

R Textbook Companion for
Probability And Statistics
by Morris H. Degroot, Mark J. Schervish¹

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

July 26, 2024

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

Book Description

Title: 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	4
1 Introduction to probability	5
2 Conditional Probability	26
3 Random Variables and Distributions	40
4 Expectation	77
5 Special Distributions	106
6 Large Random Samples	119
7 Estimation	130
8 Sampling Distributions of Estimators	143
9 Testing Hypotheses	156
10 Categorical Data and Nonparametric Methods	184
11 Linear Statistical Models	197
12 Simulation	229

List of R Codes

Exa 1.4.1	Rolling a Die	5
Exa 1.4.2	Subset	5
Exa 1.4.3	Complement	6
Exa 1.5.2	A loaded die	6
Exa 1.5.3	Diagnosing diseases	7
Exa 1.5.4	Demands for utilities	7
Exa 1.6.2	Fiber breaks	8
Exa 1.6.3	Tossing coins	9
Exa 1.6.4	Genetics	9
Exa 1.6.5	Rolling two dice	10
Exa 1.7.1	Routes between cities	11
Exa 1.7.3	Possible outcomes of two dice	11
Exa 1.7.4	Tossing several coins	11
Exa 1.7.5	Combination lock	12
Exa 1.7.7	Current population survey	13
Exa 1.7.8	Choosing officers	13
Exa 1.7.9	Arranging books	13
Exa 1.7.12	Approximating the number of permutations	14
Exa 1.8.2	Selecting a committee	14
Exa 1.8.3	Choosing jobs	15
Exa 1.8.4	Blood types	15
Exa 1.8.5	Selecting baked goods	16
Exa 1.8.6	Selecting baked goods from boxes	16
Exa 1.8.7	Tossing a coin	17
Exa 1.8.8	Sampling without replacement	18
Exa 1.8.9	Playing cards	18
Exa 1.8.10	Playing cards revisited	19
Exa 1.8.11	Lottery ticket	20

Exa 1.9.1	Choosing committees	20
Exa 1.9.2	Multinomial coefficient	21
Exa 1.9.3	Rolling dice	21
Exa 1.9.4	Distributing heart cards	22
Exa 1.10.1	Student enrollment	23
Exa 2.1.2	Lottery ticket	26
Exa 2.1.3	Rolling dice	27
Exa 2.1.4	A clinical trial	27
Exa 2.1.5	Rolling dice repeatedly	28
Exa 2.1.9	Selecting bolts	29
Exa 2.1.10	Achieving a high score	29
Exa 2.1.12	A clinical trial II	30
Exa 2.2.1	Tossing coins	30
Exa 2.2.2	Machine operation	31
Exa 2.2.3	Rolling a Die	31
Exa 2.2.4	Pairwise independence	32
Exa 2.2.7	Tossing a coin until a head appears	33
Exa 2.2.9	People v Collins	33
Exa 2.2.10	Inspecting items	34
Exa 2.3.2	Selecting bolts II	34
Exa 2.3.3	Test for a disease	35
Exa 2.3.4	Identifying the source of a defective item	35
Exa 2.3.5	Identifying genotypes	36
Exa 2.3.6	Learning about a proportion	36
Exa 2.3.7	A clinical trial III	37
Exa 2.4.1	The probability of winning in a fair game	38
Exa 2.4.2	The probability of winning in an unfavorable game	39
Exa 3.1.5	Demands for utilities	40
Exa 3.1.8	Daily Numbers	40
Exa 3.2.3	Incompletely Specified PDF	41
Exa 3.2.4	Probabilities from a PDF	41
Exa 3.2.5	Unbounded Random Variables	42
Exa 3.2.7	Truncated Voltage	42
Exa 3.3.1	Voltage I	43
Exa 3.3.3	Voltage II	43
Exa 3.3.9	Quantiles of a Binomial Distribution	44
Exa 3.3.10	Uniform Distribution on Integers	44
Exa 3.4.3	Discrete Joint Distribution	45

Exa 3.4.7	Normalizing Constant	46
Exa 3.4.8	Probability from a Joint PDF	47
Exa 3.4.9	Joint PDF by Geometric Methods	47
Exa 3.4.11	Joint PDF	48
Exa 3.4.12	A Clinical Trial I	49
Exa 3.5.2	Marginal PF	50
Exa 3.5.4	Marginal PDF from a Joint PDF	50
Exa 3.5.7	Games of Chance	51
Exa 3.5.8	Independency of Random Variable	52
Exa 3.5.9	Probability involving Independent Random Variable	52
Exa 3.5.11	Factorization of a Joint PDF	53
Exa 3.6.1	Auto Insurance	54
Exa 3.6.2	Conditional PF from a Joint PF	55
Exa 3.6.5	Processing Times	55
Exa 3.6.6	Conditional PDF from a Joint PDF	56
Exa 3.6.9	Defective Parts	56
Exa 3.6.10	Uniform Distributions	57
Exa 3.7.7	Arrivals at a Queue	57
Exa 3.7.8	A Clinical Trial II	58
Exa 3.7.9	Service Times in a Queue I	58
Exa 3.7.14	Service Times in a Queue II	59
Exa 3.7.15	Marginal Bivariate PDF	60
Exa 3.7.16	Service Times in a Queue III	61
Exa 3.7.19	Rate of Service I	61
Exa 3.7.20	Rate of Service II	62
Exa 3.8.1	Distance from the Middle I	62
Exa 3.8.2	Distance from the Middle II	63
Exa 3.8.4	PDF of square of Uniform Distribution	64
Exa 3.8.6	Generating Independent Values from a Specified PDF	65
Exa 3.9.2	Bull Market	65
Exa 3.9.5	An Investment Portfolio	66
Exa 3.9.6	Maximum and Minimum of a Random Sample	66
Exa 3.10.2	Shopping for Toothpaste I	67
Exa 3.10.4	Shopping for Toothpaste II	68
Exa 3.10.5	Number of Occupied Telephone Lines I	68
Exa 3.10.6	Plant Breeding Experiment I	69
Exa 3.10.7	Single Server Queue I	70
Exa 3.10.8	Number of Occupied Telephone Lines II	71

Exa 3.10.9 Plant Breeding Experiment II	72
Exa 3.10.10Single Server Queue II	73
Exa 3.10.11Number of Occupied Telephone Lines III	73
Exa 3.10.12Distribution for Telephone Lines	74
Exa 3.10.14Toothpaste Shopping	74
Exa 3.10.15 Alternating Chain	75
Exa 4.1.2 Stock Price Change	77
Exa 4.1.6 Expected Failure Time	77
Exa 4.1.7 Failure after Warranty	78
Exa 4.1.9 The Cauchy Distribution I	78
Exa 4.1.11 Failure Rate and Time to Failure I	79
Exa 4.1.12 Failure Rate and Time to Failure II	80
Exa 4.1.13 Expectation of square root of X	80
Exa 4.1.14 Option Pricing	81
Exa 4.1.16 Expectation of a Function of Two Variables	81
Exa 4.2.1 Expectation of a Linear Function	82
Exa 4.2.2 Investment I	82
Exa 4.2.3 Investment Portfolio I	83
Exa 4.2.6 Expected Number of Matches	83
Exa 4.2.7 Expectation of a Combination of Random Variables	84
Exa 4.2.8 Repeated Filtering	84
Exa 4.2.10 Expected Waiting Time	85
Exa 4.3.1 Stock Price Changes I	85
Exa 4.3.2 Stock Price Changes II	85
Exa 4.3.3 Variance and Standard Deviation of a Discrete Distribution	86
Exa 4.3.4 Variance of a Discrete Distribution	87
Exa 4.3.5 Modification of a Bernoulli Distribution	87
Exa 4.3.6 Variance and Standard Deviation of a Linear Function	88
Exa 4.3.7 Investment Portfolio II	88
Exa 4.3.9 The Cauchy Distribution II	89
Exa 4.4.2 Skewness of Binomial Distributions	90
Exa 4.4.3 Moment Generating Function	91
Exa 4.5.1 Median of a Discrete Distribution I	92
Exa 4.5.2 Median of a Discrete Distribution II	92
Exa 4.5.3 Median of a Continuous Distribution	92
Exa 4.5.7 Last Lottery Number I	93
Exa 4.5.8 Last Lottery Number II	93

Exa 4.5.9	Discrete Uniform Random Variable	94
Exa 4.6.2	Test Scores I	94
Exa 4.6.3	Test Scores II	96
Exa 4.6.4	Uncorrelated Random Variables	96
Exa 4.6.5	Uniform Distribution Inside a Circle	97
Exa 4.7.2	Household Survey I	98
Exa 4.7.4	Household Survey II	99
Exa 4.7.6	Points from Uniform Distributions	99
Exa 4.7.8	A Clinical Trial	100
Exa 4.8.1	Choice of Gambles I	100
Exa 4.8.2	Choice of Gambles II	101
Exa 4.8.3	Linear Utility Function	102
Exa 4.8.4	Cubic Utility Function	102
Exa 4.8.5	Logarithmic Utility Function	103
Exa 4.8.6	Quadratic Utility Function	103
Exa 4.8.7	Square Root Utility Function	104
Exa 4.8.9	Bounding a Random Variable	104
Exa 4.8.10	Investment II	105
Exa 5.2.7	Group Testing	106
Exa 5.4.2	Customer Arrivals I	107
Exa 5.4.3	Customer Arrivals II	108
Exa 5.4.4	Approximating a Probability	108
Exa 5.4.6	Radioactive Particles	108
Exa 5.4.7	Cryptosporidium in Drinking Water	109
Exa 5.5.3	Triples in the Lottery I	109
Exa 5.5.4	Triples in the Lottery II	109
Exa 5.6.4	Probabilities for a Normal Distribution	110
Exa 5.6.5	Quantiles of Normal Distributions	110
Exa 5.6.6	Heights of Men and Women	111
Exa 5.6.7	Determining a Sample Size	111
Exa 5.6.8	Interval for Mean	112
Exa 5.6.9	Failure Times of Ball Bearings	112
Exa 5.6.10	Stock and Option Prices	113
Exa 5.7.2	Mean and Variance of Lifetime of a Light Bulb	114
Exa 5.8.4	Castaneda v Partida	114
Exa 5.8.5	A Clinical Trial	115
Exa 5.9.2	Attendance at a Baseball Game	116
Exa 5.9.3	Blood Types	117

Exa 5.10.6	Heights of Husbands and Wives	117
Exa 6.1.1	Proportion of Heads	119
Exa 6.2.1	Required Number of Observations	120
Exa 6.2.2	A Simulation	120
Exa 6.2.3	Tossing a Coin	121
Exa 6.2.4	Rate of Service I	121
Exa 6.2.5	Rate of Service II	122
Exa 6.2.6	Binomial Random Variable	123
Exa 6.3.1	A Large Sample I	123
Exa 6.3.2	Tossing a Coin	124
Exa 6.3.3	Sampling from a Uniform Distribution	125
Exa 6.3.5	Determining a Simulation Size	125
Exa 6.3.9	Examination Questions I	125
Exa 6.4.1	A Large Sample II	126
Exa 6.4.2	A Large Sample III	127
Exa 6.4.3	Examination Questions II	128
Exa 6.4.4	Coin Tossing	128
Exa 7.2.6	Lifetimes of Fluorescent Lamps I	130
Exa 7.2.8	Lifetimes of Fluorescent Lamps II	132
Exa 7.3.3	Glove Use by Nurses	133
Exa 7.3.8	Automobile Emissions	134
Exa 7.3.10	Calorie Counts on Food Labels	134
Exa 7.3.12	Lifetimes of Electronic Components I	135
Exa 7.4.5	Estimating the Parameter of a Bernoulli Distribution	136
Exa 7.4.8	Lifetimes of Electronic Components II	136
Exa 7.5.2	Lifetimes of Electronic Components III	137
Exa 7.5.3	Test for a Disease	137
Exa 7.6.2	Lifetimes of Electronic Components IV	138
Exa 7.6.6	Sampling from a Gamma Distribution	138
Exa 7.6.10	Sampling from a Uniform Distribution	139
Exa 7.6.12	Prussian Army Deaths	139
Exa 7.6.13	Lifetimes of Electronic Components V	140
Exa 7.6.15	Heights and Weights	140
Exa 7.6.16	Mixture of Normal Distributions	141
Exa 8.1.5	Lifetimes of Electronic Components I	143
Exa 8.1.6	A Clinical Trial	143
Exa 8.2.3	Acid Concentration in Cheese I	145
Exa 8.3.2	Rain from Seeded Clouds I	145

Exa 8.4.3	Rain from Seeded Clouds II	145
Exa 8.5.2	Rain from Seeded Clouds III	146
Exa 8.5.3	Rain from Seeded Clouds IV	146
Exa 8.5.4	Acid Concentration in Cheese II	147
Exa 8.5.6	Rain from Seeded Clouds V	148
Exa 8.5.7	Lifetimes of Electronic Components II	149
Exa 8.5.10	Approximate Confidence Interval for Poisson Mean . .	149
Exa 8.5.11	Uniforms on an Interval of Length One	149
Exa 8.6.2	Acid Concentration in Cheese III	150
Exa 8.6.3	Nursing Homes in New Mexico	151
Exa 8.6.4	An Improper Prior for Seeded Cloud Rainfall	152
Exa 8.7.3	Lifetimes of Electronic Components III	152
Exa 8.8.12	The Posterior Distribution of the Standard Deviation .	154
Exa 9.1.5	Testing Hypotheses about the Mean of a Normal Dis- tribution with Known Variance I	156
Exa 9.1.7	Testing Hypotheses about a Uniform Distribution . . .	157
Exa 9.1.9	Testing Hypotheses about a Bernoulli Parameter I . .	157
Exa 9.1.10	Testing Hypotheses about the Mean of a Normal Dis- tribution with Known Variance II	158
Exa 9.1.11	Testing Hypotheses about the Mean of a Normal Dis- tribution with Known Variance III	158
Exa 9.1.12	Testing Hypotheses about a Bernoulli Parameter II .	159
Exa 9.1.16	One Sided Confidence Interval for a Bernoulli Parameter	159
Exa 9.1.18	Likelihood Ratio Test of Two Sided Hypotheses about a Bernoulli Parameter I	160
Exa 9.1.19	Likelihood Ratio Test of Two Sided Hypotheses about a Bernoulli Parameter II	161
Exa 9.2.1	Service Times in a Queue I	161
Exa 9.2.2	Service Times in a Queue II	162
Exa 9.2.3	Service Times in a Queue III	162
Exa 9.2.4	Service Times in a Queue IV	163
Exa 9.2.5	Random Sample from a Normal Distribution	163
Exa 9.2.6	Sampling from a Bernoulli Distribution	164
Exa 9.3.6	Service Times in a Queue V	164
Exa 9.3.7	Testing Hypotheses about the Proportion of Defective Items	165
Exa 9.4.2	Egyptian Skulls I	165
Exa 9.4.4	Service Times in a Queue VI	166

Exa 9.4.5	Testing an Interval Null Hypothesis	167
Exa 9.5.3	Nursing Homes in New Mexico	167
Exa 9.5.4	Lengths of Fibers I	168
Exa 9.5.5	Lengths of Fibers II	169
Exa 9.5.6	Lengths of Fibers III	169
Exa 9.5.7	Crash Test Dummies	170
Exa 9.5.9	Egyptian Skulls II	171
Exa 9.5.10	Lengths of Fibers IV	172
Exa 9.5.11	Lengths of Fibers V	172
Exa 9.6.2	Rain from Seeded Clouds I	173
Exa 9.6.3	Roman Pottery in Britain I	173
Exa 9.6.4	Roman Pottery in Britain II	174
Exa 9.6.5	Comparing Copper Ores I	175
Exa 9.6.6	Comparing Copper Ores II	176
Exa 9.6.7	Comparing Copper Ores III	177
Exa 9.7.2	Quantile of an F Distribution	178
Exa 9.7.3	F Test	179
Exa 9.7.4	Rain from Seeded Clouds II	180
Exa 9.8.4	Calorie Counts on Food Labels	180
Exa 9.8.5	Pesticide Residue on Celery	181
Exa 9.8.6	Roman Pottery in Britain III	181
Exa 9.8.7	Roman Pottery in Britain IV	182
Exa 10.1.3	Blood Types I	184
Exa 10.1.4	Montana Outlook Poll I	185
Exa 10.1.5	Testing Hypotheses about a Proportion	185
Exa 10.1.6	Failure Times of Ball Bearings	186
Exa 10.1.7	Blood Types II	187
Exa 10.2.5	Failure Times of Ball Bearings	188
Exa 10.2.6	Prussian Army Deaths	189
Exa 10.3.2	College Survey	189
Exa 10.3.3	Montana Outlook Poll II	190
Exa 10.4.3	A Clinical Trial	191
Exa 10.6.2	Testing Whether a Sample Comes from a Standard Normal Distribution	192
Exa 10.6.4	Calcium Supplements and Blood Pressure I	192
Exa 10.7.3	Rain from Seeded Clouds	193
Exa 10.8.2	Calorie Counts in Hotdogs	194
Exa 10.8.4	Comparing Copper Ores	194

Exa 10.8.5 Calcium Supplements and Blood Pressure II	195
Exa 11.1.1 Blood Pressure I	197
Exa 11.1.2 Blood Pressure II	197
Exa 11.1.3 Fitting a Parabola	199
Exa 11.1.5 Fitting a Linear Function of Two Variables	200
Exa 11.2.1 Pressure and the Boiling Point of Water I	200
Exa 11.2.5 Predicting Pressure from the Boiling Point of Water	201
Exa 11.3.2 Gasoline Mileage I	202
Exa 11.3.3 Pressure and the Boiling Point of Water II	203
Exa 11.3.4 Gasoline Mileage II	204
Exa 11.3.5 Gasoline Mileage III	204
Exa 11.3.6 Pressure and the Boiling Point of Water III	205
Exa 11.3.7 Pressure and the Boiling Point of Water IV	206
Exa 11.3.9 Pressure and the Boiling Point of Water V	207
Exa 11.4.2 Pressure and the Boiling Point of Water VI	207
Exa 11.4.3 Pressure and the Boiling Point of Water VII	208
Exa 11.4.4 Gasoline Mileage IV	209
Exa 11.5.1 Unemployment in the 1950s I	210
Exa 11.5.3 Unemployment in the 1950s II	211
Exa 11.5.4 Dishwasher Shipments I	212
Exa 11.5.5 Dishwasher Shipments II	212
Exa 11.5.6 Unemployment in the 1950s III	213
Exa 11.5.7 Predicting Dishwasher Shipments	213
Exa 11.5.8 Unemployment in the 1950s IV	214
Exa 11.5.9 Unemployment in the 1950s V	215
Exa 11.5.10 Dishwasher Shipments III	217
Exa 11.6.4 Calories in Hot Dogs I	219
Exa 11.6.5 Calories in Hot Dogs II	219
Exa 11.6.6 Calories in Hot Dogs III	220
Exa 11.7.2 Radioactive Isotope in Milk I	221
Exa 11.7.3 Radioactive Isotope in Milk II	223
Exa 11.7.5 Testing for Differences among the Dairies	223
Exa 11.7.6 Testing for Differences among the Methods of Measure- ment	224
Exa 11.8.2 Gasoline Consumption I	224
Exa 11.8.3 Gasoline Consumption II	225
Exa 11.8.4 Gasoline Consumption III	226
Exa 11.8.5 Gasoline Consumption IV	227

Exa 12.1.1 The Mean of a Distribution	229
Exa 12.1.2 A Normal Probability	230
Exa 12.1.3 Waiting for a Break	231
Exa 12.1.4 Long Run of Heads	231
Exa 12.2.11 The MSE of the Sample Median	232
Exa 12.2.12 The Median of a Complicated Distribution	233
Exa 12.2.14 A Service Queue with Impatient Customers	234
Exa 12.3.7 Bayesian Analysis of OneWay Layout	235
Exa 12.3.8 Comparing Copper Ores	237
Exa 12.3.13 Simulating a Binomial Random Variable Using the Alias Method	238
Exa 12.4.1 Choosing an Importance Function	239
Exa 12.4.2 Calculating a Mean with No Closed Form Expression	241
Exa 12.4.4 Illustration of Stratified Importance Sampling	242
Exa 12.5.2 Nursing Homes in New Mexico I	244
Exa 12.5.3 Nursing Homes in New Mexico II	245
Exa 12.5.5 Unemployment in the 1950s	245
Exa 12.5.6 Bayesian Analysis of OneWay Layout with Unequal Variances	247
Exa 12.5.7 Calories in Hot Dogs	249
Exa 12.5.8 Censored Arsenic Measurements	250
Exa 12.6.3 The MSE of the Sample Median	251
Exa 12.6.4 Confidence Interval for the Interquartile Range	253
Exa 12.6.6 Percentile t Confidence Interval for a Median	254
Exa 12.6.7 Features of the Distribution of a Sample Correlation	255
Exa 12.6.8 Correcting the Bias in the Coefficient of Variation	255
Exa 12.6.9 Estimating the Standard Deviation of a Statistic	256
Exa 12.6.10 Comparing Means When Variances Are Unequal	257

Chapter 1

Introduction to probability

R code Exa 1.4.1 Rolling a Die

```
1 # Page 006
2
3 S <- c(1, 2, 3, 4, 5, 6)
4
5 Event_A <- S[S %% 2 == 0]
6 print(Event_A)
7
8 Event_B <- S[S > 2]
9 print(Event_B)
```

R code Exa 1.4.2 Subset

```
1 # Page 007
2
3 S <- c(1,2,3,4,5,6)
4
5 Event_A <- S[S %% 2 == 0]
6 print(Event_A)
```

```

7
8 Event_C <- S[S > 1]
9 print(Event_C)
10
11 if (all(Event_A %in% Event_C)){
12   print("A is a subset of C")
13 }

```

R code Exa 1.4.3 Complement

```

1 # Page 008
2
3 S <- c(1,2,3,4,5,6)
4
5 Event_A <- S[S %% 2 == 0]
6 print(Event_A)
7
8 Event_Ac <- setdiff(S, Event_A)
9 print(Event_Ac)

```

R code Exa 1.5.2 A loaded die

```

1 # Page 017
2 library(MASS)
3
4 S <- 1:6
5
6 probabilities <- c(1/7, 1/7, 1/7, 1/7, 1/7, 2/7)
7
8 Event_A <- S[S%%2 != 0]
9 print(Event_A)
10

```



```

11 Probability_of_A <- fractions(sum(probabilities[
    Event_A]))
12
13 print(Probability_of_A )

```

R code Exa 1.5.3 Diagnosing diseases

```

1 # Page 19
2
3 Probability_of_bacterial_infection <- 0.7
4
5 Probability_of_viral_infection <- 0.4
6
7 Probability_of_Bacterial_Union_Viral <- 1
8
9 Probability_of_patient_having_both_infection <-
    Probability_of_bacterial_infection + Probability_
    of_viral_infection - Probability_of_Bacterial_
    Union_Viral
10 print(Probability_of_patient_having_both_infection)

```

R code Exa 1.5.4 Demands for utilities

```

1 # Page 019
2
3 electric_demand <- 1:150
4 water_demand <- 4:200
5
6 sample_space <- expand.grid(water_demand = water_
    demand, electric_demand = electric_demand)
7
8 Area_of_sample_space <- (max(sample_space$water_
    demand) - min(sample_space$water_demand)) * (max(

```

```

    sample_space$electric_demand) - min(sample_space$
    electric_demand))
9
10 event_A <- sample_space[sample_space$water_demand >=
    100, ]
11
12 event_B <- sample_space[sample_space$electric_demand
    >= 115, ]
13
14 Area_of_A <- (max(event_A$water_demand) - min(event_
    A$water_demand)) * (max(event_A$electric_demand)
    - min(event_A$electric_demand))
15
16 Area_of_B <- (max(event_B$water_demand) - min(event_
    B$water_demand)) * (max(event_B$electric_demand)
    - min(event_B$electric_demand))
17
18 Probability_of_A <- Area_of_A / Area_of_sample_space
19
20 Probability_of_B <- Area_of_B / Area_of_sample_space
21
22 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
    ))
23 Probability_of_A_intersection_B <- Area_of_A_
    intersection_B/Area_of_sample_space
24
25 Probability_of_A_union_B <- Probability_of_A +
    Probability_of_B - Probability_of_A_intersection_
    B
26 print(round(Probability_of_A_union_B,4))

```

R code Exa 1.6.2 Fiber breaks

```

1 #page 22
2
3 library(MASS)
4
5 i <- 1:5
6
7 Alpha <- 1/15
8
9 Probability_of_fibre_length_not_more_than_3_breaking
  <- fractions(sum(Alpha*i[i<=3]))
10
11 print(Probability_of_fibre_length_not_more_than_3_
  breaking)

```

R code Exa 1.6.3 Tossing coins

```

1 #Page 23
2
3 library(MASS)
4
5 Probability_of_tossing_three_identical_coins <- 1/8
6
7 Probability_of_two_heads <- fractions(choose(3,2) *
  Probability_of_tossing_three_identical_coins)
8
9 print(Probability_of_two_heads)

```

R code Exa 1.6.4 Genetics

```

1 #Page 23
2
3 library(MASS)
4 genotypes <- c("AA", "Aa", "aA", "aa")

```

```

5
6 probabilities <- rep(1/4, 4)
7
8 genotype_probabilities <- c(
9   "AA" = sum(probabilities[genotypes %in% "AA"]),
10  "Aa" = sum(probabilities[genotypes %in% c("Aa", "
    aA")]),
11  "aa" = sum(probabilities[genotypes %in% "aa"])
12 )
13
14 print(paste("Probability of genotype AA:", fractions
    (genotype_probabilities["AA"])))
15 print(paste("Probability of genotype Aa:", fractions
    (genotype_probabilities["Aa"])))
16 print(paste("Probability of genotype aa:", fractions
    (genotype_probabilities["aa"])))

```

R code Exa 1.6.5 Rolling two dice

```

1 #Page 24
2 library(MASS)
3
4 sample_space <- expand.grid(1:6, 1:6)
5
6 sums <- rowSums(sample_space)
7
8 sum_probabilities <- table(sums) / length(sums)
9
10 sum_probabilities <- fractions(sum_probabilities)
11
12 for (i in 2:12) {
13   print(paste("P(Sum =", i, ")=", sum_probabilities[
    as.character(i)]))
14 }

```

R code Exa 1.7.1 Routes between cities

```
1 #Page 26
2
3 Number_of_cities <- 3
4
5 Number_of_routes_from_A_to_B <- 3
6
7 Number_of_routes_from_B_to_C <- 5
8
9 Total_number_of_routes <- Number_of_routes_from_A_to
  _B*Number_of_routes_from_B_to_C
10
11 print(Total_number_of_routes)
```

R code Exa 1.7.3 Possible outcomes of two dice

```
1 #Page 26
2
3 Number_of_possible_outcomes_for_a_die <- 6
4
5 Number_of_possible_outcomes_for_two_dice <- Number_
  of_possible_outcomes_for_a_die*Number_of_possible
  _outcomes_for_a_die
6
7 print(Number_of_possible_outcomes_for_two_dice)
```

R code Exa 1.7.4 Tossing several coins

```

1 #Page 27
2
3 library(MASS)
4
5 Number_of_possible_outcomes_for_six_coins <- 2^6
6
7 Number_of_outcomes_with_six_heads <- 1
8
9 Probability_of_all_heads <- fractions(Number_of_
    outcomes_with_six_heads/Number_of_possible_
    outcomes_for_six_coins)
10 print(Probability_of_all_heads)
11
12 Number_of_outcomes_with_one_head <- 6
13
14 Probability_of_one_head <- fractions(Number_of_
    outcomes_with_one_head/Number_of_possible_
    outcomes_for_six_coins)
15 print(Probability_of_one_head)

```

R code Exa 1.7.5 Combination lock

```

1 #Page 27
2
3 Number_of_elements <- 40
4
5 Length_of_sequence <- 3
6
7 Possible_combinations <- Number_of_elements^Length_
    of_sequence
8
9 print(Possible_combinations)

```

R code Exa 1.7.7 Current population survey

```
1 #Page 28
2
3 n <- 50000
4
5 k <- 3
6
7 Number_of_outcomes <- 50000 * 49999 * 49998
8
9 print(format(Number_of_outcomes, scientific = TRUE,
10             digits = 3))
11 #The answer may vary due to difference in
12     representation
```

R code Exa 1.7.8 Choosing officers

```
1 #Page 29
2
3 Number_of_members <- 25
4
5 Number_of_positions <- 2
6
7 Possible_number_of_choices <- choose(Number_of_
8     members, Number_of_positions)*factorial(Number_of
9     _positions)
10
11 print(Possible_number_of_choices)
```

R code Exa 1.7.9 Arranging books

```
1 #Page 29
2
```

```

3 Number_of_books <- 6
4
5 Possible_permutations <- factorial(Number_of_books)
6
7 print(Possible_permutations)

```

R code Exa 1.7.12 Approximating the number of permutations

```

1 #Page 31
2
3 stirling_approx <- function(n) {
4   sqrt(2 * pi * n) * (n / exp(1))^n
5 }
6
7 Permutation_70_20 <- factorial(70)/factorial(50)
8
9 print(format(Permutation_70_20, scientific = TRUE,
10             digits = 4))
11
12 Stirling_approximation <- stirling_approx(70)/
13   stirling_approx(50)
14
15 print(format(Stirling_approximation, scientific =
16             TRUE, digits = 4))
17
18 #The answer may vary due to difference in
19   representation

```

R code Exa 1.8.2 Selecting a committee

```

1 #Page 33
2
3 People_for_committee <- 8

```



```

4
5 People_in_group <- 20
6
7 Possible_number_of_groups <- choose(People_in_group,
   People_for_committe)
8
9 print(Possible_number_of_groups)

```

R code Exa 1.8.3 Choosing jobs

```

1 #Page 33
2
3 People_for_committe <- 8
4
5 People_in_group <- 20
6
7 Number_of_jobs <- 8
8
9 Possible_number_of_assignments <- choose(People_in_
   group, People_for_committe) * factorial(8)
10
11 print(Possible_number_of_assignments)
12 #The answer provided in the textbook is wrong.

```

R code Exa 1.8.4 Blood types

```

1 #Page 34
2
3 Number_of_alleles <- 3
4
5 Pairs_with_same_allele <- Number_of_alleles
6 Pairs_with_different_allele <- choose(Number_of_
   alleles, 2)

```

```

7
8 Total_number_of_genotypes <- Pairs_with_same_allele+
   Pairs_with_different_allele
9 print(Total_number_of_genotypes)

```

R code Exa 1.8.5 Selecting baked goods

```

1 #Page 35
2
3 Number_of_items_to_choose <- 12
4 Number_of_items_in_bakery <- 7
5
6 Different_boxfuls <- choose(Number_of_items_in_
   bakery+Number_of_items_to_choose-1, Number_of_
   items_to_choose)
7 print(Different_boxfuls)

```

R code Exa 1.8.6 Selecting baked goods from boxes

```

1 #Page 35
2
3 Number_of_items_to_choose <- 12
4 Number_of_items_in_bakery <- 7
5
6 Possible_outcomes_when_choosing_random <- Number_of_
   items_in_bakery^Number_of_items_to_choose
7 signif(Possible_outcomes_when_choosing_random,3)
8
9 Probability_that_all_items_are_same_choosing_random
   <- Number_of_items_in_bakery/Possible_outcomes_
   when_choosing_random
10 signif(Probability_that_all_items_are_same_choosing_
   random,3)

```

```

11
12 Number_of_packed_boxes_with_baker <- 18564
13 Probability_that_all_items_are_same_choosing_boxes
    <- Number_of_items_in_bakery/Number_of_packed_
    boxes_with_baker
14 signif(Probability_that_all_items_are_same_choosing_
    boxes,3)
15 #The answer may vary due to difference in
    representation

```

R code Exa 1.8.7 Tossing a coin

```

1 #Page 36
2
3 Number_of_tosses <- 10
4
5 Total_possible_number_of_sequences <- 2^Number_of_
    tosses
6
7 Arrangements_with_3_heads <- choose(10,3)
8
9 Probability_of_obtaining_exactly_3_heads <-
    Arrangements_with_3_heads/Total_possible_number_
    of_sequences
10 print(round(Probability_of_obtaining_exactly_3_heads
    ,4))
11
12 Probability_of_obtaining_atmost_3_heads <- sum(
    choose(10,0:3))/Total_possible_number_of_
    sequences
13 print(round(Probability_of_obtaining_atmost_3_heads
    ,4))

```

R code Exa 1.8.8 Sampling without replacement

```
1 #Page 37
2
3 Number_of_boys <- 15
4 Number_of_girls <- 30
5 Number_of_students_to_select <- 10
6
7 Possible_number_of_combinations_of_students <-
  choose(Number_of_boys+Number_of_girls,Number_of_
    students_to_select)
8
9 Possible_combinations_of_3_boys_and_7_girls <-
  choose(Number_of_boys,3)*choose(Number_of_girls
    ,7)
10
11 Desired_probability <- Possible_combinations_of_3_
    boys_and_7_girls/Possible_number_of_combinations_
    of_students
12 signif(Desired_probability,4)
```

R code Exa 1.8.9 Playing cards

```
1 #Page 37
2
3 Number_of_cards <- 52
4 Number_of_players <- 4
5 Number_of_aces <- 4
6
7 Total_possible_combinations <- choose(Number_of_
    cards,Number_of_players)
8 Possible_combinations_for_each_player_receiving_one_
    ace <- (Number_of_cards/Number_of_players)^Number
    _of_aces
9 Probability_that_each_player_will_receive_one_ace <-
```

```

    Possible_combinations_for_each_player_receiving_
    one_ace/Total_possible_combinations
10
11 print(round(Probability_that_each_player_will_
    receive_one_ace,4))

```

R code Exa 1.8.10 Playing cards revisited

```

1 #Page 38
2
3 Number_of_cards <- 52
4 Number_of_players <- 4
5 Number_of_aces <- 4
6
7 Total_possible_outcomes <- factorial(Number_of_cards
    )
8 Possible_combinations_for_each_player_receiving_one_
    ace <- (Number_of_cards/Number_of_players)^Number
    _of_aces
9 Number_of_ways_to_arrange_aces <- factorial(Number_
    of_aces)
10 Number_of_ways_to_arrange_rest_of_cards <- factorial
    (Number_of_cards-Number_of_aces)
11 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
12 Probability_that_each_player_will_receive_one_ace <-
    Possible_outcomes_of_interest/Total_possible_
    outcomes
13
14 signif(Probability_that_each_player_will_receive_one
    _ace,4)

```

R code Exa 1.8.11 Lottery ticket

```
1 #Page 38
2
3 Numbers_present <- 30
4 Numbers_chosen <- 6
5
6 Possible_number_of_outcomes <- choose(Numbers_
    present, Numbers_chosen)*factorial(Numbers_
    chosen)
7 print(Possible_number_of_outcomes)
8
9 Possible_combinations_of_six_numbers <- choose(
    Numbers_present, Numbers_chosen)
10 print(Possible_combinations_of_six_numbers)
```

R code Exa 1.9.1 Choosing committees

```
1 #Page 42
2
3 Number_of_members <- 20
4 Number_of_committees <- 3
5 Members_needed_in_committee_A <- 8
6 Members_needed_in_committee_B <- 8
7 Members_needed_in_committee_C <- 4
8
9 Combinations_possible_for_committee_A <- choose(
    Number_of_members, Members_needed_in_committee_A)
10 Combinations_for_rest_members_to_B_and_C <- choose(
    Number_of_members-Members_needed_in_committee_A,
    Members_needed_in_committee_B)
```

```

11 Possible_ways_of_choosing_committee <- Combinations_
    possible_for_committee_A*Combinations_for_rest_
    members_to_B_and_C
12
13 print(Possible_ways_of_choosing_committee)

```

R code Exa 1.9.2 Multinomial coefficient

```

1 #Page 43
2
3 library(iterpc)
4
5 n <- 20
6 k <- 3
7 n1 <- 8
8 n2 <- 8
9 n3 <- 4
10
11 Possible_combinations <- multichoose(c(n1,n2,n3))
12 print(Possible_combinations)

```

R code Exa 1.9.3 Rolling dice

```

1 #Page 44
2
3 Number_of_dice <- 12
4 k <- 6
5
6 Possible_outcomes <- factorial(Number_of_dice)/(
    factorial(Number_of_dice/k)^k)
7 Probability_each_number_appear_twice <- Possible_
    outcomes/k^Number_of_dice
8

```

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

R code Exa 1.9.4 Distributing heart cards

```
1 #Page 44
2
3 library(combinat)
4 library(gmp)
5 library(iterpc)
6
7 Number_of_cards <- 52
8 Number_of_hearts <- 13
9 Number_of_players <- 4
10 Number_of_cards_received_by_each_player <- 13
11 Hearts_received_by_player_A <- 6
12 Hearts_received_by_player_B <- 4
13 Hearts_received_by_player_C <- 2
14 Hearts_received_by_player_D <- 1
15
16 Ways_to_distribute_cards <- factorial(Number_of_
    cards) / (factorial(Number_of_cards_received_by_
    each_player)^Number_of_players)
17
18 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))
19 signif(Distribution_of_hearts_cards)
20
21 Number_of_remaining_cards <- Number_of_cards -
    Number_of_hearts
22 Distribution_of_other_cards <- choose(Number_of_
    remaining_cards, Number_of_cards_received_by_each
    _player - Hearts_received_by_player_A) *
23 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) *
24  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) *
25  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)
26
27  Distribution_of_cards_based_on_assigned_heart_cards
        <- Distribution_of_hearts_cards * Distribution_of
        _other_cards
28
29  Probability_of_assigning <- Distribution_of_cards_
        based_on_assigned_heart_cards / Ways_to_
        distribute_cards
30  signif(Probability_of_assigning,3)
31
32  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)
33  signif(Possible_combinations,3)

```

R code Exa 1.10.1 Student enrollment

```

1 #Page 47
2
3 library(MASS)
4
5 Number_of_students <- 200
6
7 Maths_students <- 137
8 History_students <- 50
9 Music_students <- 124
10
11 Students_in_both_maths_and_history <- 33
12 Students_in_both_music_and_history <- 29
13 Students_in_both_maths_and_music <- 92
14 Students_in_all_three <- 18
15
16 Probability_of_student_is_in_maths <- Maths_students
    /Number_of_students
17 fractions(Probability_of_student_is_in_maths)
18
19 Probability_of_student_is_in_history <- History_
    students/Number_of_students
20 fractions(Probability_of_student_is_in_history)
21
22 Probability_of_student_is_in_music <- Music_students
    /Number_of_students
23 fractions(Probability_of_student_is_in_music)
24
25
26 Probability_of_student_is_in_maths_and_history <-
    Students_in_both_maths_and_history/Number_of_
    students
27 fractions(Probability_of_student_is_in_maths_and_
    history)
28
29 Probability_of_student_is_in_music_and_history <-
    Students_in_both_music_and_history/Number_of_
    students
30 fractions(Probability_of_student_is_in_music_and_

```

```

    history)
31
32 Probability_of_student_is_in_maths_and_music <-
    Students_in_both_maths_and_music/Number_of_
    students
33 fractions(Probability_of_student_is_in_maths_and_
    music)
34
35 Probability_of_student_is_in_all_three <- Students_
    in_all_three/Number_of_students
36 fractions(Probability_of_student_is_in_all_three)
37
38 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)
39 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

```
1 #Page 56
2
3 Numbers_present <- 30
4 Numbers_taken <- 6
5 Possible_combinations <- choose(Numbers_present,
  Numbers_taken)
6
7 Probability_of_choosing_15_in_winning_combination <-
  choose(Numbers_present-1,Numbers_taken-1)/
  Possible_combinations
8 print(Probability_of_choosing_15_in_winning_
  combination)
9
10 Probability_of_winning <- 1/Possible_combinations
11 signif(Probability_of_winning,3)
12
13 Probability_of_winning_given_15_is_in_winning_number
  <- Probability_of_winning/Probability_of_
  choosing_15_in_winning_combination
14 signif(Probability_of_winning_given_15_is_in_winning
  _number,2)
```

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

R code Exa 2.1.3 Rolling dice

```
1 #Page 57
2
3 library(MASS)
4
5 sample_space <- expand.grid(1:6, 1:6)
6 sums <- rowSums(sample_space)
7
8 event_A <- sums < 8
9 event_B <- sums %% 2 == 1
10
11 event_A_and_B <- event_A & event_B
12 Pr_A_and_B <- sum(event_A_and_B) / nrow(sample_space)
13 print(paste("Pr(A and B):", fractions(Pr_A_and_B)))
14
15 Pr_B <- sum(event_B) / nrow(sample_space)
16 print(paste("Pr(B):", fractions(Pr_B)))
17
18 Pr_A_given_B <- Pr_A_and_B / Pr_B
19 print(paste("Pr(A | B):", fractions(Pr_A_given_B)))
```

R code Exa 2.1.4 A clinical trial

```
1 #Page 57
2
3 library(MASS)
4
5 total_patients <- 150
```

```

6
7 patients_placebo <- 34
8 patients_lithium <- 38
9
10 relapses_placebo <- 24
11 relapses_lithium <- 13
12
13 Pr_B <- patients_placebo / total_patients
14 print(paste("Pr(B):", fractions(Pr_B)))
15
16 Pr_C <- patients_lithium / total_patients
17 print(paste("Pr(C):", fractions(Pr_C)))
18
19 Pr_A_and_B <- relapses_placebo / total_patients
20 print(paste("Pr(A      B):", fractions(Pr_A_and_B)))
21 Pr_A_given_B <- Pr_A_and_B / Pr_B
22 print(paste("Pr(A | B):", signif(Pr_A_given_B,3)))
23
24 Pr_A_and_C <- relapses_lithium / total_patients
25 print(paste("Pr(A      C):", fractions(Pr_A_and_C)))
26 Pr_A_given_C <- Pr_A_and_C / Pr_C
27 print(paste("Pr(A | C):", signif(Pr_A_given_C,3)))

```

R code Exa 2.1.5 Rolling dice repeatedly

```

1 #Page 58
2
3 library(MASS)
4
5 Pr_T_7 <- 6 / 36
6 Pr_T_8 <- 5 / 36
7
8 Pr_A <- Pr_T_7
9 Pr_B <- Pr_T_7 + Pr_T_8
10

```

```
11 p <- Pr_T_7 / Pr_B
12
13 print(fractions(p))
```

R code Exa 2.1.9 Selecting bolts

```
1 #Page 61
2
3 library(MASS)
4 Pr_B1 <- 1 / 2
5 Pr_B2 <- 1 / 2
6
7 Pr_A_given_B1 <- 60 / 100
8 Pr_A_given_B2 <- 10 / 30
9
10 Pr_A <- Pr_B1 * Pr_A_given_B1 + Pr_B2 * Pr_A_given_
    B2
11
12 print(fractions(Pr_A))
```

R code Exa 2.1.10 Achieving a high score

```
1 #Page 61
2
3 Pr_Bi <- 1/50
4 i <- 1:50
5
6 Pr_A_given_Bi <- 1/(51-i)
7 Pr_A <- sum(Pr_Bi*Pr_A_given_Bi)
8 signif(Pr_A,3)
```

R code Exa 2.1.12 A clinical trial II

```
1 #Page 62
2
3 library(MASS)
4
5 Pr_Bj <- 1/11
6 j <- 1:11
7
8 Pr_E1_given_Bj <- (j-1)/10
9
10 Pr_E1 <- sum(Pr_Bj*Pr_E1_given_Bj)
11 print(fractions(Pr_E1))
```

R code Exa 2.2.1 Tossing coins

```
1 #Page 66
2
3 library(MASS)
4
5 outcomes <- c("HH", "HT", "TH", "TT")
6 probabilities <- rep(1/4, 4)
7
8 event_A <- c("HH", "TH")
9 Pr_A <- sum(probabilities[outcomes %in% event_A])
10 cat("The probability Pr(A) that the second toss is H
      is:", Pr_A, "\n")
11
12 event_B <- c("TH", "TT")
13 Pr_B <- sum(probabilities[outcomes %in% event_B])
14
15 event_A_and_B <- c("TH")
16 Pr_A_and_B <- sum(probabilities[outcomes %in% event_
      A_and_B])
17
```



```
18 Pr_A_given_B <- Pr_A_and_B / Pr_B
19
20 print(fractions(Pr_A_given_B))
```

R code Exa 2.2.2 Machine operation

```
1 #Page 67
2
3 library(MASS)
4
5 Pr_A <- 1/3
6 Pr_B <- 1/4
7
8 Pr_A_and_B <- Pr_A * Pr_B
9 print(fractions(Pr_A_and_B))
10
11 Pr_A_or_B <- Pr_A + Pr_B - Pr_A_and_B
12 print(fractions(Pr_A_or_B))
```

R code Exa 2.2.3 Rolling a Die

```
1 #Page 67
2
3 library(MASS)
4 Pr_A <- 3/6
5 Pr_B <- 4/6
6
7 Pr_A_and_B <- 2/6
8
9 independence_check <- Pr_A_and_B == (Pr_A * Pr_B)
10
11 if (independence_check){
12   print("Events A and B are independent")
13 }
```

13 }

R code Exa 2.2.4 Pairwise independence

```
1 #Page 69
2
3 sample_space <- c("HH", "HT", "TH", "TT")
4 probabilities <- rep(1/4, 4)
5
6 event_A <- c("HH", "HT")
7 event_B <- c("HH", "TH")
8 event_C <- c("HH", "TT")
9
10 Pr_A <- sum(probabilities[sample_space %in% event_A
11 ])
12 Pr_B <- sum(probabilities[sample_space %in% event_B
13 ])
14 Pr_C <- sum(probabilities[sample_space %in% event_C
15 ])
16
17 print(paste("Pr(A) =", fractions(Pr_A)))
18 print(paste("Pr(B) =", fractions(Pr_B)))
19 print(paste("Pr(C) =", fractions(Pr_C)))
20
21 Pr_A_and_B <- sum(probabilities[sample_space %in%
22 intersect(event_A, event_B)])
23 Pr_A_and_C <- sum(probabilities[sample_space %in%
24 intersect(event_A, event_C)])
25 Pr_B_and_C <- sum(probabilities[sample_space %in%
26 intersect(event_B, event_C)])
27 Pr_A_and_B_and_C <- sum(probabilities[sample_space %
28 in% intersect(intersect(event_A, event_B), event_
29 C)])
30
31 print(paste("Pr(A and B) =", fractions(Pr_A_and_B)))
```

```

24 print(paste("Pr(A      C) =", fractions(Pr_A_and_C)))
25 print(paste("Pr(B      C) =", fractions(Pr_B_and_C)))
26 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

```

1 #Page 70
2
3 prob_n_tosses_for_first_head <- function(n) {
4   return((1/2)^n)
5 }
6
7 max_n <- 100
8 prob_sum <- sum(sapply(1:max_n, prob_n_tosses_for_
   first_head))
9 print(paste("The sum of probabilities from n = 1 to"
   , max_n, "is:", prob_sum))

```

R code Exa 2.2.9 People v Collins

```

1 #Page 70
2
3 p <- 8.3e-8
4 n <- 8000000
5
6 Pr_A <- 1 - (1 - p)^n
7
8 Pr_C <- n * p * (1 - p)^(n - 1)
9
10 Pr_B <- Pr_A - Pr_C
11
12 Pr_B_given_A <- Pr_B / Pr_A

```

```

13
14 print(paste("The probability Pr(B|A) is:", signif(Pr
    _B_given_A,4)))
15 #The answer may slightly vary due to rounding off
    values

```

R code Exa 2.2.10 Inspecting items

```

1 # Page 72
2
3 p1 <- 0.01
4 p2 <- 0.4
5 n <- 6
6 k <- 2
7
8 Pr_A_given_B1 <- choose(n, k) * p1^k * (1 - p1)^(n -
    k)
9 print(paste("The probability Pr(A|B1) is:", signif(
    Pr_A_given_B1,3)))
10
11 Pr_A_given_B2 <- choose(n, k) * p2^k * (1 - p2)^(n -
    k)
12 print(paste("The probability Pr(A|B2) is:", signif(
    Pr_A_given_B2,3)))

```

R code Exa 2.3.2 Selecting bolts II

```

1 #Page 77
2
3 library(MASS)
4
5 Pr_A <- 7/15
6 Pr_A_given_B1 <- 3/5

```

```

7 Pr_A_given_B2 <- 1/3
8 Pr_B1 <- 1/2
9 Pr_B2 <- 1/2
10
11 Pr_B1_given_A <- Pr_B1*Pr_A_given_B1/Pr_A
12 print(paste("Probability (B1|A) =",fractions(Pr_B1_
    given_A)))

```

R code Exa 2.3.3 Test for a disease

```

1 #Page 78
2
3 Pr_B1 <- 0.0001
4 Pr_B2 <- 1 - Pr_B1
5
6 Pr_A_given_B1 <- 0.90
7 Pr_A_given_B2 <- 0.10
8
9 Pr_B1_given_A <- Pr_A_given_B1*Pr_B1/(Pr_A_given_B1*
    Pr_B1+Pr_A_given_B2*Pr_B2)
10 signif(Pr_B1_given_A,2)
11 #The answer may vary due to difference in
    representation.

```

R code Exa 2.3.4 Identifying the source of a defective item

```

1 #Page 78
2
3 Pr_B1 <- 0.2
4 Pr_B2 <- 0.3
5 Pr_B3 <- 0.5
6
7 Pr_A_given_B1 <- 0.01

```

```

8 Pr_A_given_B2 <- 0.02
9 Pr_A_given_B3 <- 0.03
10
11 Pr_A <- Pr_A_given_B1 * Pr_B1 + Pr_A_given_B2 * Pr_
    B2 + Pr_A_given_B3 * Pr_B3
12
13 Pr_B2_given_A <- (Pr_A_given_B2 * Pr_B2) / Pr_A
14
15 signif(Pr_B2_given_A,2)

```

R code Exa 2.3.5 Identifying genotypes

```

1 #Page 79
2
3 library(MASS)
4
5 Pr_B <- c(1/16, 1/4, 1/8, 1/4, 1/4, 1/4)
6 Pr_E_given_B <- c(1, 1, 1, 3/4, 1/2, 0)
7
8 Pr_E <- sum(Pr_B * Pr_E_given_B)
9 fractions(Pr_E)
10
11 Pr_B1_given_E <- Pr_B[1]*Pr_E_given_B[1]/Pr_E
12 fractions(Pr_B1_given_E)
13
14 Pr_B5_given_E <- Pr_B[5]*Pr_E_given_B[5]/Pr_E
15 fractions(Pr_B5_given_E)

```

R code Exa 2.3.6 Learning about a proportion

```

1 #Page 82
2
3 Pr_B1 <- 0.9

```

```

4 Pr_B2 <- 0.1
5
6 Pr_A_given_B1 <- dbinom(2, size = 6, prob = 0.01)
7 Pr_A_given_B2 <- dbinom(2, size = 6, prob = 0.4)
8
9 Pr_B1_given_A <- (Pr_A_given_B1 * Pr_B1) / ((Pr_A_
  given_B1 * Pr_B1) + (Pr_A_given_B2 * Pr_B2))
10
11 signif(Pr_B1_given_A,1)

```

R code Exa 2.3.7 A clinical trial III

```

1 #Page 82
2
3 calculate_posterior <- function(prior_probs,
  likelihoods) {
4   unnormalized_posterior <- prior_probs *
    likelihoods
5   posterior <- unnormalized_posterior / sum(
    unnormalized_posterior)
6   return(posterior)
7 }
8
9 j <- 1:11
10
11 prior_probs <- rep(1/11, 11)
12
13 likelihoods <- seq(0, 1, by = 0.1)
14
15 observed_success <- 1
16
17 posterior_probs <- calculate_posterior(prior_probs,
  likelihoods)
18
19 print(signif(posterior_probs,4))

```

```

20
21
22 likelihood <- function(j) {
23   return(choose(40, 22) * ((j - 1) / 10)^22 * (1 - (
24     j - 1) / 10)^18)
25 }
26 update_posterior <- function(prior_probs, observed_
27   successes, observed_failures) {
28   posterior_probs <- rep(0, length(prior_probs))
29   for (j in 1:length(prior_probs)) {
30     posterior_probs[j] <- prior_probs[j] *
31       likelihood(j)
32   }
33   posterior_probs <- posterior_probs / sum(posterior_
34     _probs)
35   return(posterior_probs)
36 }
37 posterior_probs <- update_posterior(prior_probs, 22,
38   18)
39
40 prior_prob_next_success <- 1/2
41 posterior_prob_next_success <- sum(posterior_probs *
42   (0:10)/10)
43
44 signif(posterior_prob_next_success, 4)

```

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

```

1 #Page 88
2
3 i <- 98
4 k <- 100
5

```



```
6 probability_A_wins <- i / k
7
8 print(probability_A_wins)
```

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

```
1 #Page 89
2
3 library(MASS)
4
5 p <- 0.4
6 i <- 99
7 k <- 100
8
9 ai <- (((((1 - p) / p) ^ i) - 1) / (((1 - p) / p) ^
      k) - 1)
10
11 fractions(ai)
```

Chapter 3

Random Variables and Distributions

R code Exa 3.1.5 Demands for utilities

```
1 #Page 95
2
3 length_C <- 175 - 50
4 Pr_X_in_C <- (149 * length_C) / 29204
5
6 signif(Pr_X_in_C,4)
```

R code Exa 3.1.8 Daily Numbers

```
1 #Page 97
2
3 i <- 0:9
4
5 s <- expand.grid(i,i,i)
6
7 X <- 100*s[1] +10*s[2] +s[3]
```

```
8
9 Pr_of_each_integer <- 1/lengths(X)
10 print(Pr_of_each_integer)
```

R code Exa 3.2.3 Incompletely Specified PDF

```
1 #Page 104
2
3 library(MASS)
4
5 integral_function <- function(x) {
6   return(x)
7 }
8
9 integral_value <- integrate(integral_function, lower
   = 0, upper = 4)$value
10
11 c_value <- 1 / integral_value
12
13 fractions(c_value)
```

R code Exa 3.2.4 Probabilities from a PDF

```
1 #Page 105
2
3 library(MASS)
4
5 integral_function <- function(x) {
6   return(x/8)
7 }
8
9 Pr_X_between_1_and_2 <- integrate(integral_function,
   lower = 1, upper = 2)$value
```

```

10 fractions(Pr_X_between_1_and_2)
11
12 Pr_X_greater_than_2 <- integrate(integral_function,
    lower = 2, upper = 4)$value
13 fractions(Pr_X_greater_than_2)

```

R code Exa 3.2.5 Unbounded Random Variables

```

1 #Page 105
2
3 f <- function(x) {
4   ifelse(x > 0, 1 / (1 + x)^2, 0)
5 }
6
7 prob_X_greater_1000 <- integrate(f, lower = 1000,
    upper = Inf, rel.tol = 1e-15)$value
8
9 cat("Pr(X > 1000):", signif(prob_X_greater_1000,1),
    "\n")

```

R code Exa 3.2.7 Truncated Voltage

```

1 #Page 106
2
3 library(MASS)
4
5 f <- function(x) {
6   ifelse(x > 0, 1 / (1 + x)^2, 0)
7 }
8
9 prob_X_greater_equal_3 <- integrate(f_X, lower = 3,
    upper = Inf)$value
10

```

```

11 prob_0_less_X_less_3 <- integrate(f_X, lower = 0,
    upper = 3)$value
12
13 print(paste("Pr(Y = 3):", fractions(prob_X_greater_
    equal_3)))
14 print(paste("Pr(0 < Y < 3):", fractions(prob_0_less_
    X_less_3)))

```

R code Exa 3.3.1 Voltage I

```

1 #Page 107
2
3 library(MASS)
4
5 get_pmf <- function(x){
6   if(x<=0){
7     return (0)
8   } else{
9     return (1-(1/(1+x)))
10  }
11 }
12
13 Pr_X_greater_than_3 <- get_pmf(3)
14 fractions(Pr_X_greater_than_3)

```

R code Exa 3.3.3 Voltage II

```

1 #Page 110
2
3 library(MASS)
4 get_pmf <- function(x){
5   if(x<=0){
6     return (0)

```

```

7   } else{
8       return (1-(1/(1+x)))
9   }
10 }
11
12 Pr_C <- get_pmf(4)
13 fractions(Pr_C)
14
15 Pr_B <- get_pmf(2)
16 fractions(Pr_B)
17
18 Pr_A <- Pr_C - Pr_B
19 fractions(Pr_A)
20 #The answer provided in the textbook is wrong.

```

R code Exa 3.3.9 Quantiles of a Binomial Distribution

```

1 #Page 115
2
3 n <- 5
4 p <- 0.3
5
6 pf_values <- dbinom(0:n, size = n, prob = p)
7 cdf_values <- pbinom(0:n, size = n, prob = p)
8
9 results <- data.frame(x = 0:n, pf = pf_values, cdf =
    cdf_values)
10
11 print(round(results,4))

```

R code Exa 3.3.10 Uniform Distribution on Integers

```

1 #Page 116

```

```

2
3 cdf_X <- function(x) {
4   if (x < 1) {
5     return(0)
6   } else if (x < 2) {
7     return(1 / 4)
8   } else if (x < 3) {
9     return(1 / 2)
10  } else if (x < 4) {
11    return(3 / 4)
12  } else {
13    return(1)
14  }
15 }
16
17 find_quantile <- function(p) {
18   uniroot(function(x) cdf_X(x) - p, lower = 1, upper
19           = 4)$root
20 }
21 median <- find_quantile(1/2)
22
23 print(median)

```

R code Exa 3.4.3 Discrete Joint Distribution

```

1 # Page 191
2
3 joint_prob <- matrix(c(
4   0.1, 0, 0.1, 0,
5   0.3, 0, 0.1, 0.2,
6   0, 0.2, 0, 0
7 ), nrow = 3, byrow = TRUE)
8
9 rownames(joint_prob) <- c("X=1", "X=2", "X=3")

```

```

10 colnames(joint_prob) <- c("Y=1", "Y=2", "Y=3", "Y=4"
    )
11
12 print(joint_prob)
13
14 pr_x_ge_2_y_ge_2 <- sum(joint_prob[2:3, 2:4])
15 print(pr_x_ge_2_y_ge_2)
16
17 pr_x_eq_1 <- sum(joint_prob[1, ])
18 print(pr_x_eq_1)

```

R code Exa 3.4.7 Normalizing Constant

```

1 # Page 122
2
3 library(MASS)
4 f_xy <- function(x, y) {
5   ifelse(x^2 <= y & y <= 1, x^2 * y, 0)
6 }
7
8 integrate_f_xy <- function() {
9   integrate(function(x) {
10     sapply(x, function(xi) {
11       integrate(function(y) f_xy(xi, y), lower = xi
12         ^2, upper = 1)$value
13     }, lower = -1, upper = 1)$value
14 }
15
16 result <- integrate_f_xy()
17 print(paste("The integral is:", fractions(result),"*
18   c"))
19 Integral_value <- 1
20 c <- Integral_value/result
21 print(paste("The value of c is:", fractions(c)))

```

R code Exa 3.4.8 Probability from a Joint PDF

```
1 # Page 122
2
3 library(MASS)
4 f_xy <- function(x, y) {
5   ifelse(x^2 <= y & y <= 1, (21/4) * x^2 * y, 0)
6 }
7
8 integrate_f_xy <- function() {
9   integrate(function(x) {
10     sapply(x, function(xi) {
11       integrate(function(y) f_xy(xi, y), lower = xi
12         ^2, upper = xi)$value
13     }, lower = 0, upper = 1)$value
14 }
15
16 result <- integrate_f_xy()
17 print(paste("Pr(X      Y ) =", fractions(result)))
```

R code Exa 3.4.9 Joint PDF by Geometric Methods

```
1 #Page 123
2
3 library(MASS)
4
5 f_xy <- function(x, y) {
6   ifelse((x^2 + y^2) <= 9, 1, 0)
7 }
8
```

```

9
10 integrate_f_xy <- function() {
11   integrate(function(x) {
12     sapply(x, function(xi) {
13       integrate(function(y) f_xy(xi, y), lower = -
14         sqrt(9 - xi^2), upper = sqrt(9 - xi^2))$
15         value
16     })
17   }, lower = -3, upper = 3)$value
18 }
19 result <- integrate_f_xy()
20 Integral_value <- 1
21 c <- Integral_value/result
22 print(paste("The value of c is:", fractions(c*pi),"*
  (1/ )"))

```

R code Exa 3.4.11 Joint PDF

```

1 #Page 124
2
3 f_xy <- function(x, y) {
4   ifelse((y > 0) & (y < 1), (x * y^(x - 1)) / 3, 0)
5 }
6
7 total_probability <- 0
8
9 for (x in 1:3) {
10   integral <- integrate(function(y) f_xy(x, y),
11     lower = 0, upper = 1)$value
12   total_probability <- total_probability + integral
13 }
14 cat("The total integral over the joint p.d.f. is:",
  total_probability,"\n")

```

```

15
16 probability <- 0
17
18
19 for (x in 2:3) {
20   integral <- integrate(function(y) f_xy(x, y),
21     lower = 1/2, upper = 1)$value
22   probability <- probability + integral
23 }
24 cat("The probability Pr(X      2 and Y      1/2) is:",
25     signif(probability,4), "\n")
26
27
28 integral_alternative <- integrate(function(y) {
29   ifelse((y >= 1/2), (2/ 3 * y ) + y^2, 0)
30 }, lower = 1/2, upper = 1)$value
31
32 cat("The probability computed in the alternative
33     order is:", signif(integral_alternative,4), "\n")

```

R code Exa 3.4.12 A Clinical Trial I

```

1 #Page 125
2
3 library(MASS)
4
5 f_xp <- function(x, p) {
6   ifelse((p > 0) & (p < 1), (p^x)*(1-p)^(1-x), 0)
7 }
8
9 pr_x_le_0_and_p_le_half <- integrate(function(p) f_
10   xp(0, p), lower = 0, upper = 1/2)$value

```

```

11 pr_x_eq_1 <- integrate(function(p) f_xp(1, p), lower
    = 0, upper = 1)$value
12
13 print(paste("Pr(X <= 0 and P <= 1/2):", fractions(pr
    _x_le_0_and_p_le_half)))
14 print(paste("Pr(X = 1):", fractions(pr_x_eq_1)))

```

R code Exa 3.5.2 Marginal PF

```

1 #Page 131
2
3 joint_pmf <- matrix(c(0.1, 0, 0.1, 0,
4                       0.3, 0, 0.1, 0.2,
5                       0, 0.2, 0, 0),
6                       ncol = 4, byrow = TRUE)
7
8 rownames(joint_pmf) <- c("1", "2", "3")
9 colnames(joint_pmf) <- c("1", "2", "3", "4")
10
11 joint_pmf_with_totals <- addmargins(joint_pmf)
12
13 print(joint_pmf_with_totals)

```

R code Exa 3.5.4 Marginal PDF from a Joint PDF

```

1 #Page 134
2
3 library(MASS)
4
5 f_xy <- function(x, y) {
6   (x * y^(x - 1)) / 3
7 }
8

```

```

9 integrate_f1 <- function(x) {
10   integrate(function(y) f_xy(x, y), lower = 0, upper
      = 1)$value
11 }
12
13 print(paste("Integral value", fractions(integrate_f1
      (1))))

```

R code Exa 3.5.7 Games of Chance

```

1 #Page 137
2
3 library(MASS)
4
5 joint_pmf <- matrix(c(1/24, 1/24, 1/24, 1/24, 1/24,
      1/24,
6                      1/12, 1/12, 1/12, 1/12, 1/12,
7                      1/12,
8                      1/24, 1/24, 1/24, 1/24, 1/24,
9                      1/24),
10                    nrow = 3, byrow = TRUE)
11
12 row_totals <- rowSums(joint_pmf)
13
14 joint_pmf_with_totals <- cbind(joint_pmf, row_totals)
15
16 joint_pmf_with_totals <- rbind(joint_pmf_with_totals,
17                                c(colSums(joint_pmf), sum(row_totals)))
18
19 rownames(joint_pmf_with_totals) <- c("0", "1", "2",
20                                     "Total")
21
22 colnames(joint_pmf_with_totals) <- c("1", "2", "3",
23                                     "4", "5", "6", "Total")
24
25 fractions(joint_pmf_with_totals)

```

R code Exa 3.5.8 Independence of Random Variable

```
1 #Page 137
2
3 f_1_2 <- 0.087
4 f1_1 <- 0.513
5 f2_1 <- 0.253
6
7 product_marginals <- f1_1 * f2_1
8
9 cat("f(1, 2):", f_1_2, "\n")
10 cat("f1(1) * f2(1):", product_marginals, "\n")
11
12 if (f_1_2 == product_marginals) {
13   cat("X and Y are independent.\n")
14 } else {
15   cat("X and Y are not independent.\n")
16 }
```

R code Exa 3.5.9 Probability involving Independent Random Variable

```
1 #Page 138
2
3 library(MASS)
4
5 joint_pdf <- function(x, y) {
6   ifelse(x+y <= 1, 4*x*y, 0)
7 }
8
9 prob_X_plus_Y_leq_1 <- integrate(
10   function(x) {
```

```

11     sapply(x, function(xi) {
12         integrate(function(y) joint_pdf(xi, y), lower
13             = 0, upper = 1 - xi)$value
14     },
15     lower = 0, upper = 1
16 )$value
17
18 print(paste("Pr(X + Y <= 1):", fractions(prob_X_plus
19     _Y_leq_1)))

```

R code Exa 3.5.11 Factorization of a Joint PDF

```

1 #Page 139
2
3 joint_pdf <- function(x, y) {
4     return(exp(-(x + 2 * y)))
5 }
6
7 integrate_joint_pdf <- function() {
8     integrate(function(x) {
9         sapply(x, function(xi) {
10             integrate(function(y) joint_pdf(xi, y), lower
11                 = 0, upper = Inf)$value
12         }, lower = 0, upper = Inf)$value
13 }
14
15 marginal_pdf_x <- function(x) {
16     if (x >= 0) {
17         return(exp(-x))
18     } else {
19         return(0)
20     }
21 }

```

```

22
23 marginal_pdf_y <- function(y) {
24   if (y >= 0) {
25     return(2 * exp(-2 * y))
26   } else {
27     return(0)
28   }
29 }
30
31 k <- (marginal_pdf_x(1)*marginal_pdf_y(1))/joint_pdf
    (1,1)
32 cat("k :",k)

```

R code Exa 3.6.1 Auto Insurance

```

1 #Page 142
2
3 joint_pmf <- matrix(c(0.129, 0.298, 0.161, 0.280,
4   0.108,
5   0.010, 0.010, 0.001, 0.002,
6   0.001),
7   nrow = 2, byrow = TRUE)
8
9 joint_pmf_with_totals <- cbind(joint_pmf, rowSums(
10   joint_pmf))
11 joint_pmf_with_totals <- rbind(joint_pmf_with_totals
12   , c(colSums(joint_pmf), sum(colSums(joint_pmf))))
13
14 rownames(joint_pmf_with_totals) <- c("0", "1", "
15   Total")
16 colnames(joint_pmf_with_totals) <- c("1", "2", "3",
17   "4", "5", "Total")
18
19 print(joint_pmf_with_totals)

```

R code Exa 3.6.2 Conditional PF from a Joint PF

```
1 #Page 143
2
3 library(MASS)
4 joint_prob <- matrix(c(
5   0.1, 0,   0.1, 0,
6   0.3, 0,   0.1, 0.2,
7   0,   0.2, 0,   0
8 ), nrow = 3, byrow = TRUE)
9
10 marginal_prob_X2 <- 0.6
11
12 conditional_prob_Y_given_X2 <- joint_prob[2, ] /
13   marginal_prob_X2
14 names(conditional_prob_Y_given_X2) <- paste("g2("
15   ,1:4," | 2)" )
16 fractions(conditional_prob_Y_given_X2)
```

R code Exa 3.6.5 Processing Times

```
1 #Page 145
2
3 f_X <- function(x) {
4   ifelse(x >= 0, exp(4-x), 0)
5 }
6
7 conditional_prob_X_geq_9_given_Y_4 <- integrate(f_X,
8   lower = 9, upper = Inf)$value
9
10 signif(conditional_prob_X_geq_9_given_Y_4,2)
```

R code Exa 3.6.6 Conditional PDF from a Joint PDF

```
1 #Page 145
2
3 library(MASS)
4 g2 <- function(y, x) {
5   ifelse((x^2 <= y) & (y <= 1), 2 * y / (1 - x^4),
6         0)
7 }
8 integrate_g2_1 <- function(x) {
9   integrate(function(y) g2(y, x), lower = 1/4, upper
10             = 1)$value
11 }
12 prob1 <- integrate_g2_1(1/2)
13
14 print(prob1)
15
16 integrate_g2_2 <- function(x) {
17   integrate(function(y) g2(y, x), lower = 3/4, upper
18             = 1)$value
19 }
20 prob2 <- integrate_g2_2(1/2)
21
22 fractions(prob2)
```

R code Exa 3.6.9 Defective Parts

```
1 #Page 148
```

```

2
3 library(MASS)
4
5 integrand <- function(q) {
6   q * (1 - q)
7 }
8
9 result <- integrate(integrand, lower = 0, upper = 1)
10   $value
11 fractions(result)

```

R code Exa 3.6.10 Uniform Distributions

```

1 #Page 149
2
3 f2 <- function(y) {
4   ifelse(y > 0 & y < 1, -log(1 - y), 0)
5 }
6
7 y_vals <- seq(0.01, 0.99, length.out = 100)
8
9 f2_vals <- f2(y_vals)
10
11 plot(y_vals, f2_vals, type = "l", col = "blue", lwd
12       = 2,
13       xlab = "y", ylab = "f2(y)", main = "Marginal p.
14         d.f. f2(y)",
15       ylim = c(0, 5))
16
17 abline(h = 0, col = "gray", lty = 2)

```

R code Exa 3.7.7 Arrivals at a Queue

```

1 #Page 155
2
3 sum_h13 <- function(y){
4   ifelse(y>0, 2 * exp(-2*y), 0)
5 }
6 final_result <- integrate(sum_h13 , lower = 0, upper
   = Inf)$value
7
8 print(final_result)

```

R code Exa 3.7.8 A Clinical Trial II

```

1 #Page 156
2
3 library(MASS)
4
5 integrand <- function(p) {
6   p * (1 - p)
7 }
8
9 integral_value <- integrate(integrand, lower = 0,
   upper = 1)$value
10
11 prob_x1_x2_1 <- 2 * integral_value
12
13 print(fractions(prob_x1_x2_1))

```

R code Exa 3.7.9 Service Times in a Queue I

```

1 #Page 157
2
3 fi <- function(xi) {
4   ifelse(xi > 0, 2 / (2 + xi)^2, 0)

```

```

5 }
6
7 integrand <- function(xi) {
8   sapply(xi, fi)
9 }
10
11 result <- integrate(integrand, lower = 3, upper =
    Inf)$value
12
13 cat("Pr(Xi > 3) =", result, "\n")

```

R code Exa 3.7.14 Service Times in a Queue II

```

1 #Page 160
2
3 library(cubature)
4
5 g1 <- function(x3, x4, x5, x1, x2) {
6   return(60 * (2 + x1 + x2)^3 / (2 + x1 + x2 + x3 +
    x4 + x5)^6)
7 }
8
9 x1 <- 4
10 x2 <- 6
11
12 result <- adaptIntegrate(function(x) {
13   x3 <- x[1]
14   x4 <- x[2]
15   x5 <- x[3]
16   return(g1(x3, x4, x5, x1, x2))
17 }, lowerLimit = c(3, 0, 0), upperLimit = c(Inf, Inf,
    Inf))
18
19 conditional_probability <- result$integral
20

```

```

21 print(paste("Pr(X3 > 3|X1 = 4, X2 = 6) =", round(
    conditional_probability, 3)))
22
23 x1 <- 1
24 x2 <- 1.5
25
26 result <- adaptIntegrate(function(x) {
27   x3 <- x[1]
28   x4 <- x[2]
29   x5 <- x[3]
30   return(g1(x3, x4, x5, x1, x2))
31 }, lowerLimit = c(3, 0, 0), upperLimit = c(Inf, Inf,
    Inf))
32
33 conditional_probability <- result$integral
34
35 print(paste("Pr(X3 > 3|X1 = 1, X2 = 1.5) = ", round(
    conditional_probability, 3)))

```

R code Exa 3.7.15 Marginal Bivariate PDF

```

1 #Page 161
2
3 library(MASS)
4
5 integrand <- function(x1, x2) {
6   return(4 / (2 + x1 + x2)^3)
7 }
8
9 result <- integrate(function(x2) {
10   sapply(x2, function(x2) {
11     integrate(function(x1) {
12       integrand(x1, x2)
13     }, lower = 0, upper = 4 - x2)$value
14   })

```

```
15 }, lower = 0, upper = 4)$value
16
17 fractions(result)
```

R code Exa 3.7.16 Service Times in a Queue III

```
1 #Page 162
2
3 g0 <- function(z, x1, x2) {
4   ifelse (z > 0, (1/2) * (2 + x1 + x2)^3 * z^2 * exp
5     (-z * (2 + x1 + x2)), 0)
6 }
7
8 x1 <- 1
9 x2 <- 4
10
11 integrand <- function(z) {
12   g0(z, x1, x2)
13 }
14
15 probability <- integrate(integrand, lower = 0, upper
16   = 1)$value
17
18 signif(probability,4)
```

R code Exa 3.7.19 Rate of Service I

```
1 #Page 164
2 set.seed(123)
3
4 n <- 100
5 rate <- 2
6
```

```

7 z <- rexp(1, rate = rate)
8
9 service_times <- rexp(n, rate = z)
10
11 first_10 <- service_times[1:10]
12 cat("First 10 observations:", first_10, "\n")
13
14 smallest_time <- min(service_times)
15 largest_time <- max(service_times)
16 cat("Smallest service time:", smallest_time, "\n")
17 cat("Largest service time:", largest_time, "\n")

```

R code Exa 3.7.20 Rate of Service II

```

1 #Page 165
2 set.seed(123)
3
4 n <- 100
5 rate <- 2
6
7 z <- rexp(1, rate = rate)
8
9 service_times <- rexp(n, rate = z)
10
11 service_times <- ifelse(service_times > 10, runif(
    sum(service_times > 10), 0, 10), service_times)
12
13 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


```

1 #Page 168
2
3 library(MASS)
4
5 prob_X_4_or_6 <- function() {
6   lower_bound <- 0
7   upper_bound <- 9
8
9   prob_X_4 <- dunif(4, min = lower_bound, max =
      upper_bound)
10
11   prob_X_6 <- dunif(6, min = lower_bound, max =
      upper_bound)
12
13   prob_X_4 + prob_X_6
14 }
15
16 prob_Y_1 <- prob_X_4_or_6()
17
18 fractions(prob_Y_1)

```

R code Exa 3.8.2 Distance from the Middle II

```

1 #Page 168
2
3 library(MASS)
4
5 f_X <- function(x) {
6   if (x >= 0 && x <= 9) {
7     return(1 / 9)
8   } else {
9     return(0)
10  }
11 }
12

```

```

13 g_Y <- function(y) {
14   if (y == 0) {
15     return(f_X(5))
16   } else if (y >= 1 && y <= 4) {
17     return(2 * f_X(5 - y))
18   } else {
19     return(0)
20   }
21 }
22
23 prob_Y_0 <- g_Y(0)
24
25 prob_Y_4 <- g_Y(4)
26
27 print(paste("Pr(Y = 0) =", fractions(prob_Y_0)))
28
29 print(paste("Pr(Y = 4) =", fractions(prob_Y_4)))

```

R code Exa 3.8.4 PDF of square of Uniform Distribution

```

1 #Page 169
2
3 set.seed(123)
4
5 g_y <- function(y) {
6   ifelse(y >= 0 & y <= 1, 1 / (2 * sqrt(y)), 0)
7 }
8
9 y_values <- sort(runif(50, min = 0, max = 1))
10 pdf_y <- g_y(y_values)
11
12 plot(y_values, pdf_y, type = "l", col = "blue", lwd
      = 2,
13       main = "PDF of Y = X^2 where X ~ Uniform(-1, 1)
      ",

```

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

R code Exa 3.8.6 Generating Independent Values from a Specified PDF

```
1 #Page 171
2
3 G_inv <- function(x) {
4   return(2 * (1 - (1 - x)^(1/2)))
5 }
6
7 x_values <- c(0.4125, 0.0894, 0.8302)
8
9 y_values <- G_inv(x_values)
10
11 result_df <- data.frame(x = x_values, y = round(y_
12   values,2))
13 print(result_df)
```

R code Exa 3.9.2 Bull Market

```
1 #Page 176
2
3 n_points_A <- choose(10, 3) * choose(10, 5) * choose
4   (10, 8)
5 print(n_points_A)
6
7 total_possible_values <- 2^30
8
9 g_358 <- n_points_A / total_possible_values
10
11 signif(g_358,3)
```

R code Exa 3.9.5 An Investment Portfolio

```
1 #Page 179
2
3 g_y <- function(y) {
4   ifelse(y > 1800 & y <= 2200, 8.333e-7 * (y - 1800)
5     ,
6     ifelse(y > 2200 & y <= 4800, 3.333e-4,
7       ifelse(y > 4800 & y < 5200, 8.333e-7
8         * (5200 - y), 0)))
9 }
10
11 y_vals <- seq(from = 1800, to = 5200, by = 1)
12
13 g_y_values <- sapply(y_vals, g_y)
14
15 cat("For 1800 < y <= 2200, g(y) =", g_y_values[y_
16   vals > 1800 & y_vals <= 2200][1], "\n")
17 cat("For 2200 < y <= 4800, g(y) =", g_y_values[y_
18   vals > 2200 & y_vals <= 4800][1], "\n")
19 cat("For 4800 < y < 5200, g(y) =", g_y_values[y_vals
20   > 4800 & y_vals < 5200][1], "\n")
```

R code Exa 3.9.6 Maximum and Minimum of a Random Sample

```
1 #Page 180
2
3 library(ggplot2)
4
5 set.seed(121)
6
```

```

7  n <- 5
8  num_samples <- 10000
9
10 samples <- matrix(runif(n * num_samples, min = 0,
    max = 1), nrow = num_samples, ncol = n)
11
12 single_vals <- as.vector(samples)
13 min_vals <- apply(samples, 1, min)
14 max_vals <- apply(samples, 1, max)
15 range_vals <- max_vals - min_vals
16
17 density_single <- density(single_vals)
18 density_min <- density(min_vals)
19 density_max <- density(max_vals)
20 density_range <- density(range_vals)
21
22 plot(density_single, col = "blue", xlim = c(0, 1),
    ylim = c(0, 5), xlab = "x", ylab = "Density",
23     main = "PDFs of Single Random Variable, Min,
        Max, and Range of 5 Samples")
24 lines(density_min, col = "red", lty = 2)
25 lines(density_max, col = "green", lty = 2)
26 lines(density_range, col = "purple", lty = 2)
27 legend("topright", legend = c("Single Random
    Variable", "Min of 5", "Max of 5", "Range of 5"),
28     col = c("blue", "red", "green", "purple"),
        lty = c(1, 2, 2, 2))

```

R code Exa 3.10.2 Shopping for Toothpaste I

```

1  #Page 189
2
3  library(MASS)
4
5  transition_matrix <- matrix(c(1/3, 2/3, 2/3, 1/3),

```

```

      nrow = 2, byrow = TRUE)
6  colnames(transition_matrix) <- c("A", "B")
7  rownames(transition_matrix) <- c("A", "B")
8
9  print(paste("Pr( $X_{n+1} = 1 \mid X_n = 1$ ) =", fractions(
      transition_matrix[1, 1])))
10 print(paste("Pr( $X_{n+1} = 2 \mid X_n = 1$ ) =", fractions(
      transition_matrix[1, 2])))
11 print(paste("Pr( $X_{n+1} = 1 \mid X_n = 2$ ) =", fractions(
      transition_matrix[2, 1])))
12 print(paste("Pr( $X_{n+1} = 2 \mid X_n = 2$ ) =", fractions(
      transition_matrix[2, 2])))

```

R code Exa 3.10.4 Shopping for Toothpaste II

```

1  #Page 190
2
3  library(MASS)
4
5  transition_matrix <- matrix(c(1/3, 2/3, 2/3, 1/3),
      nrow = 2, byrow = TRUE)
6  colnames(transition_matrix) <- c("A", "B")
7  rownames(transition_matrix) <- c("A", "B")
8
9  fractions(transition_matrix)

```

R code Exa 3.10.5 Number of Occupied Telephone Lines I

```

1  #Page 191
2
3  P <- matrix(c(0.1, 0.4, 0.2, 0.1, 0.1, 0.1,
4                0.2, 0.3, 0.2, 0.1, 0.1, 0.1,
5                0.1, 0.2, 0.3, 0.2, 0.1, 0.1,

```

```

6           0.1, 0.1, 0.2, 0.3, 0.2, 0.1,
7           0.1, 0.1, 0.1, 0.2, 0.3, 0.2,
8           0.1, 0.1, 0.1, 0.1, 0.4, 0.2), nrow =
           6, byrow = TRUE)
9
10  print(P)
11
12  prob_5_4 <- P[6, 5]
13  cat("The probability P(5, 4) is:", prob_5_4, "\n")
14
15  prob_0_0 <- P[1, 1]
16  cat("The probability P(0, 0) is:", prob_0_0, "\n")
17
18  prob_at_least_one <- 1 - prob_0_0
19  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

```

1  #Page 192
2
3  states <- c("{AA, AA}", "{AA, Aa}", "{AA, aa}", "{Aa,
      Aa}", "{Aa, aa}", "{aa, aa}")
4
5  transition_matrix <- matrix(0, nrow = 6, ncol = 6)
6  rownames(transition_matrix) <- states
7  colnames(transition_matrix) <- states
8
9  transition_matrix["{AA, AA}", "{AA, AA}"] <- 1
10
11 transition_matrix["{AA, Aa}", "{AA, AA}"] <- 0.25
12 transition_matrix["{AA, Aa}", "{AA, Aa}"] <- 0.50
13 transition_matrix["{AA, Aa}", "{Aa, Aa}"] <- 0.25
14
15 transition_matrix["{AA, aa}", "{Aa, Aa}"] <- 1

```

```

16
17 transition_matrix["{Aa, Aa}", "{AA, AA}"] <- 1/16
18 transition_matrix["{Aa, Aa}", "{AA, Aa}"] <- 1/4
19 transition_matrix["{Aa, Aa}", "{AA, aa}"] <- 1/8
20 transition_matrix["{Aa, Aa}", "{Aa, Aa}"] <- 1/4
21 transition_matrix["{Aa, Aa}", "{Aa, aa}"] <- 1/4
22 transition_matrix["{Aa, Aa}", "{aa, aa}"] <- 1/16
23
24 transition_matrix["{Aa, aa}", "{Aa, Aa}"] <- 1/4
25 transition_matrix["{Aa, aa}", "{Aa, aa}"] <- 1/2
26 transition_matrix["{Aa, aa}", "{aa, aa}"] <- 1/4
27
28 transition_matrix["{aa, aa}", "{aa, aa}"] <- 1
29
30 print(transition_matrix)

```

R code Exa 3.10.7 Single Server Queue I

```

1 #Page 193
2
3 P <- matrix(c(0.9, 0.1, 0.6, 0.4), nrow = 2, byrow =
  TRUE)
4
5 Pr_Xn1_Xn2_given_Xn1 <- P[1, 1] * P[1, 1]
6
7 Pr_Xn1_Xn2_given_Xn1 <- c(Pr_Xn1_Xn2_given_Xn1, P[1,
  2] * P[2, 1])
8
9 Pr_Xn2_given_Xn1_1 <- sum(Pr_Xn1_Xn2_given_Xn1)
10
11 Pr_Xn2_given_Xn1_2 <- 1 - Pr_Xn2_given_Xn1_1
12
13 Pr_Xn2_given_Xn2_1 <- sum(P[2,] * P[, 1])
14
15 Pr_Xn2_given_Xn2_2 <- 1 - Pr_Xn2_given_Xn2_1

```



```

16
17 cat("Pr( $X_{n+1} = 1, X_{n+2} = 1 | X_n = 1$ ):", Pr_Xn1_Xn2_
    given_Xn1[1], "\n")
18 cat("Pr( $X_{n+1} = 2, X_{n+2} = 1 | X_n = 1$ ):", Pr_Xn1_Xn2_
    given_Xn1[2], "\n")
19 cat("Pr( $X_{n+2} = 1 | X_n = 1$ ):", Pr_Xn2_given_Xn1_1, "\n"
    )
20 cat("Pr( $X_{n+2} = 2 | X_n = 1$ ):", Pr_Xn2_given_Xn1_2, "\n"
    )
21 cat("Pr( $X_{n+2} = 1 | X_n = 2$ ):", Pr_Xn2_given_Xn2_1, "\n"
    )
22 cat("Pr( $X_{n+2} = 2 | X_n = 2$ ):", Pr_Xn2_given_Xn2_2, "\n"
    )

```

R code Exa 3.10.8 Number of Occupied Telephone Lines II

```

1 #Page 195
2
3 P <- matrix(c(0.1, 0.4, 0.2, 0.1, 0.1, 0.1,
4               0.2, 0.3, 0.2, 0.1, 0.1, 0.1,
5               0.1, 0.2, 0.3, 0.2, 0.1, 0.1,
6               0.1, 0.1, 0.2, 0.3, 0.2, 0.1,
7               0.1, 0.1, 0.1, 0.2, 0.3, 0.2,
8               0.1, 0.1, 0.1, 0.1, 0.4, 0.2), nrow =
9               6, byrow = TRUE)
10
11
12 print(P)
13
14 P2 <- P %*% P
15
16 P3 <- P2 %*% P
17
18 cat("P2:\n")
19 print(P2)
20

```

```

19 cat("\nP3:\n")
20 print(P3)

```

R code Exa 3.10.9 Plant Breeding Experiment II

```

1 #Page 195
2
3 inheritance_matrix <- matrix(c(1.0000, 0.0000,
4                               0.0000, 0.0000, 0.0000, 0.0000,
5                               0.2500, 0.5000,
6                               0.0000, 0.2500,
7                               0.0000, 0.0000,
8                               0.0000, 0.0000,
9                               0.0000, 1.0000,
10                              0.0000, 0.0000,
11                              0.0625, 0.2500,
12                              0.1250, 0.2500,
13                              0.2500, 0.0625,
14                              0.0000, 0.0000,
15                              0.0000, 0.2500,
16                              0.5000, 0.2500,
17                              0.0000, 0.0000,
18                              0.0000, 0.0000,
19                              0.0000, 1.0000),
20                               nrow = 6, byrow = TRUE)
21
22 row_names <- c("{AA, AA}", "{AA, Aa}", "{AA, aa}", "{Aa, Aa}",
23               "{Aa, aa}", "{aa, aa}")
24 col_names <- c("{AA, AA}", "{AA, Aa}", "{AA, aa}", "{Aa, Aa}",
25               "{Aa, aa}", "{aa, aa}")
26 rownames(inheritance_matrix) <- row_names
27 colnames(inheritance_matrix) <- col_names
28
29 print(inheritance_matrix)
30

```

```

18 two_step_transition_matrix <- inheritance_matrix %*%
    inheritance_matrix
19
20 print(round(two_step_transition_matrix,4))

```

R code Exa 3.10.10 Single Server Queue II

```

1 #Page 196
2
3 P_server_is_busy <- 0.3
4
5 v <- c(P_server_is_busy, 1 - P_server_is_busy)
6
7 print(v)

```

R code Exa 3.10.11 Number of Occupied Telephone Lines III

```

1 #Page 197
2
3 P <- matrix(c(0.1, 0.4, 0.2, 0.1, 0.1, 0.1,
4              0.2, 0.3, 0.2, 0.1, 0.1, 0.1,
5              0.1, 0.2, 0.3, 0.2, 0.1, 0.1,
6              0.1, 0.1, 0.2, 0.3, 0.2, 0.1,
7              0.1, 0.1, 0.1, 0.2, 0.3, 0.2,
8              0.1, 0.1, 0.1, 0.1, 0.4, 0.2), nrow =
              6, byrow = TRUE)
9
10 v <- c(0.5, 0.3, 0.2, 0, 0, 0)
11
12 v_P <- v %*% P
13 cat("Distribution of the number of lines in use at
    time 2:", v_P, "\n")
14

```

```

15 v_P2 <- v %*% P %*% P
16 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

```

1 #Page 197
2
3 P <- matrix(c(0.1, 0.4, 0.2, 0.1, 0.1, 0.1,
4               0.2, 0.3, 0.2, 0.1, 0.1, 0.1,
5               0.1, 0.2, 0.3, 0.2, 0.1, 0.1,
6               0.1, 0.1, 0.2, 0.3, 0.2, 0.1,
7               0.1, 0.1, 0.1, 0.2, 0.3, 0.2,
8               0.1, 0.1, 0.1, 0.1, 0.4, 0.2), nrow =
9               6, byrow = TRUE)
10
11 v <- c(0.119, 0.193, 0.186, 0.173, 0.196, 0.133)
12
13 vP <- round(v %*% P, 3)
14
15 if(isTRUE(all.equal(as.vector(vP), v))) {
16   cat("The initial distribution remains the same
17       after one transition and therefore after two or
18       more transitions as well.\n")
19 } else {
20   cat("The initial distribution does not remain the
21       same after one transition.\n")
22 }

```

R code Exa 3.10.14 Toothpaste Shopping

```

1 #Page 199
2

```

```

3 library(MASS)
4
5 P <- matrix(c(1/3, 2/3, 2/3, 1/3), nrow = 2, byrow =
      TRUE)
6
7 I <- diag(nrow(P))
8 G <- P - I
9 G[,ncol(G)] <- 1
10
11 fractions(G)
12
13 G_inv <- solve(G)
14
15 fractions(G_inv)
16
17 v <- G_inv[nrow(G_inv), ]
18
19 print(fractions(v))

```

R code Exa 3.10.15 Alternating Chain

```

1 #Page 199
2
3 P <- matrix(c(0, 1, 1, 0), nrow = 2, byrow = TRUE)
4
5 I <- diag(nrow(P))
6 G <- P - I
7 G[,ncol(G)] <- 1
8
9 fractions(G)
10
11 G_inv <- solve(G)
12
13 fractions(G_inv)
14

```

```
15 v <- G_inv[nrow(G_inv), ]  
16  
17 print(v)
```

Chapter 4

Expectation

R code Exa 4.1.2 Stock Price Change

```
1 #Page 207
2
3 values <- c(-2, 0, 1, 4)
4 probabilities <- c(0.1, 0.4, 0.3, 0.2)
5
6 weighted_average <- sum(values * probabilities)
7 print(paste("Weighted Average:", weighted_average))
8
9 initial_amount <- 18
10 interest_rate <- 0.05
11 interest_earned <- initial_amount * interest_rate
12 print(paste("Interest Earned:", interest_earned))
```

R code Exa 4.1.6 Expected Failure Time

```
1 #Page 209
2
3 library(MASS)
```

```

4
5 integrand <- function(x) {
6   return(2 * x^2)
7 }
8
9 result <- integrate(integrand, lower = 0, upper = 1)
10
11 expected_value <- result$value
12 print(paste("The expected value E(X) is:", fractions
              (expected_value)))

```

R code Exa 4.1.7 Failure after Warranty

```

1 #Page 210
2
3 f <- function(x) {
4   return(2 / x^3)
5 }
6
7 integrand <- function(x){
8   return(x*f(x))
9 }
10
11 result <- integrate(integrand, lower = 1, upper =
12   Inf)
13 expected_value <- result$value
14 print(paste("The expected value E(X) is:", fractions
              (expected_value)))

```

R code Exa 4.1.9 The Cauchy Distribution I

```

1 #Page 211

```



```

2
3 library(ggplot2)
4 library(gridExtra)
5
6 f <- function(x) {
7   1 / (pi * (1 + x^2))
8 }
9
10 g <- function(x) {
11   x / (pi * (1 + x^2))
12 }
13
14 x <- seq(-10, 10, length.out = 1000)
15
16 data_f <- data.frame(x = x, y = f(x))
17 data_g <- data.frame(x = x, y = g(x))
18
19 p1 <- ggplot(data_f, aes(x, y)) +
20   geom_line(color = "blue") +
21   ggtitle("p.d.f. of the Cauchy Distribution") +
22   xlab("x") +
23   ylab("f(x)") +
24   ylim(0, 1/pi)
25
26 p2 <- ggplot(data_g, aes(x, y)) +
27   geom_line(color = "red") +
28   ggtitle("Function y = x * f(x)") +
29   xlab("x") +
30   ylab("x * f(x)") +
31   ylim(-1/(2*pi), 1/(2*pi))
32
33 grid.arrange(p1, p2, ncol = 2)

```

R code Exa 4.1.11 Failure Rate and Time to Failure I

```

1 #Page 212
2
3 library(MASS)
4
5 integrand <- function(y) {
6   return(y * 3 * y^(-4))
7 }
8
9 result <- integrate(integrand, lower = 1, upper =
  Inf)
10
11 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

```

1 #Page 214
2
3 library(MASS)
4
5 integrand <- function(x) {
6   return( 3 * x)
7 }
8
9 result <- integrate(integrand, lower = 0, upper = 1)
10
11 print(paste("The mean of X (E(X)) is:", fractions(
  result$value)))

```

R code Exa 4.1.13 Expectation of square root of X

```

1 #Page 214
2

```

```

3 library(MASS)
4
5 integrand <- function(x) {
6   return(x^(1/2) * (2 * x))
7 }
8
9 result <- integrate(integrand, lower = 0, upper = 1)
10 fractions(result$value)

```

R code Exa 4.1.14 Option Pricing

```

1 #Page 214
2
3 p <- (200 * 1.04 - 180) / (260 - 180)
4
5 c <- 60 * p / 1.04
6
7 print(paste("Resulting price", round(c, 2)))

```

R code Exa 4.1.16 Expectation of a Function of Two Variables

```

1 #Page 215
2
3 library(MASS)
4 integrand <- function(x, y) {
5   return((x^2 + y^2))
6 }
7
8 result <- integrate(Vectorize(function(y) {
9   integrate(function(x) integrand(x, y), lower = 0,
10     upper = 1)$value
11 }), lower = 0, upper = 1, abs.tol = 1e-9)

```

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

R code Exa 4.2.1 Expectation of a Linear Function

```
1 #Page 218
2
3 E_X <- 5
4
5 E_3X_minus_5 <- 3 * E_X - 5
6 print(E_3X_minus_5)
7
8 E_neg_3X_plus_15 <- -3 * E_X + 15
9 print(E_neg_3X_plus_15)
```

R code Exa 4.2.2 Investment I

```
1 #Page 218
2
3 n_sim <- 100000
4 p <- 0.03
5
6 set.seed(123)
7 returns_R1 <- runif(n_sim, min = -10, max = 20)
8 set.seed(102)
9 returns_R2 <- runif(n_sim, min = -4.5, max = 10)
10
11 expected_return_R1 <- mean(returns_R1)
12 expected_return_R2 <- mean(returns_R2)
13
14 VaR_R1 <- quantile(returns_R1, p)
15 VaR_R2 <- quantile(returns_R2, p)
16
17 total_return_R1 <- 120 * returns_R1
```

```

18 total_return_R2 <- 200 * returns_R2
19
20 total_VaR_R1 <- quantile(total_return_R1, p)
21 total_VaR_R2 <- quantile(total_return_R2, p)
22
23 expected_total_return_R1 <- 120 * expected_return_R1
24 expected_total_return_R2 <- 200 * expected_return_R2
25
26 cat("Total VaR for stock 1:", abs(round(total_VaR_R1
    ,0)), "\n")
27 cat("Total VaR for stock 2:", abs(round(total_VaR_R2
    ,0)), "\n")

```

R code Exa 4.2.3 Investment Portfolio I

```

1 #Page 220
2
3 mean_return_portfolio <- function(s1, s2) {
4   return (5 * s1 + 2.75 * s2)
5 }
6
7 s1 <- 54
8 s2 <- 110
9
10 mean_return <- mean_return_portfolio(s1, s2)
11
12 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

```

1 #Page 221
2

```

```

3 n <- 10
4
5 expected_value <- 1/n * n
6
7 cat("Expected value of X:", expected_value, "\n")

```

R code Exa 4.2.7 Expectation of a Combination of Random Variables

```

1 #Page 222
2
3 E_X <- 0
4 E_X2 <- 1
5
6 result <- E_X2 * (E_X2 - 8 * E_X2 * E_X + 16)
7
8 cat("E[X^2_1(X^2 - 4X^3)^2]:", result, "\n")

```

R code Exa 4.2.8 Repeated Filtering

```

1 #Page 222
2
3 mu <- integrate(function(x) (1 - x) * 4 * x^3, lower
4                   = 0, upper = 1)$value
5
6 E_Y <- mu^2
7
8 cat("Mean      :", mu, "\n")
9 cat("Expected value of Y:", E_Y, "\n")

```

R code Exa 4.2.10 Expected Waiting Time

```
1 #Page 224
2
3 library(MASS)
4
5 F <- function(x) {
6   ifelse(x <= 0, 0, 1 - exp(-2 * x))
7 }
8
9 E_X <- integrate(function(x) x * 2 * exp(-2 * x),
10   lower = 0, upper = Inf)$value
11
12 print(paste("Mean of X:", fractions(E_X)))
```

R code Exa 4.3.1 Stock Price Changes I

```
1 #Page 225
2
3 library(MASS)
4
5 lower_bound_A <- 25
6 upper_bound_A <- 35
7 lower_bound_B <- 15
8 upper_bound_B <- 45
9
10 Pr_B_less_than_25 <- (25 - lower_bound_B) / (upper_
11   bound_B - lower_bound_B)
12
13 print(paste("Probability that B < 25:", fractions(Pr
14   _B_less_than_25)))
```

R code Exa 4.3.2 Stock Price Changes II

```

1 #Page 226
2
3 library(MASS)
4
5 lower_bound_A <- 25
6 upper_bound_A <- 35
7 lower_bound_B <- 15
8 upper_bound_B <- 45
9
10 Var_A <- integrate(function(a) (a - 30)^2 / 10,
11                      lower_bound_A, upper_bound_A)$value
12
13 Var_B <- integrate(function(b) (b - 30)^2 / 30,
14                      lower_bound_B, upper_bound_B)$value
15
16 sd_A <- sqrt(Var_A)
17
18 sd_B <- sqrt(Var_B)
19
20 print(paste("Variance of A:", fractions(Var_A)))
21 cat("Variance of B:", fractions(Var_B), "\n")
22 cat("Standard deviation of A:", round(sd_A,2), "\n")
23 cat("Standard deviation of B:", round(sd_B,2), "\n")

```

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

```

1 #Page 227
2
3 values <- c(-2, 0, 1, 3, 4)
4 probabilities <- rep(1/5, 5)
5
6 mean_X <- sum(values * probabilities)
7
8 var_X <- sum(probabilities * (values - mean_X)^2)

```



```

9
10 sd_X <- sqrt(var_X)
11
12 cat("Mean of X:", mean_X, "\n")
13 cat("Variance of X:", var_X, "\n")
14 cat("Standard deviation of X:", round(sd_X,3), "\n")

```

R code Exa 4.3.4 Variance of a Discrete Distribution

```

1 #Page 227
2
3 values <- c(-2, 0, 1, 3, 4)
4 probabilities <- rep(1/5, 5)
5
6 E_X2 <- sum(probabilities * values^2)
7
8 mean_X <- sum(values * probabilities)
9
10 var_X <- E_X2 - mean_X^2
11
12 cat("Mean of X:", mean_X, "\n")
13 cat("Expectation of X^2:", E_X2, "\n")
14 cat("Variance of X:", var_X, "\n")

```

R code Exa 4.3.5 Modification of a Bernoulli Distribution

```

1 #Page 228
2
3 x_values <- c(0, 1, 10000)
4 probabilities <- c(0.5, 0.499, 0.001)
5
6 E_X <- sum(x_values * probabilities)
7

```

```

8 E_X2 <- sum((x_values^2) * probabilities)
9
10 Var_X <- E_X2 - (E_X^2)
11
12 cat("E(X) =", E_X, "\n")
13 cat("E(X^2) =", E_X2, "\n")
14 cat("Var(X) =", Var_X, "\n")

```

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

```

1 #Page 229
2
3 mu_X <- 1.2
4 var_X <- 4.56
5 a <- 4
6 b <- -7
7
8 var_Y <- a^2 * var_X
9
10 sd_Y <- a * sqrt(var_X)
11
12 cat("Variance of Y =", var_Y, "\n")
13 cat("Standard deviation of Y =", round(sd_Y,2), "\n")

```

R code Exa 4.3.7 Investment Portfolio II

```

1 #Page 230
2
3 mean_R1 <- 6
4 var_R1 <- 55
5 mean_R2 <- 4
6 var_R2 <- 28

```

```

7 fixed_rate <- 0.036
8 initial_investment <- 100000
9 price_R1 <- 60
10 price_R2 <- 48
11
12 s1 <- 524.7
13 s2 <- 609.7
14
15 s3 <- initial_investment - (s1 * price_R1 + s2 *
    price_R2)
16
17 mean_return <- s1 * mean_R1 + s2 * mean_R2 + fixed_
    rate * s3
18
19 variance_return <- (s1^2 * var_R1) + (s2^2 * var_R2)
20
21 cat("Mean return of the portfolio =", round(mean_
    return), "\n")
22 cat("Variance of the portfolio =", signif(variance_
    return,3), "\n")
23 cat("Shares of stock 1 =", s1, "\n")
24 cat("Shares of stock 2 =", s2, "\n")
25 cat("Amount invested at the fixed rate =", signif(s3
    ,4), "\n")
26
27 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 4.3.9 The Cauchy Distribution II

```

1 # Page 233
2
3 quantile_function_X <- function(p) {
4   return(tan(pi * (p - 0.5)))
5 }

```

```

6
7 Q_75 <- quantile_function_X(0.75)
8 Q_25 <- quantile_function_X(0.25)
9
10 IQR <- Q_75 - Q_25
11
12 cat("Interquartile Range (IQR):", IQR, "\n")
13
14
15 quantile_function_Y <- function(p) {
16   return(2 * quantile_function_X(p))
17 }
18
19 Q_Y_75 <- quantile_function_Y(0.75)
20 Q_Y_25 <- quantile_function_Y(0.25)
21
22 IQR_Y <- Q_Y_75 - Q_Y_25
23
24 cat("Interquartile Range (IQR) of Y:", IQR_Y, "\n")

```

R code Exa 4.4.2 Skewness of Binomial Distributions

```

1 #Page 236
2
3 n <- 10
4 p <- 0.25
5
6 mu <- n * p
7 sigma <- sqrt(n * p * (1 - p))
8 sigma <- round(sigma,4)
9
10 E_X_minus_mu_cubed <- sum((0:n - mu)^3 * dbinom(0:n,
11   n, p))
12 skewness <- E_X_minus_mu_cubed / sigma^3

```

```

13
14 cat("Mean (mu) =", mu, "\n")
15 cat("Standard deviation (sigma) =", round(sigma,3),
      "\n")
16 cat("E[(X - mu)^3] =", E_X_minus_mu_cubed, "\n")
17 cat("Skewness =", round(skewness,4), "\n")

```

R code Exa 4.4.3 Moment Generating Function

```

1 #Page 237
2
3 mgf_prime <- function(t) {
4   if (t < 1) {
5     return(1 / (1 - t)^2)
6   } else {
7     return(NA)
8   }
9 }
10
11 mgf_double_prime <- function(t) {
12   if (t < 1) {
13     return(2 / (1 - t)^3)
14   } else {
15     return(NA)
16   }
17 }
18
19 E_X <- mgf_prime(0)
20 E_X2 <- mgf_double_prime(0)
21
22 Var_X <- E_X2 - E_X^2
23
24 cat("E(X) =", E_X, "\n")
25 cat("E(X^2) =", E_X2, "\n")
26 cat("Var(X) =", Var_X, "\n")

```

R code Exa 4.5.1 Median of a Discrete Distribution I

```
1 # Page 242
2
3 probs <- c(0.1, 0.2, 0.3, 0.4)
4 values <- c(1, 2, 3, 4)
5
6 cdf <- cumsum(probs)
7
8 median_value <- values[min(which(cdf >= 0.5))]
9
10 cat("Median of the distribution:", median_value, "\n
    ")
```

R code Exa 4.5.2 Median of a Discrete Distribution II

```
1 # Page 242
2
3 probs <- c(0.1, 0.4, 0.3, 0.2)
4 values <- c(1, 2, 3, 4)
5
6 median_value <- mean(values)
7
8 cat("Midpoint median of the distribution:", median_
    value, "\n")
```

R code Exa 4.5.3 Median of a Continuous Distribution

```
1 #Page 242
```

```

2
3 cdf <- function(x) {
4   if (x < 0) {
5     return(0)
6   } else if (x >= 1) {
7     return(1)
8   } else {
9     return(x^4)
10  }
11 }
12
13 median_value <- uniroot(function(x) cdf(x) - 0.5,
14   lower = 0, upper = 1)$root
15 cat("The median of the distribution is:", median_
16   value, "\n")
17 # The answer may vary due to difference in
18   representation

```

R code Exa 4.5.7 Last Lottery Number I

```

1 #Page 244
2
3 p <- 0.001
4 q <- 0.999
5
6 mean_X <- p * (1 / (1 - q)^2)
7
8 cat("The mean of X is:", mean_X, "\n")

```

R code Exa 4.5.8 Last Lottery Number II

```

1 # Page 246

```

```

2
3 p <- 0.001
4 q <- 0.999
5
6 median_X <- log(0.5) / log(q)
7
8 median_X <- ceiling(median_X)
9
10 cat("The median of X is:", median_X, "\n")

```

R code Exa 4.5.9 Discrete Uniform Random Variable

```

1 #Page 246
2 values <- 1:6
3 probabilities <- rep(1/6, 6)
4
5 expected_value <- sum(values * probabilities)
6
7 median_value <- median(values)
8
9 cat("The value that minimizes the MSE is:", expected
    _value, "\n")
10
11 cat("The value that minimizes the MAE is:", median_
    value, "\n")

```

R code Exa 4.6.2 Test Scores I

```

1 #Page 249
2
3 library(MASS)
4
5 inner_integrand <- function(y, x) {

```



```

6   2 * x^2 * y + 0.5 * x
7 }
8
9 inner_integrand_vec <- Vectorize(inner_integrand,
  vectorize.args = "y")
10
11 integrate_inner <- function(x) {
12   sapply(x, function(xi) {
13     integrate(inner_integrand_vec, 0, 1, x = xi)$
      value
14   })
15 }
16
17 integrate_inner_vec <- Vectorize(integrate_inner)
18
19
20 mu_X <- integrate(integrate_inner_vec, 0, 1)$value
21 print(paste(" X =", fractions(mu_X)))
22
23 mu_Y <- mu_X
24
25 cov_xy <- function(x, y) {
26   (x - mu_X) * (y - mu_Y) * (2 * x * y + 0.5)
27 }
28
29 covariance <- integrate(
30   function(x) {
31     sapply(x, function(x) {
32       integrate(function(y) cov_xy(x, y), 0, 1)$
        value
33     })
34   }, 0, 1
35 )$value
36
37 print(paste("Cov(X, Y) =", fractions(covariance)))

```

R code Exa 4.6.3 Test Scores II

```
1 #Page 251
2
3 library(MASS)
4
5 cov_XY <- 1 / 144
6 var_X <- 11 / 144
7 var_Y <- 11 / 144
8
9 rho_XY <- cov_XY / sqrt(var_X * var_Y)
10
11 print(paste(" (X, Y) =", fractions(rho_XY)))
```

R code Exa 4.6.4 Uncorrelated Random Variables

```
1 # Page 251
2
3 values_X <- c(-1, 0, 1)
4 prob_X <- rep(1/3, 3)
5
6 values_Y <- values_X^2
7
8 E_X <- sum(values_X * prob_X)
9 E_Y <- sum((values_X^2) * prob_X)
10
11 E_XY <- sum((values_X * values_Y) * prob_X)
12
13 cov_XY <- E_XY - (E_X * E_Y)
14
15 cat("E(X) =", E_X, "\n")
16 cat("E(XY) =", E_XY, "\n")
```

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

R code Exa 4.6.5 Uniform Distribution Inside a Circle

```
1 # Page 252
2
3 library(cubature)
4
5 joint_pdf <- function(x, y) {
6   if (x^2 + y^2 <= 1) {
7     return(1 / (2 * pi))
8   } else {
9     return(0)
10  }
11 }
12
13 integrand_X <- function(xy) {
14   x <- xy[1]
15   y <- xy[2]
16   return(x * joint_pdf(x, y))
17 }
18
19
20 integrand_Y <- function(xy) {
21   x <- xy[1]
22   y <- xy[2]
23   return(y * joint_pdf(x, y))
24 }
25
26
27 integrand_XY <- function(xy) {
28   x <- xy[1]
29   y <- xy[2]
30   return(x * y * joint_pdf(x, y))
31 }
```

```

32
33 bounds <- cbind(c(-1, 1), c(-1, 1))
34
35 E_X <- adaptIntegrate(integrand_X, lowerLimit =
    bounds[,1], upperLimit = bounds[,2])$integral
36
37 E_Y <- adaptIntegrate(integrand_Y, lowerLimit =
    bounds[,1], upperLimit = bounds[,2])$integral
38
39 E_XY <- adaptIntegrate(integrand_XY, lowerLimit =
    bounds[,1], upperLimit = bounds[,2])$integral
40
41 cov_XY <- E_XY - (E_X * E_Y)
42
43 cat("E(X) =", E_X, "\n")
44 cat("E(Y) =", E_Y, "\n")
45 cat("E(XY) =", E_XY, "\n")
46 cat("Cov(X, Y) =", cov_XY, "\n")

```

R code Exa 4.7.2 Household Survey I

```

1 # Page 257
2
3 library(knitr)
4
5 joint_pmf <- matrix(c(0.040, 0.028, 0.012, 0.008,
    0.008, 0.004, 0, 0,
6                      0.048, 0.084, 0.100, 0.120,
7                      0.100, 0.060, 0.020, 0.004,
8                      0.004, 0.020, 0.040, 0.060,
9                      0.080, 0.044, 0.020, 0.012,
10                     0, 0.008, 0.012, 0.020, 0.020,
                        0.012, 0.008, 0.004),
    nrow = 4, byrow = TRUE)

```

```

11 marginal_pmf_X <- colSums(joint_pmf)
12 marginal_pmf_Y <- rowSums(joint_pmf)
13
14
15 E_Y_given_X <- numeric(length = ncol(joint_pmf))
16 for (x in 1:length(E_Y_given_X)) {
17   E_Y_given_X[x] <- sum((0:(nrow(joint_pmf) - 1)) *
18     joint_pmf[,x]) / marginal_pmf_X[x]
19 }
20 df <- data.frame(x = 1:8, 'E(Y|x)' = sprintf("%.3f",
21   E_Y_given_X))
22 knitr::kable(df, col.names = c("x", "E(Y given x)"),
23   align = "c")

```

R code Exa 4.7.4 Household Survey II

```

1 #Page 258
2
3 Z_values <- c(0.609, 1.057, 1.317, 1.442, 1.538,
4   1.533, 1.75, 2)
5 Pr_Z_values <- c(0.092, 0.140, 0.164, 0.208, 0.208,
6   0.120, 0.048, 0.020)
7 E_Z <- sum(Z_values * Pr_Z_values)
8 signif(E_Z,4)

```

R code Exa 4.7.6 Points from Uniform Distributions

```

1 #Page 259
2
3 library(MASS)

```

```

4
5 E_X <- 1 / 2
6 E_Y_given_X <- (1 / 2) * (E_X + 1)
7 E_Y <- (1 / 2) * (E_X + 1)
8 fractions(E_Y)

```

R code Exa 4.7.8 A Clinical Trial

```

1 #Page 261
2
3 x_values <- 0:40
4
5 sum_x_squared <- sum(x_values^2)
6 sum_x <- sum(x_values)
7 E_X_squared <- sum_x_squared / 41
8 E_X <- sum_x / 41
9
10 overall_MSE <- (1/((42^2)*43))*((-sum_x_squared/41)
    +(40*sum_x/41)+41)
11
12 cat(" Overall M.S.E.: ", signif(overall_MSE,4), "\n")

```

R code Exa 4.8.1 Choice of Gambles I

```

1 #Page 265
2
3 x_values <- c(500, -350)
4 f_x <- c(0.5, 0.5)
5
6 y_values <- c(40, 50, 60)
7 g_y <- c(1/3, 1/3, 1/3)
8
9 E_X <- sum(x_values * f_x)

```

```

10
11 E_Y <- sum(y_values * g_y)
12
13 cat("E(X) =", E_X, "\n")
14 cat("E(Y) =", E_Y, "\n")

```

R code Exa 4.8.2 Choice of Gambles II

```

1 #Page 265
2
3 U <- function(x) {
4   if (x >= 0) {
5     return(100 * log(x + 100) - 461)
6   } else {
7     return(x)
8   }
9 }
10
11 x_values <- c(500, -350)
12 f_x <- c(1/2, 1/2)
13
14 E_U_X <- sum(f_x * sapply(x_values, U))
15
16 y_values <- c(40, 50, 60)
17 g_y <- c(1/3, 1/3, 1/3)
18
19 E_U_Y <- sum(g_y * sapply(y_values, U))
20
21 cat("E[U(X)] =", round(E_U_X, 1), "\n")
22 cat("E[U(Y)] =", round(E_U_Y), "\n")
23 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 4.8.3 Linear Utility Function

```
1 #Page 267
2
3 x_values <- c(-3, 2.5, 6)
4 f_x <- c(0.5, 0.4, 0.1)
5
6 y_values <- c(-2, 1, 3)
7 g_y <- c(0.3, 0.4, 0.3)
8
9 E_X <- sum(f_x * x_values)
10
11 E_Y <- sum(g_y * y_values)
12
13 cat("E(X) =", E_X, "\n")
14 cat("E(Y) =", E_Y, "\n")
```

R code Exa 4.8.4 Cubic Utility Function

```
1 #Page 267
2
3 x_values <- c(-3, 2.5, 6)
4 f_x <- c(0.5, 0.4, 0.1)
5
6 y_values <- c(-2, 1, 3)
7 g_y <- c(0.3, 0.4, 0.3)
8
9 U <- function(x) {
10   return(x^3)
11 }
12
13 E_U_X <- sum(f_x * U(x_values))
14
15 E_U_Y <- sum(g_y * U(y_values))
16
```



```

17 U_0 <- U(0)
18
19 cat("E[U(X)] =", E_U_X, "\n")
20 cat("E[U(Y)] =", E_U_Y, "\n")
21 cat("U(0) =", U_0, "\n")

```

R code Exa 4.8.5 Logarithmic Utility Function

```

1 #Page 268
2
3 x_values <- c(-3, 2.5, 6)
4 f_x <- c(0.5, 0.4, 0.1)
5
6 y_values <- c(-2, 1, 3)
7 g_y <- c(0.3, 0.4, 0.3)
8
9 U <- function(x) {
10   return(log(x + 4))
11 }
12
13 E_U_X <- sum(f_x * U(x_values))
14
15 E_U_Y <- sum(g_y * U(y_values))
16
17 U_0 <- U(0)
18
19 cat("E[U(X)] =", round(E_U_X,4), "\n")
20 cat("E[U(Y)] =", round(E_U_Y,4), "\n")
21 cat("U(0) =", round(U_0,4), "\n")

```

R code Exa 4.8.6 Quadratic Utility Function

```

1 #Page 268

```

```

2
3 U <- function(x) {
4   return(x^2)
5 }
6
7 E_U_lottery <- (1/4) * U(36) + (3/4) * U(0)
8
9 print(E_U_lottery)

```

R code Exa 4.8.7 Square Root Utility Function

```

1 #Page 268
2
3 U <- function(x) {
4   return(x^(1/2))
5 }
6
7 E_U_lottery <- (1/4) * U(36) + (3/4) * U(0)
8
9 print(E_U_lottery)

```

R code Exa 4.8.9 Bounding a Random Variable

```

1 #Page 269
2
3 prob_threshold <- 10^4.8 / (10^6 + 10^4.8)
4
5 cat("Probability threshold for Pr(P >= p0):", round
    (prob_threshold,4), "\n")

```

R code Exa 4.8.10 Investment II

```
1 #Page 270
2 U <- function(x) {
3   if (x >= 0) {
4     return(x^0.8)
5   } else {
6     return(x)
7   }
8 }
9
10 integral1_negative <- integrate(function(r) 120 * r
11   * (1/30), lower = -10, upper = 0)$value
12 integral1_positive <- integrate(function(r) (120 * r
13   )^0.8 * (1/30), lower = 0, upper = 20)$value
14 E_U_120R1 <- integral1_negative + integral1_positive
15
16 integral2_negative <- integrate(function(r) 200 * r
17   * (1/14.5), lower = -4.5, upper = 0)$value
18 integral2_positive <- integrate(function(r) (200 * r
19   )^0.8 * (1/14.5), lower = 0, upper = 10)$value
20 E_U_200R2 <- integral2_negative + integral2_positive
21
22 cat("E[U(120R1)] =", round(E_U_120R1,1), "\n")
23 cat("E[U(200R2)] =", round(E_U_200R2,1), "\n")
```

Chapter 5

Special Distributions

R code Exa 5.2.7 Group Testing

```
1 # Page 278
2
3 n_groups <- 10
4 group_size <- 100
5 disease_prob <- 0.002
6
7 prob_Z1_positive <- 1 - dbinom(0, size = group_size,
  prob = disease_prob)
8
9 cat("Pr(Z1,i > 0) =", signif(prob_Z1_positive,3), "\n"
  n")
10
11 mean_100Y1 <- group_size * n_groups * prob_Z1_
  positive
12
13 cat("Mean of 100Y1:", round(mean_100Y1), "\n")
14
15 Pr_Z2_i_k_gt_0 <- 1 - 0.998^n_groups
16
17 cat("Pr(Z2,i,k > 0):", signif(Pr_Z2_i_k_gt_0,3), "\n"
  ")")
```

```

18
19 mean_10Y2 <- n_groups * group_size * Pr_Z2_i_k_gt_0
20
21 cat("Mean of 10Y2:", round(mean_10Y2,2), "\n")
22
23 expected_total_tests <- 10 + 1.81 + 19.82
24
25 cat("Expected total number of tests:", expected_
    total_tests, "\n")

```

R code Exa 5.4.2 Customer Arrivals I

```

1 #Page 289
2
3 lambda <- 3600 * 0.00125
4
5 cat("Mean for the Poisson distribution:", lambda, "\n")
6
7 n <- 3600
8 p <- 0.00125
9
10 lambda <- n * p
11
12 x <- 0:9
13
14 binomial_probs <- dbinom(x, n, p)
15
16 poisson_probs <- dpois(x, lambda)
17
18 results <- data.frame(
19   x = x,
20   Binomial = round(binomial_probs, 5),
21   Poisson = round(poisson_probs, 5)
22 )

```

```
23
24 print(results)
```

R code Exa 5.4.3 Customer Arrivals II

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

R code Exa 5.4.4 Approximating a Probability

```
1 #Page 292
2
3 n <- 200
4 p <- 0.01
5 lambda <- n*p
6 Pr <- 1 - ppois(3, lambda)
7 cat(" Pr(X      4) =", round(Pr,4),"\\n")
8 #The answer varies due to rounding off values
```

R code Exa 5.4.6 Radioactive Particles

```
1 #Page 293
2
3 lambda <- 6
4 Pr <- 1 - ppois(9, lambda)
5 cat(" Pr(X      10) =", round(Pr,4),"\\n")
6 #The answer varies due to rounding off values
```

R code Exa 5.4.7 Cryptosporidium in Drinking Water

```
1 #Page 294
2
3 lambda <- 0.2
4 p <- 0.1
5 t <- 100
6
7 prob <- 1 - exp(-p * lambda * t)
8
9 round(prob,2)
```

R code Exa 5.5.3 Triples in the Lottery I

```
1 #Page 298
2
3 No_of_triples <- 10
4 No_of_daily_numbers <- 1000
5 p <- No_of_triples / No_of_daily_numbers
6
7 print(p)
```

R code Exa 5.5.4 Triples in the Lottery II

```
1 #Page 300
2
3 p <- 0.01
4
5 E_X <- (1 - p) / p
```

```
6
7 expected_days <- E_X + 1
8
9 print(expected_days)
```

R code Exa 5.6.4 Probabilities for a Normal Distribution

```
1 #Page 308
2
3 mean_X <- 5
4 sd_X <- 2
5 lower_bound_X <- 1
6 upper_bound_X <- 8
7
8 lower_bound_Z <- (lower_bound_X - mean_X) / sd_X
9 upper_bound_Z <- (upper_bound_X - mean_X) / sd_X
10
11 P_Z_lower <- pnorm(lower_bound_Z)
12 P_Z_upper <- pnorm(upper_bound_Z)
13
14 P_X <- P_Z_upper - P_Z_lower
15
16 round(P_X,4)
17 #The answer varies due to rounding off values
```

R code Exa 5.6.5 Quantiles of Normal Distributions

```
1 #Page 308
2
3 mean_X <- 1.329
4 sd_X <- 0.4844
5
6 q_05 <- qnorm(0.05)
```



```
7
8 quantile_05_X <- mean_X + sd_X * q_05
9
10 round(quantile_05_X,4)
```

R code Exa 5.6.6 Heights of Men and Women

```
1 # Page 310
2
3 mean_W <- 65
4 mean_M <- 68
5 var_W <- 1
6 var_M <- 9
7
8 mean_diff <- mean_W - mean_M
9
10 var_diff <- var_W + var_M
11
12 sd_diff <- sqrt(var_diff)
13
14 z <- mean_diff / sd_diff
15
16 prob <- pnorm(z)
17
18 round(prob,3)
```

R code Exa 5.6.7 Determining a Sample Size

```
1 # Page 311
2
3 z_value <- 1.96
4 std_dev <- 3
5
```

```

6 min_n <- (z_value * std_dev)^2
7
8 min_n_rounded <- ceiling(min_n)
9
10 print(min_n_rounded)

```

R code Exa 5.6.8 Interval for Mean

```

1 #Page 311
2
3 Xn <- 68
4
5 n <- 36
6
7 std_dev <- 3
8
9 z <- 1.96
10
11 margin_error <- z * (std_dev / sqrt(n))
12
13 cat("Margin of Error:", margin_error, "\n")

```

R code Exa 5.6.9 Failure Times of Ball Bearings

```

1 #Page 312
2
3 library(ggplot2)
4
5 mu <- 4.15
6 sigma <- 0.53342
7 p <- 0.9
8 set.seed(123)
9 samples <- rlnorm(1000, meanlog = mu, sdlog = sigma)

```

```

10 data_df <- data.frame(data = samples)
11
12 x_seq <- seq(min(samples), max(samples), length.out
    = 1000)
13
14 y_seq <- dlnorm(x_seq, meanlog = mu, sdlog = sigma)
15
16 ggplot(data = data_df, aes(x = data)) +
17   geom_histogram(aes(y=..density..), bins = 30,
    color="black", fill="white") +
18   geom_line(data = data.frame(x = x_seq, y = y_seq),
    aes(x = x, y = y), color = "blue", size = 1) +
19   labs(x = "Time to Failure", y = "Density") +
20   theme_minimal()
21
22 q <- exp(mu + sigma * qnorm(p))
23 signif(q, 5)

```

R code Exa 5.6.10 Stock and Option Prices

```

1 #Page 313
2
3 library(qrmtools)
4
5 S0 <- 100
6 r <- 0.06
7 sigma <- 0.1
8 K <- 100
9 T <- 1
10 t <- 0
11
12 option_price <- Black_Scholes(t, S0, r, sigma, K, T,
    type = "call")
13
14 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

```
1 #Page 317
2
3 f1 <- function(x) { x * exp(-x) }
4 f2 <- function(x) { x^2 * exp(-x) }
5
6 integral1 <- integrate(f1, 0, Inf)$value
7 integral2 <- integrate(f2, 0, Inf)$value
8
9
10 mean_lifetime <- integral1
11 variance_lifetime <- integral2 - integral1^2
12
13 cat("Mean of the lifetime:", mean_lifetime, "\n")
14 cat("Variance of the lifetime:", variance_lifetime,
    "\n")
```

R code Exa 5.8.4 Castaneda v Partida

```
1 #Page 330
2
3 library(ggplot2)
4
5 beta_values <- seq(0, 100, by = 1)
6
7 probabilities <- numeric(length(beta_values))
8
9 threshold <- 0.6328
10
11 for (i in seq_along(beta_values)) {
```

```

12  beta <- beta_values[i]
13  alpha <- 3.785 * beta
14  alpha_new <- alpha + 100
15  beta_new <- beta + 120
16  probabilities[i] <- pbeta(threshold, alpha_new,
    beta_new)
17 }
18
19 data <- data.frame(beta = beta_values, probability =
    probabilities)
20
21 y_max <- max(probabilities)
22
23 ggplot(data, aes(x = beta, y = probability)) +
24   geom_line(color = "blue") +
25   geom_segment(aes(x = 51.5, xend = 51.5, y = 0,
    yend = pmin(probability[52], y_max)),
    linetype = "dashed", color = "red") +
26   geom_segment(aes(x = 0, xend = 51.5, y = 0.5, yend
    = 0.5),
    linetype = "dashed", color = "red") +
27   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))) +
28   annotate("text", x = 51.5, y = 0.05, label = " =
    51.5", color = "red", angle = 90, vjust =
    -0.5) +
29   annotate("text", x = 20, y = 0.5, label = "Pr(P
    0.6328 | X = 100) = 0.5", color = "red",
    vjust = -0.5) +
30   theme_minimal()
31
32
33
34

```

R code Exa 5.8.5 A Clinical Trial

```

1 #Page 330
2
3 alpha <- 1.18
4 beta <- 2.35
5
6 n <- 40
7 x <- 22
8
9 alpha_new <- alpha + x
10 beta_new <- beta + (n - x)
11
12 expected_p <- alpha_new / (alpha_new + beta_new)
13
14 cat("Expected value E(P|X = 22):", round(expected_p
      ,4), "\n")

```

R code Exa 5.9.2 Attendance at a Baseball Game

```

1 #Page 334
2
3 library(iterpc)
4
5 p1 <- 0.23
6 p2 <- 0.59
7 p3 <- 0.18
8
9 n1 <- 7
10 n2 <- 8
11 n3 <- 5
12
13 n <- n1 + n2 + n3
14
15 probability <- multichoose(c(n1,n2,n3)) * (p1^n1) *
      (p2^n2) * (p3^n3)
16

```

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

R code Exa 5.9.3 Blood Types

```
1 #Page 334
2
3 p_A <- 0.360
4 p_B <- 0.123
5 p_AB <- 0.038
6 p_0 <- 0.479
7
8 probability_same_blood_type <- p_A^2 + p_B^2 + p_AB
   ^2 + p_0^2
9
10 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

```
1 #Page 343
2
3 mean_wife <- 66.8
4 sd_wife <- 2
5 mean_husband <- 70
6 sd_husband <- 2
7 correlation <- 0.68
8
9 mean_diff <- mean_wife - mean_husband
10 var_diff <- sd_wife^2 + sd_husband^2 - 2 *
   correlation * sd_wife * sd_husband
11
```

```
12 sd_diff <- sqrt(var_diff)
13
14 probability <- 1 - pnorm(0, mean_diff, sd_diff)
15
16 cat("Mean difference:", mean_diff, "\n")
17 cat("Variance of the difference:", var_diff, "\n")
18 cat("Standard deviation of the difference:", sd_diff
    , "\n")
19
20 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

```
1 # Page 347
2
3 n1 <- 10
4
5 p <- 0.5
6
7 k1 <- 5
8
9 probability_10_flips <- dbinom(k1, n1, p)
10
11 n2 <- 100
12
13 k2 <- 50
14
15 probability_100_flips <- dbinom(k2, n2, p)
16
17 probability_10_flips_range <- pbinom(6, n1, p) -
  pbinom(3, n1, p)
18
19 probability_100_flips_range <- pbinom(60, n2, p) -
  pbinom(39, n2, p)
```

```

20
21 cat("The probability of getting exactly", k1, "heads
    in", n1, "flips is:", round(probability_10_flips
    ,4) , "\n")
22 cat("The probability of getting exactly", k2, "heads
    in", n2, "flips is:", round(probability_100_flips
    ,4) , "\n")
23
24 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")
25 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

```

1 # Page 350
2
3 sigma_squared <- 4
4 epsilon <- 1
5 confidence_level <- 0.99
6
7 required_sample_size <- sigma_squared / (1 -
    confidence_level)
8
9 cat("The required sample size is at least:",
    required_sample_size, "\n")

```

R code Exa 6.2.2 A Simulation

```

1 #Page 351
2

```

```

3 epsilon <- 0.005
4 confidence_level <- 0.98
5 required_samples <- 1 / (epsilon^2 * (1 - confidence
  _level))
6
7 cat("Required number of samples for the desired
  precision:", format(required_samples, scientific =
    FALSE), "\n")

```

R code Exa 6.2.3 Tossing a Coin

```

1 #Page 351
2
3 n <- 15
4 p <- 0.5
5
6 lower_bound <- ceiling(0.4 * n)
7 upper_bound <- floor(0.6 * n)
8
9 prob <- sum(dbinom(lower_bound:upper_bound, size = n
  , prob = p))
10
11 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

```

1 #Page 354
2
3 set.seed(123)
4
5 lambda <- 0.446
6 n <- 100

```

```

7
8 sample <- rexp(n, rate = lambda)
9 sample <- sample[sample <= 10]
10
11 hist(sample, breaks = 10, probability = TRUE, main =
    "Histogram of Service Times with Exponential PDF
    ",
12       xlab = "Service Time", ylab = "Density", col =
        "lightblue", border = "black", xlim = c(0,
        10))
13
14 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

```

1 #Page 354
2
3 library(ggplot2)
4
5 set.seed(123)
6
7 lambda <- 0.446
8 n <- 100
9
10 sample <- rexp(n, rate = lambda)
11 sample <- sample[sample <= 10]
12
13 data <- data.frame(service_time = sample)
14
15 ggplot(data, aes(x = service_time)) +
16   geom_histogram(aes(y = ..density..), bins = 20,
        fill = "lightblue", color = "black", boundary =
        0.5) +
17   stat_function(fun = function(x) dexp(x, rate =

```

```

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

```

R code Exa 6.2.6 Binomial Random Variable

```

1  # Page 356
2
3  bound_fn <- function(n) {
4    2 * (0.9811 ^ n)
5  }
6
7  chebyshev_bound <- function(n) {
8    25 / n
9  }
10
11 n_values <- c(100, 200, 300)
12 for (n in n_values) {
13   bound_val <- bound_fn(n)
14   chebyshev_bound_val <- chebyshev_bound(n)
15   cat("For n =", n, "\n")
16   cat("Bound value:", round(bound_val, 4), "\n")
17   cat("Chebyshev bound:", round(chebyshev_bound_val
18     , 4), "\n\n")

```

R code Exa 6.3.1 A Large Sample I

```

1  #Page 360
2
3  n <- 100

```

```

4 p <- 0.5
5
6 x <- 0:n
7 binomial_pmf <- dbinom(x, size = n, prob = p)
8
9 mu <- n * p
10 sigma <- sqrt(n * p * (1 - p))
11
12 x_norm <- seq(0, n, length.out = 1000)
13 normal_pdf <- dnorm(x_norm, mean = mu, sd = sigma)
14
15 plot(x, binomial_pmf, type = "h", lwd = 10, col = "
    skyblue", main = "Binomial Distribution vs Normal
    Approximation",
16       xlab = "Number of Patients Surviving", ylab = "
    Probability Mass Function")
17 lines(x_norm, normal_pdf, col = "red", lty = 2, lwd
    = 2)
18 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

```

1 #Page 362
2
3 n <- 900
4 p <- 0.5
5 mu <- n * p
6 sigma <- sqrt(n * p * (1 - p))
7
8 z_score <- (495 - mu) / sigma
9
10 prob_more_than_495 <- 1 - pnorm(z_score)
11

```

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

R code Exa 6.3.3 Sampling from a Uniform Distribution

```
1 #Page 362
2
3 n <- 12
4 mu <- 1/2
5 sigma <- 1/12
6
7 z <- 0.1 / sigma
8
9 prob_within_01 <- 2 * pnorm(z) - 1
10
11 round(prob_within_01,5)
```

R code Exa 6.3.5 Determining a Simulation Size

```
1 #Page 363
2
3 n <- (1/(0.005^2)) * round(qnorm(0.99),3)^2
4
5 print(n)
6
7 n_adjusted <- ceiling(n * 1/4)
8
9 print(n_adjusted)
```

R code Exa 6.3.9 Examination Questions I

```

1 #Page 367
2
3 n <- 99
4 pass_threshold <- 60
5
6 p <- 1 - (1:n)/100
7
8 mean_correct <- sum(p)
9 var_correct <- sum(p * (1 - p))
10
11 cat("Mean:", mean_correct, "\n")
12 cat("Variance:", var_correct, "\n")
13
14
15 pass_prob_clt <- 1 - pnorm((pass_threshold - mean_
    correct) / sqrt(var_correct))
16
17 round(pass_prob_clt,3)

```

R code Exa 6.4.1 A Large Sample II

```

1 #Page 371
2
3 library(ggplot2)
4
5 n <- 100
6 p <- 0.5
7 mu <- n * p
8 sigma <- sqrt(n * p * (1 - p))
9
10 x <- seq(30, 70, by = 0.1)
11
12 binom_cdf <- pbinom(x, size = n, prob = p)
13
14 norm_cdf <- pnorm(x, mean = mu, sd = sigma)

```



```

15
16 data <- data.frame(
17   x = x,
18   Binomial = binom_cdf,
19   Normal = norm_cdf
20 )
21
22 # Plot the CDFs
23 ggplot(data, aes(x = x)) +
24   geom_line(aes(y = Binomial, color = "Binomial")) +
25   geom_line(aes(y = Normal, color = "Normal")) +
26   labs(
27     title = "CDFs of Binomial and Normal
28             Distributions",
29     x = "x",
30     y = "CDF",
31     color = "Distribution"
32   ) +
33   theme_minimal()

```

R code Exa 6.4.2 A Large Sample III

```

1 #Page 373
2
3 n <- 100
4 p <- 0.5
5 mean_binomial <- n * p
6 var_binomial <- n * p * (1 - p)
7
8 mean_normal <- mean_binomial
9 sd_normal <- sqrt(var_binomial)
10 normal_dist <- rnorm(10000, mean_normal, sd_normal)
11
12 binomial_dist <- rbinom(10000, n, p)
13

```

```

14 x_values <- seq(30, 69, by = 1)
15
16 normal_cdf <- pnorm(x_values, mean_normal, sd_normal
17 )
18 binomial_cdf <- pbinom(x_values, n, p)
19
19 plot(x_values, normal_cdf, type = "l", col = "blue",
20       xlab = "x", ylab = "Pr(X <= x)", main = "
21       Comparison of Normal and Binomial CDFs")
22 lines(x_values, binomial_cdf, type = "l", col = "red
23 ")
24 legend("bottomright", legend = c("Normal
25 Approximation", "Binomial Distribution"), col = c
26 ("blue", "red"), lty = 1)

```

R code Exa 6.4.3 Examination Questions II

```

1 #Page 373
2
3 mean_X <- 49.5
4 sd_X <- 4.08
5
6 prob <- 1 - pnorm(59.5, mean_X, sd_X)
7
8 round(prob,3)

```

R code Exa 6.4.4 Coin Tossing

```

1 #Page 374
2
3 mean_X <- 10
4 sd_X <- sqrt(20 * 0.5 * 0.5)
5

```

```
6 prob <- pnorm(10.5, mean_X, sd_X) - pnorm(9.5, mean_  
  X, sd_X)  
7  
8 round(prob,3)
```

Chapter 7

Estimation

R code Exa 7.2.6 Lifetimes of Fluorescent Lamps I

```
1 #Page 388
2
3 library(ggplot2)
4 library(gridExtra)
5
6 gamma_density <- function(x, shape, rate) {
7   dgamma(x, shape, rate)
8 }
9
10 n <- 5
11 lifetimes <- c(2911, 3403, 3237, 3509, 3118)
12 y <- sum(lifetimes)
13 prior_shape1 <- 4
14 prior_rate1 <- 20000
15 posterior_shape1 <- n + prior_shape1
16 posterior_rate1 <- y + prior_rate1
17
18 prior_shape2 <- 1
19 prior_rate2 <- 1000
20 posterior_shape2 <- n + prior_shape2
21 posterior_rate2 <- y + prior_rate2
```

```

22
23 theta <- seq(0, 0.0015, length.out = 1000)
24
25 prior_density1 <- gamma_density(theta, prior_shape1,
    prior_rate1)
26 posterior_density1 <- gamma_density(theta, posterior
    _shape1, posterior_rate1)
27 prior_density2 <- gamma_density(theta, prior_shape2,
    prior_rate2)
28 posterior_density2 <- gamma_density(theta, posterior
    _shape2, posterior_rate2)
29
30 data1 <- data.frame(
31   theta = theta,
32   Prior = prior_density1,
33   Posterior = posterior_density1
34 )
35
36 data2 <- data.frame(
37   theta = theta,
38   Prior = prior_density2,
39   Posterior = posterior_density2
40 )
41
42 p1 <- ggplot(data1, aes(x = theta)) +
43   geom_line(aes(y = Prior, color = "Prior")) +
44   geom_line(aes(y = Posterior, color = "Posterior"))
    +
45   labs(
46     title = "Gamma Distributions: Prior and
        Posterior (First Set of Parameters)",
47     x = expression(theta),
48     y = "Density",
49     color = "Distribution"
50   ) +
51   theme_minimal()
52
53 p2 <- ggplot(data2, aes(x = theta)) +

```

```

54   geom_line(aes(y = Prior, color = "Prior")) +
55   geom_line(aes(y = Posterior, color = "Posterior"))
56   +
57   labs(
58     title = "Gamma Distributions: Prior and
59             Posterior (Second Set of Parameters)",
60     x = expression(theta),
61     y = "Density",
62     color = "Distribution"
63   ) +
64   theme_minimal()
65   grid.arrange(p1, p2, nrow = 2)

```

R code Exa 7.2.8 Lifetimes of Fluorescent Lamps II

```

1  #Page 392
2
3  k <- 9.555 * (10^41)
4
5  f_x6_given_x <- function(x6, k) {
6    k / (x6 + 36178)^10
7  }
8
9  probb_greater_than_3000 <- integrate(function(x6) f_
10    x6_given_x(x6, k), lower = 3000, upper = Inf)$
11    value
12
13  round(probb_greater_than_3000,4)
14
15  k_new <- 1.542 * (10^26)
16
17  f_x6_given_x_new <- function(x6, k_new) {
18    k_new / (x6 + 17178)^7
19  }

```

```

18
19 prob_greater_than_3000_new <- integrate(function(x6)
      f_x6_given_x_new(x6, k_new), lower = 3000, upper
      = Inf)$value
20
21 round(prob_greater_than_3000_new,5)

```

R code Exa 7.3.3 Glove Use by Nurses

```

1 #Page 396
2
3 observed_before = 13
4 total_before = 51
5 observed_after = 50
6 total_after = 56
7
8 prior1_alpha = 1
9 prior1_beta = 1
10
11 prior2_alpha = 13
12 prior2_beta = 38
13
14 posterior1_alpha = prior1_alpha + observed_after
15 posterior1_beta = prior1_beta + (total_after -
      observed_after)
16
17 posterior2_alpha = prior2_alpha + observed_after
18 posterior2_beta = prior2_beta + (total_after -
      observed_after)
19
20 posterior1_mean = posterior1_alpha / (posterior1_
      alpha + posterior1_beta)
21 round(posterior1_mean,2)
22 prob_posterior1_gt_before = 1 - pbeta(2 * observed_
      before / total_before, posterior1_alpha,

```

```

    posterior1_beta)
23 prob_posterior1_gt_before
24
25 posterior2_mean = posterior2_alpha / (posterior2_
    alpha + posterior2_beta)
26 round(posterior2_mean,2)
27 prob_posterior2_gt_before = 1 - pbeta(2 * observed_
    before / total_before, posterior2_alpha,
    posterior2_beta)
28 round(prob_posterior2_gt_before,2)

```

R code Exa 7.3.8 Automobile Emissions

```

1 #Page 399
2
3 n <- 46
4 sigma_sq <- 0.25
5 mu_0 <- 2
6 v_2 <- 1.0
7 x_n <- 1.329
8
9 mu_1 <- (sigma_sq * mu_0 + n * v_2 * x_n) / (sigma_
    sq + n * v_2)
10 round(mu_1,3)
11
12 v_1 <- sigma_sq * v_2 / (sigma_sq + n * v_2)
13 round(v_1,4)

```

R code Exa 7.3.10 Calorie Counts on Food Labels

```

1 #Page 400
2
3 n <- 20

```



```

4 theta_prior_mean <- 0
5 theta_prior_var <- 60
6 theta_likelihood_var <- 100
7 x_bar <- 0.125
8
9 theta_posterior_mean <- (theta_likelihood_var *
    theta_prior_mean + n * theta_prior_var * x_bar) /
    (theta_likelihood_var + n * theta_prior_var)
10 round(theta_posterior_mean,4)
11
12 theta_posterior_var <- theta_likelihood_var * theta_
    prior_var / (theta_likelihood_var + n * theta_
    prior_var)
13 round(theta_posterior_var,2)
14
15 prob_theta_gt_1 <- 1 - pnorm(1, mean = theta_
    posterior_mean, sd = sqrt(theta_posterior_var))
16 round(prob_theta_gt_1,4)

```

R code Exa 7.3.12 Lifetimes of Electronic Components I

```

1 #Page 402
2
3 n <- 3
4 alpha <- 1
5 beta <- 2
6 y <- sum(c(3, 1.5, 2.1))
7 print(y)
8
9 alpha_posterior <- alpha + n
10 print(alpha_posterior)
11
12 beta_posterior <- beta + y
13 print(beta_posterior)

```

R code Exa 7.4.5 Estimating the Parameter of a Bernoulli Distribution

```
1 #Page 411
2
3 set.seed(1)
4
5 alpha <- 22
6 beta <- 18
7
8 samples <- rbeta(10^6, alpha, beta)
9
10 mean_value <- mean(samples)
11 round(mean_value,2)
12
13 median_value <- quantile(samples, 0.5)
14 round(median_value,4)
```

R code Exa 7.4.8 Lifetimes of Electronic Components II

```
1 #Page 414
2
3 integrand <- function(theta) {
4   ((8.6^4)/6) * (theta^2) * exp(-8.6 * theta)
5 }
6
7 bayes_estimate <- integrate(integrand, lower = 0,
8   upper = Inf)$value
9 round(bayes_estimate,3)
10
11 alpha <- 8.6
12 beta <- 4
```

```
13
14 mean_1_theta <- alpha / beta
15
16 print(mean_1_theta)
```

R code Exa 7.5.2 Lifetimes of Electronic Components III

```
1 #Page 419
2
3 log_likelihood <- function(theta) {
4   3 * log(theta) - 6.6 * theta
5 }
6
7 mle_result <- optimize(log_likelihood, interval = c
8   (0.01, 10), maximum = TRUE)
9
10 mle_theta <- mle_result$maximum
11
12 round(mle_theta,3)
```

R code Exa 7.5.3 Test for a Disease

```
1 #Page 419
2
3 mle_theta <- function(x) {
4   if (x == 0) {
5     return(0.1)
6   } else if (x == 1) {
7     return(0.9)
8   } else {
9     stop("Invalid value for x. Must be 0 or 1.")
10  }
11 }
```

```

12
13 X <- 0
14
15 theta_hat <- mle_theta(X)
16
17 cat("When x=0, theta which maximizes the likelihood
      is", theta_hat, "\n")
18
19 X <- 1
20
21 theta_hat <- mle_theta(X)
22
23 cat("When x=1, theta which maximizes the likelihood
      is", theta_hat, "\n")

```

R code Exa 7.6.2 Lifetimes of Electronic Components IV

```

1 #Page 427
2
3 theta_hat <- 0.455
4
5 psi_hat <- 1 / theta_hat
6
7 round(psi_hat, 1)

```

R code Exa 7.6.6 Sampling from a Gamma Distribution

```

1 #Page 430
2
3 alpha_0 <- 3.679
4
5 alpha_1 <- alpha_0 - (round(digamma(alpha_0), 4) -
      1.220) / round(trigamma(alpha_0), 4)

```

```

6 round(alpha_1,3)
7
8 alpha_2 <- alpha_1 - (digamma(alpha_1) - 1.220) /
  trigamma(alpha_1)
9 round(alpha_2,3)
10
11 alpha_3 <- alpha_2 - (digamma(alpha_2) - 1.220) /
  trigamma(alpha_2)
12 round(alpha_3,4)
13 #The answer provided in the textbook is wrong.

```

R code Exa 7.6.10 Sampling from a Uniform Distribution

```

1 #Page 431
2
3 X <- c(0.2, 0.99, 0.01)
4 n <- length(X)
5
6 X_bar <- mean(X)
7 S_squared <- var(X)
8
9 theta_mom <- (X_bar - sqrt(n / (n-1)) * sqrt(S_
  squared))/2
10
11 cat("X_bar:", X_bar, "\n")
12 cat("Method of Moments estimate (theta):", round(
  theta_mom,1), "\n")

```

R code Exa 7.6.12 Prussian Army Deaths

```

1 #Page 432
2
3 library(MASS)

```

```

4
5 alpha <- 196
6 beta <- 280
7 n <- 280
8
9 mle_theta <- alpha / beta
10
11 mle_variance_theoretical <- mle_theta / n
12
13 fractions(mle_theta)
14 # The answer may vary due to difference in
    representation.

```

R code Exa 7.6.13 Lifetimes of Electronic Components V

```

1 #Page 433
2
3 library(MASS)
4 n <- 3
5 alpha <- 4
6 beta <- 8.6
7
8 X <- c(3, 1.5, 2.1)
9 mle_theta <- n / sum(X)
10
11 fractions(mle_theta)
12 #The answer may vary due to difference in
    representation.

```

R code Exa 7.6.15 Heights and Weights

```

1 #Page 435
2

```

```

3 mu1 <- 69.60
4 mu2 <- 194.75
5 sigma1_sq <- 2.87
6 sigma2_sq <- 14.82
7 rho <- 0.1764
8
9 heights <- c(72, 70, 73, 68, 65, NA)
10 weights <- c(197, 204, 208, NA, NA, 170)
11
12 conditional_mean_X4_2 <- mu2 + rho * sqrt(sigma2_sq)
    * ((heights[4] - mu1)/sqrt(sigma1_sq))
13 conditional_mean_X4_2
14
15 conditional_variance_X4_2 <- (1 - rho^2) * sigma2_sq
    ^2
16 round(conditional_variance_X4_2,1)
17 #The answer provided in the textbook is wrong.

```

R code Exa 7.6.16 Mixture of Normal Distributions

```

1 #Page 736
2
3 mu_1 <- -7.65
4 mu_2 <- 7.36
5 sigma2_0 <- 46.28
6 p_0 <- 0.5
7
8 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)
9
10 q_0_10 <- p_0 * exp(-(data[1] + -mu_1)^2 / (2 *
    sigma2_0)) /
11 (p_0 * exp(-(data[1] + -mu_1)^2 / (2 * sigma2_0))
    +

```

```

12      p_0 * exp(-(data[1] - mu_2)^2 / (2 * sigma2_0))
13      )
14      round(q_0_10,4)
15      q_0_8 <- p_0 * exp(-(data[2] + -mu_1)^2 / (2 *
16      sigma2_0)) /
17      (p_0 * exp(-(data[2] + -mu_1)^2 / (2 * sigma2_0))
18      +
19      p_0 * exp(-(data[2] - mu_2)^2 / (2 * sigma2_0))
20      )
21      round(q_0_8,4)

```

Chapter 8

Sampling Distributions of Estimators

R code Exa 8.1.5 Lifetimes of Electronic Components I

```
1 #Page 468
2
3 library(gamlss.dist)
4
5 cdf_gamma <- function(x) {
6   pgamma(x, shape = 3, scale = 1)
7 }
8
9 prob <- cdf_gamma(3.33) - cdf_gamma(2.73)
10 round(prob, 3)
```

R code Exa 8.1.6 A Clinical Trial

```
1 #Page 468
2
3 library(ggplot2)
```

```

4
5 f_t_given_theta <- function(t, theta) {
6   return(choose(40, 40 * t) * theta^(40 * t) * (1 -
7     theta)^(40 * (1 - t)))
8 }
9 probability_T_theta <- function(theta) {
10   sum(sapply(seq(0, 1, by = 1/40), function(t) {
11     if (abs(t - theta) < 0.1) {
12       return(f_t_given_theta(t, theta))
13     } else {
14       return(0)
15     }
16   })))
17 }
18
19 theta_values <- seq(0, 1, by = 0.01)
20
21 prob_values <- sapply(theta_values, probability_T_
22   theta)
23 data <- data.frame(theta = theta_values, probability
24   = prob_values)
25 ggplot(data, aes(x = theta, y = probability)) +
26   geom_line(color = "blue") +
27   labs(title = expression(Pr("|T - " * theta * " | <
28     0.1 | " * theta)),
29     x = expression(theta),
30     y = "Probability") +
31   theme_minimal()
32 theta_values <- seq(0, 1, by = 1/40)
33
34 prob_values <- sapply(theta_values, probability_T_
35   theta)
36 cat("Minimum value", round(min(prob_values), 4), "\n")

```

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

R code Exa 8.2.3 Acid Concentration in Cheese I

```
1 #Page 471
2
3 library(stats)
4
5 prob <- pchisq(10, df = 10)
6 round(prob, 2)
```

R code Exa 8.3.2 Rain from Seeded Clouds I

```
1 #Page 474
2
3 df <- 25
4
5 critical_value <- qchisq(0.25, df)
6
7 prob_underestimate <- pchisq(critical_value, df,
    lower.tail = TRUE)
8
9 prob_underestimate
```

R code Exa 8.4.3 Rain from Seeded Clouds II

```
1 #Page 482
2
3 n <- 26
```

```

4
5 quantile <- 1.316 / sqrt(n)
6
7 prob <- 0.9
8
9 mu <- 0
10 sigma <- 1
11
12 upper_limit <- mu + quantile * sigma
13
14 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

```

1 #Page 486
2
3 n <- 26
4
5 gamma <- 0.95
6
7 alpha <- (1 - gamma) / 2
8 quantile <- qt(1 - alpha, df = n - 1)
9
10 const <- quantile / sqrt(n)
11
12 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

```

1 #Page 487
2
3 n <- 26
4
5 x_bar <- 5.134
6 sigma <- 1.600
7 gamma <- 0.95
8
9 alpha <- (1 - gamma) / 2
10 quantile <- qt(1 - alpha, df = n - 1)
11 const <- quantile / sqrt(n)
12
13 a <- x_bar - const * sigma
14 b <- x_bar + const * sigma
15
16 cat("The observed value of x_bar is", x_bar, "\n")
17 cat("The observed value of sigma is", sigma, "\n")
18 cat("The observed value of a is", round(a,3), "\n")
19 cat("The observed value of b is", round(b,2), "\n")

```

R code Exa 8.5.4 Acid Concentration in Cheese II

```

1 #Page 487
2
3 n <- 10
4
5 gamma <- 0.9
6
7 alpha <- (1 - gamma) / 2
8 quantile <- qt(1 - alpha, df = n - 1)
9
10 lactic_acid <- c(0.86, 1.53, 1.57, 1.81, 0.99, 1.09,
11                 1.29, 1.78, 1.29, 1.58)
12
13 x_bar <- mean(lactic_acid)

```

```

13 sigma <- sd(lactic_acid)
14
15 lower_bound <- x_bar - quantile * sigma / sqrt(n)
16 upper_bound <- x_bar + quantile * sigma / sqrt(n)
17
18 cat("The observed value of x_bar is", round(x_bar,4)
    , "\n")
19 cat("The observed value of sigma is", round(sigma,4)
    , "\n")
20 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

```

1 #Page 489
2
3 n <- 26
4
5 gamma <- 0.9
6
7 quantile <- qt(gamma, df = n - 1)
8
9 x_bar <- 5.134
10 sigma <- 1.600
11
12 lower_limit <- x_bar - quantile * sigma / sqrt(n)
13
14 cat("The observed value of x_bar is", x_bar, "\n")
15 cat("The observed value of sigma is", sigma, "\n")
16 cat("The 90% lower confidence limit for mu is",
    round(lower_limit,3), "\n")
17 # The answer provided in the textbook is wrong.

```

R code Exa 8.5.7 Lifetimes of Electronic Components II

```
1 #Page 489
2
3 n <- 3
4
5 gamma <- 0.98
6
7 quantile <- qgamma(gamma, shape = n, scale = 1)
8
9 round(quantile,4)
```

R code Exa 8.5.10 Approximate Confidence Interval for Poisson Mean

```
1 # Page 490
2
3 c <- 0.196
4 n <- 100
5
6 alpha <- 2 * pnorm(c * sqrt(n)) - 1
7
8 round(alpha,2)
```

R code Exa 8.5.11 Uniforms on an Interval of Length One

```
1 #Page 492
2
3 z <- 0.1
4 c <- 0.3
5
6 conditional_prob <- ifelse(c <= (1 - z) / 2, 2 * c /
7   (1 - z), 1)
```

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

R code Exa 8.6.2 Acid Concentration in Cheese III

```
1 #Page 498
2
3 mu0 <- 1
4 lambda0 <- 1
5 alpha0 <- 0.5
6 beta0 <- 0.5
7
8 n <- 10
9 x_bar <- 1.379
10 s2_n <- 0.9663
11
12 mu1 <- (mu0 * lambda0 + n * x_bar) / (lambda0 + n)
13 lambda1 <- lambda0 + n
14 alpha1 <- alpha0 + n / 2
15 beta1 <- beta0 + (1 / 2) * (s2_n + (n * lambda0 * (x
    _bar - mu0)^2) / (lambda0 + n))
16
17 cat("Posterior hyperparameters:\n")
18 cat("mu1 =", round(mu1,3), "\n")
19 cat("lambda1 =", lambda1, "\n")
20 cat("alpha1 =", alpha1, "\n")
21 cat("beta1 =", round(beta1,4), "\n")
22
23 sigma_threshold <- 0.3
24 tau_threshold <- 1 / sigma_threshold^2
25 posterior_prob <- pgamma(tau_threshold, shape =
    alpha1, rate = beta1, lower.tail = TRUE)
26
27 cat("\nPosterior probability that sigma > 0.3 is:",
    round(posterior_prob,3))
```

R code Exa 8.6.3 Nursing Homes in New Mexico

```
1 #Page 500
2
3 x <- c(128, 281, 291, 238, 155, 148, 154, 232, 316,
        96, 146, 151, 100, 213, 208, 157, 48, 217)
4 n <- length(x)
5 xn <- mean(x)
6 sn2 <- var(x)
7
8 mu0 <- 200
9 lambda0 <- 2
10 alpha0 <- 2
11 beta0 <- 6300
12
13 mu1 <- (lambda0 * mu0 + n * xn) / (lambda0 + n)
14 lambda1 <- lambda0 + n
15 alpha1 <- alpha0 + n / 2
16 beta1 <- beta0 + 0.5 * sum((x - xn)^2) + (lambda0 *
    n * (xn - mu0)^2) / (2 * (lambda0 + n))
17
18 mu_posterior <- mu1
19 mu_posterior
20
21 var_mu_posterior <- beta1 / (lambda1 * (alpha1 - 1))
22 round(var_mu_posterior,2)
23
24 tau_posterior <- alpha1 / beta1
25 signif(tau_posterior,3)
26
27 var_tau_posterior <- alpha1 / (beta1^2)
28 signif(var_tau_posterior,3)
```

R code Exa 8.6.4 An Improper Prior for Seeded Cloud Rainfall

```
1 #Page 503
2
3 mu_hat <- 5.134
4 quantile_t <- 2.060
5 scaling_factor <- 3.188
6
7 a <- mu_hat - quantile_t / scaling_factor
8 b <- mu_hat + quantile_t / scaling_factor
9
10 interval <- c(a, b)
11 round(interval,3)
12
13 mu_0 <- 4
14 df <- 25
15
16 U_value <- scaling_factor * (mu_0 - mu_hat)
17
18 probability <- 1 - pt(U_value, df)
19 round(probability,4)
```

R code Exa 8.7.3 Lifetimes of Electronic Components III

```
1 # Page 507
2
3 theta_values <- seq(0, 2, by = 0.1)
4
5 mse_bayes <- function(theta) {
6   if (theta <= 3.1) {
7     return(4 / (2 + 1) * (theta^2 / (2 + 1)))
8   } else {
```

```

9      return((4 / (2 + 1))^2 + theta^2 / (2 + 1) * (2
      / (2 + 1))^2)
10  }
11 }
12
13 mse_mle <- function(theta) {
14   return((9 * theta^2 / 4) + (theta^2 / 4))
15 }
16
17 mse_unbiased <- function(theta) {
18   return(theta^2)
19 }
20
21 mse_1_over_t <- function(theta) {
22   return(theta^2 / 4)
23 }
24
25 mse_bayes_values <- sapply(theta_values, mse_bayes)
26 mse_mle_values <- mse_mle(theta_values)
27 mse_unbiased_values <- mse_unbiased(theta_values)
28 mse_1_over_t_values <- mse_1_over_t(theta_values)
29
30 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")
31 lines(theta_values, mse_mle_values, col = "red", lwd
      = 2)
32 lines(theta_values, mse_unbiased_values, col = "
      green", lwd = 2)
33 lines(theta_values, mse_1_over_t_values, col = "blue
      ", lwd = 2)
34 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

```
1 #Page 524
2
3 library(ggplot2)
4
5 set.seed(42)
6
7 n <- 40
8 sigma_hat <- 1.061
9
10 posterior_mean <- sigma_hat
11 posterior_variance <- sigma_hat^2 / (2 * n)
12
13 cat("Posterior mean:", posterior_mean, "\n")
14 cat("Posterior variance:", round(posterior_variance
15     ,4), "\n")
16
17 true_sigma <- 1
18 sample_data <- rnorm(n, mean = 0, sd = true_sigma)
19
20 sigma_hat <- sqrt(sum(sample_data^2) / n)
21
22 posterior_mean <- sigma_hat
23 posterior_variance <- sigma_hat^2 / (2 * n)
24
25 actual_posterior_pdf <- function(sigma) {
26   if (sigma > 0) {
27     posterior_value <- (sigma^(-n-1)) * exp(-sum(
28       sample_data^2) / (2 * sigma^2))
29     return(posterior_value)
30   } else {
31     return(0)
32   }
33 }
```

```

31   }
32 }
33
34 approx_posterior_pdf <- function(sigma) {
35   dnorm(sigma, mean = posterior_mean, sd = sqrt(
36     posterior_variance))
37 }
38 sigma_values <- seq(0.5, 2, by = 0.01)
39
40 actual_posterior_values <- sapply(sigma_values,
41   actual_posterior_pdf)
42 approx_posterior_values <- sapply(sigma_values,
43   approx_posterior_pdf)
44
45 actual_posterior_values <- actual_posterior_values /
46   sum(actual_posterior_values) / (sigma_values[2]
47   - sigma_values[1])
48
49 plot_data <- data.frame(
50   sigma = sigma_values,
51   actual_posterior = actual_posterior_values,
52   approx_posterior = approx_posterior_values
53 )
54
55 ggplot(plot_data, aes(x = sigma)) +
56   geom_line(aes(y = actual_posterior, color = "
57     Actual Posterior")) +
58   geom_line(aes(y = approx_posterior, color = "
59     Approximate Posterior")) +
60   labs(title = "Posterior and Approximate Posterior
61     p.d.f.",
62     x = "Sigma", y = "Density") +
63   scale_color_manual(values = c("Actual Posterior" =
64     "blue", "Approximate Posterior" = "red")) +
65   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

```
1 #Page 534
2
3 mu0 <- 4
4 n <- 15
5 sigma2 <- 9
6 c_values <- c(1, 2, 3)
7
8 power_function <- function(mu, c) {
9   sigma <- sqrt(sigma2)
10  z1 <- (mu0 + c - mu) / (sigma / sqrt(n))
11  z2 <- (mu0 - c - mu) / (sigma / sqrt(n))
12  1 - pnorm(z1) + pnorm(z2)
13 }
14
15 mu_range <- seq(0, 8, length.out = 100)
16
17 plot(mu_range, power_function(mu_range, c_values[1])
18      , type = "l", lwd = 2,
19      xlab = "mu", ylab = "Power", main = "Power
      Functions",
```

```

19     ylim = c(0, 1))
20
21   for (i in 2:length(c_values)) {
22     lines(mu_range, power_function(mu_range, c_values[
23       i]), col = i, lwd = 2)
24   }
25   legend("bottomright", legend = paste("c =", c_values
26     ), col = 1:length(c_values), lwd = 2)

```

R code Exa 9.1.7 Testing Hypotheses about a Uniform Distribution

```

1 #Page 536
2
3 n <- 68
4 theta <- 29 / 30
5
6 alpha_delta <- theta^n
7
8 cat("The size of      is:", round(alpha_delta,4), "\n"
9   )

```

R code Exa 9.1.9 Testing Hypotheses about a Bernoulli Parameter I

```

1 #Page 538
2
3 n <- 10
4 p0 <- 0.3
5 alpha0 <- 0.1
6
7 cumulative_prob <- function(c, n, p0) {
8   sum(dbinom(c:n, size = n, prob = p0))
9 }

```

```

10
11 prob_c6 <- cumulative_prob(6, n, p0)
12 prob_c5 <- cumulative_prob(5, n, p0)
13
14 cat("Cumulative probability for c = 6:", round(prob_
      c6,4), "\n")
15 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

```

1 #Page 538
2
3 n <- 10
4 alpha0 <- 0.05
5
6 z_critical <- qnorm(1 - alpha0/2)
7
8 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

```

1 #Page 539
2
3 Z_observed <- 2.78
4
5 p_value <- 2 * (1 - pnorm(abs(Z_observed)))
6
7 cat("P-Value:", round(p_value,4), "\n")
8
9 if (p_value < 0.01) {

```



```

10   print(" Reject H0")
11 } else {
12   print("Do not reject H0")
13 }

```

R code Exa 9.1.12 Testing Hypotheses about a Bernoulli Parameter II

```

1 #Page 540
2
3 Y_observed <- 6
4
5 p0 <- 0.3
6 n <- 10
7
8 probability <- sum(dbinom(Y_observed:n, size = n,
9   prob = p0))
10 cat(" Probability (p-value):", round(probability,4))

```

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

```

1 #Page 542
2
3 n <- 10
4 alpha0 <- 0.1
5
6 q_func <- function(y) {
7   if (y == 0) {
8     return (0)
9   }
10  p0 <- 0
11  while (sum(dbinom(y:n, n, p0)) <= alpha0) {

```

```

12     p0 <- p0 + 0.0001
13   }
14   return(p0 - 0.0001)
15 }
16
17 y_values <- 0:n
18 q_values <- sapply(y_values, q_func)
19
20 cat("n =", n, ", alpha0 =", alpha0, "\n")
21 cat("y\tq(y)\n")
22 for (i in seq_along(y_values)) {
23   cat(y_values[i], "\t", q_values[i], "\n")
24 }

```

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

```

1 #Page 544
2
3 y <- 0:10
4 n <- 10
5 theta0 <- 0.3
6
7 phi_y <- ((n * theta0 / y)^y) * ((n*(1 - theta0)/(n
  - y))^(n-y))
8 pr_y <- dbinom(y, size = 10, prob = 0.3)
9
10 table_df <- data.frame(y = y, phi_y = signif(phi_y
  ,3), pr_y = signif(pr_y,3))
11
12 print(table_df)
13 # 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

```
1 #Page 545
2
3 alpha0 <- 0.05
4 chi_sq_quantile <- qchisq(1 - alpha0, df = 1)
5 round(chi_sq_quantile,3)
```

R code Exa 9.2.1 Service Times in a Queue I

```
1 #Page 550
2
3 f1 <- function(x, n) {
4   if (all(x > 0)) {
5     return(2 * factorial(n) / ((2 + sum(x))^(n + 1)))
6   } else {
7     return(0)
8   }
9 }
10
11 f0 <- function(x, n) {
12   if (all(x > 0)) {
13     return(1/(2^n) * exp(-sum(x)/2))
14   } else {
15     return(0)
16   }
17 }
18
19 x <- seq(0.001, 10, length.out = 100)
20
```

```

21 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"
    )
22 lines(x, sapply(x, f0, n = 1), col = "red")
23 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

```

1 #Page 551
2
3 alpha <- exp(-0.5 * 4)
4 cat("Type I error probability (alpha):", round(alpha
    ,3), "\n")
5
6 beta <- 1 - 2/(2 + 4)
7 cat("Type II error probability (beta):", round(beta
    ,3), "\n")

```

R code Exa 9.2.3 Service Times in a Queue III

```

1 #Page 553
2
3 threshold <- 5.025725
4
5 alpha_star <- exp(-0.5 * threshold)
6 cat("Type I error probability (alpha*):", round(
    alpha_star,3), "\n")
7
8 beta_star <- 1 - 2/(2 + threshold)
9 cat("Type II error probability (beta*):", round(beta
    _star,3), "\n")

```

R code Exa 9.2.4 Service Times in a Queue IV

```
1 #Page 554
2
3 alpha <- 0.07
4
5 c <- -2 * log(alpha)
6 cat("Threshold value for alpha = 0.07:", round(c,3),
    "\n")
7
8 beta <- 1 - 2/(2 + c)
9 cat("Type II error probability (beta):", round(beta
    ,3), "\n")
```

R code Exa 9.2.5 Random Sample from a Normal Distribution

```
1 #Page 554
2
3 n <- 9
4
5 beta_n <- pnorm(1.645 - sqrt(n))
6 cat("beta:", round(beta_n,4), "\n")
7
8 c <- 0.577
9
10 alpha <- 1 - pnorm(c, mean = 0, sd = 1/sqrt(n))
11 cat("alpha:", round(alpha,4), "\n")
12
13 beta <- pnorm(c, mean = 1, sd = 1/sqrt(n))
14 cat("beta:", round(beta,4), "\n")
15
```

```

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

```

R code Exa 9.2.6 Sampling from a Bernoulli Distribution

```

1 #Page 555
2
3 n <- 10
4 p0 <- 0.2
5 alpha0 <- 0.05
6
7 prob_Y_gt_4 <- 1 - pbinom(4, size = n, prob = p0)
8
9 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

```

1 #Page 563
2
3 n <- 10
4 alpha0 <- 0.1
5
6 df <- 2 * n
7
8 quantile <- qchisq(alpha0, df = df, lower.tail =
    TRUE)
9
10 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

```
1 #Page 563
2
3 n <- 20
4 p <- 0.1
5 c_values <- c(6, 7)
6
7 sizes <- sapply(c_values, function(c_val) {
8   alpha <- 1 - pbinom(c_val - 1, size = n, prob = p)
9   return(alpha)
10 })
11
12 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

```
1 #Page 569
2
3 mu0 <- 140
4 sigma <- 26
5 n <- 30
6 alpha0 <- 0.05
7 Xn <- 131.37
8
9 c1 <- mu0 - qnorm(1 - alpha0/2) * sqrt(sigma) * n
   ^(-0.5)
10 c2 <- mu0 + qnorm(1 - alpha0/2) * sqrt(sigma) * n
   ^(-0.5)
11
```

```

12 if (Xn <= c1 || Xn >= c2) {
13   cat("Reject H0:      = 140\n")
14 } else {
15   cat("Do not reject H0:      = 140\n")
16 }
17
18 cat("Critical values:\n")
19 cat("c1 =", round(c1,2), "\n")
20 cat("c2 =", round(c2,2), "\n")

```

R code Exa 9.4.4 Service Times in a Queue VI

```

1 #Page 570
2
3 library(nleqslv)
4
5 n <- 3
6 alpha0 <- 0.05
7
8 solve_c1_c2 <- function(c) {
9   c1 <- c[1]
10  c2 <- c[2]
11
12  eq1 <- ((c1 / (2 * n))^n) * exp(n - c1 / 2) - ((c2
13    / (2 * n))^n) * exp(n - c2 / 2)
14
15  eq2 <- pgamma(c1, shape = n, rate = 1/2) + 1 -
16    pgamma(c2, shape = n, rate = 1/2) - alpha0
17
18  return(c(eq1, eq2))
19 }
20
21 initial_guess <- c(1,15)
22
23 solution <- nleqslv(initial_guess, solve_c1_c2)

```



```

22
23 c1 <- solution$x[1]
24 c2 <- solution$x[2]
25
26 cat("c1 =", c1, "\n")
27 cat("c2 =", c2, "\n")

```

R code Exa 9.4.5 Testing an Interval Null Hypothesis

```

1 #Page 571
2
3 target_function <- function(c) {
4   pnorm(0.4 + 4 * c) - pnorm(0.4 - 4 * c) - 0.95
5 }
6
7 solution <- uniroot(target_function, interval = c(0,
8   1))
9
10 c <- solution$root
11
12 c1 <- 10 - c
13 c2 <- 10 + c
14
15 cat("c =", round(c,3), "\n")
16 cat("c1 =", round(c1,3), "\n")
17 cat("c2 =", round(c2,3), "\n")

```

R code Exa 9.5.3 Nursing Homes in New Mexico

```

1 #Page 578
2
3 x <- c(128, 281, 291, 238, 155, 148, 154, 232, 316,
4   96, 146, 151, 100, 213, 208, 157, 48, 217)

```

```

4
5 sample_mean <- mean(x)
6
7 sample_sd <- sd(x)
8
9 df <- length(x) - 1
10
11 mu0 <- 200
12
13 U <- sqrt(df) * (sample_mean - mu0) / sample_sd
14
15 cat("Observed value of U:", U, "\n")
16
17 critical_value <- qt(0.1, df = df)
18
19 cat("Critical value from t-distribution:", critical_
    value, "\n")
20
21 if (U <= critical_value) {
22   cat("Reject H0\n")
23 } else {
24   cat("Fail to reject H0\n")
25 }

```

R code Exa 9.5.4 Lengths of Fibers I

```

1 #Page 578
2
3 sample_mean <- 5.4
4 sample_sd <- 0.4226
5 n <- 15
6 mu0 <- 5.2
7 alpha <- 0.05
8
9 U <- (sample_mean - mu0) / (sample_sd / sqrt(n))

```

```

10
11 cat(" Observed value of U:", round(U,3), "\n")
12
13 critical_value <- qt(1 - alpha, df = n - 1)
14
15 cat(" Critical value from t-distribution:", round(
      critical_value,3), "\n")
16
17 if (U > critical_value) {
18   cat(" Reject H0\n")
19 } else {
20   cat(" Fail to reject H0\n")
21 }
22
23 p_value <- 1 - pt(U, df = n - 1)
24
25 cat(" p-value:", round(p_value,4), "\n")

```

R code Exa 9.5.5 Lengths of Fibers II

```

1 #Page 579
2
3 mu0 <- 5.2
4 sigma <- 0.4226
5 n <- 15
6
7 psi <- sqrt(n) * ((mu0 + sigma/2 - mu0) / sigma)
8 round(psi,3)

```

R code Exa 9.5.6 Lengths of Fibers III

```

1 #Page 580
2

```

```

3 library(sadists)
4
5 power_func <- function(psi, df, alpha) {
6   q_alpha <- qt(1 - alpha, df = df, ncp = 0)
7   1 - pt(q_alpha, df = df, ncp = psi)
8 }
9
10 mu0 <- 5.2
11 sigma <- 0.4226
12 mu_alternative <- mu0 + sigma / 2
13 alpha <- 0.05
14 desired_power <- 0.8
15
16 calculate_psi <- function(n, mu_alternative, mu0,
17   sigma) {
18   sqrt(n) * (mu_alternative - mu0) / sigma
19 }
20 n <- 15
21 current_power <- 0
22 while (current_power < desired_power) {
23   n <- n + 1
24   psi <- calculate_psi(n, mu_alternative, mu0, sigma
25   )
26   current_power <- power_func(psi, n - 1, alpha)
27 }
28 cat("Required sample size:", n, "\n")
29 cat("Achieved power:", round(current_power, 4), "\n")

```

R code Exa 9.5.7 Crash Test Dummies

```

1 #Page 581
2
3 n <- 164

```

```

4 alpha <- 0.01
5 x_bar <- 0.2199
6 sigma <- 0.5342
7 mu0 <- 0
8
9 U <- sqrt(n) * (x_bar - mu0) / sigma
10
11 t_critical <- qt(1 - alpha, df = n - 1)
12
13 cat("Test statistic U:", round(U,4), "\n")
14 cat("Critical value:", round(t_critical,2), "\n")
15
16 mu_alternative <- sigma / 4
17 psi <- sqrt(n) * (mu_alternative - mu0) / sigma
18
19 power <- 1 - pt(t_critical, df = n - 1, ncp = psi)
20
21 cat("Noncentrality parameter:", signif(psi,3), "\n")
22 cat("Power of the test:", round(power,3), "\n")

```

R code Exa 9.5.9 Egyptian Skulls II

```

1 #Page 582
2
3 n <- 30
4 alpha <- 0.05
5 mu0 <- 140
6 x_bar <- 131.37
7 sigma <- 5.129
8
9 U <- sqrt(n) * (x_bar - mu0) / sigma
10
11 t_critical <- round(qt(1 - alpha / 2, df = n - 1),3)
12
13 cat("Test statistic U:", round(U,4), "\n")

```

```

14 cat(" Critical value (c1, c2):", -t_critical,"and", t
    _critical, "\n")
15
16 if (U < -t_critical || U > t_critical) {
17   cat(" Reject H0 at level", alpha, "\n")
18 } else {
19   cat("Do not reject H0 at level", alpha, "\n")
20 }
21 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 9.5.10 Lengths of Fibers IV

```

1 #Page 582
2
3 n <- 15
4 alpha <- 0.05
5 df <- n - 1
6 U <- 1.833
7
8 t_critical <- round(qt(1 - alpha / 2, df),3)
9
10 cat("Test statistic U:", U, "\n")
11 cat(" Critical values (c1, c2):", -t_critical, t_
    critical, "\n")
12
13 if (U <= -t_critical || U >= t_critical) {
14   cat(" Reject H0 at level", alpha, "\n")
15 } else {
16   cat("Do not reject H0 at level", alpha, "\n")
17 }

```

R code Exa 9.5.11 Lengths of Fibers V

```

1 #Page 583
2
3 df <- 14
4 U <- 1.833
5
6 p_value <- 2 * (1 - pt(U, df))
7
8 cat("The p-value is:", round(p_value,4), "\n")

```

R code Exa 9.6.2 Rain from Seeded Clouds I

```

1 #Page 589
2
3 Xm <- 5.13
4 Ym <- 3.99
5 SX2 <- 63.96
6 SY2 <- 67.39
7 n <- 26
8 alpha <- 0.01
9 df = 50
10
11 critical_value <- qt(1-alpha, df = df)
12 test_statistic <- (sqrt(df) * (Xm - Ym)) / (sqrt(1/
    26 + 1/26) * sqrt((SX2 + SY2)))
13
14 cat("Critical value:", round(critical_value,3), "\n"
    )
15 cat("Test statistic:", round(test_statistic,2), "\n"
    )

```

R code Exa 9.6.3 Roman Pottery in Britain I

```

1 #Page 590

```

```

2
3 m <- 14
4 n <- 5
5 Xm <- 12.56
6 Ym <- 17.32
7 SX2 <- 24.65
8 SY2 <- 11.01
9 alpha <- 0.005
10
11 U <- (sqrt(m+n-2)*(Xm - Ym))/(sqrt(1/m +1/n)*sqrt(
      SX2+SY2))
12
13 df <- m + n - 2
14
15 critical_value <- qt(1 - alpha, df)
16
17 p_value <- pt(U, df)
18
19 cat("Test statistic U:", round(U,3), "\n")
20 cat("Critical value:", round(critical_value,3), "\n"
    )
21 cat("p-value:", signif(p_value,1), "\n")
22 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 9.6.4 Roman Pottery in Britain II

```

1 #Page 591
2
3 m <- 14
4 n <- 5
5 alpha <- 0.01
6 psi <- 2.88
7 df <- m + n - 2
8

```



```

9 critical_value <- 1.5/sqrt(1/m +1/n)
10
11 power <- 1 - pt(qt(1 - alpha, df = df), df = df, ncp
    = psi)
12
13 cat("Critical value:", round(critical_value,2), "\n"
    )
14 cat("Power of the test:", round(power,2), "\n")

```

R code Exa 9.6.5 Comparing Copper Ores I

```

1 #Page 591
2
3 m <- 8
4 n <- 10
5 X_bar <- 2.6
6 Y_bar <- 2.3
7 S2_X <- 0.32
8 S2_Y <- 0.22
9 alpha <- 0.01
10 df <- m + n - 2
11
12 pooled_variance <- ((m - 1) * S2_X + (n - 1) * S2_Y)
    / df
13
14 U <- (sqrt(m+n-2)*(X_bar - Y_bar))/(sqrt(1/m +1/n)*
    sqrt(S2_X+S2_Y))
15
16 critical_value <- qt(1 - alpha / 2, df)
17
18 cat("Test Statistic (U):", round(U,3), "\n")
19 cat("Critical Value:", round(critical_value,3), "\n"
    )
20
21 if (abs(U) > critical_value) {

```

```

22   cat("Reject the null hypothesis at level", alpha,
      "\n")
23 } else {
24   cat("Do not reject the null hypothesis at level",
      alpha, "\n")
25 }
26 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 9.6.6 Comparing Copper Ores II

```

1  #Page 594
2
3  x_bar <- 2.6
4  y_bar <- 2.3
5  s2_x <- 0.32
6  s2_y <- 0.22
7  m <- 8
8  n <- 10
9
10 numerator <- x_bar - y_bar
11 denominator <- sqrt((s2_x / (m * (m - 1))) + (s2_y /
      (n * (n - 1))))
12 V <- numerator / denominator
13
14 nu <- ((s2_x / (m * (m - 1))) + (s2_y / (n * (n - 1)
      )))^2 /
15   ((1 / (m - 1)^3) * (s2_x / m)^2 + (1 / (n - 1)^3)
      * (s2_y / n)^2)
16
17 p_value <- 2 * (1 - pt(V, df = nu))
18
19 cat("Test Statistic V:", round(V,3), "\n")
20 cat("Degrees of Freedom nu:", round(nu,2), "\n")
21 cat("P-value:", signif(p_value,2), "\n")

```

R code Exa 9.6.7 Comparing Copper Ores III

```
1 #Page 595
2
3 x_bar <- 2.6
4 s2_x <- 0.32
5 y_bar <- 2.3
6 s2_y <- 0.22
7 m <- 8
8 n <- 10
9
10 mu_0 <- (m * x_bar + n * y_bar) / (m + n)
11
12 tolerance <- 1e-6
13 max_iter <- 13
14 iter <- 0
15
16 update_values <- function(mu, x_bar, y_bar, s2_x, s2
    _y, m, n) {
17   sigma2_1 <- (1 / m) * (s2_x + m * (x_bar - mu)^2)
18   sigma2_2 <- (1 / n) * (s2_y + n * (y_bar - mu)^2)
19   new_mu <- (m * x_bar / sigma2_1 + n * y_bar /
    sigma2_2) / (m / sigma2_1 + n / sigma2_2)
20   return(list(mu = new_mu, sigma2_1 = sigma2_1,
    sigma2_2 = sigma2_2))
21 }
22
23 for (i in 1:13) {
24   result <- update_values(mu_0, x_bar, y_bar, s2_x,
    s2_y, m, n)
25   if (abs(result$mu - mu_0) < tolerance) {
26     cat("Convergence achieved after", iter, "
    iterations.\n")
27     break
```

```

28     }
29     mu_0 <- result$mu
30 }
31
32 mu_hat <- result$mu
33 sigma2_1_hat <- result$sigma2_1
34 sigma2_2_hat <- result$sigma2_2
35
36 Lambda_xy <- 0.01356
37 LR_statistic <- -2 * log(Lambda_xy)
38
39 p_value <- 1 - pchisq(LR_statistic, df = 1)
40
41 cat("Final MLEs:\n")
42 cat("mu_hat:", round(mu_hat,3), "\n")
43 cat("sigma2_1_hat:", round(sigma2_1_hat,4), "\n")
44 cat("sigma2_2_hat:", round(sigma2_2_hat,4), "\n")
45 cat("\nLikelihood Ratio Statistic:", round(LR_
      statistic,3), "\n")
46 cat("P-value:", round(p_value,3), "\n")

```

R code Exa 9.7.2 Quantile of an F Distribution

```

1 # Page 598
2
3 df1 <- 6
4 df2 <- 12
5
6 quantile_0.05 <- qf(0.05, df1, df2)
7
8 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

```
1 #Page 600
2
3 s2_x <- 30
4 s2_y <- 40
5
6 df1 <- 5
7 df2 <- 20
8
9 f_statistic <- s2_x / s2_y
10
11 f_crit1 <- qf(0.025, df1, df2)
12 f_crit2 <- qf(0.975, df1, df2)
13
14 cat("The critical values for the F-test are:", f_
      crit1, "and", f_crit2, "\n")
15
16 if (f_statistic < f_crit1 || f_statistic > f_crit2)
17 {
18   cat("Reject the null hypothesis that the variances
      are equal at alpha = 0.05.\n")
19 } else {
20   cat("Do not reject the null hypothesis that the
      variances are equal at alpha = 0.05.\n")
21 }
22 f_stat_alt <- 2.71 / 3
23 power <- 1 - pf(f_stat_alt, df1, df2)
24 cat("The power of the test for  $1^2 = 3 \cdot 2^2$  is:",
      round(power, 3), "\n")
```

R code Exa 9.7.4 Rain from Seeded Clouds II

```
1 #Page 601
2
3 s2_x <- 63.96
4 s2_y <- 67.39
5
6 df1 <- 25
7 df2 <- 25
8
9 f_statistic <- s2_x / s2_y
10
11 f_crit1 <- qf(0.025, df1, df2)
12 f_crit2 <- qf(0.975, df1, df2)
13
14 cat("The critical values for the F-test are:", round
      (f_crit1,4), "and", round(f_crit2,4), "\n")
15
16 if (f_statistic < f_crit1 || f_statistic > f_crit2)
17   {
18     cat("Reject the null hypothesis that the variances
19         are equal.\n")
20   } else {
21     cat("Do not reject the null hypothesis that the
22         variances are equal.\n")
23   }
24 }
```

R code Exa 9.8.4 Calorie Counts on Food Labels

```
1 #Page 610
2
3 mean_posterior <- 0.1154
4 var_posterior <- 4.62
5
6 d_values <- seq(0, 5, by = 0.01)
```

```

7
8 posterior_probs <- pnorm(d_values, mean_posterior,
  sqrt(var_posterior)) -
9   pnorm(-d_values, mean_posterior, sqrt(var_
    posterior))
10
11 plot(d_values, posterior_probs, type = "l", col = "
  blue",
12       xlab = "| |", ylab = "Posterior Probability",
13       main = "Posterior Probability that | | is at
    most d",
14       xlim = c(0, 5), ylim = c(0, 1))

```

R code Exa 9.8.5 Pesticide Residue on Celery

```

1 #Page 611
2
3 df <- 76
4 prob <- 1 - pt(1.974, df)
5
6 round(prob, 3)

```

R code Exa 9.8.6 Roman Pottery in Britain III

```

1 #Page 614
2
3 m <- 14
4 n <- 5
5 Xm <- 12.56
6 Ym <- 17.32
7 SX2 <- 24.65
8 SY2 <- 11.01
9

```

```

10 U <- (sqrt(m+n-2)*(Xm - Ym))/(sqrt(1/m +1/n)*sqrt(
      SX2+SY2))
11
12 df <- m + n - 2
13
14 p_value <- pt(U, df)
15 cat("p-value:", signif(p_value,1), "\n")
16
17 p_posterior <- p_value
18 cat("Pr( 1          2 |x) =", signif(p_posterior,1), "\n")

```

R code Exa 9.8.7 Roman Pottery in Britain IV

```

1 #Page 614
2
3 library(ggplot2)
4
5 T17 <- function(x) {
6   pt(x, df = 17)
7 }
8
9 d_values <- seq(0, 10, by = 0.01)
10
11 posterior_probs <- T17(1.33 * (d_values + 4.76)) -
      T17(1.33 * (-d_values + 4.76))
12
13 data <- data.frame(d = d_values, posterior_prob =
      posterior_probs)
14
15 median_value <- 4.76
16 median_y_value <- approx(d_values, posterior_probs,
      xout = median_value)$y
17
18 ggplot(data, aes(x = d, y = posterior_prob)) +

```



```

19 geom_line(color = "blue") +
20 geom_segment(aes(x = median_value, xend = median_
    value, y = 0, yend = median_y_value),
21               linetype = "dotted", color = "red") +
22 geom_segment(aes(x = 0, xend = median_value, y =
    median_y_value, yend = median_y_value),
23               linetype = "dotted", color = "red") +
24 labs(title = "Posterior Probability that  $|1 -$ 
    2 | is at most d",
25       x = "d",
26       y = "Posterior Probability") +
27 theme_minimal()

```

Chapter 10

Categorical Data and Nonparametric Methods

R code Exa 10.1.3 Blood Types I

```
1 #Page 627
2
3 observed_counts <- c(A = 2162, B = 738, AB = 228, O
  = 2876)
4
5 total <- sum(observed_counts)
6
7 theoretical_probabilities <- c(A = 1/3, B = 1/8, AB
  = 1/24, O = 1/2)
8
9 expected_counts <- round(total * theoretical_
  probabilities,1)
10
11 chi_square_statistic <- sum((observed_counts -
  expected_counts)^2 / expected_counts)
12
13 degrees_of_freedom <- length(observed_counts) - 1
14 p_value <- 1 - pchisq(chi_square_statistic, degrees_
  of_freedom)
```

```

15
16 cat("Chi-square test statistic:", round(chi_square_
      statistic, 3), "\n")
17 cat("p-value:", signif(p_value, digits = 3), "\n")

```

R code Exa 10.1.4 Montana Outlook Poll I

```

1 #Page 627
2
3 observed_counts <- c(Worse = 58, Same = 64, Better =
      67)
4
5 total <- sum(observed_counts)
6
7 theoretical_probabilities <- rep(1/3, length(
      observed_counts))
8
9 expected_counts <- total * theoretical_probabilities
10
11 chi_square_statistic <- sum((observed_counts -
      expected_counts)^2 / expected_counts)
12
13 degrees_of_freedom <- length(observed_counts) - 1
14 p_value <- 1 - pchisq(chi_square_statistic, degrees_
      of_freedom)
15
16 cat("Chi-square test statistic:", round(chi_square_
      statistic, 4), "\n")
17 cat("p-value:", round(p_value, 3), "\n")

```

R code Exa 10.1.5 Testing Hypotheses about a Proportion

```

1 #Page 627

```

```

2
3 N1 <- 16
4 N2 <- 84
5
6 n <- N1 + N2
7
8 p0_1 <- 0.10
9 p0_2 <- 0.90
10
11 expected_N1 <- n * p0_1
12 expected_N2 <- n * p0_2
13
14 Q <- sum((c(N1, N2) - c(expected_N1, expected_N2))^2
          / c(expected_N1, expected_N2))
15
16 p_value <- 1 - pchisq(Q, df = 1)
17
18 cat("Chi-square test statistic:", round(Q, 3), "\n")
19 cat("p-value:", round(p_value, 4), "\n")

```

R code Exa 10.1.6 Failure Times of Ball Bearings

```

1 #Page 629
2
3 mean_log <- log(50)
4 var_log <- 0.25
5
6 quantile_25 <- qnorm(0.25, mean = mean_log, sd =
          sqrt(var_log))
7 quantile_50 <- qnorm(0.5, mean = mean_log, sd = sqrt
          (var_log))
8 quantile_75 <- qnorm(0.75, mean = mean_log, sd =
          sqrt(var_log))
9
10 cat("25th quantile:", round(quantile_25, 3), "\n")

```

```

11 cat("50th quantile:", round(quantile_50, 3), "\n")
12 cat("75th quantile:", round(quantile_75, 3), "\n")
13
14 observed_counts <- c(3, 4, 8, 8)
15
16 n <- sum(observed_counts)
17
18 expected_probs <- rep(0.25, 4)
19
20 expected_counts <- n * expected_probs
21
22 Q <- sum((observed_counts - expected_counts)^2 /
           expected_counts)
23
24 cat("Chi-square test statistic:", round(Q, 3), "\n")

```

R code Exa 10.1.7 Blood Types II

```

1 #Page 630
2
3 observed_counts <- c(2162, 738, 228, 2876)
4
5 expected_counts <- c(2001.3, 750.5, 250.2, 3002.0)
6
7 test_statistic <- -2 * sum(observed_counts * log(
   expected_counts / observed_counts))
8
9 df <- length(observed_counts) - 1
10
11 p_value <- 1 - pchisq(test_statistic, df)
12
13 cat("Test statistic:", round(test_statistic, 2), "\n
   ")
14 cat("p-value:", signif(p_value, digits = 3), "\n")
15 # The answer may slightly vary due to rounding off

```

values.

R code Exa 10.2.5 Failure Times of Ball Bearings

```
1 #Page 638
2
3 observed_counts <- c(3, 4, 8, 8)
4 n <- sum(observed_counts)
5 mu_hat <- 4.150
6 sigma2_hat <- 0.2722
7 sigma_hat <- sqrt(sigma2_hat)
8
9 intervals <- c(3.575, 3.912, 4.249)
10
11 pi1 <- pnorm(intervals[1], mean = mu_hat, sd = sigma
12 _hat)
13 pi2 <- pnorm(intervals[2], mean = mu_hat, sd = sigma
14 _hat) - pi1
15 pi3 <- pnorm(intervals[3], mean = mu_hat, sd = sigma
16 _hat) - pi2 - pi1
17 pi4 <- 1 - pnorm(intervals[3], mean = mu_hat, sd =
18 sigma_hat)
19
20 pi <- round(c(pi1, pi2, pi3, pi4),4)
21 print(pi)
22
23 expected_counts <- n * pi
24
25 test_statistic <- sum((observed_counts - expected_
26 counts)^2 / expected_counts)
27
28 cat("Test statistic Q:", round(test_statistic, 3), "
29 \n")
30
31 # The answer may slightly vary due to rounding off
32 values.
```

R code Exa 10.2.6 Prussian Army Deaths

```
1 #Page 638
2
3 counts <- c(0, 1, 2, 3, 4)
4 observations <- c(144, 91, 32, 11, 2)
5 total_obs <- sum(observations)
6
7 theta_hat <- 196 / 280
8
9 pi_theta_hat <- round(dpois(counts, lambda = theta_
    hat),4)
10
11 Q <- sum((observations - total_obs * pi_theta_hat)^2
    / (total_obs * pi_theta_hat))
12
13 cat("Poisson probabilities at MLE 0.7:", pi_theta_
    hat, "\n")
14 cat("Q statistic:", round(Q, 3), "\n")
15 # The answer may slightly vary due to rounding off
    values.
16 # The answer provided in the textbook is wrong.
```

R code Exa 10.3.2 College Survey

```
1 #Page 644
2
3 observed <- matrix(c(24, 23, 12,
4                      24, 14, 10,
5                      17, 8, 13,
6                      27, 19, 9),
```

```

7             nrow = 4, byrow = TRUE)
8
9 row_totals <- c(59, 48, 38, 55)
10 col_totals <- c(92, 64, 44)
11 total <- sum(row_totals)
12
13 expected <- outer(row_totals, col_totals, "*") /
    total
14
15 Q <- sum((observed - expected)^2 / expected)
16
17 df <- (nrow(observed) - 1) * (ncol(observed) - 1)
18
19 cat("Q statistic:", round(Q, 2), "\n")
20 cat("Degrees of freedom:", df, "\n")

```

R code Exa 10.3.3 Montana Outlook Poll II

```

1 #Page 644
2
3 observed <- matrix(c(20, 15, 12,
4                     24, 27, 32,
5                     14, 22, 23),
6                   nrow = 3, byrow = TRUE)
7
8 expected <- matrix(c(14.42, 15.92, 16.66,
9                     25.47, 28.11, 29.42,
10                    18.11, 19.98, 20.92),
11                  nrow = 3, byrow = TRUE)
12
13 Q <- sum((observed - expected)^2 / expected)
14
15 df <- (nrow(observed) - 1) * (ncol(observed) - 1)
16
17 p_value <- 1 - pchisq(Q, df)

```



```

18
19 cat("Q statistic:", round(Q, 2), "\n")
20 cat("Degrees of freedom:", df, "\n")
21 cat("p-value:", round(p_value, 3), "\n")

```

R code Exa 10.4.3 A Clinical Trial

```

1 #Page 650
2
3 observed <- matrix(c(18, 22,
4                      13, 25,
5                      22, 16,
6                      24, 10),
7                    nrow = 4, byrow = TRUE)
8
9 total <- colSums(observed)
10 total_group <- rowSums(observed)
11 total_all <- sum(observed)
12
13 expected <- outer(total_group, total) / total_all
14
15 Q <- sum((observed - expected)^2 / expected)
16
17 df <- (nrow(observed) - 1) * (ncol(observed) - 1)
18
19 p_value <- 1 - pchisq(Q, df)
20
21 quantile <- pchisq(Q, df)
22
23
24 cat("Q statistic:", round(Q, 2), "\n")
25 cat("Degrees of freedom:", df, "\n")
26 cat("p-value:", round(p_value, 3), "\n")
27 cat("Quantile:", round(quantile, 3), "\n")

```

R code Exa 10.6.2 Testing Whether a Sample Comes from a Standard Normal Distribution

```
1 #Page 661
2
3 D_n_star <- 0.3372 - 0.16
4
5 n <- 25
6 ks_statistic <- sqrt(n) * D_n_star
7
8 H_value <- 0.6
9 tail_area <- 1 - H_value
10
11 print(paste("D*_n:", round(D_n_star, 4)))
12 print(paste("KS Statistic (sqrt(n) * D*_n):", round(
    ks_statistic, 4)))
13 print(paste("Tail Area (1 - H):", round(tail_area,
    4)))
```

R code Exa 10.6.4 Calcium Supplements and Blood Pressure I

```
1 #Page 664
2
3 t <- c(0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60,
    0.65, 0.70, 0.75,
4         0.80, 0.85, 0.90, 0.95, 1.00, 1.05, 1.10,
    1.15)
5 H_t <- c(0.0000, 0.0003, 0.0028, 0.0126, 0.0361,
    0.0772, 0.1357, 0.2080,
6         0.2888, 0.3728, 0.4559, 0.5347, 0.6073,
    0.6725, 0.7300, 0.7798,
7         0.8223, 0.8580)
```

```

8
9 Dmn <- 0.409
10 test_statistic <- sqrt(110/21) * Dmn
11 alpha0 <- 0.346
12 H <- qnorm(1 - alpha0)
13
14 cat("Test statistic:", round(test_statistic,3), "\n"
      )
15
16 index <- max(which(t <= test_statistic))
17 t1 <- t[index]
18 t2 <- t[index + 1]
19 H_t1 <- H_t[index]
20 H_t2 <- H_t[index + 1]
21
22 H <- H_t1 + (H_t2 - H_t1) * (test_statistic - t1) /
      (t2 - t1)
23
24 cat("H(0.936):", round(H,3), "\n")

```

R code Exa 10.7.3 Rain from Seeded Clouds

```

1 #Page 673
2
3 library(MASS)
4
5 log_rainfalls <- c(1.411, 1.685, 2.398, 3.123,
6                   3.172, 3.515, 4.912, 5.123, 5.285, 5.345,
7                   5.378, 5.396, 5.412, 5.467,
8                   5.478, 5.512, 5.789, 6.123,
9                   6.478, 6.789,
10                  6.912, 7.123, 7.512, 7.789,
11                  8.123, 8.456)
12
13 data <- data.frame(y = log_rainfalls)

```

```

10
11 fit <- rlm(y ~ 1, data = data, psi = psi.huber, k =
    2.5)
12
13 theta_hat <- coef(fit)
14
15 round(theta_hat,3)
16 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 10.8.2 Calorie Counts in Hotdogs

```

1 #Page 679
2
3 calorie_counts <- c(186, 181, 176, 149, 184, 190,
    158, 139, 175, 148, 152, 111, 141, 153, 190, 157,
    131, 149, 135, 132)
4
5 hypothesized_median <- 150
6
7 W <- sum(calorie_counts < hypothesized_median)
8
9 n <- length(calorie_counts)
10
11 p_value <- 2 * min(pbinom(W, n, 0.5), pbinom(W - 1,
    n, 0.5, lower.tail = FALSE))
12
13 cat("p-value:", round(p_value,4), "\n")

```

R code Exa 10.8.4 Comparing Copper Ores

```

1 #Page 681
2

```

```

3 S <- 104
4 mu <- 76
5 sigma <- sqrt(126.67)
6
7 Z <- (S - mu) / sigma
8
9 p_value <- 2 * (1 - pnorm(abs(Z)))
10
11 cat("Z-score:", round(Z,2), "\n")
12 cat("p-value:", signif(p_value,3), "\n")
13 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 10.8.5 Calcium Supplements and Blood Pressure II

```

1 #Page 682
2
3 calculate_test <- function(S, m, n) {
4   N <- m + n
5   mu <- m * (N + 1) / 2
6   sigma <- sqrt((m * n * (N + 1)) / 12)
7   Z <- (S - mu) / sigma
8   p_value <- 2 * (1 - pnorm(abs(Z)))
9   return(list(p_value = round(p_value,2)))
10 }
11
12 m <- 10
13 n <- 11
14
15 S1 <- 123
16 result1 <- calculate_test(S1, m, n)
17
18 S2 <- 126
19 result2 <- calculate_test(S2, m, n)
20

```

```
21 cat("p-value:", result1$p_value, "\n\n")
22
23 cat("p-value:", result2$p_value, "\n")
```

Chapter 11

Linear Statistical Models

R code Exa 11.1.1 Blood Pressure I

```
1 #Page 689
2
3 xi <- c(1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 4.6, 1.6,
4         5.5, 3.4)
5
6 yi <- c(0.7, -1.0, -0.2, -1.2, -0.1, 3.4, 0.0, 0.8,
7         3.7, 2.0)
8
9 plot(xi, yi,
10      xlim = range(c(xi, yi)), ylim = range(c(xi, yi))
11      ),
12      xlab = "x", ylab = "y",
13      main = "Reactions to Two Drugs",
14      pch = 16, col = "blue")
15
16 text(xi, yi, labels = 1:10, pos = 4, offset = 0.5,
17      cex = 0.8)
```

R code Exa 11.1.2 Blood Pressure II

```

1 #Page 691
2
3 xi <- c(1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 4.6, 1.6,
         5.5, 3.4)
4 yi <- c(0.7, -1.0, -0.2, -1.2, -0.1, 3.4, 0.0, 0.8,
         3.7, 2.0)
5
6 model <- lm(yi ~ xi)
7
8 plot(xi, yi,
9       xlim = range(c(xi, yi)), ylim = range(c(xi, yi))
10      ,
11      xlab = "Reaction to Drug A (x)", ylab = "
          Reaction to Drug B (y)",
12      main = "Reactions to Two Drugs with Fitted Line
          ",
13      pch = 16, col = "blue")
14
15 abline(model, col = "red", lwd = 2)
16
17 for(i in 1:length(xi)) {
18   segments(xi[i], yi[i], xi[i], predict(model,
19     newdata = data.frame(xi = xi[i])),
20     col = "green", lty = 2)
21 }
22
23 text(xi, yi, labels = 1:10, pos = 4, offset = 0.5,
24       cex = 0.8)
25
26 legend("topleft",
27       legend = c("Observed values", "Fitted line",
28         "Vertical deviations"),
29       col = c("blue", "red", "green"),
30       pch = c(16, NA, NA),
31       lty = c(NA, 1, 2),
32       lwd = c(NA, 2, 1))
33
34 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

```
1 #Page 694
2
3 xi <- c(1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 4.6, 1.6,
4         5.5, 3.4)
5
6 yi <- c(0.7, -1.0, -0.2, -1.2, -0.1, 3.4, 0.0, 0.8,
7         3.7, 2.0)
8
9 linear_model <- lm(yi ~ xi)
10
11 plot(xi, yi,
12       xlim = c(-1, 6), ylim = c(-2, 5),
13       xlab = "Reaction to Drug A (x)", ylab = "
14         Reaction to Drug B (y)",
15       main = "Reactions to Two Drugs with Fitted
16         Lines",
17       pch = 16, col = "blue")
18
19 abline(linear_model, col = "red", lwd = 2)
20
21 x_seq <- seq(-1, 6, length.out = 100)
22 y_parabola <- -0.744 + 0.616*x_seq + 0.013*x_seq^2
23 lines(x_seq, y_parabola, col = "green", lwd = 2)
24
25 legend("topleft",
26       legend = c("Observed values", "Linear fit", "
27         Parabolic fit"),
28       col = c("blue", "red", "green"),
29       pch = c(16, NA, NA),
30       lty = c(NA, 1, 1),
31       lwd = c(NA, 2, 2))
```

R code Exa 11.1.5 Fitting a Linear Function of Two Variables

```
1 #Page 696
2
3 xi1 <- c(1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 4.6, 1.6,
          5.5, 3.4)
4 xi2 <- c(66, 62, 64, 61, 63, 70, 68, 62, 68, 66)
5 yi <- c(0.7, -1.0, -0.2, -1.2, -0.1, 3.4, 0.0, 0.8,
          3.7, 2.0)
6
7 data <- data.frame(xi1, xi2, yi)
8
9 model <- lm(yi ~ xi1 + xi2, data=data)
10
11 coefficients <- round(coef(model),4)
12 cat(" Coefficients:\n")
13 cat(" 0 =", coefficients[1], "\n")
14 cat(" 1 =", coefficients[2], "\n")
15 cat(" 2 =", coefficients[3], "\n")
16
17 cat("\nLeast-Squares Linear Function:\n")
18 cat("y =", coefficients[1], "+", coefficients[2], "*
      x1 +", coefficients[3], "* x2\n")
19 #The answer may slightly vary due to rounding off
    values.
```

R code Exa 11.2.1 Pressure and the Boiling Point of Water I

```
1 #Page 698
2
3 boiling_point <- c(194.5, 194.3, 197.9, 198.4,
                    199.4, 199.9, 200.9, 201.1, 201.4, 201.3,
```

```

4             203.6, 204.6, 209.5, 208.6,
              210.7, 211.9, 212.2)
5 pressure <- c(20.79, 20.79, 22.40, 22.67, 23.15,
              23.35, 23.89, 23.99, 24.02, 24.01,
6             25.14, 26.57, 28.49, 27.76, 29.04,
              29.88, 30.06)
7
8 data <- data.frame(boiling_point, pressure)
9
10 model <- lm(pressure ~ boiling_point, data=data)
11
12 coefficients <- round(coef(model),4)
13 cat(" Coefficients:\n")
14 cat("  0 =", coefficients[1], "\n")
15 cat("  1 =", coefficients[2], "\n")
16
17 cat("\nLeast-Squares Line:\n")
18 cat("y =", coefficients[1], "+", coefficients[2], "*
      x\n")
19 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 11.2.5 Predicting Pressure from the Boiling Point of Water

```

1 #Page 704
2
3 boiling_point <- c(194.5, 194.3, 197.9, 198.4,
                  199.4, 199.9, 200.9, 201.1, 201.4, 201.3,
4                  203.6, 204.6, 209.5, 208.6,
                  210.7, 211.9, 212.2)
5 pressure <- c(20.79, 20.79, 22.40, 22.67, 23.15,
              23.35, 23.89, 23.99, 24.02, 24.01,
6             25.14, 26.57, 28.49, 27.76, 29.04,
              29.88, 30.06)
7

```

```

8 data <- data.frame(boiling_point, pressure)
9
10 model <- lm(pressure ~ boiling_point, data=data)
11
12 coefficients <- coef(model)
13 cat(" Coefficients:\n")
14 cat(" 0 =", coefficients[1], "\n")
15 cat(" 1 =", coefficients[2], "\n")
16
17 predicted_pressure <- coefficients[1] + coefficients
    [2] * 201.5
18 cat("\nPredicted Pressure for boiling point 201.5:",
    round(predicted_pressure,1), "\n")
19
20 plot(data$boiling_point, data$pressure, main="
    Boiling Point vs Pressure",
21       xlab="Boiling Point", ylab="Pressure", pch=19,
    col="blue")
22 abline(model, col="red")
23
24 points(201.5, predicted_pressure, col="green", pch
    =19)
25 text(201.5, predicted_pressure, labels=round(
    predicted_pressure, 2), pos=4, col="green")
26
27 segments(201.5, par("usr")[3], 201.5, predicted_
    pressure, col="green", lty=2)
28 segments(par("usr")[1], predicted_pressure, 201.5,
    predicted_pressure, col="green", lty=2)

```

R code Exa 11.3.2 Gasoline Mileage I

```

1 #Page 714
2
3 beta0 <- 0.01537

```

```

4 beta1 <- 1.396e-4
5 sigma <- 7.181e-3
6 x_mean <- 183.97
7 sx <- 1036.9
8
9 U1 <- sx * (beta1 - 0) / sigma
10
11 cat("Test statistic U1:", round(U1,2), "\n")
12 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 11.3.3 Pressure and the Boiling Point of Water II

```

1 #Page 715
2
3 beta0 <- -81.049
4 beta1 <- 0.5228
5 n <- 17
6 s2x <- 530.78
7 x_mean <- 202.95
8 sigma <- 0.2328
9 x_test <- 201.5
10 y_test <- 24.5
11
12 U01 <- sqrt(1/n + (x_mean - x_test)^2 / s2x) * (
    beta0 + x_test * beta1 - y_test) / sigma
13
14 df <- n - 2
15
16 p_value <- 2 * pt(abs(U01), df, lower.tail = FALSE)
17
18 cat("Test statistic U01:", round(U01,4), "\n")
19 cat("Degrees of freedom:", df, "\n")
20 cat("p-value:", round(p_value,4), "\n")

```

R code Exa 11.3.4 Gasoline Mileage II

```
1 #Page 716
2
3 beta_hat_1 <- 1.396e-4
4 sigma <- 7.181e-3
5 x_bar <- 183.97
6 sx <- 1036.9
7 n <- 173
8 alpha_0 <- 0.2
9
10 se_beta_1 <- sigma / (sx)
11
12 t_value <- qt(1 - alpha_0/2, df = n-2)
13
14 margin_of_error <- se_beta_1 * t_value
15
16 ci_lower <- beta_hat_1 - margin_of_error
17 ci_upper <- beta_hat_1 + margin_of_error
18
19 cat("t-value for 0.8 confidence interval:", round(t_
    value,3), "\n")
20
21 cat("80% Confidence Interval for 1:\n")
22 cat("Lower bound:", format(ci_lower,digits = 4,
    scientific = TRUE), "\n")
23 cat("Upper bound:", format(ci_upper,digits = 4,
    scientific = TRUE), "\n")
```

R code Exa 11.3.5 Gasoline Mileage III

```
1 #Page 717
```

```

2
3 beta_0 <- 0.01537
4 beta_1 <- 1.396e-4
5 sigma <- 7.181e-3
6 x_bar <- 183.97
7 sx <- 1036.9
8 n <- 173
9 alpha_0 <- 0.1
10 x <- 100
11
12 y_hat <- beta_0 + beta_1 * x
13
14 se_pred <- sigma * sqrt(1 + 1/n + (x - x_bar)^2 / (
    sx^2 * (n-1)))
15
16 t_value <- qt(1 - alpha_0/2, df = n-2)
17
18 margin_of_error <- t_value * round(se_pred,5)
19
20 lower_gpm <- round(y_hat - margin_of_error,5)
21 upper_gpm <- round(y_hat + margin_of_error,5)
22
23 lower_mpg <- 1 / upper_gpm
24 upper_mpg <- 1 / lower_gpm
25
26 cat("90% Prediction Interval:\n")
27 cat("For Gallons per Mile:\n")
28 cat("Lower bound:", lower_gpm, "\n")
29 cat("Upper bound:", upper_gpm, "\n\n")
30 cat("For Miles per Gallon:\n")
31 cat("Lower bound:", round(lower_mpg, 2), "\n")
32 cat("Upper bound:", round(upper_mpg, 2), "\n")
33 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 11.3.6 Pressure and the Boiling Point of Water III

```
1 #Page 717
2
3 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)
4 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)
5
6 beta0 <- -81.06
7 beta1 <- 0.5229
8 y_hat <- beta0 + beta1 * xi
9 e_i <- yi - y_hat
10
11 par(mfrow=c(1,2))
12
13 plot(xi, e_i, main="Residuals vs Boiling Point",
      xlab="Boiling Point", ylab="Residuals")
14 abline(h=0, col="red", lty=2)
15
16 log_yi <- log(yi)
17 log_beta0 <- -0.9709
18 log_beta1 <- 0.0206
19 log_y_hat <- log_beta0 + log_beta1 * xi
20 log_e_i <- log_yi - log_y_hat
21
22 plot(xi, log_e_i, main="Log Residuals vs Boiling
      Point", xlab="Boiling Point", ylab="Log Residuals
      ")
23 abline(h=0, col="red", lty=2)
24 par(mfrow=c(1,1))
```

R code Exa 11.3.7 Pressure and the Boiling Point of Water IV


```

1 # Page 720
2
3 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)
4 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)
5
6 log_yi <- log(yi)
7 log_beta0 <- -0.9709
8 log_beta1 <- 0.0206
9 log_y_hat <- log_beta0 + log_beta1 * xi
10 log_e_i <- log_yi - log_y_hat
11
12 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

```

1 #Page 725
2
3 n <- 16
4
5 t_quantile <- qt(0.9917, df = n - 2)
6
7 cat("The t-quantile for a 0.9917 confidence level
      with", n - 2, "degrees of freedom is:", t_
      quantile, "\n")
8 # The answer may slightly vary due to rounding off
      values.

```

R code Exa 11.4.2 Pressure and the Boiling Point of Water VI

```

1 #Page
2
3 alpha0 <- 0.1
4 n <- 16
5 beta1_hat <- 0.0205
6 sx <- 449.2
7
8 df <- n - 1
9
10 t_quantile <- qt(1 - alpha0 / 2, df = df)
11
12 margin_of_error <- round(t_quantile / sx,4)
13 lower_bound <- beta1_hat - margin_of_error
14 upper_bound <- beta1_hat + margin_of_error
15
16 cat("The t-quantile for a", 1 - alpha0, "confidence
      level with", df, "degrees of freedom is:", round(
      t_quantile,4), "\n")
17 cat("The interval estimate for beta1 is:", "(",
      lower_bound, ", ", upper_bound, ")\n")
18 # The answer may slightly vary due to rounding off
    values.

```

R code Exa 11.4.3 Pressure and the Boiling Point of Water VII

```

1 #Page 733
2
3 beta0_hat <- -0.9518
4 beta1_hat <- 0.0205
5 x <- 208
6 n <- 16
7 x_bar <- 202.85
8 s2_x <- 527.9
9 sigma_hat <- 2.616e-3
10 alpha0 <- 0.1

```

```

11 t_value <- 1.761
12
13 Y_hat <- beta0_hat + beta1_hat * x
14
15 se <- sigma_hat * sqrt(1 + 1/n + (x - x_bar)^2 / s2_
    x)
16
17 margin_of_error <- t_value * se
18
19 log_pressure_lower <- Y_hat - margin_of_error
20 log_pressure_upper <- Y_hat + margin_of_error
21
22 pressure_lower <- exp(log_pressure_lower)
23 pressure_upper <- exp(log_pressure_upper)
24
25 cat("The interval for log-pressure is: (", round(log_
    _pressure_lower,3), ", ", round(log_pressure_upper
    ,3), ")\n")
26 cat("The interval for pressure is: (", round(
    pressure_lower,2), ", ", round(pressure_upper,2),
    ")\n")

```

R code Exa 11.4.4 Gasoline Mileage IV

```

1 #Page 734
2
3 sx <- 1036.9
4 sigma_hat <- 7.181e-3
5 beta1_hat <- 1.396e-4
6 n <- 173
7 df <- n - 2
8
9 T171 <- function(x) pt(x, df)
10
11 posterior_cdf <- function(c) {

```

```

12   T171(sx / sigma_hat * (c - beta1_hat)) - T171(sx /
      sigma_hat * (-c - beta1_hat))
13 }
14
15 c_values <- seq(0, 2e-4, by = 1e-6)
16 cdf_values <- sapply(c_values, posterior_cdf)
17
18 plot(c_values, cdf_values, type = "l", col = "blue",
      lwd = 2, xlim = c(0.00010, 0.00018),
19      xlab = "|beta1|", ylab = "Posterior CDF", main
      = "Posterior CDF of |beta1|")
20 abline(v = 1.2e-4, col = "red", lty = 2)
21 abline(v = 1.6e-4, col = "green", lty = 2)
22
23 legend("bottomright", legend = c("|beta1| = 1.2e-4",
      "|beta1| = 1.6e-4"),
24      col = c("red", "green"), lty = 2, cex = 0.8)

```

R code Exa 11.5.1 Unemployment in the 1950s I

```

1 #Page 736
2
3 unemployment <- c(3.1, 1.9, 1.7, 1.6, 3.2, 2.7, 2.6,
      2.9, 4.7, 3.8)
4 index_production <- c(113, 123, 127, 138, 130, 146,
      151, 152, 141, 159)
5 year <- c(1950, 1951, 1952, 1953, 1954, 1955, 1956,
      1957, 1958, 1959)
6
7 par(mfrow = c(1, 2))
8
9 plot(index_production, unemployment,
10      main = "Unemployment vs. Index of Production",
11      xlab = "Index of Production",
12      ylab = "Unemployment",

```

```

13     pch = 19,
14     col = "blue")
15
16 plot(year, unemployment,
17       main = "Unemployment vs. Year",
18       xlab = "Year",
19       ylab = "Unemployment",
20       pch = 19,
21       col = "red")
22
23 par(mfrow = c(1, 1))

```

R code Exa 11.5.3 Unemployment in the 1950s II

```

1 #Page 741
2
3 y <- c(3.1, 1.9, 1.7, 1.6, 3.2, 2.7, 2.6, 2.9, 4.7,
4       3.8)
5 index_production <- c(113, 123, 127, 138, 130, 146,
6                       151, 152, 141, 159)
7 year <- c(1950, 1951, 1952, 1953, 1954, 1955, 1956,
8           1957, 1958, 1959)
9
10 Z <- cbind(rep(1, 10), index_production, year -
11            1949)
12
13 ZtZ_inv <- solve(t(Z) %*% Z)
14
15 Zty <- t(Z) %*% y
16
17 beta_hat <- ZtZ_inv %*% Zty
18
19 beta_hat[1] <- round(beta_hat[1], 2)
20 print(round(beta_hat, 4))

```

R code Exa 11.5.4 Dishwasher Shipments I

```
1 #Page 743
2
3 cov_beta1_beta2 <- -0.0005636
4 var_beta1 <- 0.001136
5 var_beta2 <- 0.0007026
6
7 rho <- cov_beta1_beta2 / sqrt(var_beta1 * var_beta2)
8
9 print(round(rho,4))
```

R code Exa 11.5.5 Dishwasher Shipments II

```
1 #Page 746
2
3 beta_hat_1 <- 66.91
4 sigma_hat <- 352.9
5 var_beta1 <- 0.001136
6 n <- 26
7 p <- 3
8
9 U <- beta_hat_1 / (sqrt(var_beta1) * sigma_hat)
10 round(U,3)
11
12 df <- n - p
13 print(df)
14
15 p_value <- 2 * pt(-abs(U), df)
16 signif(p_value,1)
```

R code Exa 11.5.6 Unemployment in the 1950s III

```
1 #Page 746
2
3 beta_hat_2 <- 0.6594
4 sigma_hat <- 0.4011
5 var_beta2 <- 0.06762
6 n <- 10
7 p <- 3
8 c <- 0.4
9
10 U <- (beta_hat_2 - c) / (sqrt(var_beta2) * sigma_hat
11    )
12 round(U,3)
13 df <- n - p
14 print(df)
15
16 p_value <- 1 - pt(U, df)
17 signif(p_value,3)
```

R code Exa 11.5.7 Predicting Dishwasher Shipments

```
1 #Page 748
2
3 beta_hat_0 <- -1314
4 beta_hat_1 <- 66.91
5 beta_hat_2 <- 58.86
6 sigma_hat <- 352.9
7 alpha <- 0.1
8 investment_1986 <- 67.2
9 n <- 10
```

```

10 p <- 3
11 c <- 1986
12
13 z <- c(1, c - 1960, investment_1986)
14
15 Y_hat <- trunc(sum(z * c(beta_hat_0, beta_hat_1,
    beta_hat_2)))
16
17 se <- sigma_hat * sqrt(1 + 0.2136)
18
19 t_quantile <- qt(1 - alpha/2, df = 23)
20
21 lower_bound <- trunc(Y_hat - t_quantile * se)
22 upper_bound <- trunc(Y_hat + t_quantile * se)
23
24 print(lower_bound)
25 print(upper_bound)

```

R code Exa 11.5.8 Unemployment in the 1950s IV

```

1 #Page 748
2
3 unemployment <- c(3.1, 1.9, 1.7, 1.6, 3.2, 2.7, 2.6,
    2.9, 4.7, 3.8)
4
5 y10 <- 2.82
6
7 sum_of_squares <- sum((unemployment - y10)^2)
8
9 sigma_hat_squared <- 1.126 / (10 - 3)
10
11 R_squared <- 1 - (1.126 / sum_of_squares)
12
13 print(paste("y10:", y10))
14 print(paste("Sum of Squares:", sum_of_squares))

```



```
15 print(paste("R^2:", round(R_squared,4)))
```

R code Exa 11.5.9 Unemployment in the 1950s V

```
1 #Page 749
2
3 library(ggplot2)
4 library(gridExtra)
5
6 data <- data.frame(
7   Unemployment = c(3.1, 1.9, 1.7, 1.6, 3.2, 2.7,
8     2.6, 2.9, 4.7, 3.8),
9   Production = c(113, 123, 127, 138, 130, 146, 151,
10    152, 141, 159),
11   Year = c(1950, 1951, 1952, 1953, 1954, 1955, 1956,
12    1957, 1958, 1959)
13 )
14
15 model_full <- lm(Unemployment ~ Production, data =
16   data)
17 residuals_full <- resid(model_full)
18
19 data_subset <- subset(data, Year >= 1951)
20 model_subset <- lm(Unemployment ~ Production, data =
21   data_subset)
22 residuals_subset <- resid(model_subset)
23
24 data_full <- data.frame(Production = data$Production
25   , Residuals = residuals_full, Year = data$Year)
26 data_subset_plot <- data.frame(Production = data_
27   subset$Production, Residuals = residuals_subset,
28   Year = data_subset$Year)
29
30 p1 <- ggplot(data_full, aes(x = Production, y =
31   Residuals)) +
```

```

23   geom_point() +
24   geom_hline(yintercept = 0, linetype = "dashed") +
25   labs(title = "Residuals vs Production (1950–1959)"
26         ,
27         x = "Production",
28         y = "Residuals") +
29   theme_minimal()
30 p2 <- ggplot(data_full, aes(x = Year, y = Residuals)) +
31   geom_point() +
32   geom_hline(yintercept = 0, linetype = "dashed") +
33   labs(title = "Residuals vs Year (1950–1959)",
34         x = "Year",
35         y = "Residuals") +
36   theme_minimal()
37
38 p3 <- ggplot(data_subset_plot, aes(x = Production, y
39   = Residuals)) +
40   geom_point() +
41   geom_hline(yintercept = 0, linetype = "dashed") +
42   labs(title = "Residuals vs Production (1951–1959)"
43         ,
44         x = "Production",
45         y = "Residuals") +
46   theme_minimal()
47
48 p4 <- ggplot(data_subset_plot, aes(x = Year, y =
49   Residuals)) +
50   geom_point() +
51   geom_hline(yintercept = 0, linetype = "dashed") +
52   labs(title = "Residuals vs Year (1951–1959)",
53         x = "Year",
54         y = "Residuals") +
55   theme_minimal()
56 grid.arrange(p1, p2, p3, p4, ncol = 2)

```

R code Exa 11.5.10 Dishwasher Shipments III

```
1 #Page 750
2
3 library(ggplot2)
4 library(gridExtra)
5
6 data <- data.frame(
7   Year = 1960:1985,
8   Dishwasher_Shipments = c(555, 620, 720, 880, 1050,
9     1290, 1528, 1586, 1960, 2118, 2116, 2477,
10    3199, 3702, 3320, 2702, 3140, 3356, 3558, 3488,
11    2738, 2484, 2170, 3092, 3491, 3536),
12   Private_Residential_Investment = c(34.2, 34.3,
13     37.7, 42.5, 43.1, 42.7, 38.2, 37.1, 43.1, 43.6,
14     41.0, 53.7, 63.8, 62.3, 48.2, 42.2, 51.2,
15     60.7, 62.4, 59.1, 47.1, 44.7, 37.8, 52.7, 60.3,
16     61.4)
17 )
18
19 model <- lm(Dishwasher_Shipments ~ Private_
20   Residential_Investment, data = data)
21 residuals <- resid(model)
22 fitted_values <- fitted(model)
23
24 plot_data <- data.frame(
25   Year = data$Year,
26   Private_Residential_Investment = data$Private_
27     Residential_Investment,
28   Residuals = residuals,
29   Fitted_Values = fitted_values
30 )
31
32 p1 <- ggplot(plot_data, aes(x = Year, y = Residuals))
```

```

    ) +
24   geom_point() +
25   geom_hline(yintercept = 0, linetype = "dashed") +
26   labs(title = "Residuals vs Year",
27         x = "Year",
28         y = "Residuals") +
29   theme_minimal()
30
31 p2 <- ggplot(plot_data, aes(x = Private_Residential_
    Investment, y = Residuals)) +
32   geom_point() +
33   geom_hline(yintercept = 0, linetype = "dashed") +
34   labs(title = "Residuals vs Private Residential
    Investment",
35         x = "Private Residential Investment (billions
    of 1972 dollars)",
36         y = "Residuals") +
37   theme_minimal()
38
39 p3 <- ggplot(plot_data, aes(x = Fitted_Values, y =
    Residuals)) +
40   geom_point() +
41   geom_hline(yintercept = 0, linetype = "dashed") +
42   labs(title = "Residuals vs Fitted Values",
43         x = "Fitted Values",
44         y = "Residuals") +
45   theme_minimal()
46
47 successive_residuals <- data.frame(
48   Residuals = residuals[-length(residuals)],
49   Successive_Residuals = residuals[-1]
50 )
51
52 p4 <- ggplot(successive_residuals, aes(x = Residuals
    , y = Successive_Residuals)) +
53   geom_point() +
54   geom_hline(yintercept = 0, linetype = "dashed") +
55   geom_vline(xintercept = 0, linetype = "dashed") +

```

```

56   labs(title = "Pairs of Successive Residuals",
57         x = "Residuals",
58         y = "Successive Residuals") +
59   theme_minimal()
60
61   grid.arrange(p1, p2, p3, p4, ncol = 2)

```

R code Exa 11.6.4 Calories in Hot Dogs I

```

1  #Page 759
2
3  calories <- c(
4    186, 181, 176, 149, 184, 190, 158, 139, 175, 148,
5      152, 111, 141, 153, 190, 157, 131, 149, 135,
6      132,
7    173, 191, 182, 190, 172, 147, 146, 139, 175, 136,
8      179, 153, 107, 195, 135, 140, 138,
9    129, 132, 102, 106, 94, 102, 87, 99, 107, 113,
10     135, 142, 86, 143, 152, 146, 144,
11     155, 170, 114, 191, 162, 146, 140, 187, 180
12  )
13
14 type <- factor(c(
15   rep("Beef", 20), rep("Meat", 17), rep("Poultry",
16     17), rep("Specialty", 9)
17 ))
18
19 anova_result <- aov(calories ~ type)
20
21 summary(anova_result)[[1]][, -c(4,5)]

```

R code Exa 11.6.5 Calories in Hot Dogs II

```

1 #Page 760
2
3 calories <- c(
4   186, 181, 176, 149, 184, 190, 158, 139, 175, 148,
5     152, 111, 141, 153, 190, 157, 131, 149, 135,
6     132,
7   173, 191, 182, 190, 172, 147, 146, 139, 175, 136,
8     179, 153, 107, 195, 135, 140, 138,
9   129, 132, 102, 106, 94, 102, 87, 99, 107, 113,
10    135, 142, 86, 143, 152, 146, 144,
11    155, 170, 114, 191, 162, 146, 140, 187, 180
12 )
13
14 type <- factor(c(
15   rep("Beef", 20), rep("Meat", 17), rep("Poultry",
16     17), rep("Specialty", 9)
17 ))
18
19 anova_result <- aov(calories ~ type)
20
21 anova_table <- summary(anova_result)
22
23 f_statistic <- anova_table[[1]]$'F value'[1]
24 round(f_statistic,2)
25
26 p_value <- anova_table[[1]]$'Pr(>F)'[1]
27 signif(p_value,2)
28 #The answer may slightly vary due to rounding off
29 values.

```

R code Exa 11.6.6 Calories in Hot Dogs III

```

1 #Page 760
2
3 calories <- c(

```

```

4    186, 181, 176, 149, 184, 190, 158, 139, 175, 148,
      152, 111, 141, 153, 190, 157, 131, 149, 135,
      132,
5    173, 191, 182, 190, 172, 147, 146, 139, 175, 136,
      179, 153, 107, 195, 135, 140, 138,
6    129, 132, 102, 106, 94, 102, 87, 99, 107, 113,
      135, 142, 86, 143, 152, 146, 144,
7    155, 170, 114, 191, 162, 146, 140, 187, 180
8  )
9
10 type <- factor(c(
11   rep("Beef", 20), rep("Meat", 17), rep("Poultry",
      17), rep("Specialty", 9)
12 ))
13
14 anova_result <- aov(calories ~ type)
15 residuals <- resid(anova_result)
16
17 par(mfrow=c(1, 2))
18 stripchart(residuals ~ type, vertical=TRUE, method="
      jitter",
19           main="Residuals vs. Hot Dog Type", xlab="
      Hot Dog Type", ylab="Residuals",
20           pch=16, col="blue")
21
22 symbols <- c(1, 2, 3, 4)
23 symbol_mapping <- symbols[type]
24 qqnorm(residuals, main="Normal Q-Q Plot of Residuals
      ", pch=symbol_mapping, col="blue")
25 legend("topleft", legend=levels(type), pch=symbols,
      col="blue")
26
27 par(mfrow=c(1,1))

```

R code Exa 11.7.2 Radioactive Isotope in Milk I

```

1 #Page 765
2
3 data <- matrix(c(6.4, 3.2, 6.9,
4                 8.5, 7.8, 10.1,
5                 9.3, 6.0, 9.6,
6                 8.8, 5.6, 8.4),
7               nrow = 4, byrow = TRUE)
8
9 row_averages <- rowMeans(data)
10 col_averages <- colMeans(data)
11 grand_average <- mean(data)
12
13 alpha <- row_averages - grand_average
14 beta <- col_averages - grand_average
15 mu <- grand_average
16
17 fitted_values <- matrix(NA, nrow = 4, ncol = 3)
18 for (i in 1:4) {
19   for (j in 1:3) {
20     fitted_values[i, j] <- mu + alpha[i] + beta[j]
21   }
22 }
23
24 observed_values <- as.vector(data)
25 fitted_values_vec <- as.vector(fitted_values)
26
27 sum_of_squares <- sum((observed_values - fitted_
28                       values_vec)^2)
29
30 sigma_squared <- sum_of_squares / 12
31
32 cat("Row averages: ", row_averages, "\n")
33 cat("Column averages: ", col_averages, "\n")
34 cat("Grand average: ", grand_average, "\n")
35 cat("Fitted values:\n")
36 print(fitted_values)
37 cat("Sum of squares of differences: ", sum_of_
38     squares, "\n")

```



```
37 cat("Estimate of sigma^2: ", round(sigma_squared,3),  
      "\n")
```

R code Exa 11.7.3 Radioactive Isotope in Milk II

```
1 #Page 768  
2  
3 data <- data.frame(  
4   Dairy = factor(rep(1:4, each = 3)),  
5   Method = factor(rep(1:3, times = 4)),  
6   Value = c(6.4, 3.2, 6.9, 8.5, 7.8, 10.1, 9.3, 6.0,  
7             9.6, 8.8, 5.6, 8.4)  
8 )  
9 model <- aov(Value ~ Dairy + Method, data = data)  
10  
11 summary(model)[[1]][, -c(4,5)]
```

R code Exa 11.7.5 Testing for Differences among the Dairies

```
1 #Page 770  
2  
3 MSE_Dairy <- 6.33  
4 MSE_Residuals <- 0.4567  
5  
6 U2A <- MSE_Dairy / MSE_Residuals  
7  
8 p_value <- pf(U2A, df1 = 3, df2 = 6, lower.tail =  
9               FALSE)  
10  
11 print(paste("U2A:", round(U2A,2)))  
12 print(paste("p-value:", signif(p_value,1)))
```

R code Exa 11.7.6 Testing for Differences among the Methods of Measurement

```
1 #Page 770
2
3 MSE_Method <- 11.08
4 MSE_Residuals <- 0.4567
5
6 U2B <- MSE_Method / MSE_Residuals
7
8 p_value <- pf(U2B, df1 = 2, df2 = 6, lower.tail =
  FALSE)
9
10 print(paste("U2B:", round(U2B,2)))
11 print(paste("p-value:", signif(p_value,1)))
```

R code Exa 11.8.2 Gasoline Consumption I

```
1 #Page 775
2
3 data <- data.frame(
4   Device = rep(c("Equipped", "Not Equipped"), each =
5     3),
6   Model = rep(c("Compact", "Intermediate", "Standard
7     "), times = 2),
8   Average = c(8.58, 10.12, 11.08, 8.94, 10.20,
9     11.40)
10 )
11 mu_hat <- mean(data$Average)
12
13 row_means <- tapply(data$Average, data$Device, mean)
```

```

12 alpha_hat <- row_means - mu_hat
13
14 col_means <- tapply(data$Average, data$Model, mean)
15 beta_hat <- col_means - mu_hat
16
17 cell_means <- tapply(data$Average, list(data$Device,
    data$Model), mean)
18
19 gamma_hat <- cell_means - outer(row_means, col_means
    , "+") + mu_hat
20
21 cat(" Estimates:\n")
22 cat("      =", round(mu_hat,4), "\n\n")
23
24 cat("      (Device effect):\n")
25 print(round(alpha_hat,4))
26 cat("\n")
27
28 cat("      (Model effect):\n")
29 print(round(beta_hat,4))
30 cat("\n")
31
32 cat("      (Interaction effect):\n")
33 print(round(gamma_hat,4))

```

R code Exa 11.8.3 Gasoline Consumption II

```

1 #Page 776
2
3 ss_device <- 0.4813
4 ss_model <- 30.92
5 ss_interactions <- 0.1147
6 ss_residuals <- 18.22
7
8 df_device <- 1

```

```

9 df_model <- 2
10 df_interactions <- 2
11 df_residuals <- 24
12
13 ms_device <- ss_device / df_device
14 ms_model <- ss_model / df_model
15 ms_interactions <- ss_interactions / df_interactions
16 ms_residuals <- ss_residuals / df_residuals
17
18 anova_table <- data.frame(
19   Source = c("Main effects of device", "Main effects
             of model", "Interactions", "Residuals", "Total
             "),
20   'Degrees of freedom' = c(df_device, df_model, df_
             interactions, df_residuals, 29),
21   'Sum of squares' = c(ss_device, ss_model, ss_
             interactions, ss_residuals, sum(c(ss_device, ss_
             _model, ss_interactions, ss_residuals, -0.006)))
             ,
22   'Mean square' = round(c(ms_device, ms_model, ms_
             interactions, ms_residuals, NA), 4)
23 )
24
25 print("ANOVA table")
26 print(anova_table)

```

R code Exa 11.8.4 Gasoline Consumption III

```

1 #Page 778
2
3 MS_interaction <- 0.0573
4 MS_residual <- 0.7590
5
6 df_interaction <- 2
7 df_residual <- 24

```

```

8
9 F_statistic <- round(MS_interaction / MS_residual,4)
10
11 p_value <- 1 - pf(F_statistic, df_interaction, df_
    residual)
12
13 cat("F-statistic (U^2_AB):", round(F_statistic,3), "
    \n")
14 cat("p-value:", round(p_value,4), "\n")

```

R code Exa 11.8.5 Gasoline Consumption IV

```

1 #Page 779
2
3 MS_device <- 0.4813
4 MS_interaction <- 0.1147
5 MS_residual <- 0.7590
6 SS_residual <- 18.22
7
8 df_device <- 1
9 df_interaction <- 2
10 df_residual <- 24
11
12 U_squared_A <- 24 * (MS_device + MS_interaction) /
    (3 * SS_residual)
13 p_value_U_A <- 1 - pf(U_squared_A, df_device + df_
    interaction, df_residual)
14
15 V_squared_A <- MS_device / MS_residual
16 p_value_V_A <- 1 - pf(V_squared_A, df_device, df_
    residual)
17
18 cat("Test for Device Effect (without considering
    interaction)\n")
19 cat("

```

```

n")
20 cat(" U A :", round(U_squared_A,4), "\n")
21 cat(" Degrees of freedom:", df_device + df_
    interaction, "and", df_residual, "\n")
22 cat(" p-value:", round(p_value_U_A,4), "\n\n")
23
24 cat(" Test for Device Effect (considering interaction
    )\n")
25 cat("
    n")
26 cat(" V A :", round(V_squared_A,4), "\n")
27 cat(" p-value:", round(p_value_V_A,2), "\n")

```

Chapter 12

Simulation

R code Exa 12.1.1 The Mean of a Distribution

```
1 #Page 787
2
3 set.seed(12)
4
5 generate_sample_mean <- function(n) {
6   round(mean(runif(n)),3)
7 }
8
9 num_replications <- 5
10
11 n_values <- c(100, 1000, 10000, 100000)
12
13 results <- matrix(NA, nrow = length(n_values), ncol
14   = num_replications + 1)
15 colnames(results) <- c("n", paste0("Rep", 1:num_
16   replications))
17
18 for (i in 1:length(n_values)) {
19   n <- n_values[i]
20   results[i, 1] <- n
21   for (j in 1:num_replications) {
```

```

20     results[i, j + 1] <- generate_sample_mean(n)
21   }
22 }
23
24 results_df <- as.data.frame(results)
25
26 print("Results of several different simulations:")
27 print(results_df)
28 #The answer may be different due to randomization in
    the problem

```

R code Exa 12.1.2 A Normal Probability

```

1 #Page 787
2
3 set.seed(1234)
4
5 generate_proportion <- function(n) {
6   round(mean(rnorm(n) >= 1.0), 3)
7 }
8
9 num_replications <- 5
10
11 n_values <- c(100, 1000, 10000, 100000)
12
13 results <- matrix(NA, nrow = length(n_values), ncol
    = num_replications + 1)
14 colnames(results) <- c("n", paste0("Rep", 1:num_
    replications))
15
16 for (i in 1:length(n_values)) {
17   n <- n_values[i]
18   results[i, 1] <- n
19   for (j in 1:num_replications) {
20     results[i, j + 1] <- generate_proportion(n)

```



```

21   }
22 }
23
24 results_df <- as.data.frame(results)
25
26 print("Results of several different simulations:")
27 print(results_df)
28 #The answer may be different due to randomization in
    the problem

```

R code Exa 12.1.3 Waiting for a Break

```

1 #Page 788
2
3 set.seed(123)
4
5 num_simulations <- 10000
6
7 service_rate <- 0.3
8
9 X <- rgamma(num_simulations, shape = 10, rate =
    service_rate)
10 Y <- rgamma(num_simulations, shape = 10, rate =
    service_rate)
11
12 Z <- abs(X - Y)
13
14 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

```

1 #Page 789
2
3 set.seed(123)
4
5 longest_run <- function(flips) {
6   runs <- rle(flips)$lengths
7   max_run <- max(runs)
8   return(max_run)
9 }
10
11
12 n_simulations <- 10000
13 n_flips <- 100
14
15 longest_runs <- numeric(n_simulations)
16 for (i in 1:n_simulations) {
17   flips <- sample(c(0, 1), n_flips, replace = TRUE)
18   longest_runs[i] <- longest_run(flips)
19 }
20
21
22 hist(longest_runs,
23       breaks = seq(0.5, max(longest_runs) + 0.5, by =
24         1),
25       main = "Histogram of Longest Runs in 100 Coin
26         Flips",
27       xlab = "Length of Longest Run",
28       ylab = "Frequency",
29       xlim = c(2,15))

```

R code Exa 12.2.11 The MSE of the Sample Median

```

1 #Page 798
2
3 v0 <- 1000

```

```

4 n <- 20
5 sigma_hat <- 0.3892
6 gamma <- 0.95
7 epsilon <- 0.01
8
9 v <- ceiling((round(qnorm((gamma + 1) / 2), 2) *
  sigma_hat / epsilon)^2)
10
11 print(paste("Estimated number of simulations needed:
  ", v))

```

R code Exa 12.2.12 The Median of a Complicated Distribution

```

1 #Page 799
2
3 v0 <- 10000
4 gamma <- 0.99
5 epsilon <- 0.001
6 sigma_hat <- 0.3570
7
8 v <- ceiling((qnorm((gamma + 1) / 2) * sigma_hat /
  epsilon)^2)
9
10 print(paste("Estimated number of simulations needed:
  ", v))
11
12 total_simulations <- 900000
13 k <- 100
14 m <- 6200
15 sigma_hat_new <- 0.4529
16
17 v_new <- ceiling((qnorm((gamma + 1) / 2) * sigma_hat
  _new / epsilon)^2)
18
19 print(paste("New estimated number of simulations

```

```

    needed:", v_new))
20 #The answer may be different due to randomization in
    the problem

```

R code Exa 12.2.14 A Service Queue with Impatient Customers

```

1 #Page 800
2
3 set.seed(123)
4
5 service_times <- c(0.251, 2.215, 2.855, 0.666,
6   2.505)
7 interarrival_times <- c(0.215, 0.713, 1.44, 0.174,
8   0.342)
9 t_target <- 3
10
11 T <- numeric(5)
12 Z <- numeric(5)
13 W <- numeric(5)
14 V <- numeric(5)
15 N <- 0
16
17 for (j in 1:5) {
18   if (j == 1) {
19     T[j] <- interarrival_times[j]
20   } else {
21     T[j] <- T[j-1] + interarrival_times[j]
22   }
23
24   if (j == 1 || T[j] > max(W[1:(j-1)])) {
25     Z[j] <- T[j]
26   } else {
27     r <- sum(W[1:(j-1)] > T[j])
28     if (r > 0) {
29       p <- (r - 1) / r
30     }
31   }
32 }

```

```

28     V[j] <- rbinom(1, 1, p)
29     if (V[j] == 1) {
30         Z[j] <- T[j]
31     } else {
32         Z[j] <- max(W[1:(j-1)])
33     }
34 } else {
35     Z[j] <- max(W[1:(j-1)])
36 }
37 }
38
39 W[j] <- Z[j] + service_times[j]
40 }
41
42 N <- sum(W > t_target)-1
43
44 print(data.frame(Customer = 1:5, T = T, Z = Z, W = W
45 ))
46 cat("Number of customers in queue at t =", t_target,
47     ":", N, "\n")

```

R code Exa 12.3.7 Bayesian Analysis of OneWay Layout

```

1 #Page 808
2
3 library(ggplot2)
4 library(tidyr)
5 library(dplyr)
6
7 calories_beef <- c(186, 181, 176, 149, 184, 190,
8                   158, 139, 175, 148, 152, 111, 141, 153, 190, 157,
9                   131, 149, 135, 132)
10 calories_meat <- c(173, 191, 182, 190, 172, 147,
11                   146, 139, 175, 136, 179, 153, 107, 195, 135, 140,
12                   138)

```

```

9  calories_poultry <- c(129, 132, 102, 106, 94, 102,
    87, 99, 107, 113, 135, 142, 86, 143, 152, 146,
    144)
10 calories_specialty <- c(155, 170, 114, 191, 162,
    146, 140, 187, 180)
11
12 pairwise_diff <- function(means) {
13   n <- length(means)
14   diffs <- c()
15   for (i in 1:(n-1)) {
16     for (j in (i+1):n) {
17       diffs <- c(diffs, abs(means[i] - means[j]))
18     }
19   }
20   return(diffs)
21 }
22
23 set.seed(123)
24 n_sim <- 60000
25 max_diff <- numeric(n_sim)
26 min_diff <- numeric(n_sim)
27 avg_diff <- numeric(n_sim)
28
29 for (i in 1:n_sim) {
30   means <- c(
31     mean(sample(calories_beef, replace = TRUE, size
      = length(calories_beef))),
32     mean(sample(calories_meat, replace = TRUE, size
      = length(calories_meat))),
33     mean(sample(calories_poultry, replace = TRUE,
      size = length(calories_poultry))),
34     mean(sample(calories_specialty, replace = TRUE,
      size = length(calories_specialty)))
35   )
36   diffs <- pairwise_diff(means)
37   max_diff[i] <- max(diffs)
38   min_diff[i] <- min(diffs)
39   avg_diff[i] <- mean(diffs)

```

```

40 }
41
42 df <- data.frame(
43   max = max_diff,
44   min = min_diff,
45   avg = avg_diff
46 )
47
48 df_long <- pivot_longer(df, cols = everything(),
49   names_to = "type", values_to = "difference")
50
51 ggplot(df_long, aes(x = difference, color = type)) +
52   stat_ecdf(geom = "step") +
53   scale_color_manual(values = c("max" = "red", "min"
54     = "blue", "avg" = "green"),
55     labels = c("Largest", "Smallest", "Average")) +
56   labs(x = "i - j", y = "Cumulative
57     Probability",
58     title = "CDFs of Largest, Smallest, and
59     Average Pairwise Differences",
60     color = "Difference Type") +
61   theme_minimal() +
62   theme(legend.position = "bottom")

```

R code Exa 12.3.8 Comparing Copper Ores

```

1 #Page 810
2
3 set.seed(123)
4
5 v <- 10000
6
7 alpha_x <- 3.5
8 beta_x <- 0.16

```

```

 9  alpha_y <- 4.5
10  beta_y <- 0.11
11  mu_x <- 2.6
12  mu_y <- 2.3
13  n_x <- 8
14  n_y <- 10
15
16  tau_x <- rgamma(v, shape = alpha_x, rate = beta_x)
17  tau_y <- rgamma(v, shape = alpha_y, rate = beta_y)
18
19  mu_x_values <- rnorm(v, mean = mu_x, sd = sqrt(1 / (
    n_x * tau_x)))
20  mu_y_values <- rnorm(v, mean = mu_y, sd = sqrt(1 / (
    n_y * tau_y)))
21
22  diff_values <- mu_x_values - mu_y_values
23
24  abs_diff_values <- abs(diff_values)
25
26  par(mfrow = c(1, 2))
27
28  hist(diff_values, breaks = 30, probability = TRUE,
29       main = "Histogram of  $x - y$ ", xlab = " $x -$ 
         $y$ ", ylab = "Density")
30
31  plot(ecdf(abs_diff_values), main = "Posterior CDF of
     $|x - y|$ ",
32       xlab = " $|x - y|$ ", ylab = "CDF", col = "blue",
        lwd = 2)
33  par(mfrow = c(1, 1))

```

R code Exa 12.3.13 Simulating a Binomial Random Variable Using the Alias Method


```

2
3 set.seed(123)
4
5 xi <- c(0, 2, 1, 6, 7, 3, 8, 9, 4,5)
6 yi <- c(2, 3, 3, 3, 3, 4, 4, 4, 5, 0)
7 gi_xi <- c(0.101, 0.713, 0.605, 0.743, 0.212, 0.781,
            0.035, 0.003, 0.327, 1)
8
9 num_sim <- 10000
10
11 simulated_values <- numeric(num_sim)
12
13 for (i in 1:num_sim) {
14   Y <- runif(1)
15   I <- floor(length(xi) * Y) + 1
16   U <- length(xi) * Y + 1 - I
17
18   if (U <= gi_xi[I]) {
19     simulated_values[i] <- xi[I]
20   } else {
21     simulated_values[i] <- yi[ceiling(length(xi) *
22                                   runif(1))]
23   }
24 }
25 hist(simulated_values, breaks = 50, probability =
      TRUE, main = "Histogram of 10,000 Simulated
      Values",
26       xlab = "Simulated Values", border = "black")

```

R code Exa 12.4.1 Choosing an Importance Function

```

1 #Page 817
2
3 set.seed(123)

```

```

4
5 n <- 10000
6
7 U <- runif(n)
8
9 g <- function(x) ifelse(0 < x & x < 1, exp(-x) / (1
    + x^2), 0)
10
11 inverse_f0 <- function(u) u
12 inverse_f1 <- function(u) -log(1 - u)
13 inverse_f2 <- function(u) tan(pi * (u - 0.5))
14 inverse_f3 <- function(u) -log(1 - u * (1 - exp(-1)))
    )
15 inverse_f4 <- function(u) tan(pi * u / 4)
16
17 X0 <- inverse_f0(U)
18 X1 <- inverse_f1(U)
19 X2 <- inverse_f2(U)
20 X3 <- inverse_f3(U)
21 X4 <- inverse_f4(U)
22
23 Y0 <- g(X0)
24 Y1 <- g(X1) / exp(-X1)
25 Y2 <- g(X2) / ((1 + X2^2) / pi)
26 Y3 <- g(X3) / (exp(-X3) / (1 - exp(-1)))
27 Y4 <- g(X4) / (4 * (1 + X4^2) / pi)
28
29 Y0_est <- mean(Y0)
30 Y1_est <- mean(Y1)
31 Y2_est <- mean(Y2)
32 Y3_est <- mean(Y3)
33 Y4_est <- mean(Y4)
34
35 var_Y0 <- var(Y0)
36 var_Y1 <- var(Y1)
37 var_Y2 <- var(Y2)
38 var_Y3 <- var(Y3)
39 var_Y4 <- var(Y4)

```

```

40
41 table <- data.frame(
42   j = 0:4,
43   Y_j = c(Y0_est, Y1_est, Y2_est, Y3_est, Y4_est),
44   sigma_j = sqrt(c(var_Y0, var_Y1, var_Y2, var_Y3,
45                     var_Y4))
45 )
46 print(round(table,4))
47 #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

```

1 #Page 818
2
3 set.seed(12)
4
5 n <- 10000
6
7 alpha <- 5
8 alpha_minus_2 <- alpha - 2
9
10 integrand <- function(x) {
11   1 / (1 + x + x^2)
12 }
13
14 X <- rgamma(n, shape = alpha_minus_2, rate = 1)
15 Y <- rgamma(n, shape = alpha, rate = 1)
16
17 weight <- dgamma(X, shape = alpha, rate = 1) /
18   dgamma(X, shape = alpha_minus_2, rate = 1)
19
20 importance_samples <- integrand(X) * weight
21 importance_mean <- mean(importance_samples)
22 importance_sd <- sd(importance_samples)

```

```

22
23 direct_samples <- integrand(Y)
24 direct_mean <- mean(direct_samples)
25 direct_sd <- sd(direct_samples)
26
27 results <- data.frame(
28   Method = c("Importance Sampling", "Direct
      Simulation"),
29   Mean = round(c(importance_mean, direct_mean),5),
30   SD = round(c(importance_sd, direct_sd),5)
31 )
32
33 print(results)
34 #The answer may be different due to randomization in
    the problem

```

R code Exa 12.4.4 Illustration of Stratified Importance Sampling

```

1 #Page 821
2
3 set.seed(1)
4
5 n <- 10000
6 k_10 <- 10
7 k_100 <- 100
8 m_10 <- n / k_10
9 m_100 <- n / k_100
10
11 g <- function(x) {
12   exp(-x) / (1 + x + x^2)
13 }
14
15 f3 <- function(x) {
16   exp(-x) / (1 - exp(-1))
17 }

```

```

18
19 inverse_f3 <- function(u) {
20   -log(1 - u * (1 - exp(-1)))
21 }
22
23 U_10 <- runif(m_10 * k_10)
24 strata_10 <- rep(1:k_10, each = m_10)
25 X_10 <- inverse_f3((U_10 + strata_10 - 1) / k_10)
26 Y_10 <- g(X_10) / f3(X_10)
27
28 Y_10_est <- mean(Y_10)
29 stratified_var_10 <- sum(tapply(Y_10, strata_10, var
   ) * m_10) / (n^2)
30 stratified_se_10 <- sqrt(stratified_var_10)
31
32 U_100 <- runif(m_100 * k_100)
33 strata_100 <- rep(1:k_100, each = m_100)
34 X_100 <- inverse_f3((U_100 + strata_100 - 1) / k_
   100)
35 Y_100 <- g(X_100) / f3(X_100)
36
37 Y_100_est <- mean(Y_100)
38 stratified_var_100 <- sum(tapply(Y_100, strata_100,
   var) * m_100) / (n^2)
39 stratified_se_100 <- sqrt(stratified_var_100)
40
41 results <- data.frame(
42   Method = c("Stratified Importance Sampling (k=10)"
   , "Stratified Importance Sampling (k=100)"),
43   Estimate = round(c(Y_10_est, Y_100_est),4),
44   SE = signif(c(stratified_se_10, stratified_se_100)
   ,4)
45 )
46
47 print(results)
48 #The answer may be different due to randomization in
   the problem

```

R code Exa 12.5.2 Nursing Homes in New Mexico I

```
1 #Page 826
2
3 library(coda)
4
5 set.seed(12)
6
7 alpha1 <- 11
8 beta1 <- 50925.37
9 mu1 <- 183.95
10 lambda1 <- 20
11
12 initial_mu <- c(182.17, 227, 272, 137, 82)
13
14 iterations <- 100
15
16 gibbs_sampling <- function(initial_mu, iterations) {
17   mu_samples <- numeric(iterations)
18   tau_samples <- numeric(iterations)
19
20   mu <- initial_mu
21   for (i in 1:iterations) {
22     shape <- alpha1 + 0.5
23     scale <- beta1 + 0.5 * lambda1 * (mu - mu1)^2
24     tau <- rgamma(1, shape = shape, rate = scale)
25
26     mean_mu <- mu1
27     sd_mu <- sqrt(1 / (lambda1 * tau))
28     mu <- rnorm(1, mean = mean_mu, sd = sd_mu)
29
30     mu_samples[i] <- mu
31     tau_samples[i] <- tau
32   }
```

```

33
34   return(list(mu_samples = mu_samples, tau_samples =
              tau_samples))
35 }
36
37 samples_list <- lapply(initial_mu, function(mu)
              gibbs_sampling(mu, iterations))
38
39 mcmc_list <- lapply(samples_list, function(samples)
              mcmc(samples$mu_samples))
40
41 gelman_diag <- gelman.diag(mcmc_list[1:2])
42 f_statistics <- gelman_diag$psrf
43
44 print(f_statistics[1])
45 #The answer may be different due to randomization in
   the problem

```

R code Exa 12.5.3 Nursing Homes in New Mexico II

```

1 #Page 827
2
3 mu1 <- 183.95
4 beta1 <- 50925.37
5 alpha1 <- 11
6
7 true_posterior_mean <- mu1 + 1.645 * sqrt(beta1) * (
   gamma(alpha1 - 0.5) / gamma(alpha1))
8
9 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

```

1 #Page 830
2
3 library(coda)
4
5 set.seed(123)
6
7 x1 <- 140.7778
8 x2 <- 6
9 s11 <- 179585
10 sy <- 2.789
11 s22 <- 384
12 s12 <- 7837
13 s1y <- 3580.9
14 s2y <- 169.2
15 syy <- 78.29
16
17 initial_beta1 <- c(-0.1452, -0.1067, -0.1181,
18                   -0.1079, -0.1142)
19
20 burn_in <- 4546
21 iterations <- 10000
22
23 gibbs_sampling_beta1 <- function(initial_beta1,
24                                   iterations, burn_in) {
25   beta1_samples <- numeric(iterations)
26
27   beta1 <- initial_beta1
28   for (i in 1:(iterations + burn_in)) {
29     beta1 <- rnorm(1, mean = beta1, sd = 0.01)
30
31     if (i > burn_in) {
32       beta1_samples[i - burn_in] <- beta1
33     }
34   }
35   return(beta1_samples)
36 }

```



```

37 samples_list <- lapply(initial_beta1, function(beta1
    ) gibbs_sampling_beta1(beta1, iterations, burn_in
    ))
38
39 combined_samples <- unlist(samples_list)
40
41 interval <- quantile(combined_samples, probs = c
    (0.05, 0.95))
42
43 chain_quantiles <- lapply(samples_list, function(
    samples) quantile(samples, probs = c(0.05, 0.95))
    )
44
45 print(interval)
46
47 print(do.call(rbind, chain_quantiles))
48
49 S <- 0.01567
50 m0 <- 10000
51
52 sigma_cap <- S * sqrt(m0)
53
54 print(sigma_cap)
55 #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

```

1 #Page 831
2
3 library(ggplot2)
4 library(dplyr)
5
6 set.seed(123)

```

```

7
8 E_mu <- 147.60
9 SD_mu <- sqrt(1 / (1 * rgamma(1, 1, 0.1)))
10
11 simulated_diffs <- replicate(10000, {
12   mu <- rnorm(4, mean = E_mu, sd = SD_mu)
13   abs_diff <- abs(outer(mu, mu, "-"))
14   diag(abs_diff) <- NA
15   c(max(abs_diff, na.rm = TRUE), min(abs_diff, na.rm
      = TRUE), mean(abs_diff, na.rm = TRUE))
16 })
17
18 simulated_diffs_df <- as.data.frame(t(simulated_
  diffs))
19 names(simulated_diffs_df) <- c("max_diff", "min_diff
  ", "avg_diff")
20
21 ggplot() +
22   stat_ecdf(data = simulated_diffs_df, aes(x = max_
    diff, color = "Maximum difference"), geom = "
    step", size = 1) +
23   stat_ecdf(data = simulated_diffs_df, aes(x = min_
    diff, color = "Minimum difference"), geom = "
    step", size = 1) +
24   stat_ecdf(data = simulated_diffs_df, aes(x = avg_
    diff, color = "Average difference"), geom = "
    step", size = 1) +
25   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.") +
26   scale_color_manual(name = "Differences",
27     values = c("Maximum difference"
        = "blue", "Minimum
        difference" = "red", "
        Average difference" = "green
        ")) +
28
29   theme_minimal()

```

R code Exa 12.5.7 Calories in Hot Dogs

```
1 #Page 833
2
3 library(coda)
4 set.seed(123)
5
6 calories_beef <- c(186, 181, 176, 149, 184, 190,
7                   158, 139, 175, 148, 152, 111, 141, 153, 190, 157,
8                   131, 149, 135, 132)
9 calories_meat <- c(173, 191, 182, 190, 172, 147,
10                  146, 139, 175, 136, 179, 153, 107, 195, 135, 140,
11                  138)
12 calories_poultry <- c(129, 132, 102, 106, 94, 102,
13                     87, 99, 107, 113, 135, 142, 86, 143, 152, 146,
14                     144)
15 calories_specialty <- c(155, 170, 114, 191, 162,
16                       146, 140, 187, 180)
17
18 E_mu_beef <- mean(calories_beef)
19 SD_mu_beef <- sd(calories_beef)
20 E_mu_poultry <- mean(calories_poultry)
21 SD_mu_poultry <- sd(calories_poultry)
22
23 n_simulations <- 10000
24
25 chains <- list()
26 for (i in 1:6) {
27   Y1_chain <- rnorm(n_simulations, mean = E_mu_beef,
28                   sd = SD_mu_beef)
29   Y3_chain <- rnorm(n_simulations, mean = E_mu_
30                   poultry, sd = SD_mu_poultry)
31   D_chain <- Y1_chain - Y3_chain
32   chains[[i]] <- mcmc(D_chain)
```

```

24 }
25
26 mcmc_list <- mcmc.list(chains)
27
28 quantiles <- quantile(as.numeric(unlist(mcmc_list)),
29                        c(0.05, 0.95))
30
31 simulation_standard_error <- function(mcmc_list,
32                                       quantile_prob) {
33   Z <- sapply(mcmc_list, function(chain) quantile(
34     chain, quantile_prob))
35   S <- sqrt(var(Z))
36   return(S / sqrt(length(mcmc_list)))
37 }
38
39 se_0.05 <- simulation_standard_error(mcmc_list,
40                                     0.05)
41 se_0.95 <- simulation_standard_error(mcmc_list,
42                                     0.95)
43
44 list(
45   quantiles = quantiles,
46   se_0.05 = se_0.05,
47   se_0.95 = se_0.95
48 )
49 #The answer may be different due to randomization in
50   the problem

```

R code Exa 12.5.8 Censored Arsenic Measurements

```

1 #Page 834
2
3 set.seed(123)
4
5 k <- 10

```

```

6 iterations <- 10000
7
8 mu <- rnorm(k, mean = 0, sd = 1)
9 tau <- rgamma(k, shape = 1, rate = 1)
10
11 log_Y_samples <- numeric(iterations * k)
12
13 for (i in 1:k) {
14   for (j in 1:iterations) {
15     Z <- rnorm(1)
16     log_Y_samples[(i - 1) * iterations + j] <- mu[i]
17       + Z * sqrt(1 / tau[i])
18   }
19 }
20 Y_samples <- exp(log_Y_samples)
21
22 hist(log_Y_samples, breaks = 23, border = "black",
23      main = "Histogram of Simulated log(Y(i)) Values",
24      xlab = "log(Y(i))", ylab = "Count",
25      xlim = c(-10,10), ylim=c(0,25000))

```

R code Exa 12.6.3 The MSE of the Sample Median

```

1 #Page 841
2
3 library(ggplot2)
4
5 yi <- c(-2.46, -2.11, -1.23, -0.99, -0.42, -0.39,
6        -0.21, -0.15, -0.10, -0.07,
7        -0.02, 0.27, 0.40, 0.42, 0.44, 0.70, 0.81,
8        0.88, 1.07, 1.39,
9        1.40, 1.47, 1.62, 1.64, 1.76)
10 theta_hat <- 0.40

```

```

9
10 v <- 10000
11
12 bootstrap_median <- function(data, n) {
13   sample_medians <- numeric(n)
14   for (i in 1:n) {
15     bootstrap_sample <- sample(data, length(data),
16                               replace = TRUE)
17     sample_medians[i] <- median(bootstrap_sample)
18   }
19   return(sample_medians)
20 }
21 bootstrap_medians <- bootstrap_median(yi, v)
22
23 squared_errors <- (bootstrap_medians - theta_hat)^2
24
25 mse <- mean(squared_errors)
26
27 sample_variance <- var(squared_errors)
28
29 simulation_se <- sqrt(sample_variance / v)
30
31 df <- data.frame(medians = bootstrap_medians)
32 ggplot(df, aes(x = medians)) +
33   geom_histogram(binwidth = 0.05, fill = "grey") +
34   labs(
35     title = "Histogram of Bootstrap Sample Medians",
36     x = "Sample Median",
37     y = "Frequency"
38   ) +
39   theme_minimal()
40
41 cat("M.S.E. of the sample median:", signif(mse,3), "\n")
42 cat("Sample variance of the squared errors:", signif(
43   sample_variance,3), "\n")
43 cat("Simulation standard error:", format(simulation_

```

```
se,scientific = TRUE,digits = 4), "\n")
44 #The answer may be different due to randomization in
    the problem
```

R code Exa 12.6.4 Confidence Interval for the Interquartile Range

```
1 #Page 843
2
3 library(boot)
4
5 set.seed(123)
6
7 calc_iqr <- function(data, indices) {
8   sample_data <- data[indices]
9   return(IQR(sample_data))
10 }
11
12 yi <- c(-2.46, -2.11, -1.23, -0.99, -0.42, -0.39,
13        -0.21, -0.15, -0.10, -0.07, -0.02,
14        0.27, 0.40, 0.42, 0.44, 0.70, 0.81, 0.88,
15        1.07, 1.39, 1.40, 1.47, 1.62, 1.64, 1.76)
16
17 IQR_Fn <- 1.46
18
19 bootstrap_results <- boot(data = yi, statistic =
20   calc_iqr, R = 10000)
21
22 T_values <- bootstrap_results$t / IQR_Fn
23
24 T_values_sorted <- sort(T_values)
25 T_500 <- T_values_sorted[500]
26 T_9500 <- T_values_sorted[9500]
27
28 CI_lower <- IQR_Fn / T_9500
29 CI_upper <- IQR_Fn / T_500
```

```

27 CI <- c(CI_lower, CI_upper)
28
29 round(CI,4)
30 #The answer may be different due to randomization in
    the problem

```

R code Exa 12.6.6 Percentile t Confidence Interval for a Median

```

1 #Page 844
2
3 M <- 1.41
4 Y <- 0.245
5 alpha <- 0.1
6
7 q_lower <- -2.133
8 q_upper <- 1.581
9
10 theta_lower <- M - q_upper * Y
11 theta_upper <- M - q_lower * Y
12
13 q_lower_percentile <- -0.32
14 q_upper_percentile <- 0.16
15
16 theta_lower_percentile <- M - q_upper_percentile
17 theta_upper_percentile <- M - q_lower_percentile
18
19 cat("Percentile-t bootstrap confidence interval:",
    round(theta_lower, 3), "-", round(theta_upper, 3)
    , "\n")
20 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

```
1 #Page 845
2
3 R <- 0.6401
4 avg_R_bootstrap <- 0.6354
5
6 bias_R <- avg_R_bootstrap - R
7
8 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

```
1 #Page 845
2
3 mu <- 4.150
4 sigma_mle <- 0.5217
5 theta_mle <- 0.5593
6 n <- 23
7 df <- n - 1
8 num_samples <- 10000
9
10 set.seed(123)
11
12 chi_squared_samples <- rchisq(num_samples, df = df)
13
14 scaled_variances <- chi_squared_samples * (sigma_mle
      ^2 / n)
15
16 bias_estimate <- mean(scaled_variances) - sigma_mle
      ^2
17
18 theta_corrected <- theta_mle - bias_estimate
19
```

```

20 simulation_se <- sd(scaled_variances) / sqrt(num_
    samples)
21
22 cat("Sample Average of the T(i) values (Bias
    Estimate): ", signif(bias_estimate,4), "\n")
23 cat("Simulation Standard Error: ", signif(simulation
    _se,4), "\n")
24 cat("Original MLE of theta: ", theta_mle, "\n")
25 cat("Bias-corrected Estimate of theta: ", signif(
    theta_corrected,4), "\n")
26 #The answer may be different due to randomization in
    the problem

```

R code Exa 12.6.9 Estimating the Standard Deviation of a Statistic

```

1 #Page 846
2
3 set.seed(123)
4
5 observed_data <- c(128, 281, 291, 238, 155, 148,
    154, 232, 316, 96, 146, 151, 100, 213, 208, 157,
    48, 217)
6 n <- length(observed_data)
7 mu_hat <- 182.17
8 sigma_hat <- 72.22
9 theta_hat <- pnorm((200 - mu_hat) / sigma_hat)
10
11 num_simulations <- 10000
12
13 calculate_theta <- function(sample) {
14     mu_sample <- mean(sample)
15     sigma_sample <- sd(sample)
16     return(pnorm((200 - mu_sample) / sigma_sample))
17 }
18

```

```

19 theta_stars <- numeric(num_simulations)
20 for (i in 1:num_simulations) {
21   sample <- rnorm(n, mean = mu_hat, sd = sigma_hat)
22   theta_stars[i] <- calculate_theta(sample)
23 }
24
25 theta_star_avg <- mean(theta_stars)
26 theta_star_sd <- sd(theta_stars)
27
28 cat("Average of simulated theta_stars:", signif(
    theta_star_avg,4), "\n")
29 cat("Standard deviation of simulated theta_stars:",
    signif(theta_star_sd,4), "\n")
30 #The answer may be different due to randomization in
    the problem

```

R code Exa 12.6.10 Comparing Means When Variances Are Unequal

```

1 #Page 847
2
3 library(boot)
4
5 v <- 10000
6 alpha0_corrected <- 0.05
7
8 set.seed(132)
9
10 bootstrap_analysis <- function(data, alpha0) {
11   bootstrap_t_statistic <- function(data, indices) {
12     t.test(data[indices], mu = 200)$statistic
13   }
14
15   bootstrap_results <- boot(data = data, statistic =
    bootstrap_t_statistic, R = v)
16   U <- bootstrap_results$t

```

```

17
18   quantile_1_minus_alpha0_2 <- quantile(U, 1 -
      alpha0/2)
19   return(list(quantile = quantile_1_minus_alpha0_2,
      bootstrap_results = bootstrap_results)) #
      Return both values
20 }
21
22 n <- 18
23 data <- rnorm(n, mean = 182.17, sd = 72.22)
24 result <- bootstrap_analysis(data, alpha0_corrected)
25 quantile_1_minus_alpha0_2 <- result$quantile
26 U <- result$bootstrap_results$t
27 simulation_se <- (2*sd(U)) / (100*sqrt(10))
28
29 cat("Bootstrap estimate (Z):", round(qt(1 - alpha0_
      corrected/2, df = 17), 2), "\n")
30 cat("Simulation standard error of Z:", signif(
      simulation_se, 2), "\n")
31 #The answer may be different due to randomization in
      the problem

```
