

R Textbook Companion for
Applied Statistics and Probability for
Engineers
by Douglas C. Montgomery and George C.
Runger¹

Created by
Shaik Sameer
B.Tech.
Information Technology
Indian Institute of Information Technology, Vadodara
Cross-Checked by
R TBC Team

May 23, 2020

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

Book Description

Title: Applied Statistics and Probability for Engineers

Author: Douglas C. Montgomery and George C. Runger

Publisher: John Wiley & Sons, USA

Edition: 6

Year: 2014

ISBN: 9781118539712

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
2 Probability	5
3 Discrete Random Variables and Probability Distributions	24
4 Continuous Random Variables and Probability Distributions	40
5 Joint Probability Distributions	53
6 Descriptive Statistics	65
7 Point Estimation of Parameters and Sampling Distributions	67
8 Statistical intervals for a single sample	71
9 Tests of Hypotheses for a Single Sample	78
10 Statistical Inference for Two Samples	91
11 Simple Linear Regression and Correlation	102
12 Multiple Linear Regression	108
13 Design and Analysis of Single Factor Experiments The Analysis of Variance	121
14 Design of Experiments with Several Factors	126

List of R Codes

Exa 2.1	Camera Flash	5
Exa 2.4	Automobile Options	6
Exa 2.5	Automobile Colors	7
Exa 2.8	Hospital Emergency Visits	7
Exa 2.9	Web site design	8
Exa 2.10	Printed circuit board	9
Exa 2.11	Hospital schedule	9
Exa 2.12	Bar code 39	9
Exa 2.13	Printed circuit board layout	10
Exa 2.14	Sampling without replacement	10
Exa 2.15	Laser diodes	11
Exa 2.16	Probabilities of Events	11
Exa 2.17	Contamination particles	12
Exa 2.18	Manufacturing Inspection	13
Exa 2.19	Semiconductor Wafers	13
Exa 2.20	Semiconductor wafers and location	14
Exa 2.22	Surface flaws and defectives	14
Exa 2.23	Tree Diagram	15
Exa 2.24	Random inspection	16
Exa 2.25	Continuation of Chapter 2 Example 24	17
Exa 2.26	Machine stages	17
Exa 2.27	Semiconductor contamination	18
Exa 2.28	Semiconductor failures	18
Exa 2.29	Sampling with replacement	19
Exa 2.30	Flaws and functions	19
Exa 2.31	Unconditional Probability	20
Exa 2.32	Series circuit	20
Exa 2.33	Semiconductor wafers	21

Exa 2.34	Parallel circuit	21
Exa 2.35	Advanced circuit	22
Exa 2.37	Medical Diagnostic	22
Exa 3.2	Camera flash tests	24
Exa 3.5	Wafer contamination	24
Exa 3.6	Digital channel	25
Exa 3.7	Cumulative distribution function	26
Exa 3.8	Sampling without replacement	26
Exa 3.9	Digital channel	27
Exa 3.10	Marketing	28
Exa 3.11	Messages	29
Exa 3.12	Digital channel continuation	29
Exa 3.14	Number of voice lines	30
Exa 3.15	Proportion of voice lines	30
Exa 3.16	Digital channel	31
Exa 3.17	Binomial coefficient	31
Exa 3.18	Organic pollution	32
Exa 3.19	Mean and Variance	32
Exa 3.20	Digital channel	33
Exa 3.21	Wafer contamination	33
Exa 3.22	Mean and standard deviation	34
Exa 3.24	Digital Channel	34
Exa 3.25	Camera flashes	35
Exa 3.26	Sampling without replacement	35
Exa 3.27	Parts from suppliers	36
Exa 3.28	Mean and variance	37
Exa 3.29	Customer Sample	37
Exa 3.31	Calculations of wire flaws	38
Exa 3.32	Magnetic storage and contamination	38
Exa 4.1	Electric Current	40
Exa 4.2	Hole diameter	40
Exa 4.5	Reaction time	41
Exa 4.6	Electric current	41
Exa 4.7	Continuation of Chapter 4 Example 1	42
Exa 4.8	Hole diameter	42
Exa 4.9	Uniform current	43
Exa 4.12	standard normal distribution	43
Exa 4.13	Normally distributed current	44

Exa 4.14	Normally distributed current	45
Exa 4.15	Signal detection	45
Exa 4.16	Shaft diameter	46
Exa 4.18	The digital communication	47
Exa 4.19	Normal Approximation to Binomial	47
Exa 4.20	Normal Approximation to Poisson	48
Exa 4.21	Computer Usage	48
Exa 4.22	Lack of memory property	49
Exa 4.23	Processor failure	49
Exa 4.24	Poisson process	50
Exa 4.25	Bearing Wear	50
Exa 4.26	Semiconductor Laser	51
Exa 4.27	Commercial development	52
Exa 4.28	Generalized beta distribution	52
Exa 5.2	Server Access Time	53
Exa 5.3	Marginal Distribution	53
Exa 5.4	Server Access Time	54
Exa 5.5	Conditional Probabilities for Mobile Response Time	54
Exa 5.6	Conditional Probability	55
Exa 5.8	Conditional Mean And Variance	55
Exa 5.9	Conditional Mean And Variance	55
Exa 5.12	Machined Dimensions	56
Exa 5.14	Component Lifetimes	57
Exa 5.15	Probability as a Ratio of Volumes	57
Exa 5.16	Marginal probability distribution	58
Exa 5.18	Layer Thickness	58
Exa 5.19	Expected Value of a Function of Two Random Variables	59
Exa 5.21	Covariance	60
Exa 5.23	Independence Implies Zero Covariance	61
Exa 5.24	Digital Channel	62
Exa 5.25	Digital Channel	62
Exa 5.31	Error Propagation	63
Exa 5.32	Linear Function of Independent Normal Random Variables	63
Exa 5.33	Beverage Volume	64
Exa 6.1	Sample mean	65
Exa 6.2	Sample variance	65
Exa 6.3	Sample variance shortcut	66

Exa 7.1	Resistors	67
Exa 7.3	Aircraft engine life	67
Exa 7.5	Thermal conductivity	68
Exa 7.9	Gamma Distribution Moment Estimators	69
Exa 7.17	Bayes Estimator for the Mean of a Normal Distribution	69
Exa 8.1	Metallic Material Transition	71
Exa 8.2	Metallic Material Transition	71
Exa 8.3	One Sided Confidence Bound	72
Exa 8.4	The Exponential Distribution	72
Exa 8.5	Mercury Contamination	73
Exa 8.6	Alloy Adhesion	73
Exa 8.7	Detergent Filling	74
Exa 8.8	Crankshaft Bearings	74
Exa 8.9	Crankshaft Bearings	75
Exa 8.10	The Agresti Coull CI on a Proportion	75
Exa 8.11	Alloy Adhesion	76
Exa 8.12	Alloy Adhesion	77
Exa 9.2	Propellant Burning Rate	78
Exa 9.3	Propellant Burning Rate Type II Error	79
Exa 9.4	Propellant Burning Rate Type II Error From OC Curve	79
Exa 9.6	Golf Club Design	80
Exa 9.7	Golf Club Design Sample Size	80
Exa 9.8	Automated Filling	81
Exa 9.9	Automated Filling Sample Size	82
Exa 9.10	Automobile Engine Controller	82
Exa 9.11	Automobile Engine Controller Type II Error	83
Exa 9.12	Printed Circuit Board Defects Poisson Distribution .	84
Exa 9.13	Power Supply Distribution Continuous Distribution . .	85
Exa 9.14	Health Insurance Plan Preference	86
Exa 9.15	Propellant Shear Strength Sign Test	87
Exa 9.16	Propellant Shear Strength Wilcoxon Signed Rank Test	88
Exa 9.17	Two one sided tests	89
Exa 10.1	Paint Drying Time	91
Exa 10.2	Paint Drying Time Sample Size from OC Curves . . .	92
Exa 10.3	Paint Drying Time Sample Size	92
Exa 10.4	Aluminum Tensile Strength	93
Exa 10.5	Yield from a Catalyst	93
Exa 10.6	Arsenic in Drinking Water	94

Exa 10.8	Yield from Catalyst Sample Size	95
Exa 10.9	Cement Hydration	96
Exa 10.10	Axial Stress	97
Exa 10.11	Shear Strength of Steel Girder	97
Exa 10.12	Parallel Park Cars	98
Exa 10.13	Semiconductor Etch Variability	99
Exa 10.15	Surface Finish for Titanium Alloy	99
Exa 10.16	St Johns Wort	100
Exa 10.17	Defective Bearings	101
Exa 11.1	Oxygen purity	102
Exa 11.2	Oxygen Purity Tests of Coefficients	102
Exa 11.3	Oxygen Purity ANOVA	103
Exa 11.4	Oxygen Purity Confidence Interval on the Slope . . .	104
Exa 11.5	Oxygen Purity Confidence Interval on the Mean Re- sponse	104
Exa 11.6	Oxygen Purity Prediction Interval	105
Exa 11.8	Wire Bond Pull Strength	106
Exa 12.1	Wire Bond Strength	108
Exa 12.2	Wire Bond Strength With Matrix Notation	109
Exa 12.3	Wire Bond Strength ANOVA	110
Exa 12.4	Wire Bond Strength Coefficient Test	110
Exa 12.5	Wire Bond Strength One Sided Coefficient Test	112
Exa 12.6	Wire Bond Strength General Regression Test	113
Exa 12.7	Wire Bond Strength Confidence Interval	113
Exa 12.8	Wire Bond Strength Confidence Interval on the Mean Response	115
Exa 12.9	Wire Bond Strength Confidence Interval	116
Exa 12.11	Wire Bond Strength Confidence Interval	117
Exa 12.12	Airplane Sidewall Panels	118
Exa 12.13	Surface Finish	119
Exa 13.1	Tensile Strength ANOVA	121
Exa 13.2	Hardwood concentration experiment	123
Exa 13.4	Textile Manufacturing	124
Exa 13.5	Fabric Strength	125
Exa 14.1	Aircraft Primer Paint	126
Exa 14.3	Epitaxial Process	128
Exa 14.4	Surface Roughness	129
Exa 14.5	Plasma Etch	130

Exa 14.6	Process Yield	130
Exa 14.8	Plasma Etch	131
Exa 15.1	Vane Opening	133
Exa 15.2	Chemical Process Concentration	134
Exa 15.3	Electrical Current	135
Exa 15.4	Ceramic Substrate	136
Exa 15.5	Printed Circuit Boards	137
Exa 15.6	Chemical Process Concentration CUSUM Tabular Cusum	137
Exa 15.7	Chemical Process Concentration EWMA	138

Chapter 2

Probability

R code Exa 2.1 Camera Flash

```
1 #Camera flash (Pg no. 18)
2 foo<-function(){
3   min_recycle_time = 1.5
4   max_recycle_time = 5
5   x<-readline(prompt="Enter 1: " )
6   if(min_recycle_time<x && x<max_recycle_time)
7   {
8     print("Outcome of Set S={low , medium ,high} is
          medium")
9     print("Outcome of Set S={yes , no} is yes")
10  } else if(min_recycle_time >= x)
11  {
12    print("Outcome of Set S={low , medium ,high} is
          low")
13    print("Outcome of Set S={yes , no} is no")
14  } else if(max_recycle_time <= x)
15  {
16    print("Outcome of Set S={low , medium ,high} is
          high")
17    print("Outcome of Set S={yes , no} is no")
18  }
```

```
19 }
20
21 foo()
```

R code Exa 2.4 Automobile Options

```
1 #Automobile options(Pg no. 19)
2 foo<-function(){
3   number_of_possible_outcomes_of_automatic_
4     transmission = 2
5   #Set S = {With an automatic transmission, Without
6     an automatic transmission}
7
8   number_of_possible_outcomes_of_sunroof = 2
9   #Set S = {With a sunroof, Without a sunroof}
10
11   number_of_possible_outcomes_of_sterio_system = 3
12   #Set S = {Sterio system 1, Sterio system 2, Sterio
13     system 3}
14
15   number_of_possible_outcomes_of_exterior_color = 4
16   #Set S = {Exterior color 1, Exterior color 2,
17     Exterior color 3, Exterior color 4}
18
19   total_no_of_outcome = number_of_possible_outcomes_
20     of_automatic_transmission*
21     number_of_possible_outcomes_of_sunroof*
22     number_of_possible_outcomes_of_sterio_system*
23     number_of_possible_outcomes_of_exterior_color
24
25   cat("The sample space contains",total_no_of_
26     outcome,"outcomes")
27 }
28
29 foo()
```

R code Exa 2.5 Automobile Colors

```
1 #Automobile Colors(Pg no. 20)
2 foo<-function(){
3   vehicle_types = 12
4
5   exterior_color_black = 2
6   #Set S = {black, red}
7
8   exterior_color_white = 4
9   #Set S = {black, white, blue, brown}
10
11  exterior_color_blue = 3
12  #Set S = {black, red, blue}
13
14  exterior_color_brown = 1
15  #Set S = {brown}
16
17  total_no_of_outcome = vehicle_types*exterior_color
18    _black+
19    vehicle_types*exterior_color_white+
20    vehicle_types*exterior_color_blue+
21    vehicle_types*exterior_color_brown
22  cat("The sample space contains",total_no_of_
23    outcome,"vehicle types")
24
25  }
```

R code Exa 2.8 Hospital Emergency Visits

```

1 #Hospital Emergency Visits(Pg no. 21)
2
3 total = c(5292,6991,5640,4329)
4 lwbs = c(195,270,246,242)
5 admitted = c(1277,1558,666,984)
6 not_admitted = c(3820,5163,4728,3103)
7 df = data.frame("total" = total, "lwbs" = lwbs, "
      admitted" = admitted, "not_admitted" = not_
      admitted)
8 # A is the event that a visit is to hospital 1
9 # B is the event that the result of the visit is
    LWBS
10 A_intersection_B = df$lwbs[1]
11 Abar = sum(df$total[2:4])
12 A_union_B = df$total[1] + sum(df$lwbs[2:4])
13
14 cat("A union B =",A_union_B)
15 cat("A intersection B =",A_intersection_B)
16 cat("Abar =",Abar)
17
18 # The answer given in textbook for A intersection B
    is wrong

```

R code Exa 2.9 Web site design

```

1 #Web site design(Pg no. 23)
2 colors = 4
3 fonts = 3
4 image_positions = 3
5
6 designs = colors*fonts*image_positions
7
8 cat(designs,"different designs are possible")

```

R code Exa 2.10 Printed circuit board

```
1 #Printed circuit board(Pg no. 24)
2
3 locations = 8
4 components = 4
5 designs = factorial(locations)/factorial(components)
6
7 cat(designs," different designs are possible")
```

R code Exa 2.11 Hospital schedule

```
1 #Hospital schedule(Pg no. 24)
2
3 knee_surgery = 3
4 hip_surgery = 2
5 sequences = factorial(knee_surgery+hip_surgery)/(
    factorial(knee_surgery)*factorial(hip_surgery))
6
7 cat(sequences," sequences are possible")
```

R code Exa 2.12 Bar code 39

```
1 #Bar code 39(Pg no. 24)
2
3 blackBars = 5
4 narrowBars = 3
5 wideBars = 2
6 bars = factorial(blackBars)/(factorial(narrowBars)
    *factorial(wideBars))
```



```

7
8 narrow_spaces = 3
9 wide_spaces = 1
10 codes = bars*(narrow_spaces+wide_spaces)
11
12 cat("The number of possible codes is",codes)

```

R code Exa 2.13 Printed circuit board layout

```

1 #Printed circuit board layout(Pg no. 25)
2
3 locations = 8
4 identical_components = 5
5 designs = factorial(locations)/(factorial(identical_
      components)*factorial(locations-identical_
      components))
6
7 cat("Number of possible designs is",designs)

```

R code Exa 2.14 Sampling without replacement

```

1 #Sampling without replacement(Pg no. 25)
2
3 size = 6
4 total_parts = 50
5 defective_parts = 3
6 non_defective_parts = 47
7
8 step_1 = choose(defective_parts,2)
9
10 remaining_parts = size - 2
11

```

```

12 step_2 = choose(non_defective_parts, remaining_parts
13 )
14 num_of_subsets = step_1*step_2
15
16 cat("the number of subsets of size 6 that contain
    exactly 2 defective parts is",num_of_subsets)
17
18 diff_subsets = choose(total_parts, size)
19
20 cat("total number of different subsets of size 6 is"
    ,diff_subsets)

```

R code Exa 2.15 Laser diodes

```

1 #Laser diodes(Pg no. 30)
2
3 num_of_diodes = 30
4 individual_probability = 0.01
5
6 total_probability = num_of_diodes*individual_
    probability
7
8 cat("Probability of E is", total_probability)

```

R code Exa 2.16 Probabilities of Events

```

1 #Probabilities of Events(Pg no. 31)
2
3 a = 0.1
4 b = 0.3
5 c = 0.5
6 d = 0.1

```

```

7
8 A = a + b
9 B = b + c + d
10 C = d
11
12 cat("P(A) =",A)
13 cat("P(B) =",B)
14 cat("P(C) =",C)
15
16 cat("P(A_bar) =",1-A)
17 cat("P(B_bar) =",1-B)
18 cat("P(C_bar) =",1-C)
19
20 cat("P(A_intersection_B) =",b)
21 cat("P(A_union_B) =",a+b+c+d)
22 cat("P(A_intersection_c) =",0)

```

R code Exa 2.17 Contamination particles

```

1 #Contamination particles(Pg no. 31)
2
3 zero_contaminant = 0.4
4 three_contaminants = 0.10
5 four_contaminants = 0.05
6 five_and_above_contaminants = 0.10
7
8 E = three_contaminants+four_contaminants+five_and_
   above_contaminants
9
10 cat("P(E) = ",E,"is the probability that a wafer
    contains three or more particles in the inspected
    location")

```

R code Exa 2.18 Manufacturing Inspection

```
1 #Refer to code EX2_14(Sampling without replacement)
2
3 #Manufacturing Inspection(Pg no. 32)
4
5 size = 6
6 non_defective_parts = 47
7 num_of_subsets = 535095
8 diff_subsets = 15890700
9
10 P1 = num_of_subsets/diff_subsets
11
12 #subset with no defective parts
13 subset2 = choose(non_defective_parts,size)
14
15 P2 = subset2/diff_subsets
16
17 cat("Probability that a sample contains exactly 2
    defective parts is",P1)
18 cat("Probability that no defective parts are
    selected is",P2)
19 #The answer might slightly vary due to rounding off
    values
```

R code Exa 2.19 Semiconductor Wafers

```
1 #install.packages("MASS")
2 library(MASS)
3
4 #Semiconductor Wafers(Pg no. 35)
5
6 total = 940
7 high = 358
8 center = 626
```

```

9 high_and_center = 112
10 high_or_center = (high+center-high_and_center)
11 ans = high_or_center/total
12 ans = fractions(ans)
13 print(ans)
14
15 #The answer might slightly vary due to reducing of
    values

```

R code Exa 2.20 Semiconductor wafers and location

```

1 #Semiconductor wafers and location(Pg no. 36)
2
3 E1 = 0.15 #probability that a wafer contains four or
    more particles
4 E2 = 0.28 #probability that a wafer was at the edge
5 E1_and_E2 = 0.04
6 E1_or_E2 = E1+E2-E1_and_E2
7
8 E3 = 0.60 #probability that a wafer contains less
    than two particles
9 E4 = 0.03 #probability that a wafer is both at the
    edge and contains more than four particles
10 E3_and_E4 = 0 #mutually exclusive
11 E3_or_E4 = E3+E4-E3_and_E4
12
13 cat("probability of E1 union E2 is",E1_or_E2)
14 cat("probability of E3 union E4 is",E3_or_E4)

```

R code Exa 2.22 Surface flaws and defectives

```

1 #Surface flaws and defectives(Pg no. 40)
2

```

```

3 f1 = 40 #total_surface flaws
4 d1 = 10 #defectiv and surface flawed
5 P1 = d1/f1
6
7 d2 = 18 #defective but not surface flawed
8 f2 = 400 - f1 #without surface flawed
9 P2 = d2/f2
10
11 cat("conditional probability P(d1|f1) is",P1)
12 cat("conditional probability P(d2|f2) is",P2)

```

R code Exa 2.23 Tree Diagram

```

1 #install.packages("MASS")
2 library(MASS)
3
4 #Tree Diagram(Pg no. 41)
5
6 D_and_F = 10/400
7 F = 40/400
8 D = 28/400
9 Dbar_and_F = 30/400
10 D_and_Fbar = 18/400
11 Dbar_and_Fbar = 342/400
12 Fbar = 1-F
13
14 D_given_F = D_and_F/F
15 D_given_F = fractions(D_given_F)
16 print("P(D|F) is")
17 print(D_given_F)
18
19 F_given_D = D_and_F/D
20 F_given_D = fractions(F_given_D)
21 print("P(F|D) is")
22 print(F_given_D)

```

```

23
24 Dbar_given_F = Dbar_and_F/F
25 Dbar_given_F = fractions(Dbar_given_F)
26 print("P(Dbar|F) is")
27 print(Dbar_given_F)
28
29 D_given_Fbar = D_and_Fbar/Fbar
30 D_given_Fbar = fractions(D_given_Fbar)
31 print("P(D|Fbar) is")
32 print(D_given_Fbar)
33
34 Dbar_given_Fbar = Dbar_and_Fbar/Fbar
35 Dbar_given_Fbar = fractions(Dbar_given_Fbar)
36 print("P(Dbar|Fbar) is")
37 print(Dbar_given_Fbar)
38
39 #The answer might slightly vary due to reducing of
    values

```

R code Exa 2.24 Random inspection

```

1 #install.packages("MASS")
2 library(MASS)
3
4 #Random inspection(Pg no. 42)
5
6 remaining_parts = 49 #total remaining parts
7 defective_parts = 2 #total defective parts remaining
8
9 P = defective_parts/remaining_parts
10 P = fractions(P)
11 print(P)
12 cat("conditional probability P(B|A) is ",P)

```

R code Exa 2.25 Continuation of Chapter 2 Example 24

```
1 #Continue to EX2_24
2
3 #(Pg no. 43)
4
5 total = 50
6 defective_parts = 3
7 non_defective_parts = 47
8
9 p = (defective_parts/total)*((defective_parts-1)/(
    total-1))*(non_defective_parts/(total-2))
10
11 cat(p,"is the probability that the first two parts
    selected are defective and the third is not
    defective")
12
13 #The answer may slightly vary due to rounding off
    values
```

R code Exa 2.26 Machine stages

```
1 #Machine stages(Pg no. 45)
2
3 A = 0.90 #probability of that first stage of
    machining meets specifications
4 B_given_A = 0.95 #Given A, probability that a second
    stage of machining meets specifications
5 A_and_B = B_given_A*A
6
7 cat(A_and_B,"is the probability that both stages
    meet specifications")
```

R code Exa 2.27 Semiconductor contamination

```
1 #Semiconductor contamination(Pg no. 46)
2
3 high = 0.20 #probability of high level of
   contamination
4 not_high = 1 - high #probability of not high level
   of contamination
5 product_fail_given_high = 0.10
6 product_fail_given_not_high = 0.005
7
8 product_fail = (product_fail_given_high*high)+(
   product_fail_given_not_high*not_high)
9
10 cat(product_fail,"is probability that the product
   fails")
```

R code Exa 2.28 Semiconductor failures

```
1 #Semiconductor failures(Pg no. 47)
2
3 high = 0.20 #probability of high level of
   contamination
4 medium = 0.30 #probability of medium level of
   contamination
5 low = 0.50 #probability of low level of
   contamination
6 product_fail_given_high = 0.10
7 product_fail_given_medium = 0.01
8 product_fail_given_low = 0.001
9
```

```

10 product_fail = (product_fail_given_high*high)+(
    product_fail_given_medium*medium)+(product_fail_
    given_low*low)
11
12 cat(product_fail,"is probability that the product
    fails")

```

R code Exa 2.29 Sampling with replacement

```

1 #install.packages("MASS")
2 library(MASS)
3
4 #Sampling with replacement(Pg no. 49)
5
6 # A : event that first part is defective
7 # B : event that second part is defective
8
9 B_given_A = 3
10 A = 3
11 total = 50
12 P = fractions((B_given_A*A)/(total*total))
13 print(P)
14 cat("probability that both parts are defective is",P
    )

```

R code Exa 2.30 Flaws and functions

```

1 #install.packages("MASS")
2 library(MASS)
3
4 #Flaws and functions(Pg no. 49)
5
6 defective_and_surface_flawed = 2

```

```

7 total_defective_parts = 20
8 total_surface_flawed = 40
9 P = fractions((defective_and_surface_flawed*
    defective_and_surface_flawed)/(total_defective_
    parts*total_surface_flawed))
10 print(P)
11 cat("probability of surface flawed and defevtive is"
    ,P)
12
13 #The answer may slightly vary due to rounding off
    values

```

R code Exa 2.31 Unconditional Probability

```

1 #Continuation of Ex2_14
2
3 #(Pg no. 50)
4
5 B_given_A = 2/49
6 A = 3/50
7 B_given_Abar = 3/49
8 Abar = 1-A
9
10 B = (B_given_A*A)+(B_given_Abar*Abar)
11 cat("P(B) =",B)

```

R code Exa 2.32 Series circuit

```

1 #Series circuit(Pg no. 51)
2
3 l = 0.80 #probability that left device operate
4 r = 0.90 #probability that right device operate
5

```

```

6 l_and_r = l*r
7
8 cat("The probability that the circuit operates is",l
    _and_r)

```

R code Exa 2.33 Semiconductor wafers

```

1 #Semiconductor wafers(Pg no. 51)
2
3 contamination = 0.01
4 no_contamination = 1- contamination
5 wafers = 15
6
7 total_probability = no_contamination^wafers
8
9 cat("The probability that no large particles are
    found is ",total_probability)
10
11 #The answer may slightly vary due rounding off
    values

```

R code Exa 2.34 Parallel circuit

```

1 #Parallel circuit(Pg no. 51)
2
3 top = 0.95 #top device does operate
4 bottom = 0.95 #bottom device does operate
5
6 t = 1 - top #top device does not operate
7 b = 1 - bottom #bottom device does not operate
8
9 t_and_b = t*b
10

```

```

11 top_or_bottom = 1 - t_and_b
12
13 cat("The probability that the circuit operates is",
      top_or_bottom)

```

R code Exa 2.35 Advanced circuit

```

1 #Advanced circuit (Pg no. 52)
2
3 left = 1 - 0.1^3
4 middle = 1 - 0.05^2
5 right = 0.99
6
7 operates = left*middle*right
8
9 cat(operates,"is the probability that the circuit
      operates")
10 #The answer may slightly vary due to rounding off
    values

```

R code Exa 2.37 Medical Diagnostic

```

1 #Medical Diagnostic (Pg no. 55)
2
3 p_given_i = 0.99 #test signals positive with illness
4 n_given_i = 0.95 #test signals negative with illness
5 i = 0.0001 #illness in population
6
7 p_given_not_i = 0.05
8
9 i_given_p = (p_given_i*i)/((p_given_i*i)+(p_given_
      not_i*(1-i)))
10

```

```
11 cat(i_given_p,"is the probability that you have the  
    illness given test is positive")  
12 #The answer may slightly vary due to rounding off  
    values
```

Chapter 3

Discrete Random Variables and Probability Distributions

R code Exa 3.2 Camera flash tests

```
1 #Camera flash tests(Pg no. 66)
2
3 pass = 0.8
4 fail = 1 - pass
5
6 ppf = pass*pass*fail
7
8 cat("the probability that the first and second
    cameras pass the test and third one fails is",ppf
    )
```

R code Exa 3.5 Wafer contamination

```
1 #Wafer contamination(Pg no. 68)
2 foo = function()
3 {
```

```

4      p = 0.01 #contaminent present
5      a = 1 - p #contaminent absent
6      x = readline(prompt = "Enter x:")
7      x = as.integer(x)
8      if (x>=1)
9      {
10         px = (a^(x-1))*p
11         cat("probability P( X =",x," ) is ",px)
12     }
13     else
14     {
15         cat("Error! input must be >=1")
16     }
17
18 }
19
20 foo()

```

R code Exa 3.6 Digital channel

```

1 #Digital channel(Pg no. 71)
2
3 #probability of error in given number(X) of bits
4 zero = 0.6561 # P(X=0)
5 one = 0.2916 # P(X=1)
6 two = 0.0486 # P(X=2)
7 three = 0.0036 # P(X=3)
8
9
10 X = zero + one + two + three
11 Y = zero + one + two
12
13 cat("P(X<=3) =",X)
14 cat("P(X=3) =",X-Y)

```

R code Exa 3.7 Cumulative distribution function

```
1 #Cumulative distribution function(Pg no. 72)
2
3 "
4  $F(x) = 0 \text{ --- } \{ x < -2\}$ 
5  $F(x) = 0.2 \text{ --- } \{ -2 \leq x < 0\}$ 
6  $F(x) = 0.7 \text{ --- } \{ 0 \leq x < 2\}$ 
7  $F(x) = 1 \text{ --- } \{ 2 \leq x\}$ 
8
9  $f(-2) = ?$ 
10  $f(0) = ?$ 
11  $f(2) = ?$ 
12 "
13 P1 = 0.2 - 0.0
14 P2 = 0.7 - 0.2
15 P3 = 1.0 - 0.7
16
17 cat("f(-2) =", P1)
18 cat("f(0) =", P2)
19 cat("f(2) =", P3)
```

R code Exa 3.8 Sampling without replacement

```
1 #Sampling without replacement(Pg no. 72)
2
3 total = 850
4 unconforming = 50
5 conforming = total - unconforming
6
7 X0 = (conforming/total)*((conforming-1)/(total-1))
8 X1 = 2*(conforming/total)*(unconforming/(total-1))
```

```

9  X2 = (unconforming/total)*((unconforming-1)/(total
    -1))
10
11  F0 = round(X0, digits = 3)
12  F0
13  F1 = round(X0 + X1, digits = 3)
14  F1
15  F2 = X0 + X1 + X2
16  F2
17  cat(
18  "
19  F(x) = 0 --- { x < 0}
20  F(x) = ",F0," --- { 0 <= x < 1}
21  F(x) = ",F1," --- { 1 <= x < 2}
22  F(x) = ",F2," --- { 2 <= x}
23
24  ")
25  #The values may slightly vary due to rounding off
    values

```

R code Exa 3.9 Digital channel

```

1  #Digital channel(Pg no. 75)
2
3  error_bits = c(0,1,2,3,4)
4  probability = c(0.656,0.2916,0.0486,0.0036,0.0001)
5  df = data.frame('error_bits' = error_bits, '
    probability' = probability)
6
7  index = c(1:length(error_bits))
8
9  u = weighted.mean(error_bits, probability) #
    expectation
10
11  v = weighted.mean((error_bits-u)^2,probability) #

```

```

        variance
12
13
14 cat(" Expectation E(X) =",round(u, digits = 1))
15 cat(" Variance V(X) =",v)
16 #The answer may slightly vary due to rounding off
    values

```

R code Exa 3.10 Marketing

```

1 #Marketing(Pg no. 75)
2
3 B_revenue1 = 7 # in million dollars
4 B_probability1 = 0.3
5
6 B_revenue2 = 2 # in million dollars
7 B_probability2 = 0.7
8
9 u = B_revenue1*B_probability1 + B_revenue2*B_
    probability2
10 #expectation in million dollars
11
12 v = ((B_revenue1-u)^2)*B_probability1 + ((B_revenue2
    -u)^2)*B_probability2
13 #variance in million dollars squared
14
15 sd = sqrt(v)
16
17 cat(" Expectation E(Y) =",u)
18 cat(" Variance V(Y) =",v)
19 cat(" Standard Deviation SD(Y) =",sd)
20 #The answer may slightly vary due to rounding off
    values

```

R code Exa 3.11 Messages

```
1 #Messages(Pg no. 76)
2
3 x = c(10,11,12,13,14,15) # number of messages
4 probability = c(0.08,0.15,0.30,0.20,0.20,0.07)
5
6 u = weighted.mean(x,probability) #expectation
7
8 v = weighted.mean((x-u)^2,probability)#variance
9
10 sd = sqrt(v)
11 cat(" Expectation E(X) =",u)
12 cat(" Variance V(X) =",v)
13 cat(" Standard Deviation SD(X) =",sd)
14 #The answer may slightly vary due to rounding off
    values
```

R code Exa 3.12 Digital channel continuation

```
1 #Digital channel continuation(Pg no. 76)
2
3 error_bits = list(0,1,2,3,4)
4 probability = list
    (0.656,0.2916,0.0486,0.0036,0.0001)
5
6 index = c(1:length(error_bits))
7 u = 0.00 #expectation
8 for (i in index) {
9     u = u + (((as.double(error_bits[i]))^2)*as.double(
        probability[i]))
10 }
```

```
11
12 cat(" Expectation E[h(X)] =",u)
```

R code Exa 3.14 Number of voice lines

```
1 #Number of voice lines (Pg no. 78)
2
3 max_lines = 48
4 min_lines = 0
5
6 u = (max_lines+min_lines)/2 #expectation
7
8 sd = sqrt((((max_lines-min_lines+1)^2)-1)/12) #
   standard deviation
9
10 cat(" Expectation E(X) =",u)
11 cat(" Standard Deviation SD(X) =",sd)
12 #The answer may slightly vary due to rounding off
   values
```

R code Exa 3.15 Proportion of voice lines

```
1 #Refer to EX3_14(Number of voice lines)
2
3 #Proportion of voice lines (Pg no. 79)
4
5 u = 24 #expectation E(X)
6 v = 199.94 #variance Var(X)
7
8 u1 = u/48 #expectation E(Y)
9 v1 = v/(48^2) #variance Var(Y)
10
11 cat(" Expectation E(Y) =",u1)
```

```
12 cat(" Variance   Var(Y) =",v1)
13 #The answer may slightly vary due to rounding off
    values
```

R code Exa 3.16 Digital channel

```
1 #install.packages("combinat")
2 library(combinat)
3
4 #Digital channel(Pg no. 80)
5
6 error = 0.1
7 not_error = 1 - error
8
9 P = error*not_error*error*not_error #probability of
    2 errors
10
11 outcomes = dim(combn(4,2))[2]
12
13 total_probability = outcomes*P
14
15 cat(" Probability P(X=2) =",total_probability)
```

R code Exa 3.17 Binomial coefficient

```
1 #install.packages("combinat")
2 library(combinat)
3
4 #Binomial coefficient(Pg no. 82)
5
6
7 A = dim(combn(10,3))[2]
8 A
```

```

9
10 B = dim(combn(15,10))[2]
11 B
12
13 C = dim(combn(100,4))[2]
14 C

```

R code Exa 3.18 Organic pollution

```

1 #Organic pollution(Pg no. 83)
2
3 n = 18 #number of samples
4 p = 0.1
5
6 # P(X=2)
7 A = dbinom(2,n,p)
8 A
9
10 # P(X>=4) == 1 - P(X<4)
11 B = 1 - pbinom(3,n,p)
12 B
13
14 # P(3<=X && X<7)
15 C = dbinom(3,n,p) + dbinom(4,n,p) + dbinom(5,n,p) +
      dbinom(6,n,p)
16 C

```

R code Exa 3.19 Mean and Variance

```

1 #Mean and Variance(Pg no. 84)
2
3 n = 4
4 p = 0.1

```

```

5
6 u = n*p
7
8 v = n*p*(1-p)
9
10 cat("E(X) =",u)
11 cat("Var(X) =",v)

```

R code Exa 3.20 Digital channel

```

1 #Digital channel(Pg no. 87)
2
3 error = 0.1
4 not_error = 1 - error
5
6 P = (not_error^4)*error
7
8 cat("P(X=5) =",P)
9 #The answer may slightly vary due to rounding off
  values

```

R code Exa 3.21 Wafer contamination

```

1 #Wafer contamination(Pg no. 88)
2
3 contaminated = 0.01
4
5 n = 125 #num of wafers
6
7 p = dgeom(n,contaminated)
8
9 cat("P(X=125) =",p)

```



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

R code Exa 3.22 Mean and standard deviation

```
1 #Mean and standard deviation(Pg no. 88)
2
3 p = 0.1
4
5 u = 1/p
6
7 sd = sqrt((1-p)/(p^2))
8
9 cat("The mean until the first error is",u)
10 cat("The standard deviation before the first error
    is",sd)
11 #The answer may slightly vary due to rounding off
    values
```

R code Exa 3.24 Digital Channel

```
1 #Digital Channel(Pg no. 89)
2
3 error_prob = 0.1
4 n = 10
5 n_error = 3
6 ans = dnbinom((n-n_error-1),(n_error+1),error_prob)
7
8 cat("Answer is",ans)
```

R code Exa 3.25 Camera flashes

```
1 #Camera flashes (Pg no. 91)
2
3 p = 0.2
4 r = 3
5 x = 5
6
7 #P(X<=5)
8 B = pnbinom((x-r),r,p)
9
10 cat(B,"is the probability that the third failure is
      obtained in five or fewer tests")
11 #The answer may slightly vary due to rounding off
    values
```

R code Exa 3.26 Sampling without replacement

```
1 #Sampling without replacement (Pg no. 93)
2
3 # P(X=0)
4 A = (choose(50,0)*choose(800,2))/choose(850,2)
5 A
6
7
8 # P(X=1)
9 B = (choose(50,1)*choose(800,1))/choose(850,2)
10 B
11
12 # P(X=2)
13 C = (choose(50,2)*choose(800,0))/choose(850,2)
14 C
15
16 #The answer may slightly vary due to rounding off
    values
```

R code Exa 3.27 Parts from suppliers

```
1 #Parts from suppliers(Pg no. 94)
2
3 supplier1 = 100 #parts from local supplier
4 supplier2 = 200 #parts from distant supplier
5
6 #P(X=4)
7 A = (choose(supplier1,4)*choose(supplier2,0))/choose
      (supplier1+supplier2,4)
8
9 #P(X>=2)
10 B = 0.0
11 for (i in 2:4)
12 {
13   B = B + (choose(supplier1,i)*choose(supplier2,4-i)
              )/choose(supplier1+supplier2,4)
14 }
15
16 #P(X>=1) == 1 - P(X=0)
17 temp = ((choose(supplier1,0)*choose(supplier2,4))/
           choose(supplier1+supplier2,4))
18 C = 1 - temp
19
20 cat(A,"is the probability that all 4 parts are from
      the local supplier")
21 cat(B,"is the probability that two or more parts in
      the sample are from the local supplier")
22 cat(C,"is the probability that at least one part in
      the sample is from the local supplier")
23
24 #The answer may slightly vary due to rounding off
    values
```

R code Exa 3.28 Mean and variance

```
1 #Mean and variance(Pg no. 95)
2
3 size = 4
4 p = 100/300
5
6 u = size*p #expectation
7
8 v = size*(1/3)*(2/3)*((300-4)/299) #variance
9
10 cat("E(X) =",u)
11 cat("Var(X) =",v)
12
13 #The answer may slightly vary due to rounding off
    values
```

R code Exa 3.29 Customer Sample

```
1 #Customer Sample(Pg no. 96)
2
3 p = 0.7 #P(A)
4 n = 50 #no.of sample customers
5
6 #P(X>45)
7 B = 1 - pbinom(45,50,p)
8
9 cat(B,"is the probability that X>45 have purchased
    from the corporation in the last three months")
10 #The answer may slightly vary due to rounding off
    values
```

R code Exa 3.31 Calculations of wire flaws

```
1 #Calculations of wire flaws(Pg no. 99)
2
3 mean = 2.3 #flaws per mm(millimeter)
4 t = 1 #mm
5 #P(X=2)
6 A = dpois(2,lambda = mean*t)
7
8 t = 5 #mm
9 #P(X=10)
10 B = dpois(10, lambda = mean*t)
11
12 t = 2 #mm
13 #P(X>=1) == 1 - P(X=0)
14 C = 1 - dpois(0,lambda = mean*t)
15
16 cat("the probability of two flaws in 1 millimeter of
      wire is",A)
17 cat("the probability of 10 flaws in 5 millimeters of
      wire is",B)
18 cat("the probability of at least one flaw in 2
      millimeters of wire is",C)
19 #The answer may slightly vary due to rounding off
    values
```

R code Exa 3.32 Magnetic storage and contamination

```
1 #Magnetic storage and contamination(Pg no. 101)
2
3 mean = 0.1 #particles per cm(centimeter)^2
4 t = 100 #cm^2
```

```
5
6 #P(X=12)
7 A = dpois(12, lambda = mean*t)
8
9 #P(X=0)
10 B = dpois(0, lambda = mean*t)
11
12 #P(X<=12)
13 C = ppois(12, lambda = mean*t)
14
15 cat("P(X=12) =",A)
16 cat("P(X=0) =",B)
17 cat("P(X<=12) =",C)
18 #The answer may slightly vary due to rounding off
    values
```

Chapter 4

Continuous Random Variables and Probability Distributions

R code Exa 4.1 Electric Current

```
1 #Electric Current(Pg no. 110)
2
3 #P(X<5)
4 integrand <- function(x) {5}
5 q = integrate(Vectorize(integrand), lower = 4.9,
6               upper = 5)
7
8 #P(4.95<X<5.1)
9 p = integrate(Vectorize(integrand), lower = 4.95,
10               upper = 5.1)
11 cat("P(4.95<X<5.1) =", p$value)
```

R code Exa 4.2 Hole diameter

```
1 #Hole diameter(Pg no. 110)
```

```

2
3 # P {X>12.60}
4 integrand <- function(x) {20*(exp(-20*(x-12.5)))}
5 P = integrate(integrand, lower = 12.6, upper = Inf)
6 P
7
8 # P {12.50<X<12.60}
9 integrand <- function(x) {20*(exp(-20*(x-12.5)))}
10 Q = integrate(integrand, lower = 12.5, upper = 12.6)
11 Q
12
13 #The answer may slightly vary due to rounding off
    values

```

R code Exa 4.5 Reaction time

```

1 #install.packages("Deriv")
2 library(Deriv)
3
4 #Reaction time(Pg no. 113)
5
6 #P(X<200)
7 f = function(x){1-(exp(-(0.01*x)))}
8 f(200)
9 #The answer may slightly vary due to rounding off
    values

```

R code Exa 4.6 Electric current

```

1 #Electric current(Pg no. 115)
2
3 # Expectation
4 integrand <- function(x) {5*x}

```



```

5 P = integrate(integrand, lower = 4.9, upper = 5.1)
6 P
7
8 # Variance
9 integrand <- function(x) {5*(x-10)^2}
10 Q = integrate(integrand, lower = 4.9, upper = 5.1)
11 Q
12
13 #The answer provided in textbook is wrong

```

R code Exa 4.7 Continuation of Chapter 4 Example 1

```

1 #Continuation of EX4_1
2
3 #(Pg no. 115)
4
5 integrand <- function(x) {0.0001*(x^2)}
6 P = integrate(integrand, lower = 4.9, upper = 5.1)
7 P #watts

```

R code Exa 4.8 Hole diameter

```

1 #Hole diameter(Pg no. 115)
2
3 # Expectation
4 integrand <- function(x) {x*20*(exp(-20*(x-12.5)))}
5 u = integrate(integrand, lower = 12.5, upper = Inf)
6 u
7
8 # Variance
9 integrand <- function(x) {((x - 12.55)^2)*20*(exp
  (-20*(x-12.5)))}
10 v = integrate(integrand, lower = 12.5, upper = Inf)

```

```

11 v
12
13 sd = sqrt(v$value) #Standard deviation
14 sd
15 #The answer may slightly vary due to rounding off
    values

```

R code Exa 4.9 Uniform current

```

1 #Uniform current(Pg no. 117)
2
3 # P {4.95<X<5}
4 integrand <- function(x) {5}
5 P = integrate(Vectorize(integrand),lower = 4.95,
    upper = 5.0)
6 P
7
8 a=4.9
9 b=5.1
10
11 u = (a+b)/2 # expectation in mA
12 u
13 v = ((0.2)^2)/12 #variance in mA^2
14 v
15 sd = sqrt(v) #standard deviation in mA
16 sd
17
18 #The answer may slightly vary due to rounding off
    values

```

R code Exa 4.12 standard normal distribution

```

1 options(scipen = 999) #for disabling scientific
  notation
2
3 #(Pg no. 121)
4
5 #P(X > 1.26)
6 p1 = 1 - pnorm(1.26,0,1)
7 cat("P(X>1.26) =",p1)
8
9 #P(X < 20.86)
10 p1 = pnorm(20.86,0,1)
11 cat("P(X<20.86) =",p1)
12
13 #P(X > -1.37) = P(X < 1.37)
14 p1 = pnorm(1.37,0,1)
15 cat("P(X>-1.37) =",p1)
16
17 #P(-1.25 < X < 0.37)
18 pmin = pnorm(0.37,0,1)
19 pmax = pnorm(-1.25,0,1)
20 cat("P(-1.25 < Z < 0.37) =",pmin-pmax)
21
22 #P(X <= -4.6)
23 p = pnorm(-3.99,0,1)
24 cat("Since P(X<=-3.99) =",round(p, digits = 6),"and
      P(X<=      4.6) < P(X<=      3.99) P(X <= -4.6)
      almost equal to zero")
25 #the answer given in textbook is wrong for P(X <
    20.86)

```

R code Exa 4.13 Normally distributed current

```

1 #Normally distributed current (Pg no. 123)
2
3 mean = 10 #mA

```

```

4 v = 4 #mA^2
5 x = 13 #mA
6
7 p = 1 - pnorm(x, mean ,sqrt(v))
8 p
9
10 #The answer may slightly vary due to rounding off
    values

```

R code Exa 4.14 Normally distributed current

```

1 #Normally distributed current(Pg no. 124)
2
3 mean = 10 #mA
4 v = 4 #mA^2
5
6 #P(9<X<11)
7 p1 = 1 - pnorm(9, mean ,sqrt(v))
8 p1
9
10 p2 = 1 - pnorm(11, mean , sqrt(v))
11 p2
12
13 result = p1- p2
14 result
15
16 z =2.06
17 x = z*sqrt(v) + mean
18 x #mA
19 #The answer may slightly vary due to rounding off
    values

```

R code Exa 4.15 Signal detection

```

1 #Signal detection(Pg no. 124)
2
3 result = 1 - pnorm(0.9, 0, 0.45)
4 result
5
6 cat(result,"is the probability of detecting a
    digital 1 when none was sent")

```

R code Exa 4.16 Shaft diameter

```

1 #Shaft diameter(Pg no. 125)
2
3 mean = 0.2508 #inches
4 sd = 0.0005 #inches
5
6 # P(0.2485<X<0.2515)
7 p1 = 1 - pnorm(0.2485, mean ,sd)
8
9 p2 = 1 - pnorm(0.2515, mean, sd)
10
11 result = p1 - p2
12
13 cat(result,"proportion of shafts conforms to
    specifications")
14
15 new_mean = 0.2500
16
17 p3 = 1 - pnorm(0.2485, new_mean ,sd)
18
19 p4 = 1 - pnorm(0.2515, new_mean, sd)
20
21 result1 = p3 - p4
22 cat(result1,"proportion of shafts conforms to
    specifications")

```

R code Exa 4.18 The digital communication

```
1 #The digital communication(Pg no. 130)
2
3 np = (16*10^6)*(1*10^-5)
4
5 #P(X <= 150.5)
6 z = (150.5-np)/sqrt(np*(1-10^-5))
7 p = pnorm(z,0,1)
8 cat("P(X <= 150.5) =",p)
```

R code Exa 4.19 Normal Approximation to Binomial

```
1 #Normal Approximation to Binomial(Pg no. 130)
2
3 n = 50
4 p = 0.1
5
6 #P(X<=2)
7 x = pbinom(2,n,p)
8 cat("P(X<=2) =",round(x, digits = 3))
9
10 p1 = pnorm(2.5, 5, (sqrt(n*p*(1-p))))
11 cat("Based on the normal approximation P(X<=2) =",p1
    )
12
13 #P(9<=X)
14 p2 = pnorm(8.5, 5, 2.12, lower.tail = FALSE)
15 cat("P(9<=X) =",round(p2, digits = 2))
```

R code Exa 4.20 Normal Approximation to Poisson

```
1 #Normal Approximation to Poisson(Pg no. 131)
2
3 #P(X <= 950.5)
4 z = (950.5-1000)/sqrt(1000)
5 p = pnorm(z,0,1)
6 cat("P(X <= 950.5) =",p)
```

R code Exa 4.21 Computer Usage

```
1 #Computer Usage(Pg no. 134)
2
3 mean = 25 #log-on per hour
4 x = 0.1 #6 min = 0.1 hr
5
6 #P(X>0.1)
7 p1 = pexp(x,mean,lower.tail = FALSE)
8
9 #P(0.033<X<0.05)
10 p2 = (pexp(0.033,25,lower.tail = FALSE) + pexp
      (0.05,25,lower.tail = TRUE))-1
11
12 cat(round(p1,digits = 3),"is the probability that
      there are no log-ons in an interval of six
      minutes")
13 cat(round(p2, digits = 3),"is the probability that
      the time until the next log-on is between two and
      three minutes")
14
15 #P(X > x) = 0.90
16 x = log(0.90)/(-mean)
17 cat("the length of time x such that P ( X > x ) = 0
      . 90 is",round(x*60, digits = 2),"minutes")
18 cat("the mean time until the next log-on is", (1/
```

```

    mean)*60,"minutes")
19 cat("The standard deviation of the time until the
    next log-on is",(1/mean)*60,"minutes")

```

R code Exa 4.22 Lack of memory property

```

1 #Lack of memory property(Pg no. 135)
2
3 u = 1.4 #expectation in min
4 #P(X<0.5min)
5 f = 1 - exp(-(0.5/1.4))
6 cat("probability that we detect a particle within 30
    seconds of starting the counter is",f)
7
8 #P(3< X <3.5)
9 f1 = (1 - exp(-(3.5/1.4))) - (1 - exp(-(3/1.4)))
10
11 #P(X >3)
12 f2 = 1 - (1 - exp(-(3/1.4)))
13
14 #P(X <3.5 | X >3) = P(3< X <3.5) / P(X >3)
15 result = f1/f2
16 round(result,digits = 2)

```

R code Exa 4.23 Processor failure

```

1 #Processor failure(Pg no. 139)
2
3 mean = 0.0001 #failure per hr
4 failures = 40000
5
6 u = mean * failures #expectation
7

```



```

8 #P(X > 40,000) = P(N <= 3)
9 p = ppois(3, lambda = u)
10 round(p,digits = 3)

```

R code Exa 4.24 Poisson process

```

1
2 #(Pg no. 141)
3
4 #P(X>25)
5 p = ppois(9,lambda = 12.5)
6 round(p, digits = 3)
7
8 r= 10
9 lambda = 0.5
10
11 mean = r/lambda
12 mean
13
14 variance = r/(lambda^2)
15 variance
16
17 sd = variance^0.5
18 round(sd, digits = 2)
19
20 #P(X <= x) = 0.95
21 x = qgamma(p = 0.95, shape = r, scale = 1/lambda)
22 cat("A schedule that allows '",x," hours to prepare
    10 slides should be met 95% of the time")

```

R code Exa 4.25 Bearing Wear

```

1 #Bearing Wear(Pg no. 144)

```

```

2
3 mean = 5000*0.5*sqrt(pi)
4 mean
5
6 #P(X>6000)
7 p = pweibull(q = 6000, shape = 2, scale = 5000, lower.tail = FALSE)
8
9 cat("only ", round(p*100, digits = 1), "% of all
    bearings last at least 6000 hours")

```

R code Exa 4.26 Semiconductor Laser

```

1 #Semiconductor Laser (Pg no. 147)
2
3 o = 10
4 w = 1.5
5
6 #P(X > 10000)
7 p = 1 - plnorm(10000, meanlog = o, sdlog = w)
8 cat("P(X > 10000) =", p)
9
10 x = qlnorm(0.99, o, w, lower.tail = FALSE)
11 cat("x =", round(x, digits = 2), "hours")
12
13 E_x = exp(o + (w^2/2))
14 cat("mean =", round(E_x, digits = 1), "hours")
15
16 V_x = (exp((2*o) + (w^2))) * (exp((w^2)) - 1)
17 cat("standard deviation =", sqrt(V_x), "hours")
18
19 #The answer may slightly vary due to rounding off
    values

```

R code Exa 4.27 Commercial development

```
1
2 #(Pg no. 149)
3
4 alpha = 2.5
5 beta = 1
6
7
8 #P(X > 0.7)
9 p = pbeta(q = 0.7, shape1 = alpha, shape2 = beta,
10          lower.tail = FALSE)
11 cat("P(X > 0.7) =", round(p, digits = 2))
```

R code Exa 4.28 Generalized beta distribution

```
1
2 #(Pg no. 150)
3
4 a = 8
5 b = 20
6 m = 16
7
8 mean = (a+(4*m)+b)/6
9 cat("mean =", round(mean, digits = 3))
10 alpha = ((mean-a)*((2*m)-a-b))/((m-mean)*(b-a))
11 cat("alpha =", round(alpha, digits = 3))
12 beta = (alpha*(b-mean))/(mean-a)
13 cat("beta =", round(beta, digits = 3))
```

Chapter 5

Joint Probability Distributions

R code Exa 5.2 Server Access Time

```
1 #Server Access Time(Pg no. 158)
2
3 fnc = function(x) {(exp((- (0.002*x) - (0.001*x)))) /
  (0.002)}
4
5 an1 = integrate(fnc, 0, Inf)$value
6 an1 = an1*6*10^(-6)
7 an1
```

R code Exa 5.3 Marginal Distribution

```
1 #Marginal Distribution(Pg no. 159)
2
3 P3_1 = 0.25
4 P3_2 = 0.2
5 P3_3 = 0.05
6 P3_4 = 0.05
7
```

```

8 f3 = P3_1 + P3_2 + P3_3 + P3_4
9 f3

```

R code Exa 5.4 Server Access Time

```

1 #Server Access Time(Pg no. 160)
2
3 #P(Y>2000)
4 f <- function(y)
5 {
6   (exp(-(0.002*y)))*(1-exp(-(0.001*y)))
7 }
8 v = integrate(f, lower = 2000, upper = Inf)
9 ans = (6*10^-3)*(v$value)
10 cat("P(Y>2000) =",round(ans,digits = 2))

```

R code Exa 5.5 Conditional Probabilities for Mobile Response Time

```

1 #Conditional Probabilities for Mobile Response Time(
   Pg no. 162)
2
3 P3_1 = 0.25
4 P3_2 = 0.2
5 P3_3 = 0.05
6 P3_4 = 0.05
7
8 f3 = P3_1 + P3_2 + P3_3 + P3_4
9
10 cat("P(Y=1|X=3) =",round((P3_1/f3),digits = 3))
11 cat("P(Y=2|X=3) =",round((P3_2/f3),digits = 3))
12 cat("P(Y=3|X=3) =",round((P3_3/f3),digits = 3))
13 cat("P(Y=4|X=3) =",round((P3_4/f3),digits = 3))
14

```

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

R code Exa 5.6 Conditional Probability

```
1 #Conditional Probability(Pg no. 163)
2
3 #P(Y>2000|X=1500)
4 f <- function(y)
5 {
6   0.002*(exp((0.002*1500)-(0.002*y)))
7 }
8 v = integrate(f, lower = 2000, upper = Inf)
9 ans = v$value
10 cat("P(Y>2000|X=1500) =",round(ans,digits = 3))
```

R code Exa 5.8 Conditional Mean And Variance

```
1 #Conditional Mean And Variance(Pg no. 164)
2
3 #E(Y|X=1500)
4 f <- function(y)
5 {
6   y*(0.002*(exp((0.002*1500)-(0.002*y))))
7 }
8 v = integrate(f, lower = 1500, upper = Inf)
9 ans = v$value #ms
10 cat("E(Y|X=1500) =",ans)
```

R code Exa 5.9 Conditional Mean And Variance

```

1 #(Pg no. 164)
2 y = c(1,2,3,4)
3 prob = c(0.05,0.1,0.1,0.75)
4 u = weighted.mean(y,prob)
5 cat("E(Y|1) =",u)
6
7 v = weighted.mean((y-u)^2, prob)
8 cat("V(Y|1) =",round(v, digits = 3))

```

R code Exa 5.12 Machined Dimensions

```

1 #Machined Dimensions(Pg no. 166)
2
3 xmean = 10.5
4 xvar = 0.0025
5 xstd_dev = sqrt(xvar)
6 ymean = 3.2
7 yvar = 0.0036
8 ystd_dev = sqrt(yvar)
9
10 #P(10.4<X<10.6,3.15<Y<3.25)
11 zx_min = pnorm(10.4,xmean,xstd_dev,lower.tail =
    FALSE)
12 zx_max = pnorm(10.6,xmean,xstd_dev)
13
14 zy_min = pnorm(3.15,ymean,ystd_dev, lower.tail =
    FALSE)
15 zy_max = pnorm(3.25,ymean,ystd_dev)
16
17 p = zx_min*zx_max*zy_min*zy_max
18 cat("P(10.4<X<10.6,3.15<Y<3.25) =",round(p,digits =
    1))
19
20 # The answer may slightly vary due to rounding off
    values

```

R code Exa 5.14 Component Lifetimes

```
1 #Component Lifetimes(Pg no. 167)
2
3 f = function(x1,x2,x3,x4)
4 {
5   (exp(-(0.001*x1)-(0.002*x2)-(0.0015*x3)-(0.003*x4)
6     ))
7 }
8 #P(X1>1000,X2>1000,X3>1000,X4>1000)
9
10 p = f(1000,1000,1000,1000)
11 cat("P(X1>1000,X2>1000,X3>1000,X4>1000)=",round(p,
12   digits = 5))
```

R code Exa 5.15 Probability as a Ratio of Volumes

```
1 #install.packages("MASS")
2 library(MASS)
3
4 #Probability as a Ratio of Volumes(Pg no. 168)
5
6 area1 = 4*pi
7 area2 = pi
8
9 p = area2/area1
10 cat("Prabability is",fractions(p))
11 ans = fractions(p)
12 ans
```

R code Exa 5.16 Marginal probability distribution

```
1 #(Pg no. 168)
2
3 #P(X2 = 0)
4 P_0 = 0.4
5
6 #P(X2 = 1)
7 P_1 = 0.3
8
9 #P(X2 = 2)
10 P_2 = 0.2
11
12 #P(X2 = 3)
13 P_3 = 0.1
14
15 x = c(0,1,2,3)
16
17 E = weighted.mean(x,c(P_0,P_1,P_2,P_3))
18 cat("E(X2) =",E)
```

R code Exa 5.18 Layer Thickness

```
1 #Layer Thickness(Pg no. 170)
2
3 mean1 = 10000
4 std_dev1 = 250
5 mean2 = 1000
6 std_dev2 = 20
7 mean3 = 80
8 std_dev3 = 4
9
```

```

10 #P(9200<X1<10800,950<X2<1050,75<X3<85)
11 zx1_min = pnorm(9200,mean1,std_dev1,lower.tail =
      FALSE)
12 zx1_max = pnorm(10800,mean1,std_dev1,lower.tail =
      FALSE)
13
14 zx2_min = pnorm(950,mean2,std_dev2,lower.tail =
      FALSE)
15 zx2_max = pnorm(1050,mean2,std_dev2, lower.tail =
      FALSE)
16
17 zx3_min = pnorm(75,mean3,std_dev3,lower.tail = FALSE
      )
18 zx3_max = pnorm(85,mean3,std_dev3, lower.tail =
      FALSE)
19
20 p = (zx1_min-zx1_max)*(zx2_min-zx2_max)*(zx3_min-zx3
      _max)
21 cat("the requested probability is P(9200<X1
      <10800,950<X2<1050,75<X3<85) =",round(p,digits =
      4))

```

R code Exa 5.19 Expected Value of a Function of Two Random Variables

```

1 #Expected Value of a Function of Two Random
  Variables(Pg no. 174)
2
3 y1 = c(0.01,0.02,0.25)
4 y2 = c(0.02,0.03,0.2)
5 y3 = c(0.02,0.1,0.05)
6 y4 = c(0.15,0.1,0.05)
7
8 #joint probability matrix
9 M = matrix(c(y1,y2,y3,y4), nrow = 4, ncol = 3, byrow
      = TRUE)

```

```

10 ux = 2.35
11 uy = 2.49
12 M
13 sum_total= 0.0
14 for(i in 1:4)
15 {
16     for(j in 1:3)
17     {
18         sum_total = sum_total + ((j-ux)*(i-uy)*M[i,j])
19     }
20 }
21
22 cat("E[(X-ux)(Y-uy)] =",sum_total)

```

R code Exa 5.21 Covariance

```

1 #Covariance(Pg no. 176)
2 x = c(0,1,1,2,2,3)
3 y = c(0,1,2,1,2,3)
4 w = c(0.2,0.1,0.1,0.1,0.1,0.4)
5
6 E_xy = 0.0
7 for(i in 1:length(x))
8 {
9     E_xy = E_xy + (x[i]*y[i]*w[i])
10 }
11
12 E_x = 0*0.2+1*0.2+2*0.2+3*0.4
13 E_x
14 V_x = ((0-E_x)^2*0.2)+((1-E_x)^2*0.2)+((2-E_x)^2*
15         0.2)+((3-E_x)^2*0.4)
16 E_y = E_x
17 V_y = V_x
18 V_x

```

```

19 sigma_xy = E_xy - E_x*E_y
20 sigma_xy
21
22 p_xy = sigma_xy/(sqrt(V_x)*sqrt(V_y))
23 round(p_xy,digits = 3)

```

R code Exa 5.23 Independence Implies Zero Covariance

```

1 #install.packages("pracma")
2 #install.packages("MASS")
3 library(pracma)
4 library(MASS)
5
6 #Independence Implies Zero Covariance(Pg no. 177)
7
8 fun_xy <- function(x, y) (1/16)*(x^2)*(y^2)
9
10 E_xy = integral2(fun_xy, 0, 2, 0, 4, reltol = 1e-10)
11 E_xy = E_xy$Q
12 cat("E(XY) =",E_xy)
13 fractions(E_xy)
14
15 fun_x <- function(x, y) (1/16)*(x^2)*(y)
16
17 E_x = integral2(fun_x, 0, 2, 0, 4, reltol = 1e-10)
18 E_x = E_x$Q
19 cat("E(X) =",E_x)
20 fractions(E_x)
21
22 fun_y <- function(x, y) (1/16)*(x)*(y^2)
23
24 E_y = integral2(fun_y, 0, 2, 0, 4, reltol = 1e-10)
25 E_y = E_y$Q
26 cat("E(Y) =",E_y)
27 fractions(E_y)

```

```

28
29 if(ceiling(E_xy-(E_x*E_y)) == 0)
30 {
31     print("The two random variables are independent")
32 }

```

R code Exa 5.24 Digital Channel

```

1 #Digital Channel(Pg no. 179)
2
3 e = 0.6
4 g = 0.3
5 f = 0.08
6 p = 0.02
7
8 prob = (e^14)*(g^3)*(f^2)*(p^1)
9 prob
10
11 no_seq = factorial(20)/(factorial(14)*factorial(3)*
    factorial(2)*factorial(1))
12 no_seq
13
14 P = no_seq*prob
15 cat("P(14E's, 3 G's, 2 F's, and 1 P) =",round(P,
    digits = 4))

```

R code Exa 5.25 Digital Channel

```

1 #Digital Channel(Pg no.180)
2
3 x = c(0.6,0.3,0.08,0.02)
4 P = dmultinom(c(12,6,2,0), prob = x)

```

```
5 cat("P(12E's, 6 G's, 2 F's, and 0 P) =",round(P,
  digits = 4))
```

R code Exa 5.31 Error Propagation

```
1 #Error Propagation(Pg no. 185)
2
3 var_x = 25+40+30
4 cat("V(X) =",var_x,"nm^2")
5 cat("standard deviation =",round(sqrt(var_x),digits
  = 2),"nm")
```

R code Exa 5.32 Linear Function of Independent Normal Random Variables

```
1 #Linear Function of Independent Normal Random
  Variables(Pg no. 186)
2
3 E_X1 = 2
4 E_X2 = 5
5 std_dev1 = 0.1
6 std_dev2 = 0.2
7 Y = 14.5
8 E_Y = 14
9
10 var_Y = (4*(0.1)^2)+(4*(0.2)^2)
11 z = (Y-E_Y)/sqrt(var_Y)
12
13 p = pnorm(z,0,1,lower.tail = FALSE)
14 cat("the probability that the perimeter exceeds 14.5
  cm is",round(p,digits = 2))
```

R code Exa 5.33 Beverage Volume

```
1 #Beverage Volume(Pg no. 187)
2
3 E_Xbar = 12.1
4 V_Xbar = 0.1^2/10
5 Xbar = 12
6
7 z = (Xbar - E_Xbar)/sqrt(V_Xbar)
8
9 p = pnorm(z,0,1,lower.tail = TRUE)
10 cat("the probability that the average volume of 10
      cans selected from this process is less than 12
      oz is",round(p,digits = 5))
11
12 #The answer may slightly vary due to rounding off
    values
```

Chapter 6

Descriptive Statistics

R code Exa 6.1 Sample mean

```
1 #Sample mean(Pg no. 200)
2
3 obs = c(12.6,12.9,13.4,12.3,13.6,13.5,12.6,13.1)
4 x = mean(obs)
5
6 cat("sample mean is",x,"pounds")
```

R code Exa 6.2 Sample variance

```
1 #Sample variance(Pg no. 202)
2
3 obs = c(12.6,12.9,13.4,12.3,13.6,13.5,12.6,13.1)
4
5 v = var(obs)
6 sd = sd(obs)
7
8 cat("sample variance is ",round(v,digits = 4)) #
  pounds^2
```



```
9 cat("sample standard deviation is",round(sd,digits =  
2)) #pounds
```

R code Exa 6.3 Sample variance shortcut

```
1 #Sample variance shortcut(Pg no. 203)  
2  
3 samples = c(12.6,12.9,13.4,12.3,13.6,13.5,12.6,13.1)  
4 n = length(samples) #number of samples  
5 x = sum(samples) # sum of individual samples  
6 y = sum(samples^2) # sum of square of individual  
  samples  
7  
8 v = (y - ((x)^2/n))/(n-1)  
9  
10 cat("sample variance is ",v) #pounds^2  
11 cat("sample standard deviation is",round(sqrt(v),  
  digits = 2)) #pounds
```

Chapter 7

Point Estimation of Parameters and Sampling Distributions

R code Exa 7.1 Resistors

```
1 #Resistors(Pg no. 175)
2
3 n = 25
4 mean = 100 #ohms
5 x = 95 #ohms
6
7 sd = mean/(sqrt(n)) #standard deviation
8 z = (x - mean)/sd
9
10 #P(x<95) = P(Z<-2.5)
11 p = pnorm(-2.5)
12 round(p, digits = 4)
```

R code Exa 7.3 Aircraft engine life

```
1 #Aircraft engine life(Pg no. 147)
```

```

2
3 mean1 = 5000
4 mean2 = 5050
5
6 variance1 = 10
7 variance2 = 6
8
9 mean = mean2 - mean1
10 varianve = (variance2)^2 + (variance1)^2
11
12 x = 25
13
14 z = (x-mean)/sqrt(varianve)
15
16 #P(X2 - X1 >=25) = P(Z >= z)
17 p = 1 - pnorm(z)
18 p
19
20 cat(round(p,digits = 4),"is the probability that the
      difference in sample means between the new and
      the old process will be at least 25 hours")
21
22 # The answer may slightly vary due to rounding off
    of values

```

R code Exa 7.5 Thermal conductivity

```

1 #Thermal conductivity(Pg no. 252)
2
3 n = 10
4 obs = c(41.60 , 41.48 , 42.34 , 41.95 , 41.86 ,
          42.18 , 41.72 , 42.26 , 41.81 , 42.04)
5 sd = sd(obs) #standard deviation
6
7 se = sd/sqrt(n) #standard error

```

```
8
9 cat("standard error is ",round(se,digits = 4))
```

R code Exa 7.9 Gamma Distribution Moment Estimators

```
1 #Gamma Distribution Moment Estimators(Pg no. 258)
2
3 n = 8
4 x = 21.65
5 y = 6639.40 #sum of square of individual samples
6
7 r = x^2/(((1/n)*y)-(x)^2)
8 r
9
10 lamda = x/(((1/n)*y)-(x)^2)
11 round(lamda,digits = 4)
12
13 # The answer may slightly vary due to rounding off
    of values
```

R code Exa 7.17 Bayes Estimator for the Mean of a Normal Distribution

```
1 #Bayes Estimator for the Mean of a Normal
    Distribution(Pg no. 265)
2
3 n = 10
4 variance = 4
5 mean0 = 0
6 variance0 = 1
7 sample_mean = 0.75
8
9 estimated_mean = ((variance/n)*mean0 + (variance0)*
    sample_mean)/(1 + (variance/n))
```

```
10 cat(" Estimated mean is ",round(estimated_mean,digits  
    = 3))
```

Chapter 8

Statistical intervals for a single sample

R code Exa 8.1 Metallic Material Transition

```
1 #Metallic Material Transition(Pg no. 274)
2
3 n = 10
4 sd = 1
5 z = qnorm(0.025,lower.tail = FALSE)
6 mean = 64.46
7
8 min = mean - z*(sd/sqrt(n))
9 max = mean + z*(sd/sqrt(n))
10
11 cat("The resulting 95% CI is",round(min,digits = 2),
      "<= u <= ",round(max,digits = 2))
```

R code Exa 8.2 Metallic Material Transition

```
1 #Metallic Material Transition(Pg no. 276)
```

```

2
3 e = 0.5 #error in estimation
4 z = qnorm(0.025,lower.tail = FALSE)
5 sd = 1
6
7 n = ((z*sd)/e)^2
8 n = ceiling(n)
9
10 cat("the required sample size is",n)

```

R code Exa 8.3 One Sided Confidence Bound

```

1 #One-Sided Confidence Bound(Pg no. 277)
2
3 n = 10
4 sd = 1
5 z = qnorm(0.05,lower.tail = FALSE)
6 mean = 64.46
7
8 min = mean - z*(sd/sqrt(n))
9
10 cat("the interval is",round(min,digits = 2),"<= u")

```

R code Exa 8.4 The Exponential Distribution

```

1 #The Exponential Distribution(Pg no. 278)
2
3 x = c
      (2.84,2.37,7.52,2.76,3.83,1.32,8.43,2.25,1.63,0.27)

4 n = length(x)
5 df = 2*n #degrees of freedom
6 C1 = qchisq(0.025,20,lower.tail = TRUE)

```

```

7 Cu = qchisq(0.025,20,lower.tail = FALSE)
8 x_sum = sum(x)
9
10 upper_bound = Cu/(2*x_sum)
11 lower_bound = Cl/(2*x_sum)
12
13 cat("The 95% two-sided CI on lambda is",round(lower_
      bound,digits = 4),"<= lambda <=",round(upper_
      bound,digits = 4))
14
15 # The values may slightly vary due to rounding off
    of values

```

R code Exa 8.5 Mercury Contamination

```

1 #Mercury Contamination(Pg no. 279)
2
3 n = 53
4 sd = 0.3486
5 z = qnorm(0.025,lower.tail = FALSE)
6 mean = 0.5250
7
8 min = mean - z*(sd/sqrt(n))
9 max = mean + z*(sd/sqrt(n))
10
11 cat("The approximate 95% CI is",round(min,digits =
      4),"<= u <= ",round(max,digits = 4))

```

R code Exa 8.6 Alloy Adhesion

```

1 #Alloy Adhesion(Pg no. 285)

```



```

2 obs = c
      (19.8,15.4,11.4,19.5,10.1,18.5,14.1,8.8,14.9,7.9,17.6,13.6,7.5,12
3 ans = t.test(obs)
4
5 min = ans$conf.int[1]
6 max = ans$conf.int[2]
7 cat("The resulting 95% CI is",round(min,digits = 2),
      "<= u <= ",round(max,digits = 3))

```

R code Exa 8.7 Detergent Filling

```

1 #Detergent Filling(Pg no. 289)
2
3 sample_variance = 0.0153 #fluidounce
4 n = 20 #bottles
5 x = qchisq(0.95,19, lower.tail = FALSE)
6 variance = ((n-1)*sample_variance)/x
7 sd = sqrt(variance)
8
9 cat("A 95% upper confidence bound on variance is",
      round(variance,digits = 4))
10 cat("A 95% upper confidence bound on standard
      deviation is",round(sd, digits = 2))

```

R code Exa 8.8 Crankshaft Bearings

```

1 #Crankshaft Bearings(Pg no. 292)
2
3 n = 85
4 x = 10
5 p = round(x/n, digits = 2)
6 p

```

```

7 z = round(qnorm(0.025, lower.tail = FALSE), digits =
  2)
8 z
9
10 p_min = p - (z*sqrt((p*(1-p))/n))
11 p_max = p + (z*sqrt((p*(1-p))/n))
12
13 cat("A 95% two-sided confidence interval for p is",
  round(p_min, digits = 4), "<= p <=", round(p_max,
  digits = 4))
14
15 # The answer given in the textbook is wrong for the
  upper bound

```

R code Exa 8.9 Crankshaft Bearings

```

1 #Crankshaft Bearings(Pg no. 293)
2
3 p = 0.12
4 e = 0.05
5 z = qnorm(0.025, lower.tail = FALSE)
6
7 n = (z/e)^2 * p * (1-p)
8 n = ceiling(n)
9
10 cat("sample size is", n)
11
12 n = (z/e)^2 * 0.25
13 n = ceiling(n)
14 n

```

R code Exa 8.10 The Agresti Coull CI on a Proportion

```

1 #The Agresti–Coull CI on a Proportion(Pg no. 294)
2
3 p = 0.12
4 n = 85
5 z = qnorm(0.025, lower.tail = FALSE)
6
7 ucl = (p+((z^2)/(2*n))+(z*sqrt(((p*(1-p))/n)+((z^2)/(
      4*(n^2)))))))/(1+((z^2)/n))
8 lcl = (p+((z^2)/(2*n))-(z*sqrt(((p*(1-p))/n)+((z^2)/(
      4*(n^2)))))))/(1+((z^2)/n))
9
10 cat("UCL =",round(ucl,digits = 4))
11 cat("LCL =",round(lcl,digits = 4))
12 #The answer may slightly vary due to rounding off
    values

```

R code Exa 8.11 Alloy Adhesion

```

1 #Alloy Adhesion(Pg no. 298)
2
3 n = 22
4 sd = 3.55
5 df = n - 1
6 t = qt(0.025,df, lower.tail = FALSE)
7 mean = 13.71
8
9 min = mean - t*sd*sqrt(1+(1/n))
10 max = mean + t*sd*sqrt(1+(1/n))
11
12 cat("A 95% prediction interval on the load at
      failure for this specimen is",round(min,digits =
      2),"<= X <= ",round(max,digits = 2))

```

R code Exa 8.12 Alloy Adhesion

```
1 #install.packages("tolerance")
2 library(tolerance)
3
4 #Alloy Adhesion(Pg no. 299)
5
6 n = 22
7 sd = 3.55
8 mean = 13.71
9 k = K.factor(n, f = NULL, 0.05, 0.90, side = 2)
10
11 min = mean - k*sd
12 max = mean + k*sd
13
14 cat("The desired tolerance interval is [",round(min,
15       digits = 2),",",round(max,digits = 2),"]")
16 # The answer may slightly vary due to rounding off
17     values
```

Chapter 9

Tests of Hypotheses for a Single Sample

R code Exa 9.2 Propellant Burning Rate

```
1 #Propellant Burning Rate(Pg no. 324)
2
3 foo = function()
4 {
5     n = 25
6     mean = 50
7     std_dev = 2
8     Xbar = 51.3
9     alpha = 0.05
10    statistic = (Xbar-mean)/(std_dev/sqrt(n))
11    z = qnorm(alpha/2,0,1,lower.tail = FALSE)
12    p_value = 2*(1-pnorm(statistic,0,1))
13    if(p_value < alpha)
14    {
15        print("Null hypothesis is rejected")
16    }
17    else
18    {
19        print("Unable to reject the null hypothesis")
20    }
21 }
```

```

20   }
21 }
22 foo()

```

R code Exa 9.3 Propellant Burning Rate Type II Error

```

1 #Propellant Burning Rate Type II Error(Pg no. 326)
2
3 n = 25
4 std_dev = 2
5 alpha = 0.05
6 z = qnorm(alpha/2,0,1,lower.tail = FALSE)
7 lim1 = z - (sqrt(n)/std_dev)
8 lim2 = -z - (sqrt(n)/std_dev)
9 beta = pnorm(lim1,0,1) - pnorm(lim2,0,1)
10 beta
11
12 beta2 = 0.10
13 z1 = qnorm(alpha/2,0,1,lower.tail = FALSE)
14 z2 = qnorm(beta2,0,1,lower.tail = FALSE)
15 new_sample_size = (((z1+z2)^2)*(std_dev^2))/1^2
16 new_sample_size = round(new_sample_size)
17 cat("new n =",new_sample_size)

```

R code Exa 9.4 Propellant Burning Rate Type II Error From OC Curve

```

1 #install.packages("MASS")
2 library(MASS)
3
4 #Propellant Burning Rate Type II Error From OC Curve
  (Pg no. 327)
5
6 u =51

```

```

7 u0 = 50
8 std_dev = 2
9
10 d = (u-u0)/std_dev
11 fractions(d)

```

R code Exa 9.6 Golf Club Design

```

1 #Golf Club Design(Pg no. 334)
2
3 foo = function()
4 {
5     obs = c
6         (0.8411,0.8580,0.8042,0.8191,0.8532,0.8730,0.8182,0.8483,0.8282
7
8     ans = t.test(obs, mu = 0.82, alternative = "
9         greater")
10    p_value = ans$p.value
11
12    if(0.005<p_value && p_value<0.01)
13    {
14        print("Null hypothesis rejected and hence the
15            mean coefficient of restitution exceeds 0.82"
16        )
17    }
18    else
19    {
20        print("Unable to reject the null hypothesis")
21    }
22 }
23 foo()

```

R code Exa 9.7 Golf Club Design Sample Size

```

1 #install.packages("pwr")
2 library(pwr)
3
4 #Golf Club Design Sample Size(Pg no. 337)
5 s = 0.02456
6 diff = 0.02
7 sample_size = 15
8 d = diff/s
9 d
10 alpha = 0.05
11 ans = pwr.t.test(n = sample_size, d = d, sig.level =
    alpha, type = "one.sample", alternative = "
    greater")$power
12 cat("Since the probability of rejecting null
    hypothesis is approx.",round(ans, digits = 1),"we
    conclude that a sample size of n = 15 is adequate
    to provide the desired sensitivity")

```

R code Exa 9.8 Automated Filling

```

1 #Automated Filling(Pg no. 342)
2 foo = function()
3 {
4   n = 20
5   sample_var = 0.0153
6   var = 0.01
7   alpha = 0.05
8
9   statistic = ((n-1)*sample_var)/var
10
11   compare = qchisq((1-alpha),(n-1))
12
13   if(statistic>compare)
14   {
15     print("Reject null hypothesis")

```



```

16 }
17 else
18 {
19     print("there is no strong evidence of a problem
           with incorrectly filled bottles")
20 }
21 }
22 foo()

```

R code Exa 9.9 Automated Filling Sample Size

```

1 #Automated Filling Sample Size(Pg no. 343)
2
3 sd = 0.125
4 sd0 = 0.10
5 n1 = 20
6 alpha = 0.05
7
8 lambda = sd/sd0
9
10 cat("With n =",n1,"and lambda =",lambda,"there is
      only 40% chance that the null hypothesis will be
      rejected")

```

R code Exa 9.10 Automobile Engine Controller

```

1 #Automobile Engine Controller(Pg no. 346)
2
3 foo= function()
4 {
5     x = 4
6     n = 200
7     p0 = 0.05

```

```

8
9   statistic = (x - (n*p0))/sqrt(n*p0*(1-p0))
10  statistic
11
12  p_value = pnorm(statistic,0,1)
13  p_value
14
15  if(p_value < p0)
16  {
17      print("Null hypothesis is rejected")
18  }
19  else
20  {
21      print("Unable to reject the null hypothesis")
22  }
23 }
24 foo()

```

R code Exa 9.11 Automobile Engine Controller Type II Error

```

1  #Automobile Engine Controller Type II Error(Pg no.
   347)
2
3  p0 = 0.05
4  n = 200
5  p = 0.03
6  z_alpha = qnorm(p0,0,1,lower.tail = FALSE)
7
8  x = (p0 - p - (z*sqrt((p0*(1-p0))/n)))/sqrt((p*(1-p)
   )/n)
9
10 beta = 1 - pnorm(x,0,1)
11
12 z_beta = qnorm(0.10,0,1,lower.tail = FALSE)
13 z_beta

```

```

14
15 n = (((z_alpha*sqrt(p0*(1-p0))) + (z_beta*sqrt(p*(1-
      p))))/(p-p0))^2
16 n = floor(n)
17 n

```

R code Exa 9.12 Printed Circuit Board Defects Poisson Distribution

```

1 #Printed Circuit Board Defects-Poisson Distribution(
    Pg no. 351)
2
3 foo = function()
4 {
5     n = 60
6     defects = c(0,1,2,3)
7     obsv_freq = c(32,15,9,4)
8     observed_mean = weighted.mean(defects,obsv_freq)
9
10    p1 = dpois(0,lambda = observed_mean)
11    p2 = dpois(1,lambda = observed_mean)
12    p3 = dpois(2,lambda = observed_mean)
13    p4 = 1 - (p1+p2+p3)
14
15    obsv_freq = c(obsv_freq[1],obsv_freq[2],(obsv_freq
        [3]+obsv_freq[4]))
16
17    expected_freq = c(n*p1,n*p2,(n*p3+n*p4))
18
19    statistic = 0.0
20
21    for(i in 1:3)
22    {
23        statistic = statistic + (((obsv_freq[i] -
            expected_freq[i])^2)/expected_freq[i])
24    }

```

```

25     statistic
26
27     x1 = qchisq(0.9,1)
28     x2 = qchisq(0.95,1)
29
30     if(x1<statistic && statistic<x2)
31     {
32         print("Unable to reject the null hypothesis")
33     }
34     else
35     {
36         print("Null hypothesis is rejected")
37     }
38 }
39 foo()

```

R code Exa 9.13 Power Supply Distribution Continuous Distribution

```

1 #Power Supply Distribution-Continuous Distribution(
  Pg no. 352)
2
3 foo = function()
4 {
5     alpha = 0.05
6     obsv_freq = c(12,14,12,13,12,11,12,14)
7     ans = chisq.test(obsv_freq)
8     statistic = ans$statistic
9
10    x = qchisq((1-alpha),5)
11
12    if(statistic<x)
13    {
14        print("Unable to reject the null hypothesis")
15    }
16    else

```

```

17   {
18       print("Null hypothesis is rejected")
19   }
20 }
21 foo()

```

R code Exa 9.14 Health Insurance Plan Preference

```

1 #Health Insurance Plan Preference(Pg no. 355)
2
3 foo = function()
4 {
5     alpha = 0.05
6     n = 500
7     u1 = (340/n)
8     u2 = (160/n)
9     v1 = (200/n)
10    v2 = (200/n)
11    v3 = (100/n)
12
13    O11 = 160
14    O12 = 140
15    O13 = 40
16    O21 = 40
17    O22 = 60
18    O23 = 60
19
20    observed = c(O11,O12,O13,O21,O22,O23)
21    observed
22
23    E11 = n*u1*v1
24    E12 = n*u1*v2
25    E13 = n*u1*v3
26    E21 = n*u2*v1
27    E22 = n*u2*v2

```

```

28     E23 = n*u2*v3
29
30     expected = c(E11,E12,E13,E21,E22,E23)
31     expected
32
33     r=2
34     c=3
35
36     df = (r-1)*(c-1) #degrees of freedom
37
38     statistic = 0.0
39     for(i in 1:6)
40     {
41         statistic = statistic + ((observed[i]-expected[i
42             ])^2)/expected[i]
43     }
44     statistic
45
46     x = qchisq(0.95,df)
47     x
48
49     if(statistic<x)
50     {
51         print("Unable to reject the null hypothesis")
52     }
53     else
54     {
55         print("Null hypothesis is rejected")
56     }
57 }
foo()

```

R code Exa 9.15 Propellant Shear Strength Sign Test

```
1 #install.packages("BSDA")
```

```

2 library(BSDA)
3 #Propellant Shear Strength Sign Test(Pg no. 359)
4
5 foo = function()
6 {
7     x = c
8         (2158.7,1678.15,2316.00,2061.30,2207.50,1708.30,1784.70,2575.10
9
10     ans = SIGN.test(x-2000)
11     p_value = ans$p.value
12     print(p_value)
13     alpha = 0.05
14
15     if(p_value<alpha)
16     {
17         print("Null hypothesis is rejected")
18     }
19     else
20     {
21         print("Unable to reject the null hypothesis")
22     }
23 }
24 foo()

```

R code Exa 9.16 Propellant Shear Strength Wilcoxon Signed Rank Test

```

1 #Propellant Shear Strength–Wilcoxon Signed–Rank Test
  (Pg no. 363)
2 foo = function()
3 {
4     obs = c
5         (2158.7,1678.15,2316.00,2061.30,2207.50,1708.30,1784.70,2575.10
6
7     alpha = 0.05
8     ans = wilcox.test(obs,mu = 2000)

```

```

7   p_value = ans$p.value
8
9   if(p_value<alpha)
10  {
11    print("Null hypothesis rejected")
12  }
13  else
14  {
15    print("Unable to reject the null hypothesis")
16  }
17 }
18 foo()

```

R code Exa 9.17 Two one sided tests

```

1  options(scipen=999)#For disabling scientific
   notation
2
3  #(Pg no. 366)
4
5  foo = function()
6  {
7    n = 50
8    sample_mean = 79.98
9    sample_sd = 0.10
10   x = 80 #ohms
11   u1 = 80.05
12   u2 = 79.95
13   error = 0.01 #ohms
14
15   t_statistic1 = (sample_mean-80.05)/(sample_sd/sqrt
                     (n))
16   p1 = pnorm(t_statistic1,0,1)
17
18   t_statistic2 = (sample_mean-79.95)/(sample_sd/sqrt

```



```

        (n))
19  p2 = pnorm(t_statistic2,0,1,lower.tail = FALSE)
20
21  if(p1<error)
22  {
23      print("the mean resistance is less than 80.05")
24  }
25  else
26  {
27      print("the mean resistance is not less than
          80.05")
28  }
29
30  if(p2<0.025)
31  {
32      print("the mean resistance is significantly
          greater than 79.95")
33  }
34  else
35  {
36      print("the mean resistance is not greater than
          79.95")
37  }
38 }
39 foo()

```

Chapter 10

Statistical Inference for Two Samples

R code Exa 10.1 Paint Drying Time

```
1 #Paint Drying Time(Pg no. 377)
2
3 foo = function()
4 {
5   X1bar = 121
6   X2bar = 112
7   std_dev1 = std_dev2 = 8
8   n1 = n2 = 10
9   delta0 = 0
10
11   statistic = (X1bar-X2bar-delta0)/sqrt((std_dev1^2/
12     n1)+(std_dev2^2/n2))
13   statistic
14
15   p = 1 - pnorm(statistic,0,1,lower.tail = TRUE)
16   p
17   if(p<0.05)
18   {
```

```

19     print("Null hypothesis is rejected ")
20 }
21 else
22 {
23     print("Null hypothesis is accepted")
24 }
25 }
26 foo()

```

R code Exa 10.2 Paint Drying Time Sample Size from OC Curves

```

1 #Paint Drying Time, Sample Size from OC Curves(Pg no
  . 378)
2
3 delta = 10
4 std_dev1 = std_dev2 = 8
5
6 d = (delta)/sqrt(std_dev1^2 + std_dev1^2)
7
8 round(d, digits = 2)

```

R code Exa 10.3 Paint Drying Time Sample Size

```

1 #Paint Drying Time Sample Size(Pg no. 379)
2
3 alpha = 0.05
4 beta = 0.10
5 delta = 10
6 delta0 = 0
7 std_dev1 = std_dev1 = 8
8
9 z_alpha = qnorm(alpha,0,1,lower.tail = FALSE)
10 z_alpha

```

```

11 z_beta = qnorm(beta,0,1,lower.tail = FALSE)
12 z_beta
13
14 n = ((z_alpha+z_beta)^2 * (std_dev1^2 + std_dev2^2))
      /(delta-delta0)^2
15 n = round(n)
16 n

```

R code Exa 10.4 Aluminum Tensile Strength

```

1 #Aluminum Tensile Strength(Pg no. 380)
2
3 X1bar = 87.6
4 X2bar = 74.5
5 std_dev1 = 1.0
6 std_dev2 = 1.5
7 n1 = 10
8 n2 =12
9 alpha = 0.1
10
11 z_alpha = qnorm(alpha/2,0,1,lower.tail = FALSE)
12
13 lim1 = X1bar - X2bar - (z_alpha * sqrt(((std_dev1^2)
      /n1)+((std_dev2^2)/n2)))
14 lim2 = X1bar - X2bar + (z_alpha * sqrt(((std_dev1^2)
      /n1)+((std_dev2^2)/n2)))
15
16 cat("90% CI is",round(lim1, digits = 2),"<= u - u0
      <=",round(lim2, digits = 2))

```

R code Exa 10.5 Yield from a Catalyst

```

1 #Yield from a Catalyst(Pg no. 384)

```

```

2
3 foo = function()
4 {
5   catalyst1 = c
        (91.50,94.18,92.18,95.39,91.79,89.07,94.72,89.21)

6   catalyst2 = c
        (89.19,90.95,90.46,93.21,97.19,97.04,91.07,92.75)

7   ans = t.test(catalyst1,catalyst2)
8   statistic = ans$statistic
9   cat("t-statistic =",round(statistic, digits = 2),"
        \n")
10
11   l1 = qt(0.40, df = 14, lower.tail = FALSE)
12   l2 = qt(0.25, df = 14, lower.tail = FALSE)
13
14   if(l1<abs(statistic) && abs(statistic)<l2)
15   {
16     print("Unable to reject the null hypothesis")
17
18   }
19   else
20   {
21     print("Null hypothesis rejected")
22   }
23 }
24 foo()

```

R code Exa 10.6 Arsenic in Drinking Water

```

1 #Arsenic in Drinking Water(Pg no.387)
2
3 foo = function()
4 {

```

```

5   X1bar = 12.5
6   X2bar = 27.5
7   std_dev1 = 7.63
8   std_dev2 = 15.3
9   n1 = 10
10  n2 = 10
11
12  v = ((std_dev1^2/n1)+(std_dev2^2/n2))^2 / (((std_dev1^2/n1)^2/(n1-1))+((std_dev2^2/n2)^2/(n2-1)))
13  v = round(v)
14  v
15
16  statistic = (X1bar - X2bar)/sqrt((std_dev1^2/n1)+(std_dev2^2/n2))
17  cat("t-statistic = ",round(statistic, digits = 2),
18      "\n")
19
20  l1 = qt(0.025, df = 13)
21
22  if(statistic<l1)
23  {
24      print("Null hypothesis rejected")
25  }
26  else
27  {
28      print("Unable to reject the null hypothesis")
29  }
30  }
31  foo()

```

R code Exa 10.8 Yield from Catalyst Sample Size

```

1  #Yield from Catalyst Sample Size(Pg no. 390)
2
3  sp = 2.70

```

```

4 delta = 4.0
5 std_dev = (std_dev1+std_dev2)/2
6
7 d = delta/(2*std_dev)
8 d
9
10 n1 = 20
11
12 n = (n1+1)/2
13 n = ceiling(n)
14 n

```

R code Exa 10.9 Cement Hydration

```

1 #Cement Hydration(Pg no. 391)
2
3 X1bar = 90.0
4 X2bar = 87.0
5
6 s1 = 5.0
7 s2 = 4.0
8
9 n1 = 10
10 n2 = 15
11
12 sp = sqrt(((n1-1)*s1^2)+((n2-1)*s2^2))/(n1+n2-2)
13 sp
14
15 t = qt(0.025,23,lower.tail = FALSE)
16 t
17 lim1 = X1bar-X2bar-(t*sp*sqrt((1/n1)+(1/n2)))
18 lim2 = X1bar-X2bar+(t*sp*sqrt((1/n1)+(1/n2)))
19
20 cat("The 95% confidence interval is found as",round
    (lim1, digits = 2),"<= u-u0 <=",round(lim2,

```

```

    digits = 2))
21
22 # The answer may slightly vary due to rounding off
    values

```

R code Exa 10.10 Axial Stress

```

1 #Axial Stress (Pg no. 397)
2
3 n1 = n2 = 10
4 rank1 = c(2,3,4,8,9,11,13,15,16,18)
5
6 w1 = sum(rank1)
7
8 w2 = (((n1+n2)*(n1+n2+1))/2) - w1
9 w2

```

R code Exa 10.11 Shear Strength of Steel Girder

```

1 #Shear Strength of Steel Girder (Pg no. 401)
2
3 foo = function()
4 {
5     km = c
6         (1.186,1.151,1.322,1.339,1.200,1.402,1.365,1.537,1.559)
7
8     lm = c
9         (1.061,0.992,1.063,1.062,1.065,1.178,1.037,1.086,1.052)
10
11     ans = t.test(km,lm, alternative = "two.sided",
12                 paired = TRUE)
13     statistic = ans$statistic
14     print(statistic)

```



```

10     alpha = 0.05
11     p_value = ans$p.value
12
13     if(p_value<alpha)
14     {
15         print("the strength prediction methods yield
                different results")
16     }
17     else
18     {
19         print("the strength prediction methods yield
                similar results")
20     }
21 }
22 foo()

```

R code Exa 10.12 Parallel Park Cars

```

1 #Parallel Park Cars(Pg no. 403)
2
3 x1 = c
      (37.0,25.8,16.2,24.2,22.0,33.4,23.8,58.2,33.6,24.4,23.4,21.2,36.2
4
5 x2 = c
      (17.8,20.2,16.8,41.4,21.4,38.4,16.8,32.2,27.8,23.2,29.6,20.6,32.2
6
7 ans = t.test(x1,x2,conf.level = 0.9, paired = TRUE)
8 ans
9
10 l1 = ans$conf.int[1]
11 l2 = ans$conf.int[2]
12
13 cat(l1,"<= uD <=",l2)
14
15 #The answer may slightly vary due to rounding off of
    values

```

R code Exa 10.13 Semiconductor Etch Variability

```
1 #Semiconductor Etch Variability(Pg no. 410)
2
3 s1 = 1.96
4 s2 = 2.13
5 alpha = 0.05
6 n1 = n2 =16
7
8 statistic = s1^2/s2^2
9 statistic
10
11 f_max = qf(0.025,n1-1,n2-1,lower.tail = FALSE)
12 f_min = qf(0.975,15,15,lower.tail = FALSE)
13
14 if(f_min < statistic && statistic < f_max)
15 {
16   cat("we cannot reject the null hypothesis at 0.05
17     level of significance")
18 }
```

R code Exa 10.15 Surface Finish for Titanium Alloy

```
1 #Surface Finish for Titanium Alloy(Pg no. 412)
2
3 s1 = 5.1
4 s2 = 4.7
5 n1 = 11
6 n2 =16
7
8 statistic = s1^2/s2^2
```

```

9 statistic
10
11 f_max = qf(0.05,n2-1,n1-1,lower.tail = FALSE)
12 f_max
13 f_min = qf(0.95,n2-1,n1-1,lower.tail = FALSE)
14 f_min
15
16
17 cat("the interval is (",round(sqrt(statistic*f_min),
    digits = 3),",",round(sqrt(statistic*f_max),
    digits = 3),")")
18
19 #The answer may slightly vary due to rounding off of
    values

```

R code Exa 10.16 St Johns Wort

```

1 #St. John's Wort(Pg no. 415)
2
3 x1 = 19
4 x2 = 27
5 n1 = n2 = 100
6 p1 = x2/n2
7 p2 = x1/n1
8
9 p = (x1+x2)/(n1+n2)
10 cat("p =",p)
11
12 statistic = (p1-p2)/sqrt(p*(1-p)*((1/n1)+(1/n2)))
13 statistic
14
15 p = 2*(1-pnorm(statistic,0,1,lower.tail = TRUE))
16 cat("P-value =",p)
17
18 if(p>=0.05)

```

```

19 {
20   cat("we cannot reject the null hypothesis")
21 }

```

R code Exa 10.17 Defective Bearings

```

1 #Defective Bearings(Pg no. 419)
2
3 n1 = n2 = 85
4 p1 = 10/n1
5 p2 = 8/n2
6
7 z = qnorm(0.025,0,1,lower.tail = FALSE)
8
9 lim1 = p1 - p2 - z*sqrt((p1*(1-p1))/n1 + (p2*(1-p2))/
10   n2)
11
12 lim2= p1 - p2 + z*sqrt((p1*(1-p1))/n1 + (p2*(1-p2))/
13   n2)
14
15 cat("the required interval is (",round(lim1, digits
16   = 4),",",round(lim2, digits = 4),")")
17
18 # The answer may slightly vary duet to rounding off
19   values

```

Chapter 11

Simple Linear Regression and Correlation

R code Exa 11.1 Oxygen purity

```
1 #Oxygen purity(Pg no. 433)
2
3 x = c
    (0.99,1.02,1.15,1.29,1.46,1.36,0.87,1.23,1.55,1.40,1.19,1.15,0.98
4 y = c
    (90.01,89.05,91.43,93.74,96.73,94.45,87.59,91.77,99.42,93.65,93.5
5 df = data.frame('y' = y, 'x' = x)
6 model = lm(y~x, data = df)
7
8 cat("The fitted simple linear regression model is y
    = ",round(model$coefficients[1],digits = 3),"+",
    round(model$coefficients[2],digits = 3),"x")
```

R code Exa 11.2 Oxygen Purity Tests of Coefficients

```

1 #Oxygen Purity Tests of Coefficients (Pg no. 442)
2
3 foo = function()
4 {
5   alpha = 0.01
6   beta1 = 14.947
7   S_xx = 0.68088
8   sample_var = 1.18
9
10  statistic = beta1/sqrt(sample_var/S_xx)
11  round(statistic, digits = 2)
12
13  t = qt(0.005,18,lower.tail = FALSE)
14  round(t,digits = 2)
15
16  if(t < statistic)
17  {
18    print("Null hypothesis is rejected as the test
          statistic is very far into the critical
          region")
19  }
20  else
21  {
22    print("Unable to reject the null hypothesis")
23  }
24 }
25 foo()

```

R code Exa 11.3 Oxygen Purity ANOVA

```

1 #Oxygen Purity ANOVA (Pg no. 444)
2
3 x = c
    (0.99,1.02,1.15,1.29,1.46,1.36,0.87,1.23,1.55,1.40,1.19,1.15,0.98

```

```

4 y = c
    (90.01,89.05,91.43,93.74,96.73,94.45,87.59,91.77,99.42,93.65,93.54)

5 df = data.frame('y' = y, 'x' = x)
6 ans = aov(formula = y~x, data = df)
7 ans = summary(ans)
8 ans
9 # The answer may slightly vary due to rounding off
    values

```

R code Exa 11.4 Oxygen Purity Confidence Interval on the Slope

```

1 #Oxygen Purity Confidence Interval on the Slope(Pg
    no.447)
2
3 x = c
    (0.99,1.02,1.15,1.29,1.46,1.36,0.87,1.23,1.55,1.40,1.19,1.15,0.98)

4 y = c
    (90.01,89.05,91.43,93.74,96.73,94.45,87.59,91.77,99.42,93.65,93.54)

5 df = data.frame('y' = y, 'x' = x)
6 model = lm(y~x, data = df)
7 ans = confint(model)
8
9 lim1 = ans[2]
10 lim2 = ans[4]
11
12 cat(round(lim1, digits = 3), "<= Beta <=", round(lim2,
    digits = 3))

```

R code Exa 11.5 Oxygen Purity Confidence Interval on the Mean Response

```

1 #Oxygen Purity Confidence Interval on the Mean
  Response(Pg no. 448)
2
3 x = c
    (0.99,1.02,1.15,1.29,1.46,1.36,0.87,1.23,1.55,1.40,1.19,1.15,0.98
4 y = c
    (90.01,89.05,91.43,93.74,96.73,94.45,87.59,91.77,99.42,93.65,93.5
5 df = data.frame('y' = y, 'x' = x)
6 model = lm(y~x, data = df)
7
8 ans = predict(model,data.frame(x = 1),interval = "
    confidence")
9 cat("prediction of mean oxygen purity when x0 =
    1.00% is",ans[1])
10
11 min = ans[2]
12 max = ans[3]
13
14 cat(round(min, digits = 2),"<= u <=",round(max,
    digits = 2))
15
16 # The answer may slightly vary due to rounding off
    values

```

R code Exa 11.6 Oxygen Purity Prediction Interval

```

1 #Oxygen Purity Prediction Interval(Pg no. 450)
2
3 x = c
    (0.99,1.02,1.15,1.29,1.46,1.36,0.87,1.23,1.55,1.40,1.19,1.15,0.98
4 y = c
    (90.01,89.05,91.43,93.74,96.73,94.45,87.59,91.77,99.42,93.65,93.5

```



```

5 df = data.frame('y' = y, 'x' = x)
6 model = lm(y~x, data = df)
7
8 ans = predict(model,data.frame(x = 1),interval = "
      prediction")
9 cat("prediction of mean oxygen purity when x0 =
      1.00% is",ans[1])
10
11 min = ans[2]
12 max = ans[3]
13
14 cat(round(min, digits = 2),"<= y0 <=",round(max,
      digits = 2))

```

R code Exa 11.8 Wire Bond Pull Strength

```

1 #Wire Bond Pull Strength(Pg no. 460)
2
3 S_xx = 698.56
4 S_xy = 2027.7132
5 SSt = 6105.9
6 n = 25
7
8 r = S_xy/sqrt(S_xx*SSt)
9
10 statistic = (r*sqrt(n-2))/sqrt(1-(r^2))
11 statistic
12
13 compare = qt(0.025,n-2,lower.tail = FALSE)
14 compare
15
16 z = qnorm(0.025,0,1,lower.tail = FALSE)
17 z
18

```

```
19
20 lim1 = tanh(atanh(r) - (z/sqrt(n-3)))
21 lim2 = tanh(atanh(r) + (z/sqrt(n-3)))
22
23 cat(round(lim1, digits = 4), "<= P <= ", round(lim2,
    digits = 4))
```

Chapter 12

Multiple Linear Regression

R code Exa 12.1 Wire Bond Strength

```
1 #Wire Bond Strength(Pg no. 482)
2
3 x1 = c
      (2,8,11,10,8,4,2,2,9,8,4,11,12,2,4,4,20,1,10,15,15,16,17,6,5)
4
5 x2 = c
      (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,600,
70,100,110,120,130,140,150,160,170,180,190,200,210,220,230,240,250,260,270,280,290,300,310,320,330,340,350,360,370,380,390,400,410,420,430,440,450,460,470,480,490,500,510,520,530,540,550,560,570,580,590,600,610,620,630,640,650,660,670,680,690,700,710,720,730,740,750,760,770,780,790,800,810,820,830,840,850,860,870,880,890,900,910,920,930,940,950,960,970,980,990,1000)
6
7 y = c
      (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,18.75,20.12,21.45,22.78,24.11,25.44,26.77,28.10,29.43,30.76,32.09,33.42,34.75,36.08,37.41,38.74,40.07,41.40,42.73,44.06,45.39,46.72,48.05,49.38,50.71,52.04,53.37,54.70,56.03,57.36,58.69,60.02,61.35,62.68,64.01,65.34,66.67,68.00,69.33,70.66,71.99,73.32,74.65,75.98,77.31,78.64,79.97,81.30,82.63,83.96,85.29,86.62,87.95,89.28,90.61,91.94,93.27,94.60,95.93,97.26,98.59,99.92,101.25,102.58,103.91,105.24,106.57,107.90,109.23,110.56,111.89,113.22,114.55,115.88,117.21,118.54,119.87,121.20,122.53,123.86,125.19,126.52,127.85,129.18,130.51,131.84,133.17,134.50,135.83,137.16,138.49,139.82,141.15,142.48,143.81,145.14,146.47,147.80,149.13,150.46,151.79,153.12,154.45,155.78,157.11,158.44,159.77,161.10,162.43,163.76,165.09,166.42,167.75,169.08,170.41,171.74,173.07,174.40,175.73,177.06,178.39,179.72,181.05,182.38,183.71,185.04,186.37,187.70,189.03,190.36,191.69,193.02,194.35,195.68,197.01,198.34,199.67,201.00,202.33,203.66,204.99,206.32,207.65,208.98,210.31,211.64,212.97,214.30,215.63,216.96,218.29,219.62,220.95,222.28,223.61,224.94,226.27,227.60,228.93,230.26,231.59,232.92,234.25,235.58,236.91,238.24,239.57,240.90,242.23,243.56,244.89,246.22,247.55,248.88,250.21,251.54,252.87,254.20,255.53,256.86,258.19,259.52,260.85,262.18,263.51,264.84,266.17,267.50,268.83,270.16,271.49,272.82,274.15,275.48,276.81,278.14,279.47,280.80,282.13,283.46,284.79,286.12,287.45,288.78,290.11,291.44,292.77,294.10,295.43,296.76,298.09,299.42,300.75,302.08,303.41,304.74,306.07,307.40,308.73,310.06,311.39,312.72,314.05,315.38,316.71,318.04,319.37,320.70,322.03,323.36,324.69,326.02,327.35,328.68,330.01,331.34,332.67,334.00,335.33,336.66,337.99,339.32,340.65,341.98,343.31,344.64,345.97,347.30,348.63,349.96,351.29,352.62,353.95,355.28,356.61,357.94,359.27,360.60,361.93,363.26,364.59,365.92,367.25,368.58,369.91,371.24,372.57,373.90,375.23,376.56,377.89,379.22,380.55,381.88,383.21,384.54,385.87,387.20,388.53,389.86,391.19,392.52,393.85,395.18,396.51,397.84,399.17,400.50,401.83,403.16,404.49,405.82,407.15,408.48,409.81,411.14,412.47,413.80,415.13,416.46,417.79,419.12,420.45,421.78,423.11,424.44,425.77,427.10,428.43,429.76,431.09,432.42,433.75,435.08,436.41,437.74,439.07,440.40,441.73,443.06,444.39,445.72,447.05,448.38,449.71,451.04,452.37,453.70,455.03,456.36,457.69,459.02,460.35,461.68,463.01,464.34,465.67,467.00,468.33,469.66,470.99,472.32,473.65,474.98,476.31,477.64,478.97,480.30,481.63,482.96,484.29,485.62,486.95,488.28,489.61,490.94,492.27,493.60,494.93,496.26,497.59,498.92,500.25,501.58,502.91,504.24,505.57,506.90,508.23,509.56,510.89,512.22,513.55,514.88,516.21,517.54,518.87,520.20,521.53,522.86,524.19,525.52,526.85,528.18,529.51,530.84,532.17,533.50,534.83,536.16,537.49,538.82,540.15,541.48,542.81,544.14,545.47,546.80,548.13,549.46,550.79,552.12,553.45,554.78,556.11,557.44,558.77,560.10,561.43,562.76,564.09,565.42,566.75,568.08,569.41,570.74,572.07,573.40,574.73,576.06,577.39,578.72,580.05,581.38,582.71,584.04,585.37,586.70,588.03,589.36,590.69,592.02,593.35,594.68,596.01,597.34,598.67,600.00,601.33,602.66,603.99,605.32,606.65,607.98,609.31,610.64,611.97,613.30,614.63,615.96,617.29,618.62,619.95,621.28,622.61,623.94,625.27,626.60,627.93,629.26,630.59,631.92,633.25,634.58,635.91,637.24,638.57,639.90,641.23,642.56,643.89,645.22,646.55,647.88,649.21,650.54,651.87,653.20,654.53,655.86,657.19,658.52,659.85,661.18,662.51,663.84,665.17,666.50,667.83,669.16,670.49,671.82,673.15,674.48,675.81,677.14,678.47,679.80,681.13,682.46,683.79,685.12,686.45,687.78,689.11,690.44,691.77,693.10,694.43,695.76,697.09,698.42,699.75,701.08,702.41,703.74,705.07,706.40,707.73,709.06,710.39,711.72,713.05,714.38,715.71,717.04,718.37,719.70,721.03,722.36,723.69,725.02,726.35,727.68,729.01,730.34,731.67,733.00,734.33,735.66,736.99,738.32,739.65,740.98,742.31,743.64,744.97,746.30,747.63,748.96,750.29,751.62,752.95,754.28,755.61,756.94,758.27,759.60,760.93,762.26,763.59,764.92,766.25,767.58,768.91,770.24,771.57,772.90,774.23,775.56,776.89,778.22,779.55,780.88,782.21,783.54,784.87,786.20,787.53,788.86,790.19,791.52,792.85,794.18,795.51,796.84,798.17,799.50,800.83,802.16,803.49,804.82,806.15,807.48,808.81,810.14,811.47,812.80,814.13,815.46,816.79,818.12,819.45,820.78,822.11,823.44,824.77,826.10,827.43,828.76,830.09,831.42,832.75,834.08,835.41,836.74,838.07,839.40,840.73,842.06,843.39,844.72,846.05,847.38,848.71,850.04,851.37,852.70,854.03,855.36,856.69,858.02,859.35,860.68,862.01,863.34,864.67,866.00,867.33,868.66,869.99,871.32,872.65,873.98,875.31,876.64,877.97,879.30,880.63,881.96,883.29,884.62,885.95,887.28,888.61,889.94,891.27,892.60,893.93,895.26,896.59,897.92,899.25,900.58,901.91,903.24,904.57,905.90,907.23,908.56,909.89,911.22,912.55,913.88,915.21,916.54,917.87,919.20,920.53,921.86,923.19,924.52,925.85,927.18,928.51,929.84,931.17,932.50,933.83,935.16,936.49,937.82,939.15,940.48,941.81,943.14,944.47,945.80,947.13,948.46,949.79,951.12,952.45,953.78,955.11,956.44,957.77,959.10,960.43,961.76,963.09,964.42,965.75,967.08,968.41,969.74,971.07,972.40,973.73,975.06,976.39,977.72,979.05,980.38,981.71,983.04,984.37,985.70,987.03,988.36,989.69,991.02,992.35,993.68,995.01,996.34,997.67,999.00,1000.33,1001.66,1003.0,1004.33,1005.66,1007.0,1008.33,1009.66,1011.0,1012.33,1013.66,1015.0,1016.33,1017.66,1019.0,1020.33,1021.66,1023.0,1024.33,1025.66,1027.0,1028.33,1029.66,1031.0,1032.33,1033.66,1035.0,1036.33,1037.66,1039.0,1040.33,1041.66,1043.0,1044.33,1045.66,1047.0,1048.33,1049.66,1051.0,1052.33,1053.66,1055.0,1056.33,1057.66,1059.0,1060.33,1061.66,1063.0,1064.33,1065.66,1067.0,1068.33,1069.66,1071.0,1072.33,1073.66,1075.0,1076.33,1077.66,1079.0,1080.33,1081.66,1083.0,1084.33,1085.66,1087.0,1088.33,1089.66,1091.0,1092.33,1093.66,1095.0,1096.33,1097.66,1099.0,1100.33,1101.66,1103.0,1104.33,1105.66,1107.0,1108.33,1109.66,1111.0,1112.33,1113.66,1115.0,1116.33,1117.66,1119.0,1120.33,1121.66,1123.0,1124.33,1125.66,1127.0,1128.33,1129.66,1131.0,1132.33,1133.66,1135.0,1136.33,1137.66,1139.0,1140.33,1141.66,1143.0,1144.33,1145.66,1147.0,1148.33,1149.66,1151.0,1152.33,1153.66,1155.0,1156.33,1157.66,1159.0,1160.33,1161.66,1163.0,1164.33,1165.66,1167.0,1168.33,1169.66,1171.0,1172.33,1173.66,1175.0,1176.33,1177.66,1179.0,1180.33,1181.66,1183.0,1184.33,1185.66,1187.0,1188.33,1189.66,1191.0,1192.33,1193.66,1195.0,1196.33,1197.66,1199.0,1200.33,1201.66,1203.0,1204.33,1205.66,1207.0,1208.33,1209.66,1211.0,1212.33,1213.66,1215.0,1216.33,1217.66,1219.0,1220.33,1221.66,1223.0,1224.33,1225.66,1227.0,1228.33,1229.66,1231.0,1232.33,1233.66,1235.0,1236.33,1237.66,1239.0,1240.33,1241.66,1243.0,1244.33,1245.66,1247.0,1248.33,1249.66,1251.0,1252.33,1253.66,1255.0,1256.33,1257.66,1259.0,1260.33,1261.66,1263.0,1264.33,1265.66,1267.0,1268.33,1269.66,1271.0,1272.33,1273.66,1275.0,1276.33,1277.66,1279.0,1280.33,1281.66,1283.0,1284.33,1285.66,1287.0,1288.33,1289.66,1291.0,1292.33,1293.66,1295.0,1296.33,1297.66,1299.0,1300.33,1301.66,1303.0,1304.33,1305.66,1307.0,1308.33,1309.66,1311.0,1312.33,1313.66,1315.0,1316.33,1317.66,1319.0,1320.33,1321.66,1323.0,1324.33,1325.66,1327.0,1328.33,1329.66,1331.0,1332.33,1333.66,1335.0,1336.33,1337.66,1339.0,1340.33,1341.66,1343.0,1344.33,1345.66,1347.0,1348.33,1349.66,1351.0,1352.33,1353.66,1355.0,1356.33,1357.66,1359.0,1360.33,1361.66,1363.0,1364.33,1365.66,1367.0,1368.33,1369.66,1371.0,1372.33,1373.66,1375.0,1376.33,1377.66,1379.0,1380.33,1381.66,1383.0,1384.33,1385.66,1387.0,1388.33,1389.66,1391.0,1392.33,1393.66,1395.0,1396.33,1397.66,1399.0,1400.33,1401.66,1403.0,1404.33,1405.66,1407.0,1408.33,1409.66,1411.0,1412.33,1413.66,1415.0,1416.33,1417.66,1419.0,1420.33,1421.66,1423.0,1424.33,1425.66,1427.0,1428.33,1429.66,1431.0,1432.33,1433.66,1435.0,1436.33,1437.66,1439.0,1440.33,1441.66,1443.0,1444.33,1445.66,1447.0,1448.33,1449.66,1451.0,1452.33,1453.66,1455.0,1456.33,1457.66,1459.0,1460.33,1461.66,1463.0,1464.33,1465.66,1467.0,1468.33,1469.66,1471.0,1472.33,1473.66,1475.0,1476.33,1477.66,1479.0,1480.33,1481.66,1483.0,1484.33,1485.66,1487.0,1488.33,1489.66,1491.0,1492.33,1493.66,1495.0,1496.33,1497.66,1499.0,1500.33,1501.66,1503.0,1504.33,1505.66,1507.0,1508.33,1509.66,1511.0,1512.33,1513.66,1515.0,1516.33,1517.66,1519.0,1520.33,1521.66,1523.0,1524.33,1525.66,1527.0,1528.33,1529.66,1531.0,1532.33,1533.66,1535.0,1536.33,1537.66,1539.0,1540.33,1541.66,1543.0,1544.33,1545.66,1547.0,1548.33,1549.66,1551.0,1552.33,1553.66,1555.0,1556.33,1557.66,1559.0,1560.33,1561.66,1563.0,1564.33,1565.66,1567.0,1568.33,1569.66,1571.0,1572.33,1573.66,1575.0,1576.33,1577.66,1579.0,1580.33,1581.66,1583.0,1584.33,1585.66,1587.0,1588.33,1589.66,1591.0,1592.33,1593.66,1595.0,1596.33,1597.66,1599.0,1600.33,1601.66,1603.0,1604.33,1605.66,1607.0,1608.33,1609.66,1611.0,1612.33,1613.66,1615.0,1616.33,1617.66,1619.0,1620.33,1621.66,1623.0,1624.33,1625.66,1627.0,1628.33,1629.66,1631.0,1632.33,1633.66,1635.0,1636.33,1637.66,1639.0,1640.33,1641.66,1643.0,1644.33,1645.66,1647.0,1648.33,1649.66,1651.0,1652.33,1653.66,1655.0,1656.33,1657.66,1659.0,1660.33,1661.66,1663.0,1664.33,1665.66,1667.0,1668.33,1669.66,1671.0,1672.33,1673.66,1675.0,1676.33,1677.66,1679.0,1680.33,1681.66,1683.0,1684.33,1685.66,1687.0,1688.33,1689.66,1691.0,1692.33,1693.66,1695.0,1696.33,1697.66,1699.0,1700.33,1701.66,1703.0,1704.33,1705.66,1707.0,1708.33,1709.66,1711.0,1712.33,1713.66,1715.0,1716.33,1717.66,1719.0,1720.33,1721.66,1723.0,1724.33,1725.66,1727.0,1728.33,1729.66,1731.0,1732.33,1733.66,1735.0,1736.33,1737.66,1739.0,1740.33,1741.66,1743.0,1744.33,1745.66,1747.0,1748.33,1749.66,1751.0,1752.33,1753.66,1755.0,1756.33,1757.66,1759.0,1760.33,1761.66,1763.0,1764.33,1765.66,1767.0,1768.33,1769.66,1771.0,1772.33,1773.66,1775.0,1776.33,1777.66,1779.0,1780.33,1781.66,1783.0,1784.33,1785.66,1787.0,1788.33,1789.66,1791.0,1792.33,1793.66,1795.0,1796.33,1797.66,1799.0,1800.33,1801.66,1803.0,1804.33,1805.66,1807.0,1808.33,1809.66,1811.0,1812.33,1813.66,1815.0,1816.33,1817.66,1819.0,1820.33,1821.66,1823.0,1824.33,1825.66,1827.0,1828.33,1829.66,1831.0,1832.33,1833.66,1835.0,1836.33,1837.66,1839.0,1840.33,1841.66,1843.0,1844.33,1845.66,1847.0,1848.33,1849.66,1851.0,1852.33,1853.66,1855.0,1856.33,1857.66,1859.0,1860.33,1861.66,1863.0,1864.33,1865.66,1867.0,1868.33,1869.66,1871.0,1872.33,1873.66,1875.0,1876.33,1877.66,1879.0,1880.33,1881.66,1883.0,1884.33,1885.66,1887.0,1888.33,1889.66,1891.0,1892.33,1893.66,1895.0,1896.33,1897.66,1899.0,1900.33,1901.66,1903.0,1904.33,1905.66,1907.0,1908.33,1909.66,1911.0,1912.33,1913.66,1915.0,1916.33,1917.66,1919.0,1920.33,1921.66,1923.0,1924.33,1925.66,1927.0,1928.33,1929.66,1931.0,1932.33,1933.66,1935.0,1936.33,1937.66,1939.0,1940.33,1941.66,1943.0,1944.33,1945.66,1947.0,1948.33,1949.66,1951.0,1952.33,1953.66,1955.0,1956.33,1957.66,1959.0,1960.33,1961.66,1963.0,1964.33,1965.66,1967.0,1968.33,1969.66,1971.0,1972.33,1973.66,1975.0,1976.33,1977.66,1979.0,1980.33,1981.66,1983.0,1984.33,1985.66,1987.0,1988.33,1989.66,1991.0,1992.33,1993.66,1995.0,1996.33,1997.66,1999.0,2000.33,2001.66,2003.0,2004.33,2005.66,2007.0,2008.33,2009.66,2011.0,2012.33,2013.66,2015.0,2016.33,2017.66,2019.0,2020.33,2021.66,2023.0,2024.33,2025.66,2027.0,2028.33,2029.66,2031.0,2032.33,2033.66,2035.0,2036.33,2037.66,2039.0,2040.33,2041.66,2043.0,2044.33,2045.66,2047.0,2048.33,2049.66,2051.0,2052.33,2053.66,2055.0,2056.33,2057.66,2059.0,2060.33,2061.66,2063.0,2064.33,2065.66,2067.0,2068.33,2069.66,2071.0,2072.33,2073.66,2075.0,2076.33,2077.66,2079.0,2080.33,2081.66,2083.0,2084.33,2085.66,2087.0,2088.33,2089.66,2091.0,2092.33,2093.66,2095.0,2096.33,2097.66,2099.0,2100.33,2101.66,2103.0,2104.33,2105.66,2107.0,2108.33,2109.66,2111.0,2112.33,2113.66,2115.0,2116.33,2117.66,2119.0,2120.33,2121.66,2123.0,2124.33,2125.66,2127.0,2128.33,2129.66,2131.0,2132.33,2133
```

R code Exa 12.2 Wire Bond Strength With Matrix Notation

```
1 #install.packages("matlib")
2 library(matlib)
3
4 #Wire Bond Strength With Matrix Notation(Pg no. 485)
5
6
7 x1 = c
      (2,8,11,10,8,4,2,2,9,8,4,11,12,2,4,4,20,1,10,15,15,16,17,6,5)
8
9 x2 = c
      (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,600,
10
11
12 x3 = c(rep.int(1,25))
13 y = c
      (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,
14
15
16
17 X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow =
      FALSE)
18
19 Y = matrix(c(y), nrow = 25, ncol = 1, byrow = FALSE)
20
21 X_t = t(X)
22
23 XX_t = X_t %*% X
24
25
26
27 XX_t
28
29 Xt_y = X_t %*% Y
30
31
32
33 Xt_y
34
35
36
37 beta = inv(XX_t) %*% Xt_y
38
39
40
41 cat("the fitted regression model is y =",beta[1,1],"
      +",beta[2,1],"x1 +",beta[3,1],"x2")
42
43 y0 = beta[1,1] + beta[2,1]*2 + beta[3,1]*50
```

```

29 y0
30 y1 = 9.95
31
32 e = y1 - y0
33 e = round(e, digits = 2)
34 e
35 # The answer may slightly vary due to rounding off
    values

```

R code Exa 12.3 Wire Bond Strength ANOVA

```

1 #Wire Bond Strength ANOVA(Pg no. 498)
2
3 wire_lt = c
      (2,8,11,10,8,4,2,2,9,8,4,11,12,2,4,4,20,1,10,15,15,16,17,6,5)
4
4 die_ht = c
      (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,600,
5
5 pull_strgth = c
      (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,
6
6
7 df = data.frame('pull_strength' = pull_strgth, 'wire
      _length' = wire_lt, 'die_height' = die_ht)
8 model = lm(pull_strength~wire_length+die_height,
      data = df)
9 model
10 ans = aov(model)
11 summary(ans)

```

R code Exa 12.4 Wire Bond Strength Coefficient Test

```

1 #install.packages("matlib")
2 library(matlib)
3 #Wire Bond Strength Coefficient Test(Pg no. 500)
4
5 foo= function()
6 {
7     x1 = c
8         (2,8,11,10,8,4,2,2,9,8,4,11,12,2,4,4,20,1,10,15,15,16,17,6,5)
9
10    x2 = c
11        (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,
12
13    x3 = c(rep.int(1,25))
14    y = c
15        (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.0
16
17    X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow
18        = FALSE)
19    Y = matrix(c(y), nrow = 25, ncol = 1, byrow =
20        FALSE)
21    X_t = t(X)
22    XX_t = X_t %*% X
23    Xt_y = X_t %*% Y
24    beta = inv(XX_t) %*% Xt_y
25
26    sample_var = 5.2352
27    inv_XtX = inv(X_t%*%X)
28    inv_XtX
29
30    C22 = inv_XtX[3,3]
31
32    statistic = beta[3,1]/sqrt(sample_var*C22)
33    statistic
34
35    compare = qt(0.025,22,lower.tail = FALSE)
36    compare
37
38
39
40

```

```

31  if(statistic>compare)
32  {
33      print("We reject the null hypothesis and
           conclude that the variable x2 (die height)
           contributes significantly to the model")
34  }
35  else
36  {
37      print("Unable to reject the null hypothesis")
38  }
39  }
40  foo()

```

R code Exa 12.5 Wire Bond Strength One Sided Coefficient Test

```

1  #Wire Bond Strength One-Sided Coefficient Test(Pg no
   . 501)
2
3  foo = function()
4  {
5      beta2 = 0.012528
6      std_err = 0.002793
7
8      statistic = (beta2 - 0.01)/std_err
9      statistic
10
11     t0 = qt(0.25,22,lower.tail = FALSE)
12     t0
13     t1 = qt(0.10,22,lower.tail = FALSE)
14     t1
15
16     p_max = 1 - pnorm(t0,0,1)
17     p_min = 1 - pnorm(t1,0,1)
18
19     p = 1 - pnorm(statistic,0,1)

```

```

20     p
21
22     if(p_min<p && p<p_max)
23     {
24         print("One cannot conclude that the coefficient
                exceeds 0.01 at common levels of significance
                ")
25     }
26     else
27     {
28         print("One can conclude that the coefficient
                exceeds 0.01 at common levels of significance
                ")
29     }
30 }
31 foo()

```

R code Exa 12.6 Wire Bond Strength General Regression Test

```

1 #Wire Bond Strength General Regression Test(Pg no.
  502)
2
3 SSr1 = 6024.0 #4 degrees of freedom
4 SSr2 = SSr1 - 5990.8 #2 degrees of freedom
5 MSe = 4.1
6
7 statistic = (SSr2/2)/MSe
8 statistic = round(statistic, digits = 2)
9 statistic

```

R code Exa 12.7 Wire Bond Strength Confidence Interval

```

1 #install.packages("matlib")

```



```

2 library(matlib)
3
4 #Wire Bond Strength Confidence Interval(Pg no. 507)
5
6 x1 = c
      (2,8,11,10,8,4,2,2,9,8,4,11,12,2,4,4,20,1,10,15,15,16,17,6,5)
7
8 x2 = c
      (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,600,
9
10
11 x3 = c(rep.int(1,25))
12 y = c
      (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,
13
14
15 X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow =
      FALSE)
16 Y = matrix(c(y), nrow = 25, ncol = 1, byrow = FALSE)
17
18 X_t = t(X)
19 XX_t = X_t %*% X
20 Xt_y = X_t %*% Y
21 XtX = X_t%*%X
22 inv_XtX = inv(XtX)
23 beta = inv(XX_t) %*% Xt_y
24
25 C11 = inv_XtX[2,2]
26 sample_var = 5.2352
27 t = qt(0.025,22,lower.tail = FALSE)
28
29 min = beta[2,1] - (t*sqrt(sample_var*C11))
30 max = beta[2,1] + (t*sqrt(sample_var*C11))
31
32 cat("the interval is(",min,"",max,"")")
33
34 # The answer may slightly vary due to rounding off
   values

```

R code Exa 12.8 Wire Bond Strength Confidence Interval on the Mean Response

```
1 #install.packages("matlib")
2 library(matlib)
3
4 #Wire Bond Strength Confidence Interval on the Mean
  Response(Pgno. 508)
5
6 x1 = c
      (2,8,11,10,8,4,2,2,9,8,4,11,12,2,4,4,20,1,10,15,15,16,17,6,5)
7
7 x2 = c
      (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,600,
8
8 x3 = c(rep.int(1,25))
9 y = c
      (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,
10
11 X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow =
      FALSE)
12 Y = matrix(c(y), nrow = 25, ncol = 1, byrow = FALSE)
13
14 X_t = t(X)
15 XX_t = X_t %*% X
16 Xt_y = X_t %*% Y
17 beta = inv(XX_t) %*% Xt_y
18
19 sample_var = 5.2352
20 x0 = matrix(c(1,8,275), nrow = 3, ncol = 1, byrow =
      FALSE)
21 u = t(x0)%*%beta
22
```

```

23 u_var = sample_var * ((t(x0))%%inv(X_t%%X)%%x0)
24 u_var
25
26 t = qt(0.025,22,lower.tail = FALSE)
27
28 min = u - (t*sqrt(u_var))
29 max = u + (t*sqrt(u_var))
30
31 cat("the interval is(",min,"",max,"")")
32
33 # The answer may slightly vary due to rounding off
    values

```

R code Exa 12.9 Wire Bond Strength Confidence Interval

```

1 #install.packages("matlib")
2 library(matlib)
3 #Wire Bond Strength Confidence Interval
4
5 x1 = c
      (2,8,11,10,8,4,2,2,9,8,4,11,12,2,4,4,20,1,10,15,15,16,17,6,5)
6
7 x2 = c
      (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,600,
7
8 y = c
      (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,
9
10 X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow =
      FALSE)
11 Y = matrix(c(y), nrow = 25, ncol = 1, byrow = FALSE)
12
13 X_t = t(X)

```

```

14 XX_t = X_t %*% X
15 Xt_y = X_t %*% Y
16 XtX = X_t%*%X
17 inv_XtX = inv(XtX)
18 beta = inv(XX_t) %*% Xt_y
19
20 sample_var = 5.2352
21 x0 = matrix(c(1,8,275), nrow = 3, ncol = 1, byrow =
      FALSE)
22 y0 = t(x0)%*%beta
23
24 y0_var = sample_var * (1+((t(x0))%*%inv(X_t%*%X)%*%
      x0))
25 y0_var
26
27 t = qt(0.025,22,lower.tail = FALSE)
28
29 min = y0 - (t*sqrt(y0_var))
30 max = y0 + (t*sqrt(y0_var))
31
32 cat("the interval is(",min,"",max,"")")
33
34 # The answer may slightly vary due to rounding off
    of beta values

```

R code Exa 12.11 Wire Bond Strength Confidence Interval

```

1 #install.packages("matlib")
2 library(matlib)
3 #Wire Bond Strength Confidence Interval(Pg no. 515)
4
5 x1 = c
      (2,8,11,10,8,4,2,2,9,8,4,11,12,2,4,4,20,1,10,15,15,16,17,6,5)
6
6 x2 = c

```

```

(50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,600,
7 x3 = c(rep.int(1,25))
8 y = c
  (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,
9
10 X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow =
  FALSE)
11 Y = matrix(c(y), nrow = 25, ncol = 1, byrow = FALSE)
12
13 X_t = t(X)
14
15 H = X%%inv(X_t%%X)%%X_t
16 sample_var = 5.2352
17 e1 = 1.57
18 r1 = e1/sqrt(sample_var*(1-H[1,1]))
19 p = 3
20 MSe = 5.2352
21
22 D1 = ((r1^2)/p)*(H[1,1]/(1-H[1,1]))
23 D1 = round(D1, digits = 3)
24 D1

```

R code Exa 12.12 Airplane Sidewall Panels

```

1 #install.packages("matlib")
2 library(matlib)
3
4 #Airplane Sidewall Panels(Pg no. 518)
5
6 x1 = c(20,25,30,35,40,50,60,65,70,75,80,90)
7 x2 = c
  (400,625,900,1225,1600,2500,3600,4225,4900,5625,6400,8100)

```

```

8 x3 = c(rep.int(1,12))
9 y = c
    (1.81,1.70,1.65,1.55,1.48,1.40,1.30,1.26,1.24,1.21,1.20,1.18)

10
11 X = matrix(c(x3,x1,x2),nrow = 12, ncol = 3, byrow =
    FALSE)
12 Y = matrix(c(y), nrow = 12, ncol = 1, byrow = FALSE)
13
14 X_t = t(X)
15
16 XX_t = X_t %*% X
17
18 XX_t
19
20 Xt_y = X_t %*% Y
21
22 Xt_y
23
24 beta = inv(XX_t) %*% Xt_y
25 beta
26
27 cat("the fitted regression model is y =",beta[1,1],
    ",beta[2,1], "x +",-beta[3,1], "x^2")

```

R code Exa 12.13 Surface Finish

```

1 #Surface Finish(Pg no. 520)
2
3 x2 = c
    (225,200,250,245,235,237,265,259,221,218,224,212,248,260,243,238,
4 y = c
    (45.44,42.03,50.10,48.75,47.92,47.79,52.26,50.52,45.58,44.78,33.5

```

```
5 tool = c(rep.int(302,10),rep.int(416,10))
6
7 df = data.frame("surface" = y, "rpm" = x2, "tool" =
    tool)
8 df$tool.f = factor(df$tool)
9 is.factor(df$tool.f)
10 df$tool.f[1:20]
11 model = lm(surface ~ rpm+tool.f, data = df)
12
13 cat("the fitted regression model is y =",coef(model)
    [1],"",coef(model)[2],"x +",coef(model)[3],"x^2")
```

Chapter 13

Design and Analysis of Single Factor Experiments The Analysis of Variance

R code Exa 13.1 Tensile Strength ANOVA

```
1 #Tensile Strength ANOVA(Pg no. 546)
2
3 foo = function()
4 {
5   obs1 = c(7,12,14,19)
6   obs2 = c(8,17,18,25)
7   obs3 = c(15,13,19,22)
8   obs4 = c(11,18,17,23)
9   obs5 = c(9,19,16,18)
10  obs6 = c(10,15,18,20)
11
12  N = 24
13  n = 6
14
15  M = matrix(c(obs1,obs2,obs3,obs4,obs5,obs6), nrow
              = 4, ncol = 6, byrow = FALSE)
16  M
```



```

17  y_total = sum(M)
18
19  sum_y_sqr = sum(M^2)
20
21  SSt = sum_y_sqr - y_total^2/N
22  SSt
23
24  sum_row1 = sum(M[1,])
25  sum_row2 = sum(M[2,])
26  sum_row3 = sum(M[3,])
27  sum_row4 = sum(M[4,])
28
29  SStreatments = ((sum_row1^2+sum_row2^2+sum_row3^2+
    sum_row4^2)/n) - y_total^2/N
30  SStreatments
31
32  SSe = SSt - SStreatments
33  SSe
34
35  f = qf(0.01, df1=3, df2=20, lower.tail = FALSE)
36  f0 = (SStreatments/3)/(SSe/20)
37
38  if(f<f0)
39  {
40      print("Null hypothesis is rejected and conclude
        that hardwood concentration in the pulp
        significantly affects the mean strength of
        the paper")
41  }
42  else
43  {
44      print("Unable to reject the null hypothesis")
45  }
46 }
47 foo()
48
49 concn = c(5,10,15,20)
50 obs1 = c(7,12,14,19)

```

```

51 obs2 = c(8,17,18,25)
52 obs3 = c(15,13,19,22)
53 obs4 = c(11,18,17,23)
54 obs5 = c(9,19,16,18)
55 obs6 = c(10,15,18,20)
56
57 df = data.frame("concn" = concn, "obs1" = obs1, "
    obs2" = obs2, "obs3" = obs3, "obs4" = obs4, "obs5
    " = obs5, "obs6" = obs6)
58 df$concn = factor(df$concn)

```

R code Exa 13.2 Hardwood concentration experiment

```

1 #(Pg no. 549)
2
3 obs1 = c(7,12,14,19)
4 obs2 = c(8,17,18,25)
5 obs3 = c(15,13,19,22)
6 obs4 = c(11,18,17,23)
7 obs5 = c(9,19,16,18)
8 obs6 = c(10,15,18,20)
9
10 M = matrix(c(obs1,obs2,obs3,obs4,obs5,obs6), nrow =
    4, ncol = 6, byrow = FALSE)
11
12 sum_row1 = sum(M[1,])
13 sum_row2 = sum(M[2,])
14 sum_row3 = sum(M[3,])
15 sum_row4 = sum(M[4,])
16 a = 4
17 n = 6
18 MSe = 6.51
19 t = qt(0.025,20,lower.tail = FALSE)
20 t
21 y1 = sum_row1/n

```

```

22 y2 = sum_row2/n
23 y3 = sum_row3/n
24 y4 = sum_row4/n
25
26 LSD = t*(sqrt((2*MSe)/n))
27 LSD
28
29 cat("4 vs 1 =",y4-y1,">",LSD)
30 cat("4 vs 2 =",y4-y2,">",LSD)
31 cat("4 vs 3 =",y4-y3,">",LSD)
32 cat("3 vs 1 =",y3-y1,">",LSD)
33 cat("3 vs 2 =",y3-y2,">",LSD)
34 cat("2 vs 1 =",y2-y1,">",LSD)

```

R code Exa 13.4 Textile Manufacturing

```

1
2 #Textile Manufacturing(Pg no. 560)
3 obs1 = c(98,91,96,95)
4 obs2 = c(97,90,95,96)
5 obs3 = c(99,93,97,99)
6 obs4 = c(96,92,95,98)
7
8 data = rbind(obs1,obs2,obs3,obs4)
9
10 tag = c("obs1","obs2","obs3","obs4")
11
12 M = c(t(as.matrix(data)))
13 n_row = 4
14 n_col = 4
15
16 op <- gl(n_row,1,n_col*n_row,factor(tag))
17
18 m <- gl(n_col,n_row,n_row*n_col)
19

```

```
20 summary(aov(M~op+m))
```

R code Exa 13.5 Fabric Strength

```
1 #Fabric Strength(Pg no. 565)
2
3 obs1 = c(1.3,2.2,1.8,3.9)
4 obs2 = c(1.6,2.4,1.7,4.4)
5 obs3 = c(0.5,0.4,0.6,2.0)
6 obs4 = c(1.2,2.0,1.5,4.1)
7 obs5 = c(1.1,1.8,1.3,3.4)
8
9 data = rbind(obs1,obs2,obs3,obs4,obs5)
10
11 tag = c("obs1","obs2","obs3","obs4","obs5")
12
13 M = c(t(as.matrix(data)))
14 n_row <- 4
15 n_col <- 5
16
17 op <- gl(n_row,1,n_col*n_row,factor(tag))
18
19 m <- gl(n_col,n_row,n_row*n_col)
20
21
22 summary(aov(M~op+m))
23
24 # The answer may slightly vary due to rounding off
    values
```

Chapter 14

Design of Experiments with Several Factors

R code Exa 14.1 Aircraft Primer Paint

```
1 #Aircraft Primer Paint(pg no. 586)
2
3 foo = function()
4 {
5   n = 3
6   a = 3
7   b = 2
8   dripping = matrix(c
9     (4.0,4.5,4.3,5.6,4.9,5.4,3.8,3.7,4.0),nrow = 3,
10    ncol = 3,byrow = TRUE)
11   spraying = matrix(c
12     (5.4,4.9,5.6,5.8,6.1,6.3,5.5,5.0,5.0),nrow = 3,
13    ncol = 3,byrow = TRUE)
14   types = c(28.7,34.1,27.0)
15   methods = c(40.2,49.6)
16   interaction = c(12.8, 15.9, 15.9, 18.2, 11.5,
17     15.5)
18
19   total = sum(dripping) + sum(spraying)
```

```

15
16 Yijk = sum(dripping^2) + sum(spraying^2)
17 SSt = Yijk - (total^2/(a*b*n))
18 cat("SSt =", SSt, "\n")
19
20 Yi = sum(types^2)
21 SStypes = (Yi/(b*n)) - (total^2/(a*b*n))
22 cat("SStypes =", SStypes, "\n")
23
24 Yj = sum(methods^2)
25 SSmethods = (Yj/(a*n)) - (total^2/(a*b*n))
26 cat("SSmethods =", SSmethods, "\n")
27
28 Yij = sum(interaction^2)
29 SSinteraction = (Yij/n) - (total^2/(a*b*n)) -
    SStypes - SSmethods
30 cat("SSinteraction =", SSinteraction, "\n")
31
32 SSe = SSt - SStypes - SSmethods - SSinteraction
33 cat("SSe =", SSe, "\n")
34
35 f2 = qf(0.05, df1=2, df2=12, lower.tail = FALSE)
36 f1 = qf(0.05, df1=1, df2=12, lower.tail = FALSE)
37
38 f0_types = (SStypes/2)/(SSe/12)
39 f0_methods = (SSmethods/1)/(SSe/12)
40 f0_interaction = (SSinteraction/2)/(SSe/12)
41
42 if(f2 < f0_types)
43 {
44     print("the main effects of primer type affect
        adhesion force")
45 }
46 else
47 {
48     print("the main effects of primer type do not
        affect adhesion force")
49 }

```

```

50
51  if(f1<f0_methods)
52  {
53      print("the main effects of application method
          affect adhesion force")
54  }
55  else
56  {
57      print("the main effects of application method do
          not affect adhesion force")
58  }
59
60  if(f2<f0_interaction)
61  {
62      print("there is indication of interaction
          between these factors")
63  }
64  else
65  {
66      print("there is no indication of interaction
          between these factors")
67  }
68 }
69 foo()

```

R code Exa 14.3 Epitaxial Process

```

1  #Epitaxial Process(Pg no. 596)
2
3  obs1 = c(14.037,14.165,13.972,13.907)
4  obs2 = c(14.821,14.757,14.843,14.878)
5  obs3 = c(13.880,13.860,14.032,13.914)
6  obs4 = c(14.888,14.921,14.415,14.932)
7
8  M = matrix(c(obs1,obs2,obs3,obs4), nrow = 4, ncol =

```

```

      4, byrow = TRUE)
9
10 sum_y_sqr = sum(M^2)
11
12 init = sum(M[1,])
13 a = sum(M[2,])
14 b = sum(M[3,])
15 ab = sum(M[4,])
16
17 n = 4
18 N = 16
19
20 A = (1/(2*n)) * (a+ab-b-init)
21 B = (1/(2*n)) * (b+ab-a-init)
22 AB = (1/(2*n)) * (ab+init-a-b)
23 SSa = ((a+ab-b-init)^2)/N
24 SSa
25 SSb = ((b+ab-a-init)^2)/N
26 SSb
27 SSab = ((ab+init-a-b)^2)/N
28 SSab
29 SSt = sum_y_sqr - ((init+a+b+ab)^2)/N
30 SSt

```

R code Exa 14.4 Surface Roughness

```

1 #Surface Roughness(Pg no. 604)
2
3 n=2
4 k=3
5 init=9+7
6 a=10+12
7 b=9+11
8 ab=12+15
9 c=11+10

```



```

10 ac=10+13
11 bc=10+8
12 abc=16+14
13
14 A = (1/(4*n)) * (a+ab+ac+abc-init-b-c-bc)
15 A
16 SSa = ((a+ab+ac+abc-init-b-c-bc)^2)/(n*(2^k))
17 SSa

```

R code Exa 14.5 Plasma Etch

```

1 #Plasma Etch(Pg no. 607)
2
3 A = (1/8) * (669 + 650 + 642 + 635 + 749 + 868 + 860
      + 729-550-604-633-601-1037-1052 - 1075 - 1063)
4 A
5
6 cat("Because A = ",-101.625,"the effect of
      increasing the gap between the cathode and anode
      is to decrease the etch rate")

```

R code Exa 14.6 Process Yield

```

1 #Process Yield(Pg no. 612)
2
3 yc = c(40.3,40.5,40.7,40.2,40.6)
4 yf = c(40.0,41.5,39.3,40.9)
5
6 ycbar = sum(yc)/length(yc)
7 yfbar = sum(yf)/length(yf)
8 nf = length(yf)
9 nc = length(yc)
10 SSe = 0.0

```

```

11 for(i in 1:nc)
12 {
13   SSe = SSe + (yc[i] - ycbar)^2
14 }
15 SSe
16 MSe = SSe/(nc-1)
17 MSe
18
19 SScurvature = (nc*nf*((yfbar-ycbar)^2))/(nc+nf)
20 SScurvature
21
22 statistic = ((yfbar-ycbar))/sqrt(MSe*((1/nc)+(1/nf)))
23 )
24 statistic = round(statistic, digits = 2)
25 statistic

```

R code Exa 14.8 Plasma Etch

```

1 #Plasma Etch(Pg no. 629)
2
3 init = 550
4 ad = 749
5 bd = 1052
6 ab = 650
7 cd = 1075
8 ac = 642
9 bc = 601
10 abcd = 729
11
12 lA = (1/4) * (-init+ad-bd+ab-cd+ac-bc+abcd)
13 lA
14 lB = (1/4) * (-init-ad+bd+ab-cd-ac+bc+abcd)
15 lB
16 lC = (1/4) * (-init-ad-bd-ab+cd+ac+bc+abcd)
17 lC

```

```
18 1D = (1/4) * (-init+ad+bd-ab+cd-ac-bc+abcd)
19 1D
20
21 1AB = (1/4) * (init-ad-bd+ab+cd-ac-bc+abcd)
22 1AB
23 1AC = (1/4) * (init-ad+bd-ab-cd+ac-bc+abcd)
24 1AC
25 1AD = (1/4) * (init+ad-bd-ab-cd-ac+bc+abcd)
26 1AD
```

Chapter 15

Statistical Quality Control

R code Exa 15.1 Vane Opening

```
1 #Vane Opening(Pg no. 677)
2
3 xbar = c
      (31.6,33.4,35.0,32.2,33.8,38.4,31.6,36.8,35.0,34.0,29.8,34.0,33.0
4
5
6 Xbar = sum(xbar)/20
7 rbar = sum(r)/20
8 n = 5
9 A2 = 0.577
10 UCL = Xbar + A2*rbar
11 LCL = Xbar - A2*rbar
12 cat("the trial control limits for the Xbar chart are
      UCL = ",round(UCL,digits = 2),"LCL = ",round(LCL
      ,digits = 2))
13
14 D4 = 2.115
15 D3 = 0
16
17 cat("the trial control limits for the rbar chart are
```

```

      UCl = ",round(D4*rbar, digits = 2),"LCl = ",(D3*
rbar))
18
19 s = c
      (1.67332,2.60768,1.58114,1.64317,0.83666,1.14018,1.51658,4.38178,

20 sbar = sum(s)/20
21 c4 = 0.94
22
23 lim = ((3*sbar)/c4)*sqrt(1-(c4^2))
24
25 cat("the trial control limits for the sbar chart are
      UCl = ",(sbar+lim),"LCl = ",(sbar-lim))
26
27 lim4 = Xbar + (3*sbar)/(c4*sqrt(n))
28 lim3 = Xbar - (3*sbar)/(c4*sqrt(n))
29
30 cat("If sbar is used to determine the control limits
      for the Xbar chart then limits are(",lim3,"",
      lim4,")")
31
32 Xbar = 33.21
33 rbar = 5.0
34 UCL = Xbar + A2*rbar
35 LCL = Xbar - A2*rbar
36 cat("new revised limits for the Xbar chart are UCl =
      ",round(UCL, digits = 2),"LCl = ",round(LCL,
      digits = 2))

```

R code Exa 15.2 Chemical Process Concentration

```

1 #Chemical Process Concentration(Pg no. 685)
2
3 x = c
      (102.0,94.8,98.3,98.4,102.0,98.5,99.0,97.7,100.0,98.1,101.3,98.7,

```

```

4  mr = c
      (7.2,3.5,0.1,3.6,3.5,0.5,1.3,2.3,1.9,3.2,2.6,2.4,2.7,1.4,0.3,3.6,
5
6  xbar = sum(x)/length(x)
7  xbar
8
9  mrbar = sum(mr)/length(mr)
10 mrbar
11
12 d2 = 1.128
13
14 UCL = xbar + (3*(mrbar/d2))
15 cat("UCL =",UCL)
16
17 LCL = xbar - (3*(mrbar/d2))
18 cat("LCL =",LCL)
19
20 CL = xbar
21 cat("CL =",CL)
22
23 # The answer may slightly vary due to rounding off
    values

```

R code Exa 15.3 Electrical Current

```

1  #Electrical Current(Pg no. 692)
2
3  x = 110
4  mean = 107.0
5  std_dev = 1.5
6
7  PCR = (x-90)/(6*std_dev)
8  cat("PCR =",PCR)

```

```

9 PCRk = (x-mean)/(3*std_dev)
10 cat("PCRk =",PCRk)
11
12 LSL = (90-mean)/std_dev
13 USL = (x - mean)/std_dev
14
15 #P(X<LSL)
16 z1 = pnorm(LSL,0,1)
17 z1 = round(z1)
18 z1
19
20 #P(X>USL)
21 z2 = pnorm(USL,0,1,lower.tail = FALSE)
22 z2 = round(z2 , digits = 3)
23 z2

```

R code Exa 15.4 Ceramic Substrate

```

1 #Ceramic Substrate(Pg no. 696)
2
3 defectives = c
      (44,48,32,50,29,31,46,52,44,48,36,52,35,41,42,30,46,38,26,30)
4
5 samples = 20
6 size = 100
7 total_size = samples*size
8
9 pbar = sum(defectives)/total_size
10 pbar
11
12 UCL = pbar + (3*sqrt((pbar*(1-pbar))/size))
13
14 LCL = pbar - (3*sqrt((pbar*(1-pbar))/size))
15 cat("UCL =",UCL)
16
17 cat("LCL =",LCL)

```

```

16
17 CL = pbar
18 cat("CL =", CL)
19
20 # The answer may slightly vary due to rounding off
    values

```

R code Exa 15.5 Printed Circuit Boards

```

1 #Printed Circuit Boards(Pg no. 697)
2
3 defects = c
      (6,4,8,10,9,12,16,2,3,10,9,15,8,10,8,2,7,1,7,13)
4 samples = 20
5 size = 5
6
7 defects_p_unit = defects/size
8 ubar = sum(defects_p_unit)/samples
9 ubar
10
11 UCL = ubar + 3*sqrt(ubar/size)
12 cat("UCL =", UCL)
13
14 LCL = ubar - 3*sqrt(ubar/size)
15 cat("LCL =", LCL, "< 0")

```

R code Exa 15.6 Chemical Process Concentration CUSUM Tabular Cusum

```

1 #Chemical Process Concentration CUSUM Tabular Cusum(
    Pg no. 706)
2

```



```

3  obs = c
      (102.0,94.8,98.3,98.4,102.0,98.5,99.0,97.7,100.0,98.1,101.3,98.7,
4  Sh_0 = 0
5  Sl_0 = 0
6  u0 = 99
7  k = 1
8
9  Sh_1 = max(0, obs[1]-(u0+k)+Sh_0)
10 Sh_1
11 Sl_1 = max(0, (u0-k)-obs[1]+Sh_0)
12 Sl_1

```

R code Exa 15.7 Chemical Process Concentration EWMA

```

1  #Chemical Process Concentration EWMA(Pg no. 711)
2
3  lambda = 0.2
4  n = 1
5  xbar = 99.1
6  mrbar = 2.59
7  u0 = 99.1
8  sigma = 2.59/1.128
9
10 LCL = u0 - (3*(sigma/sqrt(n))*sqrt((lambda/(2-lambda
      ))*(1-(1-lambda)^(2))))
11 UCL = u0 + (3*(sigma/sqrt(n))*sqrt((lambda/(2-lambda
      ))*(1-(1-lambda)^(2))))
12 cat("The control limits for z1 are LCL=",LCL,"UCL ="
      ,UCL)
13
14 # The answer may slightly vary due to rounding off
      values

```
