R Textbook Companion for Linear Algebra by Jim Hefferon[1]

Created by
Ajay Ragh
B.Tech.
Information Technology
APJ Abdul Kalam Technological University
Cross-Checked by
R TBC Team

May 10, 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

Author: Jim Hefferon

Publisher: Orthogonal Publishing L3C

Edition: 3

Year: 2017

ISBN: 978-1944325039

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 Systems

R code Exa 2.1 solution of system of linear equations

```
#Example 2.1,chapter1,section 1.2,page 13
#package used matlib v0.9.1
#install package using command: install.packages("matlib")
#Github reposiory of matlib :https://github.com/friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(2,1,3,0,-1,-1,1,-1,0),ncol=3)
b <- c(3,1,4)
Ab <- cbind(A,b)
Ab <- rowadd(Ab,1,2,-0.5)
Ab <- rowadd(Ab,1,3,-1.5)
Ab <- rowadd(Ab,2,3,-1)
Ab
#from the result we can see that the system doesnot have a unique solution.
```

```
#We can represent the solution set by representing the variables that lead(x,y)by the variable that does not lead(z).
#so solution set {((3/2)-(1/2)z,(1/2)-(3/2)z,z)|z belongs to R}
```

R code Exa 1.3 Solving Linear Systems

```
#Example 1.3
#Section I. Solving Linear Systems,page 3
A <- matrix(c(3,-1,2,1),ncol = 2)
b <- c(7,6)
x <- solve(A) %*% b
x
```

R code Exa 2.3 solution of system of linear equations

```
#Example 2.3,chapter1,section 1.2,page 13
#package used matlib v0.9.1
#install package using command: install.packages("matlib")
#Github reposiory of matlib :https://github.com/friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,0,3,0,1,1,0,-1,1,-1,6,1,-1,1,-6,-1),ncol = 4)
b <- c(1,-1,6,1)
Ab <- cbind(A,b)
```

```
Ab <- rowadd(Ab,1,3,-3)
Ab <- rowadd(Ab,2,3,3)
Ab <- rowadd(Ab,2,4,1)
Ab
#from the reduced augmented matrix Ab,we can see
   that only x and y are leading,while z and w are
   free.
#We represent the solution set by expressing the
   leading variables in terms of the free variables
#The solution set: {(2 - 2z + 2w,-1 + z - w,z,w)|z,w
    belongs to R}
```

R code Exa 3.3 system of homogeneous equations

```
#Example 3.3,section 1.3,chapter 1,page 24
#package used matlib v0.9.1
#install package using command: install.packages("
   matlib")
#Github repository of matlib :https://github.com/
   friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(3,2,4,-1),ncol = 2)
b <- c(3,1)
c <- c(0,0)
Ab <- cbind(A,b) # linear system
Ac <- cbind(A,c) #system of homogeneous equations
#reduction of original system
Ab <- rowadd(Ab,1,2,-2/3)
#reduction of homogeneous system
Ac <- rowadd(Ac,1,2,-2/3)
```

```
#comparing both
Ab
Ac
#Obviously the two reductions go in the same way.
#We can study how to reduce a linear systems by
   instead studying how to reduce the associated
   homogeneous system.
```

R code Exa 1.4 Solving Linear Systems

```
#Example 1.4,Section I. Solving Linear Systems
#page 3
#package used: matlib
#installation run command : install.packages("matlib
   ")
#package repo : https://github.com/friendly/matlib

#installing and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(0,1,1/3,0,5,2,3,-2,0),ncol = 3,nrow =
   3)
b <- c(9,2,3)
Ab <- cbind(A,b)
Ab <- rowswap(Ab,1,3)
Ab <- rowmult(Ab,1,3)
Ab <- rowmult(Ab,1,-1)
Ab <- rowadd(Ab,1,2,1)
Ab
#from Ab (X3=3)(since drom row3,3*X3=9)
#from Ab row2 -1*X2-2*x3=-7,so X2=1
#from Ab row1 X1+6*x2=9,so X1 = 3
```

R code Exa 2.5 finding the type of solution set of a system of linear equation

```
#Example 2.5,section 1.2,page 15
#package used matlib v0.9.1
#install package using command: install.packages("matlib")
#Github reposiory of matlib :https://github.com/friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,2,3,2,0,2,0,1,1,0,0,-1),ncol = 4)
b <- c(1,2,4)
Ab <- cbind(A,b) #augmented matrix
Ab <- rowadd(Ab,1,2,-2)
Ab <- rowadd(Ab,1,3,-3)
Ab <- rowadd(Ab,2,3,-1)
Ab
#The leading variables are x, y, and w. The variable z is free.
#although there are infinitely many solutions, the value of w doesn't vary but is constant w = −1.
```

R code Exa 3.5 gauss method to reduce system of linear equations

```
#Example 3.5,chapter 1,section 1.3,page 25
#package used matlib v0.9.1
```

```
#install package using command: install.packages("
    matlib")
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(7,8,0,0,0,1,1,3,-7,-5,-3,-6,0,-2,0,-1)
    ,ncol=4)
b <- c(0,0,0,0)
Ab <- cbind(A,b)
Ab <- rowadd(Ab,1,2,-8/7)
Ab <- rowadd(Ab,2,3,-1)
Ab <- rowadd(Ab,2,4,-3)
Ab <- rowadd(Ab,3,4,-5/2)
Ab
```

R code Exa 1.7 solution of system of linear equations

```
#Example 1.7,page nO:5
#package used: matlib
#installation run command : install.packages("matlib
    ")
#package repo : https://github.com/friendly/matlib

#installing and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,2,1,1,-1,-2,0,3,-1),ncol = 3)
b <- c(0,3,3)
```

```
Ab <- cbind(A,b)
Ab <- rowadd(Ab,1,2,-2)
Ab <- rowadd(Ab,1,3,-1)
Ab <- rowadd(Ab,2,3,-1)
Ab
#from row3 : −4z=0,so z=0
#from row2 : −3y+3z=3,so y=−1
#from row1 : x+y=0,so x=1
```

R code Exa 2.7 solution of system of linear equations

```
#Example 2.7,chapter one,section 1.2,page 16
#package used: matlib
#installation run command : install.packages("matlib
    ")
#package repo : https://github.com/friendly/matlib

#installing and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,0,1,2,1,0,0,-1,2),ncol = 3)
b <- c(4,0,4)
#creating augmented matrix
Ab <- cbind(A,b)
Ab
#applying reduction techniques
Ab <- rowadd(Ab,1,3,-1)
Ab <- rowadd(Ab,2,3,2)
Ab
#second row: y−z=0
#first row: x+2*y=4
#so solution set is : {(4−2*z = 0)}
```

R code Exa 1.8 solution of system of linear equations

```
#Example 1.8,page 5
#package used: matlib
#installation run command : install.packages("matlib
    ")
#package repo : https://github.com/friendly/matlib

#installing and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(40,-50,15,25),ncol = 2)
b <- c(100,50)
Ab <- cbind(A,b)
Ab <- rowadd(Ab,1,2,1.25)
Ab
#from row2 : 43.75c = 175,so c= 4
#from row1 : 40h+15c=100,so h=1
```

R code Exa 1.9 gauss method to reduce system of linear equations

```
#Example 1.9,chapter1 linear systems page 6
#Example showing gauss method to reduce given system
     of linear equations
#package used matlib v0.9.1
#install package using command: install.packages("
    matlib")
#Github reposiory of matlib :https://github.com/
    friendly/matlib

```

```
#installing and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,2,3,1,4,6,1,-3,-5),ncol = 3)
b <-c(9,1,0)
Ab <- cbind(A,b)
Ab <-rowadd(Ab,1,2,-2)
Ab <-rowadd(Ab,1,3,-3)
Ab <-rowadd(Ab,2,3,-(3/2))
Ab
#from Ab row3: z=3
#from Ab row2: y=-1
#from Ab row1: x=7
```

R code Exa 1.11 gauss method to reduce system of linear equations

```
#Example 1.11,page 6
#package used: matlib
#installation run command : install.packages("matlib")
#package repo : https://github.com/friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,2,0,0,-1,-2,1,0,0,1,0,2,0,2,1,1),ncol=4)
b <- c(0,4,0,5)
Ab <- cbind(A,b)
Ab <- rowadd(Ab,1,2,-1)
Ab <- rowswap(Ab,2,3)
```

```
Ab <- rowadd(Ab,3,4,-2)
Ab
#Back−substitution gives w = 1, z = 2 , y = −1, and
    x = −1.
```

R code Exa 1.12 gaussian elimination technique

```
#Example 1.12 Section I. Solving Linear Systems
    page7
#package used matlib v0.9.1
#install package using command: install.packages("
    matlib")
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installing and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,2,2,3,1,2),ncol = 2)
b <- c(1,-3,-2)
#for this problem we cannot use normal method
    because the number of equations is more than
    number of variables
#so we use gaussian elimination technique.
gaussianElimination(A, b, tol = sqrt(.Machine$double
    .eps), verbose = FALSE,
                    latex = FALSE, fractions = FALSE
                        )
# result shows that one of the equations is
    redundant, here x=−2,y=1
```

R code Exa 2.12 vector addition

```
#Example 2.12,section 1.2,chapter 1,page 18.
#vector addition
a <- c(2,3,1)
b <- c(3,-1,4)
a+b
c <- c(1,4,-1,-3)
7*c
```

R code Exa 1.13 solution of system of linear equations

```
#Example 1.13,page 7
#package used matlib v0.9.1
#install package using command: install.packages("
    matlib")
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,2,2,3,1,2),ncol = 2)
b <- c(1,-3,0)
Ab <- cbind(A,b)
Ab <- rowadd(Ab,1,2,-2)
Ab <- rowadd(Ab,1,3,-2)
Ab <- rowadd(Ab,2,3,-(4/5))
Ab
#the echelon form shows that the system is
    inconsistent,hence the solution set is empty
```

Chapter 2

Vector Spaces

R code Exa 3.2 Linear Independence

```
#Example 3.2,Section III. Basis and Dimension,page
   127
#package used matlib v0.9.1
#install package using command: install.packages("
   matlib")
#Github reposiory of matlib :https://github.com/
   friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(2,4,3,6),ncol = 2)
#Rowspace(A) is this subspace of the space of two-
   component row vectors
#{c1.(2 3) + c2.(4 6) | c1,c2 belongs to R}
#simplifying A
A <- rowadd(A,1,2,-2)
A
#From the simplified matrix,the second row vector is
```

```
    linearly dependent on the first and so we can
    simplify the above description to
#{c.(2 3)|c belongs to R}
```

R code Exa 3.5 Basis for the column space of the given matrix

```
#Example 3.5,Section III. Basis and Dimension,page
    128
#package used matlib v0.9.1
#install package using command: install.packages("
    matlib")
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,1,2,3,4,0,1,1,5),ncol = 3)
#From any matrix, we can produce a basis for the row
     space by
#performing Gauss's Method and taking the nonzero
    rows of the resulting echelon form matrix
#simplifying to echelon form
A <- rowadd(A,1,2,-1)
A <- rowadd(A,1,3,-2)
A <- rowadd(A,2,3,6)
A
#on simplification:produces the basis h(1 3 1); (0 1
     0); (0 0 3)i for the row space. This is a basis
#for the row space of both the starting and ending
    matrices, since the two row spaces are equal.
```

R code Exa 1.6 Linear Independence

```
#Chapter 2.
#Section II. Linear Independence
#Example 1.6,page 103
#package used matlib v0.9.1
#install package using command: install.packages("matlib")
#Github reposiory of matlib :https://github.com/friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
#c1(40 15) + c2(-50 25) = (0 0),check if {(40 15),(-50 25)} is linearly independent
A <- matrix(c(40,15,-50,25),ncol = 2)
b <- c(0,0)
Ab <- cbind(A,b)
Ab <- rowadd(Ab,1,2,-15/40)
Ab
#from Ab,Both c1 and c2 are zero. So the only linear relationship between the two given row vectors is the trivial relationship.
```

R code Exa 3.7 Basis for the column space of the given matrix

```
#Example 3.7,Section III. Basis and Dimension,page 129
#package used matlib v0.9.1
```

```
#install package using command: install.packages("
    matlib")
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,2,0,4,3,3,1,0,7,8,2,4),ncol = 3)
#to get a basis for the column space, temporarily
    turn the columns into rows and reduce.
A <- t(A)
A <- rowadd(A,1,2,-3)
A <- rowadd(A,1,3,-7)
A <- rowadd(A,2,3,-2)
#Now turn the rows back to columns
A <- t(A)
A
#The result is a basis for the column space of the
    given matrix.
```

R code Exa 1.9 Linear Independence

```
#Chapter 2.
#Section II. Linear Independence
#Example 1.9,page 104
v1 <- c(3,4,5)
v2 <- c(2,9,2)
v3 <- c(4,18,4)
(0*v1)+(2*v2)-1*v3
#the set S = {v1,v2,v3} is linearly dependent
    because this is a relationship where not all of
    the scalars are zero
```

R code Exa 3.9 Finding a basis for the given span

```
#Example 3.5,Section III. Basis and Dimension,page
    130
#package used matlib v0.9.1
#install package using command: install.packages("
    matlib")
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
#To get a basis for the span of {x^2 + x^4,2x^2 + 3x
    ^4,-x^2 - 3x^4 } in the space row4
A <- matrix(c(0,0,0,0,0,0,1,2,-1,0,0,0,1,3,-3),ncol
    = 5,nrow = 3)
#applying gauss method
A <- rowadd(A,1,2,-2)
A <- rowadd(A,1,3,1)
A <- rowadd(A,2,3,2)
A
#we get the basis (x^2 + x^4,x^4)
```

R code Exa 3.10 finding basis from reduced echelon form

```
#Example 3.10,Section III. Basis and Dimension,page
    131
#package used pracma
```

```
#install package using command: install.packages("
    pracma")

#installation and loading library
#install.packages("pracma")
library("pracma")

#program
A <- matrix(c(1,2,1,3,6,3,1,3,1,6,16,6),ncol = 4,
    nrow = 3)
#finding row reduced echelon form("using gauss-
    jordan reduction")
rref(A)
#Thus, for a reduced echelon form matrix we can find
     bases for the row and column spaces in
    essentially the same way, by taking the parts of
    the matrix, the rows or columns, containing the
    leading entries.
```

R code Exa 1.16 finding the coordinates of that vector with respect to the basis

```
#Example 1.16,Section III. Basis and Dimension,page
    118
#package used matlib v0.9.1
#install package using command: install.packages("
    matlib")
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
```

```
v <- c(3,2)
A <- matrix(c(1,1,0,2),ncol = 2)
Av <- cbind(A,v)
Av <- echelon(A,v,reduced = TRUE)
Av
#from Av, c1 = 3 and c2 = −1=2.
```

Chapter 3

Maps Between Spaces

R code Exa 1.1 correspondence between vectors

```
#Example 1.1,Section I. page 166
a <- c(1,2)
a <-t(a) #a becomes row vector
b <-t(a) #b becomes column vector
c <-c(3,4)
c <-t(c)
d <-t(c)
a+c
b+d
#these corresponding vectors add to corresponding
   totals
5*a
5*b
#correspondence respecting scalar multiplication
```

R code Exa 4.1.1 Matrix Operations

```
#Example1.1,section IV:Matrix Operations,chapter3,
   page 224
```

```
#Let f : V -> W be a linear function represented with respect to some bases by this matrix.
f <- matrix(c(1,1,0,1),ncol = 2)
#find the map that is the scalar multiple 5f: V -> W.
5*f
#Changing from the map f to the map 5f has the effect on the representation of the output vector of multiplying each entry by 5.
#Therefore, going from the matrix representing f to the one representing 5f means multiplying all the matrix entries by 5.
```

R code Exa 4.1.2 Matrix Operations

```
#Example1.2,section IV:Matrix Operations,chapter3,page 225
f <- matrix(c(1,2,3,0),ncol = 2)
g <- matrix(c(-2,2,-1,4),ncol = 2)
#two linear maps with the same domain and codomain f; g: R^2 -> R^2 are represented with respect to bases B and D by these matrices.
#find effect of f + g on the map
f+g
#if f does v -> u and g does v -> w then f + g is the function whose action is v -> u + w.
```

R code Exa 3.1.11 Computing linear maps

```
#Example1.11,section III:Computing linear maps,chapter3,page 211
a <- matrix(c(1,2,0,0,-1,3),ncol=3)
b <- c(2,-1,1)
```

```r
a %*% b
#the above result can also be obtained by the method
x <- matrix(c(1,2),ncol = 1)
y <- matrix(c(0,0),ncol = 1)
z <- matrix(c(-1,3),ncol = 1)
#splitting the matrix a into component columns and
    now multiplying
2*x-1*y+1*z
#we see that both methods are equal.
```

R code Exa 4.2.4 Matrix Operations

```r
#Example2.4,section IV:Matrix Operations,chapter3,
    page 229
A <- matrix(c(2,4,8,0,6,2),ncol = 2)
B <- matrix(c(1,5,3,7),ncol = 2)
A %*% B
```

R code Exa 4.2.5 Matrix Operations

```r
#Example2.5,section IV:Matrix Operations,chapter3,
    page 229
A <- matrix(c(1,3,2,4),ncol = 2)
B <- matrix(c(-1,2,0,-2),ncol=2)
A %*% B
```

R code Exa 4.2.6 Matrix Operations

```r
#Example2.6,section IV:Matrix Operations,chapter3,
    page 230
```

```
A <- matrix(c(1,0,1,1,1,0),ncol=2)
B <- matrix(c(4,5,6,7,8,9,2,3),ncol = 4)
A %*% B
```

R code Exa 4.3.15 Mechanics of Matrix Multiplication

```
#Example3.15,section IV.3:Mechanics of Matrix
    Multiplication,chapter3,page 241
#A permutation matrix is square and is all 0's
    except for a single 1 in each row and column.
A <- matrix(c(0,1,0,0,0,1,1,0,0),ncol = 3) #
    permutation matrix
B <- matrix(c(1,4,7,2,5,8,3,6,9),ncol = 3)
#From the left these matrices permute rows.
A %*% B
#From the right they permute columns.
B %*% A
```

R code Exa 4.4.9 Inverses

```
#Example4.9,section IV.4:Inverses,chapter3,page 251
#package used matlib v0.9.1
#install package using command: install.packages("
    matlib")
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
# Augmented matrix
```

```
A <- matrix(c(0,1,1,3,0,-1,-1,1,0),ncol = 3)
B <- matrix(c(1,0,0,0,1,0,0,0,1),ncol = 3)
AB <- cbind(A,B)
echelon(A,B)
```

R code Exa 2.5 Computing linear maps

```
#Example2.5, section III:Computing linear maps,
   chapter3, page 218
#package used pracma
#install package using command: install.packages("
   pracma")

#installation and loading library
#install.packages("pracma")
library("pracma")

#program
A <- matrix(c(1,1,0,0,2,2,0,0,2,1,3,2),ncol = 3)
Rank(A)
#Any map represented by above matrix must have three
   −dimensional domain and a four−dimensional
   codomain.
#Since the rank of this matrix is found to be 2 by
   above code;
#Any map represented by this matrix has a two−
   dimensional range space.
```

R code Exa 1.9 automorphism

```
#Example 1.9, Section I. page 170
#space P5 of polynomials of degree 5 or less and the
    map f that sends a polynomial p(x) to p(x − 1).
```

```
#under this map x^2 ->(x-1)^2 = x^2-2x+1 and x^3+2x
    -> (x-1)^3+2(x-1) = x^3-3x^2+5x-3.
curve(x^2,from = -1000,to=1000)
curve((x-1)^2,from = -1000,to=1000)
curve(x^3,from = -1000,to=1000)
curve((x-1)^3,from = -1000,to=1000)
#from these plots we can say that this map is an
    automorphism of this space.
```

R code Exa 2.10 Computing linear maps

```
#Example2.10,section III:Computing linear maps,
    chapter3,page 219
#package used pracma
#install package using command: install.packages("
    pracma")

#installation and loading library
#install.packages("pracma")
library("pracma")

#program
A <- matrix(c(1,0,2,3),ncol = 2)
Rank(A)
#Any map from R^2 to P1 represented with respect to
    any pair of bases by the above matrix;
#is nonsingular because this matrix has rank two.
```

R code Exa 1.11 value of h on the basis vectors

```
#Example 1.11,Section II. page 186
#given map: h(1,0)=(-1,1) and h(0,1)=(-4,4)
#h(3,-2)=h(3*(1,0)-2*(0,1))=3*h(1,0)-2*h(0,1)
```

```r
a <- c(-1,1) # h(1,0)
b <- c(-4,4) # h(0,1)
3*a-2*b
#the value of h on this argument is a direct consequence of the value of h on the basis vectors.
```

R code Exa 2.11 Computing linear maps

```r
#Example2.11,section III:Computing linear maps, chapter3,page 220
A <- matrix(c(1,3,2,6),ncol=2)
det(A)
#map g: V -> W represented by above matrix is singular because this matrix is singular.
```

Chapter 4

Determinants

R code Exa 3.1 Determinants

```
#Example3.1,chapter4,section1,page 329
X <- matrix(c(4,-2,2,6),ncol = 2)
#scalars come out of each row separately,not from
  the entire matrix at once.
#so X will become
A <- matrix(c(2,-1,1,3),ncol = 2)
#with scalars 2 & 2 coming out of rows 1 and 2
  respectively
c <- det(X)
d <- 4*det(A)
all.equal(c,d)
```

R code Exa 1.4 Laplace Formula to find determinant

```
#Example 1.4,chapter 4,Section III. Laplace's
  Formula,page 354
#package used matlib v0.9.1 and pracma
#Github reposiory of matlib :https://github.com/
  friendly/matlib
```

```
#installation and loading library
#install.packages("matlib")
library("matlib")

#program
T <- matrix(c(1,4,7,2,5,8,3,6,9),ncol = 3)
T12 <- cofactor(T,1,2)
T22 <- cofactor(T,2,2)
T12
T22
```

R code Exa 3.4 Determinants

```
#Example3.4,chapter 4,page 330
A <- matrix(c(2,4,1,3),ncol = 2)
#using multinlinearity property to break up the
    matrix
a <- matrix(c(2,4,0,0),ncol = 2)
b <- matrix(c(2,0,0,3),ncol = 2)
c <- matrix(c(0,4,1,0),ncol = 2)
d <- matrix(c(0,0,1,3),ncol = 2)
#verifying the property
x <- det(A)
y <- det(a)+det(b)+det(c)+det(d)
all.equal(x,y)
```

R code Exa 2.5 Determinants

```
#Example 2.5,chapter4,Section I. Definition,page 325
#package used matlib v0.9.1
#install package using command: install.packages("
    matlib")
```

```
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(2,4,0,2,4,-3,6,3,5),ncol=3)
b <- det(A)
#determinant by normal ,ethod
A <- rowadd(A,1,2,-2)
A <- rowswap(A,2,3)
A
#reducing with gaussian reduction now multiplying
    the diagonal terms,keeping in mind the sign
    change due to row swap to find determinant.
c <- -1*A[1,1]*A[2,2]*A[3,3]
all.equal(b,c)
#so the determinant by both techniques are the same
```

R code Exa 1.6 Laplace Formula to find determinant

```
#Example 1.6,chapter 4,Section III. Laplace's
    Formula,page 355\
#package used matlib v0.9.1 and pracma
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
t <- matrix(c(1,4,7,2,5,8,3,6,9),ncol = 3)
```

```
x <- det(t)
#computing the determinant by expanding along the first row,
y <- t[1,1]*cofactor(t,1,1)+t[1,2]*cofactor(t,1,2)+t[1,3]*cofactor(t,1,3)
#computing the determinant by expanding along the second row,
z <- t[2,1]*cofactor(t,2,1)+t[2,2]*cofactor(t,2,2)+t[2,3]*cofactor(t,2,3)
all.equal(x,y)
all.equal(x,z)
```

R code Exa 2.6 Determinants

```
#Example 2.6,chapter4,Section I. Definition,page 326
#package used matlib v0.9.1 and pracma
#Github reposiory of matlib :https://github.com/friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
A <- matrix(c(1,0,0,0,0,1,0,1,1,1,0,0,3,4,5,1),ncol = 4)
A <- rowadd(A,2,4,-1)
A <- rowswap(A,3,4)
A
#multiplying diagonal terms and multiplying it with -1 to compensate for rowswap
-1*A[1,1]*A[2,2]*A[3,3]*A[4,4]
```

R code Exa 1.7 Laplace Formula to find determinant

```
#Example 1.7,chapter 4,Section III. Laplace's
   Formula,page 355
#package used matlib v0.9.1 and pracma
#Github reposiory of matlib :https://github.com/
   friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
t <- matrix(c(1,2,3,5,1,-1,0,1,0),ncol = 3)
#computing the determinant by expanding along the
   third column.
y <- t[1,3]*cofactor(t,1,3)+t[2,3]*cofactor(t,2,3)+t
   [3,3]*cofactor(t,3,3)
y
```

R code Exa 1.10 Laplace Formula to find inverse

```
#Example 1.10,chapter 4,Section III. Laplace's
   Formula,page 356
#package used matlib v0.9.1 and pracma
#Github reposiory of matlib :https://github.com/
   friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")

#program
t <- matrix(c(1,2,1,0,1,0,4,-1,1),ncol = 3)
#finding inverse: if T has an inverse, if |T| != 0,
```

```
    then T^-1 = (1/|T|)   adj(T)
a <- adjoint(t)
b <- 1/det(t)
i <- b*a
i
```

Chapter 5

Similarity

R code Exa 2.2 Complex Vector Spaces

```
#Example 2.2,chapter 5,Section I. Complex Vector
    Spaces,page 387
a <- complex(real = 1,imaginary = 1)
b <- complex(real = 0,imaginary = 1)
c <- complex(real = 2,imaginary = -0)
d <- complex(real = -2,imaginary = 3)
e <- complex(real = 1,imaginary = 0)
f <- complex(real = 0,imaginary = 3)
g <- complex(real = 1,imaginary = -0)
h <- complex(real = 0,imaginary = -1)
A <- matrix(c(a,b,c,d),ncol = 2)
B <- matrix(c(e,f,g,h),ncol = 2)
A %*% B
```

R code Exa 2.2.2 Checking if matrix is diagonalizable

```
#Example 2.2,chapter 5,section II.2
    Diagonalizability,page 393
```

```
#matrix: T is diagonalizable if there is a
    nonsingular P such that PTP−1 is diagonal.
T <- matrix(c(4,1,-2,1),ncol = 2)
P <- matrix(c(-1,1,2,-1),ncol = 2)
#checking whether diagonizable
P %*% T %*% Inverse(P) # diagonal matrix
# so matrix P is diagonizable
```

R code Exa 1.3 Similar Matrix

```
#Example 1.3,chapter 5,Section II. Similarity,page
    390
#package used matlib v0.9.1
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")
P <- matrix(c(2,1,1,1),ncol = 2)
T <- matrix(c(2,1,-3,-1),ncol = 2)
#finding similar matrix of T
T1 <- P %*% T %*% Inverse(P)
T1
```

R code Exa 2.4 finding null spaces

```
#Example 2.4,chapter 5,section IV.2,page 432
#package used pracma v1.9.9
#installing and loading library
#install.packages("pracma")
library("pracma")
T <- matrix(c(2,1,-1,4),ncol=2)
```

```
a <- eigen(T)
a$values
#so T has only the single eigenvalue 3.
I <- matrix(c(1,0,0,1),ncol = 2)
T-(3*I)
# so for this, the only eigenvalue is 0 and T -3I is
    nilpotent.
#to ease this computation we find nulspaces
x <- nullspace(T)
x
```

R code Exa 3.6 Eigenvalues and Eigenvectors

```
#Example3.6, chapter 5, section II.3 Eigenvalues and
    Eigenvectors, page 398
T <- matrix(c(2,0,0,0),ncol = 2)
a <-eigen(T)
a$values
a$vectors
```

R code Exa 2.10 Nilpotent matrix

```
#Example 2.10, chapter 5, scetion III.2, page 414
#package used matlib v0.9.1
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")
N <- matrix(c(0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0),ncol
    =4)
```

```r
P <- matrix(c(1,0,1,0,0,2,1,0,1,1,1,0,0,0,0,1),ncol
    = 4)
A <- P %*% N %*% Inverse(P)
A
#The new matrix ,A is nilpotent; its fourth power is
    the zero matrix.
x <- P %*% N^4 %*% Inverse(P)
#since (PNP^-1)^4 = P * N^4 *P^-1
y <- det(x)
all.equal(y,0)
```

R code Exa 2.17 Nilpotence index

```r
#Example 2.17,chapter 5,scetion III.2,page 419
#package used matlib v0.9.1
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")
M <- matrix(c(1,1,-1,-1),ncol = 2)
#finding nilpotent index
A <- matrix(c(0,0,0,0),ncol = 2)
count <- 1
Y <- M
repeat{
  Y <- Y %*% M
  if (all.equal(Y,A)){
    print(count+1)
    break()
  }
 count= count+1
}
```

R code Exa 2.18 nilpotence index

```
#Example 2.18,chapter 5,scetion III.2,page 420
#package used matlib v0.9.1
#Github reposiory of matlib :https://github.com/
    friendly/matlib

#installation and loading library
#install.packages("matlib")
library("matlib")
M <- matrix(c
    (0,1,-1,0,1,0,0,1,1,0,0,0,1,0,-1,0,0,-1,0,1,0,0,1,0,-1)
    ,ncol = 5)
#finding nilpotent index
A <- matrix(c(0),ncol = 5,nrow = 5)
count <- 1
Y <- M
repeat{
  Y <- Y %*% M
  if (all.equal(Y,A)== TRUE){
    print(count+1)
    break()
  }
  count=count+1
}
```

R code Exa 3.19 Eigenvalues and Eigenvectors

```
#Example3.19,chapter 5,section II.3 Eigenvalues and
    Eigenvectors,page 404
T <- matrix(c(2,0,-4,-2,1,8,2,1,3),ncol = 3)
a <- eigen(T)
```

```
a$values
```