

R Textbook Companion for
Introduction to Probability
by Dimitri P. Bertsekas and John N. Tsitsiklis¹

Created by
Midhun C Kachappilly
B.Tech.
Electrical Engineering
Government Engineering College Barton Hill
Cross-Checked by
R TBC Team

May 26, 2020

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

Book Description

Title: Introduction to Probability

Author: Dimitri P. Bertsekas and John N. Tsitsiklis

Publisher: Athena Scientific

Edition: 2

Year: 2008

ISBN: 978-1-886529-23-6

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

Contents

List of R Codes	4
1 sample space and probability	5
2 Discrete Random Variable	17
3 General Random Variable	30
4 Further Topics on Random Variables and Expectations	38
5 Stochastic Processes	47
6 Markov Chains	51
7 Limit Theorems	56

List of R Codes

Exa 1.2	Discrete Models	5
Exa 1.3	Probabilistic Models	6
Exa 1.6	Conditional probability	7
Exa 1.8	Conditional probability	7
Exa 1.9	Conditional probability	8
Exa 1.10	Conditional probability	10
Exa 1.11	Total Probabilty Theorem and Bayes Rule	11
Exa 1.12	Total Probabilty Theorem and Bayes Rule	12
Exa 1.18	Independence	13
Exa 1.28	Counting	13
Exa 1.29	Counting	14
Exa 1.30	Counting	15
Exa 2.1	functions and random variables	17
Exa 2.2	Expectation Mean and Variance	18
Exa 2.3	Expectation Mean and Variance	19
Exa 2.4	Expectation Mean and Variance	21
Exa 2.5	Expectation Mean and Variance	22
Exa 2.7	Expectation Mean and Variance	23
Exa 2.8	Expectation Mean and Variance	23
Exa 2.9	Joint PMFs of Multiple Random Variable	24
Exa 2.11	Conditioning	24
Exa 2.13	conditioning	25
Exa 2.14	Conditioning	26
Exa 2.16	Independence	27
Exa 2.18	Independence	28
Exa 3.2	Continuous Random Variables And PDFs	30
Exa 3.3	Continuous Random Variables And PDFs	31
Exa 3.4	Continuous Random Variables And PDFs	31

Exa 3.5	Continuous Random Variables And PDFs	32
Exa 3.7	cumulative distribution Function	33
Exa 3.8	Normal Random Variable	33
Exa 3.9	Normal Random Variable	34
Exa 3.11	conditioning on an event	35
Exa 3.12	conditioning on an event	36
Exa 3.22	Derived Distributions	36
Exa 4.1	Transforms	38
Exa 4.4	Transforms	39
Exa 4.5	Transforms	39
Exa 4.6	Transforms	40
Exa 4.7	Transforms	41
Exa 4.9	Transforms	41
Exa 4.13	Transforms	42
Exa 4.16	conditional expectation as random variable	44
Exa 4.20	conditional expectation as random variable	44
Exa 4.21	Sum of a Random Number of Independent Random Variable	45
Exa 5.5	Bernolli process	47
Exa 5.6	The Poisson Process	48
Exa 5.7	The Poisson Process	48
Exa 5.9	The Poisson Process	49
Exa 5.12	The Poisson Process	49
Exa 5.17	The Poisson Process	50
Exa 6.1	the discrete time markov chains	51
Exa 6.2	Discrete Time Markov Chains	51
Exa 6.4	Steady State Behavior	52
Exa 6.11	Absorption Probabilities and Expected Time to Absorp- tion	53
Exa 6.13	Absorption Probabilities and Expected Time to Absorp- tion	54
Exa 6.14	more general markov chains	54
Exa 7.1	Some Useful Inequalities	56
Exa 7.4	convergence in probability	57
Exa 7.8	The central Limit Theorem	57
Exa 7.9	The central Limit Theorem	58
Exa 7.10	The central Limit Theorem	59
Exa 7.11	The central Limit Theorem	60

Chapter 1

sample space and probability

R code Exa 1.2 Discrete Models

```
1 # EX1_2
2 #page 10
3 number_flips <- 100
4 # created coin object with head and tail
5 coin <- c("heads", "tails")
6 #simulating the flip of the object coin
7 flips <- sample(coin, size=number_flips, replace=
  TRUE)
8 #counting the number of heads and tails in the flips
9 freq <- table(flips)
10 #typing the frequency of heads and tails
11 freq
12 #probability of getting head if we specify that head
  and tail is equally likely
13 dbinom(1, size=1, prob=0.5)
14 #total probability of head and tail
15 dbinom(1, size=1, prob=0.5)+
16 dbinom(1, size=1, prob=0.5)
```

R code Exa 1.3 Probabilistic Models

```
1 #EX_1_3
2 #page 11
3 Dice<- seq(1:4)
4 d<-0
5 c<-0
6 a<- numeric(2)#creating an array
7 Sample_Space<-expand.grid(Dice,Dice)#creating the
  sample space
8 Sum_Matrix <- matrix(nrow=4,ncol=4) #creating a
  sample matrix
9 #storing the sum of sample space of rolling 2 dice
10 for (i in 1:4)
11 {
12     for (j in 1:4)
13     {
14         a[1]<-i
15         a[2]<-j
16         Sum_Matrix[i,j]=sum(a)#stores the sum of the
          sample in the matrix b
17         if((Sum_Matrix[i,j]%%2)== 0)#to check
          whether the sum is even
18         {
19             print(Sum_Matrix[i,j])
20             d <-d+1#count the even sums
21         }else
22         {
23             c<-c+1#count odd sums
24         }
25     }
26 }
27 total_sample_space<- nrow(Sample_Space)#finding the
  number of sample space
28 n<-total_sample_space#printing the sample space
  count
29 Even_Sum<- d/n
30 Even_Sum #printing the probability of getting even
```



```

      numbers
31 Odd_Sum<-c/n
32 Odd_Sum #printing the probability of getting odd
      numbers

```

R code Exa 1.6 Conditional probability

```

1 #EX1_6.R
2 #page 19
3 coins<- c("H","T","H","T","H","T")
4 for(i in 1:8){
5 flips <- sample(coins, size=3, replace=FALSE)
6 print(flips)
7 }
8 A<-dbinom(2, size=3, prob=0.5)
9 dbinom(3, size=3, prob=0.5)
10 #probability of getting more heads than tails
11 p<-dbinom(3, size=3, prob=0.5)+
12 dbinom(2, size=3, prob=0.5)
13 print(p)
14 #probability of first toss is a head
15 q<- 1-p
16 #probability of (A and B)
17 s<- 3/8
18 #conditional probability p(A|B)
19 r<- s/q
20 print(r)

```

R code Exa 1.8 Conditional probability

```

1 #EX_1_8
2 #page 19
3 p1<- 2/3

```

```

4 #probability of team n succes
5 #p(SS)+P(FS)=1/2
6 p2<- 1/2
7 #probability atleast one got selected =p(SS)+p(SF)+p(
  FS)
8 p3<-3/4
9 #Difference of p3 and p1= p(FS)
10 FS<-sum(p3,-p1)
11 FS
12 SF<-sum(p3,-p2)
13 SF
14 p4<-sum(p1,p2)
15 SS<-sum(p4,-p3)
16 SS
17 #the probability that it was designed by team N
18 #conditional probabiltiy
19 #P({FS}|{SF,FS})
20 PN<- FS/sum(FS,SF)
21 PN#prints the probability that it is designed by N

```

R code Exa 1.9 Conditional probability

```

1 #EX_1_9
2 #page 21
3 Rate<-100000# specifying the number of trials
4 Aircraft<- sample(c("no","yes"),size=Rate,replace=
  TRUE,prob=c(0.95,0.05)) #created the sample space
  of presence of aircraft
5 Radar<- rep(NA,Rate)#creating the radar sample array
6 #creating the sample space in which radar detects
  the aircraft eventhough there is not an aircraft
7 Radar[Aircraft=="no"]<-sample(c("detects","notdetect
  "),size=sum(Aircraft=="no"),replace=TRUE,prob=c
  (0.10,0.90))
8 #Creating the sample space in which radar detects

```

```

    aircraft with the presence of aircraft
9  Radar[Aircraft=="yes"]<-sample(c("detects","
    notdetect"),size=sum(Aircraft=="yes"),replace=
    TRUE,prob=c(0.99,0.01))
10 p1<-mean(Radar[Aircraft=="yes"]=="detects")#
    probability radar detects aircraft given aircraft
    present
11 p2<-mean(Radar[Aircraft=="yes"]=="notdetect")#
    probability radar not detects given aircraft is
    present
12 p3<- mean(Radar[Aircraft=="no"]=="detects")#
    probability radar detects given aircraft is not
    present
13 p4<-mean(Radar[Aircraft=="no"]=="notdetect")#
    probability radar not detects given aircraft is
    not present
14 p5<-mean(Aircraft=="yes")#probability of presence of
    aircraft
15 p6<-mean(Aircraft=="no")#probability aircraft is not
    present
16 #A = {an aircraft is present}, B = {the radar
    registers an aircraft presence}
17 #A! = {an aircraft is not present}, B! = {the radar
    does not register an aircraft presence}.
18 probability <- matrix(c(p1,p2,p3,p4,p5,p6), nrow=6,
    byrow=TRUE, dimnames =list(c("P(B|A)", "P(B!|A)",
    "P(B|A!)", "P(B!|A!)", "P(A)", "P(A!)"), c("
    probability"))))
19 probability
20 Con_prob<- function(a,b)
21 {
22     return(a*b)
23
24 }
25 #P(false alarm) = P(A!B)=P(A!)P(B|A!)
26 p7<- Con_prob(p6,p3)#P(A!B)
27 p7
28 #P(missed detection) = P(AB!)=P(A)P(B!|A)

```

```
29 p8<- Con_prob(p5,p2)# P(AB!)
30 p8
```

R code Exa 1.10 Conditional probability

```
1 #Example 1.10.
2 #page 10
3 #simulation of a deck of cards
4 deck <- c(rep("Diamonds",13), rep("Clubs",13), rep("
    Hearts",13), rep("Spades",13))
5 deck
6 #simulation of picking 3 cards from a deck of 52
    cards
7 picks <- sample(deck, size=3, replace= FALSE)
8 picks
9 #counting number of elements in a pick
10 count<-table(picks)
11 #initializing a list "alpha" to store the
    probability of not getting a heart for each 3
    picks
12 alpha<-numeric(3)
13 #function to calculate probability
14 eventProbability <- function(cardnumber,decknumber)
15 {
16     notHeartprobability <- (cardnumber/decknumber)
17     return (notHeartprobability)
18 }
19 #loop to store probability in list alpha
20 for (i in 1:3){
21     # number of cards in deck before picking a card
22     deckNumber <- 52
23     # number of cards other than heart before picking a
        card
24     notHeartNumber <- 39
25     #after picking cards without replacement
```

```

26 cardsDrawn <- (i-1)
27 deckNumber <- deckNumber- cardsDrawn
28 print(deckNumber)
29 HeartDrawn <- (i-1)
30 notHeartNumber <- notHeartNumber - HeartDrawn
31 print(notHeartNumber)
32 #finding probability for each picking
33 heartprobability <- eventProbability (notHeartNumber
    ,deckNumber)
34 #storing the probability to list
35 alpha [i]<- heartprobability
36 print(alpha)
37 print (heartprobability)
38 }
39
40 print(alpha)
41 #finding the total probabilitly of not getting a heart
    when piking 3 cards from 52 cards
42 probabilityNotHeart <- (alpha[1]*alpha[2]*alpha[3])
43 print(probabilityNotHeart)
44 print(heartprobability)

```

R code Exa 1.11 Total Probabilty Theorem and Bayes Rule

```

1 #EX_1_11
2 #page25
3 #initialize variables
4 decreaseInGroups<- 4
5 studentDecrease <- 1
6 studentSlotInGroups <-12
7 studentSlot <- 15
8 #initialize the array of probability
9 alpha<-numeric(3)
10 #function to calculate probability
11 probability <- function(students ,studSlot)

```

```

12 {prob <- (students/studSlot)
13 return (prob)
14 }
15 for(i in 1:3){
16 probs <- probability(studentSlotInGroups, studentSlot
17 )
17 print(probs)
18 alpha[i] <- probs
19 print(alpha)
20 studentSlotInGroups <- studentSlotInGroups -
21 decreaseInGroups
21 print (studentSlotInGroups)
22 studentSlot <- studentSlot - studentDecrease
23 print(studentSlot)
24 }#loop to calculate probability
25 #calculating probability of having gaduate student
26 in each group
26 totalProbability <- alpha[1]*alpha[2]*alpha[3]
27 print (totalProbability)

```

R code Exa 1.12 Total Probabilty Theorem and Bayes Rule

```

1 #EX_1_12
2 #page26
3 mat<-numeric(3)
4 win <- matrix(c(0.3, 0.4, 0.5),nrow=3,byrow=TRUE,
5   dimnames=list(c("p(B|A1)", "P(B|A2)", "P(B|A3)"), c(
6     "probability"))) #the probability of winning
7   given playing with i'th opponent
8 win
9 playing <- matrix(c(0.5, 0.25, 0.25),nrow=3,byrow=
10   TRUE,dimnames=list(c("p(A1)", "P(A2)", "P(A3)"), c(
11     "probability"))) #the probability of playing with
12   i'th opponent
13 playing

```

```

8 for(i in 1:3){mat[i]<-playing[i]*win[i]} #to do the
  total multiplication theorem
9 mat#printing the product
10 totalprobability<-sum(mat) #total probability of
  winning,P(B)
11 totalprobability#prints the total probabilty of
  winning p(B)

```

R code Exa 1.18 Independence

```

1 #EX_1_18
2 #page34
3 number_flips <- 100
4 # created coin object with head and tail
5 coin <- c("heads","tails")
6 #simulating the flip of the object coin
7 flips <- sample(coin, size=number_flips, replace=
  TRUE)
8 #probability of getting head if we specify that head
  and tail is equally likely
9 dbinom(1, size=1, prob=0.5)
10 #total probability of head and tail
11 dbinom(1, size=1, prob=0.5)#p(H1|D)
12 dbinom(1, size=1, prob=0.5)#p(H2|D)

```

R code Exa 1.28 Counting

```

1 #EX_1_28
2 #page45
3 install.packages("prob")
4 library(prob)
5 permsn(4,2)#permutation of 4 letters out of which 2
  is taken

```

```

6 combn(4,2)#combination of 4 letters out of which 2
  is picked up
7 ncol(combn(4,2))#number of combinations when 4
  letters from which 2 is picked out

```

R code Exa 1.29 Counting

```

1 #EX_1_29
2 #page46
3 TATTOO <- list("T","A","T","T","O","O")# listing the
  letters of Tattoo
4 L<-length(TATTOO)
5 M<-0
6 N<-0
7 R<-0
8 for(i in 1:L)#loop to count the number of same type
  of letters in tattoo
9 {
10   if(as.character(TATTOO[i])=="T")
11   {
12     M<-(M+1)
13   }
14
15   else if(as.character(TATTOO[i])=="A")
16   {
17     N<- (N+1)
18   }
19   else if(as.character(TATTOO[i])=="O")
20   {
21     R<- (R+1)
22   }
23 }
24 #the counters in the loop will count the number of
  same letters
25 repetition<- matrix(c(M,N,R),nrow=3,byrow=T,dimnames

```



```

      = list(c("T","A","O"),c("repetition"))
26 repetition#matrix give the number of repeated
    letters
27 Per<- function(p,q,s,t)
28 {
29   X<-prod(factorial(p))/(factorial(q)*factorial(s)*
    factorial(t))
30   return(X)
31
32 }#function to calculate the permutation
33 p<-Per(L,M,N,R)
34 p

```

R code Exa 1.30 Counting

```

1 #EX_1_30
2 #page47
3 install.packages("prob")
4 library(prob)
5 total_sample_space<- prod(factorial(16))/(factorial
    (4)*factorial(4)*factorial(4)*factorial(4))
6 graduate<-letters[1:4]#creating sample space of 4
    graduate
7 permsn(graduate,4)# sample space combination of 4
    graduate in 4 groups
8 f1<-ncol(permsn(graduate,4))# number of combination
    of 4 graduate in 4 groups
9 #Take the remaining 12 undergraduate students and
    distribute them to the four groups (3 students in
    each).
10 c<-prod(factorial(12))/(factorial(3)*factorial(3)*
    factorial(3)*factorial(3))
11 c
12 f2<-f1*c#total possibility of dividing the 4
    graduate and 12 undergraduate students is

```

```
    randomly divided into four groups of 4.  
13 f2  
14 p<-f2/total_sample_space  
15 p#total probability of the sample space
```

Chapter 2

Discrete Random Variable

R code Exa 2.1 functions and random variables

```
1 #EX_2_1
2 #page 10
3 x<-numeric(9)#creating the sample array
4 y<-numeric(4)
5 X<-c(-4:4)#creating the sample space of x and y
6 y<-c(1:4)
7 #function to caculate sample space of p(x)
8 px<-function(x)
9 {
10   if (-4<=x&& x<=4)
11   {
12     return(1/9)
13   }else{
14     return(0)
15   }
16 }
17 #creating the probability function of y
18 py<-function(y)
19 {
20   if (1<=y&& y<=4)
21   {
```

```

22     return(2/9)
23 }else if(y==0)
24 {
25     return(1/9)
26 }else{
27     return(0)
28 }
29 }
30 #printing the sample space of p(x)
31 for(i in 1:9)
32 {
33     print(px(i-5))
34 }
35 #printing the sample space of p(y)
36 for(i in 1:5)
37 {
38     print(py(i-1))
39 }
40 }

```

R code Exa 2.2 Expectation Mean and Variance

```

1 #EX_2_2
2 #page12
3 install.packages("prob")
4 library(prob)
5 fx<-numeric(3)#initializing the probability mass
  function
6 x<-numeric(3)#initializing the x value
7 M<-numeric(3)#initializing the array to have the
  loop value to calculate mean
8 V<- numeric(3)#initializing the array to have the
  loop value of standard deviation
9 fx <-c(((1/4)^2),(2*1/4*3/4),(3/4)^2)#initializing
  the array of PMF

```

```

10 x<- c(0,1,2)#initializing the x variable
11 coin<- c("H","T")# initializing object coin
12 iidspace(coin,ntrials=2,probs=(c(0.75,0.25)))#sample
    space of tossing 2 coin with probability .75 of
    getting head
13 #for loop to calculate the product of PMF and x
14 for(i in 1:3)
15 {
16   M[i]<-prod(fx[i],x[i])
17 }
18 mean<-sum(M)#calculated the mean
19 mean# print the mean
20 #loop to calculate the variance
21 for(i in 1:3)
22 {
23   V[i]<-(x[i]-mean)^2
24 }
25 variance<-V#calculated the variance
26 variance# print the variance
27 standard_deviation<-sqrt(variance)#standard
    deviation is the square root of variance
28 standard_deviation# print the standard deviation

```

R code Exa 2.3 Expectation Mean and Variance

```

1 #EX_2_3
2 #page 14
3 M<- numeric(9)#created the sample array
4 x<-numeric(9)
5 z<- numeric(9)
6 Z<-numeric(5)
7 V<- numeric(5)
8 PMFZ<-numeric(5)
9 x<-c(-4:4)
10 #function to create the sample space of PX(x)

```

```

11 PMF<-function(x)
12 {
13     if(1<=x&& x<=9)
14     {
15         return(1/9)
16     }else{
17         return(0)
18     }
19 }
20 for(i in 1:9)
21 {
22     M[i]<-x[i]*PMF(i)
23 }
24 Ex<-sum(M)
25 Ex#the expected valor of Px(x)
26 #loop to calculate the Z
27 for(i in 1:9)
28 {
29     z[i]<-(x[i]-mean)^2
30 }
31 Z<-z[5:9]
32 Z#calculating the sample space of Z
33 PMFz<-function(z)
34 {
35     if((z==1||z==4||z==9||z==16)&&z!=0)
36     {
37         return(2/9)
38     }else if(z==0){
39         return(1/9)
40     }else{
41         return(0)
42     }
43 }
44 #loop to print the PMF(z)
45 for(i in 1:5)
46 {
47     print("PMF(z)")
48     print(PMFz(Z[i]))

```

```

49 }
50 #loop to calculate the multiplication of each
    element
51 for(i in 2:5)
52 {
53   V[i] <- Z[i]*PMFz(1)
54 }
55 V
56 variance<-sum(V)
57 variance#calculated total variance

```

R code Exa 2.4 Expectation Mean and Variance

```

1 #EX_2_4
2 #page 17
3 #let p=0.15
4 #(1-p)=0.85
5 X<-numeric(2)#initializing the array of size 2 to
    store the variable
6 px<-numeric(2)#initializing the array PX to store
    the probability of occurring the event
7 px1<-numeric(2)#initializing the array to store mean
    of X
8 px2<-numeric(2)#initializing the array to store mean
    of X^2
9 X<-c(1,0)#the events
10 PX<-c(0.15,0.85)#probability of events
11 for(i in 1:2)#loop to calculate the product of events
    and the probability of occurring the events
12 {
13   px1[i] <- X[i]*PX[i]
14 }
15 px1<-sum(px1)#calculated the mean
16 for(i in 1:2)#loop to calculate the product of
    square of event and the probability of occurring

```

```

    the events
17 {
18   px2[i] <- X[i]^2 * PX[i]
19 }
20 px2 <- sum(px2) #calculated the mean of square of the
    events
21 variance <- px2 - px1^2 #calculated the variance
22 variance
23 prod(PX)

```

R code Exa 2.5 Expectation Mean and Variance

```

1 #EX_2_5
2 #page 18
3 x <- numeric(6) #initialize the array of size 6 to
    store the events
4 px <- numeric(6) #initialize the array to store the
    probaility of events
5 ex <- numeric(6) #initialize the array to store the
    product of probailty and the events
6 ex1 <- numeric(6) # initialize the array to store the
    product between the probaility and the events
7 x <- c(1:6) #the events
8 px <- 1/6 #the probailty of events
9 for(i in 1:6) #loop to calcualte the product between
    the probailty of events and between the square
    of events
10 {
11   ex[i] <- prod(x[i], px)
12   ex1[i] <- prod(x[i]^2, px)
13 }
14 EX <- sum(ex) #calculated the sum of E(X)
15 EX1 <- sum(ex1) #calculated the sum of E(X^2)
16 EX
17 EX1

```



```

18 variance<-EX1-EX^2#calculated the variance
19 variance# print the calculated variance

```

R code Exa 2.7 Expectation Mean and Variance

```

1 #EX_2_7
2 #page 20
3 #Quiz problem
4 x1<-numeric(3)#creating the sample list
5 x2<-numeric(3)
6 p1<-numeric(3)
7 p2<-numeric(3)
8 ex1<-numeric(3)
9 ex2<-numeric(3)
10 x1<-c(0,100,300)#creating the sample space of the x1
    and x2
11 x2<-c(0,200,300)
12 p1<-c(0.2,0.8*0.5,0.5*0.8)#creating the sample space
    of the probabily of both x1 and x2
13 p2<-c(0.5,0.5*0.2,0.5*0.8)
14 #loop to calculate the multiplication of both
    probabilities
15 for(i in 1:3)
16 {
17   ex1[i]<-prod(x1[i],p1[i])
18   ex2[i]<-prod(x2[i],p2[i])
19 }
20 sum(ex1)#the expected values of the both x1 and x2
21 sum(ex2)

```

R code Exa 2.8 Expectation Mean and Variance

```

1 #EX_2_8

```

```

2 #page 21
3 # Average Speed Versus Average Time
4 t<-numeric(2)#initialize the array to store the time
5 p<-numeric(2)#initialize the array to store the
  probability
6 et<-numeric(2)#array to store the product
7 t<-c(2/5,2/30)#timre array
8 p<-c(0.6,0.4)#probability array
9 for(i in 1:2)#loop to calculate the product of
  probability and the time
10 {
11   et[i]<-prod(t[i],p[i])
12 }
13 sum(et)#calcualted the mean

```

R code Exa 2.9 Joint PMFs of Multiple Random Variable

```

1 #EX_2_9
2 #page 25
3 # Mean of the Binomial
4 x<-300#each 300 student get 1 PMF
5 p<-1/3#probability of each getting A
6 e<-prod(x,p)#the mean  $E[X]=\text{Sum}((i=1\text{to } 300)*1/3)$ 
7 e#printing the mean

```

R code Exa 2.11 Conditioning

```

1 #EX_2_11
2 #page 29
3 sum<-0
4 mat<-matrix(c(16/48,12/48,9/48,0,4/48,6/48,0,0,1/48)
  , nrow=3,ncol=3,byrow=T,dimnames= list(c(x=0:2),c
  (y=0:2)))

```

```

5 mat#the matrix of the joint PMF
6 x[2]
7 mat[1,1]
8
9 for(i in 2:3)
10 {
11   for(j in 2:3)
12   {
13     sum<-sum+mat[i,j]
14   }
15 }
16 sum# probabiltly of atleast one wrong

```

R code Exa 2.13 conditioning

```

1 #EX_2_13
2 #page 31
3 #x travel time of given message
4 #y the length of the given message
5 py<-function(y)
6 {
7   if(y==100)
8   {
9     return(5/6)
10  }else if(y==10^4){
11    return(1/6)
12  }
13 }#function to calculate the PMF (y)
14 pxy<-function(x)
15 {
16   if(x==0.01)
17   {
18     return(1/2)
19   }else if(x==0.1){
20     return(1/3)

```

```

21   }else if(x==1){
22       return(1/6)
23   }
24 }#function to calculate the PMF(x,10^2)
25 PXY<-function(x)
26 {
27   if(x==1)
28   {
29       return(1/2)
30   }else if(x==10){
31       return(1/3)
32   }else if(x==100){
33       return(1/6)
34   }
35 }#function to calculate the PMF(x,10^4)
36 #using the probability formula calculated each
    probability
37 px0.01<-py(100)*pxy(0.01)
38 px0.01
39 px0.1<-py(100)*pxy(0.1)
40 px0.1
41 px1<-(py(100)*pxy(1))+py(10^4)*PXY(1)
42 px1

```

R code Exa 2.14 Conditioning

```

1 #EX_2_14
2 #page 34
3 #E[X] is easily calculated using the total expectation
    theorem as
4 p<-numeric(3)
5 t<-numeric(3)
6 ex<-numeric(3)
7 p<-c(0.5,0.3,0.2)
8 t<-c(0.05,0.1,0.3)

```

```

9  for(i in 1:3)
10 {
11   ex[i]<-prod(p[i],t[i])
12 }
13 sum(ex)#E(x) is simply calculated using total
      expectation theorem

```

R code Exa 2.16 Independence

```

1  #EX_2_16
2  #page 34
3  install.packages("prob")
4  library(prob)
5  PXx<-numeric(3)
6  PXAx<-numeric(3)
7  toss<-matrix(nrow=4,ncol=4)
8  p<-numeric(4)
9  mat<-matrix(nrow=3,ncol=3)
10 coin<-c("H","T")
11 toss<-expand.grid(coin,coin)
12 toss#gives the sample space of all combination of 2
      independent toss
13 table(toss)
14 nrow(toss)#gives number of rows
15 ncol(toss)
16 probspace(toss)#gives the probability of each sample
      in sample space
17 mat<-noorder(probspace(toss))#table the repeating
      probabiltiy
18
19 mat[3]#takes the probabiltiy
20 #Let X be the number of heads and
21 #function to calculate the PMF of x
22 pxx<-function(x)
23 {

```

```

24   if(x==0)
25   {
26       return(1/4)
27   }else if(x==1){
28       return(0)
29   }else if(x==2){
30       return(1/2)
31   }
32 }
33 #function to calculate the conditional PMF
34 pxax<-function(x)
35 {
36     if(x==0)
37     {
38         return(1/2)
39     }else if(x==1){
40         return(0)
41     }else if(x==2){
42         return(1/2)
43     }
44 }
45 #loop to print the PMF(x)
46 for(i in 1:3)
47 {
48     PXx[i]<-pxx(i-1)
49     PXAx[i]<-pxax(i-1)
50 }
51 PXx#print the PMF(x)
52 PXAx#print the conditonal PMF(X|A)

```

R code Exa 2.18 Independence

```

1  #EX_2_18
2  #page 40
3  n<-100000

```

```

4  binomial<-numeric(2)
5  #Xi be a random variable that encodes the response
    of the i th person:
6  Xi<-c(1,0)#1 if i th person approves C's performnce
7  #0 if the ith person dissapproves C's performance
8  binomial<-rbinom(Xi,n,0.5)#creating a random
    variable of the approval of c's performance
9  binomial#printing the random variable
10 p<-1/2#the common mean of appproval
11 q<-1-p#the common mean of dissapproval
12 sn<-binomial[1]/n#sn is the mean from the sample
    random variable
13 sn
14 Esn<-p#printing the expectation of the mean of
    sample space is the common mean
15 varsn<-prod(p,q)/n
16 varsn#variance of the mean

```

Chapter 3

General Random Variable

R code Exa 3.2 Continuous Random Variables And PDFs

```
1 #EX_3_2
2 #page 5
3 #Piecewise Constant PDF
4 Fx<-numeric(11)
5 f <- function(c1)c1#representing the function of
   constant variable to integrate
6 f2<-function(c2)c2
7 cum<-integrate(f,15,20)
8 p_sunnyday<-cum$value/17.5#to calculate the
   probabiltiy of sunny day
9 p_sunnyday
10 cum2<-integrate(f2,20,25)
11 p_rainyday<-(cum2$value/22.5)#to calcualte the
   probability of rainy day
12 p_rainyday
13 c1<-(2/3)/p_sunnyday#calculting the c1
14 c2<-(1/3)/p_rainyday#calculating the c2
15 fx<-c(c1,c2)#sample space of fx(x)
16 c1
17 c2
18 fx
```

R code Exa 3.3 Continuous Random Variables And PDFs

```
1 #EX_3_3
2 #page 6
3 #function to print the sample space of fX(x)
4 fx<-function(x)
5   {
6     if(0<x&& x<=1)
7       {
8         return(1/(sqrt(x)*2))
9       }
10    else
11      {
12        return(0)
13      }
14  }
15 FX<-c(fx(0),fx(1))
16 FX #PDF of random variable x
17 int<-integrate(fx,0,1)
18 int$value#PDF of fX(x)
```

R code Exa 3.4 Continuous Random Variables And PDFs

```
1 #EX_3_4
2 #page 8
3 #function to calculate the gx
4 gx<-function(x)
5   {
6     if(x<=1/3)
7       {
8         return(1)
```

```

9      }else if(x>1/3){
10          return(2)
11      }
12  }
13  #function to calcualte the PMF Py
14  pY<-function(gx)
15  {
16      if(gx==1)
17      {
18          return(1/3)
19      }else if(gx==2){
20          return(2/3)
21      }
22  }
23  #to calcualte the E(Y)
24  EY<-sum(pY(1)*gx(1/3),pY(2)*gx(2/3))
25  EY

```

R code Exa 3.5 Continuous Random Variables And PDFs

```

1  #EX_3_5
2  #page 10
3  lamda<-1/10
4  px<-function(a,lamda)
5  {
6      return(exp(1)^(-lamda*a))
7  }
8  #function to calculate the probability
9  PX<-function(x)
10 {
11     if(1/4<=x || x<=3/4)
12     {
13         return(px(1/4,1/10)-px(3/4,1/10))
14     }
15 }

```

```
16 PX(1/4)#probabilty of meteorite lands between 6 am
    and 6 pm on the first day
```

R code Exa 3.7 cumulative distribution Function

```
1 #EX_3_7
2 #page 15
3 px<-numeric(10)
4 #we compute the FX(k) first and then the PMF
5 #function to calculate the FX(k)
6 fx<-function(k)
7 {
8   return((k/10)^3)
9 }
10 #function to calculate the FX(k-1)
11 fx1<-function(k)
12 {
13   return(((k-1)/10)^3)
14 }
15 #to print the PMF
16 for(i in 1:10)
17 {
18   px[i]<-fx(i)-fx1(i)
19 }
20 px#PMF calculated
```

R code Exa 3.8 Normal Random Variable

```
1 #EX_3_8
2 #page 19
3 #Using the Normal Table
4 # Its CDF is denoted by phi,
5 pi<-3.14
```

```

6 #function to calculate the CDF normal random
  variable
7 f<-function(t)
8 {
9   return((1/sqrt(2*pi))*exp(1)^(-(t^2)/2))
10 }
11 #to calculate the CDF of Y<=0.5
12 f_0.5<-integrate(f,-Inf,0.5)
13 f_0.5$val
14 #to calculate the CDF of Y<=-0.5
15 f_negative_0.5<-(1-f_0.5$val)
16 f_negative_0.5
17 sd<-20#standard deviation
18 mean<-60#mean
19 y<-(80-mean)/20#calculating the Y
20 y#Y is 1
21 #calculate the CDF of Y<=80-60/20 is phi(1)
22 f_1<-integrate(f,-Inf,1)
23 f_1$val
24 #to calculate the CDF of Y>=80-60/20
25 p_x_greater_80<-(1-f_1$val)
26 p_x_greater_80

```

R code Exa 3.9 Normal Random Variable

```

1 #EX_3_9
2 binary_message<-c(-1,1)#the message send may be -1,1
3 mean<-0#mean and standard deviation is given
4 sd<-1
5 pi<-3.14
6 variance<-sd^2
7 #function to calculate the normal table
8 f<-function(y)
9 {
10   return((1/sqrt(2*pi))*e^((-y^2)/2))

```

```

11 }
12 #to calculate the CDF of sending sending message is
    -1 is normal table phi(1)
13 f1<-integrate(f,0,1)
14 f1$val
15 #probabilty of error
16 p_N_greater_1<-1-f1$val
17 p_N_greater_1

```

R code Exa 3.11 conditioning on an event

```

1 #EX_3_11
2 #page 24
3 # Mean and Variance of a Piecewise Constant PDF
4 x<-readline(prompt="x: ")#enter the random variable
    x in the console
5 #this enters the constant PDF of x
6 if(0<=x&&x<=1)
7 {
8   pA1<-1/3
9   print("pA1:")
10  return(pA1)
11 }else if(1<x&&x<=2){
12   pA2<-2/3
13   print("pA2:")
14   return(pA2)
15 }else {
16   return(0)
17 }
18 # the mean of a uniform random variable on an
    interval [a,b] is ( a+b)/2 and its second moment
    is (a2 +ab+b2)/3.
19 ex<-function(a,b)#function to return the mean
20 {
21   return(sum(a,b)/2)

```

```

22 }
23 ex2<-function(a,b)#function to return the variance
24 {
25     return(sum(a^2,prod(a,b),b^2)/3)
26 }
27 ex(0,1)#mean when x in between 0&1
28 ex2(0,1)#variance when x in 0&1
29 ex(1,2)#mean of x in 1&2
30 ex2(1,2)#variance of x in 1&2

```

R code Exa 3.12 conditioning on an event

```

1 #EX_3_12
2 #page 25
3 #metro train problem
4 A1<-numeric(5)
5 A2<-numeric(15)
6 A1<-sample(c(1:5),replace = FALSE)
7 A2<-sample(c(1:15),replace = FALSE)
8 pA1<-1/4
9 fyA1<-1/length(A1)
10 fyA2<-1/length(A2)
11 for(i in 1:15)
12 {
13     if(A2[i]<5)
14     {
15         print(sum(prod(pA1,fyA1),prod((1-pA1),fyA2)))
16     }else{
17         print(prod((1-pA1),fyA2))
18     }
19 }

```

R code Exa 3.22 Derived Distributions

```

1 #EX_3_22
2 #page 40
3 x<-numeric(30)
4 gx<-numeric(30)
5 x<-runif(30,30,60)
6 x#to print the uniform distribution of time between
   30,60
7 #to print the g(x),PDF,CDF of X
8 for(i in 1:30)
9 {
10   print("x:")
11   print(x[i])
12   gx[i]<-180/x[i]
13   print("gx")
14   print(gx[i])
15   if(30<=x[i]||x[i]<=60){
16     print("fx")
17     print(1/30)
18     print("FX")
19     print((x[i]-30)/30)
20   }else if(60<=x[i]){
21     print("FX")
22     print(1)
23   }else if(30>=x[i]){
24     print("FX")
25     print(0)
26   }else{
27     print("fx")
28     print(0)
29   }
30 }

```

Chapter 4

Further Topics on Random Variables and Expectations

R code Exa 4.1 Transforms

```
1 #EX_4_1
2 #page 2
3 #Transform
4 x<-c(2,3,5)#creating the uniform random variable of
   x
5 px<-c(1/2,1/6,1/3)#creating the pdf function of x
6 Ms<-sum(1/2*exp(1)^2,1/6*exp(1)^3,1/3*exp(1)^5)#
   calculating the transform
7 Ms
8 par(mfrow=c(2,2))#creating the space for the plots
   to be plotted
9 curve((1/2*exp(1)^(2*x)), -10,10, col="red")#curve of
   the Ms function of x=2
10 curve(1/3*exp(1)^(3*x), -10,10, col="violet")#curve of
   the Ms function of x=3
11 curve(1/5*exp(1)^(3*x), -10,10, col="black")#curve of
   the Ms function of x=5
12 plot(x,px,type="h", col="red")#plot the x vs px graph
```

R code Exa 4.4 Transforms

```
1 #EX_4_4
2 #page 4
3 #function to calculate the exponential randm
  variable of x
4 exponential_transform<-function(l,s){
5   return(1/1-s)
6 }
7 #function to calculate the exponential transform of
  y
8 y<-function(a,b,l,s)
9 {
10  (exp(1)^b*s)*1/1-a*s
11 }
12 print("1/1-s")
13 exponential_transform(1,0)#printing the both
  transform by giving certain values
14 y(2,3,1,1)
```

R code Exa 4.5 Transforms

```
1 #EX_4_5
2 #page 5
3 x<-numeric(3)
4 px<-numeric(3)
5 derrivative<-numeric(3)
6 x<-c(2,3,5)#creating the uniform random variable of
  x
7 px<-c(1/2,1/6,1/3)#creating the pdf function of x
8 Ms<-sum(1/2*exp(1)^2,1/6*exp(1)^3,1/3*exp(1)^5)#
  calculating the transform
```

```

9  ex<-expression(px*x*exp(1)^x*s)
10 derrivative<-D(ex,"s")
11 derrivative
12 mx<-expression((1/2*exp(1)^(2*s))+(1/6*exp(1)^(3*s))
    +(1/3*exp(1)^(5*s)))#to print the mx
13 ex<-D(mx,"s")#it gives the derrivative of mx
14 ex
15 ex2<-D(ex,"s")#it gives the second derrivative
16 ex2

```

R code Exa 4.6 Transforms

```

1  #EX_4_6
2  #page 6
3  x<-numeric(3)
4  px<-numeric(3)
5  derrivative<-numeric(3)
6  x<-c(2,3,5)#creating the uniform random variable of
    x
7  px<-c(1/2,1/6,1/3)#creating the pdf function of x
8  Ms<-sum(1/2*exp(1)^2,1/6*exp(1)^3,1/3*exp(1)^5)#
    calculating the transform
9  Ms
10 mx<-expression((1/2*exp(1)^(2*s))+(1/6*exp(1)^(3*s))
    +(1/3*exp(1)^(5*s)))#giving the expression
11 ex<-D(mx,"s")#finding the first derrivative of
    expression
12 ex
13 ex2<-D(ex,"s")#finding the second derrivative of the
    expression
14 ex2
15 #finding the values of transforms with s=1 and s=0
16 Mx<-((1/2*exp(1)^(2))+(1/6*exp(1)^(3))+(1/3*exp(1)
    ^5))#s=1
17 Mx

```

```

18 dMx<-(1/2*2+(1/6*3)+(1/3*5))#s=0
19 dMx
20 d2Mx<-(1/2*4)+(1/6*9)+(1/3*25)#s=0
21 d2Mx

```

R code Exa 4.7 Transforms

```

1 #EX_4_7
2 #page 8
3 x<-c(-1,0,4,5)
4 px<-c(1/4,1/2,1/8,1/8)
5 Ms<-expression((1/4)*exp(1)^(-1*s)+(1/2)*exp(1)^(0*s
    )+(1/8)*exp(1)^(4*s)+(1/8)*exp(1)*(5*s))#
    expressing the transform function
6 s<-1#giving free variable s as 1
7 Ms1<-function(a,b){
8   return((1/a)*exp(1)^b)
9 }#function for calculating the transform at s=1
10 Ms<-sum(Ms1(1/4,-1),Ms1(1/2,0),Ms1(1/8,4),Ms1(1/8,5)
    )#the value of transform at free variable =1
11 Ms

```

R code Exa 4.9 Transforms

```

1 #EX_4_9
2 #page 9
3 lamda<-c(6,4)#expressing lamda
4 p<-1/3#initiating probabiltly of selecting one teller
5 s<-1#expressed the free variable as 1
6 fx<-function(x)
7 {
8   return((exp(1)^x)*((2/3)*6*exp(1)^(-6*x)+(1/3)*4*
    exp(1)^(-4*x)))

```

```

9 }#function for calculating the M(s)
10 Ms<-integrate(fx,0,Inf) #integrate to get the M(s)
11 Ms$val#giving the value of M(s)

```

R code Exa 4.13 Transforms

```

1 #EX_4_13
2 #page 14
3 x<-numeric(3)#initializing the variables x,y,w
4 y<-numeric(3)
5 py<-numeric(3)#initializing the probability of each
  variable
6 w<-numeric(5)
7 py1<-numeric(3)
8 pw1<-numeric(5)
9 x<-c(1:3)#representing the sample space of each
  variable
10 y<-c(0:2)
11 w<-c(1:5)
12 px<-function(x)#function to print the probabilty of
  x
13 {
14   if(1<=x&& x<=3)
15     {
16       return(1/3)
17     }
18   else
19     {
20       return(0)
21     }
22 }
23 py<-function(y)#function to print the sample space
  of probability of y
24 {
25   if(y==0){

```

```

26     return(1/2)
27 }else if(y==1){
28     return(1/3)
29 }else if(y==2){
30     return(1/6)
31 }else{
32     return(0)
33 }
34 }
35 for(i in 1:3)#loop to print the probabiltly of y
36 {
37     py1[i]<-py(i-1)
38 }
39 py1#printing the probability
40 pw<-function(w)#function to print the sample space
    of probabiltly of w
41 {
42     if(w==1)
43     {
44         return(px(1)*py(0))
45     }else if(w==2){
46         return(sum(prod(px(1),py(1)),prod(px(2),py(0))))
47     }else if(w==3){
48         return(sum(prod(px(1),py(2)),prod(px(2),py(1)),
            prod(px(3),py(0))))
49     }else if(w==4){
50         return(sum(prod(px(2),py(2)),prod(px(3),py(1))))
51     }else if(w==5){
52         return(prod(px(3),py(2)))
53     }else{
54         return(0)
55     }
56 }
57 for(i in 1:5)#loop to print the probabiltly of w
58 {
59     pw1[i]<-pw(i)
60 }
61 pw1#printing the probabiltly

```

R code Exa 4.16 conditional expectation as random variable

```
1 #EX_4_16
2 #page 23
3 l<-8#define length of stick as 8
4 vary<-function(l)
5 {
6   return((l^2)/12)
7 }#function to calculate var(y)
8 f<-function(y)
9 {
10   return((y^2)/(12*8))
11 }
12 varxy<-1/4*vary(l)
13 varxy#to print var(x|y)
14 integral<-integrate(f,0,l)#to calculate E(var(x|y))
15 Evarxy<-integral$val
16 Evarxy
17 varx<-sum(Evarxy,varxy)
18 varx#to final calculation of var(x)
```

R code Exa 4.20 conditional expectation as random variable

```
1 #EX_4_20
2 #page 24
3 x<-numeric(3)#initializing the variables
4 x1<-numeric(3)
5 Y<-numeric(3)
6 x<-c(0:2)#sample space of x
7 fx<-c(1/3,1/3,2/3)#sample space of fx
8 #function to calculate the sample space of y
```

```

9  y<-function(x)
10 {
11   if(x<1){
12     return(1)
13   }else if(x>=1){
14     return(2)
15   }
16 }
17 #loop to print the y sample space
18 for(i in 1:3)
19 {
20   print(y(i-1))
21 }
22 Exy<-c(1/2,3/2)#sample space of Exy
23 #function to calculate the probability of Exy
24 pExy<-function(Exy)
25 {
26   if(Exy==1/2){
27     return(1/3)
28   }else if(Exy==3/2){
29     return(2/3)
30   }
31 }
32 MeanExy<-7/6#mean of E(x|y)
33 varExy<-sum(prod(pExy(1/2),((1/2-MeanExy)^2)),prod(
    pExy(3/2),((3/2-MeanExy)^2)))#calculating the
    variance of E(x|y)
34 varExy
35 varxy<-1/12
36 Evarxy<-1/12
37 varx<-sum(Evarxy,varExy)#calculating the variance of
    x
38 varx

```

R code Exa 4.21 Sum of a Random Number of Independent Random Variable

```
1 #EX-4_21
2 #page 27
3 gas<-runif(1000,0,1000)
4 p<-1/2
5 s<-1
6 MNs<-1/8*(1+exp(1)^3)#the transform of binomial
   random variable of N open gas station
7 MNs
8 Mxs<-(((exp(1)^(1000*s))-1)/(1000*s))#transform of
   amount of gas available
9 Mxs
10 Mys<-(1/8)*(1+Mxs)^3
11 Mys#transform associated with y
```

Chapter 5

Stochastic Processes

R code Exa 5.5 Bernolli process

```
1 #EX_5_5
2 #page 13
3 p<-0.01
4 q<-1-p
5 pz1<-numeric(4)#representing a sample list
6 px1<-numeric(4)
7 #function to calculate PX(x)
8 px<-function(x)
9 {
10   if(x==0)
11   {
12     return((1-0.01)^100)
13   }else if(x==2||x==5||x==10){
14     return(prod(factorial(n),(p^x),q^(n-x))/prod(
15       factorial(n-x),factorial(x)))
16   }
17 }
18 #printing the PX(x)
19 px1<-c(px(0),px(2),px(5),px(10))
20 #function to calculate PZ(x)
21 pz<-function(x)
```

```

21 {
22   (exp(1)^-1)/factorial(x)
23 }
24 #printing the PZ(x)
25 pz1<-c(pz(0),pz(2),pz(5),pz(10))
26 px1
27 pz1

```

R code Exa 5.6 The Poisson Process

```

1 #EX_5_6
2 #page 14
3 p<-0.0001#initializing the variables p,n,n1
4 n<-(log(0.999,base=exp(1)))/(log(0.9999,base=exp(1))
   )
5 n1<-(-log(0.999,base=exp(1)))/p
6 Ps<-1-(1-p)^n#calculating the probability of free
   variable S
7 Ps
8 poisS<-1-exp(1)^-(p*n1)#calculating the probability
   of free variable using the poisson approximation
9 poisS

```

R code Exa 5.7 The Poisson Process

```

1 #EX_5_7
2 #page 19
3 lamda<-0.2# initializing the variables lamda
4 #function to calculate the probability using the
   poisson PMF
5 PMF<-function(lamda,T,k)
6 {

```

```

7   return(prod((lamda*T)^k,(exp(1)^-(lamda*T)))/
      factorial(k))
8 }
9 PMF(0.2,1,0)#PMF of different lamda,Time,and k value
      is being calculated
10 PMF(0.2,1,1)
11 PMF(0.2,24,0)
12 PMF24<-(PMF(0.2,1,0))^24#use poisson PMF
13 PMF24

```

R code Exa 5.9 The Poisson Process

```

1 #EX_5_9
2 #page 19
3 mue1<-5
4 mue2<-3
5 # the PMF of the total number of accidents between
      8 am and 11 am?
6 PMF<-sum(5,(3*2))
7 PMF# sum of independent poisson random variable with
      parmeters 5& 3*2

```

R code Exa 5.12 The Poisson Process

```

1 #EX_5_12
2 #page 24
3 n<-56
4 lamda<-2#callers depart with poisson process a rate
      of lamda
5 #the waiting time Y
6 EY<-n/lamda
7 #the function to calculate the probabiltiy you have
      to wait for more than an hour

```

```

8 PY60<-function(y)
9 {
10   return((lamda^n)*(y^(n-1))*(exp(1)^(-lamda*y))/
           factorial(n-1))
11 }
12 probability<-integrate(PY60, 60, Inf)#the integral
    function to calculate the probability of waiting
    more than an hour
13 probability$val

```

R code Exa 5.17 The Poisson Process

```

1 #EX_5_17
2 #page 30
3 # Random incidence in a non-Poisson arrival process
4 T1<-15
5 T2<-45
6 #person arrives at interarrival time of 15 minute
    with probabiltiy 1/4
7 p1<-1/4
8 #person arrives at interarrival time of 45 is of
    probability 3/4
9 p2<-3/4
10 #the expected value of chosen interarrival time is
11 T<-sum((T1*p1),(T2*p2))
12 T

```

Chapter 6

Markov Chains

R code Exa 6.1 the discrete time markov chains

```
1 #EX_6_1
2 #page 2
3 library(markovchain)#loading libraries
4 library(diagram)
5 p<-c(0.8,0.2,0.6,0.4)
6 probability<-matrix(p,nrow=2,ncol=2,byrow=T)
7 probability#probabilty matrix
8 plotmat(probability)
```

R code Exa 6.2 Discrete Time Markov Chains

```
1 #EX_6_2
2 #page 4
3 library(markovchain)#loading libraries
4 library(diagram)
5 matrix<-matrix(NA,nrow=4,ncol=4,byrow=T)#creating
  the sample matrix
6 #loop to enter the probabilty in the matrix
```

```

7  for(i in 1:4)
8  {
9    for(j in 1:4)
10   {
11     if(i==1&&j==1)
12     {
13       matrix[i,j]<-1
14     }else if(i==4&&j==4){
15       matrix[i,j]<-1
16     }else if(i==j&&i>1){
17       matrix[i,j]<-0.4
18     }else if(i>=2&&(j==(i-1)||j==(i+1))){
19       matrix[i,j]<-0.3
20     }else{
21       matrix[i,j]<-0
22     }
23   }
24 }
25 matrix#printing the matrix
26 plotmat(matrix)#markov chain representation of the
    matrix

```

R code Exa 6.4 Steady State Behavior

```

1  #EX_6_4
2  #page 15
3  p<-c(0.8,0.6,0.2,0.4)
4  prob<-matrix(p,nrow=2,ncol=2,byrow=T)#matrix of
    multiples of pi
5  prob
6  pi<-c(0.75,0.25)
7  mat<-matrix(pi,nrow=2,ncol=1,byrow=T)#created matrix
    of pi giving value of pi
8  mat
9  solve(prob,mat)#generate the solution of the pi

```


R code Exa 6.13 Absorption Probabilities and Expected Time to Absorption

```
1 #EX_6_13
2 #page 30
3 m<-4
4 i<-c(2,3)
5 mue<-c(0.6,-0.3,0.7,-0.4)#representing the
    multiplication vectors with mu
6 mat<-matrix(mue,nrow=2,ncol=2,byrow=T)#representing
    to matrix
7 mat
8 b<-matrix(c(1,1),nrow=2,ncol=1,byrow=T)#the solution
    matrix
9 b
10 m<-solve(mat,b)#this solve the both matrix to give
    the value of mu
11 mu<-matrix(m,nrow=2,ncol=1,byrow=T)#representing
    the value of mu in a matrix
12 row.names(mu)<-c("mu1","mu2")
13 mu#represent the values of mu1, mu2 in the matrix "
    mu"
14 #let m=5
15 transition<-c(1,0,0,0.3,0.4,0.3,0,0.3,0.4)
16 transition_mat<-matrix(transition,nrow=3,ncol=3,
    byrow=T)
17 plotmat(transition_mat)#markov chain representation
    of the transition matrix
```

R code Exa 6.14 more general markov chains

```
1 #EX_6_14
```



```

2 #page 32
3 p<-c(0.8,0.2,0.6,0.4)
4 mat<-matrix(p,nrow=2,ncol=2,byrow=T)#probabilty
    matrix
5 mat
6 t<-c(0,0.6,1,0.2)
7 T<-matrix(t,nrow=2,ncol=2,byrow=T)#matrix to
    represent the t matrix
8 T
9 b<-matrix(c(1,1),nrow=2,ncol=1,byrow=T)
10 b
11 t1<-solve(T,b)#calculate the first passage time to
    state 1 from state2
12 b1<-matrix(c(0,1),nrow=2,ncol=1,byrow=T)
13 t2<-solve(T,b1)#calculate the mean recurrence time
14 t1[1,1]#mean first passage time to statel starting
    from sate2
15 t2[1,1]#mean recurrence time to state 1

```

Chapter 7

Limit Theorems

R code Exa 7.1 Some Useful Inequalities

```
1 #EX_7_1
2 #page 3
3 EX<-2#expected mean
4 #Markov inequality asserts that
5 #function to calculate the Markov Inequality
6 PX<-function(x)
7 {
8   if(x>=2&& x<3)
9   {
10     return(1)
11   }else if(x>=3&& x<4){
12     return(2/3)
13   }else if(x>=4){
14     return(2/4)
15   }
16 }
17 #function to calculate the normal probability
18 px<-function(x)
19 {
20   if(x>=2&& x<3)
21   {
```

```

22     return(0.5)
23   }else if(x>=3&& x<4){
24     return(0.25)
25   }else if(x>=4){
26     return(0)
27   }
28 }
29
30 c<-c(PX(2),PX(3),PX(4),px(2),px(3),px(4))
31 compare<-matrix(c,nrow=3,ncol=2,byrow=T)
32 compare#matrix to compare both Markov Inequality and
        normal probability

```

R code Exa 7.4 convergence in probability

```

1 #EX_7_4
2 #page 6
3 #polling
4 PMnp<-function(n,e)
5 {
6   return(1/(4*n*e^2))
7 }#function to calculate the chebyshev inequality
8 PMnp(100,0.1)
9 PMnp(1000000,0.01)# calculated the chebyshev
        inequality

```

R code Exa 7.8 The central Limit Theorem

```

1 #EX_7_8
2 #page 11
3 n<-100#number of packages loaded
4 p1<-5#weights are uniformly distributed between 5and
        50

```

```

5 p2<-50
6 pi<-3.14
7 #mean and variance of single package
8 mue<-sum(p1,p2)/2
9 mue
10 var<-sum(50,-5)^2/12
11 var
12 z<-(3000-prod(100,mue))/sqrt(var*100)#normalized
    value of the mean and variance
13 z
14 #function to calculate the CDF normal random
    variable
15 f<-function(t)
16 {
17   return((1/sqrt(2*pi))*exp(1)^(-(t^2)/2))
18 }
19 phi1.92<-integrate(f,-Inf,1.92)#calculate the CDF of
    normal random variable from the normal table
20 phi1.92$val
21 p_greater_3000<-1-phi1.92$val#the desired
    probability that the total weight exceeds 3000
    pounds
22 p_greater_3000

```

R code Exa 7.9 The central Limit Theorem

```

1 #EX_7_9
2 #page 12
3 #processing time is independent random variable
    between 1 and 5
4 mue<-3#the variance and the mean
5 var<-16/12
6 var
7 n<-100#the number of parts
8 z<-(320-(n*mue))/sqrt(var*n)

```

```

9 z#calculated the normalized value
10 f<-function(t)
11 {
12   return((1/sqrt(2*pi))*exp(1)^(-(t^2)/2))
13 }#function to calculate the CDF normal random
    variable
14 phi1.73<-integrate(f,-Inf,1.73)#the desired
    approximation gives p(S100>320) it is t from the
    normal table
15 phi1.73$val

```

R code Exa 7.10 The central Limit Theorem

```

1 #EX_7_10
2 #page 12
3 n<-100#consider the case n=100 and e=0.1
4 e<-0.1
5 z<-function(e,n)
6 {
7   return(2*e*sqrt(n))
8 }#function to calculate the standardized value
9 Z<-z(e,n)
10 Z#2*0.01*sqrt(n)>= 1.96
11 f<-function(t)
12 {
13   return((1/sqrt(2*pi))*exp(1)^(-(t^2)/2))
14 }#function to calculate the CDF normal random
    variable
15 phi<-integrate(f,-Inf,Z)#the normal CDF of 2 from
    normal table
16 phi$val#phi(2)
17 p<-2-(2*phi$val)
18 p#2-2phi(2*0.01 sqrt(n))<= 0.05

```

R code Exa 7.11 The central Limit Theorem

```
1 #EX_7_11
2 #page 15
3 n<-36
4 p<-0.5
5 P21<-numeric(22)
6 comb<-function(n,x)
7 {
8   return(factorial(n)/factorial(n-x)/factorial(x))
9 }#function to calculate the combination
10 for(i in 1:22)
11 {
12   P21[i]<-comb(n,(i-1))
13 }
14 P21#exact valuep(Sn<=21)
15 p<-sum(P21*(0.5^36))
16 f<-function(t)
17 {
18   return((1/sqrt(2*pi))*exp(1)^(-(t^2)/2))
19 }#function to calculate the CDF normal random
    variable
20 p21<-integrate(f,-Inf,1)
21 p21$val#the central limit approximation
22 P21<-integrate(f,Inf,1.17)#Using the proposed re???
    nement
23 P21$val#which is much closer to the exact value
24 z1<-(19.5-18)/3
25 z2<-(18.5-18)/3
26 p1<-integrate(f,-Inf,z1)
27 p2<-integrate(f,-Inf,z2)
28 p19<-p1$val-p2$val
29 p19# de Moivre – Laplace formula also allows us to
    approximate the probability of a single value
```

```
30 P19<-comb(n,19)*(0.5^36)
31 P19#exact value P(Sn=19)
```
