R Textbook Companion for A First Course in Probability by Sheldon Ross[1]

Created by
Anuj Singh
B.Tech.
Computer Science and Engineering
Indian Institute of Technology, Bhilai
Cross-Checked by
R TBC Team

June 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: A First Course in Probability

Author: Sheldon Ross

Publisher: Pearson, USA

Edition: 8

Year: 2008

ISBN: 013603313X

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

Combinatorial Analysis

R code Exa 2.a Mother and child of the year

```
# Page No. 2

no_of_women = 10
children_per_women = 3

no_of_choices = no_of_women * children_per_women

print(no_of_choices)
```

R code Exa 2.b Committee selection

```
# Page No. 2

no_of_freshmen = 3
no_of_sophmores = 4
no_of_juniors = 5
no_of_seniors = 2

```

```
ans = no_of_seniors * no_of_juniors * no_of_
    sophmores * no_of_freshmen

print(ans)
```

R code Exa 2.c Licence plates

```
# Page No. 3

no_of_letters = 26
no_of_digits = 10

ans = (no_of_digits^4) * (no_of_letters^3)

print(ans)
```

R code Exa 2.e Licence plates without repititions

```
# Page No. 3

no_of_letters = 26
no_of_digits = 10
ans = 1

for(i in 0:3)
{
  ans = ans * (no_of_digits - i);
}
for(i in 0:2)
{
  ans = ans * (no_of_letters - i);
}

```

```
print(ans)
```

R code Exa 3.a Batting orders

```
# Page No. 3

no_of_players = 9

ans = factorial(no_of_players)

print(ans)
```

R code Exa 3.b.a Universal rankings

```
# Page No. 4

no_of_women = 4
no_of_men = 6

ans = factorial(no_of_men + no_of_women)

print(ans)
```

R code Exa 3.b.b Gender rankings

```
# Page No. 4

no_of_women = 4
no_of_men = 6

```

```
ans = factorial(no_of_men) * factorial(no_of_women)

print(ans)
```

R code Exa 3.c Book arrangements

```
# Page No. 4

no_of_math_books = 4
no_of_chem_books = 3
no_of_history_books = 2
no_of_lang_book = 1
no_of_types_of_books = 4

no_of_orderings = factorial(no_of_types_of_books)
ans = no_of_orderings * factorial(no_of_math_books)
   * factorial(no_of_chem_books) * factorial(no_of_
   history_books) * factorial(no_of_lang_book)

print(ans)
```

R code Exa 3.d Pepper permuatations

```
# Page No. 4

no_of_p = 3
no_of_r = 1
no_of_e = 2

total_chars = no_of_p + no_of_r + no_of_e
ans = factorial(total_chars) / (factorial(no_of_p) *
    factorial(no_of_r) * factorial(no_of_e))

```

```
print(ans)
```

R code Exa 3.e Chess tournament outcomes

```
# Page No. 5

no_of_russians = 4
no_of_americans = 3
no_of_brits = 2
no_of_brazillians = 1

ans = factorial(no_of_brazillians + no_of_americans
   + no_of_brits + no_of_russians) / (factorial(no_
   of_russians) * factorial(no_of_brits) * factorial
   (no_of_americans) * factorial(no_of_brazillians))

print(ans)
```

R code Exa 3.f Flag signals

```
# Page No. 5

no_of_white_flags = 4
no_of_red_flags = 3
no_of_blue_flags = 2

ans = factorial(no_of_white_flags + no_of_red_flags
   + no_of_blue_flags) / (factorial(no_of_white_
   flags) * factorial(no_of_red_flags) * factorial(
   no_of_blue_flags))

print(ans)
```

R code Exa 4.a Possible committees

```
# Page No. 6

total_people = 20
committee_size = 3

ans = choose(total_people, committee_size)

print(ans)
```

R code Exa 4.b Committee selection with genders and feuds

```
# Page No. 6

total_men = 7
req_men = 3
total_women = 5
req_women = 2

ans1 = choose(total_men, req_men) * choose(total_
    women, req_women)

print(ans1)

no_of_feuding_men = 2

feuding_groups = choose(no_of_feuding_men, no_of_
    feuding_men) * choose(total_men - no_of_feuding_
    men, req_men - no_of_feuding_men)
ans2 = (choose(total_men, req_men) - feuding_groups)
     * choose(total_women, req_women)
```

```
print(ans2)
```

R code Exa 5.a Possible divisions

```
# Page No. 10

patrollers = 5
station_officers = 2
reserve = 3

ans = factorial(patrollers + station_officers +
   reserve) / (factorial(patrollers) * factorial(
   station_officers) * factorial(reserve))

print(ans)
```

R code Exa 5.b Division into team A and team B

```
# Page No. 10

teamA_size = 5
teamB_size = 5

ans = factorial(teamA_size + teamB_size) / (
   factorial(teamA_size) * factorial(teamB_size))

print(ans)
```

R code Exa 5.c Basketball divisions

```
# Page No. 10

team_size = 5
no_of_teams = 2

ans = factorial(team_size * no_of_teams) / ((
   factorial(team_size)^2) * factorial(no_of_teams))

print(ans)
```

R code Exa 5.d Tournaments

```
# Page No. 11

no_of_players = 8
no_of_winners = 4

ans1 = factorial(no_of_winners) * choose(no_of_
   players,no_of_winners)

cat("Ans to a)", ans1, "\n")

ans2 = factorial(no_of_players)

cat("Ans to b)", ans2, "\n")

# The answer may vary due to difference in
   representation.
```

R code Exa 6.a Distinct non-negative integral solutions

```
# Page No. 13

```

```
const = 3
no_of_vars = 2

ans = choose(const + no_of_vars - 1, no_of_vars - 1)

print(ans)
```

R code Exa 6.b Investments

```
# Page No. 13

no_of_vars = 4
const = 20

ans1 = choose(const + no_of_vars - 1, no_of_vars -
    1)

print(ans1)

no_of_vars_updated = 5

ans2 = choose(const + no_of_vars_updated - 1, no_of_
    vars_updated - 1)

print(ans2)
```

Chapter 2

Axioms of Probability

R code Exa 3.b Even number on a dice roll

```
# Page No. 28

p_of_a_side = 1/6
no_of_even_sides = 3

ans = p_of_a_side * no_of_even_sides

print(ans)

# The answer may vary due to difference in
   representation.
```

R code Exa 4.a Books and vacations

```
# Page No. 30

p_book1 = 0.5
p_book2 = 0.4
```

```
p_both_books = 0.3

ans = 1 - (p_book1 + p_book2 - p_both_books)

print(ans)
```

R code Exa 5.a Sum of dice rolls

```
# Page No. 34

library(MASS)

favourable_outcomes = 6
total_outcomes = 36

ans = favourable_outcomes / total_outcomes

print(fractions(ans))
```

R code Exa 5.b Drawing balls from a bowl

```
# Page No. 34

library(MASS)

total_white_balls = 6
total_black_balls = 5
fav_no_of_white_balls = 1
fav_no_of_black_balls = 2
drawn_balls = 3

ans = (choose(total_white_balls, fav_no_of_white_
   balls) * choose(total_black_balls, fav_no_of_
```

```
    black_balls)) / choose(total_black_balls + total_
    white_balls, drawn_balls)

print(fractions(ans))
```

R code Exa 5.c Committee selection

```
# Page No. 35

library(MASS)

total_men = 6
total_women = 9
fav_no_of_men = 3
fav_no_of_women = 2
commmittee_size = 5

ans = (choose(total_women, fav_no_of_women) * choose
    (total_men, fav_no_of_men)) / choose(total_men +
    total_women, commmittee_size )

print(fractions(ans))
```

R code Exa 5.f Probability of a straight

```
# Page No. 36

hand_size = 5
total_cards = 52

no_of_straights = 10 * (4^5 - 4)
ans = no_of_straights / choose(total_cards, hand_
    size)
```

```

print(ans)
```

R code Exa 5.g Probability of full house

```
# Page No. 37

total_hands = 5
total_cards = 52

total_full_houses = 13 * 12 * choose(4, 2) * choose
    (4, 3)
ans = total_full_houses / choose(total_cards, total_
    hands)

print(ans)
```

R code Exa 5.h.a Bridge player gets all spades

```
# Page No. 37

total_cards = 52
suite_size = 13

p_13spades = 1 / choose(total_cards, suite_size)
ans = 4 * p_13spades

print(ans)
```

R code Exa 5.h.b Bridge each player gets 1 spade

```r
# Page No. 37

total_cards = 52
suite_size = 13
no_of_aces = 4
ace_per_suite = 1

ans = (factorial(no_of_aces) * (factorial(total_
   cards - no_of_aces) / factorial(suite_size - ace_
   per_suite)^4)) / (factorial(total_cards) /
   factorial(suite_size)^4)

print(ans)
```

R code Exa 5.i Birthday problem

```r
# Page No. 38

days_in_a_year = 365
n = 0
p = 1
i = 0

while(p >= 0.5)
{
  p = p * (days_in_a_year - i) / days_in_a_year
  n = n + 1
  i = i + 1
}

print(n)
```

R code Exa 5.k Football team

```
# Page No. 39

P_2i <- function(i, o = 20, d = 20)
{
  k = (factorial(o - 2 * i) / (2^(o / 2 - i) *
     factorial(o / 2 - i)))^2
  numer = ((dim(combn(o, 2 * i))[2])^2) * factorial
     (2 * i) * k
  denom = (factorial(o + d) / (2^20 * factorial(o)))
  return(numer / denom)
}

for(i in 0:10)
{
  cat("P(", i,") =", P_2i(i), "\n")
}

# The answer may vary due to difference in
   representation.
```

R code Exa 5.1 Club sports

```
# Page No. 40

no_tennis = 36
no_squash = 28
no_badminton = 18
no_tennis_badminton = 12
no_tennis_squash = 22
no_badminton_squash = 9
no_all = 4

ans = no_tennis + no_squash + no_badminton - no_
   badminton_squash - no_tennis_badminton - no_
   tennis_squash + no_all
```

```

print(ans)
```

R code Exa 5.n Married couples

```
# Page No. 42

no_of_couples = 10
ans = 0

for(i in 1:10)
{
  ans = ans + ((-1)^(i + 1)) * choose(10, i) * (2^i)
      * factorial(19 - i) / factorial(19)
}
ans = 1 - ans

print(ans)
```

R code Exa 7.a Horse race

```
# Page No. 48

P = c(0.2, 0.2, 0.15, 0.15, 0.1, 0.1, 0.1)

v1 = P[1] + P[2] + P[3]

print(v1)

v2 = P[1] + P[5] + P[6] + P[7]

if(v1 > v2)
{
```

```
  print("First wager")
} else
{
  print("Second wager")
}
```

Chapter 3

Conditional Probability and Independence

R code Exa 2.a Student exams

```
# Page No. 59

P_Lx <- function(x)
{
  return(x/2)
}
P_F = 1 - P_Lx(1)
ans = P_F / (1 - P_Lx(0.75))

print(ans)
```

R code Exa 2.b.a First coin flip heads

```
# Page No. 59

library(MASS)
```

```
P_B = 1/4
P_F = 1/2

ans = P_B / P_F

print(fractions(ans))
```

R code Exa 2.b.b At least one flip heads

```
# Page No. 59

library(MASS)

P_B = 1/4
P_A = 3/4

ans = P_B / P_A

print(fractions(ans))
```

R code Exa 2.c Bridge

```
# Page No. 60

total_north_south = 26
suite_size = 13

ans = choose(5,3) * choose(total_north_south - 5, suite_size - 3) / choose(total_north_south, suite_size)

print(ans)
```

R code Exa 2.e Course selection

```
# Page No. 61

library(MASS)

P_C = 1/2
P_AgC = 2/3

ans = P_C * P_AgC

print(fractions(ans))
```

R code Exa 2.f.a Drawing red balls from urn

```
# Page No. 62

library(MASS)

no_of_red_balls = 8
no_of_blue_balls = 4

P_R1 = no_of_red_balls / (no_of_blue_balls + no_of_
    red_balls)
P_R2gR1 = (no_of_red_balls - 1 ) / (no_of_blue_balls
     + no_of_red_balls - 1)
ans = P_R1 * P_R2gR1

print(fractions(ans))
```

R code Exa 2.h Each pile contains 1 ace

```
# Page No. 64

P_E1 = 1
P_E2gE1 = 39/51
P_E3gE1E2 = 26/50
P_E4gE1E2E3 = 13/49

ans = P_E1 * P_E2gE1 * P_E3gE1E2 * P_E4gE1E2E3

print(ans)
```

R code Exa 3.d Laboratory blood test

```
# Page No. 67

P_D = 0.005
P_EgD = 0.95
P_EgDc = 0.01

ans = P_EgD * P_D / (P_EgD * P_D + P_EgDc * (1 - P_D
  ))

print(ans)
```

R code Exa 3.e Surgery dilemma

```
# Page No. 68

P_D = 0.6
P_EgD = 1
P_EgDc = 0.3
```

```

ans = P_D * P_EgD / (P_EgD * P_D + P_EgDc * (1 - P_D
    ))

print(ans)
```

R code Exa 3.f Criminal investigation

```
# Page No. 69

P_G = 0.6
P_CgG = 1
P_CgGc = 0.2

ans = P_G * P_CgG / (P_G * P_CgG + (1 - P_G) * P_
    CgGc)

print(ans)
```

R code Exa 3.i Urn and coins

```
# Page No. 72

library(MASS)

P_A = 2/3
P_headsgA = 1/4
P_B = 1/3
P_headsgB = 3/4

ans = P_A * P_headsgA / (P_B * P_headsgB)

print(fractions(ans))
```

R code Exa 3.l Coloured cards

```
# Page No. 75

library(MASS)

P_RgRB = 1/2
P_RB = 1/3
P_RgRR = 1
P_RR = 1/3
P_RgBB = 0
P_BB = 1/3

ans = P_RgRB * P_RB / (P_RgBB * P_BB + P_RgRB * P_RB
    + P_RgRR * P_RR)

print(fractions(ans))
```

R code Exa 3.n.a Flashlight gives more than 100 hours of use

```
# Page No. 77

P_F1 = 0.2
P_F2 = 0.3
P_F3 = 0.5
P_AgF1 = 0.7
P_AgF2 = 0.4
P_AgF3 = 0.3

ans = P_AgF1 * P_F1 + P_AgF2 * P_F2 + P_AgF3 * P_F3

print(ans)
```

R code Exa 3.n.b Type of flashlight

```
# Page No. 77

library(MASS)

P_FgA <- function(P_F, P_AgF)
{
  P_A = 0.41
  return(P_F * P_AgF / P_A)
}

P_AgF1 = 0.7
P_F1 = 0.2
P_AgF2 = 0.4
P_F2 = 0.3
P_AgF3 = 0.3
P_F3 = 0.5

cat("P(F1 | A) = ")
print(fractions(P_FgA(P_F1, P_AgF1)))
cat("P(F2 | A) = ")
print(fractions(P_FgA(P_F2, P_AgF2)))
cat("P(F3 | A) = ")
print(fractions(P_FgA(P_F3, P_AgF3)))
```

R code Exa 3.o Crime probability

```
# Page No. 77

ex_criminals = 10000
```

```
tot_pop = 1000000
P_hair_match = 10^-5

alpha = function(c)
{
  c / (ex_criminals * c + tot_pop - ex_criminals)
}
P_MgG = (1 - P_hair_match)^(ex_criminals - 1)
P_all_aj = function(c)
{
  (1 - ex_criminals * alpha(c)) / (1 - alpha(c))
}
P_MgGc = function(c)
{
  P_hair_match * P_all_aj(c) * (1 - P_hair_match)^(ex_criminals - 1)
}
P_G = alpha
P_GgM = function(c)
{
  (P_MgG * P_G(c)) / (P_MgG * P_G(c) + P_MgGc(c) * (1 - P_G(c)))
}

c1 = 100
ans1 = P_GgM(c1)

cat("For c =", c1, "alpha =", alpha(c1), "and P(G|M) =", P_GgM(c1), "\n")

c2 = 10
ans2 = P_GgM(c2)

cat("For c =", c2, "alpha =", alpha(c2), "and P(G|M) =", P_GgM(c2), "\n")

c3 = 1
ans3 = P_GgM(c3)
```

```
cat("For c =", c3, "alpha =", alpha(c3), "and P(G|M)
    =", P_GgM(c3), "\n")

# The answer may vary due to difference in
    representation.
```

R code Exa 4.h Independent die trials

```
# Page No. 83

library(MASS)

P_En <- function(n)
{
  ans = ((13/18)^(n - 1)) * 1/9
  return(ans)
}

ans = 0
i = 2

ix = P_En(1)

while(ix != 0)
{
  ans = ans + ix
  ix = P_En(i)
  i = i + 1
}

print(fractions(ans))
```

R code Exa 5.a Conditional probability of accident prone policy holder

```
# Page No. 94

library(MASS)

P_A1gA = 0.4
P_A = 0.3
P_A1 = 0.26

P_AgA1 = P_A1gA * P_A / P_A1

P_A2gAA1 = 0.4
P_A2gAcA1 = 0.2

P_A2gA1 = P_A2gAA1 * P_AgA1 + P_A2gAcA1 * (1 - P_
    AgA1)

print(P_A2gA1)
```

R code Exa 3.a.1 Accident within a year of purchasing policy

```
# Page No. 66

P_A1gA = 0.4
P_A = 0.3

P_A1gAc = 0.2
ans = P_A1gA * P_A + P_A1gAc * (1 - P_A)

print(ans)
```

R code Exa 3.a.2 Policy holder is accident prone

```
# Page No. 66

library(MASS)

P_A = 0.3
P_A1 = 0.26
P_A1gA = 0.4

ans = P_A*P_A1gA/P_A1

print(fractions(ans))
```

Chapter 4

Random Variables

R code Exa 1.b Urn with replacement

```
# Page No. 118

P <- function(i)
{
  x = choose(i - 1,2) / choose(20,3)
  return(x)
}

ans = 0

for(i in 17:20)
{
  ans = ans + P(i)
}

print(ans)
```

R code Exa 1.d Winnings from urn experiment

```
# Page No. 119

library(MASS)

P <- rep(4)

P[0] <- (choose(5,3) + choose(3,1) * choose(3,1) * choose(5,1)) / choose(11,3)
P[1] <- (choose(3,1) * choose(5,2) + choose(3,2) * choose(3,1)) / choose(11,3)
P[2] <- (choose(3,2) * choose(5,1)) / choose(11,3)
P[3] <- choose(3,3) / choose(11,3)

print(fractions(sum(P)))
```

R code Exa 10.a Properties of random variables

```
# Page No. 169

library(MASS)

sentinal = exp(-12)

F_x = function(x)
{
  if(x < 0)
  {
    return(0)
  }
  if(x < 1)
  {
    return(x/2)
  }
  if(x < 2)
  {
```

```
    return(2/3)
  }
  if(x < 3)
  {
    return(11/12)
  }
  else
  {
    return(1)
  }
}
P_X = function(F, b)
{
  return(F(b - sentinal))
}
ans1 = P_X(F_x, 3)

cat("a: ")
print(fractions(ans1))

ans2 = F_x(1) - P_X(F_x, 1)

cat("b: ")
print(fractions(ans2))

ans3 = 1 - F_x(1/2)

cat("c: ")
print(fractions(ans3))

ans4 = F_x(4) - F_x(2)

cat("d: ")
print(fractions(ans4))
```

R code Exa 3.a Expectation of a die roll

```
# Page No. 126

library(MASS)

X = 1:6
w = rep(1/6, 6)

ans = weighted.mean(X, w)

print(fractions(ans))
```

R code Exa 3.d Expectation of number of students

```
# Page No. 127

P_Xe36 = 36/120
P_Xe40 = 40/120
P_Xe44 = 44/120
w = c(P_Xe36, P_Xe40, P_Xe44)
X = c(36, 40, 44)

ans = weighted.mean(X, w)

print(ans)
```

R code Exa 4.a Expectation of a random variable squared

```
# Page No. 128

P_Ye1 = 0.3 + 0.2
P_Ye0 = 0.5
```

```
w = c(P_Ye0, P_Ye1)
X = 0:1

ans = weighted.mean(X, w)

print(ans)
```

R code Exa 5.a Variance of a die roll

```
# Page No. 133

library(MASS)

X = 1:6
w = rep(1/6, 6)

E_X = weighted.mean(X, w)
X1 = X^2
E_X1 = weighted.mean(X1, w)
Var_X = E_X1 - E_X^2

print(fractions(Var_X))
```

R code Exa 6.a Binomial random variable coin experiment

```
# Page No. 135

library("MASS")

n = 5
p = 1/2
x = 0:5

```

```r
ans = dbinom(x, size = n, prob = p)

for(i in x)
{
  cat("P{ X =", i, "} =")
  print(fractions(ans[i + 1]))
}

# The answer may vary due to difference in representation.
```

R code Exa 6.b Package replacement

```r
# Page No. 135

p = 0.01
n = 10

ans = 1 - pbinom(1, size = n, prob = p)

print(ans)
```

R code Exa 6.c Wheel of fortune

```r
# Page No. 136

library(MASS)

n = 3
p = 1/6
x = 0:3
X = c(-1, 1, 2, 3)

```

```
P_X = dbinom(x, size = n, prob = p)
ans = weighted.mean(X, P_X)

print(fractions(ans))
```

R code Exa 6.d Genetic traits

```
# Page No. 136

library(MASS)

n = 4
p = 3/4

ans = dbinom(3, size = n, prob = p)

print(fractions(ans))
```

R code Exa 6.h Computing binomial distribution function

```
# Page No. 142

n = 6
p = 0.4
x = 0:6

P_X = dbinom(x, size = n, prob = p)

for(i in 0:6)
{
  cat("P{ X =", i, "} =", P_X[i + 1], "\n")
}
```

R code Exa 6.i Binomial distribution generation

```
# Page No. 143

n = 100
p = 0.75

ans1 = dbinom(70, size = n, prob = p)

cat("P{X = 70} = ", ans1, "\n")

ans2 = pbinom(70, size = n, prob = p)

cat("P{X <= 70} = ", ans2, "\n")
```

R code Exa 7.a Typos in a page

```
# Page No. 144

l = 1/2

ans = 1 - dpois(0, lambda = l)

print(ans)
```

R code Exa 7.b Defective item

```
# Page No. 145

```

```
n = 10
p = 0.1
x = 1

binom_ans = pbinom(x, size = n, prob = p)

cat(binom_ans, "\n")

l = n * p
pois_ans = ppois(x, l)

cat(pois_ans)
```

R code Exa 7.c Radioactive particles

```
# Page No. 145

l = 3.2
q = 2

ans = ppois(q, lambda = l)

print(ans)
```

R code Exa 7.e.a Earthquake occourence

```
# Page No. 154

no_of_weeks = 2
x = 2

l = 2 * no_of_weeks
ans = 1 - ppois(x, lambda = l)
```

```
print(ans)

# The answer may vary due to difference in
    representation.
```

R code Exa 7.f Generating poisson distribution

```
# Page No. 155

l1 = 100
x1 = 90

ans1 = ppois(x1, l1)

cat("a) ", ans1, "\n")

l2 = 1000
x2 = 1075

ans2 = ppois(x2, l2)

cat("b) ", ans2)

# The answer may slightly vary due to rounding off
    values.
```

R code Exa 8.g Throws of die required

```
# Page No. 160

r = 4
p = 1/6
```

```
E_X = r / p
Var_X = r * (1 - p) / p^2

cat(E_X, "\n")
cat(Var_X, "\n")
```

R code Exa 8.i Electrical components

```
# Page No. 161

lot_size = 10

P_4defectives = 0.3
P_1defective = 0.7

inspect_size = 3

P_acceptance = choose(4,0) * choose(6,3) * P_4
    defectives / choose(lot_size, inspect_size) +
    choose(1,0) * choose(9,3) * P_1defective / choose
    (lot_size, inspect_size)

ans = 1 - P_acceptance

print(ans*100)
```

Chapter 5

Continuous Random Variables

R code Exa 1.a.a Finding the constant in "fx"

```
# Page No. 187

library(MASS)

integrand = function(x)
{
  4 * x - 2 * x^2
}
C = 1 / integrate(integrand, lower = 0, upper = 2)$
   value

print(fractions(C))
```

R code Exa 1.a.b Probability of a continuous random variable

```
# Page No. 187

library("MASS")
```

```
integrand = function(x)
{
  C = 3/8
  C * (4 * x - 2 * x^2)
}
ans = integrate(integrand, lower = 1, upper = 2)$
   value

print(fractions(ans))
```

R code Exa 1.b.a Computer functioning "a"

```
# Page No. 188

integrand = function(x)
{
  exp(-x/100)
}
l = 1 / integrate(integrand, lower = 0, upper = Inf)
   $value
ans = l * integrate(integrand, lower = 50, upper =
   150)$value

print(ans)
```

R code Exa 1.b.b Computer functioning "b"

```
# Page No. 188

l = 0.01

integrand = function(x)
```

```
{
  exp(-x/100)
}
ans = 1 * integrate(integrand, lower = 0, upper =
    100)$value

print(ans)
```

R code Exa 1.c Lifetime of radio tube

```
# Page No. 188

library(MASS)

n = 5
x = 2

integrand = function(x)
{
  100 / x^2
}
P_Ei = integrate(integrand, lower = 100, upper =
    150)$value
ans = dbinom(x, size = n, prob = P_Ei)

print(fractions(ans))
```

R code Exa 2.a Expectation of a continuous random variable

```
# Page No. 190

library("MASS")

```

```
integrand = function(x)
{
  2 * x^2
}
ans = integrate(integrand, lower = 0, upper = 1)$value

print(fractions(ans))
```

R code Exa 2.e Variance of a continuous random variable

```
# Page No. 194

library("MASS")

E_X = 2/3

integrand = function(x)
{
  2 * x^3
}
E_X2 = integrate(integrand, lower = 0, upper = 1)$value
ans = E_X2 - E_X^2

print(fractions(ans))
```

R code Exa 3.b Uniform distribution

```
# Page No. 196

library("MASS")

```

```
min = 0
max = 10

ans1 = punif(3, min = min, max = max)

cat("Ans to a)")
print(fractions(ans1))

ans2 = 1 - punif(6, min = min, max = max)

cat("Ans to b)")
print(fractions(ans2))

ans3 = punif(8, min = min, max = max) - punif(3, min
    = min, max = max)

cat("Ans to c)")
print(fractions(ans3))

# The answer may vary due to difference in
    representation.
```

R code Exa 3.c Waiting time for bus

```
# Page No. 196

library("MASS")

uni = function(lower, upper)
{
  alpha = 0
  beta = 30

  P_X = punif(upper, min = alpha, max = beta) -
     punif(lower, min = alpha, max = beta)
```

```
  return(P_X)
}
ans1 = uni(10, 15) + uni(25, 30)

cat("Ans for a)")
print(fractions(ans1))

ans2 = uni(0, 5) + uni(15, 20)

cat("Ans for b)")
print(fractions(ans2))
```

R code Exa 4.b Normal random variable

```
# Page No. 202

mu = 3
sigma = sqrt(9)

ans1 = pnorm(5, mu, sigma) - pnorm(2, mu, sigma)

cat("Ans to a) ", ans1, "\n")

ans2 = 1 - pnorm(0, mu, sigma)

cat("Ans to b) ", ans2, "\n")

ans3 = 1 - pnorm(9, mu, sigma) + pnorm(-3, mu, sigma
   )

cat("Ans to c) ", ans3, "\n")
```

R code Exa 4.c Grading on the curve

```
# Page No. 202

cat("A:", (1 - pnorm(1)) * 100, "%\n")
cat("B:", (pnorm(1) - pnorm(0)) * 100, "%\n")
cat("C:", (pnorm(0) - pnorm(-1)) * 100, "%\n")
cat("D:", (pnorm(2) - pnorm(1)) * 100, "%\n")
cat("E:", (pnorm(-2)) * 100, "%\n")
```

R code Exa 4.d Paternity suit

```
# Page No. 203

mu = 270
sigma = sqrt(100)

ans =  1 - pnorm(290, mu, sigma) + pnorm(240, mu, sigma)

print(ans)
```

R code Exa 4.e Signals

```
# Page No. 203

P_EgOne = 1 - pnorm(1.5)

cat("P{1} =", P_EgOne, "\n")

P_EgZero = 1 - pnorm(2.5)

cat("P{0} =", P_EgZero, "\n")
```

R code Exa 4.f Normal approximation

```
# Page No. 204

n = 40
p = 1/2
x = 20

mu = n * p
sigma = sqrt(n * p * (1 - p))
ans1 = pnorm(x + 0.5, mean = mu, sd = sigma) - pnorm
    (x - 0.5, mean = mu, sd = sigma)

cat("Ans via approx:", ans1, "\n")

ans2 = dbinom(x, size = n, prob = p)

cat("Ans:", ans2)

# The answer may slightly vary due to rounding off
    values.
```

R code Exa 4.g Probability of attendance in a college

```
# Page No. 205

n = 450
p = 0.3

mu = n * p
sigma = sqrt(n * p * (1 - p))
ans = 1 - pnorm(150.5, mu, sigma)
```

```

print(ans)
```

R code Exa 4.h Effectiveness of diet

```
# Page No. 206

n = 100
p = 1/2

mu = n * p
sigma = sqrt(n * p * (1 - p))
ans = 1 - pnorm(64.5, mu, sigma)

print(ans)
```

R code Exa 4.i Outlawing cigaretts

```
# Page No. 206

P_Sn = function(n)
{
  p = 0.52
  mu = p * n
  sigma = sqrt(n * p * (1 - p))
  ans = 1 - pnorm(0.5 * n, mu, sigma)
  return(ans)
}

N = c(11, 101, 1001)

for(i in N)
{
```

```
  cat("For n =", i, "P(Sn > 0.5 * n) =", P_Sn(i), "\
    n")
}

i = 1

while(P_Sn(i) < 0.95)
{
  i = i + 1
}

cat("For at least 95%, n =", i, "\n")
```

R code Exa 5.b Exponential random variable

```
# Page No. 209

lambda = 1/10

ans1 = 1 - pexp(10, lambda)

cat("Ans to a)", ans1, "\n")

ans2 = pexp(20, lambda) - pexp(10, lambda)

cat("Ans to b)", ans2, "\n")
```

R code Exa 5.e Laplace distribution

```
# Page No. 212

library(rmutil)

```

```
P_EgOne = plaplace(-1.5)
P_EgZero = plaplace(-2.5)

cat("P{1} =", P_EgOne, "\n")
cat("P{0} =", P_EgZero, "\n")
```

Chapter 6

Jointly Distributed Random Variables

R code Exa 1.a Joint PMF of drawing balls from urn

```
# Page No. 233

library(MASS)

P_XY = function(x, y)
{
  blue = 5
  red = 3
  white = 4
  lot_size = 3

  ans = choose(blue, lot_size - x - y) * choose(red,
      x) * choose(white, y) / choose(blue + white +
     red, lot_size)
}

lot_size = 3

for(i in 0:lot_size)
```

```
{
  for(j in 0:(lot_size - i))
  {
    cat("p(", i,",",j,") =")
    print(fractions(P_XY(i, j)))
  }
}

# The answer may vary due to difference in
   representation.
```

R code Exa 1.b Joint PMF of number of boys and girls in a family

```
# Page No. 234

P = c(.15, .2, .35, .3)

P_BG <- function(i, j)
{
  if((i + j) >= 4)
  {
    return(0)
  }else
  {
    ans = P[(i + j) + 1] * ((factorial(i + j) / (
       factorial(i) * factorial(j))) / 2^(i + j))
    return(ans)
  }
}

max_children = 3

for(i in 0:max_children)
{
  for(j in 0:max_children)
```

```
  {
    cat("p(", i,",",j,") =", P_BG(i, j), "\n")
  }
}
```

R code Exa 1.c.a Joint density function "a"

```
# Page No. 236

myfun = function(x, y) (2 * exp(-x) * exp(-2 * y))
llimx =  1
ulimx = Inf
llimy = 0
ulimy = 1

f = function()
{
  return(integrate(function(y)
  {
    sapply(y, function(y)
    {
      integrate(function(x) myfun(x,y), llimx, ulimx
         )$value
    })
  }, llimy, ulimy))
}
ans = f()$value

print(ans)

# The answer may vary due to difference in
   representation.
```

R code Exa 1.c.b Joint density function "b"

```
# Page No. 236

library(MASS)

myfun <- function(x,y) (2 * exp(-x) * exp(-2 * y))
llimx <- 0
llimy <- 0
ulimy <- Inf

f <- function()
{
  return(integrate(function(y)
  {
    sapply(y, function(y)
     {
      integrate(function(x) myfun(x,y), llimx, y)$value
    })
  }, llimy, ulimy))
}
ans = f()$value

print(fractions(ans))
```

R code Exa 2.c Waiting time

```
# Page No. 243

library(pracma)
library(MASS)

integrand = function(x, y)
{
```

```
  (1/60)^2
}
xm = function(y)
{
  (y - 10)
}
ans = 2 * integral2(integrand, xmin = 10, xmax =
   60, ymin =  0, ymax = xm)$Q

print(fractions(ans))
```

R code Exa 3.c.a Basketball wins "a"

```
# Page No. 257

na = 26
pa = 0.4

E_Xa = na * pa
Var_Xa = na * pa * (1 - pa)

nb = 18
pb = 0.7

E_Xb = nb * pb
Var_Xb = nb * pb * (1 - pb)
E_Xab = E_Xa + E_Xb
Var_Xab = Var_Xa + Var_Xb
ans = 1 - pnorm(25 - 1/2, E_Xab, sqrt(Var_Xab))

print(ans)
```

R code Exa 3.c.b Basketball wins "b"

```
# Page No. 257

na = 26
pa = 0.4

E_Xa = na * pa
Var_Xa = na * pa * (1 - pa)

nb = 18
pb = 0.7

E_Xb = nb * pb
Var_Xb = nb * pb * (1 - pb)
E_Xab = E_Xa - E_Xb
Var_Xab = Var_Xa + Var_Xb
ans = 1 - pnorm(1 - 1/2, E_Xab, sqrt(Var_Xab))

print(ans)
```

R code Exa 3.d.a Price of security "a"

```
# Page No. 258

mu = 0.0165
sigma = 0.0730

p = 1 - pnorm(0, mu, sigma)
ans = p^2

print(ans)
```

R code Exa 3.d.b Price of security "b"

```
# Page No. 258

mu = 0.0165
sigma = 0.0730

mu = mu * 2
sigma = sqrt(2 * sigma^2)
ans = 1 - pnorm(0, mu, sigma)

print(ans)
```

R code Exa 4.a Conditional probability on joint PMF

```
# Page No. 264

library(MASS)

p00 = 0.4
p01 = 0.2
p10 = 0.1
p11 = 0.3

py1 = p01 + p11
px0 = p01 / py1
px1 = p11 / py1

print(fractions(px0))
print(fractions(px1))
```

R code Exa 6.b Probability of sample median

```
# Page No. 272

```

```
library(MASS)

integrand = function(x)
{
  x * (1 - x)
}
ans = factorial(3) / factorial(1)^2 * integrate(
   integrand, lower = 1/4, upper = 3/4)$value

print(fractions(ans))
```

Chapter 7

Properties of Expectation

R code Exa 2.r Chipmunks and groves

```
# Page No. 312

no_of_chipmunks = 15
total_trees = 52
p = 7/52

Xi = rep(1, no_of_chipmunks)
E_Xi = Xi * p
E_X = sum(E_Xi)

print(E_X)

# The answer may vary due to difference in
   representation.
```

R code Exa 5.e.a Game of craps "a"

```
# Page No. 335
```

```

Pi = function(i)
{
  if(i > 7)
  {
    i = 14 - i
  }
  ans = (i - 1) / 36
}
E_RgSi = function(i)
{
  if(i == 2 || i == 3 || i == 7 || i == 11 || i ==
     12)
  {
    return(1)
  }
  else
  {
    ans = 1 + 1 / (Pi(i) + Pi(7))
    return(ans)
  }
}

E_R = 0

for(i in 2:12)
{
  E_R = E_R + E_RgSi(i) * Pi(i)
}

print(E_R)
```

R code Exa 5.e.b Game of craps ”b”

```
# Page No. 335
```

```
Pi = function(i)
{
  if(i > 7)
  {
    i = 14 - i
  }
  ans = (i - 1) / 36
}
P_Si = function(c)
{
  if(i == 7 || i == 11)
  {
    return(Pi(i))
  }
  if( i == 2 || i == 3 || i == 12)
  {
    return(0)
  }
  else
  {
    ans = Pi(i)^2 / (Pi(i) + Pi(7))
  }
}

p = 0

for(i in 2:12)
{
  p = p + P_Si(i)
}

print(p)
```

R code Exa 5.e.c Game of craps ”c”

```
# Page No. 335

p = 0.493
E_R = 3.376

E_RgSi = function(i)
{
  if(i == 2 || i == 3 || i == 7 || i == 11 || i ==
     12)
  {
    return(1)
  }
  else
  {
    ans = 1 + 1 / (Pi(i) + Pi(7))
    return(ans)
  }
}
Pi = function(i)
{
  if(i > 7)
  {
    i = 14 - i
  }
  ans = (i - 1) / 36
}
Qi = function(i)
{
  if( i == 2 || i == 3 || i == 12)
  {
    return(0)
  }
  if( i == 7 || i == 11)
  {
    return(Pi(i) / p)
  }
  else
  {
```

```
    ans = Pi(i)^2 / (p * (Pi(i) + Pi(7)))
    return(ans)
  }
}

E_Rgwin = 0

for(i in 2:12)
{
  E_Rgwin = E_Rgwin + E_RgSi(i) * Qi(i)
}

E_Rglose = (E_R - E_Rgwin * p) / (1 - p)

print(E_Rglose)
```

Chapter 8

Limit Theorems

R code Exa 2.b Chebyshev inequality

```r
# Page No. 390

alpha = 0
beta = 10

E_X = (alpha + beta) / 2
Var_X = (beta - alpha)^2 / 12

ans1 = Var_X / 4^2

cat("Using inequality, the ans is ", ans1, "\n")

ans2 = punif(1, min = alpha, max = beta) + 1 - punif(9, min = 0, max = 10)

cat("Exact ans is ", ans2, "\n")
```

R code Exa 3.a Astronomical distances

```
# Page No. 393

P = function(n)
{
  acc = 0.5
  ans = pnorm(0.5 * sqrt(n) / 2, 0, 1) - pnorm(-0.5
     * sqrt(n) / 2, 0, 1)
  return(ans)
}

i = 1

while(P(i) < 0.95)
{
  i = i + 1
}

print(i)
```

R code Exa 3.b Probability of a professor teaching two sections

```
# Page No. 396

mu = 100
var = mu

ans = 1 - pnorm(120 - 1/2, mu, sqrt(var))

print(ans)
```

R code Exa 3.c Sum of dice rolls

```
# Page No. 397
```

```

p = 1/6
E_X = 0
E_X2 = 0
no_of_die_rolls = 10

for(i in 1:6)
{
  E_X = E_X + p * i
  E_X2 = E_X2 + p * i^2
}

print(E_X)

Var_X = (E_X2 - E_X^2)

print(Var_X)

Var_X = (E_X2 - E_X^2) * no_of_die_rolls
E_X = E_X *  no_of_die_rolls
ans = pnorm(40 + 1/2, E_X, sqrt(Var_X)) - pnorm(30 -
    1/2, E_X, sqrt(Var_X))

print(ans)

# The answer may vary due to difference in
   representation.
```

R code Exa 3.d Approximation of uniform random variables

```
# Page No. 398

E_X = 1/2
Var_X = 1/12

```

```
n = 10

E_X = 1/2*10
Var_X = 1/12*10

ans = 1 - pnorm(6, E_X, sqrt(Var_X))

print(ans)
```

R code Exa 3.e Time for grading exams

```
# Page No. 398

mu = 20
sd = 4
n = 25

mu = mu * n
sd = sd * sqrt(n)
ans = pnorm(450, mu, sd)

print(ans)
```

R code Exa 5.a Factory production

```
# Page No. 405

library(MASS)

mu = 100
var = 400
a = 20

```

```
ans = var / (var + a^2)

print(fractions(ans))
```

R code Exa 5.b Pairs

```
# Page No. 405

tot_men = 100
tot_people = 200

E_Xi = tot_men / (tot_people - 1)
E_X = E_Xi * tot_men
Var_X = E_X * (tot_men - 1) / (tot_people - 1) + 2 *
    choose(tot_men, 2) * ((tot_men*(tot_men - 1)) /
   ((tot_people - 1) * (tot_people - 3)) - (tot_men
   / (tot_people - 1))^2)
a = E_X - 30
ans = Var_X / (Var_X + a^2)

print(ans)
```