.v1))*v1')
x1=v1
a1=t(v2) %*% v1
a2=t(v1) %*% v1
av=(a1/a2)
x2=v2-av[1]*v1
x1=x1/(sqrt(x1[1,1]^2+x1[2,1]^2+x1[3,1]^2))
u1=x1
x1=x2/(sqrt(x2[1,1]^2+x2[2,1]^2+x2[3,1]^2))
print(x2)
x2=x2*5
print(x2)
print('An orthonormal basis is:')
print('u1=')
print(u1)
print('u2=')
print(x1)
cat('for lambda=',evv[3])
print('A-(lambda)I=')
b1=a-evv[3]*diag(3)
print(b1)
print('To find eigenvector, form an augmented matrix')
cm1=cbind(b1,zm)
print(cm1)
print('performing row operations')
cm1[2,]=cm1[2,]/2
cm1[c(1,2),]=cm1[c(2,1),]
cm1[1,]=cm1[1,]/-1
print(cm1)
cm1[2,]=round(cm1[2,]-5*cm1[1,])
cm1[3,]=cm1[3,]-4*cm1[1,]
print(cm1)
```

```
cm1[3,]=round(cm1[3,]-cm1[2,])
cm1[2,]=cm1[2,]/18
print(cm1)
cm1[1,]=round(cm1[1,]+4*cm1[2,])
print(cm1)
print(’With x3 as free variable’)
print(’x1=x3 and x2=−.5x3’)
print(’Thus a basis for the eigenspace is:’)
v3=matrix(c(-2,-1,2),3,1,TRUE)
print(v3)
print(’upon normalizing’)
v3=v3/(sqrt(v3[1,1]^2+v3[2,1]^2+v3[3,1]^2))
print(v3)
print(’Thus, matrix P=’)
print(cbind(u1,x1,v3))
print(’Corresponding matrix D=’)
D1=cbind(evv[1],0,0)
D2=cbind(0,evv[2],0)
D3=cbind(0,0,evv[3])
D=rbind(D1,D2,D3)
print(D)
```

R code Exa 7.5 5

```
#Chapter 7 − Symmetric Matrices And Quadratic Forms
#Quadratic Form
#Page No.226 / 7−6
#Prob 1
#7.2.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))
```

```

print('given matrix A and vector x')
a = matrix(
  c(5,(1/3),(1/3),1),
  nrow = 2,
  ncol = 2,
  byrow = TRUE)

print('A=')
print(a)

x = matrix(
  c(6,1),
  nrow = 2,
  ncol = 1,
  byrow = TRUE)
print('x=')
print(x)

print('Product of transpose of x and A and x=')
tx=t(x)
p=tx %*% a %*% x
print(p)
print('New value of vector x=')

x = matrix(
  c(1,3),
  nrow = 2,
  ncol = 1,
  byrow = TRUE)

print(x)
print('Product of transpose of x and A and x=')
tx1=t(x)
p=tx1  %*% a %*% x
print(p)
```

R code Exa 7.6 6

```r
#Chapter 7 - Symmetric Matrices And Quadratic Forms
#Page No.226 / 7-6
#Prob 7
#7.2.7

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

a=matrix(c(1,5,5,1),2,2,TRUE)
print('A=')
print(a)
eig=eigen(a)
eigv<-eig$values
print('Eigen values are:')
print(eigv)
ev=eig$vectors
ev1=(ev[,1]/(ev[1,1]))
print('For lambda=6')
print('an eigen vector is')
print(ev1)
print('normalized u1=')
print(eig$vectors[,1])
ev2=(ev[,2]/(ev[1,1]))
print('For lambda=-4')
print('an eigen vector is')
print(ev2)
print('normalized u2=')
print(eig$vectors[,2])
pm=eig$vectors
```

```
print('P=')
print(pm)
Dm=t(pm) %*% a %*% pm
print('D=')
print(round(Dm))
```

R code Exa 7.7 7

```
#Chapter 7 - Symmetric Matrices And Quadratic Forms
#Page No.227 / 7-7
#Prob 13
#7.2.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

a=matrix(c(1,-3,-3,9),2,2,TRUE)
print('A=')
print(a)
eig=eigen(a)
eigv<-round(eig$values)
print('Eigen values are:')
print(eigv)
ev=eig$vectors
ev1=(ev[,1]/(ev[1,1]))
print('For lambda=10')
print('an eigen vector is')
print(ev1)
print('normalized u1=')
print(-eig$vectors[,1])
ev2=(ev[,2]/(ev[1,1]))
print('For lambda=0')
print(’an eigen vector is’)
print(ev2)
print(’normalized u2=’)
print(-eig$vectors[,2])
pm=-eig$vectors
print(’P=’)
print(pm)
Dm=t(pm) %*% a %*% pm
print(’D=’)
print(round(Dm))
```

R code Exa 7.8 8

```
#Chapter 7 - Symmetric Matrices And Quadratic Forms
#Page No.229 / 7-9
#Prob 1
#7.3.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

av<-c(5,2,0,2,6,-2,0,-2,7)
a=matrix(av,3,3,TRUE)
print(’A=’)
print(a)
eig=eigen(a)
eigv<-eig$values
print(’eigen values are:’)
print(eigv)
print(’for lambda=9’)
print(’normalized eigenvector u1=’)
print(-eig$vectors[,1])
print(’for lambda=6’)
print(’normalized eigenvector u2=’)
print(eig$vectors[,2])
print(’for lambda=3’)
print(’normalized eigenvector u3=’)
print(-eig$vectors[,3])
print(’x=Py’)
print(’P=’)
p1=-eig$vectors[,1]
p2=eig$vectors[,2]
p3=-eig$vectors[,3]
p=cbind(p1,p2,p3)
print(p)
```

R code Exa 7.9 9

```
#Chapter 7 – Symmetric Matrices And Quadratic Forms
#Page No.229 / 7−9
#Prob 7
#7.3.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

av<-c(-2,2,0,2,-1,2,0,2,0)
a=matrix(av,3,3,TRUE)
print(’A=’)
print(a)
eig=eigen(a)
eigv<-eig$values
print(’eigen values are:’)
print(eigv)
print('2 is the greatest eigen value')
eigve<-eig$vectors[,1]
v1=eigve/eigve[3]
print('v1=')
print(v1)
v1=v1*2
print('scaled v1=')
print(v1)
print('u1=')
print(eigve)
```

R code Exa 7.10 10

```
#Chapter 7 - Symmetric Matrices And Quadratic Forms
#The Singular Value Decomposition
#Page No.232 / 7-12
#Prob 1
#7.4.1

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

a=matrix(c(1,0,0,-3),2,2,TRUE)
print('A=')
print(a)
print('ATA')
ata=t(a) %*% a
print(ata)
eig=eigen(ata)
eigv<-eig$values
print('eigen values are:')
print(eigv)
print('eigen vectors are:')
print('for lambda=9')
print(eig$vectors[,1])
print('for lambda=1')
print(-eig$vectors[,2])
r1=eig$vectors[,1]
r2=-eig$vectors[,2]
V=cbind(r1,r2)
print('V=')
print(V)
sv1=sqrt(eigv[1])
sv2=sqrt(eigv[2])
print('Singular values are:')
print(sv1)
print(sv2)
sigmam=matrix(c(sv1,0,0,sv2),2,2,TRUE)
print('The matrix is the same shape as A=')
print(sigmam)
Av1=a %*% r1
Av2=a %*% r2
print('Av1=')
print(Av1)
print('Av2=')
print(Av2)
u1=Av1/sv1
u2=Av2/sv2
print('normalize:')
print('u1=')
print(u1)
print('u2=')
print(u2)
um=cbind(u1,u2)
print(um)
print('A=')
print(um)
print('*')
print(sigmam)
print('*')
print(t(V))
```

R code Exa 7.11 11

```
#Chapter 7 – Symmetric Matrices And Quadratic Forms
#The Singular Value Decomposition
#Page No.232 / 7−12
#Prob 7
#7.4.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

a=matrix(c(2,-1,2,2),2,2,TRUE)
print(’A=’)
print(a)
ata=t(a) %*% a
print(’ATA=’)
print(ata)
eig=eigen(ata)
print(’eigen values are:’)
eigv<-eig$values
print(eigv)
eigve1<-(-eig$vectors[,1])
eigve2<-(-eig$vectors[,2])
print(’for lambda=9’)
print(eigve1)
print(’for lambda=4’)
print(eigve2)
V=cbind(eigve1,eigve2)
print(V)
sv1=sqrt(eigv[1])
```

```
sv2=sqrt(eigv[2])
print('Singular values are')
print(sv1)
print(sv2)
c1<-c(sv1,0)
c2<-c(0,sv2)
sigmam=cbind(c1,c2)
av1=a %*% eigve1
av2=a %*% eigve2
print('Av1')
print(av1)
print('Av2')
print(av2)
print('Check orthogonality')
co=av1*av2
co=co[1]+co[2]
print('Av1.Av2=')
print(co)
u1=av1/sv1
u2=av2/sv2
print('u1=')
print(u1)
print('u2=')
print(u2)
um=cbind(u1,u2)
print(um)
print('A=')
print(um)
print('*')
print(sigmam)
print('*')
print(t(V))
```

R code Exa 7.12 12

```
#Chapter 7 - Symmetric Matrices And Quadratic Forms
#The Singular Value Decomposition
#Page No.233 / 7-13
#Prob 13
#7.4.13

#clear console
cat("\014")

#clear variables
rm(list=ls(all=TRUE))

av<-c(3,2,2,2,3,-2)
a=matrix(av,2,3,TRUE)
print(a)
aat=a %*% t(a)
print(aat)
eig=eigen(aat)
eigv<-eig$values
print('eigen values are')
print(eigv)
v1=eig$vectors[,1]
v2=eig$vectors[,2]
print('eigen vectors are')
print('for lambda=25')
print(v1)
print('for lambda=9')
print(v2)
V=cbind(v1,v2)
print(V)
sv1=sqrt(eigv[1])
sv2=sqrt(eigv[2])
print('singular values are')
print(sv1)
print(sv2)
zm=matrix(0,1,2,TRUE)
c1=matrix(c(sv1,0),1,2,TRUE)
c2=matrix(c(0,sv2),1,2,TRUE)
```

```
sigmam=rbind(c1,c2,zm)
print(sigmam)
print(t(a) %*% V)
u1=(t(a) %*% v1)/sv1
u2=(t(a) %*% v2)/sv2
print(’u1=’)
print(u1)
print(’u2=’)
print(u2)
u3=matrix(c((-2/3),(2/3),(1/3)),3,1,TRUE)
print(’u3=’)
print(u3)
u=cbind(u1,u2,u3)
print(’A=’)
print(V)
print(’*’)
print(t(sigmam))
print(’*’)
print(t(u))
```

R code Exa 7.13 13

```
#Chapter 7 – Symmetric Matrices And Quadratic Forms
#Applications to Image Processing and Statistics
#Page No.237 / 7−17
#Prob 1
#7.5.1

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

xv<-c(19,22,6,3,2,20,22,6,9,15,13,5)
```

```r
x=matrix(xv,2,6,TRUE)
B=x
print(x)
M=matrix(c(12,10),2,1,TRUE)
print(M)
B[1,]=x[1,] - M[1]
B[2,]=x[2,] - M[2]
print(B)
#There is a calculation error in the textbook
S1=B %*% t(B)
S=S1/(dim(B)[2]-1)
print(S)
```

R code Exa 7.14 14

```r
#Chapter 7 – Symmetric Matrices And Quadratic Forms
#Applications to Image Processing and Statistics
#Page No.237 / 7–17
#Prob 7
#7.5.7

#clear console
cat(”\014”)

#clear variables
rm(list=ls(all=TRUE))

# Constructing Quadratic Formula
result <- function(a,b,c){
  if(delta(a,b,c) > 0){ # first case D>0
    x_1 = (-b+sqrt(delta(a,b,c)))/(2*a)
    x_2 = (-b-sqrt(delta(a,b,c)))/(2*a)
    result = c(x_1,x_2)
  }
  else if(delta(a,b,c) == 0){ # second case D=0
```

```
    x = -b/(2*a)
  }
  else {”There are no real roots.”} # third case D<0
}

# Constructing delta
delta<-function(a,b,c){
  b^2-4*a*c
}

a <- result(1,-102,647)
print(a)
print(’Two possible choices for new variable are:’)
print(’y1=−.95x1+.32x2 and’)
print(’y1=.95x1−.32x2’)
print(’variance of y1 is’)
print(a[1])
print(’total variance is’)
print(a[1]+a[2])
print(’the new variable y1 explains about’)
print(a[1]/(a[1]+a[2]))
print(’% of the variance in the data’)
```