R Textbook Companion for Probability And Statistics by Morris H. Degroot, Mark J. Schervish[1]

Created by
E B Benson
Integrated MSc
Computer Science
Cochin University of Science and Technology
Cross-Checked by
R TBC Team

August 29, 2024

[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: Probability And Statistics

Author: Morris H. Degroot, Mark J. Schervish

Publisher: Pearson

Edition: 4

Year: 2012

ISBN: 978-0-321-50046-5

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

Introduction to probability

R code Exa 1.4.1 Rolling a Die

```r
# Page 006

S <- c(1, 2, 3, 4, 5, 6)

Event_A <- S[S %% 2 == 0]
print(Event_A)

Event_B <- S[S > 2]
print(Event_B)
```

R code Exa 1.4.2 Subset

```r
# Page 007

S <- c(1,2,3,4,5,6)

Event_A <- S[S %% 2 == 0]
print(Event_A)
```

```r

Event_C <- S[S > 1]
print(Event_C)

if (all(Event_A %in% Event_C)){
  print("A is a subset of C")
}
```

R code Exa 1.4.3 Complement

```r
# Page 008

S <- c(1,2,3,4,5,6)

Event_A <- S[S %% 2 == 0]
print(Event_A)

Event_Ac <- setdiff(S, Event_A)
print(Event_Ac)
```

R code Exa 1.5.2 A loaded die

```r
# Page 017
library(MASS)

S <- 1:6

probabilities <- c(1/7, 1/7, 1/7, 1/7, 1/7, 2/7)

Event_A <- S[S%%2 != 0]
print(Event_A)

```

```
Probability_of_A <- fractions(sum(probabilities[
    Event_A]))

print(Probability_of_A )
```

R code Exa 1.5.3 Diagnosing diseases

```
# Page 19

Probability_of_bacterial_infection <- 0.7

Probability_of_viral_infection <- 0.4

Probability_of_Bacterial_Union_Viral <- 1

Probability_of_patient_having_both_infection <-
    Probability_of_bacterial_infection + Probability_
    of_viral_infection - Probability_of_Bacterial_
    Union_Viral
print(Probability_of_patient_having_both_infection)
```

R code Exa 1.5.4 Demands for utilities

```
# Page 019

electric_demand <- 1:150
water_demand <- 4:200

sample_space <- expand.grid(water_demand = water_
    demand, electric_demand = electric_demand)

Area_of_sample_space <- (max(sample_space$water_
    demand) - min(sample_space$water_demand)) * (max(
```

```
sample_space$electric_demand) - min(sample_space$electric_demand))

event_A <- sample_space[sample_space$water_demand >= 100, ]

event_B <- sample_space[sample_space$electric_demand >= 115, ]

Area_of_A <- (max(event_A$water_demand) - min(event_A$water_demand)) * (max(event_A$electric_demand) - min(event_A$electric_demand))

Area_of_B <- (max(event_B$water_demand) - min(event_B$water_demand)) * (max(event_B$electric_demand) - min(event_B$electric_demand))

Probability_of_A <- Area_of_A / Area_of_sample_space

Probability_of_B <- Area_of_B / Area_of_sample_space

Area_of_A_intersection_B <- (max(event_A$water_demand) - min(event_A$water_demand)) * (max(event_B$electric_demand) - min(event_B$electric_demand))
Probability_of_A_intersection_B <- Area_of_A_intersection_B/Area_of_sample_space

Probability_of_A_union_B <- Probability_of_A + Probability_of_B - Probability_of_A_intersection_B
print(round(Probability_of_A_union_B,4))
```

R code Exa 1.6.2 Fiber breaks

```
#page 22

library(MASS)

i <- 1:5

Alpha <- 1/15

Probability_of_fibre_length_not_more_than_3_breaking
    <- fractions(sum(Alpha*i[i<=3]))

print(Probability_of_fibre_length_not_more_than_3_
    breaking)
```

R code Exa 1.6.3 Tossing coins

```
#Page 23

library(MASS)

Probability_of_tossing_three_identical_coins <- 1/8

Probability_of_two_heads <- fractions(choose(3,2) *
    Probability_of_tossing_three_identical_coins)

print(Probability_of_two_heads)
```

R code Exa 1.6.4 Genetics

```
#Page 23

library(MASS)
genotypes <- c("AA", "Aa", "aA", "aa")
```

```
probabilities <- rep(1/4, 4)

genotype_probabilities <- c(
  "AA" = sum(probabilities[genotypes %in% "AA"]),
  "Aa" = sum(probabilities[genotypes %in% c("Aa", "aA")]),
  "aa" = sum(probabilities[genotypes %in% "aa"])
)

print(paste("Probability of genotype AA:", fractions(genotype_probabilities["AA"])))
print(paste("Probability of genotype Aa:", fractions(genotype_probabilities["Aa"])))
print(paste("Probability of genotype aa:", fractions(genotype_probabilities["aa"])))
```

R code Exa 1.6.5 Rolling two dice

```
#Page 24
library(MASS)

sample_space <- expand.grid(1:6, 1:6)

sums <- rowSums(sample_space)

sum_probabilities <- table(sums) / length(sums)

sum_probabilities <- fractions(sum_probabilities)

for (i in 2:12) {
  print(paste("P(Sum =", i,")=", sum_probabilities[as.character(i)]))
}
```

R code Exa 1.7.1 Routes between cities

```r
#Page 26

Number_of_cities <- 3

Number_of_routes_from_A_to_B <- 3

Number_of_routes_from_B_to_C <- 5

Total_number_of_routes <- Number_of_routes_from_A_to_B*Number_of_routes_from_B_to_C

print(Total_number_of_routes)
```

R code Exa 1.7.3 Possible outcomes of two dice

```r
#Page 26

Number_of_possible_outcomes_for_a_die <- 6

Number_of_possible_outcomes_for_two_dice <- Number_of_possible_outcomes_for_a_die*Number_of_possible_outcomes_for_a_die

print(Number_of_possible_outcomes_for_two_dice)
```

R code Exa 1.7.4 Tossing several coins

```
#Page 27

library(MASS)

Number_of_possible_outcomes_for_six_coins <- 2^6

Number_of_outcomes_with_six_heads <- 1

Probability_of_all_heads <- fractions(Number_of_outcomes_with_six_heads/Number_of_possible_outcomes_for_six_coins)
print(Probability_of_all_heads)

Number_of_outcomes_with_one_head <- 6

Probability_of_one_head <- fractions(Number_of_outcomes_with_one_head/Number_of_possible_outcomes_for_six_coins)
print(Probability_of_one_head)
```

R code Exa 1.7.5 Combination lock

```
#Page 27

Number_of_elements <- 40

Length_of_sequence <- 3

Possible_combinations <- Number_of_elements^Length_of_sequence

print(Possible_combinations)
```

R code Exa 1.7.7 Current population survey

```
#Page 28

n <- 50000

k <- 3

Number_of_outcomes <- 50000 * 49999 * 49998

print(format(Number_of_outcomes, scientific = TRUE,
   digits = 3))
#The answer may vary due to difference in
   representation
```

R code Exa 1.7.8 Choosing officers

```
#Page 29

Number_of_members <- 25

Number_of_positions <- 2

Possible_number_of_choices <- choose(Number_of_
   members, Number_of_positions)*factorial(Number_of
   _positions)

print(Possible_number_of_choices)
```

R code Exa 1.7.9 Arranging books

```
#Page 29

```

```
Number_of_books <- 6

Possible_permutations <- factorial(Number_of_books)

print(Possible_permutations)
```

R code Exa 1.7.12 Approximating the number of permutations

```
#Page 31

stirling_approx <- function(n) {
  sqrt(2 * pi * n) * (n / exp(1))^n
}

Permutation_70_20 <- factorial(70)/factorial(50)

print(format(Permutation_70_20, scientific = TRUE,
    digits = 4))

Stirling_approximation <- stirling_approx(70)/
    stirling_approx(50)

print(format(Stirling_approximation, scientific =
    TRUE, digits = 4))

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

R code Exa 1.8.2 Selecting a committee

```
#Page 33

People_for_committe <- 8
```

```

People_in_group <- 20

Possible_number_of_groups <- choose(People_in_group,People_for_committe)

print(Possible_number_of_groups)
```

R code Exa 1.8.3 Choosing jobs

```
#Page 33

People_for_committe <- 8

People_in_group <- 20

Number_of_jobs <- 8

Possible_number_of_assignments <- choose(People_in_group,People_for_committe) * factorial(8)

print(Possible_number_of_assignments)
#The answer provided in the textbook is wrong.
```

R code Exa 1.8.4 Blood types

```
#Page 34

Number_of_alleles <- 3

Pairs_with_same_allele <- Number_of_alleles
Pairs_with_different_allele <- choose(Number_of_alleles,2)
```

```

Total_number_of_genotypes <- Pairs_with_same_allele+Pairs_with_different_allele
print(Total_number_of_genotypes)
```

R code Exa 1.8.5 Selecting baked goods

```
#Page 35

Number_of_items_to_choose <- 12
Number_of_items_in_bakery <- 7

Different_boxfuls <- choose(Number_of_items_in_bakery+Number_of_items_to_choose-1, Number_of_items_to_choose)
print(Different_boxfuls)
```

R code Exa 1.8.6 Selecting baked goods from boxes

```
#Page 35

Number_of_items_to_choose <- 12
Number_of_items_in_bakery <- 7

Possible_outcomes_when_choosing_random <- Number_of_items_in_bakery^Number_of_items_to_choose
signif(Possible_outcomes_when_choosing_random,3)

Probability_that_all_items_are_same_choosing_random <- Number_of_items_in_bakery/Possible_outcomes_when_choosing_random
signif(Probability_that_all_items_are_same_choosing_random,3)
```

```
Number_of_packed_boxes_with_baker <- 18564
Probability_that_all_items_are_same_choosing_boxes
    <- Number_of_items_in_bakery/Number_of_packed_
    boxes_with_baker
signif(Probability_that_all_items_are_same_choosing_
    boxes,3)
#The answer may vary due to difference in
    representation
```

R code Exa 1.8.7 Tossing a coin

```
#Page 36

Number_of_tosses <- 10

Total_possible_number_of_sequences <- 2^Number_of_
    tosses

Arrangements_with_3_heads <- choose(10,3)

Probability_of_obtaining_exactly_3_heads <-
    Arrangements_with_3_heads/Total_possible_number_
    of_sequences
print(round(Probability_of_obtaining_exactly_3_heads
    ,4))

Probability_of_obtaining_atmost_3_heads <- sum(
    choose(10,0:3))/Total_possible_number_of_
    sequences
print(round(Probability_of_obtaining_atmost_3_heads
    ,4))
```

R code Exa 1.8.8 Sampling without replacement

```
#Page 37

Number_of_boys <- 15
Number_of_girls <- 30
Number_of_students_to_select <- 10

Possible_number_of_combinations_of_students <-
    choose(Number_of_boys+Number_of_girls,Number_of_
    students_to_select)

Possible_combinations_of_3_boys_and_7_girls <-
    choose(Number_of_boys,3)*choose(Number_of_girls
    ,7)

Desired_probability <- Possible_combinations_of_3_
    boys_and_7_girls/Possible_number_of_combinations_
    of_students
signif(Desired_probability,4)
```

R code Exa 1.8.9 Playing cards

```
#Page 37

Number_of_cards <- 52
Number_of_players <- 4
Number_of_aces <- 4

Total_possible_combinations <- choose(Number_of_
    cards,Number_of_players)
Possible_combinations_for_each_player_receiving_one_
    ace <- (Number_of_cards/Number_of_players)^Number
    _of_aces
Probability_that_each_player_will_receive_one_ace <-
```

```
 Possible_combinations_for_each_player_receiving_one_ace/Total_possible_combinations

print(round(Probability_that_each_player_will_receive_one_ace,4))
```

R code Exa 1.8.10 Playing cards revisited

```
#Page 38

Number_of_cards <- 52
Number_of_players <- 4
Number_of_aces <- 4

Total_possible_outcomes <- factorial(Number_of_cards)
Possible_combinations_for_each_player_receiving_one_ace <- (Number_of_cards/Number_of_players)^Number_of_aces
Number_of_ways_to_arrange_aces <- factorial(Number_of_aces)
Number_of_ways_to_arrange_rest_of_cards <- factorial(Number_of_cards-Number_of_aces)
Possible_outcomes_of_interest <- Possible_combinations_for_each_player_receiving_one_ace*Number_of_ways_to_arrange_aces*Number_of_ways_to_arrange_rest_of_cards
Probability_that_each_player_will_receive_one_ace <- Possible_outcomes_of_interest/Total_possible_outcomes

signif(Probability_that_each_player_will_receive_one_ace,4)
```

R code Exa 1.8.11 Lottery ticket

```
#Page 38

Numbers_present <- 30
Numbers_choosen <- 6

Possible_number_of_outcomes <- choose(Numbers_
    present,Numbers_choosen)*factorial(Numbers_
    choosen)
print(Possible_number_of_outcomes)

Possible_combinations_of_six_numbers <- choose(
    Numbers_present, Numbers_choosen)
print(Possible_combinations_of_six_numbers)
```

R code Exa 1.9.1 Choosing committees

```
#Page 42

Number_of_members <- 20
Number_of_committees <- 3
Members_needed_in_committee_A <- 8
Members_needed_in_committee_B <- 8
Members_needed_in_committee_C <- 4

Combinations_possible_for_committee_A <- choose(
    Number_of_members,Members_needed_in_committee_A)
Combinations_for_rest_members_to_B_and_C <- choose(
    Number_of_members-Members_needed_in_committee_A,
    Members_needed_in_committee_B)
```

```
Possible_ways_of_choosing_committee <- Combinations_possible_for_committee_A*Combinations_for_rest_members_to_B_and_C

print(Possible_ways_of_choosing_committee)
```

R code Exa 1.9.2 Multinomial coefficient

```
#Page 43

library(iterpc)

n <- 20
k <- 3
n1 <- 8
n2 <- 8
n3 <- 4

Possible_combinations <- multichoose(c(n1,n2,n3))
print(Possible_combinations)
```

R code Exa 1.9.3 Rolling dice

```
#Page 44

Number_of_dice <- 12
k <- 6

Possible_outcomes <- factorial(Number_of_dice)/(factorial(Number_of_dice/k)^k)
Probability_each_number_appear_twice <- Possible_outcomes/k^Number_of_dice

```

```
print(round(Probability_each_number_appear_twice ,4))
```

R code Exa 1.9.4 Distributing heart cards

```
#Page 44

library(combinat)
library(gmp)
library(iterpc)

Number_of_cards <- 52
Number_of_hearts <- 13
Number_of_players <- 4
Number_of_cards_received_by_each_player <- 13
Hearts_received_by_player_A <- 6
Hearts_received_by_player_B <- 4
Hearts_received_by_player_C <- 2
Hearts_received_by_player_D <- 1

Ways_to_distribute_cards <- factorial(Number_of_cards) / (factorial(Number_of_cards_received_by_each_player)^Number_of_players)

Distribution_of_hearts_cards <- multichoose(c(Hearts_received_by_player_A, Hearts_received_by_player_B, Hearts_received_by_player_C, Hearts_received_by_player_D))
signif(Distribution_of_hearts_cards)

Number_of_remaining_cards <- Number_of_cards - Number_of_hearts
Distribution_of_other_cards <- choose(Number_of_remaining_cards, Number_of_cards_received_by_each_player - Hearts_received_by_player_A) *
  choose(Number_of_remaining_cards - (Number_of_
```

```
cards_received_by_each_player - Hearts_received_by_player_A), Number_of_cards_received_by_each_player - Hearts_received_by_player_B) *
  choose(Number_of_remaining_cards - (Number_of_cards_received_by_each_player - Hearts_received_by_player_A) - (Number_of_cards_received_by_each_player - Hearts_received_by_player_B), Number_of_cards_received_by_each_player - Hearts_received_by_player_C) *
  choose(Number_of_remaining_cards - (Number_of_cards_received_by_each_player - Hearts_received_by_player_A) - (Number_of_cards_received_by_each_player - Hearts_received_by_player_B) - (Number_of_cards_received_by_each_player - Hearts_received_by_player_C), Number_of_cards_received_by_each_player - Hearts_received_by_player_D)

Distribution_of_cards_based_on_assigned_heart_cards <- Distribution_of_hearts_cards * Distribution_of_other_cards

Probability_of_assigning <- Distribution_of_cards_based_on_assigned_heart_cards / Ways_to_distribute_cards
signif(Probability_of_assigning,3)

Possible_combinations <- prod(choose(Number_of_hearts,c(Hearts_received_by_player_A,Hearts_received_by_player_B,Hearts_received_by_player_C,Hearts_received_by_player_D)))/choose(Number_of_cards,Number_of_cards_received_by_each_player)
signif(Possible_combinations,3)
```

R code Exa 1.10.1 Student enrollment

```
#Page 47

library(MASS)

Number_of_students <- 200

Maths_students <- 137
History_students <- 50
Music_students <- 124

Students_in_both_maths_and_history <- 33
Students_in_both_music_and_history <- 29
Students_in_both_maths_and_music <- 92
Students_in_all_three <- 18

Probability_of_student_is_in_maths <- Maths_students/Number_of_students
fractions(Probability_of_student_is_in_maths)

Probability_of_student_is_in_history <- History_students/Number_of_students
fractions(Probability_of_student_is_in_history)

Probability_of_student_is_in_music <- Music_students/Number_of_students
fractions(Probability_of_student_is_in_music)


Probability_of_student_is_in_maths_and_history <- Students_in_both_maths_and_history/Number_of_students
fractions(Probability_of_student_is_in_maths_and_history)

Probability_of_student_is_in_music_and_history <- Students_in_both_music_and_history/Number_of_students
fractions(Probability_of_student_is_in_music_and_
```

```
history)

Probability_of_student_is_in_maths_and_music <- Students_in_both_maths_and_music/Number_of_students
fractions(Probability_of_student_is_in_maths_and_music)

Probability_of_student_is_in_all_three <- Students_in_all_three/Number_of_students
fractions(Probability_of_student_is_in_all_three)

Probability_of_student_enrolled_in_at_least_one_of_the_three_classes <- Probability_of_student_is_in_all_three + (Probability_of_student_is_in_maths + Probability_of_student_is_in_history + Probability_of_student_is_in_music) - (Probability_of_student_is_in_maths_and_history+Probability_of_student_is_in_maths_and_music+Probability_of_student_is_in_music_and_history)
fractions(Probability_of_student_enrolled_in_at_least_one_of_the_three_classes)
```

Chapter 2

Conditional Probability

R code Exa 2.1.2 Lottery ticket

```
#Page 56

Numbers_present <- 30
Numbers_taken <- 6
Possible_combinations <- choose(Numbers_present,
   Numbers_taken)

Probability_of_choosing_15_in_winning_combination <-
    choose(Numbers_present-1,Numbers_taken-1)/
   Possible_combinations
print(Probability_of_choosing_15_in_winning_
   combination)

Probability_of_winning <- 1/Possible_combinations
signif(Probability_of_winning,3)

Probability_of_winning_given_15_is_in_winning_number
    <- Probability_of_winning/Probability_of_
   choosing_15_in_winning_combination
signif(Probability_of_winning_given_15_is_in_winning
   _number,2)
```

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

R code Exa 2.1.3 Rolling dice

```
#Page 57

library(MASS)

sample_space <- expand.grid(1:6, 1:6)
sums <- rowSums(sample_space)

event_A <- sums < 8
event_B <- sums %% 2 == 1

event_A_and_B <- event_A & event_B
Pr_A_and_B <- sum(event_A_and_B) / nrow(sample_space)
print(paste("Pr(A     B):", fractions(Pr_A_and_B)))

Pr_B <- sum(event_B) / nrow(sample_space)
print(paste("Pr(B):", fractions(Pr_B)))

Pr_A_given_B <- Pr_A_and_B / Pr_B
print(paste("Pr(A | B):", fractions(Pr_A_given_B)))
```

R code Exa 2.1.4 A clinical trial

```
#Page 57

library(MASS)

total_patients <- 150
```

```r

patients_placebo <- 34
patients_lithium <- 38

relapses_placebo <- 24
relapses_lithium <- 13

Pr_B <- patients_placebo / total_patients
print(paste("Pr(B):", fractions(Pr_B)))

Pr_C <- patients_lithium / total_patients
print(paste("Pr(C):", fractions(Pr_C)))

Pr_A_and_B <- relapses_placebo / total_patients
print(paste("Pr(A  B):", fractions(Pr_A_and_B)))
Pr_A_given_B <- Pr_A_and_B / Pr_B
print(paste("Pr(A | B):", signif(Pr_A_given_B,3)))

Pr_A_and_C <- relapses_lithium / total_patients
print(paste("Pr(A  C):", fractions(Pr_A_and_C)))
Pr_A_given_C <- Pr_A_and_C / Pr_C
print(paste("Pr(A | C):", signif(Pr_A_given_C,3)))
```

R code Exa 2.1.5 Rolling dice repeatedly

```r
#Page 58

library(MASS)

Pr_T_7 <- 6 / 36
Pr_T_8 <- 5 / 36

Pr_A <- Pr_T_7
Pr_B <- Pr_T_7 + Pr_T_8

```

```
p <- Pr_T_7 / Pr_B

print(fractions(p))
```

R code Exa 2.1.9 Selecting bolts

```
#Page 61

library(MASS)
Pr_B1 <- 1 / 2
Pr_B2 <- 1 / 2

Pr_A_given_B1 <- 60 / 100
Pr_A_given_B2 <- 10 / 30

Pr_A <- Pr_B1 * Pr_A_given_B1 + Pr_B2 * Pr_A_given_B2

print(fractions(Pr_A))
```

R code Exa 2.1.10 Achieving a high score

```
#Page 61

Pr_Bi <- 1/50
i <- 1:50

Pr_A_given_Bi <- 1/(51-i)
Pr_A <- sum(Pr_Bi*Pr_A_given_Bi)
signif(Pr_A,3)
```

R code Exa 2.1.12 A clinical trial II

```r
#Page 62

library(MASS)

Pr_Bj <- 1/11
j <- 1:11

Pr_E1_given_Bj <- (j-1)/10

Pr_E1 <- sum(Pr_Bj*Pr_E1_given_Bj)
print(fractions(Pr_E1))
```

R code Exa 2.2.1 Tossing coins

```r
#Page 66

library(MASS)

outcomes <- c("HH", "HT", "TH", "TT")
probabilities <- rep(1/4, 4)

event_A <- c("HH", "TH")
Pr_A <- sum(probabilities[outcomes %in% event_A])
cat("The probability Pr(A) that the second toss is H is:", Pr_A, "\n")

event_B <- c("TH", "TT")
Pr_B <- sum(probabilities[outcomes %in% event_B])

event_A_and_B <- c("TH")
Pr_A_and_B <- sum(probabilities[outcomes %in% event_A_and_B])

```

```
Pr_A_given_B <- Pr_A_and_B / Pr_B

print(fractions(Pr_A_given_B))
```

R code Exa 2.2.2 Machine operation

```
#Page 67

library(MASS)

Pr_A <- 1/3
Pr_B <- 1/4

Pr_A_and_B <- Pr_A * Pr_B
print(fractions(Pr_A_and_B))

Pr_A_or_B <- Pr_A + Pr_B - Pr_A_and_B
print(fractions(Pr_A_or_B))
```

R code Exa 2.2.3 Rolling a Die

```
#Page 67

library(MASS)
Pr_A <- 3/6
Pr_B <- 4/6

Pr_A_and_B <- 2/6

independence_check <- Pr_A_and_B == (Pr_A * Pr_B)

if (independence_check){
  print("Events A and B are independent")
```

```
}
```

R code Exa 2.2.4 Pairwise independence

```
#Page 69

sample_space <- c("HH", "HT", "TH", "TT")
probabilities <- rep(1/4, 4)

event_A <- c("HH", "HT")
event_B <- c("HH", "TH")
event_C <- c("HH", "TT")

Pr_A <- sum(probabilities[sample_space %in% event_A])
Pr_B <- sum(probabilities[sample_space %in% event_B])
Pr_C <- sum(probabilities[sample_space %in% event_C])

print(paste("Pr(A) =", fractions(Pr_A)))
print(paste("Pr(B) =", fractions(Pr_B)))
print(paste("Pr(C) =", fractions(Pr_C)))

Pr_A_and_B <- sum(probabilities[sample_space %in% intersect(event_A, event_B)])
Pr_A_and_C <- sum(probabilities[sample_space %in% intersect(event_A, event_C)])
Pr_B_and_C <- sum(probabilities[sample_space %in% intersect(event_B, event_C)])
Pr_A_and_B_and_C <- sum(probabilities[sample_space %in% intersect(intersect(event_A, event_B), event_C)])

print(paste("Pr(A   B) =", fractions(Pr_A_and_B)))
```

```
print(paste("Pr(A      C) =", fractions(Pr_A_and_C)))
print(paste("Pr(B      C) =", fractions(Pr_B_and_C)))
print(paste("Pr(A      B      C) =", fractions(Pr_A_and_B_and_C)))
```

R code Exa 2.2.7 Tossing a coin until a head appears

```
#Page 70

prob_n_tosses_for_first_head <- function(n) {
  return((1/2)^n)
}

max_n <- 100
prob_sum <- sum(sapply(1:max_n, prob_n_tosses_for_first_head))
print(paste("The sum of probabilities from n = 1 to", max_n, "is:", prob_sum))
```

R code Exa 2.2.9 People v Collins

```
#Page 70

p <- 8.3e-8
n <- 8000000

Pr_A <- 1 - (1 - p)^n

Pr_C <- n * p * (1 - p)^(n - 1)

Pr_B <- Pr_A - Pr_C

Pr_B_given_A <- Pr_B / Pr_A
```

```
print(paste("The probability Pr(B|A) is:", signif(Pr_B_given_A,4)))
#The answer may slightly vary due to rounding off values
```

R code Exa 2.2.10 Inspecting items

```
# Page 72

p1 <- 0.01
p2 <- 0.4
n <- 6
k <- 2

Pr_A_given_B1 <- choose(n, k) * p1^k * (1 - p1)^(n - k)
print(paste("The probability Pr(A|B1) is:", signif(Pr_A_given_B1,3)))

Pr_A_given_B2 <- choose(n, k) * p2^k * (1 - p2)^(n - k)
print(paste("The probability Pr(A|B2) is:", signif(Pr_A_given_B2,3)))
```

R code Exa 2.3.2 Selecting bolts II

```
#Page 77

library(MASS)

Pr_A <- 7/15
Pr_A_given_B1 <- 3/5
```

```
Pr_A_given_B2 <- 1/3
Pr_B1 <- 1/2
Pr_B2 <- 1/2

Pr_B1_given_A <- Pr_B1*Pr_A_given_B1/Pr_A
print(paste("Probability(B1|A) =",fractions(Pr_B1_
    given_A)))
```

R code Exa 2.3.3 Test for a disease

```
#Page 78

Pr_B1 <- 0.0001
Pr_B2 <- 1 - Pr_B1

Pr_A_given_B1 <- 0.90
Pr_A_given_B2 <- 0.10

Pr_B1_given_A <- Pr_A_given_B1*Pr_B1/(Pr_A_given_B1*
    Pr_B1+Pr_A_given_B2*Pr_B2)
signif(Pr_B1_given_A,2)
#The answer may vary due to difference in
    representation.
```

R code Exa 2.3.4 Identifying the source of a defective item

```
#Page 78

Pr_B1 <- 0.2
Pr_B2 <- 0.3
Pr_B3 <- 0.5

Pr_A_given_B1 <- 0.01
```

```
Pr_A_given_B2 <- 0.02
Pr_A_given_B3 <- 0.03

Pr_A <- Pr_A_given_B1 * Pr_B1 + Pr_A_given_B2 * Pr_
    B2 + Pr_A_given_B3 * Pr_B3

Pr_B2_given_A <- (Pr_A_given_B2 * Pr_B2) / Pr_A

signif(Pr_B2_given_A,2)
```

R code Exa 2.3.5 Identifying genotypes

```
#Page 79

library(MASS)

Pr_B <- c(1/16, 1/4, 1/8, 1/4, 1/4, 1/4)
Pr_E_given_B <- c(1, 1, 1, 3/4, 1/2, 0)

Pr_E <- sum(Pr_B * Pr_E_given_B)
fractions(Pr_E)

Pr_B1_given_E <- Pr_B[1]*Pr_E_given_B[1]/Pr_E
fractions(Pr_B1_given_E)

Pr_B5_given_E <- Pr_B[5]*Pr_E_given_B[5]/Pr_E
fractions(Pr_B5_given_E)
```

R code Exa 2.3.6 Learning about a proportion

```
#Page 82

Pr_B1 <- 0.9
```

```
Pr_B2 <- 0.1

Pr_A_given_B1 <- dbinom(2, size = 6, prob = 0.01)
Pr_A_given_B2 <- dbinom(2, size = 6, prob = 0.4)

Pr_B1_given_A <- (Pr_A_given_B1 * Pr_B1) / ((Pr_A_
    given_B1 * Pr_B1) + (Pr_A_given_B2 * Pr_B2))

signif(Pr_B1_given_A,1)
```

R code Exa 2.3.7 A clinical trial III

```
#Page 82

calculate_posterior <- function(prior_probs,
    likelihoods) {
  unnormalized_posterior <- prior_probs *
     likelihoods
  posterior <- unnormalized_posterior / sum(
     unnormalized_posterior)
  return(posterior)
}

j <- 1:11

prior_probs <- rep(1/11, 11)

likelihoods <- seq(0, 1, by = 0.1)

observed_success <- 1

posterior_probs <- calculate_posterior(prior_probs,
    likelihoods)

print(signif(posterior_probs,4))
```

```
likelihood <- function(j) {
  return(choose(40, 22) * ((j - 1) / 10)^22 * (1 - (
     j - 1) / 10)^18)
}

update_posterior <- function(prior_probs, observed_
    successes, observed_failures) {
  posterior_probs <- rep(0, length(prior_probs))
  for (j in 1:length(prior_probs)) {
    posterior_probs[j] <- prior_probs[j] *
       likelihood(j)
  }
  posterior_probs <- posterior_probs / sum(posterior
     _probs)
  return(posterior_probs)
}

posterior_probs <- update_posterior(prior_probs, 22,
    18)

prior_prob_next_success <- 1/2
posterior_prob_next_success <- sum(posterior_probs *
    (0:10)/10)

signif(posterior_prob_next_success,4)
```

R code Exa 2.4.1 The probability of winning in a fair game

```
#Page 88

i <- 98
k <- 100

```

```
probability_A_wins <- i / k

print(probability_A_wins)
```

R code Exa 2.4.2 The probability of winning in an unfavorable game

```
#Page 89

library(MASS)

p <- 0.4
i <- 99
k <- 100

ai <- ((((1 - p) / p) ^ i) - 1) / ((((1 - p) / p) ^
   k) - 1)

fractions(ai)
```

Chapter 3

Random Variables and Distributions

R code Exa 3.1.5 Demands for utilities

```
#Page 95

length_C <- 175 - 50
Pr_X_in_C <- (149 * length_C) / 29204

signif(Pr_X_in_C,4)
```

R code Exa 3.1.8 Daily Numbers

```
#Page 97

i <- 0:9

s <- expand.grid(i,i,i)

X <- 100*s[1] +10*s[2] +s[3]
```

```

Pr_of_each_integer <- 1/lengths(X)
print(Pr_of_each_integer)
```

R code Exa 3.2.3 Incompletely Specified PDF

```
#Page 104

library(MASS)

integral_function <- function(x) {
  return(x)
}

integral_value <- integrate(integral_function, lower
    = 0, upper = 4)$value

c_value <- 1 / integral_value

fractions(c_value)
```

R code Exa 3.2.4 Probabilities from a PDF

```
#Page 105

library(MASS)

integral_function <- function(x) {
  return(x/8)
}

Pr_X_between_1_and_2 <- integrate(integral_function,
    lower = 1, upper = 2)$value
```

```
fractions(Pr_X_between_1_and_2)

Pr_X_greater_than_2 <- integrate(integral_function, lower = 2, upper = 4)$value
fractions(Pr_X_greater_than_2)
```

R code Exa 3.2.5 Unbounded Random Variables

```
#Page 105

f <- function(x) {
  ifelse(x > 0, 1 / (1 + x)^2, 0)
}

prob_X_greater_1000 <- integrate(f, lower = 1000, upper = Inf, rel.tol = 1e-15)$value

cat("Pr(X > 1000):", signif(prob_X_greater_1000,1), "\n")
```

R code Exa 3.2.7 Truncated Voltage

```
#Page 106

library(MASS)

f <- function(x) {
  ifelse(x > 0, 1 / (1 + x)^2, 0)
}

prob_X_greater_equal_3 <- integrate(f_X, lower = 3, upper = Inf)$value

```

```r
prob_0_less_X_less_3 <- integrate(f_X, lower = 0, upper = 3)$value

print(paste("Pr(Y = 3):", fractions(prob_X_greater_equal_3)))
print(paste("Pr(0 < Y < 3):", fractions(prob_0_less_X_less_3)))
```

R code Exa 3.3.1 Voltage I

```r
#Page 107

library(MASS)

get_pmf <- function(x){
  if(x<=0){
    return (0)
  } else{
    return (1-(1/(1+x)))
  }
}

Pr_X_greater_than_3 <- get_pmf(3)
fractions(Pr_X_greater_than_3)
```

R code Exa 3.3.3 Voltage II

```r
#Page 110

library(MASS)
get_pmf <- function(x){
  if(x<=0){
    return (0)
```

```
  } else{
    return (1-(1/(1+x)))
  }
}

Pr_C <- get_pmf(4)
fractions(Pr_C)

Pr_B <- get_pmf(2)
fractions(Pr_B)

Pr_A <- Pr_C - Pr_B
fractions(Pr_A)
#The answer provided in the textbook is wrong.
```

R code Exa 3.3.9 Quantiles of a Binomial Distribution

```
#Page 115

n <- 5
p <- 0.3

pf_values <- dbinom(0:n, size = n, prob = p)
cdf_values <- pbinom(0:n, size = n, prob = p)

results <- data.frame(x = 0:n, pf = pf_values, cdf =
    cdf_values)

print(round(results,4))
```

R code Exa 3.3.10 Uniform Distribution on Integers

```
#Page 116
```

```

cdf_X <- function(x) {
  if (x < 1) {
    return(0)
  } else if (x < 2) {
    return(1 / 4)
  } else if (x < 3) {
    return(1 / 2)
  } else if (x < 4) {
    return(3 / 4)
  } else {
    return(1)
  }
}

find_quantile <- function(p) {
  uniroot(function(x) cdf_X(x) - p, lower = 1, upper = 4)$root
}

median <- find_quantile(1/2)

print(median)
```

R code Exa 3.4.3 Discrete Joint Distribution

```
# Page 191

joint_prob <- matrix(c(
  0.1, 0, 0.1, 0,
  0.3, 0, 0.1, 0.2,
  0, 0.2, 0, 0
), nrow = 3, byrow = TRUE)

rownames(joint_prob) <- c("X=1", "X=2", "X=3")
```

```
colnames(joint_prob) <- c("Y=1", "Y=2", "Y=3", "Y=4")

print(joint_prob)

pr_x_ge_2_y_ge_2 <- sum(joint_prob[2:3, 2:4])
print(pr_x_ge_2_y_ge_2)

pr_x_eq_1 <- sum(joint_prob[1, ])
print(pr_x_eq_1)
```

R code Exa 3.4.7 Normalizing Constant

```
# Page 122

library(MASS)
f_xy <- function(x, y) {
  ifelse(x^2 <= y & y <= 1,  x^2 * y, 0)
}

integrate_f_xy <- function() {
  integrate(function(x) {
    sapply(x, function(xi) {
      integrate(function(y) f_xy(xi, y), lower = xi^2, upper = 1)$value
    })
  }, lower = -1, upper = 1)$value
}

result <- integrate_f_xy()
print(paste("The integral is:", fractions(result),"*c"))
Integral_value <- 1
c <- Integral_value/result
print(paste("The value of c is:", fractions(c)))
```

R code Exa 3.4.8 Probability from a Joint PDF

```
# Page 122

library(MASS)
f_xy <- function(x, y) {
  ifelse(x^2 <= y & y <= 1, (21/4) * x^2 * y, 0)
}

integrate_f_xy <- function() {
  integrate(function(x) {
    sapply(x, function(xi) {
      integrate(function(y) f_xy(xi, y), lower = xi^2, upper = xi)$value
    })
  }, lower = 0, upper = 1)$value
}

result <- integrate_f_xy()
print(paste("Pr(X      Y ) =", fractions(result)))
```

R code Exa 3.4.9 Joint PDF by Geometric Methods

```
#Page 123

library(MASS)

f_xy <- function(x, y) {
  ifelse((x^2 + y^2) <= 9, 1, 0)
}

```

```
integrate_f_xy <- function() {
  integrate(function(x) {
    sapply(x, function(xi) {
      integrate(function(y) f_xy(xi, y), lower = -sqrt(9 - xi^2), upper = sqrt(9 - xi^2))$value
    })
  }, lower = -3, upper = 3)$value
}

result <- integrate_f_xy()
Integral_value <- 1
c <- Integral_value/result
print(paste("The value of c is:", fractions(c*pi),"*(1/  )"))
```

R code Exa 3.4.11 Joint PDF

```
#Page 124

f_xy <- function(x, y) {
  ifelse((y > 0) & (y < 1), (x * y^(x - 1)) / 3, 0)
}

total_probability <- 0

for (x in 1:3) {
  integral <- integrate(function(y) f_xy(x, y), lower = 0, upper = 1)$value
  total_probability <- total_probability + integral
}

cat("The total integral over the joint p.d.f. is:", total_probability,"\n")
```

```
probability <- 0


for (x in 2:3) {
  integral <- integrate(function(y) f_xy(x, y), lower = 1/2, upper = 1)$value
  probability <- probability + integral
}

cat("The probability Pr(X    2 and Y    1/2) is:", signif(probability,4),"\n")



integral_alternative <- integrate(function(y) {
  ifelse((y >= 1/2), (2/ 3 * y ) + y^2, 0)
}, lower = 1/2, upper = 1)$value

cat("The probability computed in the alternative order is:", signif(integral_alternative,4), "\n")
```

R code Exa 3.4.12 A Clinical Trial I

```
#Page 125

library(MASS)

f_xp <- function(x, p) {
  ifelse((p > 0) & (p < 1), (p^x)*(1-p)^(1-x), 0)
}

pr_x_le_0_and_p_le_half <- integrate(function(p) f_xp(0, p), lower = 0, upper = 1/2)$value

```

```
pr_x_eq_1 <- integrate(function(p) f_xp(1, p), lower
    = 0, upper = 1)$value

print(paste("Pr(X <= 0 and P <= 1/2):", fractions(pr
    _x_le_0_and_p_le_half)))
print(paste("Pr(X = 1):", fractions(pr_x_eq_1)))
```

R code Exa 3.5.2 Marginal PF

```
#Page 131

joint_pmf <- matrix(c(0.1, 0, 0.1, 0,
                      0.3, 0, 0.1, 0.2,
                      0, 0.2, 0, 0),
                    ncol = 4, byrow = TRUE)

rownames(joint_pmf) <- c("1", "2", "3")
colnames(joint_pmf) <- c("1", "2", "3", "4")

joint_pmf_with_totals <- addmargins(joint_pmf)

print(joint_pmf_with_totals)
```

R code Exa 3.5.4 Marginal PDF from a Joint PDF

```
#Page 134

library(MASS)

f_xy <- function(x, y) {
  (x * y^(x - 1)) / 3
}

```

```
integrate_f1 <- function(x) {
  integrate(function(y) f_xy(x, y), lower = 0, upper
     = 1)$value
}

print(paste("Integral value", fractions(integrate_f1
   (1))))
```

R code Exa 3.5.7 Games of Chance

```
#Page 137

library(MASS)

joint_pmf <- matrix(c(1/24, 1/24, 1/24, 1/24, 1/24,
   1/24,
                      1/12, 1/12, 1/12, 1/12, 1/12,
                         1/12,
                      1/24, 1/24, 1/24, 1/24, 1/24,
                         1/24),
                    nrow = 3, byrow = TRUE)

row_totals <- rowSums(joint_pmf)

joint_pmf_with_totals <- cbind(joint_pmf, row_totals
   )
joint_pmf_with_totals <- rbind(joint_pmf_with_totals
   , c(colSums(joint_pmf), sum(row_totals)))

rownames(joint_pmf_with_totals) <- c("0", "1", "2",
   "Total")
colnames(joint_pmf_with_totals) <- c("1", "2", "3",
   "4", "5", "6", "Total")

fractions(joint_pmf_with_totals)
```

R code Exa 3.5.8 Independency of Random Variable

```
#Page 137

f_1_2 <- 0.087
f1_1 <- 0.513
f2_1 <- 0.253

product_marginals <- f1_1 * f2_1

cat("f(1, 2):", f_1_2, "\n")
cat("f1(1) * f2(1):", product_marginals, "\n")

if (f_1_2 == product_marginals) {
  cat("X and Y are independent.\n")
} else {
  cat("X and Y are not independent.\n")
}
```

R code Exa 3.5.9 Probability involving Independent Random Variable

```
#Page 138

library(MASS)

joint_pdf <- function(x, y) {
  ifelse(x+y <= 1, 4*x*y, 0)
}

prob_X_plus_Y_leq_1 <- integrate(
  function(x) {
```

```
    sapply(x, function(xi) {
      integrate(function(y) joint_pdf(xi, y), lower = 0, upper = 1 - xi)$value
    })
  },
  lower = 0, upper = 1
)$value

print(paste("Pr(X + Y <= 1):", fractions(prob_X_plus_Y_leq_1)))
```

R code Exa 3.5.11 Factorization of a Joint PDF

```
#Page 139

joint_pdf <- function(x, y) {
  return(exp(-(x + 2 * y)))
}

integrate_joint_pdf <- function() {
  integrate(function(x) {
    sapply(x, function(xi) {
      integrate(function(y) joint_pdf(xi, y), lower = 0, upper = Inf)$value
    })
  }, lower = 0, upper = Inf)$value
}

marginal_pdf_x <- function(x) {
  if (x >= 0) {
    return(exp(-x))
  } else {
    return(0)
  }
}
```

```r

marginal_pdf_y <- function(y) {
  if (y >= 0) {
    return(2 * exp(-2 * y))
  } else {
    return(0)
  }
}

k <- (marginal_pdf_x(1)*marginal_pdf_y(1))/joint_pdf(1,1)
cat("k :",k)
```

R code Exa 3.6.1 Auto Insurance

```r
#Page 142

joint_pmf <- matrix(c(0.129, 0.298, 0.161, 0.280, 0.108,
                      0.010, 0.010, 0.001, 0.002, 0.001),
                    nrow = 2, byrow = TRUE)

joint_pmf_with_totals <- cbind(joint_pmf, rowSums(joint_pmf))
joint_pmf_with_totals <- rbind(joint_pmf_with_totals, c(colSums(joint_pmf), sum(colSums(joint_pmf))))

rownames(joint_pmf_with_totals) <- c("0", "1", "Total")
colnames(joint_pmf_with_totals) <- c("1", "2", "3", "4", "5", "Total")

print(joint_pmf_with_totals)
```

R code Exa 3.6.2 Conditional PF from a Joint PF

```
#Page 143

library(MASS)
joint_prob <- matrix(c(
  0.1, 0,   0.1, 0,
  0.3, 0,   0.1, 0.2,
  0,   0.2, 0,   0
), nrow = 3, byrow = TRUE)

marginal_prob_X2 <- 0.6

conditional_prob_Y_given_X2 <- joint_prob[2, ] / marginal_prob_X2

names(conditional_prob_Y_given_X2) <- paste("g2(",1:4,"| 2)" )
fractions(conditional_prob_Y_given_X2)
```

R code Exa 3.6.5 Processing Times

```
#Page 145

f_X <- function(x) {
  ifelse(x >= 0, exp(4-x), 0)
}

conditional_prob_X_geq_9_given_Y_4 <- integrate(f_X, lower = 9, upper = Inf)$value

signif(conditional_prob_X_geq_9_given_Y_4,2)
```

R code Exa 3.6.6 Conditional PDF from a Joint PDF

```
#Page 145

library(MASS)
g2 <- function(y, x) {
  ifelse((x^2 <= y) & (y <= 1), 2 * y / (1 - x^4), 0)
}

integrate_g2_1 <- function(x) {
  integrate(function(y) g2(y, x), lower = 1/4, upper = 1)$value
}

prob1 <- integrate_g2_1(1/2)

print(prob1)

integrate_g2_2 <- function(x) {
  integrate(function(y) g2(y, x), lower = 3/4, upper = 1)$value
}

prob2 <- integrate_g2_2(1/2)

fractions(prob2)
```

R code Exa 3.6.9 Defective Parts

```
#Page 148
```

```r

library(MASS)

integrand <- function(q) {
  q * (1 - q)
}

result <- integrate(integrand, lower = 0, upper = 1)
    $value

fractions(result)
```

R code Exa 3.6.10 Uniform Distributions

```r
#Page 149

f2 <- function(y) {
  ifelse(y > 0 & y < 1, -log(1 - y), 0)
}

y_vals <- seq(0.01, 0.99, length.out = 100)

f2_vals <- f2(y_vals)

plot(y_vals, f2_vals, type = "l", col = "blue", lwd
    = 2,
     xlab = "y", ylab = "f2(y)", main = "Marginal p.
        d.f. f2(y)",
     ylim = c(0, 5))

abline(h = 0, col = "gray", lty = 2)
```

R code Exa 3.7.7 Arrivals at a Queue

```r
#Page 155

sum_h13 <- function(y){
  ifelse(y>0, 2 * exp(-2*y), 0)
}
final_result <- integrate(sum_h13 , lower = 0, upper
    = Inf)$value

print(final_result)
```

R code Exa 3.7.8 A Clinical Trial II

```r
#Page 156

library(MASS)

integrand <- function(p) {
  p * (1 - p)
}

integral_value <- integrate(integrand, lower = 0,
   upper = 1)$value

prob_x1_x2_1 <- 2 * integral_value

print(fractions(prob_x1_x2_1))
```

R code Exa 3.7.9 Service Times in a Queue I

```r
#Page 157

fi <- function(xi) {
  ifelse(xi > 0, 2 / (2 + xi)^2, 0)
```

```
}

integrand <- function(xi) {
  sapply(xi, fi)
}

result <- integrate(integrand, lower = 3, upper =
   Inf)$value

cat("Pr(Xi > 3) =", result, "\n")
```

R code Exa 3.7.14 Service Times in a Queue II

```
#Page 160

library(cubature)

g1 <- function(x3, x4, x5, x1, x2) {
  return(60 * (2 + x1 + x2)^3 / (2 + x1 + x2 + x3 +
     x4 + x5)^6)
}

x1 <- 4
x2 <- 6

result <- adaptIntegrate(function(x) {
  x3 <- x[1]
  x4 <- x[2]
  x5 <- x[3]
  return(g1(x3, x4, x5, x1, x2))
}, lowerLimit = c(3, 0, 0), upperLimit = c(Inf, Inf,
    Inf))

conditional_probability <- result$integral

```

```
print(paste("Pr(X3 > 3|X1 = 4, X2 = 6) =",round(
    conditional_probability,3)))

x1 <- 1
x2 <- 1.5

result <- adaptIntegrate(function(x) {
  x3 <- x[1]
  x4 <- x[2]
  x5 <- x[3]
  return(g1(x3, x4, x5, x1, x2))
}, lowerLimit = c(3, 0, 0), upperLimit = c(Inf, Inf,
     Inf))

conditional_probability <- result$integral

print(paste("Pr(X3 > 3|X1 = 1, X2 = 1.5) = ", round(
    conditional_probability,3)))
```

R code Exa 3.7.15 Marginal Bivariate PDF

```
#Page 161

library(MASS)

integrand  <- function(x1, x2) {
  return(4 / (2 + x1 + x2)^3)
}

result <- integrate(function(x2) {
  sapply(x2, function(x2) {
    integrate(function(x1) {
      integrand(x1, x2)
    }, lower = 0, upper = 4 - x2)$value
  })
```

```
}, lower = 0, upper = 4)$value

fractions(result)
```

R code Exa 3.7.16 Service Times in a Queue III

```
#Page 162

g0 <- function(z, x1, x2) {
  ifelse (z > 0, (1/2) * (2 + x1 + x2)^3 * z^2 * exp(-z * (2 + x1 + x2)), 0)
}

x1 <- 1
x2 <- 4

integrand <- function(z) {
  g0(z, x1, x2)
}

probability <- integrate(integrand, lower = 0, upper = 1)$value

signif(probability,4)
```

R code Exa 3.7.19 Rate of Service I

```
#Page 164
set.seed(123)

n <- 100
rate <- 2

```

```
z <- rexp(1, rate = rate)

service_times <- rexp(n, rate = z)

first_10 <- service_times[1:10]
cat("First 10 observations:", first_10,"\n")

smallest_time <- min(service_times)
largest_time <- max(service_times)
cat("Smallest service time:",smallest_time,"\n")
cat("Largest service time:",largest_time,"\n")
```

R code Exa 3.7.20 Rate of Service II

```
#Page 165
set.seed(123)

n <- 100
rate <- 2

z <- rexp(1, rate = rate)

service_times <- rexp(n, rate = z)

service_times <- ifelse(service_times > 10, runif(
    sum(service_times > 10), 0, 10), service_times)

hist(service_times, breaks = seq(0, 10, by = 1),
    main = "Histogram of Service Times", xlab = "
    Service Time", border = "black")
```

R code Exa 3.8.1 Distance from the Middle I

```r
#Page 168

library(MASS)

prob_X_4_or_6 <- function() {
  lower_bound <- 0
  upper_bound <- 9

  prob_X_4 <- dunif(4, min = lower_bound, max =
     upper_bound)

  prob_X_6 <- dunif(6, min = lower_bound, max =
     upper_bound)

  prob_X_4 + prob_X_6
}

prob_Y_1 <- prob_X_4_or_6()

fractions(prob_Y_1)
```

R code Exa 3.8.2 Distance from the Middle II

```r
#Page 168

library(MASS)

f_X <- function(x) {
  if (x >= 0 && x <= 9) {
    return(1 / 9)
  } else {
    return(0)
  }
}

```

```
g_Y <- function(y) {
  if (y == 0) {
    return(f_X(5))
  } else if (y >= 1 && y <= 4) {
    return(2 * f_X(5 - y))
  } else {
    return(0)
  }
}

prob_Y_0 <- g_Y(0)

prob_Y_4 <- g_Y(4)

print(paste("Pr(Y = 0) =", fractions(prob_Y_0)))

print(paste("Pr(Y = 4) =", fractions(prob_Y_4)))
```

R code Exa 3.8.4 PDF of square of Uniform Distribution

```
#Page 169

set.seed(123)

g_y <- function(y) {
  ifelse(y >= 0 & y <= 1, 1 / (2 * sqrt(y)), 0)
}

y_values <- sort(runif(50, min = 0, max = 1))
pdf_y <- g_y(y_values)

plot(y_values, pdf_y, type = "l", col = "blue", lwd = 2,
     main = "PDF of Y = X^2 where X ~ Uniform(-1, 1)",
```

```
      xlab = "y", ylab = "g(y)")
```

R code Exa 3.8.6 Generating Independent Values from a Specified PDF

```
#Page 171

G_inv <- function(x) {
  return(2 * (1 - (1 - x)^(1/2)))
}

x_values <- c(0.4125, 0.0894, 0.8302)

y_values <- G_inv(x_values)

result_df <- data.frame(x = x_values, y = round(y_
    values,2))

print(result_df)
```

R code Exa 3.9.2 Bull Market

```
#Page 176

n_points_A <- choose(10, 3) * choose(10, 5) * choose
    (10, 8)

print(n_points_A)

total_possible_values <- 2^30

g_358 <- n_points_A / total_possible_values

signif(g_358,3)
```

R code Exa 3.9.5 An Investment Portfolio

```
#Page 179

g_y <- function(y) {
  ifelse(y > 1800 & y <= 2200, 8.333e-7 * (y - 1800)
    ,
         ifelse(y > 2200 & y <= 4800, 3.333e-4,
                ifelse(y > 4800 & y < 5200, 8.333e-7
                    * (5200 - y), 0)))
}

y_vals <- seq(from = 1800, to = 5200, by = 1)

g_y_values <- sapply(y_vals, g_y)

cat("For 1800 < y <= 2200, g(y) =", g_y_values[y_
   vals > 1800 & y_vals <= 2200][1], "\n")
cat("For 2200 < y <= 4800, g(y) =", g_y_values[y_
   vals > 2200 & y_vals <= 4800][1], "\n")
cat("For 4800 < y < 5200, g(y) =", g_y_values[y_vals
    > 4800 & y_vals < 5200][1], "\n")
```

R code Exa 3.9.6 Maximum and Minimum of a Random Sample

```
#Page 180

library(ggplot2)

set.seed(121)

```

```
n <- 5
num_samples <- 10000

samples <- matrix(runif(n * num_samples, min = 0,
    max = 1), nrow = num_samples, ncol = n)

single_vals <- as.vector(samples)
min_vals <- apply(samples, 1, min)
max_vals <- apply(samples, 1, max)
range_vals <- max_vals - min_vals

density_single <- density(single_vals)
density_min <- density(min_vals)
density_max <- density(max_vals)
density_range <- density(range_vals)

plot(density_single, col = "blue", xlim = c(0, 1),
    ylim = c(0, 5), xlab = "x", ylab = "Density",
      main = "PDFs of Single Random Variable, Min,
        Max, and Range of 5 Samples")
lines(density_min, col = "red", lty = 2)
lines(density_max, col = "green", lty = 2)
lines(density_range, col = "purple", lty = 2)
legend("topright", legend = c("Single Random
    Variable", "Min of 5", "Max of 5", "Range of 5"),
        col = c("blue", "red", "green", "purple"),
           lty = c(1, 2, 2, 2))
```

R code Exa 3.10.2 Shopping for Toothpaste I

```
#Page 189

library(MASS)

transition_matrix <- matrix(c(1/3, 2/3, 2/3, 1/3),
```

```
    nrow = 2, byrow = TRUE)
colnames(transition_matrix) <- c("A", "B")
rownames(transition_matrix) <- c("A", "B")

print(paste("Pr(Xn+1 = 1 | Xn = 1) =", fractions(transition_matrix[1, 1])))
print(paste("Pr(Xn+1 = 2 | Xn = 1) =", fractions(transition_matrix[1, 2])))
print(paste("Pr(Xn+1 = 1 | Xn = 2) =", fractions(transition_matrix[2, 1])))
print(paste("Pr(Xn+1 = 2 | Xn = 2) =", fractions(transition_matrix[2, 2])))
```

R code Exa 3.10.4 Shopping for Toothpaste II

```
#Page 190

library(MASS)

transition_matrix <- matrix(c(1/3, 2/3, 2/3, 1/3),
    nrow = 2, byrow = TRUE)
colnames(transition_matrix) <- c("A", "B")
rownames(transition_matrix) <- c("A", "B")

fractions(transition_matrix)
```

R code Exa 3.10.5 Number of Occupied Telephone Lines I

```
#Page 191

P <- matrix(c(0.1, 0.4, 0.2, 0.1, 0.1, 0.1,
              0.2, 0.3, 0.2, 0.1, 0.1, 0.1,
              0.1, 0.2, 0.3, 0.2, 0.1, 0.1,
```

```
                 0.1, 0.1, 0.2, 0.3, 0.2, 0.1,
                 0.1, 0.1, 0.1, 0.2, 0.3, 0.2,
                 0.1, 0.1, 0.1, 0.1, 0.4, 0.2), nrow = 6, byrow = TRUE)

print(P)

prob_5_4 <- P[6, 5]
cat("The probability P(5, 4) is:", prob_5_4, "\n")

prob_0_0 <- P[1, 1]
cat("The probability P(0, 0) is:", prob_0_0, "\n")

prob_at_least_one <- 1 - prob_0_0
cat("The probability that at least one line will be in use is:", prob_at_least_one, "\n")
```

R code Exa 3.10.6 Plant Breeding Experiment I

```
#Page 192

states <- c("{AA, AA}", "{AA, Aa}", "{AA, aa}", "{Aa, Aa}", "{Aa, aa}", "{aa, aa}")

transition_matrix <- matrix(0, nrow = 6, ncol = 6)
rownames(transition_matrix) <- states
colnames(transition_matrix) <- states

transition_matrix["{AA, AA}", "{AA, AA}"] <- 1

transition_matrix["{AA, Aa}", "{AA, AA}"] <- 0.25
transition_matrix["{AA, Aa}", "{AA, Aa}"] <- 0.50
transition_matrix["{AA, Aa}", "{Aa, Aa}"] <- 0.25

transition_matrix["{AA, aa}", "{Aa, Aa}"] <- 1
```

```

transition_matrix["{Aa, Aa}", "{AA, AA}"] <- 1/16
transition_matrix["{Aa, Aa}", "{AA, Aa}"] <- 1/4
transition_matrix["{Aa, Aa}", "{AA, aa}"] <- 1/8
transition_matrix["{Aa, Aa}", "{Aa, Aa}"] <- 1/4
transition_matrix["{Aa, Aa}", "{Aa, aa}"] <- 1/4
transition_matrix["{Aa, Aa}", "{aa, aa}"] <- 1/16

transition_matrix["{Aa, aa}", "{Aa, Aa}"] <- 1/4
transition_matrix["{Aa, aa}", "{Aa, aa}"] <- 1/2
transition_matrix["{Aa, aa}", "{aa, aa}"] <- 1/4

transition_matrix["{aa, aa}", "{aa, aa}"] <- 1

print(transition_matrix)
```

R code Exa 3.10.7 Single Server Queue I

```
#Page 193

P <- matrix(c(0.9, 0.1, 0.6, 0.4), nrow = 2, byrow =
    TRUE)

Pr_Xn1_Xn2_given_Xn1 <- P[1, 1] * P[1, 1]

Pr_Xn1_Xn2_given_Xn1 <- c(Pr_Xn1_Xn2_given_Xn1, P[1,
    2] * P[2, 1])

Pr_Xn2_given_Xn1_1 <- sum(Pr_Xn1_Xn2_given_Xn1)

Pr_Xn2_given_Xn1_2 <- 1 - Pr_Xn2_given_Xn1_1

Pr_Xn2_given_Xn2_1 <- sum(P[2,] * P[, 1])

Pr_Xn2_given_Xn2_2 <- 1 - Pr_Xn2_given_Xn2_1
```

```
cat("Pr(Xn+1 = 1, Xn+2 = 1|Xn = 1):", Pr_Xn1_Xn2_given_Xn1[1], "\n")
cat("Pr(Xn+1 = 2, Xn+2 = 1|Xn = 1):", Pr_Xn1_Xn2_given_Xn1[2], "\n")
cat("Pr(Xn+2 = 1|Xn = 1):", Pr_Xn2_given_Xn1_1, "\n")
cat("Pr(Xn+2 = 2|Xn = 1):", Pr_Xn2_given_Xn1_2, "\n")
cat("Pr(Xn+2 = 1|Xn = 2):", Pr_Xn2_given_Xn2_1, "\n")
cat("Pr(Xn+2 = 2|Xn = 2):", Pr_Xn2_given_Xn2_2, "\n")
```

R code Exa 3.10.8 Number of Occupied Telephone Lines II

```
#Page 195

P <- matrix(c(0.1, 0.4, 0.2, 0.1, 0.1, 0.1,
              0.2, 0.3, 0.2, 0.1, 0.1, 0.1,
              0.1, 0.2, 0.3, 0.2, 0.1, 0.1,
              0.1, 0.1, 0.2, 0.3, 0.2, 0.1,
              0.1, 0.1, 0.1, 0.2, 0.3, 0.2,
              0.1, 0.1, 0.1, 0.1, 0.4, 0.2), nrow = 6, byrow = TRUE)

print(P)

P2 <- P %*% P

P3 <- P2 %*% P

cat("P2:\n")
print(P2)

```

```
cat("\nP3:\n")
print(P3)
```

R code Exa 3.10.9 Plant Breeding Experiment II

```
#Page 195

inheritance_matrix <- matrix(c(1.0000, 0.0000,
   0.0000, 0.0000, 0.0000, 0.0000,
                              0.2500, 0.5000,
                                 0.0000, 0.2500,
                                 0.0000, 0.0000,
                              0.0000, 0.0000,
                                 0.0000, 1.0000,
                                 0.0000, 0.0000,
                              0.0625, 0.2500,
                                 0.1250, 0.2500,
                                 0.2500, 0.0625,
                              0.0000, 0.0000,
                                 0.0000, 0.2500,
                                 0.5000, 0.2500,
                              0.0000, 0.0000,
                                 0.0000, 0.0000,
                                 0.0000, 1.0000),
                            nrow = 6, byrow = TRUE)

row_names <- c("{AA, AA}", "{AA, Aa}", "{AA, aa}", "
   {Aa, Aa}", "{Aa, aa}", "{aa, aa}")
col_names <- c("{AA, AA}", "{AA, Aa}", "{AA, aa}", "
   {Aa, Aa}", "{Aa, aa}", "{aa, aa}")
rownames(inheritance_matrix) <- row_names
colnames(inheritance_matrix) <- col_names

print(inheritance_matrix)

```

```
two_step_transition_matrix <- inheritance_matrix %*%
    inheritance_matrix

print(round(two_step_transition_matrix,4))
```

R code Exa 3.10.10 Single Server Queue II

```
#Page 196

P_server_is_busy <- 0.3

v <- c(P_server_is_busy, 1 - P_server_is_busy)

print(v)
```

R code Exa 3.10.11 Number of Occupied Telephone Lines III

```
#Page 197

P <- matrix(c(0.1, 0.4, 0.2, 0.1, 0.1, 0.1,
              0.2, 0.3, 0.2, 0.1, 0.1, 0.1,
              0.1, 0.2, 0.3, 0.2, 0.1, 0.1,
              0.1, 0.1, 0.2, 0.3, 0.2, 0.1,
              0.1, 0.1, 0.1, 0.2, 0.3, 0.2,
              0.1, 0.1, 0.1, 0.1, 0.4, 0.2), nrow =
                6, byrow = TRUE)

v <- c(0.5, 0.3, 0.2, 0, 0, 0)

v_P <- v %*% P
cat("Distribution of the number of lines in use at
   time 2:", v_P, "\n")

```

```
v_P2 <- v %*% P %*% P
cat("Distribution of the number of lines in use at
    time 3:", v_P2, "\n")
```

R code Exa 3.10.12 Distribution for Telephone Lines

```
#Page 197

P <- matrix(c(0.1, 0.4, 0.2, 0.1, 0.1, 0.1,
              0.2, 0.3, 0.2, 0.1, 0.1, 0.1,
              0.1, 0.2, 0.3, 0.2, 0.1, 0.1,
              0.1, 0.1, 0.2, 0.3, 0.2, 0.1,
              0.1, 0.1, 0.1, 0.2, 0.3, 0.2,
              0.1, 0.1, 0.1, 0.1, 0.4, 0.2), nrow =
                 6, byrow = TRUE)

v <- c(0.119, 0.193, 0.186, 0.173, 0.196, 0.133)

vP <- round(v %*% P,3)

if(isTRUE(all.equal(as.vector(vP), v))) {
  cat("The initial distribution remains the same
      after one transition and therefore after two or
       more transitions as well.\n")
} else {
  cat("The initial distribution does not remain the
      same after one transition.\n")
}
```

R code Exa 3.10.14 Toothpaste Shopping

```
#Page 199

```

```
library(MASS)

P <- matrix(c(1/3, 2/3, 2/3, 1/3), nrow = 2, byrow =
    TRUE)

I <- diag(nrow(P))
G <- P - I
G[,ncol(G)] <- 1

fractions(G)

G_inv <- solve(G)

fractions(G_inv)

v <- G_inv[nrow(G_inv), ]

print(fractions(v))
```

R code Exa 3.10.15 Alternating Chain

```
#Page 199

P <- matrix(c(0, 1, 1, 0), nrow = 2, byrow = TRUE)

I <- diag(nrow(P))
G <- P - I
G[,ncol(G)] <- 1

fractions(G)

G_inv <- solve(G)

fractions(G_inv)

```

```
v <- G_inv[nrow(G_inv), ]

print(v)
```

Chapter 4

Expectation

R code Exa 4.1.2 Stock Price Change

```
#Page 207

values <- c(-2, 0, 1, 4)
probabilities <- c(0.1, 0.4, 0.3, 0.2)

weighted_average <- sum(values * probabilities)
print(paste("Weighted Average:", weighted_average))

initial_amount <- 18
interest_rate <- 0.05
interest_earned <- initial_amount * interest_rate
print(paste("Interest Earned:", interest_earned))
```

R code Exa 4.1.6 Expected Failure Time

```
#Page 209

library(MASS)
```

```
integrand <- function(x) {
  return(2 * x^2)
}

result <- integrate(integrand, lower = 0, upper = 1)

expected_value <- result$value
print(paste("The expected value E(X) is:", fractions
   (expected_value)))
```

R code Exa 4.1.7 Failure after Warranty

```
#Page 210

f <- function(x) {
  return(2 / x^3)
}

integrand <- function(x){
  return(x*f(x))
}

result <- integrate(integrand, lower = 1, upper =
   Inf)

expected_value <- result$value
print(paste("The expected value E(X) is:", fractions
   (expected_value)))
```

R code Exa 4.1.9 The Cauchy Distribution I

```
#Page 211
```

```
library(ggplot2)
library(gridExtra)

f <- function(x) {
  1 / (pi * (1 + x^2))
}

g <- function(x) {
  x / (pi * (1 + x^2))
}

x <- seq(-10, 10, length.out = 1000)

data_f <- data.frame(x = x, y = f(x))
data_g <- data.frame(x = x, y = g(x))

p1 <- ggplot(data_f, aes(x, y)) +
  geom_line(color = "blue") +
  ggtitle("p.d.f. of the Cauchy Distribution") +
  xlab("x") +
  ylab("f(x)")+
  ylim(0,1/pi)

p2 <- ggplot(data_g, aes(x, y)) +
  geom_line(color = "red") +
  ggtitle("Function y = x * f(x)") +
  xlab("x") +
  ylab("x * f(x)") +
  ylim(-1/(2*pi),1/(2*pi))

grid.arrange(p1, p2, ncol = 2)
```

R code Exa 4.1.11 Failure Rate and Time to Failure I

```
#Page 212

library(MASS)

integrand <- function(y) {
  return(y * 3 * y^(-4))
}

result <- integrate(integrand, lower = 1, upper = Inf)

print(paste("The mean of Y (E(Y)) is:", fractions(result$value)))
```

R code Exa 4.1.12 Failure Rate and Time to Failure II

```
#Page 214

library(MASS)

integrand <- function(x) {
  return( 3 * x)
}

result <- integrate(integrand, lower = 0, upper = 1)

print(paste("The mean of X (E(X)) is:", fractions(result$value)))
```

R code Exa 4.1.13 Expectation of square root of X

```
#Page 214

```

```
library(MASS)

integrand <- function(x) {
  return(x^(1/2) * (2 * x))
}

result <- integrate(integrand, lower = 0, upper = 1)
fractions(result$value)
```

R code Exa 4.1.14 Option Pricing

```
#Page 214

p <- (200 * 1.04 - 180) / (260 - 180)

c <- 60 * p / 1.04

print(paste("Resulting price",round(c,2)))
```

R code Exa 4.1.16 Expectation of a Function of Two Variables

```
#Page 215

library(MASS)
integrand <- function(x, y) {
  return((x^2 + y^2))
}

result <- integrate(Vectorize(function(y) {
  integrate(function(x) integrand(x, y), lower = 0,
     upper = 1)$value
}), lower = 0, upper = 1, abs.tol = 1e-9)

```

```
fractions(result$value)
```

R code Exa 4.2.1 Expectation of a Linear Function

```
#Page 218

E_X <- 5

E_3X_minus_5 <- 3 * E_X - 5
print(E_3X_minus_5)

E_neg_3X_plus_15 <- -3 * E_X + 15
print(E_neg_3X_plus_15)
```

R code Exa 4.2.2 Investment I

```
#Page 218

n_sim <- 100000
p <- 0.03

set.seed(123)
returns_R1 <- runif(n_sim, min = -10, max = 20)
set.seed(102)
returns_R2 <- runif(n_sim, min = -4.5, max = 10)

expected_return_R1 <- mean(returns_R1)
expected_return_R2 <- mean(returns_R2)

VaR_R1 <- quantile(returns_R1, p)
VaR_R2 <- quantile(returns_R2, p)

total_return_R1 <- 120 * returns_R1
```

```
total_return_R2 <- 200 * returns_R2

total_VaR_R1 <- quantile(total_return_R1, p)
total_VaR_R2 <- quantile(total_return_R2, p)

expected_total_return_R1 <- 120 * expected_return_R1
expected_total_return_R2 <- 200 * expected_return_R2

cat("Total VaR for stock 1:", abs(round(total_VaR_R1
    ,0)), "\n")
cat("Total VaR for stock 2:", abs(round(total_VaR_R2
    ,0)), "\n")
```

R code Exa 4.2.3 Investment Portfolio I

```
#Page 220

mean_return_portfolio <- function(s1, s2) {
  return (5 * s1 + 2.75 * s2)
}

s1 <- 54
s2 <- 110

mean_return <- mean_return_portfolio(s1, s2)

cat("Mean return for the portfolio with s1 =", s1, "
    and s2 =", s2, ":", mean_return, "\n")
```

R code Exa 4.2.6 Expected Number of Matches

```
#Page 221

```

```r
n <- 10

expected_value <- 1/n * n

cat("Expected value of X:", expected_value, "\n")
```

R code Exa 4.2.7 Expectation of a Combination of Random Variables

```r
#Page 222

E_X <- 0
E_X2 <- 1

result <- E_X2 * (E_X2 - 8 * E_X2 * E_X + 16)

cat("E[X^2_1(X^2 - 4X^3)^2]:", result, "\n")
```

R code Exa 4.2.8 Repeated Filtering

```r
#Page 222

mu <- integrate(function(x) (1 - x) * 4 * x^3, lower = 0, upper = 1)$value

E_Y <- mu^2


cat("Mean    :", mu, "\n")
cat("Expected value of Y:", E_Y, "\n")
```

R code Exa 4.2.10 Expected Waiting Time

```
#Page 224

library(MASS)

F <- function(x) {
  ifelse(x <= 0, 0, 1 - exp(-2 * x))
}

E_X <- integrate(function(x) x * 2 * exp(-2 * x),
   lower = 0, upper = Inf)$value

print(paste("Mean of X:", fractions(E_X)))
```

R code Exa 4.3.1 Stock Price Changes I

```
#Page 225

library(MASS)

lower_bound_A <- 25
upper_bound_A <- 35
lower_bound_B <- 15
upper_bound_B <- 45

Pr_B_less_than_25 <- (25 - lower_bound_B) / (upper_
   bound_B - lower_bound_B)

print(paste("Probability that B < 25:", fractions(Pr
   _B_less_than_25)))
```

R code Exa 4.3.2 Stock Price Changes II

```
#Page 226

library(MASS)

lower_bound_A <- 25
upper_bound_A <- 35
lower_bound_B <- 15
upper_bound_B <- 45

Var_A <- integrate(function(a) (a - 30)^2 / 10,
    lower_bound_A, upper_bound_A)$value

Var_B <- integrate(function(b) (b - 30)^2 / 30,
    lower_bound_B, upper_bound_B)$value

sd_A <- sqrt(Var_A)

sd_B <- sqrt(Var_B)

print(paste("Variance of A:", fractions(Var_A)))
cat("Variance of B:", fractions(Var_B), "\n")
cat("Standard deviation of A:", round(sd_A,2), "\n")
cat("Standard deviation of B:", round(sd_B,2), "\n")
```

R code Exa 4.3.3 Variance and Standard Deviation of a Discrete Distribution

```
#Page 227

values <- c(-2, 0, 1, 3, 4)
probabilities <- rep(1/5, 5)

mean_X <- sum(values * probabilities)

var_X <- sum(probabilities * (values - mean_X)^2)
```

```
sd_X <- sqrt(var_X)

cat("Mean of X:", mean_X, "\n")
cat("Variance of X:", var_X, "\n")
cat("Standard deviation of X:", round(sd_X,3), "\n")
```

R code Exa 4.3.4 Variance of a Discrete Distribution

```
#Page 227

values <- c(-2, 0, 1, 3, 4)
probabilities <- rep(1/5, 5)

E_X2 <- sum(probabilities * values^2)

mean_X <- sum(values * probabilities)

var_X <- E_X2 - mean_X^2

cat("Mean of X:", mean_X, "\n")
cat("Expectation of X^2:", E_X2, "\n")
cat("Variance of X:", var_X, "\n")
```

R code Exa 4.3.5 Modification of a Bernoulli Distribution

```
#Page 228

x_values <- c(0, 1, 10000)
probabilities <- c(0.5, 0.499, 0.001)

E_X <- sum(x_values * probabilities)

```

```
E_X2 <- sum((x_values^2) * probabilities)

Var_X <- E_X2 - (E_X^2)

cat("E(X) =", E_X, "\n")
cat("E(X^2) =", E_X2, "\n")
cat("Var(X) =", Var_X, "\n")
```

R code Exa 4.3.6 Variance and Standard Deviation of a Linear Function

```
#Page 229

mu_X <- 1.2
var_X <- 4.56
a <- 4
b <- -7

var_Y <- a^2 * var_X

sd_Y <- a * sqrt(var_X)

cat("Variance of Y =", var_Y, "\n")
cat("Standard deviation of Y =", round(sd_Y,2), "\n"
  )
```

R code Exa 4.3.7 Investment Portfolio II

```
#Page 230

mean_R1 <- 6
var_R1 <- 55
mean_R2 <- 4
var_R2 <- 28
```

```r
fixed_rate <- 0.036
initial_investment <- 100000
price_R1 <- 60
price_R2 <- 48

s1 <- 524.7
s2 <- 609.7

s3 <- initial_investment - (s1 * price_R1 + s2 *
    price_R2)

mean_return <- s1 * mean_R1 + s2 * mean_R2 + fixed_
    rate * s3

variance_return <- (s1^2 * var_R1) + (s2^2 * var_R2)

cat("Mean return of the portfolio =", round(mean_
    return), "\n")
cat("Variance of the portfolio =", signif(variance_
    return,3), "\n")
cat("Shares of stock 1 =", s1, "\n")
cat("Shares of stock 2 =", s2, "\n")
cat("Amount invested at the fixed rate =", signif(s3
    ,4), "\n")

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

R code Exa 4.3.9 The Cauchy Distribution II

```r
# Page 233

quantile_function_X <- function(p) {
  return(tan(pi * (p - 0.5)))
}
```

```

Q_75 <- quantile_function_X(0.75)
Q_25 <- quantile_function_X(0.25)

IQR <- Q_75 - Q_25

cat("Interquartile Range (IQR):", IQR, "\n")


quantile_function_Y <- function(p) {
  return(2 * quantile_function_X(p))
}

Q_Y_75 <- quantile_function_Y(0.75)
Q_Y_25 <- quantile_function_Y(0.25)

IQR_Y <- Q_Y_75 - Q_Y_25

cat("Interquartile Range (IQR) of Y:", IQR_Y, "\n")
```

R code Exa 4.4.2 Skewness of Binomial Distributions

```
#Page 236

n <- 10
p <- 0.25

mu <- n * p
sigma <- sqrt(n * p * (1 - p))
sigma <- round(sigma,4)

E_X_minus_mu_cubed <- sum((0:n - mu)^3 * dbinom(0:n, n, p))

skewness <- E_X_minus_mu_cubed / sigma^3
```

```
cat("Mean (mu) =", mu, "\n")
cat("Standard deviation (sigma) =", round(sigma,3),
  "\n")
cat("E[(X - mu)^3] =", E_X_minus_mu_cubed, "\n")
cat("Skewness =", round(skewness,4), "\n")
```

R code Exa 4.4.3 Moment Generating Function

```
#Page 237

mgf_prime <- function(t) {
  if (t < 1) {
    return(1 / (1 - t)^2)
  } else {
    return(NA)
  }
}

mgf_double_prime <- function(t) {
  if (t < 1) {
    return(2 / (1 - t)^3)
  } else {
    return(NA)
  }
}

E_X <- mgf_prime(0)
E_X2 <- mgf_double_prime(0)

Var_X <- E_X2 - E_X^2

cat("E(X) =", E_X, "\n")
cat("E(X^2) =", E_X2, "\n")
cat("Var(X) =", Var_X, "\n")
```

R code Exa 4.5.1 Median of a Discrete Distribution I

```
# Page  242

probs <- c(0.1, 0.2, 0.3, 0.4)
values <- c(1, 2, 3, 4)

cdf <- cumsum(probs)

median_value <- values[min(which(cdf >= 0.5))]

cat("Median of the distribution:", median_value, "\n
  ")
```

R code Exa 4.5.2 Median of a Discrete Distribution II

```
# Page 242

probs <- c(0.1, 0.4, 0.3, 0.2)
values <- c(1, 2, 3, 4)

median_value <- mean(values)

cat("Midpoint median of the distribution:", median_
  value, "\n")
```

R code Exa 4.5.3 Median of a Continuous Distribution

```
#Page 242
```

```

cdf <- function(x) {
  if (x < 0) {
    return(0)
  } else if (x >= 1) {
    return(1)
  } else {
    return(x^4)
  }
}

median_value <- uniroot(function(x) cdf(x) - 0.5,
   lower = 0, upper = 1)$root

cat("The median of the distribution is:", median_
   value, "\n")
# The answer may vary due to difference in
   representation
```

R code Exa 4.5.7 Last Lottery Number I

```
#Page 244

p <- 0.001
q <- 0.999

mean_X <- p * (1 / (1 - q)^2)

cat("The mean of X is:", mean_X, "\n")
```

R code Exa 4.5.8 Last Lottery Number II

```
# Page 246
```

```
p <- 0.001
q <- 0.999

median_X <- log(0.5) / log(q)

median_X <- ceiling(median_X)

cat("The median of X is:", median_X, "\n")
```

R code Exa 4.5.9 Discrete Uniform Random Variable

```
#Page 246
values <- 1:6
probabilities <- rep(1/6, 6)

expected_value <- sum(values * probabilities)

median_value <- median(values)

cat("The value that minimizes the MSE is:", expected
    _value, "\n")

cat("The value that minimizes the MAE is:", median_
    value, "\n")
```

R code Exa 4.6.2 Test Scores I

```
#Page 249

library(MASS)

inner_integrand <- function(y, x) {
```

```
  2 * x^2 * y + 0.5 * x
}

inner_integrand_vec <- Vectorize(inner_integrand, vectorize.args = "y")

integrate_inner <- function(x) {
  sapply(x, function(xi) {
    integrate(inner_integrand_vec, 0, 1, x = xi)$value
  })
}

integrate_inner_vec <- Vectorize(integrate_inner)


mu_X <- integrate(integrate_inner_vec, 0, 1)$value
print(paste("  X =", fractions(mu_X)))

mu_Y <- mu_X

cov_xy <- function(x, y) {
  (x - mu_X) * (y - mu_Y) * (2 * x * y + 0.5)
}

covariance <- integrate(
  function(x) {
    sapply(x, function(x) {
      integrate(function(y) cov_xy(x, y), 0, 1)$value
    })
  }, 0, 1
)$value

print(paste("Cov(X, Y) =", fractions(covariance)))
```

R code Exa 4.6.3 Test Scores II

```r
#Page 251

library(MASS)

cov_XY <- 1 / 144
var_X <- 11 / 144
var_Y <- 11 / 144

rho_XY <- cov_XY / sqrt(var_X * var_Y)

print(paste("  (X, Y) =", fractions(rho_XY)))
```

R code Exa 4.6.4 Uncorrelated Random Variables

```r
# Page 251

values_X <- c(-1, 0, 1)
prob_X <- rep(1/3, 3)

values_Y <- values_X^2

E_X <- sum(values_X * prob_X)
E_Y <- sum((values_X^2) * prob_X)

E_XY <- sum((values_X * values_Y) * prob_X)

cov_XY <- E_XY - (E_X * E_Y)

cat("E(X) =", E_X, "\n")
cat("E(XY) =", E_XY, "\n")
```

```
cat("Cov(X, Y) =", cov_XY, "\n")
```

R code Exa 4.6.5 Uniform Distribution Inside a Circle

```
# Page 252

library(cubature)

joint_pdf <- function(x, y) {
  if (x^2 + y^2 <= 1) {
    return(1 / (2 * pi))
  } else {
    return(0)
  }
}

integrand_X <- function(xy) {
  x <- xy[1]
  y <- xy[2]
  return(x * joint_pdf(x, y))
}


integrand_Y <- function(xy) {
  x <- xy[1]
  y <- xy[2]
  return(y * joint_pdf(x, y))
}


integrand_XY <- function(xy) {
  x <- xy[1]
  y <- xy[2]
  return(x * y * joint_pdf(x, y))
}
```

```
bounds <- cbind(c(-1, 1), c(-1, 1))

E_X <- adaptIntegrate(integrand_X, lowerLimit = bounds[,1], upperLimit = bounds[,2])$integral

E_Y <- adaptIntegrate(integrand_Y, lowerLimit = bounds[,1], upperLimit = bounds[,2])$integral

E_XY <- adaptIntegrate(integrand_XY, lowerLimit = bounds[,1], upperLimit = bounds[,2])$integral

cov_XY <- E_XY - (E_X * E_Y)

cat("E(X) =", E_X, "\n")
cat("E(Y) =", E_Y, "\n")
cat("E(XY) =", E_XY, "\n")
cat("Cov(X, Y) =", cov_XY, "\n")
```

R code Exa 4.7.2 Household Survey I

```
# Page 257

library(knitr)

joint_pmf <- matrix(c(0.040, 0.028, 0.012, 0.008, 0.008, 0.004, 0, 0,
                      0.048, 0.084, 0.100, 0.120, 0.100, 0.060, 0.020, 0.004,
                      0.004, 0.020, 0.040, 0.060, 0.080, 0.044, 0.020, 0.012,
                      0, 0.008, 0.012, 0.020, 0.020, 0.012, 0.008, 0.004),
                    nrow = 4, byrow = TRUE)

```

```
marginal_pmf_X <- colSums(joint_pmf)
marginal_pmf_Y <- rowSums(joint_pmf)


E_Y_given_X <- numeric(length = ncol(joint_pmf))
for (x in 1:length(E_Y_given_X)) {
  E_Y_given_X[x] <- sum((0:(nrow(joint_pmf) - 1)) *
     joint_pmf[,x]) / marginal_pmf_X[x]
}

df <- data.frame(x = 1:8, `E(Y|x)` = sprintf("%.3f",
    E_Y_given_X))

knitr::kable(df, col.names = c("x", "E(Y given x)"),
    align = "c")
```

R code Exa 4.7.4 Household Survey II

```
#Page 258

Z_values <- c(0.609, 1.057, 1.317, 1.442, 1.538,
   1.533, 1.75, 2)
Pr_Z_values <- c(0.092, 0.140, 0.164, 0.208, 0.208,
   0.120, 0.048, 0.020)

E_Z <- sum(Z_values * Pr_Z_values)
signif(E_Z,4)
```

R code Exa 4.7.6 Points from Uniform Distributions

```
#Page 259

library(MASS)
```

```
E_X <- 1 / 2
E_Y_given_X <- (1 / 2) * (E_X + 1)
E_Y <- (1 / 2) * (E_X + 1)
fractions(E_Y)
```

R code Exa 4.7.8 A Clinical Trial

```
#Page 261

x_values <- 0:40

sum_x_squared <- sum(x_values^2)
sum_x <- sum(x_values)
E_X_squared <- sum_x_squared / 41
E_X <- sum_x / 41

overall_MSE <- (1/((42^2)*43))*((-sum_x_squared/41)
   +(40*sum_x/41)+41)

cat("Overall M.S.E.:", signif(overall_MSE,4), "\n")
```

R code Exa 4.8.1 Choice of Gambles I

```
#Page 265

x_values <- c(500, -350)
f_x <- c(0.5, 0.5)

y_values <- c(40, 50, 60)
g_y <- c(1/3, 1/3, 1/3)

E_X <- sum(x_values * f_x)
```

```
E_Y <- sum(y_values * g_y)

cat("E(X) =", E_X, "\n")
cat("E(Y) =", E_Y, "\n")
```

R code Exa 4.8.2 Choice of Gambles II

```
#Page 265

U <- function(x) {
  if (x >= 0) {
    return(100 * log(x + 100) - 461)
  } else {
    return(x)
  }
}

x_values <- c(500, -350)
f_x <- c(1/2, 1/2)

E_U_X <- sum(f_x * sapply(x_values, U))

y_values <- c(40, 50, 60)
g_y <- c(1/3, 1/3, 1/3)

E_U_Y <- sum(g_y * sapply(y_values, U))

cat("E[U(X)] =", round(E_U_X,1), "\n")
cat("E[U(Y)] =", round(E_U_Y), "\n")
#The answer may slightly vary due to rounding off
    values.
```

R code Exa 4.8.3 Linear Utility Function

```
#Page 267

x_values <- c(-3, 2.5, 6)
f_x <- c(0.5, 0.4, 0.1)

y_values <- c(-2, 1, 3)
g_y <- c(0.3, 0.4, 0.3)

E_X <- sum(f_x * x_values)

E_Y <- sum(g_y * y_values)

cat("E(X) =", E_X, "\n")
cat("E(Y) =", E_Y, "\n")
```

R code Exa 4.8.4 Cubic Utility Function

```
#Page 267

x_values <- c(-3, 2.5, 6)
f_x <- c(0.5, 0.4, 0.1)

y_values <- c(-2, 1, 3)
g_y <- c(0.3, 0.4, 0.3)

U <- function(x) {
  return(x^3)
}

E_U_X <- sum(f_x * U(x_values))

E_U_Y <- sum(g_y * U(y_values))

```

```
U_0 <- U(0)

cat("E[U(X)] =", E_U_X, "\n")
cat("E[U(Y)] =", E_U_Y, "\n")
cat("U(0) =", U_0, "\n")
```

R code Exa 4.8.5 Logarithmic Utility Function

```
#Page 268

x_values <- c(-3, 2.5, 6)
f_x <- c(0.5, 0.4, 0.1)

y_values <- c(-2, 1, 3)
g_y <- c(0.3, 0.4, 0.3)

U <- function(x) {
  return(log(x + 4))
}

E_U_X <- sum(f_x * U(x_values))

E_U_Y <- sum(g_y * U(y_values))

U_0 <- U(0)

cat("E[U(X)] =", round(E_U_X,4), "\n")
cat("E[U(Y)] =", round(E_U_Y,4), "\n")
cat("U(0) =", round(U_0,4), "\n")
```

R code Exa 4.8.6 Quadratic Utility Function

```
#Page 268
```

```
U <- function(x) {
  return(x^2)
}

E_U_lottery <- (1/4) * U(36) + (3/4) * U(0)

print(E_U_lottery)
```

R code Exa 4.8.7 Square Root Utility Function

```
#Page 268

U <- function(x) {
  return(x^(1/2))
}

E_U_lottery <- (1/4) * U(36) + (3/4) * U(0)

print(E_U_lottery)
```

R code Exa 4.8.9 Bounding a Random Variable

```
#Page 269

prob_threshold <- 10^4.8 / (10^6 + 10^4.8)

cat("Probability threshold for Pr(P    p0):", round
   (prob_threshold,4), "\n")
```

R code Exa 4.8.10 Investment II

```r
#Page 270
U <- function(x) {
  if (x >= 0) {
    return(x^0.8)
  } else {
    return(x)
  }
}

integral1_negative <- integrate(function(r) 120 * r * (1/30), lower = -10, upper = 0)$value
integral1_positive <- integrate(function(r) (120 * r)^0.8 * (1/30), lower = 0, upper = 20)$value
E_U_120R1 <- integral1_negative + integral1_positive

integral2_negative <- integrate(function(r) 200 * r * (1/14.5), lower = -4.5, upper = 0)$value
integral2_positive <- integrate(function(r) (200 * r)^0.8 * (1/14.5), lower = 0, upper = 10)$value
E_U_200R2 <- integral2_negative + integral2_positive

cat("E[U(120R1)] =", round(E_U_120R1,1), "\n")
cat("E[U(200R2)] =", round(E_U_200R2,1), "\n")
```

Chapter 5

Special Distributions

R code Exa 5.2.7 Group Testing

```r
# Page 278

n_groups <- 10
group_size <- 100
disease_prob <- 0.002

prob_Z1_positive <- 1 - dbinom(0, size = group_size, prob = disease_prob)

cat("Pr(Z1,i > 0) =", signif(prob_Z1_positive,3), "\n")

mean_100Y1 <- group_size * n_groups * prob_Z1_positive

cat("Mean of 100Y1:", round(mean_100Y1), "\n")

Pr_Z2_i_k_gt_0 <- 1 - 0.998^n_groups

cat("Pr(Z2,i,k > 0):", signif(Pr_Z2_i_k_gt_0,3), "\n")
```

```r

mean_10Y2 <- n_groups * group_size * Pr_Z2_i_k_gt_0

cat("Mean of 10Y2:", round(mean_10Y2,2), "\n")

expected_total_tests <- 10 + 1.81 + 19.82

cat("Expected total number of tests:", expected_
   total_tests, "\n")
```

R code Exa 5.4.2 Customer Arrivals I

```r
#Page 289

lambda <- 3600 * 0.00125

cat("Mean for the Poisson distribution:", lambda, "\
   n")

n <- 3600
p <- 0.00125

lambda <- n * p

x <- 0:9

binomial_probs <- dbinom(x, n, p)

poisson_probs <- dpois(x, lambda)

results <- data.frame(
  x = x,
  Binomial = round(binomial_probs, 5),
  Poisson = round(poisson_probs, 5)
)
```

```
print(results)
```

R code Exa 5.4.3 Customer Arrivals II

```
#Page 291
X <- 4.5
Y <- 4.5
lambda <- X+Y
Pr <- 1 - ppois(11, lambda)
cat(" Pr(X      12) =", round(Pr,3))
```

R code Exa 5.4.4 Approximating a Probability

```
#Page 292

n <- 200
p <- 0.01
lambda <- n*p
Pr <- 1 - ppois(3, lambda)
cat("Pr(X      4) =", round(Pr,4),"\n")
#The answer varies due to rounding off values
```

R code Exa 5.4.6 Radioactive Particles

```
#Page 293

lambda <- 6
Pr <- 1 - ppois(9, lambda)
cat("Pr(X      10) =", round(Pr,4),"\n")
#The answer varies due to rounding off values
```

R code Exa 5.4.7 Cryptosporidium in Drinking Water

```
#Page 294

lambda <- 0.2
p <- 0.1
t <- 100

prob <- 1 - exp(-p * lambda * t)

round(prob,2)
```

R code Exa 5.5.3 Triples in the Lottery I

```
#Page 298

No_of_triples <- 10
No_of_daily_numbers <- 1000
p <- No_of_triples / No_of_daily_numbers

print(p)
```

R code Exa 5.5.4 Triples in the Lottery II

```
#Page 300

p <- 0.01

E_X <- (1 - p) / p
```

```

expected_days <- E_X + 1

print(expected_days)
```

R code Exa 5.6.4 Probabilities for a Normal Distribution

```
#Page 308

mean_X <- 5
sd_X <- 2
lower_bound_X <- 1
upper_bound_X <- 8

lower_bound_Z <- (lower_bound_X - mean_X) / sd_X
upper_bound_Z <- (upper_bound_X - mean_X) / sd_X

P_Z_lower <- pnorm(lower_bound_Z)
P_Z_upper <- pnorm(upper_bound_Z)

P_X <- P_Z_upper - P_Z_lower

round(P_X,4)
#The answer varies due to rounding off values
```

R code Exa 5.6.5 Quantiles of Normal Distributions

```
#Page 308

mean_X <- 1.329
sd_X <- 0.4844

q_05 <- qnorm(0.05)
```

```

quantile_05_X <- mean_X + sd_X * q_05

round(quantile_05_X,4)
```

R code Exa 5.6.6 Heights of Men and Women

```
# Page 310

mean_W <- 65
mean_M <- 68
var_W <- 1
var_M <- 9

mean_diff <- mean_W - mean_M

var_diff <- var_W + var_M

sd_diff <- sqrt(var_diff)

z <- mean_diff / sd_diff

prob <- pnorm(z)

round(prob,3)
```

R code Exa 5.6.7 Determining a Sample Size

```
# Page 311

z_value <- 1.96
std_dev <- 3

```

```
min_n <- (z_value * std_dev)^2

min_n_rounded <- ceiling(min_n)

print(min_n_rounded)
```

R code Exa 5.6.8 Interval for Mean

```
#Page 311

Xn <- 68

n <- 36

std_dev <- 3

z <- 1.96

margin_error <- z * (std_dev / sqrt(n))

cat("Margin of Error:", margin_error, "\n")
```

R code Exa 5.6.9 Failure Times of Ball Bearings

```
#Page 312

library(ggplot2)

mu <- 4.15
sigma <- 0.53342
p <- 0.9
set.seed(123)
samples <- rlnorm(1000, meanlog = mu, sdlog = sigma)
```

```
data_df <- data.frame(data = samples)

x_seq <- seq(min(samples), max(samples), length.out
    = 1000)

y_seq <- dlnorm(x_seq, meanlog = mu, sdlog = sigma)

ggplot(data = data_df, aes(x = data)) +
  geom_histogram(aes(y=..density..), bins = 30,
      color="black", fill="white") +
  geom_line(data = data.frame(x = x_seq, y = y_seq),
       aes(x = x, y = y), color = "blue", size = 1) +
  labs(x = "Time to Failure", y = "Density") +
  theme_minimal()

q <- exp(mu + sigma * qnorm(p))
signif(q, 5)
```

R code Exa 5.6.10 Stock and Option Prices

```
#Page 313

library(qrmtools)

S0 <- 100
r <- 0.06
sigma <- 0.1
K <- 100
T <- 1
t <- 0

option_price <- Black_Scholes(t, S0, r, sigma, K, T,
     type = "call")

cat("Option price using Black-Scholes formula:",
```

```
    round(option_price,2)/100, "* S0\n")
```

R code Exa 5.7.2 Mean and Variance of Lifetime of a Light Bulb

```
#Page 317

f1 <- function(x) { x * exp(-x) }
f2 <- function(x) { x^2 * exp(-x) }

integral1 <- integrate(f1, 0, Inf)$value
integral2 <- integrate(f2, 0, Inf)$value


mean_lifetime <- integral1
variance_lifetime <- integral2 - integral1^2

cat("Mean of the lifetime:", mean_lifetime, "\n")
cat("Variance of the lifetime:", variance_lifetime,
    "\n")
```

R code Exa 5.8.4 Castaneda v Partida

```
#Page 330

library(ggplot2)

beta_values <- seq(0, 100, by = 1)

probabilities <- numeric(length(beta_values))

threshold <- 0.6328

for (i in seq_along(beta_values)) {
```

```
  beta <- beta_values[i]
  alpha <- 3.785 * beta
  alpha_new <- alpha + 100
  beta_new <- beta + 120
  probabilities[i] <- pbeta(threshold, alpha_new, beta_new)
}

data <- data.frame(beta = beta_values, probability = probabilities)

y_max <- max(probabilities)

ggplot(data, aes(x = beta, y = probability)) +
  geom_line(color = "blue") +
  geom_segment(aes(x = 51.5, xend = 51.5, y = 0, yend = pmin(probability[52], y_max)),
               linetype = "dashed", color = "red") +
  geom_segment(aes(x = 0, xend = 51.5, y = 0.5, yend = 0.5),
               linetype = "dashed", color = "red") +
  labs(title = expression(Pr(P <= 0.6328 | X == 100) ~ "for various values of" ~ beta),
       x = expression(beta),
       y = expression(Pr(P <= 0.6328 | X == 100))) +
  annotate("text", x = 51.5, y = 0.05, label = "  = 51.5", color = "red", angle = 90, vjust = -0.5) +
  annotate("text", x = 20, y = 0.5, label = "Pr(P  0.6328 | X = 100) = 0.5", color = "red", vjust = -0.5) +
  theme_minimal()
```

R code Exa 5.8.5 A Clinical Trial

```
#Page 330

alpha <- 1.18
beta <- 2.35

n <- 40
x <- 22

alpha_new <- alpha + x
beta_new <- beta + (n - x)

expected_p <- alpha_new / (alpha_new + beta_new)

cat("Expected value E(P|X = 22):", round(expected_p
    ,4), "\n")
```

R code Exa 5.9.2 Attendance at a Baseball Game

```
#Page 334

library(iterpc)

p1 <- 0.23
p2 <- 0.59
p3 <- 0.18

n1 <- 7
n2 <- 8
n3 <- 5

n <- n1 + n2 + n3

probability <- multichoose(c(n1,n2,n3)) * (p1^n1) *
    (p2^n2) * (p3^n3)

```

```r
cat("The probability is:", round(probability,4), "\n
   ")
```

R code Exa 5.9.3 Blood Types

```r
#Page 334

p_A <- 0.360
p_B <- 0.123
p_AB <- 0.038
p_O <- 0.479

probability_same_blood_type <- p_A^2 + p_B^2 + p_AB
   ^2 + p_O^2

cat("The probability that two randomly selected
   individuals have the same blood type is:", round(
   probability_same_blood_type,3), "\n")
```

R code Exa 5.10.6 Heights of Husbands and Wives

```r
#Page 343

mean_wife <- 66.8
sd_wife <- 2
mean_husband <- 70
sd_husband <- 2
correlation <- 0.68

mean_diff <- mean_wife - mean_husband
var_diff <- sd_wife^2 + sd_husband^2 - 2 *
   correlation * sd_wife * sd_husband

```

```
sd_diff <- sqrt(var_diff)

probability <- 1 - pnorm(0, mean_diff, sd_diff)

cat("Mean difference:", mean_diff, "\n")
cat("Variance of the difference:", var_diff, "\n")
cat("Standard deviation of the difference:", sd_diff, "\n")

cat("The probability that the wife will be taller than her husband is:", round(probability,5), "\n")
```

Chapter 6

Large Random Samples

R code Exa 6.1.1 Proportion of Heads

```
# Page 347

n1 <- 10

p <- 0.5

k1 <- 5

probability_10_flips <- dbinom(k1, n1, p)

n2 <- 100

k2 <- 50

probability_100_flips <- dbinom(k2, n2, p)

probability_10_flips_range <- pbinom(6, n1, p) -
   pbinom(3, n1, p)

probability_100_flips_range <- pbinom(60, n2, p) -
   pbinom(39, n2, p)
```

```
cat("The probability of getting exactly", k1, "heads in", n1, "flips is:", round(probability_10_flips,4) , "\n")
cat("The probability of getting exactly", k2, "heads in", n2, "flips is:",round(probability_100_flips,4) , "\n")

cat("The probability that the proportion of heads is within 0.1 of 1/2 for n = 100 is:", round(probability_100_flips_range,4), "\n")
cat("The probability that the proportion of heads is within 0.1 of 1/2 for n = 10 is:", round(probability_10_flips_range,5), "\n")
```

R code Exa 6.2.1 Required Number of Observations

```
# Page 350

sigma_squared <- 4
epsilon <- 1
confidence_level <- 0.99

required_sample_size <- sigma_squared / (1 - confidence_level)

cat("The required sample size is at least:", required_sample_size, "\n")
```

R code Exa 6.2.2 A Simulation

```
#Page 351

```

```r
epsilon <- 0.005
confidence_level <- 0.98
required_samples <- 1 / (epsilon^2 * (1 - confidence_level))

cat("Required number of samples for the desired precision:", format(required_samples,scientific = FALSE), "\n")
```

R code Exa 6.2.3 Tossing a Coin

```r
#Page 351

n <- 15
p <- 0.5

lower_bound <- ceiling(0.4 * n)
upper_bound <- floor(0.6 * n)

prob <- sum(dbinom(lower_bound:upper_bound, size = n, prob = p))

cat("Pr(0.4 <= Xn <= 0.6) for n =", n, "is", round(prob,2), "\n")
```

R code Exa 6.2.4 Rate of Service I

```r
#Page 354

set.seed(123)

lambda <- 0.446
n <- 100
```

```
sample <- rexp(n, rate = lambda)
sample <- sample[sample <= 10]

hist(sample, breaks = 10, probability = TRUE, main = "Histogram of Service Times with Exponential PDF",
     xlab = "Service Time", ylab = "Density", col = "lightblue", border = "black", xlim = c(0, 10))

curve(dexp(x, rate = lambda), from = 0.001, to = 10, add = TRUE, col = "red", lwd = 2)
```

R code Exa 6.2.5 Rate of Service II

```
#Page 354

library(ggplot2)

set.seed(123)

lambda <- 0.446
n <- 100

sample <- rexp(n, rate = lambda)
sample <- sample[sample <= 10]

data <- data.frame(service_time = sample)

ggplot(data, aes(x = service_time)) +
  geom_histogram(aes(y = ..density..), bins = 20, fill = "lightblue", color = "black", boundary = 0.5) +
  stat_function(fun = function(x) dexp(x, rate =
```

```
    lambda), color = "red", size = 1) +
  labs(title = "Histogram of Service Times with
    Exponential PDF",
       x = "Service Time", y = "Density") +
  xlim(0, 10)
```

R code Exa 6.2.6 Binomial Random Variable

```
# Page 356

bound_fn <- function(n) {
  2 * (0.9811 ^ n)
}

chebyshev_bound <- function(n) {
  25 / n
}

n_values <- c(100, 200, 300)
for (n in n_values) {
  bound_val <- bound_fn(n)
  chebyshev_bound_val <- chebyshev_bound(n)
  cat("For n =", n, "\n")
  cat("Bound value:", round(bound_val,4), "\n")
  cat("Chebyshev bound:", round(chebyshev_bound_val
    ,4), "\n\n")
}
```

R code Exa 6.3.1 A Large Sample I

```
#Page 360

n <- 100
```

```
p <- 0.5

x <- 0:n
binomial_pmf <- dbinom(x, size = n, prob = p)

mu <- n * p
sigma <- sqrt(n * p * (1 - p))

x_norm <- seq(0, n, length.out = 1000)
normal_pdf <- dnorm(x_norm, mean = mu, sd = sigma)

plot(x, binomial_pmf, type = "h", lwd = 10, col = "skyblue", main = "Binomial Distribution vs Normal Approximation",
     xlab = "Number of Patients Surviving", ylab = "Probability Mass Function")
lines(x_norm, normal_pdf, col = "red", lty = 2, lwd = 2)
legend("topright", legend = c("Binomial PMF", "Normal Approximation"), col = c("skyblue", "red"), lty = c(1, 2), lwd = c(10, 2))
```

R code Exa 6.3.2 Tossing a Coin

```
#Page 362

n <- 900
p <- 0.5
mu <- n * p
sigma <- sqrt(n * p * (1 - p))

z_score <- (495 - mu) / sigma

prob_more_than_495 <- 1 - pnorm(z_score)

```

```
round(prob_more_than_495,4)
```

R code Exa 6.3.3 Sampling from a Uniform Distribution

```
#Page 362

n <- 12
mu <- 1/2
sigma <- 1/12

z <- 0.1 / sigma

prob_within_01 <- 2 * pnorm(z) - 1

round(prob_within_01,5)
```

R code Exa 6.3.5 Determining a Simulation Size

```
#Page 363

n <- (1/(0.005^2)) * round(qnorm(0.99),3)^2

print(n)

n_adjusted <- ceiling(n * 1/4)

print(n_adjusted)
```

R code Exa 6.3.9 Examination Questions I

```r
#Page 367

n <- 99
pass_threshold <- 60

p <- 1 - (1:n)/100

mean_correct <- sum(p)
var_correct <- sum(p * (1 - p))

cat("Mean:", mean_correct, "\n")
cat("Variance:", var_correct, "\n")


pass_prob_clt <- 1 - pnorm((pass_threshold - mean_
    correct) / sqrt(var_correct))

round(pass_prob_clt,3)
```

R code Exa 6.4.1 A Large Sample II

```r
#Page 371

library(ggplot2)

n <- 100
p <- 0.5
mu <- n * p
sigma <- sqrt(n * p * (1 - p))

x <- seq(30, 70, by = 0.1)

binom_cdf <- pbinom(x, size = n, prob = p)

norm_cdf <- pnorm(x, mean = mu, sd = sigma)
```

```
data <- data.frame(
  x = x,
  Binomial = binom_cdf,
  Normal = norm_cdf
)

# Plot the CDFs
ggplot(data, aes(x = x)) +
  geom_line(aes(y = Binomial, color = "Binomial")) +
  geom_line(aes(y = Normal, color = "Normal")) +
  labs(
    title = "CDFs of Binomial and Normal
       Distributions",
    x = "x",
    y = "CDF",
    color = "Distribution"
  ) +
  theme_minimal()
```

R code Exa 6.4.2 A Large Sample III

```
#Page 373

n <- 100
p <- 0.5
mean_binomial <- n * p
var_binomial <- n * p * (1 - p)

mean_normal <- mean_binomial
sd_normal <- sqrt(var_binomial)
normal_dist <- rnorm(10000, mean_normal, sd_normal)

binomial_dist <- rbinom(10000, n, p)

```

```
x_values <- seq(30, 69, by = 1)

normal_cdf <- pnorm(x_values, mean_normal, sd_normal
  )
binomial_cdf <- pbinom(x_values, n, p)

plot(x_values, normal_cdf, type = "l", col = "blue",
   xlab = "x", ylab = "Pr(X <= x)", main = "
  Comparison of Normal and Binomial CDFs")
lines(x_values, binomial_cdf, type = "l", col = "red
  ")
legend("bottomright", legend = c("Normal
  Approximation", "Binomial Distribution"), col = c
  ("blue", "red"), lty = 1)
```

R code Exa 6.4.3 Examination Questions II

```
#Page 373

mean_X <- 49.5
sd_X <- 4.08

prob <- 1 - pnorm(59.5, mean_X, sd_X)

round(prob,3)
```

R code Exa 6.4.4 Coin Tossing

```
#Page 374

mean_X <- 10
sd_X <- sqrt(20 * 0.5 * 0.5)

```

```
prob <- pnorm(10.5, mean_X, sd_X) - pnorm(9.5, mean_X, sd_X)

round(prob,3)
```

Chapter 7

Estimation

R code Exa 7.2.6 Lifetimes of Fluorescent Lamps I

```
#Page 388

library(ggplot2)
library(gridExtra)

gamma_density <- function(x, shape, rate) {
  dgamma(x, shape, rate)
}

n <- 5
lifetimes <- c(2911, 3403, 3237, 3509, 3118)
y <- sum(lifetimes)
prior_shape1 <- 4
prior_rate1 <- 20000
posterior_shape1 <- n + prior_shape1
posterior_rate1 <- y + prior_rate1

prior_shape2 <- 1
prior_rate2 <- 1000
posterior_shape2 <- n + prior_shape2
posterior_rate2 <- y + prior_rate2
```

```

theta <- seq(0, 0.0015, length.out = 1000)

prior_density1 <- gamma_density(theta, prior_shape1, prior_rate1)
posterior_density1 <- gamma_density(theta, posterior_shape1, posterior_rate1)
prior_density2 <- gamma_density(theta, prior_shape2, prior_rate2)
posterior_density2 <- gamma_density(theta, posterior_shape2, posterior_rate2)

data1 <- data.frame(
  theta = theta,
  Prior = prior_density1,
  Posterior = posterior_density1
)

data2 <- data.frame(
  theta = theta,
  Prior = prior_density2,
  Posterior = posterior_density2
)

p1 <- ggplot(data1, aes(x = theta)) +
  geom_line(aes(y = Prior, color = "Prior")) +
  geom_line(aes(y = Posterior, color = "Posterior")) +
  labs(
    title = "Gamma Distributions: Prior and Posterior (First Set of Parameters)",
    x = expression(theta),
    y = "Density",
    color = "Distribution"
  ) +
  theme_minimal()

p2 <- ggplot(data2, aes(x = theta)) +
```

```
  geom_line(aes(y = Prior, color = "Prior")) +
  geom_line(aes(y = Posterior, color = "Posterior"))
    +
  labs(
    title = "Gamma Distributions: Prior and
       Posterior (Second Set of Parameters)",
    x = expression(theta),
    y = "Density",
    color = "Distribution"
  ) +
  theme_minimal()

grid.arrange(p1, p2, nrow = 2)
```

R code Exa 7.2.8 Lifetimes of Fluorescent Lamps II

```
#Page 392

k <- 9.555 * (10^41)

f_x6_given_x <- function(x6, k) {
  k / (x6 + 36178)^10
}

prob_greater_than_3000 <- integrate(function(x6) f_
   x6_given_x(x6, k), lower = 3000, upper = Inf)$
   value

round(prob_greater_than_3000,4)

k_new <- 1.542 * (10^26)

f_x6_given_x_new <- function(x6, k_new) {
  k_new / (x6 + 17178)^7
}
```

```
prob_greater_than_3000_new <- integrate(function(x6)
    f_x6_given_x_new(x6, k_new), lower = 3000, upper
    = Inf)$value

round(prob_greater_than_3000_new,5)
```

R code Exa 7.3.3 Glove Use by Nurses

```
#Page 396

observed_before = 13
total_before = 51
observed_after = 50
total_after = 56

prior1_alpha = 1
prior1_beta = 1

prior2_alpha = 13
prior2_beta = 38

posterior1_alpha = prior1_alpha + observed_after
posterior1_beta = prior1_beta + (total_after -
   observed_after)

posterior2_alpha = prior2_alpha + observed_after
posterior2_beta = prior2_beta + (total_after -
   observed_after)

posterior1_mean = posterior1_alpha / (posterior1_
   alpha + posterior1_beta)
round(posterior1_mean,2)
prob_posterior1_gt_before = 1 - pbeta(2 * observed_
   before / total_before, posterior1_alpha,
```

```
posterior1_beta)
prob_posterior1_gt_before

posterior2_mean = posterior2_alpha / (posterior2_alpha + posterior2_beta)
round(posterior2_mean,2)
prob_posterior2_gt_before = 1 - pbeta(2 * observed_before / total_before, posterior2_alpha, posterior2_beta)
round(prob_posterior2_gt_before,2)
```

R code Exa 7.3.8 Automobile Emissions

```
#Page 399

n <- 46
sigma_sq <- 0.25
mu_0 <- 2
v_2 <- 1.0
x_n <- 1.329

mu_1 <- (sigma_sq * mu_0 + n * v_2 * x_n) / (sigma_sq + n * v_2)
round(mu_1,3)

v_1 <- sigma_sq * v_2 / (sigma_sq + n * v_2)
round(v_1,4)
```

R code Exa 7.3.10 Calorie Counts on Food Labels

```
#Page 400

n <- 20
```

```r
theta_prior_mean <- 0
theta_prior_var <- 60
theta_likelihood_var <- 100
x_bar <- 0.125

theta_posterior_mean <- (theta_likelihood_var * theta_prior_mean + n * theta_prior_var * x_bar) / (theta_likelihood_var + n * theta_prior_var)
round(theta_posterior_mean,4)

theta_posterior_var <- theta_likelihood_var * theta_prior_var / (theta_likelihood_var + n * theta_prior_var)
round(theta_posterior_var,2)

prob_theta_gt_1 <- 1 - pnorm(1, mean = theta_posterior_mean, sd = sqrt(theta_posterior_var))
round(prob_theta_gt_1,4)
```

R code Exa 7.3.12 Lifetimes of Electronic Components I

```r
#Page 402

n <- 3
alpha <- 1
beta <- 2
y <- sum(c(3, 1.5, 2.1))
print(y)

alpha_posterior <- alpha + n
print(alpha_posterior)

beta_posterior <- beta + y
print(beta_posterior)
```

R code Exa 7.4.5 Estimating the Parameter of a Bernoulli Distribution

```
#Page 411

set.seed(1)

alpha <- 22
beta <- 18

samples <- rbeta(10^6, alpha, beta)

mean_value <- mean(samples)
round(mean_value,2)

median_value <- quantile(samples, 0.5)
round(median_value,4)
```

R code Exa 7.4.8 Lifetimes of Electronic Components II

```
#Page 414

integrand <- function(theta) {
  ((8.6^4)/6) * (theta^2) * exp(-8.6 * theta)
}

bayes_estimate <- integrate(integrand, lower = 0,
   upper = Inf)$value

round(bayes_estimate,3)

alpha <- 8.6
beta <- 4
```

```

mean_1_theta <- alpha / beta

print(mean_1_theta)
```

R code Exa 7.5.2 Lifetimes of Electronic Components III

```
#Page 419

log_likelihood <- function(theta) {
  3 * log(theta) - 6.6 * theta
}

mle_result <- optimize(log_likelihood, interval = c
   (0.01, 10), maximum = TRUE)

mle_theta <- mle_result$maximum

round(mle_theta,3)
```

R code Exa 7.5.3 Test for a Disease

```
#Page 419

mle_theta <- function(x) {
  if (x == 0) {
    return(0.1)
  } else if (x == 1) {
    return(0.9)
  } else {
    stop("Invalid value for x. Must be 0 or 1.")
  }
}
```

```
X <- 0

theta_hat <- mle_theta(X)

cat("When x=0, theta which maximizes the likelihood
   is",theta_hat,"\n")

X <- 1

theta_hat <- mle_theta(X)

cat("When x=1, theta which maximizes the likelihood
   is",theta_hat,"\n")
```

R code Exa 7.6.2 Lifetimes of Electronic Components IV

```
#Page 427

theta_hat <- 0.455

psi_hat <- 1 / theta_hat

round(psi_hat,1)
```

R code Exa 7.6.6 Sampling from a Gamma Distribution

```
#Page 430

alpha_0 <- 3.679

alpha_1 <- alpha_0 - (round(digamma(alpha_0),4) -
   1.220) / round(trigamma(alpha_0),4)
```

```
round(alpha_1,3)

alpha_2 <- alpha_1 - (digamma(alpha_1) - 1.220) /
    trigamma(alpha_1)
round(alpha_2,3)

alpha_3 <- alpha_2 - (digamma(alpha_2) - 1.220) /
    trigamma(alpha_2)
round(alpha_3,4)
#The answer provided in the textbook is wrong.
```

R code Exa 7.6.10 Sampling from a Uniform Distribution

```
#Page 431

X <- c(0.2, 0.99, 0.01)
n <- length(X)

X_bar <- mean(X)
S_squared <- var(X)

theta_mom <- (X_bar - sqrt(n / (n-1)) * sqrt(S_
    squared))/2

cat("X_bar:", X_bar, "\n")
cat("Method of Moments estimate (theta):", round(
    theta_mom,1), "\n")
```

R code Exa 7.6.12 Prussian Army Deaths

```
#Page 432

library(MASS)
```

```

alpha <- 196
beta <- 280
n <- 280

mle_theta <- alpha / beta

mle_variance_theoretical <- mle_theta / n

fractions(mle_theta)
# The answer may vary due to difference in
    representation.
```

R code Exa 7.6.13 Lifetimes of Electronic Components V

```
#Page 433

library(MASS)
n <- 3
alpha <- 4
beta <- 8.6

X <- c(3, 1.5, 2.1)
mle_theta <- n / sum(X)

fractions(mle_theta)
#The answer may vary due to difference in
    representation.
```

R code Exa 7.6.15 Heights and Weights

```
#Page 435

```

```
mu1 <- 69.60
mu2 <- 194.75
sigma1_sq <- 2.87
sigma2_sq <- 14.82
rho <- 0.1764

heights <- c(72, 70, 73, 68, 65, NA)
weights <- c(197, 204, 208, NA, NA, 170)

conditional_mean_X4_2 <- mu2 + rho * sqrt(sigma2_sq)
    * ((heights[4] - mu1)/sqrt(sigma1_sq))
conditional_mean_X4_2

conditional_variance_X4_2 <- (1 - rho^2) * sigma2_sq
   ^2
round(conditional_variance_X4_2,1)
#The answer provided in the textbook is wrong.
```

R code Exa 7.6.16 Mixture of Normal Distributions

```
#Page 736

mu_1 <- -7.65
mu_2 <- 7.36
sigma2_0 <- 46.28
p_0 <- 0.5

data <- c(-4.0, 9.0, -5.5, 10.2, -6.8, 11.5, -8.3,
   13.0, -7.2, 12.3, 2.4, 12.8, 3.8, 13.1, 1.7,
   14.6, 0.2, 14.9, -2.3, 15.4)

q_0_10 <- p_0 * exp(-(data[1] + -mu_1)^2 / (2 *
   sigma2_0)) /
  (p_0 * exp(-(data[1] + -mu_1)^2 / (2 * sigma2_0))
     +
```

```
      p_0 * exp(-(data[1] - mu_2)^2 / (2 * sigma2_0)))
round(q_0_10,4)

q_0_8 <- p_0 * exp(-(data[2] + -mu_1)^2 / (2 * sigma2_0)) /
  (p_0 * exp(-(data[2] + -mu_1)^2 / (2 * sigma2_0)) +
      p_0 * exp(-(data[2] - mu_2)^2 / (2 * sigma2_0)))
round(q_0_8,4)
```

Chapter 8

Sampling Distributions of Estimators

R code Exa 8.1.5 Lifetimes of Electronic Components I

```
#Page 468

library(gamlss.dist)

cdf_gamma <- function(x) {
  pgamma(x, shape = 3, scale = 1)
}

prob <- cdf_gamma(3.33) - cdf_gamma(2.73)
round(prob,3)
```

R code Exa 8.1.6 A Clinical Trial

```
#Page 468

library(ggplot2)
```

```r

f_t_given_theta <- function(t, theta) {
  return(choose(40, 40 * t) * theta^(40 * t) * (1 - theta)^(40 * (1 - t)))
}

probability_T_theta <- function(theta) {
  sum(sapply(seq(0, 1, by = 1/40), function(t) {
    if (abs(t - theta) < 0.1) {
      return(f_t_given_theta(t, theta))
    } else {
      return(0)
    }
  }))
}

theta_values <- seq(0, 1, by = 0.01)

prob_values <- sapply(theta_values, probability_T_theta)

data <- data.frame(theta = theta_values, probability = prob_values)

ggplot(data, aes(x = theta, y = probability)) +
  geom_line(color = "blue") +
  labs(title = expression(Pr("|T - " * theta * "| < 0.1 | " * theta)),
       x = expression(theta),
       y = "Probability") +
  theme_minimal()

theta_values <- seq(0, 1, by = 1/40)

prob_values <- sapply(theta_values, probability_T_theta)

cat("Minimum value",round(min(prob_values),4), "\n")
```

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

R code Exa 8.2.3 Acid Concentration in Cheese I

```
#Page 471

library(stats)

prob <- pchisq(10, df = 10)
round(prob,2)
```

R code Exa 8.3.2 Rain from Seeded Clouds I

```
#Page 474

df <- 25

critical_value <- qchisq(0.25, df)

prob_underestimate <- pchisq(critical_value, df,
   lower.tail = TRUE)

prob_underestimate
```

R code Exa 8.4.3 Rain from Seeded Clouds II

```
#Page 482

n <- 26
```

```
quantile <- 1.316 / sqrt(n)

prob <- 0.9

mu <- 0
sigma <- 1

upper_limit <- mu + quantile * sigma

cat("With probability", prob, "the value of Xn will
   be no more than", round(upper_limit,4), "times
   sigma above mu.")
```

R code Exa 8.5.2 Rain from Seeded Clouds III

```
#Page 486

n <- 26

gamma <- 0.95

alpha <- (1 - gamma) / 2
quantile <- qt(1 - alpha, df = n - 1)

const <- quantile / sqrt(n)

cat("For n =", n, "and gamma =", gamma, ", the
   quantile is", round(quantile,3), "and the
   constant is", round(const,3))
```

R code Exa 8.5.3 Rain from Seeded Clouds IV

```
#Page 487

n <- 26

x_bar <- 5.134
sigma <- 1.600
gamma <- 0.95

alpha <- (1 - gamma) / 2
quantile <- qt(1 - alpha, df = n - 1)
const <- quantile / sqrt(n)

a <- x_bar - const * sigma
b <- x_bar + const * sigma

cat("The observed value of x_bar is", x_bar, "\n")
cat("The observed value of sigma is", sigma, "\n")
cat("The observed value of a is", round(a,3), "\n")
cat("The observed value of b is", round(b,2), "\n")
```

R code Exa 8.5.4 Acid Concentration in Cheese II

```
#Page 487

n <- 10

gamma <- 0.9

alpha <- (1 - gamma) / 2
quantile <- qt(1 - alpha, df = n - 1)

lactic_acid <- c(0.86, 1.53, 1.57, 1.81, 0.99, 1.09,
    1.29, 1.78, 1.29, 1.58)

x_bar <- mean(lactic_acid)
```

```
sigma <- sd(lactic_acid)

lower_bound <- x_bar - quantile * sigma / sqrt(n)
upper_bound <- x_bar + quantile * sigma / sqrt(n)

cat("The observed value of x_bar is", round(x_bar,4)
    , "\n")
cat("The observed value of sigma is", round(sigma,4)
    , "\n")
cat("The 90% confidence interval for mu is (", round
    (lower_bound,3), ",", round(upper_bound,3), ")")
```

R code Exa 8.5.6 Rain from Seeded Clouds V

```
#Page 489

n <- 26

gamma <- 0.9

quantile <- qt(gamma, df = n - 1)

x_bar <- 5.134
sigma <- 1.600

lower_limit <- x_bar - quantile * sigma / sqrt(n)

cat("The observed value of x_bar is", x_bar, "\n")
cat("The observed value of sigma is", sigma, "\n")
cat("The 90% lower confidence limit for mu is",
    round(lower_limit,3),"\n")
# The answer provided in the textbook is wrong.
```

R code Exa 8.5.7 Lifetimes of Electronic Components II

```
#Page 489

n <- 3

gamma <- 0.98

quantile <- qgamma(gamma, shape = n, scale = 1)

round(quantile,4)
```

R code Exa 8.5.10 Approximate Confidence Interval for Poisson Mean

```
# Page 490

c <- 0.196
n <- 100

alpha <- 2 * pnorm(c * sqrt(n)) - 1

round(alpha,2)
```

R code Exa 8.5.11 Uniforms on an Interval of Length One

```
#Page 492

z <- 0.1
c <- 0.3

conditional_prob <- ifelse(c <= (1 - z) / 2, 2 * c /
    (1 - z), 1)

```

```
round(conditional_prob,4)
```

R code Exa 8.6.2 Acid Concentration in Cheese III

```
#Page 498

mu0 <- 1
lambda0 <- 1
alpha0 <- 0.5
beta0 <- 0.5

n <- 10
x_bar <- 1.379
s2_n <- 0.9663

mu1 <- (mu0 * lambda0 + n * x_bar) / (lambda0 + n)
lambda1 <- lambda0 + n
alpha1 <- alpha0 + n / 2
beta1 <- beta0 + (1 / 2) * (s2_n + (n * lambda0 * (x_bar - mu0)^2) / (lambda0 + n))

cat("Posterior hyperparameters:\n")
cat("mu1 =", round(mu1,3), "\n")
cat("lambda1 =", lambda1, "\n")
cat("alpha1 =", alpha1, "\n")
cat("beta1 =", round(beta1,4), "\n")

sigma_threshold <- 0.3
tau_threshold <- 1 / sigma_threshold^2
posterior_prob <- pgamma(tau_threshold, shape = alpha1, rate = beta1, lower.tail = TRUE)

cat("\nPosterior probability that sigma > 0.3 is:", round(posterior_prob,3))
```

R code Exa 8.6.3 Nursing Homes in New Mexico

```
#Page 500

x <- c(128, 281, 291, 238, 155, 148, 154, 232, 316,
   96, 146, 151, 100, 213, 208, 157, 48, 217)
n <- length(x)
xn <- mean(x)
sn2 <- var(x)

mu0 <- 200
lambda0 <- 2
alpha0 <- 2
beta0 <- 6300

mu1 <- (lambda0 * mu0 + n * xn) / (lambda0 + n)
lambda1 <- lambda0 + n
alpha1 <- alpha0 + n / 2
beta1 <- beta0 + 0.5 * sum((x - xn)^2) + (lambda0 *
   n * (xn - mu0)^2) / (2 * (lambda0 + n))

mu_posterior <- mu1
mu_posterior

var_mu_posterior <- beta1 / (lambda1 * (alpha1 - 1))
round(var_mu_posterior,2)

tau_posterior <- alpha1 / beta1
signif(tau_posterior,3)

var_tau_posterior <- alpha1 / (beta1^2)
signif(var_tau_posterior,3)
```

R code Exa 8.6.4 An Improper Prior for Seeded Cloud Rainfall

```
#Page 503

mu_hat <- 5.134
quantile_t <- 2.060
scaling_factor <- 3.188

a <- mu_hat - quantile_t / scaling_factor
b <- mu_hat + quantile_t / scaling_factor

interval <- c(a, b)
round(interval,3)

mu_0 <- 4
df <- 25

U_value <- scaling_factor * (mu_0 - mu_hat)

probability <- 1 - pt(U_value, df)
round(probability,4)
```

R code Exa 8.7.3 Lifetimes of Electronic Components III

```
# Page 507

theta_values <- seq(0, 2, by = 0.1)

mse_bayes <- function(theta) {
  if (theta <= 3.1) {
    return(4 / (2 + 1) * (theta^2 / (2 + 1)))
  } else {
```

```r
    return((4 / (2 + 1))^2 + theta^2 / (2 + 1) * (2 / (2 + 1))^2)
  }
}

mse_mle <- function(theta) {
  return((9 * theta^2 / 4) + (theta^2 / 4))
}

mse_unbiased <- function(theta) {
  return(theta^2)
}

mse_1_over_t <- function(theta) {
  return(theta^2 / 4)
}

mse_bayes_values <- sapply(theta_values, mse_bayes)
mse_mle_values <- mse_mle(theta_values)
mse_unbiased_values <- mse_unbiased(theta_values)
mse_1_over_t_values <- mse_1_over_t(theta_values)

plot(theta_values, mse_bayes_values, type = "l", col = "purple", ylim = c(0, max(mse_mle_values)), ylab = "MSE", xlab = "Theta", lwd = 2, main = "MSE of Different Estimators")
lines(theta_values, mse_mle_values, col = "red", lwd = 2)
lines(theta_values, mse_unbiased_values, col = "green", lwd = 2)
lines(theta_values, mse_1_over_t_values, col = "blue", lwd = 2)
legend("topleft", legend = c("Bayes", "MLE", "Unbiased", "1/T"), col = c("blue", "red", "green", "purple"), lty = 1, lwd = 2)
```

R code Exa 8.8.12 The Posterior Distribution of the Standard Deviation

```r
#Page 524

library(ggplot2)

set.seed(42)

n <- 40
sigma_hat <- 1.061

posterior_mean <- sigma_hat
posterior_variance <- sigma_hat^2 / (2 * n)

cat("Posterior mean:", posterior_mean, "\n")
cat("Posterior variance:", round(posterior_variance,4), "\n")

true_sigma <- 1

sample_data <- rnorm(n, mean = 0, sd = true_sigma)

sigma_hat <- sqrt(sum(sample_data^2) / n)

posterior_mean <- sigma_hat
posterior_variance <- sigma_hat^2 / (2 * n)

actual_posterior_pdf <- function(sigma) {
  if (sigma > 0) {
    posterior_value <- (sigma^(-n-1)) * exp(-sum(sample_data^2) / (2 * sigma^2))
    return(posterior_value)
  } else {
    return(0)
```

```
  }
}

approx_posterior_pdf <- function(sigma) {
  dnorm(sigma, mean = posterior_mean, sd = sqrt(posterior_variance))
}

sigma_values <- seq(0.5, 2, by = 0.01)

actual_posterior_values <- sapply(sigma_values, actual_posterior_pdf)
approx_posterior_values <- sapply(sigma_values, approx_posterior_pdf)

actual_posterior_values <- actual_posterior_values / sum(actual_posterior_values) / (sigma_values[2] - sigma_values[1])

plot_data <- data.frame(
  sigma = sigma_values,
  actual_posterior = actual_posterior_values,
  approx_posterior = approx_posterior_values
)

ggplot(plot_data, aes(x = sigma)) +
  geom_line(aes(y = actual_posterior, color = "Actual Posterior")) +
  geom_line(aes(y = approx_posterior, color = "Approximate Posterior")) +
  labs(title = "Posterior and Approximate Posterior p.d.f.",
       x = "Sigma", y = "Density") +
  scale_color_manual(values = c("Actual Posterior" = "blue", "Approximate Posterior" = "red")) +
  theme_minimal()
```

Chapter 9

Testing Hypotheses

R code Exa 9.1.5 Testing Hypotheses about the Mean of a Normal Distribution with Known Variance I

```
#Page 534

mu0 <- 4
n <- 15
sigma2 <- 9
c_values <- c(1, 2, 3)

power_function <- function(mu, c) {
  sigma <- sqrt(sigma2)
  z1 <- (mu0 + c - mu) / (sigma / sqrt(n))
  z2 <- (mu0 - c - mu) / (sigma / sqrt(n))
  1 - pnorm(z1) + pnorm(z2)
}

mu_range <- seq(0, 8, length.out = 100)

plot(mu_range, power_function(mu_range, c_values[1])
    , type = "l", lwd = 2,
     xlab = "mu", ylab = "Power", main = "Power
        Functions",
```

```
      ylim = c(0, 1))

for (i in 2:length(c_values)) {
  lines(mu_range, power_function(mu_range, c_values[
     i]), col = i, lwd = 2)
}

legend("bottomright", legend = paste("c =", c_values
   ), col = 1:length(c_values), lwd = 2)
```

R code Exa 9.1.7 Testing Hypotheses about a Uniform Distribution

```
#Page 536

n <- 68
theta <- 29 / 30

alpha_delta <- theta^n

cat("The size of    is:", round(alpha_delta,4), "\n"
   )
```

R code Exa 9.1.9 Testing Hypotheses about a Bernoulli Parameter I

```
#Page 538

n <- 10
p0 <- 0.3
alpha0 <- 0.1

cumulative_prob <- function(c, n, p0) {
  sum(dbinom(c:n, size = n, prob = p0))
}
```

```r

prob_c6 <- cumulative_prob(6, n, p0)
prob_c5 <- cumulative_prob(5, n, p0)

cat("Cumulative probability for c = 6:", round(prob_c6,4), "\n")
cat("Cumulative probability for c = 5:", round(prob_c5,4), "\n")
```

R code Exa 9.1.10 Testing Hypotheses about the Mean of a Normal Distribution with Known Variance II

```r
#Page 538

n <- 10
alpha0 <- 0.05

z_critical <- qnorm(1 - alpha0/2)

cat("\nCritical Value:", round(z_critical,2))
```

R code Exa 9.1.11 Testing Hypotheses about the Mean of a Normal Distribution with Known Variance III

```r
#Page 539

Z_observed <- 2.78

p_value <- 2 * (1 - pnorm(abs(Z_observed)))

cat("P-Value:", round(p_value,4),"\n")

if (p_value < 0.01) {
```

```
  print("Reject H0")
} else {
  print("Do not reject H0")
}
```

R code Exa 9.1.12 Testing Hypotheses about a Bernoulli Parameter II

```
#Page 540

Y_observed <- 6

p0 <- 0.3
n <- 10

probability <- sum(dbinom(Y_observed:n, size = n,
   prob = p0))

cat("Probability (p-value):", round(probability,4))
```

R code Exa 9.1.16 One Sided Confidence Interval for a Bernoulli Parameter

```
#Page 542

n <- 10
alpha0 <- 0.1

q_func <- function(y) {
  if (y == 0) {
    return (0)
  }
  p0 <- 0
  while (sum(dbinom(y:n, n, p0)) <= alpha0) {
```

```
    p0 <- p0 + 0.0001
  }
  return(p0 - 0.0001)
}

y_values <- 0:n
q_values <- sapply(y_values, q_func)

cat("n =", n, ", alpha0 =", alpha0, "\n")
cat("y\tq(y)\n")
for (i in seq_along(y_values)) {
  cat(y_values[i], "\t", q_values[i], "\n")
}
```

R code Exa 9.1.18 Likelihood Ratio Test of Two Sided Hypotheses about a Bernoulli Parameter I

```
#Page 544

y <- 0:10
n <- 10
theta0 <- 0.3

phi_y <- ((n * theta0 / y)^y )* ((n*(1 - theta0)/(n
   - y))^(n-y))
pr_y <- dbinom(y, size = 10, prob = 0.3)

table_df <- data.frame(y = y, phi_y = signif(phi_y
   ,3), pr_y = signif(pr_y,3))

print(table_df)
# The answer may slightly vary due to rounding off
   values.
```

R code Exa 9.1.19 Likelihood Ratio Test of Two Sided Hypotheses about a Bernoulli Parameter II

```
#Page 545

alpha0 <- 0.05
chi_sq_quantile <- qchisq(1 - alpha0, df = 1)
round(chi_sq_quantile,3)
```

R code Exa 9.2.1 Service Times in a Queue I

```
#Page 550

f1 <- function(x, n) {
  if (all(x > 0)) {
    return(2 * factorial(n) / ((2 + sum(x))^(n + 1))
      )
  } else {
    return(0)
  }
}

f0 <- function(x, n) {
  if (all(x > 0)) {
    return(1/(2^n) * exp(-sum(x)/2))
  } else {
    return(0)
  }
}

x <- seq(0.001, 10, length.out = 100)

```

```
plot(x, sapply(x, f1, n = 1), type = "l", col = "blue", ylim = c(0, 1), ylab = "Density", xlab = "x", main = "Comparison of PDFs for Service Times")
lines(x, sapply(x, f0, n = 1), col = "red")
legend("topright", legend = c("Model f1", "Model f0"), col = c("blue", "red"), lty = 1)
```

R code Exa 9.2.2 Service Times in a Queue II

```
#Page 551

alpha <- exp(-0.5 * 4)
cat("Type I error probability (alpha):", round(alpha,3), "\n")

beta <- 1 - 2/(2 + 4)
cat("Type II error probability (beta):", round(beta,3), "\n")
```

R code Exa 9.2.3 Service Times in a Queue III

```
#Page 553

threshold <- 5.025725

alpha_star <- exp(-0.5 * threshold)
cat("Type I error probability (alpha*):", round(alpha_star,3), "\n")

beta_star <- 1 - 2/(2 + threshold)
cat("Type II error probability (beta*):", round(beta_star,3), "\n")
```

R code Exa 9.2.4 Service Times in a Queue IV

```
#Page 554

alpha <- 0.07

c <- -2 * log(alpha)
cat("Threshold value for alpha = 0.07:", round(c,3),
    "\n")

beta <- 1 - 2/(2 + c)
cat("Type II error probability (beta):", round(beta
    ,3), "\n")
```

R code Exa 9.2.5 Random Sample from a Normal Distribution

```
#Page 554

n <- 9

beta_n <- pnorm(1.645 - sqrt(n))
cat("beta:", round(beta_n,4), "\n")

c <- 0.577

alpha <- 1 - pnorm(c, mean = 0, sd = 1/sqrt(n))
cat("alpha:", round(alpha,4), "\n")

beta <- pnorm(c, mean = 1, sd = 1/sqrt(n))
cat("beta:", round(beta,4), "\n")

```

```
min_value <- 2 * alpha + beta
cat("Minimum value of 2*alpha + beta:", signif(min_value,5), "\n")
```

R code Exa 9.2.6 Sampling from a Bernoulli Distribution

```
#Page 555

n <- 10
p0 <- 0.2
alpha0 <- 0.05

prob_Y_gt_4 <- 1 - pbinom(4, size = n, prob = p0)

cat("Reject H0 when Y > 4 with alpha( ) =", round(prob_Y_gt_4,4), "\n")
```

R code Exa 9.3.6 Service Times in a Queue V

```
#Page 563

n <- 10
alpha0 <- 0.1

df <- 2 * n

quantile <- qchisq(alpha0, df = df, lower.tail = TRUE)

cat("For n =", n, "and alpha0 =", alpha0, ", the quantile is:", round(quantile,2))
```

R code Exa 9.3.7 Testing Hypotheses about the Proportion of Defective Items

```
#Page 563

n <- 20
p <- 0.1
c_values <- c(6, 7)

sizes <- sapply(c_values, function(c_val) {
  alpha <- 1 - pbinom(c_val - 1, size = n, prob = p)
  return(alpha)
})

cat("The alpha values for c values 6 and 7 are", round(sizes[1],4),"and",round(sizes[2],4))
```

R code Exa 9.4.2 Egyptian Skulls I

```
#Page 569

mu0 <- 140
sigma <- 26
n <- 30
alpha0 <- 0.05
Xn <- 131.37

c1 <- mu0 - qnorm(1 - alpha0/2) * sqrt(sigma) * n^(-0.5)
c2 <- mu0 + qnorm(1 - alpha0/2) * sqrt(sigma) * n^(-0.5)

```

```
if (Xn <= c1 || Xn >= c2) {
  cat("Reject H0:    = 140\n")
} else {
  cat("Do not reject H0:    = 140\n")
}

cat("Critical values:\n")
cat("c1 =", round(c1,2), "\n")
cat("c2 =", round(c2,2), "\n")
```

R code Exa 9.4.4 Service Times in a Queue VI

```
#Page 570

library(nleqslv)

n <- 3
alpha0 <- 0.05

solve_c1_c2 <- function(c) {
  c1 <- c[1]
  c2 <- c[2]

  eq1 <- ((c1 / (2 * n))^n) * exp(n - c1 / 2) - ((c2
      / (2 * n))^n) * exp(n - c2 / 2)

  eq2 <- pgamma(c1, shape = n, rate = 1/2) + 1 -
     pgamma(c2, shape = n, rate = 1/2) - alpha0

  return(c(eq1, eq2))
}

initial_guess <- c(1,15)

solution <- nleqslv(initial_guess, solve_c1_c2)
```

```
c1 <- solution$x[1]
c2 <- solution$x[2]

cat("c1 =", c1, "\n")
cat("c2 =", c2, "\n")
```

R code Exa 9.4.5 Testing an Interval Null Hypothesis

```
#Page 571

target_function <- function(c) {
  pnorm(0.4 + 4 * c) - pnorm(0.4 - 4 * c) - 0.95
}

solution <- uniroot(target_function, interval = c(0,
    1))

c <- solution$root

c1 <- 10 - c
c2 <- 10 + c

cat("c =", round(c,3), "\n")
cat("c1 =", round(c1,3), "\n")
cat("c2 =", round(c2,3), "\n")
```

R code Exa 9.5.3 Nursing Homes in New Mexico

```
#Page 578

x <- c(128, 281, 291, 238, 155, 148, 154, 232, 316,
   96, 146, 151, 100, 213, 208, 157, 48, 217)
```

```
sample_mean <- mean(x)

sample_sd <- sd(x)

df <- length(x) -1

mu0 <- 200

U <- sqrt(df) * (sample_mean - mu0) / sample_sd

cat("Observed value of U:", U, "\n")

critical_value <- qt(0.1, df = df)

cat("Critical value from t-distribution:", critical_
   value, "\n")

if (U <= critical_value) {
  cat("Reject H0\n")
} else {
  cat("Fail to reject H0\n")
}
```

R code Exa 9.5.4 Lengths of Fibers I

```
#Page 578

sample_mean <- 5.4
sample_sd <- 0.4226
n <- 15
mu0 <- 5.2
alpha <- 0.05

U <- (sample_mean - mu0) / (sample_sd / sqrt(n))
```

```r

cat("Observed value of U:", round(U,3), "\n")

critical_value <- qt(1 - alpha, df = n - 1)

cat("Critical value from t-distribution:", round(critical_value,3), "\n")

if (U > critical_value) {
  cat("Reject H0\n")
} else {
  cat("Fail to reject H0\n")
}

p_value <- 1 - pt(U, df = n - 1)

cat("p-value:", round(p_value,4), "\n")
```

R code Exa 9.5.5 Lengths of Fibers II

```r
#Page 579

mu0 <- 5.2
sigma <- 0.4226
n <- 15

psi <- sqrt(n) * ((mu0 + sigma/2 - mu0) / sigma)
round(psi,3)
```

R code Exa 9.5.6 Lengths of Fibers III

```r
#Page 580

```

```r
library(sadists)

power_func <- function(psi, df, alpha) {
  q_alpha <- qt(1 - alpha, df = df, ncp = 0)
  1 - pt(q_alpha, df = df, ncp = psi)
}

mu0 <- 5.2
sigma <- 0.4226
mu_alternative <- mu0 + sigma / 2
alpha <- 0.05
desired_power <- 0.8

calculate_psi <- function(n, mu_alternative, mu0,
    sigma) {
  sqrt(n) * (mu_alternative - mu0) / sigma
}

n <- 15
current_power <- 0
while (current_power < desired_power) {
  n <- n + 1
  psi <- calculate_psi(n, mu_alternative, mu0, sigma
    )
  current_power <- power_func(psi, n - 1, alpha)
}

cat("Required sample size:", n, "\n")
cat("Achieved power:", round(current_power,4), "\n")
```

R code Exa 9.5.7 Crash Test Dummies

```r
#Page 581

n <- 164
```

```
alpha <- 0.01
x_bar <- 0.2199
sigma <- 0.5342
mu0 <- 0

U <- sqrt(n) * (x_bar - mu0) / sigma

t_critical <- qt(1 - alpha, df = n - 1)

cat("Test statistic U:", round(U,4), "\n")
cat("Critical value:", round(t_critical,2), "\n")

mu_alternative <- sigma / 4
psi <- sqrt(n) * (mu_alternative - mu0) / sigma

power <- 1 - pt(t_critical, df = n - 1, ncp = psi)

cat("Noncentrality parameter:", signif(psi,3), "\n")
cat("Power of the test:", round(power,3), "\n")
```

R code Exa 9.5.9 Egyptian Skulls II

```
#Page 582

n <- 30
alpha <- 0.05
mu0 <- 140
x_bar <- 131.37
sigma <- 5.129

U <- sqrt(n) * (x_bar - mu0) / sigma

t_critical <- round(qt(1 - alpha / 2, df = n - 1),3)

cat("Test statistic U:", round(U,4), "\n")
```

```r
cat("Critical value (c1, c2):", -t_critical,"and", t_critical, "\n")

if (U < -t_critical || U > t_critical) {
  cat("Reject H0 at level", alpha, "\n")
} else {
  cat("Do not reject H0 at level", alpha, "\n")
}
#The answer may slightly vary due to rounding off values.
```

R code Exa 9.5.10 Lengths of Fibers IV

```r
#Page 582

n <- 15
alpha <- 0.05
df <- n - 1
U <- 1.833

t_critical <- round(qt(1 - alpha / 2, df),3)

cat("Test statistic U:", U, "\n")
cat("Critical values (c1, c2):", -t_critical, t_critical, "\n")

if (U <= -t_critical || U >= t_critical) {
  cat("Reject H0 at level", alpha, "\n")
} else {
  cat("Do not reject H0 at level", alpha, "\n")
}
```

R code Exa 9.5.11 Lengths of Fibers V

```
#Page 583

df <- 14
U <- 1.833

p_value <- 2 * (1 - pt(U, df))

cat("The p-value is:", round(p_value,4), "\n")
```

R code Exa 9.6.2 Rain from Seeded Clouds I

```
#Page 589

Xm <- 5.13
Ym <- 3.99
SX2 <- 63.96
SY2 <- 67.39
n <- 26
alpha <- 0.01
df = 50

critical_value <- qt(1-alpha, df = df)
test_statistic <- (sqrt(df) * (Xm - Ym)) / (sqrt(1/
    26 + 1/26) * sqrt((SX2 + SY2)))

cat("Critical value:", round(critical_value,3), "\n"
    )
cat("Test statistic:", round(test_statistic,2), "\n"
    )
```

R code Exa 9.6.3 Roman Pottery in Britain I

```
#Page 590
```

```
m <- 14
n <- 5
Xm <- 12.56
Ym <- 17.32
SX2 <- 24.65
SY2 <- 11.01
alpha <- 0.005

U <- (sqrt(m+n-2)*(Xm - Ym))/(sqrt(1/m +1/n)*sqrt(
    SX2+SY2))

df <- m + n - 2

critical_value <- qt(1 - alpha, df)

p_value <- pt(U, df)

cat("Test statistic U:", round(U,3), "\n")
cat("Critical value:", round(critical_value,3), "\n"
    )
cat("p-value:", signif(p_value,1), "\n")
#The answer may slightly vary due to rounding off
    values.
```

R code Exa 9.6.4 Roman Pottery in Britain II

```
#Page 591

m <- 14
n <- 5
alpha <- 0.01
psi <- 2.88
df <- m + n - 2

```

```
critical_value <- 1.5/sqrt(1/m +1/n)

power <- 1 - pt(qt(1 - alpha, df = df), df = df, ncp
    = psi)

cat("Critical value:", round(critical_value,2), "\n"
  )
cat("Power of the test:", round(power,2), "\n")
```

R code Exa 9.6.5 Comparing Copper Ores I

```
#Page 591

m <- 8
n <- 10
X_bar <- 2.6
Y_bar <- 2.3
S2_X <- 0.32
S2_Y <- 0.22
alpha <- 0.01
df <- m + n - 2

pooled_variance <- ((m - 1) * S2_X + (n - 1) * S2_Y)
    / df

U <- (sqrt(m+n-2)*(X_bar - Y_bar))/(sqrt(1/m +1/n)*
  sqrt(S2_X+S2_Y))

critical_value <- qt(1 - alpha / 2, df)

cat("Test Statistic (U):", round(U,3), "\n")
cat("Critical Value:", round(critical_value,3), "\n"
  )

if (abs(U) > critical_value) {
```

```
  cat("Reject the null hypothesis at level", alpha, "\n")
} else {
  cat("Do not reject the null hypothesis at level", alpha, "\n")
}
#The answer may slightly vary due to rounding off values.
```

R code Exa 9.6.6 Comparing Copper Ores II

```
#Page 594

x_bar <- 2.6
y_bar <- 2.3
s2_x <- 0.32
s2_y <- 0.22
m <- 8
n <- 10

numerator <- x_bar - y_bar
denominator <- sqrt((s2_x / (m * (m - 1))) + (s2_y / (n * (n - 1))))
V <- numerator / denominator

nu <- ((s2_x / (m * (m - 1))) + (s2_y / (n * (n - 1))))^2 /
  ((1 / (m - 1)^3) * (s2_x / m)^2 + (1 / (n - 1)^3) * (s2_y / n)^2)

p_value <- 2 * (1 - pt(V, df = nu))

cat("Test Statistic V:", round(V,3), "\n")
cat("Degrees of Freedom nu:", round(nu,2), "\n")
cat("P-value:", signif(p_value,2), "\n")
```

R code Exa 9.6.7 Comparing Copper Ores III

```r
#Page 595

x_bar <- 2.6
s2_x <- 0.32
y_bar <- 2.3
s2_y <- 0.22
m <- 8
n <- 10

mu_0 <- (m * x_bar + n * y_bar) / (m + n)

tolerance <- 1e-6
max_iter <- 13
iter <- 0

update_values <- function(mu, x_bar, y_bar, s2_x, s2_y, m, n) {
  sigma2_1 <- (1 / m) * (s2_x + m * (x_bar - mu)^2)
  sigma2_2 <- (1 / n) * (s2_y + n * (y_bar - mu)^2)
  new_mu <- (m * x_bar / sigma2_1 + n * y_bar / sigma2_2) / (m / sigma2_1 + n / sigma2_2)
  return(list(mu = new_mu, sigma2_1 = sigma2_1, sigma2_2 = sigma2_2))
}

for (i in 1:13) {
  result <- update_values(mu_0, x_bar, y_bar, s2_x, s2_y, m, n)
  if (abs(result$mu - mu_0) < tolerance) {
    cat("Convergence achieved after", iter, " iterations.\n")
    break
```

```
  }
  mu_0 <- result$mu
}

mu_hat <- result$mu
sigma2_1_hat <- result$sigma2_1
sigma2_2_hat <- result$sigma2_2

Lambda_xy <- 0.01356
LR_statistic <- -2 * log(Lambda_xy)

p_value <- 1 - pchisq(LR_statistic, df = 1)

cat("Final MLEs:\n")
cat("mu_hat:", round(mu_hat,3), "\n")
cat("sigma2_1_hat:", round(sigma2_1_hat,4), "\n")
cat("sigma2_2_hat:", round(sigma2_2_hat,4), "\n")
cat("\nLikelihood Ratio Statistic:", round(LR_statistic,3), "\n")
cat("P-value:", round(p_value,3), "\n")
```

R code Exa 9.7.2 Quantile of an F Distribution

```
# Page 598

df1 <- 6
df2 <- 12

quantile_0.05 <- qf(0.05, df1, df2)

cat("The 0.05 quantile of the F-distribution with", df1, "and", df2, "degrees of freedom is:", round(quantile_0.05,2), "\n")
```

R code Exa 9.7.3 F Test

```r
#Page 600

s2_x <- 30
s2_y <- 40

df1 <- 5
df2 <- 20

f_statistic <- s2_x / s2_y

f_crit1 <- qf(0.025, df1, df2)
f_crit2 <- qf(0.975, df1, df2)

cat("The critical values for the F-test are:", f_crit1, "and", f_crit2, "\n")

if (f_statistic < f_crit1 || f_statistic > f_crit2) {
  cat("Reject the null hypothesis that the variances are equal at alpha = 0.05.\n")
} else {
  cat("Do not reject the null hypothesis that the variances are equal at alpha = 0.05.\n")
}

f_stat_alt <- 2.71 / 3
power <- 1 - pf(f_stat_alt, df1, df2)
cat("The power of the test for  1 ^2 = 3  2 ^2 is:", round(power,3), "\n")
```

R code Exa 9.7.4 Rain from Seeded Clouds II

```
#Page 601

s2_x <- 63.96
s2_y <- 67.39

df1 <- 25
df2 <- 25

f_statistic <- s2_x / s2_y

f_crit1 <- qf(0.025, df1, df2)
f_crit2 <- qf(0.975, df1, df2)

cat("The critical values for the F-test are:", round
   (f_crit1,4), "and", round(f_crit2,4), "\n")

if (f_statistic < f_crit1 || f_statistic > f_crit2)
   {
  cat("Reject the null hypothesis that the variances
      are equal.\n")
} else {
  cat("Do not reject the null hypothesis that the
     variances are equal.\n")
}
```

R code Exa 9.8.4 Calorie Counts on Food Labels

```
#Page 610

mean_posterior <- 0.1154
var_posterior <- 4.62

d_values <- seq(0, 5, by = 0.01)
```

```
posterior_probs <- pnorm(d_values, mean_posterior, sqrt(var_posterior)) -
  pnorm(-d_values, mean_posterior, sqrt(var_posterior))

plot(d_values, posterior_probs, type = "l", col = "blue",
     xlab = "|  |", ylab = "Posterior Probability",
     main = "Posterior Probability that |  | is at most d",
     xlim = c(0, 5), ylim = c(0, 1))
```

R code Exa 9.8.5 Pesticide Residue on Celery

```
#Page 611

df <- 76
prob <- 1 - pt(1.974,df)

round(prob,3)
```

R code Exa 9.8.6 Roman Pottery in Britain III

```
#Page 614

m <- 14
n <- 5
Xm <- 12.56
Ym <- 17.32
SX2 <- 24.65
SY2 <- 11.01

```

```
U <- (sqrt(m+n-2)*(Xm - Ym))/(sqrt(1/m +1/n)*sqrt(SX2+SY2))

df <- m + n - 2

p_value <- pt(U, df)
cat("p-value:", signif(p_value,1), "\n")

p_posterior <- p_value
cat("Pr(  1       2 |x) =", signif(p_posterior,1), "\n")
```

R code Exa 9.8.7 Roman Pottery in Britain IV

```
#Page 614

library(ggplot2)

T17 <- function(x) {
  pt(x, df = 17)
}

d_values <- seq(0, 10, by = 0.01)

posterior_probs <- T17(1.33 * (d_values + 4.76)) - T17(1.33 * (-d_values + 4.76))

data <- data.frame(d = d_values, posterior_prob = posterior_probs)

median_value <- 4.76
median_y_value <- approx(d_values, posterior_probs, xout = median_value)$y

ggplot(data, aes(x = d, y = posterior_prob)) +
```

```
  geom_line(color = "blue") +
  geom_segment(aes(x = median_value, xend = median_value, y = 0, yend = median_y_value),
               linetype = "dotted", color = "red") +
  geom_segment(aes(x = 0, xend = median_value, y = median_y_value, yend = median_y_value),
               linetype = "dotted", color = "red") +
  labs(title = "Posterior Probability that | 1 − 2 | is at most d",
       x = "d",
       y = "Posterior Probability") +
  theme_minimal()
```

Chapter 10

Categorical Data and Nonparametric Methods

R code Exa 10.1.3 Blood Types I

```
#Page 627

observed_counts <- c(A = 2162, B = 738, AB = 228, O
    = 2876)

total <- sum(observed_counts)

theoretical_probabilities <- c(A = 1/3, B = 1/8, AB
    = 1/24, O = 1/2)

expected_counts <- round(total * theoretical_
    probabilities,1)

chi_square_statistic <- sum((observed_counts -
    expected_counts)^2 / expected_counts)

degrees_of_freedom <- length(observed_counts) - 1
p_value <- 1 - pchisq(chi_square_statistic, degrees_
    of_freedom)
```

```
cat("Chi-square test statistic:", round(chi_square_statistic, 3), "\n")
cat("p-value:", signif(p_value, digits = 3), "\n")
```

R code Exa 10.1.4 Montana Outlook Poll I

```
#Page 627

observed_counts <- c(Worse = 58, Same = 64, Better = 67)

total <- sum(observed_counts)

theoretical_probabilities <- rep(1/3, length(observed_counts))

expected_counts <- total * theoretical_probabilities

chi_square_statistic <- sum((observed_counts - expected_counts)^2 / expected_counts)

degrees_of_freedom <- length(observed_counts) - 1
p_value <- 1 - pchisq(chi_square_statistic, degrees_of_freedom)

cat("Chi-square test statistic:", round(chi_square_statistic, 4), "\n")
cat("p-value:", round(p_value, 3), "\n")
```

R code Exa 10.1.5 Testing Hypotheses about a Proportion

```
#Page 627
```

```
N1 <- 16
N2 <- 84

n <- N1 + N2

p0_1 <- 0.10
p0_2 <- 0.90

expected_N1 <- n * p0_1
expected_N2 <- n * p0_2

Q <- sum((c(N1, N2) - c(expected_N1, expected_N2))^2
    / c(expected_N1, expected_N2))

p_value <- 1 - pchisq(Q, df = 1)

cat("Chi-square test statistic:", round(Q, 3), "\n")
cat("p-value:", round(p_value, 4), "\n")
```

R code Exa 10.1.6 Failure Times of Ball Bearings

```
#Page 629

mean_log <- log(50)
var_log <- 0.25

quantile_25 <- qnorm(0.25, mean = mean_log, sd =
    sqrt(var_log))
quantile_50 <- qnorm(0.5, mean = mean_log, sd = sqrt
    (var_log))
quantile_75 <- qnorm(0.75, mean = mean_log, sd =
    sqrt(var_log))

cat("25th quantile:", round(quantile_25, 3), "\n")
```

```r
cat("50th quantile:", round(quantile_50, 3), "\n")
cat("75th quantile:", round(quantile_75, 3), "\n")

observed_counts <- c(3, 4, 8, 8)

n <- sum(observed_counts)

expected_probs <- rep(0.25, 4)

expected_counts <- n * expected_probs

Q <- sum((observed_counts - expected_counts)^2 / expected_counts)

cat("Chi-square test statistic:", round(Q, 3), "\n")
```

R code Exa 10.1.7 Blood Types II

```r
#Page 630

observed_counts <- c(2162, 738, 228, 2876)

expected_counts <- c(2001.3, 750.5, 250.2, 3002.0)

test_statistic <- -2 * sum(observed_counts * log(expected_counts / observed_counts))

df <- length(observed_counts) - 1

p_value <- 1 - pchisq(test_statistic, df)

cat("Test statistic:", round(test_statistic, 2), "\n")
cat("p-value:", signif(p_value, digits = 3), "\n")
# The answer may slightly vary due to rounding off
```

```
values.
```

R code Exa 10.2.5 Failure Times of Ball Bearings

```
#Page 638

observed_counts <- c(3, 4, 8, 8)
n <- sum(observed_counts)
mu_hat <- 4.150
sigma2_hat <- 0.2722
sigma_hat <- sqrt(sigma2_hat)

intervals <- c(3.575, 3.912, 4.249)

pi1 <- pnorm(intervals[1], mean = mu_hat, sd = sigma_hat)
pi2 <- pnorm(intervals[2], mean = mu_hat, sd = sigma_hat) - pi1
pi3 <- pnorm(intervals[3], mean = mu_hat, sd = sigma_hat) - pi2 - pi1
pi4 <- 1 - pnorm(intervals[3], mean = mu_hat, sd = sigma_hat)

pi <- round(c(pi1, pi2, pi3, pi4),4)
print(pi)

expected_counts <- n * pi

test_statistic <- sum((observed_counts - expected_counts)^2 / expected_counts)

cat("Test statistic Q:", round(test_statistic, 3), "\n")
# The answer may slightly vary due to rounding off values.
```

R code Exa 10.2.6 Prussian Army Deaths

```r
#Page 638

counts <- c(0, 1, 2, 3, 4)
observations <- c(144, 91, 32, 11, 2)
total_obs <- sum(observations)

theta_hat <- 196 / 280

pi_theta_hat <- round(dpois(counts, lambda = theta_hat),4)

Q <- sum((observations - total_obs * pi_theta_hat)^2 / (total_obs * pi_theta_hat))

cat("Poisson probabilities at MLE 0.7:", pi_theta_hat, "\n")
cat("Q statistic:", round(Q, 3), "\n")
# The answer may slightly vary due to rounding off values.
# The answer provided in the textbook is wrong.
```

R code Exa 10.3.2 College Survey

```r
#Page 644

observed <- matrix(c(24, 23, 12,
                     24, 14, 10,
                     17, 8, 13,
                     27, 19, 9),
```

```
                     nrow = 4, byrow = TRUE)

row_totals <- c(59, 48, 38, 55)
col_totals <- c(92, 64, 44)
total <- sum(row_totals)

expected <- outer(row_totals, col_totals, "*") /
   total

Q <- sum((observed - expected)^2 / expected)

df <- (nrow(observed) - 1) * (ncol(observed) - 1)

cat("Q statistic:", round(Q, 2), "\n")
cat("Degrees of freedom:", df, "\n")
```

R code Exa 10.3.3 Montana Outlook Poll II

```
#Page 644

observed <- matrix(c(20, 15, 12,
                     24, 27, 32,
                     14, 22, 23),
                   nrow = 3, byrow = TRUE)

expected <- matrix(c(14.42, 15.92, 16.66,
                     25.47, 28.11, 29.42,
                     18.11, 19.98, 20.92),
                   nrow = 3, byrow = TRUE)

Q <- sum((observed - expected)^2 / expected)

df <- (nrow(observed) - 1) * (ncol(observed) - 1)

p_value <- 1 - pchisq(Q, df)
```

```r

cat("Q statistic:", round(Q, 2), "\n")
cat("Degrees of freedom:", df, "\n")
cat("p-value:", round(p_value, 3), "\n")
```

R code Exa 10.4.3 A Clinical Trial

```r
#Page 650

observed <- matrix(c(18, 22,
                     13, 25,
                     22, 16,
                     24, 10),
                   nrow = 4, byrow = TRUE)

total <- colSums(observed)
total_group <- rowSums(observed)
total_all <- sum(observed)

expected <- outer(total_group, total) / total_all

Q <- sum((observed - expected)^2 / expected)

df <- (nrow(observed) - 1) * (ncol(observed) - 1)

p_value <- 1 - pchisq(Q, df)

quantile <- pchisq(Q, df)


cat("Q statistic:", round(Q, 2), "\n")
cat("Degrees of freedom:", df, "\n")
cat("p-value:", round(p_value, 3), "\n")
cat("Quantile:", round(quantile, 3), "\n")
```

R code Exa 10.6.2 Testing Whether a Sample Comes from a Standard Normal Distribution

```r
#Page 661

D_n_star <- 0.3372 - 0.16

n <- 25
ks_statistic <- sqrt(n) * D_n_star

H_value <- 0.6
tail_area <- 1 - H_value

print(paste("D*_n:", round(D_n_star, 4)))
print(paste("KS Statistic (sqrt(n) * D*_n):", round(ks_statistic, 4)))
print(paste("Tail Area (1 - H):", round(tail_area, 4)))
```

R code Exa 10.6.4 Calcium Supplements and Blood Pressure I

```r
#Page 664

t <- c(0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75,
       0.80, 0.85, 0.90, 0.95, 1.00, 1.05, 1.10, 1.15)
H_t <- c(0.0000, 0.0003, 0.0028, 0.0126, 0.0361, 0.0772, 0.1357, 0.2080,
         0.2888, 0.3728, 0.4559, 0.5347, 0.6073, 0.6725, 0.7300, 0.7798,
         0.8223, 0.8580)
```

```
Dmn <- 0.409
test_statistic <- sqrt(110/21) * Dmn
alpha0 <- 0.346
H <- qnorm(1 - alpha0)

cat("Test statistic:", round(test_statistic,3), "\n"
   )

index <- max(which(t <= test_statistic))
t1 <- t[index]
t2 <- t[index + 1]
H_t1 <- H_t[index]
H_t2 <- H_t[index + 1]

H <- H_t1 + (H_t2 - H_t1) * (test_statistic - t1) /
   (t2 - t1)

cat("H(0.936):", round(H,3), "\n")
```

R code Exa 10.7.3 Rain from Seeded Clouds

```
#Page 673

library(MASS)

log_rainfalls <- c(1.411, 1.685, 2.398, 3.123,
   3.172, 3.515, 4.912, 5.123, 5.285, 5.345,
                   5.378, 5.396, 5.412, 5.467,
                      5.478, 5.512, 5.789, 6.123,
                      6.478, 6.789,
                   6.912, 7.123, 7.512, 7.789,
                      8.123, 8.456)

data <- data.frame(y = log_rainfalls)
```

```
fit <- rlm(y ~ 1, data = data, psi = psi.huber, k =
    2.5)

theta_hat <- coef(fit)

round(theta_hat,3)
#The answer may slightly vary due to rounding off
    values.
```

R code Exa 10.8.2 Calorie Counts in Hotdogs

```
#Page 679

calorie_counts <- c(186, 181, 176, 149, 184, 190,
    158, 139, 175, 148, 152, 111, 141, 153, 190, 157,
     131, 149, 135, 132)

hypothesized_median <- 150

W <- sum(calorie_counts < hypothesized_median)

n <- length(calorie_counts)

p_value <- 2 * min(pbinom(W, n, 0.5), pbinom(W - 1,
    n, 0.5, lower.tail = FALSE))

cat("p-value:", round(p_value,4), "\n")
```

R code Exa 10.8.4 Comparing Copper Ores

```
#Page 681

```

```
S <- 104
mu <- 76
sigma <- sqrt(126.67)

Z <- (S - mu) / sigma

p_value <- 2 * (1 - pnorm(abs(Z)))

cat("Z-score:", round(Z,2), "\n")
cat("p-value:", signif(p_value,3), "\n")
#The answer may slightly vary due to rounding off
    values.
```

R code Exa 10.8.5 Calcium Supplements and Blood Pressure II

```
#Page 682

calculate_test <- function(S, m, n) {
  N <- m + n
  mu <- m * (N + 1) / 2
  sigma <- sqrt((m * n * (N + 1)) / 12)
  Z <- (S - mu) / sigma
  p_value <- 2 * (1 - pnorm(abs(Z)))
  return(list(p_value = round(p_value,2)))
}

m <- 10
n <- 11

S1 <- 123
result1 <- calculate_test(S1, m, n)

S2 <- 126
result2 <- calculate_test(S2, m, n)

```

```
cat("p-value:", result1$p_value, "\n\n")

cat("p-value:", result2$p_value, "\n")
```

Chapter 11

Linear Statistical Models

R code Exa 11.1.1 Blood Pressure I

```
#Page 689

xi <- c(1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 4.6, 1.6, 5.5, 3.4)
yi <- c(0.7, -1.0, -0.2, -1.2, -0.1, 3.4, 0.0, 0.8, 3.7, 2.0)

plot(xi, yi,
     xlim = range(c(xi, yi)), ylim = range(c(xi, yi)),
     xlab = "x", ylab = "y",
     main = "Reactions to Two Drugs",
     pch = 16, col = "blue")

text(xi, yi, labels = 1:10, pos = 4, offset = 0.5, cex = 0.8)
```

R code Exa 11.1.2 Blood Pressure II

```
#Page 691

xi <- c(1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 4.6, 1.6, 5.5, 3.4)
yi <- c(0.7, -1.0, -0.2, -1.2, -0.1, 3.4, 0.0, 0.8, 3.7, 2.0)

model <- lm(yi ~ xi)

plot(xi, yi,
     xlim = range(c(xi, yi)), ylim = range(c(xi, yi)),
     xlab = "Reaction to Drug A (x)", ylab = "Reaction to Drug B (y)",
     main = "Reactions to Two Drugs with Fitted Line",
     pch = 16, col = "blue")

abline(model, col = "red", lwd = 2)

for(i in 1:length(xi)) {
  segments(xi[i], yi[i], xi[i], predict(model, newdata = data.frame(xi = xi[i])),
           col = "green", lty = 2)
}

text(xi, yi, labels = 1:10, pos = 4, offset = 0.5, cex = 0.8)

legend("topleft",
       legend = c("Observed values", "Fitted line", "Vertical deviations"),
       col = c("blue", "red", "green"),
       pch = c(16, NA, NA),
       lty = c(NA, 1, 2),
       lwd = c(NA, 2, 1))

cat("Regression equation: y =", round(coef(model)
```

```
    [1], 3), "+", round(coef(model)[2], 3), "x\n")
```

R code Exa 11.1.3 Fitting a Parabola

```
#Page 694

xi <- c(1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 4.6, 1.6,
    5.5, 3.4)
yi <- c(0.7, -1.0, -0.2, -1.2, -0.1, 3.4, 0.0, 0.8,
    3.7, 2.0)

linear_model <- lm(yi ~ xi)

plot(xi, yi,
     xlim = c(-1, 6), ylim = c(-2, 5),
     xlab = "Reaction to Drug A (x)", ylab = "
        Reaction to Drug B (y)",
     main = "Reactions to Two Drugs with Fitted
        Lines",
     pch = 16, col = "blue")

abline(linear_model, col = "red", lwd = 2)

x_seq <- seq(-1, 6, length.out = 100)
y_parabola <- -0.744 + 0.616*x_seq + 0.013*x_seq^2
lines(x_seq, y_parabola, col = "green", lwd = 2)

legend("topleft",
       legend = c("Observed values", "Linear fit", "
          Parabolic fit"),
       col = c("blue", "red", "green"),
       pch = c(16, NA, NA),
       lty = c(NA, 1, 1),
       lwd = c(NA, 2, 2))
```

R code Exa 11.1.5 Fitting a Linear Function of Two Variables

```
#Page 696

xi1 <- c(1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 4.6, 1.6, 5.5, 3.4)
xi2 <- c(66, 62, 64, 61, 63, 70, 68, 62, 68, 66)
yi <- c(0.7, -1.0, -0.2, -1.2, -0.1, 3.4, 0.0, 0.8, 3.7, 2.0)

data <- data.frame(xi1, xi2, yi)

model <- lm(yi ~ xi1 + xi2, data=data)

coefficients <- round(coef(model),4)
cat("Coefficients:\n")
cat("  0 =", coefficients[1], "\n")
cat("  1 =", coefficients[2], "\n")
cat("  2 =", coefficients[3], "\n")

cat("\nLeast-Squares Linear Function:\n")
cat("y =", coefficients[1], "+", coefficients[2], "* x1 +", coefficients[3], "* x2\n")
#The answer may slightly vary due to rounding off values.
```

R code Exa 11.2.1 Pressure and the Boiling Point of Water I

```
#Page 698

boiling_point <- c(194.5, 194.3, 197.9, 198.4, 199.4, 199.9, 200.9, 201.1, 201.4, 201.3,
```

```
                     203.6, 204.6, 209.5, 208.6, 210.7, 211.9, 212.2)
pressure <- c(20.79, 20.79, 22.40, 22.67, 23.15, 23.35, 23.89, 23.99, 24.02, 24.01,
               25.14, 26.57, 28.49, 27.76, 29.04, 29.88, 30.06)

data <- data.frame(boiling_point, pressure)

model <- lm(pressure ~ boiling_point, data=data)

coefficients <- round(coef(model),4)
cat("Coefficients:\n")
cat("  0 =", coefficients[1], "\n")
cat("  1 =", coefficients[2], "\n")

cat("\nLeast-Squares Line:\n")
cat("y =", coefficients[1], "+", coefficients[2], "*x\n")
#The answer may slightly vary due to rounding off values.
```

R code Exa 11.2.5 Predicting Pressure from the Boiling Point of Water

```
#Page 704

boiling_point <- c(194.5, 194.3, 197.9, 198.4, 199.4, 199.9, 200.9, 201.1, 201.4, 201.3,
                     203.6, 204.6, 209.5, 208.6, 210.7, 211.9, 212.2)
pressure <- c(20.79, 20.79, 22.40, 22.67, 23.15, 23.35, 23.89, 23.99, 24.02, 24.01,
               25.14, 26.57, 28.49, 27.76, 29.04, 29.88, 30.06)

```

```
data <- data.frame(boiling_point, pressure)

model <- lm(pressure ~ boiling_point, data=data)

coefficients <- coef(model)
cat("Coefficients:\n")
cat("  0 =", coefficients[1], "\n")
cat("  1 =", coefficients[2], "\n")

predicted_pressure <- coefficients[1] + coefficients[2] * 201.5
cat("\nPredicted Pressure for boiling point 201.5:", round(predicted_pressure,1), "\n")

plot(data$boiling_point, data$pressure, main="Boiling Point vs Pressure",
     xlab="Boiling Point", ylab="Pressure", pch=19, col="blue")
abline(model, col="red")

points(201.5, predicted_pressure, col="green", pch=19)
text(201.5, predicted_pressure, labels=round(predicted_pressure, 2), pos=4, col="green")

segments(201.5, par("usr")[3], 201.5, predicted_pressure, col="green", lty=2)
segments(par("usr")[1], predicted_pressure, 201.5, predicted_pressure, col="green", lty=2)
```

R code Exa 11.3.2 Gasoline Mileage I

```
#Page 714

beta0 <- 0.01537
```

```
beta1 <- 1.396e-4
sigma <- 7.181e-3
x_mean <- 183.97
sx <- 1036.9

U1 <- sx * (beta1 - 0) / sigma

cat("Test statistic U1:", round(U1,2), "\n")
#The answer may slightly vary due to rounding off values.
```

R code Exa 11.3.3 Pressure and the Boiling Point of Water II

```
#Page 715

beta0 <- -81.049
beta1 <- 0.5228
n <- 17
s2x <- 530.78
x_mean <- 202.95
sigma <- 0.2328
x_test <- 201.5
y_test <- 24.5

U01 <- sqrt(1/n + (x_mean - x_test)^2 / s2x) * (beta0 + x_test * beta1 - y_test) / sigma

df <- n - 2

p_value <- 2 * pt(abs(U01), df, lower.tail = FALSE)

cat("Test statistic U01:", round(U01,4), "\n")
cat("Degrees of freedom:", df, "\n")
cat("p-value:",round(p_value,4), "\n")
```

R code Exa 11.3.4 Gasoline Mileage II

```r
#Page 716

beta_hat_1 <- 1.396e-4
sigma <- 7.181e-3
x_bar <- 183.97
sx <- 1036.9
n <- 173
alpha_0 <- 0.2

se_beta_1 <- sigma / (sx)

t_value <- qt(1 - alpha_0/2, df = n-2)

margin_of_error <- se_beta_1 * t_value

ci_lower <- beta_hat_1 - margin_of_error
ci_upper <- beta_hat_1 + margin_of_error

cat("t-value for 0.8 confidence interval:", round(t_value,3), "\n")

cat("80% Confidence Interval for  1:\n")
cat("Lower bound:", format(ci_lower,digits = 4, scientific = TRUE), "\n")
cat("Upper bound:", format(ci_upper,digits = 4, scientific = TRUE), "\n")
```

R code Exa 11.3.5 Gasoline Mileage III

```r
#Page 717
```

```r

beta_0 <- 0.01537
beta_1 <- 1.396e-4
sigma <- 7.181e-3
x_bar <- 183.97
sx <- 1036.9
n <- 173
alpha_0 <- 0.1
x <- 100

y_hat <- beta_0 + beta_1 * x

se_pred <- sigma * sqrt(1 + 1/n + (x - x_bar)^2 / (sx^2 * (n-1)))

t_value <- qt(1 - alpha_0/2, df = n-2)

margin_of_error <- t_value * round(se_pred,5)

lower_gpm <- round(y_hat - margin_of_error,5)
upper_gpm <- round(y_hat + margin_of_error,5)

lower_mpg <- 1 / upper_gpm
upper_mpg <- 1 / lower_gpm

cat("90% Prediction Interval:\n")
cat("For Gallons per Mile:\n")
cat("Lower bound:", lower_gpm, "\n")
cat("Upper bound:", upper_gpm, "\n\n")
cat("For Miles per Gallon:\n")
cat("Lower bound:", round(lower_mpg, 2), "\n")
cat("Upper bound:", round(upper_mpg, 2), "\n")
#The answer may slightly vary due to rounding off values.
```

R code Exa 11.3.6 Pressure and the Boiling Point of Water III

```
#Page 717

xi <- c(194.5, 194.3, 197.9, 198.4, 199.4, 199.9,
   200.9, 201.1, 201.4, 201.3, 203.6, 204.6, 209.5,
   208.6, 210.7, 211.9, 212.2)
yi <- c(20.79, 20.79, 22.40, 22.67, 23.15, 23.35,
   23.89, 23.99, 24.02, 24.01, 25.14, 26.57, 28.49,
   27.76, 29.04, 29.88, 30.06)

beta0 <- -81.06
beta1 <- 0.5229
y_hat <- beta0 + beta1 * xi
e_i <- yi - y_hat

par(mfrow=c(1,2))

plot(xi, e_i, main="Residuals vs Boiling Point",
   xlab="Boiling Point", ylab="Residuals")
abline(h=0, col="red", lty=2)

log_yi <- log(yi)
log_beta0 <- -0.9709
log_beta1 <- 0.0206
log_y_hat <- log_beta0 + log_beta1 * xi
log_e_i <- log_yi - log_y_hat

plot(xi, log_e_i, main="Log Residuals vs Boiling
   Point", xlab="Boiling Point", ylab="Log Residuals
   ")
abline(h=0, col="red", lty=2)
par(mfrow=c(1,1))
```

R code Exa 11.3.7 Pressure and the Boiling Point of Water IV

```r
# Page 720

xi <- c(194.5, 194.3, 197.9, 198.4, 199.4, 199.9,
   200.9, 201.1, 201.4, 201.3, 203.6, 209.5, 208.6,
   210.7, 211.9, 212.2)
yi <- c(20.79, 20.79, 22.40, 22.67, 23.15, 23.35,
   23.89, 23.99, 24.02, 24.01, 25.14, 28.49, 27.76,
   29.04, 29.88, 30.06)

log_yi <- log(yi)
log_beta0 <- -0.9709
log_beta1 <- 0.0206
log_y_hat <- log_beta0 + log_beta1 * xi
log_e_i <- log_yi - log_y_hat

qqnorm(log_e_i, main="Normal Quantile Plot of Log-
   Linear Regression Residuals")
```

R code Exa 11.3.9 Pressure and the Boiling Point of Water V

```r
#Page 725

n <- 16

t_quantile <- qt(0.9917, df = n - 2)

cat("The t-quantile for a 0.9917 confidence level
   with", n - 2, "degrees of freedom is:", t_
   quantile, "\n")
# The answer may slightly vary due to rounding off
   values.
```

R code Exa 11.4.2 Pressure and the Boiling Point of Water VI

```r
#Page

alpha0 <- 0.1
n <- 16
beta1_hat <- 0.0205
sx <- 449.2

df <- n - 1

t_quantile <- qt(1 - alpha0 / 2, df = df)

margin_of_error <- round(t_quantile / sx,4)
lower_bound <- beta1_hat - margin_of_error
upper_bound <- beta1_hat + margin_of_error

cat("The t-quantile for a", 1 - alpha0, "confidence
    level with", df, "degrees of freedom is:", round(
    t_quantile,4), "\n")
cat("The interval estimate for beta1 is:", "(",
    lower_bound, ",", upper_bound, ")\n")
# The answer may slightly vary due to rounding off
    values.
```

R code Exa 11.4.3 Pressure and the Boiling Point of Water VII

```r
#Page 733

beta0_hat <- -0.9518
beta1_hat <- 0.0205
x <- 208
n <- 16
x_bar <- 202.85
s2_x <- 527.9
sigma_hat <- 2.616e-3
alpha0 <- 0.1
```

```
t_value <- 1.761

Y_hat <- beta0_hat + beta1_hat * x

se <- sigma_hat * sqrt(1 + 1/n + (x - x_bar)^2 / s2_
    x)

margin_of_error <- t_value * se

log_pressure_lower <- Y_hat - margin_of_error
log_pressure_upper <- Y_hat + margin_of_error

pressure_lower <- exp(log_pressure_lower)
pressure_upper <- exp(log_pressure_upper)

cat("The interval for log-pressure is: (", round(log
    _pressure_lower,3), ",", round(log_pressure_upper
    ,3), ")\n")
cat("The interval for pressure is: (", round(
    pressure_lower,2), ",", round(pressure_upper,2),
    ")\n")
```

R code Exa 11.4.4 Gasoline Mileage IV

```
#Page 734

sx <- 1036.9
sigma_hat <- 7.181e-3
beta1_hat <- 1.396e-4
n <- 173
df <- n - 2

T171 <- function(x) pt(x, df)

posterior_cdf <- function(c) {
```

```
  T171(sx / sigma_hat * (c - beta1_hat)) - T171(sx / sigma_hat * (-c - beta1_hat))
}

c_values <- seq(0, 2e-4, by = 1e-6)
cdf_values <- sapply(c_values, posterior_cdf)

plot(c_values, cdf_values, type = "l", col = "blue", lwd = 2, xlim = c(0.00010,0.00018),
     xlab = "|beta1|", ylab = "Posterior CDF", main = "Posterior CDF of |beta1|")
abline(v = 1.2e-4, col = "red", lty = 2)
abline(v = 1.6e-4, col = "green", lty = 2)

legend("bottomright", legend = c("|beta1| = 1.2e-4", "|beta1| = 1.6e-4"),
       col = c("red", "green"), lty = 2, cex = 0.8)
```

R code Exa 11.5.1 Unemployment in the 1950s I

```
#Page 736

unemployment <- c(3.1, 1.9, 1.7, 1.6, 3.2, 2.7, 2.6, 2.9, 4.7, 3.8)
index_production <- c(113, 123, 127, 138, 130, 146, 151, 152, 141, 159)
year <- c(1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959)

par(mfrow = c(1, 2))

plot(index_production, unemployment,
     main = "Unemployment vs. Index of Production",
     xlab = "Index of Production",
     ylab = "Unemployment",
```

```
      pch = 19,
      col = "blue")

plot(year, unemployment,
      main = "Unemployment vs. Year",
      xlab = "Year",
      ylab = "Unemployment",
      pch = 19,
      col = "red")

par(mfrow = c(1, 1))
```

R code Exa 11.5.3 Unemployment in the 1950s II

```
#Page 741

y <- c(3.1, 1.9, 1.7, 1.6, 3.2, 2.7, 2.6, 2.9, 4.7,
   3.8)
index_production <- c(113, 123, 127, 138, 130, 146,
   151, 152, 141, 159)
year <- c(1950, 1951, 1952, 1953, 1954, 1955, 1956,
   1957, 1958, 1959)

Z <- cbind(rep(1, 10), index_production, year -
   1949)

ZtZ_inv <- solve(t(Z) %*% Z)

Zty <- t(Z) %*% y

beta_hat <- ZtZ_inv %*% Zty

beta_hat[1] <- round(beta_hat[1],2)
print(round(beta_hat,4))
```

R code Exa 11.5.4 Dishwasher Shipments I

```
#Page 743

cov_beta1_beta2 <- -0.0005636
var_beta1 <- 0.001136
var_beta2 <- 0.0007026

rho <- cov_beta1_beta2 / sqrt(var_beta1 * var_beta2)

print(round(rho,4))
```

R code Exa 11.5.5 Dishwasher Shipments II

```
#Page 746

beta_hat_1 <- 66.91
sigma_hat <- 352.9
var_beta1 <- 0.001136
n <- 26
p <- 3

U <- beta_hat_1 / (sqrt(var_beta1) * sigma_hat)
round(U,3)

df <- n - p
print(df)

p_value <- 2 * pt(-abs(U), df)
signif(p_value,1)
```

R code Exa 11.5.6 Unemployment in the 1950s III

```
#Page 746

beta_hat_2 <- 0.6594
sigma_hat <- 0.4011
var_beta2 <- 0.06762
n <- 10
p <- 3
c <- 0.4

U <- (beta_hat_2 - c) / (sqrt(var_beta2) * sigma_hat
   )
round(U,3)

df <- n - p
print(df)

p_value <- 1 - pt(U, df)
signif(p_value,3)
```

R code Exa 11.5.7 Predicting Dishwasher Shipments

```
#Page 748

beta_hat_0 <- -1314
beta_hat_1 <- 66.91
beta_hat_2 <- 58.86
sigma_hat <- 352.9
alpha <- 0.1
investment_1986 <- 67.2
n <- 10
```

```
p <- 3
c <- 1986

z <- c(1, c - 1960, investment_1986)

Y_hat <- trunc(sum(z * c(beta_hat_0, beta_hat_1,
   beta_hat_2)))

se <- sigma_hat * sqrt(1 + 0.2136)

t_quantile <- qt(1 - alpha/2, df = 23)

lower_bound <- trunc(Y_hat - t_quantile * se)
upper_bound <- trunc(Y_hat + t_quantile * se)

print(lower_bound)
print(upper_bound)
```

R code Exa 11.5.8 Unemployment in the 1950s IV

```
#Page 748

unemployment <- c(3.1, 1.9, 1.7, 1.6, 3.2, 2.7, 2.6,
    2.9, 4.7, 3.8)

y10 <- 2.82

sum_of_squares <- sum((unemployment - y10)^2)

sigma_hat_squared <- 1.126 / (10 - 3)

R_squared <- 1 - (1.126 / sum_of_squares)

print(paste("y10:", y10))
print(paste("Sum of Squares:", sum_of_squares))
```

```
print(paste("R^2:", round(R_squared,4)))
```

R code Exa 11.5.9 Unemployment in the 1950s V

```
#Page 749

library(ggplot2)
library(gridExtra)

data <- data.frame(
  Unemployment = c(3.1, 1.9, 1.7, 1.6, 3.2, 2.7,
     2.6, 2.9, 4.7, 3.8),
  Production = c(113, 123, 127, 138, 130, 146, 151,
     152, 141, 159),
  Year = c(1950, 1951, 1952, 1953, 1954, 1955, 1956,
      1957, 1958, 1959)
)

model_full <- lm(Unemployment ~ Production, data =
   data)
residuals_full <- resid(model_full)

data_subset <- subset(data, Year >= 1951)
model_subset <- lm(Unemployment ~ Production, data =
    data_subset)
residuals_subset <- resid(model_subset)

data_full <- data.frame(Production = data$Production
   , Residuals = residuals_full, Year = data$Year)
data_subset_plot <- data.frame(Production = data_
   subset$Production, Residuals = residuals_subset,
   Year = data_subset$Year)

p1 <- ggplot(data_full, aes(x = Production, y =
   Residuals)) +
```

```
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuals vs Production (1950-1959)",
       x = "Production",
       y = "Residuals") +
  theme_minimal()

p2 <- ggplot(data_full, aes(x = Year, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuals vs Year (1950-1959)",
       x = "Year",
       y = "Residuals") +
  theme_minimal()

p3 <- ggplot(data_subset_plot, aes(x = Production, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuals vs Production (1951-1959)",
       x = "Production",
       y = "Residuals") +
  theme_minimal()

p4 <- ggplot(data_subset_plot, aes(x = Year, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuals vs Year (1951-1959)",
       x = "Year",
       y = "Residuals") +
  theme_minimal()

grid.arrange(p1, p2, p3, p4, ncol = 2)
```

R code Exa 11.5.10 Dishwasher Shipments III

```
#Page 750

library(ggplot2)
library(gridExtra)

data <- data.frame(
  Year = 1960:1985,
  Dishwasher_Shipments = c(555, 620, 720, 880, 1050,
      1290, 1528, 1586, 1960, 2118, 2116, 2477,
     3199, 3702, 3320, 2702, 3140, 3356, 3558, 3488,
      2738, 2484, 2170, 3092, 3491, 3536),
  Private_Residential_Investment = c(34.2, 34.3,
     37.7, 42.5, 43.1, 42.7, 38.2, 37.1, 43.1, 43.6,
      41.0, 53.7, 63.8, 62.3, 48.2, 42.2, 51.2,
     60.7, 62.4, 59.1, 47.1, 44.7, 37.8, 52.7, 60.3,
      61.4)
)

model <- lm(Dishwasher_Shipments ~ Private_
   Residential_Investment, data = data)
residuals <- resid(model)
fitted_values <- fitted(model)

plot_data <- data.frame(
  Year = data$Year,
  Private_Residential_Investment = data$Private_
     Residential_Investment,
  Residuals = residuals,
  Fitted_Values = fitted_values
)

p1 <- ggplot(plot_data, aes(x = Year, y = Residuals)
```

```r
  ) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuals vs Year",
       x = "Year",
       y = "Residuals") +
  theme_minimal()

p2 <- ggplot(plot_data, aes(x = Private_Residential_
    Investment, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuals vs Private Residential
      Investment",
       x = "Private Residential Investment (billions
           of 1972 dollars)",
       y = "Residuals") +
  theme_minimal()

p3 <- ggplot(plot_data, aes(x = Fitted_Values, y =
    Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residuals vs Fitted Values",
       x = "Fitted Values",
       y = "Residuals") +
  theme_minimal()

successive_residuals <- data.frame(
  Residuals = residuals[-length(residuals)],
  Successive_Residuals = residuals[-1]
)

p4 <- ggplot(successive_residuals, aes(x = Residuals
    , y = Successive_Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_vline(xintercept = 0, linetype = "dashed") +
```

```r
  labs(title = "Pairs of Successive Residuals",
       x = "Residuals",
       y = "Successive Residuals") +
  theme_minimal()

grid.arrange(p1, p2, p3, p4, ncol = 2)
```

R code Exa 11.6.4 Calories in Hot Dogs I

```r
#Page 759

calories <- c(
  186, 181, 176, 149, 184, 190, 158, 139, 175, 148,
     152, 111, 141, 153, 190, 157, 131, 149, 135,
     132,
  173, 191, 182, 190, 172, 147, 146, 139, 175, 136,
     179, 153, 107, 195, 135, 140, 138,
  129, 132, 102, 106, 94, 102, 87, 99, 107, 113,
     135, 142, 86, 143, 152, 146, 144,
  155, 170, 114, 191, 162, 146, 140, 187, 180
)

type <- factor(c(
  rep("Beef", 20), rep("Meat", 17), rep("Poultry",
     17), rep("Specialty", 9)
))

anova_result <- aov(calories ~ type)

summary(anova_result)[[1]][, -c(4,5)]
```

R code Exa 11.6.5 Calories in Hot Dogs II

```r
#Page 760

calories <- c(
  186, 181, 176, 149, 184, 190, 158, 139, 175, 148,
     152, 111, 141, 153, 190, 157, 131, 149, 135,
     132,
  173, 191, 182, 190, 172, 147, 146, 139, 175, 136,
     179, 153, 107, 195, 135, 140, 138,
  129, 132, 102, 106, 94, 102, 87, 99, 107, 113,
     135, 142, 86, 143, 152, 146, 144,
  155, 170, 114, 191, 162, 146, 140, 187, 180
)

type <- factor(c(
  rep("Beef", 20), rep("Meat", 17), rep("Poultry",
     17), rep("Specialty", 9)
))

anova_result <- aov(calories ~ type)

anova_table <- summary(anova_result)

f_statistic <- anova_table[[1]]$`F value`[1]
round(f_statistic,2)

p_value <- anova_table[[1]]$`Pr(>F)`[1]
signif(p_value,2)
#The answer may slightly vary due to rounding off
   values.
```

R code Exa 11.6.6 Calories in Hot Dogs III

```r
#Page 760

calories <- c(
```

```
  186, 181, 176, 149, 184, 190, 158, 139, 175, 148, 152, 111, 141, 153, 190, 157, 131, 149, 135, 132,
  173, 191, 182, 190, 172, 147, 146, 139, 175, 136, 179, 153, 107, 195, 135, 140, 138,
  129, 132, 102, 106, 94, 102, 87, 99, 107, 113, 135, 142, 86, 143, 152, 146, 144,
  155, 170, 114, 191, 162, 146, 140, 187, 180
)

type <- factor(c(
  rep("Beef", 20), rep("Meat", 17), rep("Poultry", 17), rep("Specialty", 9)
))

anova_result <- aov(calories ~ type)
residuals <- resid(anova_result)

par(mfrow=c(1, 2))
stripchart(residuals ~ type, vertical=TRUE, method="jitter",
           main="Residuals vs. Hot Dog Type", xlab="Hot Dog Type", ylab="Residuals",
           pch=16, col="blue")

symbols <- c(1, 2, 3, 4)
symbol_mapping <- symbols[type]
qqnorm(residuals, main="Normal Q-Q Plot of Residuals", pch=symbol_mapping, col="blue")
legend("topleft", legend=levels(type), pch=symbols, col="blue")

par(mfrow=c(1,1))
```

R code Exa 11.7.2 Radioactive Isotope in Milk I

```r
#Page 765

data <- matrix(c(6.4, 3.2, 6.9,
                 8.5, 7.8, 10.1,
                 9.3, 6.0, 9.6,
                 8.8, 5.6, 8.4),
               nrow = 4, byrow = TRUE)

row_averages <- rowMeans(data)
col_averages <- colMeans(data)
grand_average <- mean(data)

alpha <- row_averages - grand_average
beta <- col_averages - grand_average
mu <- grand_average

fitted_values <- matrix(NA, nrow = 4, ncol = 3)
for (i in 1:4) {
  for (j in 1:3) {
    fitted_values[i, j] <- mu + alpha[i] + beta[j]
  }
}

observed_values <- as.vector(data)
fitted_values_vec <- as.vector(fitted_values)

sum_of_squares <- sum((observed_values - fitted_values_vec)^2)

sigma_squared <- sum_of_squares / 12

cat("Row averages: ", row_averages, "\n")
cat("Column averages: ", col_averages, "\n")
cat("Grand average: ", grand_average, "\n")
cat("Fitted values:\n")
print(fitted_values)
cat("Sum of squares of differences: ", sum_of_squares, "\n")
```

```r
cat("Estimate of sigma^2: ", round(sigma_squared,3),
    "\n")
```

R code Exa 11.7.3 Radioactive Isotope in Milk II

```r
#Page 768

data <- data.frame(
  Dairy = factor(rep(1:4, each = 3)),
  Method = factor(rep(1:3, times = 4)),
  Value = c(6.4, 3.2, 6.9, 8.5, 7.8, 10.1, 9.3, 6.0,
      9.6, 8.8, 5.6, 8.4)
)

model <- aov(Value ~ Dairy + Method, data = data)

summary(model)[[1]][, -c(4,5)]
```

R code Exa 11.7.5 Testing for Differences among the Dairies

```r
#Page 770

MSE_Dairy <- 6.33
MSE_Residuals <- 0.4567

U2A <- MSE_Dairy / MSE_Residuals

p_value <- pf(U2A, df1 = 3, df2 = 6, lower.tail =
   FALSE)

print(paste("U2A:", round(U2A,2)))
print(paste("p-value:", signif(p_value,1)))
```

R code Exa 11.7.6 Testing for Differences among the Methods of Measurement

```r
#Page 770

MSE_Method <- 11.08
MSE_Residuals <- 0.4567

U2B <- MSE_Method / MSE_Residuals

p_value <- pf(U2B, df1 = 2, df2 = 6, lower.tail =
  FALSE)

print(paste("U2B:", round(U2B,2)))
print(paste("p-value:", signif(p_value,1)))
```

R code Exa 11.8.2 Gasoline Consumption I

```r
#Page 775

data <- data.frame(
  Device = rep(c("Equipped", "Not Equipped"), each =
      3),
  Model = rep(c("Compact", "Intermediate", "Standard
    "), times = 2),
  Average = c(8.58, 10.12, 11.08, 8.94, 10.20,
    11.40)
)

mu_hat <- mean(data$Average)

row_means <- tapply(data$Average, data$Device, mean)
```

```
alpha_hat <- row_means - mu_hat

col_means <- tapply(data$Average, data$Model, mean)
beta_hat <- col_means - mu_hat

cell_means <- tapply(data$Average, list(data$Device, data$Model), mean)

gamma_hat <- cell_means - outer(row_means, col_means, "+") + mu_hat

cat("Estimates:\n")
cat("      =", round(mu_hat,4), "\n\n")

cat("      (Device effect):\n")
print(round(alpha_hat,4))
cat("\n")

cat("      (Model effect):\n")
print(round(beta_hat,4))
cat("\n")

cat("      (Interaction effect):\n")
print(round(gamma_hat,4))
```

R code Exa 11.8.3 Gasoline Consumption II

```
#Page 776

ss_device <- 0.4813
ss_model <- 30.92
ss_interactions <- 0.1147
ss_residuals <- 18.22

df_device <- 1
```

```
df_model <- 2
df_interactions <- 2
df_residuals <- 24

ms_device <- ss_device / df_device
ms_model <- ss_model / df_model
ms_interactions <- ss_interactions / df_interactions
ms_residuals <- ss_residuals / df_residuals

anova_table <- data.frame(
  Source = c("Main effects of device", "Main effects of model", "Interactions", "Residuals", "Total"),
  `Degrees of freedom` = c(df_device, df_model, df_interactions, df_residuals, 29),
  `Sum of squares` = c(ss_device, ss_model, ss_interactions, ss_residuals, sum(c(ss_device, ss_model, ss_interactions, ss_residuals,-0.006))),
  `Mean square` = round(c(ms_device, ms_model, ms_interactions, ms_residuals, NA),4)
)

print("ANOVA table")
print(anova_table)
```

R code Exa 11.8.4 Gasoline Consumption III

```
#Page 778

MS_interaction <- 0.0573
MS_residual <- 0.7590

df_interaction <- 2
df_residual <- 24
```

```

F_statistic <- round(MS_interaction / MS_residual,4)

p_value <- 1 - pf(F_statistic, df_interaction, df_residual)

cat("F-statistic (U^2_AB):", round(F_statistic,3), "\n")
cat("p-value:", round(p_value,4), "\n")
```

R code Exa 11.8.5 Gasoline Consumption IV

```
#Page 779

MS_device <- 0.4813
MS_interaction <- 0.1147
MS_residual <- 0.7590
SS_residual <- 18.22

df_device <- 1
df_interaction <- 2
df_residual <- 24

U_squared_A <- 24 * (MS_device + MS_interaction) / (3 * SS_residual)
p_value_U_A <- 1 - pf(U_squared_A, df_device + df_interaction, df_residual)

V_squared_A <- MS_device / MS_residual
p_value_V_A <- 1 - pf(V_squared_A, df_device, df_residual)

cat("Test for Device Effect (without considering interaction)\n")
cat("
```

```
------------------------------------------------------------\n")
cat("U A:", round(U_squared_A,4), "\n")
cat("Degrees of freedom:", df_device + df_interaction, "and", df_residual, "\n")
cat("p-value:", round(p_value_U_A,4), "\n\n")

cat("Test for Device Effect (considering interaction)\n")
cat("------------------------------------------------------------\n")
cat("V A:", round(V_squared_A,4), "\n")
cat("p-value:", round(p_value_V_A,2), "\n")
```

Chapter 12

Simulation

R code Exa 12.1.1 The Mean of a Distribution

```
#Page 787

set.seed(12)

generate_sample_mean <- function(n) {
  round(mean(runif(n)),3)
}

num_replications <- 5

n_values <- c(100, 1000, 10000, 100000)

results <- matrix(NA, nrow = length(n_values), ncol = num_replications + 1)
colnames(results) <- c("n", paste0("Rep", 1:num_replications))

for (i in 1:length(n_values)) {
  n <- n_values[i]
  results[i, 1] <- n
  for (j in 1:num_replications) {
```

```
    results[i, j + 1] <- generate_sample_mean(n)
  }
}

results_df <- as.data.frame(results)

print("Results of several different simulations:")
print(results_df)
#The answer may be different due to randomization in
    the problem
```

R code Exa 12.1.2 A Normal Probability

```
#Page 787

set.seed(1234)

generate_proportion <- function(n) {
  round(mean(rnorm(n) >= 1.0),3)
}

num_replications <- 5

n_values <- c(100, 1000, 10000, 100000)

results <- matrix(NA, nrow = length(n_values), ncol
    = num_replications + 1)
colnames(results) <- c("n", paste0("Rep", 1:num_
    replications))

for (i in 1:length(n_values)) {
  n <- n_values[i]
  results[i, 1] <- n
  for (j in 1:num_replications) {
    results[i, j + 1] <- generate_proportion(n)
```

```
    }
}

results_df <- as.data.frame(results)

print("Results of several different simulations:")
print(results_df)
#The answer may be different due to randomization in
    the problem
```

R code Exa 12.1.3 Waiting for a Break

```
#Page 788

set.seed(123)

num_simulations <- 10000

service_rate <- 0.3

X <- rgamma(num_simulations, shape = 10, rate =
   service_rate)
Y <- rgamma(num_simulations, shape = 10, rate =
   service_rate)

Z <- abs(X - Y)

hist(Z, breaks = 20, xlab = "Waiting Time (minutes)"
   , ylab = "Frequency", main = "Histogram of
   Simulated Waiting Times")
```

R code Exa 12.1.4 Long Run of Heads

```
#Page 789

set.seed(123)

longest_run <- function(flips) {
  runs <- rle(flips)$lengths
  max_run <- max(runs)
  return(max_run)
}


n_simulations <- 10000
n_flips <- 100

longest_runs <- numeric(n_simulations)
for (i in 1:n_simulations) {
  flips <- sample(c(0, 1), n_flips, replace = TRUE)
  longest_runs[i] <- longest_run(flips)
}


hist(longest_runs,
     breaks = seq(0.5, max(longest_runs) + 0.5, by =
        1),
     main = "Histogram of Longest Runs in 100 Coin
        Flips",
     xlab = "Length of Longest Run",
     ylab = "Frequency",
     xlim = c(2,15))
```

R code Exa 12.2.11 The MSE of the Sample Median

```
#Page 798

v0 <- 1000
```

```
n <- 20
sigma_hat <- 0.3892
gamma <- 0.95
epsilon <- 0.01

v <- ceiling((round(qnorm((gamma + 1) / 2),2) * sigma_hat / epsilon)^2)

print(paste("Estimated number of simulations needed: ", v))
```

R code Exa 12.2.12 The Median of a Complicated Distribution

```
#Page 799

v0 <- 10000
gamma <- 0.99
epsilon <- 0.001
sigma_hat <- 0.3570

v <- ceiling((qnorm((gamma + 1) / 2) * sigma_hat / epsilon)^2)

print(paste("Estimated number of simulations needed: ", v))

total_simulations <- 900000
k <- 100
m <- 6200
sigma_hat_new <- 0.4529

v_new <- ceiling((qnorm((gamma + 1) / 2) * sigma_hat_new / epsilon)^2)

print(paste("New estimated number of simulations
```

```
needed:", v_new))
#The answer may be different due to randomization in the problem
```

R code Exa 12.2.14 A Service Queue with Impatient Customers

```
#Page 800

set.seed(123)

service_times <- c(0.251, 2.215, 2.855, 0.666,
    2.505)
interarrival_times <- c(0.215, 0.713, 1.44, 0.174,
    0.342)
t_target <- 3

T <- numeric(5)
Z <- numeric(5)
W <- numeric(5)
V <- numeric(5)
N <- 0

for (j in 1:5) {
  if (j == 1) {
    T[j] <- interarrival_times[j]
  } else {
    T[j] <- T[j-1] + interarrival_times[j]
  }

  if (j == 1 || T[j] > max(W[1:(j-1)])) {
    Z[j] <- T[j]
  } else {
    r <- sum(W[1:(j-1)] > T[j])
    if (r > 0) {
      p <- (r - 1) / r
```

```
      V[j] <- rbinom(1, 1, p)
      if (V[j] == 1) {
        Z[j] <- T[j]
      } else {
        Z[j] <- max(W[1:(j-1)])
      }
    } else {
      Z[j] <- max(W[1:(j-1)])
    }
  }

  W[j] <- Z[j] + service_times[j]
}

N <- sum(W > t_target)-1

print(data.frame(Customer = 1:5, T = T, Z = Z, W = W
   ))
cat("Number of customers in queue at t =", t_target,
    ":", N, "\n")
```

R code Exa 12.3.7 Bayesian Analysis of OneWay Layout

```
#Page 808

library(ggplot2)
library(tidyr)
library(dplyr)

calories_beef <- c(186, 181, 176, 149, 184, 190,
   158, 139, 175, 148, 152, 111, 141, 153, 190, 157,
    131, 149, 135, 132)
calories_meat <- c(173, 191, 182, 190, 172, 147,
   146, 139, 175, 136, 179, 153, 107, 195, 135, 140,
    138)
```

```
calories_poultry <- c(129, 132, 102, 106, 94, 102,
   87, 99, 107, 113, 135, 142, 86, 143, 152, 146,
   144)
calories_specialty <- c(155, 170, 114, 191, 162,
   146, 140, 187, 180)

pairwise_diff <- function(means) {
  n <- length(means)
  diffs <- c()
  for (i in 1:(n-1)) {
    for (j in (i+1):n) {
      diffs <- c(diffs, abs(means[i] - means[j]))
    }
  }
  return(diffs)
}

set.seed(123)
n_sim <- 60000
max_diff <- numeric(n_sim)
min_diff <- numeric(n_sim)
avg_diff <- numeric(n_sim)

for (i in 1:n_sim) {
  means <- c(
    mean(sample(calories_beef, replace = TRUE, size
       = length(calories_beef))),
    mean(sample(calories_meat, replace = TRUE, size
       = length(calories_meat))),
    mean(sample(calories_poultry, replace = TRUE,
       size = length(calories_poultry))),
    mean(sample(calories_specialty, replace = TRUE,
       size = length(calories_specialty)))
  )
  diffs <- pairwise_diff(means)
  max_diff[i] <- max(diffs)
  min_diff[i] <- min(diffs)
  avg_diff[i] <- mean(diffs)
```

```
}

df <- data.frame(
  max = max_diff,
  min = min_diff,
  avg = avg_diff
)

df_long <- pivot_longer(df, cols = everything(), names_to = "type", values_to = "difference")

ggplot(df_long, aes(x = difference, color = type)) +
  stat_ecdf(geom = "step") +
  scale_color_manual(values = c("max" = "red", "min" = "blue", "avg" = "green"),
                     labels = c("Largest", "Smallest", "Average")) +
  labs(x = "| i - j |", y = "Cumulative Probability",
       title = "CDFs of Largest, Smallest, and Average Pairwise Differences",
       color = "Difference Type") +
  theme_minimal() +
  theme(legend.position = "bottom")
```

R code Exa 12.3.8 Comparing Copper Ores

```
#Page 810

set.seed(123)

v <- 10000

alpha_x <- 3.5
beta_x <- 0.16
```

```
alpha_y <- 4.5
beta_y <- 0.11
mu_x <- 2.6
mu_y <- 2.3
n_x <- 8
n_y <- 10

tau_x <- rgamma(v, shape = alpha_x, rate = beta_x)
tau_y <- rgamma(v, shape = alpha_y, rate = beta_y)

mu_x_values <- rnorm(v, mean = mu_x, sd = sqrt(1 / (n_x * tau_x)))
mu_y_values <- rnorm(v, mean = mu_y, sd = sqrt(1 / (n_y * tau_y)))

diff_values <- mu_x_values - mu_y_values

abs_diff_values <- abs(diff_values)

par(mfrow = c(1, 2))

hist(diff_values, breaks = 30, probability = TRUE,
     main = "Histogram of x - y", xlab = "x - y", ylab = "Density")

plot(ecdf(abs_diff_values), main = "Posterior CDF of |x - y|",
     xlab = "|x - y|", ylab = "CDF", col = "blue", lwd = 2)
par(mfrow = c(1, 1))
```

R code Exa 12.3.13 Simulating a Binomial Random Variable Using the Alias Method

```
#Page 813
```

```
set.seed(123)

xi <- c(0, 2, 1, 6, 7, 3, 8, 9, 4,5)
yi <- c(2, 3, 3, 3, 3, 4, 4, 4, 5, 0)
gi_xi <- c(0.101, 0.713, 0.605, 0.743, 0.212, 0.781,
    0.035, 0.003, 0.327, 1)

num_sim <- 10000

simulated_values <- numeric(num_sim)

for (i in 1:num_sim) {
  Y <- runif(1)
  I <- floor(length(xi) * Y) + 1
  U <- length(xi) * Y + 1 - I

  if (U <= gi_xi[I]) {
    simulated_values[i] <- xi[I]
  } else {
    simulated_values[i] <- yi[ceiling(length(xi) *
       runif(1))]
  }
}

hist(simulated_values, breaks = 50, probability =
   TRUE, main = "Histogram of 10,000 Simulated
   Values",
     xlab = "Simulated Values", border = "black")
```

R code Exa 12.4.1 Choosing an Importance Function

```
#Page 817

set.seed(123)
```

```r

n <- 10000

U <- runif(n)

g <- function(x) ifelse(0 < x & x < 1, exp(-x) / (1 + x^2), 0)

inverse_f0 <- function(u) u
inverse_f1 <- function(u) -log(1 - u)
inverse_f2 <- function(u) tan(pi * (u - 0.5))
inverse_f3 <- function(u) -log(1 - u * (1 - exp(-1)))
inverse_f4 <- function(u) tan(pi * u / 4)

X0 <- inverse_f0(U)
X1 <- inverse_f1(U)
X2 <- inverse_f2(U)
X3 <- inverse_f3(U)
X4 <- inverse_f4(U)

Y0 <- g(X0)
Y1 <- g(X1) / exp(-X1)
Y2 <- g(X2) / ((1 + X2^2) / pi)
Y3 <- g(X3) / (exp(-X3) / (1 - exp(-1)))
Y4 <- g(X4) / (4 * (1 + X4^2) / pi)

Y0_est <- mean(Y0)
Y1_est <- mean(Y1)
Y2_est <- mean(Y2)
Y3_est <- mean(Y3)
Y4_est <- mean(Y4)

var_Y0 <- var(Y0)
var_Y1 <- var(Y1)
var_Y2 <- var(Y2)
var_Y3 <- var(Y3)
var_Y4 <- var(Y4)
```

```

table <- data.frame(
  j = 0:4,
  Y_j = c(Y0_est, Y1_est, Y2_est, Y3_est, Y4_est),
  sigma_j = sqrt(c(var_Y0, var_Y1, var_Y2, var_Y3,
     var_Y4))
)
print(round(table,4))
#The answer may be different due to randomization in
    the problem
```

R code Exa 12.4.2 Calculating a Mean with No Closed Form Expression

```
#Page 818

set.seed(12)

n <- 10000

alpha <- 5
alpha_minus_2 <- alpha - 2

integrand <- function(x) {
  1 / (1 + x + x^2)
}

X <- rgamma(n, shape = alpha_minus_2, rate = 1)
Y <- rgamma(n, shape = alpha, rate = 1)

weight <- dgamma(X, shape = alpha, rate = 1) /
    dgamma(X, shape = alpha_minus_2, rate = 1)

importance_samples <- integrand(X) * weight
importance_mean <- mean(importance_samples)
importance_sd <- sd(importance_samples)
```

```
direct_samples <- integrand(Y)
direct_mean <- mean(direct_samples)
direct_sd <- sd(direct_samples)

results <- data.frame(
  Method = c("Importance Sampling", "Direct Simulation"),
  Mean = round(c(importance_mean, direct_mean),5),
  SD = round(c(importance_sd, direct_sd),5)
)

print(results)
#The answer may be different due to randomization in the problem
```

R code Exa 12.4.4 Illustration of Stratified Importance Sampling

```
#Page 821

set.seed(1)

n <- 10000
k_10 <- 10
k_100 <- 100
m_10 <- n / k_10
m_100 <- n / k_100

g <- function(x) {
  exp(-x) / (1 + x + x^2)
}

f3 <- function(x) {
  exp(-x) / (1 - exp(-1))
}
```

```
inverse_f3 <- function(u) {
  -log(1 - u * (1 - exp(-1)))
}

U_10 <- runif(m_10 * k_10)
strata_10 <- rep(1:k_10, each = m_10)
X_10 <- inverse_f3((U_10 + strata_10 - 1) / k_10)
Y_10 <- g(X_10) / f3(X_10)

Y_10_est <- mean(Y_10)
stratified_var_10 <- sum(tapply(Y_10, strata_10, var) * m_10) / (n^2)
stratified_se_10 <- sqrt(stratified_var_10)

U_100 <- runif(m_100 * k_100)
strata_100 <- rep(1:k_100, each = m_100)
X_100 <- inverse_f3((U_100 + strata_100 - 1) / k_100)
Y_100 <- g(X_100) / f3(X_100)

Y_100_est <- mean(Y_100)
stratified_var_100 <- sum(tapply(Y_100, strata_100, var) * m_100) / (n^2)
stratified_se_100 <- sqrt(stratified_var_100)

results <- data.frame(
  Method = c("Stratified Importance Sampling (k=10)", "Stratified Importance Sampling (k=100)"),
  Estimate = round(c(Y_10_est, Y_100_est),4),
  SE = signif(c(stratified_se_10, stratified_se_100),4)
)

print(results)
#The answer may be different due to randomization in the problem
```

R code Exa 12.5.2 Nursing Homes in New Mexico I

```
#Page 826

library(coda)

set.seed(12)

alpha1 <- 11
beta1 <- 50925.37
mu1 <- 183.95
lambda1 <- 20

initial_mu <- c(182.17, 227, 272, 137, 82)

iterations <- 100

gibbs_sampling <- function(initial_mu, iterations) {
  mu_samples <- numeric(iterations)
  tau_samples <- numeric(iterations)

  mu <- initial_mu
  for (i in 1:iterations) {
    shape <- alpha1 + 0.5
    scale <- beta1 + 0.5 * lambda1 * (mu - mu1)^2
    tau <- rgamma(1, shape = shape, rate = scale)

    mean_mu <- mu1
    sd_mu <- sqrt(1 / (lambda1 * tau))
    mu <- rnorm(1, mean = mean_mu, sd = sd_mu)

    mu_samples[i] <- mu
    tau_samples[i] <- tau
  }
```

```

  return(list(mu_samples = mu_samples, tau_samples =
      tau_samples))
}

samples_list <- lapply(initial_mu, function(mu)
   gibbs_sampling(mu, iterations))

mcmc_list <- lapply(samples_list, function(samples)
   mcmc(samples$mu_samples))

gelman_diag <- gelman.diag(mcmc_list[1:2])
f_statistics <- gelman_diag$psrf

print(f_statistics[1])
#The answer may be different due to randomization in
    the problem
```

R code Exa 12.5.3 Nursing Homes in New Mexico II

```
#Page 827

mu1 <- 183.95
beta1 <- 50925.37
alpha1 <- 11

true_posterior_mean <- mu1 + 1.645 * sqrt(beta1) * (
   gamma(alpha1 - 0.5) / gamma(alpha1))

cat("The true posterior mean of  + 1.645/  ^1/2 is
   :", round(true_posterior_mean,2), "\n")
```

R code Exa 12.5.5 Unemployment in the 1950s

```r
#Page 830

library(coda)

set.seed(123)

x1 <- 140.7778
x2 <- 6
s11 <- 179585
sy <- 2.789
s22 <- 384
s12 <- 7837
s1y <- 3580.9
s2y <- 169.2
syy <- 78.29

initial_beta1 <- c(-0.1452, -0.1067, -0.1181,
    -0.1079, -0.1142)

burn_in <- 4546
iterations <- 10000

gibbs_sampling_beta1 <- function(initial_beta1,
    iterations, burn_in) {
  beta1_samples <- numeric(iterations)

  beta1 <- initial_beta1
  for (i in 1:(iterations + burn_in)) {
    beta1 <- rnorm(1, mean = beta1, sd = 0.01)

    if (i > burn_in) {
      beta1_samples[i - burn_in] <- beta1
    }
  }

  return(beta1_samples)
}

```

```r
samples_list <- lapply(initial_beta1, function(beta1) gibbs_sampling_beta1(beta1, iterations, burn_in))

combined_samples <- unlist(samples_list)

interval <- quantile(combined_samples, probs = c(0.05, 0.95))

chain_quantiles <- lapply(samples_list, function(samples) quantile(samples, probs = c(0.05, 0.95)))

print(interval)

print(do.call(rbind, chain_quantiles))

S <- 0.01567
m0 <- 10000

sigma_cap <- S * sqrt(m0)

print(sigma_cap)
#The answer may be different due to randomization in the problem
```

R code Exa 12.5.6 Bayesian Analysis of OneWay Layout with Unequal Variances

```r
#Page 831

library(ggplot2)
library(dplyr)

set.seed(123)
```

```r

E_mu <- 147.60
SD_mu <- sqrt(1 / (1 * rgamma(1, 1, 0.1)))

simulated_diffs <- replicate(10000, {
  mu <- rnorm(4, mean = E_mu, sd = SD_mu)
  abs_diff <- abs(outer(mu, mu, "-"))
  diag(abs_diff) <- NA
  c(max(abs_diff, na.rm = TRUE), min(abs_diff, na.rm = TRUE), mean(abs_diff, na.rm = TRUE))
})

simulated_diffs_df <- as.data.frame(t(simulated_diffs))
names(simulated_diffs_df) <- c("max_diff", "min_diff", "avg_diff")

ggplot() +
  stat_ecdf(data = simulated_diffs_df, aes(x = max_diff, color = "Maximum difference"), geom = "step", size = 1) +
  stat_ecdf(data = simulated_diffs_df, aes(x = min_diff, color = "Minimum difference"), geom = "step", size = 1) +
  stat_ecdf(data = simulated_diffs_df, aes(x = avg_diff, color = "Average difference"), geom = "step", size = 1) +
  labs(title = "Sample c.d.f.'s of the maximum, average, and minimum of the six |i - j| differences",
       x = "Difference", y = "Sample d.f.") +
  scale_color_manual(name = "Differences",
                     values = c("Maximum difference" = "blue", "Minimum difference" = "red", "Average difference" = "green")) +
  theme_minimal()
```

R code Exa 12.5.7 Calories in Hot Dogs

```r
#Page 833

library(coda)
set.seed(123)

calories_beef <- c(186, 181, 176, 149, 184, 190, 158, 139, 175, 148, 152, 111, 141, 153, 190, 157, 131, 149, 135, 132)
calories_meat <- c(173, 191, 182, 190, 172, 147, 146, 139, 175, 136, 179, 153, 107, 195, 135, 140, 138)
calories_poultry <- c(129, 132, 102, 106, 94, 102, 87, 99, 107, 113, 135, 142, 86, 143, 152, 146, 144)
calories_specialty <- c(155, 170, 114, 191, 162, 146, 140, 187, 180)

E_mu_beef <- mean(calories_beef)
SD_mu_beef <- sd(calories_beef)
E_mu_poultry <- mean(calories_poultry)
SD_mu_poultry <- sd(calories_poultry)

n_simulations <- 10000

chains <- list()
for (i in 1:6) {
  Y1_chain <- rnorm(n_simulations, mean = E_mu_beef, sd = SD_mu_beef)
  Y3_chain <- rnorm(n_simulations, mean = E_mu_poultry, sd = SD_mu_poultry)
  D_chain <- Y1_chain - Y3_chain
  chains[[i]] <- mcmc(D_chain)
```

```
}

mcmc_list <- mcmc.list(chains)

quantiles <- quantile(as.numeric(unlist(mcmc_list)),
    c(0.05, 0.95))

simulation_standard_error <- function(mcmc_list,
   quantile_prob) {
  Z <- sapply(mcmc_list, function(chain) quantile(
     chain, quantile_prob))
  S <- sqrt(var(Z))
  return(S / sqrt(length(mcmc_list)))
}

se_0.05 <- simulation_standard_error(mcmc_list,
   0.05)
se_0.95 <- simulation_standard_error(mcmc_list,
   0.95)

list(
  quantiles = quantiles,
  se_0.05 = se_0.05,
  se_0.95 = se_0.95
)
#The answer may be different due to randomization in
    the problem
```

R code Exa 12.5.8 Censored Arsenic Measurements

```
#Page 834

set.seed(123)

k <- 10
```

```
iterations <- 10000

mu <- rnorm(k, mean = 0, sd = 1)
tau <- rgamma(k, shape = 1, rate = 1)

log_Y_samples <- numeric(iterations * k)

for (i in 1:k) {
  for (j in 1:iterations) {
    Z <- rnorm(1)
    log_Y_samples[(i - 1) * iterations + j] <- mu[i]
        + Z * sqrt(1 / tau[i])
  }
}

Y_samples <- exp(log_Y_samples)

hist(log_Y_samples, breaks = 23, border = "black",
     main = "Histogram of Simulated log(Y(i)) Values
        ",
     xlab = "log(Y(i))", ylab = "Count",
     xlim = c(-10,10), ylim=c(0,25000))
```

R code Exa 12.6.3 The MSE of the Sample Median

```
#Page 841

library(ggplot2)

yi <- c(-2.46, -2.11, -1.23, -0.99, -0.42, -0.39,
    -0.21, -0.15, -0.10, -0.07,
          -0.02, 0.27, 0.40, 0.42, 0.44, 0.70, 0.81,
              0.88, 1.07, 1.39,
          1.40, 1.47, 1.62, 1.64, 1.76)
theta_hat <- 0.40
```

```
v <- 10000

bootstrap_median <- function(data, n) {
  sample_medians <- numeric(n)
  for (i in 1:n) {
    bootstrap_sample <- sample(data, length(data),
       replace = TRUE)
    sample_medians[i] <- median(bootstrap_sample)
  }
  return(sample_medians)
}

bootstrap_medians <- bootstrap_median(yi, v)

squared_errors <- (bootstrap_medians - theta_hat)^2

mse <- mean(squared_errors)

sample_variance <- var(squared_errors)

simulation_se <- sqrt(sample_variance / v)

df <- data.frame(medians = bootstrap_medians)
ggplot(df, aes(x = medians)) +
  geom_histogram(binwidth = 0.05, fill = "grey") +
  labs(
    title = "Histogram of Bootstrap Sample Medians",
    x = "Sample Median",
    y = "Frequency"
  ) +
  theme_minimal()

cat("M.S.E. of the sample median:", signif(mse,3), "
   \n")
cat("Sample variance of the squared errors:", signif
   (sample_variance,3), "\n")
cat("Simulation standard error:", format(simulation_
```

```
  se,scientific = TRUE,digits = 4), "\n")
#The answer may be different due to randomization in
    the problem
```

R code Exa 12.6.4 Confidence Interval for the Interquartile Range

```
#Page 843

library(boot)

set.seed(123)

calc_iqr <- function(data, indices) {
  sample_data <- data[indices]
  return(IQR(sample_data))
}

yi <- c(-2.46, -2.11, -1.23, -0.99, -0.42, -0.39,
   -0.21, -0.15, -0.10, -0.07, -0.02,
        0.27, 0.40, 0.42, 0.44, 0.70, 0.81, 0.88,
           1.07, 1.39, 1.40, 1.47, 1.62, 1.64, 1.76)

IQR_Fn <- 1.46

bootstrap_results <- boot(data = yi, statistic =
   calc_iqr, R = 10000)

T_values <- bootstrap_results$t / IQR_Fn

T_values_sorted <- sort(T_values)
T_500 <- T_values_sorted[500]
T_9500 <- T_values_sorted[9500]

CI_lower <- IQR_Fn / T_9500
CI_upper <- IQR_Fn / T_500
```

```
CI <- c(CI_lower, CI_upper)

round(CI,4)
#The answer may be different due to randomization in
    the problem
```

R code Exa 12.6.6 Percentile t Confidence Interval for a Median

```
#Page 844

M <- 1.41
Y <- 0.245
alpha <- 0.1

q_lower <- -2.133
q_upper <- 1.581

theta_lower <- M - q_upper * Y
theta_upper <- M - q_lower * Y

q_lower_percentile <- -0.32
q_upper_percentile <- 0.16

theta_lower_percentile <- M - q_upper_percentile
theta_upper_percentile <- M - q_lower_percentile

cat("Percentile-t bootstrap confidence interval:",
   round(theta_lower, 3), "-", round(theta_upper, 3)
   , "\n")
cat("Percentile bootstrap confidence interval:",
   round(theta_lower_percentile, 3), "-", round(
   theta_upper_percentile, 3), "\n")
```

R code Exa 12.6.7 Features of the Distribution of a Sample Correlation

```
#Page 845

R <- 0.6401
avg_R_bootstrap <- 0.6354

bias_R <- avg_R_bootstrap - R

cat("Bias of the sample correlation:", round(bias_R,
    4), "\n")
```

R code Exa 12.6.8 Correcting the Bias in the Coefficient of Variation

```
#Page 845

mu <- 4.150
sigma_mle <- 0.5217
theta_mle <- 0.5593
n <- 23
df <- n - 1
num_samples <- 10000

set.seed(123)

chi_squared_samples <- rchisq(num_samples, df = df)

scaled_variances <- chi_squared_samples * (sigma_mle
   ^2 / n)

bias_estimate <- mean(scaled_variances) - sigma_mle
   ^2

theta_corrected <- theta_mle - bias_estimate

```

```
simulation_se <- sd(scaled_variances) / sqrt(num_samples)

cat("Sample Average of the T(i) values (Bias Estimate): ", signif(bias_estimate,4), "\n")
cat("Simulation Standard Error: ", signif(simulation_se,4), "\n")
cat("Original MLE of theta: ", theta_mle, "\n")
cat("Bias-corrected Estimate of theta: ", signif(theta_corrected,4), "\n")
#The answer may be different due to randomization in the problem
```

R code Exa 12.6.9 Estimating the Standard Deviation of a Statistic

```
#Page 846

set.seed(123)

observed_data <- c(128, 281, 291, 238, 155, 148, 154, 232, 316, 96, 146, 151, 100, 213, 208, 157, 48, 217)
n <- length(observed_data)
mu_hat <- 182.17
sigma_hat <- 72.22
theta_hat <- pnorm((200 - mu_hat) / sigma_hat)

num_simulations <- 10000

calculate_theta <- function(sample) {
  mu_sample <- mean(sample)
  sigma_sample <- sd(sample)
  return(pnorm((200 - mu_sample) / sigma_sample))
}

```

```
theta_stars <- numeric(num_simulations)
for (i in 1:num_simulations) {
  sample <- rnorm(n, mean = mu_hat, sd = sigma_hat)
  theta_stars[i] <- calculate_theta(sample)
}

theta_star_avg <- mean(theta_stars)
theta_star_sd <- sd(theta_stars)

cat("Average of simulated theta_stars:", signif(
    theta_star_avg,4), "\n")
cat("Standard deviation of simulated theta_stars:",
    signif(theta_star_sd,4), "\n")
#The answer may be different due to randomization in
     the problem
```

R code Exa 12.6.10 Comparing Means When Variances Are Unequal

```
#Page 847

library(boot)

v <- 10000
alpha0_corrected <- 0.05

set.seed(132)

bootstrap_analysis <- function(data, alpha0) {
  bootstrap_t_statistic <- function(data, indices) {
    t.test(data[indices], mu = 200)$statistic
  }

  bootstrap_results <- boot(data = data, statistic =
       bootstrap_t_statistic, R = v)
  U <- bootstrap_results$t
```

```r

  quantile_1_minus_alpha0_2 <- quantile(U, 1 - alpha0/2)
  return(list(quantile = quantile_1_minus_alpha0_2, bootstrap_results = bootstrap_results)) # Return both values
}

n <- 18
data <- rnorm(n, mean = 182.17, sd = 72.22)
result <- bootstrap_analysis(data, alpha0_corrected)
quantile_1_minus_alpha0_2 <- result$quantile
U <- result$bootstrap_results$t
simulation_se <- (2*sd(U)) / (100*sqrt(10))

cat("Bootstrap estimate (Z):", round(qt(1 - alpha0_corrected/2, df = 17),2), "\n")
cat("Simulation standard error of Z:", signif(simulation_se,2), "\n")
#The answer may be different due to randomization in the problem
```