R Textbook Companion for Applied Statistics and Probability for Engineers by Douglas C. Montgomery and George C. Runger[1]

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

June 4, 2020

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

Book Description

Title: 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

Chapter 2

Probability

R code Exa 2.1 Camera Flash

```
#Camera flash(Pg no. 18)
foo<-function(){
  min_recycle_time = 1.5
  max_recycle_time = 5
  x<-readline(prompt="Enter 1: " )
  if(min_recycle_time<x && x<max_recycle_time)
  {
    print("Outcome of Set S={low, medium,high} is medium")
    print("Outcome of Set S={yes, no} is yes")
  } else if(min_recycle_time >= x)
  {
    print("Outcome of Set S={low, medium,high} is low")
    print("Outcome of Set S={yes, no} is no")
  } else if(max_recycle_time <= x)
  {
    print("Outcome of Set S={low, medium,high} is high")
    print("Outcome of Set S={yes, no} is no")
  }
```

```
}

foo()
```

R code Exa 2.4 Automobile Options

```
#Automobile options(Pg no. 19)
foo<-function(){
  number_of_possible_outcomes_of_automatic_
     transmission = 2
  #Set S = {With an automatic transmission , Without
     an automatic transmission}

  number_of_possible_outcomes_of_sunroof = 2
  #Set S = {With a sunroof , Without a sunroof}

  number_of_possible_outcomes_of_sterio_system = 3
  #Set S = {Sterio system 1, Sterio system 2, Sterio
      system 3}

  number_of_possible_outcomes_of_exterior_color = 4
  #Set S = {Exterior color 1, Exterior color 2,
     Exterior color 3, Exterior color 4}

  total_no_of_outcome = number_of_possible_outcomes_
     of_automatic_transmission*
    number_of_possible_outcomes_of_sunroof*
    number_of_possible_outcomes_of_sterio_system*
    number_of_possible_outcomes_of_exterior_color

  cat("The sample space contains",total_no_of_
     outcome,"outcomes")
}

foo()
```

R code Exa 2.5 Automobile Colors

```
#Automobile Colors(Pg no. 20)
foo<-function(){
  vehicle_types = 12

  exterior_color_black = 2
  #Set S = {black, red}

  exterior_color_white = 4
  #Set S = {black, white, blue, brown}

  exterior_color_blue = 3
  #Set S = {black, red, blue}

  exterior_color_brown = 1
  #Set S = {brown}

  total_no_of_outcome = vehicle_types*exterior_color_black+
    vehicle_types*exterior_color_white+
    vehicle_types*exterior_color_blue+
    vehicle_types*exterior_color_brown

  cat("The sample space contains",total_no_of_outcome,"vehicle types")
}

foo()
```

R code Exa 2.8 Hospital Emergency Visits

```
#Hospital Emergency Visits(Pg no. 21)

total = c(5292,6991,5640,4329)
lwbs = c(195,270,246,242)
admitted = c(1277,1558,666,984)
not_admitted = c(3820,5163,4728,3103)
df = data.frame("total" = total, "lwbs" = lwbs, "admitted" = admitted, "not_admitted" = not_admitted)
# A is the event that a visit is to hospital 1
# B is the event that the result of the visit is LWBS
A_intersection_B = df$lwbs[1]
Abar = sum(df$total[2:4])
A_union_B = df$total[1] + sum(df$lwbs[2:4])

cat("A union B =",A_union_B)
cat("A intersection B =",A_intersection_B)
cat("Abar =",Abar)

# The answer given in textbook for A intersection B is wrong
```

R code Exa 2.9 Web site design

```
#Web site design(Pg no. 23)
colors = 4
fonts = 3
image_positions = 3

designs = colors*fonts*image_positions

cat(designs,"different designs are possible")
```

R code Exa 2.10 Printed circuit board

```
#Printed circuit board(Pg no. 24)

locations = 8
components = 4
designs = factorial(locations)/factorial(components)

cat(designs,"different designs are possible")
```

R code Exa 2.11 Hospital schedule

```
#Hospital schedule(Pg no. 24)

knee_surgery = 3
hip_surgery = 2
sequences = factorial(knee_surgery+hip_surgery)/(
   factorial(knee_surgery)*factorial(hip_surgery))

cat(sequences,"sequences are possible")
```

R code Exa 2.12 Bar code 39

```
#Bar code 39(Pg no. 24)

black_bars = 5
narrow_bars = 3
wide_bars = 2
bars = factorial(black_bars)/(factorial(narrow_bars)
   *factorial(wide_bars))
```

```
narrow_spaces = 3
wide_spaces = 1
codes = bars*(narrow_spaces+wide_spaces)

cat("The number of possible codes is",codes)
```

R code Exa 2.13 Printed circuit board layout

```
#Printed circuit board layout(Pg no. 25)

locations = 8
identical_components = 5
designs = factorial(locations)/(factorial(identical_
    components)*factorial(locations-identical_
    components))

cat("Number of possible designs is",designs)
```

R code Exa 2.14 Sampling without replacement

```
#Sampling without replacement(Pg no. 25)

size = 6
total_parts = 50
defective_parts = 3
non_defective_parts = 47

step_1 = choose(defective_parts,2)

remaining_parts = size - 2

```

```
step_2 = choose(non_defective_parts, remaining_parts
   )

num_of_subsets = step_1*step_2

cat("the number of subsets of size 6 that contain
   exactly 2 defective parts is",num_of_subsets)

diff_subsets = choose(total_parts, size)

cat("total number of different subsets of size 6 is"
   ,diff_subsets)
```

R code Exa 2.15 Laser diodes

```
#Laser diodes(Pg no. 30)

num_of_diodes = 30
individual_probability = 0.01

total_probability = num_of_diodes*individual_
   probability

cat("Probability of E is", total_probability)
```

R code Exa 2.16 Probabilities of Events

```
#Probabilities of Events(Pg no. 31)

a = 0.1
b = 0.3
c = 0.5
d = 0.1
```

```
A = a + b
B = b + c + d
C = d

cat("P(A) =",A)
cat("P(B) =",B)
cat("P(C) =",C)

cat("P(A_bar) =",1-A)
cat("P(B_bar) =",1-B)
cat("P(C_bar) =",1-C)

cat("P(A_intersection_B) =",b)
cat("P(A_union_B) =",a+b+c+d)
cat("P(A_intersection_c) =",0)
```

R code Exa 2.17 Contamination particles

```
#Contamination particles(Pg no. 31)

zero_contaminant = 0.4
three_contaminants = 0.10
four_contaminants = 0.05
five_and_above_contaminants = 0.10

E = three_contaminants+four_contaminants+five_and_
    above_contaminants

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

```
#Refer to code EX2_14(Sampling without replacement)

#Manufacturing Inspection(Pg no. 32)

size = 6
non_defective_parts = 47
num_of_subsets = 535095
diff_subsets = 15890700

P1 = num_of_subsets/diff_subsets

#subset with no defective parts
subset2 = choose(non_defective_parts,size)

P2 = subset2/diff_subsets

cat("Probability that a sample contains exactly 2
    defective parts is",P1)
cat("Probability that no defective parts are
    selected is",P2)
#The answer might slightly vary due to rounding off
    values
```

R code Exa 2.19 Semiconductor Wafers

```
#install.packages("MASS")
library(MASS)

#Semiconductor Wafers(Pg no. 35)

total = 940
high = 358
center = 626
```

```
high_and_center = 112
high_or_center = (high+center-high_and_center)
ans = high_or_center/total
ans = fractions(ans)
print(ans)

#The answer might slightly vary due to reducing of values
```

R code Exa 2.20 Semiconductor wafers and location

```
#Semiconductor wafers and location(Pg no. 36)

E1 = 0.15 #probability that a wafer contains four or more particles
E2 = 0.28 #probability that a wafer was at the edge
E1_and_E2 = 0.04
E1_or_E2 = E1+E2-E1_and_E2

E3 = 0.60 #probability that a wafer contains less than two particles
E4 = 0.03 #probability that a wafer is both at the edge and contains more than four particles
E3_and_E4 = 0 #mutually exclusive
E3_or_E4 = E3+E4-E3_and_E4

cat("probability of E1 union E2 is",E1_or_E2)
cat("probability of E3 union E4 is",E3_or_E4)
```

R code Exa 2.22 Surface flaws and defectives

```
#Surface flaws and defectives(Pg no. 40)

```

```
f1 = 40 #total_surface flaws
d1 = 10 #defectiv and surface flawed
P1 = d1/f1

d2 = 18 #defective but not surface flawed
f2 = 400 - f1 #without surface flawed
P2 = d2/f2

cat("conditional probability P(d1|f1) is",P1)
cat("conditional probability P(d2|f2) is",P2)
```

R code Exa 2.23 Tree Diagram

```
#install.packages("MASS")
library(MASS)

#Tree Diagram(Pg no. 41)

D_and_F = 10/400
F = 40/400
D = 28/400
Dbar_and_F = 30/400
D_and_Fbar = 18/400
Dbar_and_Fbar = 342/400
Fbar = 1-F

D_given_F = D_and_F/F
D_given_F = fractions(D_given_F)
print("P(D|F) is")
print(D_given_F)

F_given_D = D_and_F/D
F_given_D = fractions(F_given_D)
print("P(F|D) is")
print(F_given_D)
```

```
Dbar_given_F = Dbar_and_F/F
Dbar_given_F = fractions(Dbar_given_F)
print("P(Dbar|F) is")
print(Dbar_given_F)

D_given_Fbar = D_and_Fbar/Fbar
D_given_Fbar = fractions(D_given_Fbar)
print("P(D|Fbar) is")
print(D_given_Fbar)

Dbar_given_Fbar = Dbar_and_Fbar/Fbar
Dbar_given_Fbar = fractions(Dbar_given_Fbar)
print("P(Dbar|Fbar) is")
print(Dbar_given_Fbar)

#The answer might slightly vary due to reducing of values
```

R code Exa 2.24 Random inspection

```
#install.packages("MASS")
library(MASS)

#Random inspection(Pg no. 42)

remaining_parts = 49 #total remaining parts
defective_parts = 2 #total defective parts remaining

P = defective_parts/remaining_parts
P = fractions(P)
print(P)
cat("conditional probability P(B|A) is ",P)
```

R code Exa 2.25 Continuation of Chapter 2 Example 24

```
#Continue to EX2_24

#(Pg no. 43)

total = 50
defective_parts = 3
non_defective_parts = 47

p = (defective_parts/total)*((defective_parts-1)/(total-1))*(non_defective_parts/(total-2))

cat(p,"is the probability that the first two parts selected are defective and the third is not defective")

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

R code Exa 2.26 Machine stages

```
#Machine stages(Pg no. 45)

A = 0.90 #probability of that first stage of machining meets specifications
B_given_A = 0.95 #Given A, probability that a second stage of machining meets specifications
A_and_B = B_given_A*A

cat(A_and_B,"is the probability that both stages meet specifications")
```

R code Exa 2.27 Semiconductor contamination

```
#Semiconductor contamination(Pg no. 46)

high = 0.20 #probability of high level of
    contamination
not_high = 1 - high #probability of not high level
    of contamination
product_fail_given_high = 0.10
product_fail_given__not_high = 0.005

product_fail = (product_fail_given_high*high)+(
    product_fail_given__not_high*not_high)

cat(product_fail,"is probability that the product
    fails")
```

R code Exa 2.28 Semiconductor failures

```
#Semiconductor failures(Pg no. 47)

high = 0.20 #probability of high level of
    contamination
medium = 0.30 #probability of medium level of
    contamination
low = 0.50 #probability of low level of
    contamination
product_fail_given_high = 0.10
product_fail_given_medium = 0.01
product_fail_given_low = 0.001

```

```
product_fail = (product_fail_given_high*high)+(
    product_fail_given_medium*medium)+(product_fail_
    given_low*low)

cat(product_fail,"is probability that the product
    fails")
```

R code Exa 2.29 Sampling with replacement

```
#install.packages("MASS")
library(MASS)

#Sampling with replacement(Pg no. 49)

# A : event that first part is defective
# B : event that second part is defective

B_given_A = 3
A = 3
total = 50
P = fractions((B_given_A*A)/(total*total))
print(P)
cat("probability that both parts are defective is",P
    )
```

R code Exa 2.30 Flaws and functions

```
#install.packages("MASS")
library(MASS)

#Flaws and functions(Pg no. 49)

defective_and_surface_flawed = 2
```

```
total_defective_parts = 20
total_surface_flawed = 40
P = fractions((defective_and_surface_flawed*
    defective_and_surface_flawed)/(total_defective_
    parts*total_surface_flawed))
print(P)
cat("probability of surface flawed and defevtive is"
    ,P)

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

R code Exa 2.31 Unconditional Probability

```
#Continuation of Ex2_14

#(Pg no. 50)

B_given_A = 2/49
A = 3/50
B_given_Abar = 3/49
Abar = 1-A

B = (B_given_A*A)+(B_given_Abar*Abar)
cat("P(B) =",B)
```

R code Exa 2.32 Series circuit

```
#Series circuit(Pg no. 51)

l = 0.80 #probability that left device operate
r = 0.90 #probability that right device operate

```

```
l_and_r = l*r

cat("The probability that the circuit operates is",l_and_r)
```

R code Exa 2.33 Semiconductor wafers

```
#Semiconductor wafers(Pg no. 51)

contamination = 0.01
no_contamination = 1- contamination
wafers = 15

total_probability = no_contamination^wafers

cat("The probability that no large particles are found is ",total_probability)

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

R code Exa 2.34 Parallel circuit

```
#Parallel circuit(Pg no. 51)

top = 0.95 #top device does operate
bottom = 0.95 #bottom device does operate

t = 1 - top #top device does not operate
b = 1 - bottom #bottom device does not operate

t_and_b = t*b

```

```
top_or_bottom = 1 - t_and_b

cat("The probability that the circuit operates is",
    top_or_bottom)
```

R code Exa 2.35 Advanced circuit

```
#Advanced circuit(Pg no. 52)

left = 1 - 0.1^3
middle = 1 - 0.05^2
right = 0.99

operates = left*middle*right

cat(operates,"is the probability that the circuit
    operates")
#The answer may slightly vary due to rounding off
    values
```

R code Exa 2.37 Medical Diagnostic

```
#Medical Diagnostic(Pg no. 55)

p_given_i = 0.99 #test signals positive with illness
n_given_i = 0.95 #test signals negative with illness
i = 0.0001 #illness in population

p_given_not_i = 0.05

i_given_p = (p_given_i*i)/((p_given_i*i)+(p_given_
    not_i*(1-i)))

```

```
cat(i_given_p,"is the probability that you have the
    illness given test is positive")
#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

```
#Camera flash tests(Pg no. 66)

pass = 0.8
fail = 1 - pass

ppf = pass*pass*fail

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

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

```
    p = 0.01 #contaminent present
    a = 1 - p #contaminent absent
    x = readline(prompt = "Enter x:")
    x = as.integer(x)
    if (x>=1)
      {
        px = (a^(x-1))*p
        cat("probability P( X =",x,") is ",px)
    }
    else
    {
      cat("Error! input must be >=1")
    }

}

foo()
```

R code Exa 3.6 Digital channel

```
#Digital channel(Pg no. 71)

#probability of error in given number(X) of bits
zero = 0.6561 # P(X=0)
one = 0.2916 # P(X=1)
two = 0.0486 # P(X=2)
three = 0.0036 # P(X=3)


X = zero + one + two + three
Y = zero + one + two

cat("P(X<=3) =",X)
cat("P(X=3) =",X-Y)
```

R code Exa 3.7 Cumulative distribution function

```
#Cumulative distribution function(Pg no. 72)

"
F(x) = 0 -- { x < -2}
F(x) = 0.2 -- { -2 <= x < 0}
F(x) = 0.7 -- { 0<= x < 2}
F(x) = 1 -- { 2 <= x}

f(-2) = ?
f(0) = ?
f(2) = ?
"
P1 = 0.2 - 0.0
P2 = 0.7 - 0.2
P3 = 1.0 - 0.7

cat("f(-2) =",P1)
cat("f(0) =",P2)
cat("f(2) =",P3)
```

R code Exa 3.8 Sampling without replacement

```
#Sampling without replacement(Pg no. 72)

total = 850
unconforming = 50
conforming = total - unconforming

X0 = (conforming/total)*((conforming-1)/(total-1))
X1 = 2*(conforming/total)*(unconforming/(total-1))
```

```
X2 = (unconforming/total)*((unconforming-1)/(total
   -1))

F0 = round(X0, digits = 3)
F0
F1 = round(X0 + X1, digits = 3)
F1
F2 = X0 + X1 + X2
F2
cat(
"
F(x) = 0 -- { x < 0}
F(x) = ",F0," -- { 0 <= x < 1}
F(x) = ",F1," -- { 1 <= x < 2}
F(x) = ",F2," -- { 2 <= x}

")
#The values may slightly vary due to rounding off
   values
```

R code Exa 3.9 Digital channel

```
#Digital channel(Pg no. 75)

error_bits = c(0,1,2,3,4)
probability = c(0.656,0.2916,0.0486,0.0036,0.0001)
df = data.frame('error_bits' = error_bits, '
   probability' = probability)

index = c(1:length(error_bits))

u = weighted.mean(error_bits, probability) #
   expectation

v = weighted.mean((error_bits-u)^2,probability) #
```

```
variance


cat("Expectation E(X) =",round(u, digits = 1))
cat("Variance V(X) =",v)
#The answer may slightly vary due to rounding off
    values
```

R code Exa 3.10 Marketing

```
#Marketing(Pg no. 75)

B_revenue1 = 7 # in million dollars
B_probability1 = 0.3

B_revenue2 = 2 # in million dollars
B_probability2 = 0.7

u = B_revenue1*B_probability1 + B_revenue2*B_
    probability2
#expectation in million dollars

v = ((B_revenue1-u)^2)*B_probability1 + ((B_revenue2
    -u)^2)*B_probability2
#variance in million dollars squared

sd = sqrt(v)

cat("Expectation E(Y) =",u)
cat("Variance V(Y) =",v)
cat("Standard Deviation SD(Y) =",sd)
#The answer may slightly vary due to rounding off
    values
```

R code Exa 3.11 Messages

```
#Messages(Pg no. 76)

  x = c(10,11,12,13,14,15) # number of messages
  probability = c(0.08,0.15,0.30,0.20,0.20,0.07)

  u = weighted.mean(x,probability) #expectation

  v = weighted.mean((x-u)^2,probability)#variance

  sd = sqrt(v)
  cat("Expectation E(X) =",u)
  cat("Variance V(X) =",v)
  cat("Standard Deviation SD(X) =",sd)
#The answer may slightly vary due to rounding off
   values
```

R code Exa 3.12 Digital channel continuation

```
#Digital channel continuation(Pg no. 76)

error_bits = list(0,1,2,3,4)
probability = list
   (0.656,0.2916,0.0486,0.0036,0.0001)

index = c(1:length(error_bits))
u = 0.00 #expectation
for (i in index) {
  u = u + (((as.double(error_bits[i]))^2)*as.double(
     probability[i]))
}
```

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

R code Exa 3.14 Number of voice lines

```
#Number of voice lines(Pg no. 78)

max_lines = 48
min_lines = 0

u = (max_lines+min_lines)/2 #expectation

sd = sqrt((((max_lines-min_lines+1)^2)-1)/12) #
   standard deviation

cat("Expectation E(X) =",u)
cat("Standard Deviation SD(X) =",sd)
#The answer may slightly vary due to rounding off
   values
```

R code Exa 3.15 Proportion of voice lines

```
#Refer to EX3_14(Number of voice lines)

#Proportion of voice lines(Pg no. 79)

u = 24 #expectation E(X)
v = 199.94 #variance Var(X)

u1 = u/48 #expectation E(Y)
v1 = v/(48^2) #variance Var(Y)

cat("Expectation E(Y) =",u1)
```

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

R code Exa 3.16 Digital channel

```
#install.packages("combinat")
library(combinat)

#Digital channel(Pg no. 80)

error = 0.1
not_error = 1 - error

P = error*not_error*error*not_error #probability of
    2 errors

outcomes = dim(combn(4,2))[2]

total_probability = outcomes*P

cat("Probability P(X=2) =",total_probability)
```

R code Exa 3.17 Binomial coefficient

```
#install.packages("combinat")
library(combinat)

#Binomial coefficient(Pg no. 82)


A = dim(combn(10,3))[2]
A
```

```
B = dim(combn(15,10))[2]
B

C = dim(combn(100,4))[2]
C
```

R code Exa 3.18 Organic pollution

```
#Organic pollution(Pg no. 83)

n = 18 #number of samples
p = 0.1

# P(X=2)
A = dbinom(2,n,p)
A

# P(X>=4) == 1 - P(X<4)
B = 1 - pbinom(3,n,p)
B

# P(3<=X && X<7)
C = dbinom(3,n,p) + dbinom(4,n,p) + dbinom(5,n,p) +
    dbinom(6,n,p)
C
```

R code Exa 3.19 Mean and Variance

```
#Mean and Variance(Pg no. 84)

n = 4
p = 0.1
```

```
u = n*p

v = n*p*(1-p)

cat("E(X) =",u)
cat("Var(X) =",v)
```

R code Exa 3.20 Digital channel

```
#Digital channel(Pg no. 87)

error = 0.1
not_error = 1 - error

P = (not_error^4)*error

cat("P(X=5) =",P)
#The answer may slightly vary due to rounding off
    values
```

R code Exa 3.21 Wafer contamination

```
#Wafer contamination(Pg no. 88)

contaminated = 0.01

n = 125 #num of wafers

p = dgeom(n,contaminated)

cat("P(X=125) =",p)
```

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

R code Exa 3.22 Mean and standard deviation

```
#Mean and standard deviation(Pg no. 88)

p = 0.1

u = 1/p

sd = sqrt((1-p)/(p^2))

cat("The mean until the first error is",u)
cat("The standard deviation before the first error is",sd)
#The answer may slightly vary due to rounding off values
```

R code Exa 3.24 Digital Channel

```
#Digital Channel(Pg no. 89)

error_prob = 0.1
n = 10
n_error = 3
ans = dnbinom((n-n_error-1),(n_error+1),error_prob)

cat("Answer is",ans)
```

R code Exa 3.25 Camera flashes

```
#Camera flashes(Pg no. 91)

p = 0.2
r = 3
x = 5

#P(X<=5)
B = pnbinom((x-r),r,p)

cat(B,"is the probability that the third failure is
    obtained in five or fewer tests")
#The answer may slightly vary due to rounding off
    values
```

R code Exa 3.26 Sampling without replacement

```
#Sampling without replacement(Pg no. 93)

# P(X=0)
A = (choose(50,0)*choose(800,2))/choose(850,2)
A


# P(X=1)
B = (choose(50,1)*choose(800,1))/choose(850,2)
B

# P(X=2)
C = (choose(50,2)*choose(800,0))/choose(850,2)
C

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

R code Exa 3.27 Parts from suppliers

```
#Parts from suppliers(Pg no. 94)

supplier1 = 100 #parts from local supplier
supplier2 = 200 #parts from distant supplier

#P(X=4)
A = (choose(supplier1,4)*choose(supplier2,0))/choose
    (supplier1+supplier2,4)

#P(X>=2)
B = 0.0
for (i in 2:4)
{
  B = B + (choose(supplier1,i)*choose(supplier2,4-i)
      )/choose(supplier1+supplier2,4)
}

#P(X>=1) == 1 - P(X=0)
temp = ((choose(supplier1,0)*choose(supplier2,4))/
    choose(supplier1+supplier2,4))
C = 1 - temp

cat(A,"is the probability that all 4 parts are from
    the local supplier")
cat(B,"is the probability that two or more parts in
    the sample are from the local supplier")
cat(C,"is the probability that at least one part in
    the sample is from the local supplier")

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

R code Exa 3.28 Mean and variance

```
#Mean and variance(Pg no. 95)

size = 4
p = 100/300

u = size*p #expectation

v = size*(1/3)*(2/3)*((300-4)/299) #variance

cat("E(X) =",u)
cat("Var(X) =",v)

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

R code Exa 3.29 Customer Sample

```
#Customer Sample(Pg no. 96)

p = 0.7 #P(A)
n = 50 #no.of sample suctomers

#P(X>45)
B = 1 - pbinom(45,50,p)

cat(B,"is the probability that X>45 have purchased
    from the corporation in the last three months")
#The answer may slightly vary due to rounding off
    values
```

R code Exa 3.31 Calculations of wire flaws

```
#Calculations of wire flaws(Pg no. 99)

mean = 2.3 #flaws per mm(millimeter)
t = 1 #mm
#P(X=2)
A = dpois(2,lambda = mean*t)

t = 5 #mm
#P(X=10)
B = dpois(10, lambda = mean*t)

t = 2 #mm
#P(X>-1) == 1 - P(X=0)
C = 1 - dpois(0,lambda = mean*t)

cat("the probability of two flaws in 1 millimeter of wire is",A)
cat("the probability of 10 flaws in 5 millimeters of wire is",B)
cat("the probability of at least one flaw in 2 millimeters of wire is",C)
#The answer may slightly vary due to rounding off values
```

R code Exa 3.32 Magnetic storage and contamination

```
#Magnetic storage and contamination(Pg no. 101)

mean = 0.1 #particles per cm(centiimeter)^2
t = 100 #cm^2
```

```
#P(X=12)
A = dpois(12, lambda = mean*t)

#P(X=0)
B = dpois(0, lambda = mean*t)

#P(X<=12)
C = ppois(12, lambda = mean*t)

cat("P(X=12) =",A)
cat("P(X=0) =",B)
cat("P(X<=12) =",C)
#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

```
#Electric Current(Pg no. 110)

#P(X<5)
integrand <- function(x) {5}
q = integrate(Vectorize(integrand), lower = 4.9,
    upper = 5)
cat("P(X<5) =",q$value)

#P(4.95<X<5.1)
p = integrate(Vectorize(integrand), lower = 4.95,
    upper = 5.1)
cat("P(4.95<X<5.1) =",p$value)
```

R code Exa 4.2 Hole diameter

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

```
# P {X>12.60}
integrand <- function(x) {20*(exp(-20*(x-12.5)))}
P = integrate(integrand,lower = 12.6,upper = Inf)
P

# P {12.50<X<12.60}
integrand <- function(x) {20*(exp(-20*(x-12.5)))}
Q = integrate(integrand,lower = 12.5,upper = 12.6)
Q

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

R code Exa 4.5 Reaction time

```
#install.packages("Deriv")
library(Deriv)

#Reaction time(Pg no. 113)

#P(X<200)
f = function(x){1-(exp(-(0.01*x)))}
f(200)
#The answer may slightly vary due to rounding off
   values
```

R code Exa 4.6 Electric current

```
#Electric current(Pg no. 115)

# Expectation
integrand <- function(x) {5*x}
```

```
P = integrate(integrand,lower = 4.9,upper = 5.1)
P

# Variance
integrand <- function(x) {5*(x-10)^2}
Q = integrate(integrand,lower = 4.9,upper = 5.1)
Q

#The answer provided in textbook is wrong
```

R code Exa 4.7 Continuation of Chapter 4 Example 1

```
#Continuation of EX4_1

#(Pg no. 115)

integrand <- function(x) {0.0001*(x^2)}
P = integrate(integrand,lower = 4.9,upper = 5.1)
P #watts
```

R code Exa 4.8 Hole diameter

```
#Hole diameter(Pg no. 115)

# Expectation
integrand <- function(x) {x*20*(exp(-20*(x-12.5)))}
u = integrate(integrand,lower = 12.5,upper = Inf)
u

# Variance
integrand <- function(x) {((x - 12.55)^2)*20*(exp
    (-20*(x-12.5)))}
v = integrate(integrand,lower = 12.5,upper = Inf)
```

```
v

sd = sqrt(v$value) #Standard deviation
sd
#The answer may slightly vary due to rounding off
   values
```

R code Exa 4.9 Uniform current

```
#Uniform current(Pg no. 117)

# P {4.95<X<5}
integrand <- function(x) {5}
P = integrate(Vectorize(integrand),lower = 4.95,
   upper = 5.0)
P

a=4.9
b=5.1

u = (a+b)/2 # expectation in mA
u
v = ((0.2)^2)/12 #variance in mA^2
v
sd = sqrt(v) #standard deviation in mA
sd

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

R code Exa 4.12 standard normal distribution

```
options(scipen = 999) #for disabling scientific notation

#(Pg no. 121)

#P(X > 1.26)
p1 = 1 - pnorm(1.26,0,1)
cat("P(X>1.26) =",p1)

#P(X < 20.86)
p1 = pnorm(20.86,0,1)
cat("P(X<20.86) =",p1)

#P(X > -1.37) = P(X < 1.37)
p1 = pnorm(1.37,0,1)
cat("P(X>-1.37) =",p1)

#P(-1.25 < X < 0.37)
pmin = pnorm(0.37,0,1)
pmax = pnorm(-1.25,0,1)
cat("P(-1.25 < Z < 0.37) =",pmin-pmax)

#P(X <= -4.6)
p = pnorm(-3.99,0,1)
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")
#the answer given in textbook is wrong for P(X < 20.86)
```

R code Exa 4.13 Normally distributed current

```
#Normally distributed current(Pg no. 123)

mean = 10 #mA
```

```
v = 4 #mA^2
x = 13 #mA

p = 1 - pnorm(x, mean ,sqrt(v))
p

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

R code Exa 4.14 Normally distributed current

```
#Normally distributed current(Pg no. 124)

mean = 10 #mA
v = 4 #mA^2

#P(9<X<11)
p1 = 1 - pnorm(9, mean ,sqrt(v))
p1

p2 = 1 - pnorm(11, mean, sqrt(v))
p2

result = p1- p2
result

z =2.06
x = z*sqrt(v) + mean
x #mA
#The answer may slightly vary due to rounding off
    values
```

R code Exa 4.15 Signal detection

```
#Signal detection(Pg no. 124)

result = 1 - pnorm(0.9, 0, 0.45)
result

cat(result,"is the probability of detecting a
   digital 1 when none was sent")
```

R code Exa 4.16 Shaft diameter

```
#Shaft diameter(Pg no. 125)

mean = 0.2508 #inches
sd = 0.0005 #inches

# P(0.2485<X<0.2515)
p1 = 1 - pnorm(0.2485, mean ,sd)

p2 = 1 - pnorm(0.2515, mean, sd)

result = p1 - p2

cat(result,"proportion of shafts conforms to
   specifications")

new_mean = 0.2500

p3 = 1 - pnorm(0.2485, new_mean ,sd)

p4 = 1 - pnorm(0.2515, new_mean, sd)

result1 = p3 - p4
cat(result1,"proportion of shafts conforms to
   specifications")
```

R code Exa 4.18 The digital communication

```
#The digital communication(Pg no. 130)

np = (16*10^6)*(1*10^-5)

#P(X <= 150.5)
z = (150.5-np)/sqrt(np*(1-10^-5))
p = pnorm(z,0,1)
cat("P(X <= 150.5) =",p)
```

R code Exa 4.19 Normal Approximation to Binomial

```
#Normal Approximation to Binomial(Pg no. 130)

n = 50
p = 0.1

#P(X<=2)
x = pbinom(2,n,p)
cat("P(X<=2) =",round(x, digits = 3))

p1 = pnorm(2.5, 5, (sqrt(n*p*(1-p))))
cat("Based on the normal approximation P(X<=2) =",p1
  )

#P(9<=X)
p2 = pnorm(8.5, 5, 2.12, lower.tail = FALSE)
cat("P(9<=X) =",round(p2, digits = 2))
```

R code Exa 4.20 Normal Approximation to Poisson

```
#Normal Approximation to Poisson(Pg no. 131)

#P(X <= 950.5)
z = (950.5-1000)/sqrt(1000)
p = pnorm(z,0,1)
cat("P(X <= 950.5) =",p)
```

R code Exa 4.21 Computer Usage

```
#Computer Usage(Pg no. 134)

mean = 25  #log-on per hour
x = 0.1 #6 min = 0.1 hr

#P(X>0.1)
p1 = pexp(x,mean,lower.tail = FALSE)

#P(0.033<X<0.05)
p2 = (pexp(0.033,25,lower.tail = FALSE) + pexp
   (0.05,25,lower.tail = TRUE))-1

cat(round(p1,digits = 3),"is the probability that
   there are no log-ons in an interval of six
   minutes")
cat(round(p2, digits = 3),"is the probability that
   the time until the next log-on is between two and
    three minutes")

#P(X > x) = 0.90
x = log(0.90)/(-mean)
cat("the length of time x such that P ( X > x ) = 0
   . 90 is",round(x*60, digits = 2),"minutes")
cat("the mean time until the next log-on is", (1/
```

```
    mean)*60,"minutes")
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

```
#Lack of memory property(Pg no. 135)

u = 1.4 #expectation in min
#P(X<0.5min)
f = 1 - exp(-(0.5/1.4))
cat("probability that we detect a particle within 30
    seconds of starting the counter is",f)

#P(3< X <3.5)
f1 = (1 - exp(-(3.5/1.4))) - (1 - exp(-(3/1.4)))

#P(X >3)
f2 = 1 - (1 - exp(-(3/1.4)))

#P(X <3.5 | X >3) = P(3< X <3.5) / P(X >3)
result = f1/f2
round(result,digits = 2)
```

R code Exa 4.23 Processor failure

```
#Processor failure(Pg no. 139)

mean = 0.0001 #failure per hr
failures = 40000

u = mean * failures #expectation

```

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

R code Exa 4.24 Poisson process

```

#(Pg no. 141)

#P(X>25)
p = ppois(9,lambda = 12.5)
round(p, digits = 3)

 r= 10
 lambda = 0.5

 mean = r/lambda
 mean

 variance = r/(lambda^2)
 variance

 sd = variance^0.5
 round(sd, digits = 2)

 #P(X <= x) = 0.95
 x = qgamma(p = 0.95, shape = r, scale = 1/lambda)
 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

```
#Bearing Wear(Pg no. 144)
```

```r

mean = 5000*0.5*sqrt(pi)
mean

#P(X>6000)
p = pweibull(q = 6000,shape = 2, scale = 5000,lower.
    tail = FALSE)

cat("only ",round(p*100,digits = 1),"% of all
    bearings last at least 6000 hours")
```

R code Exa 4.26 Semiconductor Laser

```r
#Semiconductor Laser(Pg no. 147)

o = 10
w = 1.5

#P(X > 10000)
p = 1 - plnorm(10000, meanlog = o, sdlog = w)
cat("P(X > 10000) =",p)

x = qlnorm(0.99,o,w,lower.tail = FALSE)
cat("x =",round(x,digits = 2),"hours")

E_x = exp(o+(w^2/2))
cat("mean =",round(E_x,digits = 1),"hours")

V_x = (exp((2*o)+(w^2)))*(exp((w^2))-1)
cat("standard deviation =",sqrt(V_x),"hours")

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

R code Exa 4.27 Commercial development

```
#(Pg no. 149)

alpha = 2.5
beta = 1


#P(X > 0.7)
 p = pbeta(q = 0.7, shape1 = alpha, shape2 = beta,
    lower.tail = FALSE)
 cat("P(X > 0.7) =",round(p,digits = 2))
```

R code Exa 4.28 Generalized beta distribution

```
#(Pg no. 150)

a = 8
b = 20
m = 16

mean = (a+(4*m)+b)/6
cat("mean =",round(mean,digits = 3))
alpha = ((mean-a)*((2*m)-a-b))/((m-mean)*(b-a))
cat("alpha =",round(alpha,digits = 3))
beta = (alpha*(b-mean))/(mean-a)
cat("beta =",round(beta,digits = 3))
```

Chapter 5

Joint Probability Distributions

R code Exa 5.2 Server Access Time

```
#Server Access Time(Pg no. 158)

fnc = function(x) {(exp((-(0.002*x)-(0.001*x))))/(0.002)}

an1 = integrate(fnc, 0, Inf)$value
an1 = an1*6*10^(-6)
an1
```

R code Exa 5.3 Marginal Distribution

```
#Marginal Distribution(Pg no. 159)

P3_1 = 0.25
P3_2 = 0.2
P3_3 = 0.05
P3_4 = 0.05

```

```
f3 = P3_1 + P3_2 + P3_3 + P3_4
f3
```

R code Exa 5.4 Server Access Time

```
#Server Access Time(Pg no. 160)

#P(Y>2000)
f <- function(y)
{
 (exp((-(0.002*y))))*(1-exp(-(0.001*y)))
}
v = integrate(f, lower = 2000, upper = Inf)
ans = (6*10^-3)*(v$value)
cat("P(Y>2000) =",round(ans,digits = 2))
```

R code Exa 5.5 Conditional Probabilities for Mobile Response Time

```
#Conditional Probabilities for Mobile Response Time(Pg no. 162)

P3_1 = 0.25
P3_2 = 0.2
P3_3 = 0.05
P3_4 = 0.05

f3 = P3_1 + P3_2 + P3_3 + P3_4

cat("P(Y=1|X=3) =",round((P3_1/f3),digits = 3))
cat("P(Y=2|X=3) =",round((P3_2/f3),digits = 3))
cat("P(Y=3|X=3) =",round((P3_3/f3),digits = 3))
cat("P(Y=4|X=3) =",round((P3_4/f3),digits = 3))
```

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

R code Exa 5.6 Conditional Probability

```
#Conditional Probability(Pg no. 163)

#P(Y>2000|X=1500)
f <- function(y)
{
  0.002*(exp((0.002*1500)-(0.002*y)))
}
v = integrate(f, lower = 2000, upper = Inf)
ans = v$value
cat("P(Y>2000|X=1500) =",round(ans,digits = 3))
```

R code Exa 5.8 Conditional Mean And Variance

```
#Conditional Mean And Variance(Pg no. 164)

#E(Y|X=1500)
f <- function(y)
{
  y*(0.002*(exp((0.002*1500)-(0.002*y))))
}
v = integrate(f, lower = 1500, upper = Inf)
ans = v$value #ms
cat("E(Y|X=1500) =",ans)
```

R code Exa 5.9 Conditional Mean And Variance

```
#(Pg no. 164)
y = c(1,2,3,4)
prob = c(0.05,0.1,0.1,0.75)
u = weighted.mean(y,prob)
cat("E(Y|1) =",u)

v = weighted.mean((y-u)^2, prob)
cat("V(Y|1) =",round(v, digits = 3))
```

R code Exa 5.12 Machined Dimensions

```
#Machined Dimensions(Pg no. 166)

xmean = 10.5
xvar = 0.0025
xstd_dev = sqrt(xvar)
ymean = 3.2
yvar = 0.0036
ystd_dev = sqrt(yvar)

#P(10.4<X<10.6,3.15<Y<3.25)
zx_min = pnorm(10.4,xmean,xstd_dev,lower.tail =
    FALSE)
zx_max = pnorm(10.6,xmean,xstd_dev)

zy_min = pnorm(3.15,ymean,ystd_dev, lower.tail =
    FALSE)
zy_max = pnorm(3.25,ymean,ystd_dev)

p = zx_min*zx_max*zy_min*zy_max
cat("P(10.4<X<10.6,3.15<Y<3.25) =",round(p,digits =
    1))

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

R code Exa 5.14 Component Lifetimes

```
#Component Lifetimes(Pg no. 167)

f = function(x1,x2,x3,x4)
{
  (exp(-(0.001*x1)-(0.002*x2)-(0.0015*x3)-(0.003*x4)
    ))
}

#P(X1>1000,X2>1000,X3>1000,X4>1000)

p = f(1000,1000,1000,1000)
cat("P(X1>1000,X2>1000,X3>1000,X4>1000)=",round(p,
   digits = 5))
```

R code Exa 5.15 Probability as a Ratio of Volumes

```
#install.packages("MASS")
library(MASS)

#Probability as a Ratio of Volumes(Pg no. 168)

area1 = 4*pi
area2 = pi

p = area2/area1
cat("Prabability is",fractions(p))
ans = fractions(p)
ans
```

R code Exa 5.16 Marginal probability distribution

```
#(Pg no. 168)

#P(X2 = 0)
P_0 = 0.4

#P(X2 = 1)
P_1 = 0.3

#P(X2 = 2)
P_2 = 0.2

#P(X2 = 3)
P_3 = 0.1

x = c(0,1,2,3)

E = weighted.mean(x,c(P_0,P_1,P_2,P_3))
cat("E(X2) =",E)
```

R code Exa 5.18 Layer Thickness

```
#Layer Thickness(Pg no. 170)

mean1 = 10000
std_dev1 = 250
mean2 = 1000
std_dev2 = 20
mean3 = 80
std_dev3 = 4

```

```
#P(9200<X1<10800,950<X2<1050,75<X3<85)
zx1_min = pnorm(9200,mean1,std_dev1,lower.tail =
    FALSE)
zx1_max = pnorm(10800,mean1,std_dev1,lower.tail =
    FALSE)

zx2_min = pnorm(950,mean2,std_dev2,lower.tail =
    FALSE)
zx2_max = pnorm(1050,mean2,std_dev2, lower.tail =
    FALSE)

zx3_min = pnorm(75,mean3,std_dev3,lower.tail = FALSE
    )
zx3_max = pnorm(85,mean3,std_dev3, lower.tail =
    FALSE)

p = (zx1_min-zx1_max)*(zx2_min-zx2_max)*(zx3_min-zx3
    _max)
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

```
#Expected Value of a Function of Two Random
    Variables(Pg no. 174)

y1 = c(0.01,0.02,0.25)
y2 = c(0.02,0.03,0.2)
y3 = c(0.02,0.1,0.05)
y4 = c(0.15,0.1,0.05)

#joint probability matrix
M = matrix(c(y1,y2,y3,y4), nrow = 4, ncol = 3, byrow
     = TRUE)
```

```
ux = 2.35
uy = 2.49
M
sum_total= 0.0
for(i in 1:4)
{
  for(j in 1:3)
  {
    sum_total = sum_total + ((j-ux)*(i-uy)*M[i,j])
  }
}

cat("E[(X–ux)(Y–uy)] =",sum_total)
```

R code Exa 5.21 Covariance

```
#Covariance(Pg no. 176)
x = c(0,1,1,2,2,3)
y = c(0,1,2,1,2,3)
w = c(0.2,0.1,0.1,0.1,0.1,0.4)

E_xy = 0.0
for(i in 1:length(x))
{
  E_xy = E_xy + (x[i]*y[i]*w[i])
}

E_x = 0*0.2+1*0.2+2*0.2+3*0.4
E_x
V_x = ((0-E_x)^2*0.2)+((1-E_x)^2*0.2)+((2-E_x)^2*
   0.2)+((3-E_x)^2*0.4)
E_y = E_x
V_y = V_x
V_x

```

```
sigma_xy = E_xy - E_x*E_y
sigma_xy

p_xy = sigma_xy/(sqrt(V_x)*sqrt(V_y))
round(p_xy,digits = 3)
```

R code Exa 5.23 Independence Implies Zero Covariance

```
#install.packages("pracma")
#install.packages("MASS")
library(pracma)
library(MASS)

#Independence Implies Zero Covariance(Pg no. 177)

fun_xy <- function(x, y) (1/16)*(x^2)*(y^2)

E_xy = integral2(fun_xy, 0, 2, 0, 4, reltol = 1e-10)
E_xy = E_xy$Q
cat("E(XY) =",E_xy)
fractions(E_xy)

fun_x <- function(x, y) (1/16)*(x^2)*(y)

E_x = integral2(fun_x, 0, 2, 0, 4, reltol = 1e-10)
E_x = E_x$Q
cat("E(X) =",E_x)
fractions(E_x)

fun_y <- function(x, y) (1/16)*(x)*(y^2)

E_y = integral2(fun_y, 0, 2, 0, 4, reltol = 1e-10)
E_y = E_y$Q
cat("E(Y) =",E_y)
fractions(E_y)
```

```

if(ceiling(E_xy-(E_x*E_y)) == 0)
{
  print("The two random variables are independent")
}
```

R code Exa 5.24 Digital Channel

```
#Digital Channel(Pg no. 179)

e = 0.6
g = 0.3
f = 0.08
p = 0.02

prob = (e^14)*(g^3)*(f^2)*(p^1)
prob

no_seq = factorial(20)/(factorial(14)*factorial(3)*
   factorial(2)*factorial(1))
no_seq

P = no_seq*prob
cat("P(14E's, 3 G's, 2 F's, and 1 P) =",round(P,
   digits = 4))
```

R code Exa 5.25 Digital Channel

```
#Digital Channel(Pg no.180)

x = c(0.6,0.3,0.08,0.02)
P = dmultinom(c(12,6,2,0), prob = x)
```

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

R code Exa 5.31 Error Propagation

```
#Error Propagation(Pg no. 185)

var_x = 25+40+30
cat("V(X) =",var_x,"nm^2")
cat("standard deviation =",round(sqrt(var_x),digits
   = 2),"nm")
```

R code Exa 5.32 Linear Function of Independent Normal Random Variables

```
#Linear Function of Independent Normal Random
   Variables(Pg no. 186)

E_X1 = 2
E_X2 = 5
std_dev1 = 0.1
std_dev2 = 0.2
Y = 14.5
E_Y = 14

var_Y = (4*(0.1)^2)+(4*(0.2)^2)
z = (Y-E_Y)/sqrt(var_Y)

p = pnorm(z,0,1,lower.tail = FALSE)
cat("the probability that the perimeter exceeds 14.5
    cm is",round(p,digits = 2))
```

R code Exa 5.33 Beverage Volume

```
#Beverage Volume(Pg no. 187)

E_Xbar = 12.1
V_Xbar = 0.1^2/10
Xbar = 12

z = (Xbar - E_Xbar)/sqrt(V_Xbar)

p = pnorm(z,0,1,lower.tail = TRUE)
cat("the probability that the average volume of 10
    cans selected from this process is less than 12
    oz is",round(p,digits = 5))

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

Chapter 6

Descriptive Statistics

R code Exa 6.1 Sample mean

```
#Sample mean(Pg no. 200)

obs = c(12.6,12.9,13.4,12.3,13.6,13.5,12.6,13.1)
x = mean(obs)

cat("sample mean is",x,"pounds")
```

R code Exa 6.2 Sample variance

```
#Sample variance(Pg no. 202)

obs = c(12.6,12.9,13.4,12.3,13.6,13.5,12.6,13.1)

v = var(obs)
sd = sd(obs)

cat("sample variance is ",round(v,digits = 4)) #
   pounds^2
```

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

R code Exa 6.3 Sample variance shortcut

```
#Sample variance shortcut(Pg no. 203)

samples = c(12.6,12.9,13.4,12.3,13.6,13.5,12.6,13.1)
n = length(samples) #number of samples
x = sum(samples) # sum of individual samples
y = sum(samples^2) # sum of square of individual
    samples

v = (y - ((x)^2/n))/(n-1)

cat("sample variance is ",v) #pounds^2
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

```
#Resistors(Pg no. 175)

n = 25
mean = 100 #ohms
x = 95 #ohms

sd = mean/(sqrt(n)) #standard deviation
z = (x - mean)/sd

#P(x<95) = P(Z<-2.5)
p = pnorm(-2.5)
round(p, digits = 4)
```

R code Exa 7.3 Aircraft engine life

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

```
mean1 = 5000
mean2 = 5050

variance1 = 10
variance2 = 6

mean = mean2 - mean1
varianve = (variance2)^2 + (variance1)^2

x = 25

z = (x-mean)/sqrt(varianve)

#P(X2 − X1 >=25) = P(Z >= z)
p = 1 - pnorm(z)
p

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")

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

R code Exa 7.5 Thermal conductivity

```
#Thermal conductivity(Pg no. 252)

n = 10
obs = c(41.60 , 41.48 , 42.34 , 41.95 , 41.86 ,
   42.18 , 41.72 , 42.26 , 41.81 , 42.04)
sd = sd(obs) #standard deviation

se = sd/sqrt(n) #standard error
```

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

R code Exa 7.9 Gamma Distribution Moment Estimators

```
#Gamma Distribution Moment Estimators(Pg no. 258)

n = 8
x = 21.65
y = 6639.40 #sum of square of individual samples

r = x^2/(((1/n)*y)-(x)^2)
r

lamda = x/(((1/n)*y)-(x)^2)
round(lamda,digits = 4)

# 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

```
#Bayes Estimator for the Mean of a Normal
  Distribution(Pg no. 265)

n = 10
variance = 4
mean0 = 0
variance0 = 1
sample_mean = 0.75

estimated_mean = ((variance/n)*mean0 + (variance0)*
  sample_mean)/(1 + (variance/n))
```

```
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

```
#Metallic Material Transition(Pg no. 274)

n = 10
sd = 1
z = qnorm(0.025,lower.tail = FALSE)
mean = 64.46

min = mean - z*(sd/sqrt(n))
max = mean + z*(sd/sqrt(n))

cat("The resulting 95% CI is",round(min,digits = 2),
   "<= u <= ",round(max,digits = 2))
```

R code Exa 8.2 Metallic Material Transition

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

```
e = 0.5 #error in estimation
z = qnorm(0.025,lower.tail = FALSE)
sd = 1

n = ((z*sd)/e)^2
n = ceiling(n)

cat("the required sample size is",n)
```

R code Exa 8.3 One Sided Confidence Bound

```
#One-Sided Confidence Bound(Pg no. 277)

n = 10
sd = 1
z = qnorm(0.05,lower.tail = FALSE)
mean = 64.46

min = mean - z*(sd/sqrt(n))

cat("the interval is",round(min,digits = 2),"<= u")
```

R code Exa 8.4 The Exponential Distribution

```
#The Exponential Distribution(Pg no. 278)

x = c(2.84,2.37,7.52,2.76,3.83,1.32,8.43,2.25,1.63,0.27)
n = length(x)
df = 2*n #degrees of freedom
Cl = qchisq(0.025,20,lower.tail = TRUE)
```

```
Cu = qchisq(0.025,20,lower.tail = FALSE)
x_sum = sum(x)

upper_bound = Cu/(2*x_sum)
lower_bound = Cl/(2*x_sum)

cat("The 95% two-sided CI on lambda is",round(lower_
    bound,digits = 4),"<= lambda <=",round(upper_
    bound,digits = 4))

# The values may slightly vary due to rounding off
    of values
```

R code Exa 8.5 Mercury Contamination

```
#Mercury Contamination(Pg no. 279)

n = 53
sd = 0.3486
z = qnorm(0.025,lower.tail = FALSE)
mean = 0.5250

min = mean - z*(sd/sqrt(n))
max = mean + z*(sd/sqrt(n))

cat("The approximate 95% CI is",round(min,digits =
    4),"<= u <= ",round(max,digits = 4))
```

R code Exa 8.6 Alloy Adhesion

```
#Alloy Adhesion(Pg no. 285)
```

```
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
ans = t.test(obs)

min = ans$conf.int[1]
max = ans$conf.int[2]
cat("The resulting 95% CI is",round(min,digits = 2), "<= u <= ",round(max,digits = 3))
```

R code Exa 8.7 Detergent Filling

```
#Detergent Filling(Pg no. 289)

sample_variance = 0.0153 #fluidounce
n = 20 #bottles
x = qchisq(0.95,19, lower.tail = FALSE)
variance = ((n-1)*sample_variance)/x
sd = sqrt(variance)

cat("A 95% upper confidence bound on variance is", round(variance,digits = 4))
cat("A 95% upper confidence bound on standard deviation is",round(sd, digits = 2))
```

R code Exa 8.8 Crankshaft Bearings

```
#Crankshaft Bearings(Pg no. 292)

n = 85
x = 10
p = round(x/n, digits = 2)
p
```

```r
z = round(qnorm(0.025,lower.tail = FALSE),digits =
    2)
z

p_min = p - (z*sqrt((p*(1-p))/n))
p_max = p + (z*sqrt((p*(1-p))/n))

cat("A 95% two-sided confidence interval for p is",
    round(p_min,digits = 4),"<= p <=",round(p_max,
    digits = 4))

# The answer given in the textbook is wrong for the
    upper bound
```

R code Exa 8.9 Crankshaft Bearings

```r
#Crankshaft Bearings(Pg no. 293)

p = 0.12
e  = 0.05
z = qnorm(0.025,lower.tail = FALSE)

n = (z/e)^2 * p * (1-p)
n = ceiling(n)

cat("sample size is",n)

n = (z/e)^2 * 0.25
n = ceiling(n)
n
```

R code Exa 8.10 The Agresti Coull CI on a Proportion

```
#The Agresti−Coull CI on a Proportion(Pg no. 294)

p = 0.12
n = 85
z = qnorm(0.025,lower.tail = FALSE)

ucl = (p+((z^2)/(2*n))+(z*sqrt(((p*(1-p))/n)+((z^2)/
    (4*(n^2))))))/(1+((z^2)/n))
lcl = (p+((z^2)/(2*n))-(z*sqrt(((p*(1-p))/n)+((z^2)/
    (4*(n^2))))))/(1+((z^2)/n))

cat("UCL =",round(ucl,digits = 4))
cat("LCL =",round(lcl,digits = 4))
#The answer may slightly vary due to rounding off
    values
```

R code Exa 8.11 Alloy Adhesion

```
#Alloy Adhesion(Pg no. 298)

n = 22
sd = 3.55
df = n - 1
t = qt(0.025,df,lower.tail = FALSE)
mean = 13.71

min = mean - t*sd*sqrt(1+(1/n))
max = mean + t*sd*sqrt(1+(1/n))

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

```r
#install.packages("tolerance")
library(tolerance)

#Alloy Adhesion(Pg no. 299)

n = 22
sd = 3.55
mean = 13.71
k = K.factor(n, f = NULL, 0.05, 0.90, side = 2)

min = mean - k*sd
max = mean + k*sd

cat("The desired tolerance interval is [",round(min,
   digits = 2),",",round(max,digits = 2),"]")

# The answer may slightly vary due to rouding off
   values
```

Chapter 9

Tests of Hypotheses for a Single Sample

R code Exa 9.2 Propellant Burning Rate

```
#Propellant Burning Rate(Pg no. 324)

foo = function()
{
  n = 25
  mean = 50
  std_dev = 2
  Xbar = 51.3
  alpha = 0.05
  statistic = (Xbar-mean)/(std_dev/sqrt(n))
  z = qnorm(alpha/2,0,1,lower.tail = FALSE)
  p_value = 2*(1-pnorm(statistic,0,1))
  if(p_value < alpha)
  {
    print("Null hypothesis is rejected")
  }
  else
  {
    print("Unable to reject the null hypothesis")
```

```
  }
}
foo()
```

R code Exa 9.3 Propellant Burning Rate Type II Error

```
#Propellant Burning Rate Type II Error(Pg no. 326)

n = 25
std_dev = 2
alpha = 0.05
z = qnorm(alpha/2,0,1,lower.tail = FALSE)
lim1 = z - (sqrt(n)/std_dev)
lim2 = -z - (sqrt(n)/std_dev)
beta = pnorm(lim1,0,1) - pnorm(lim2,0,1)
beta

beta2 = 0.10
z1 = qnorm(alpha/2,0,1,lower.tail = FALSE)
z2 = qnorm(beta2,0,1,lower.tail = FALSE)
new_sample_size = (((z1+z2)^2)*(std_dev^2))/1^2
new_sample_size = round(new_sample_size)
cat("new n =",new_sample_size)
```

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

```
#install.packages("MASS")
library(MASS)

#Propellant Burning Rate Type II Error From OC Curve
    (Pg no. 327)

u =51
```

```
u0 = 50
std_dev = 2

d = (u-u0)/std_dev
fractions(d)
```

R code Exa 9.6 Golf Club Design

```
#Golf Club Design(Pg no. 334)

foo =  function()
{
  obs = c(0.8411,0.8580,0.8042,0.8191,0.8532,0.8730,0.8182,0.8483,0.8282
  ans = t.test(obs, mu = 0.82, alternative = "greater")
  p_value = ans$p.value

  if(0.005<p_value && p_value<0.01)
  {
    print("Null hypothesis rejected  and hence the mean coefficient of restitution exceeds 0.82")
  }
  else
  {
    print("Unable to reject the null hypothesis")
  }
}
foo()
```

R code Exa 9.7 Golf Club Design Sample Size

```
#install.packages("pwr")
library(pwr)

#Golf Club Design Sample Size(Pg no. 337)
s = 0.02456
diff = 0.02
sample_size = 15
d = diff/s
d
alpha = 0.05
ans = pwr.t.test(n = sample_size, d = d, sig.level =
    alpha, type = "one.sample", alternative = "
   greater")$power
cat("Since the probability of rejecting null
   hypothesis is appox.",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

```
#Automated Filling(Pg no. 342)
foo = function()
{
n = 20
sample_var = 0.0153
var = 0.01
alpha = 0.05

statistic = ((n-1)*sample_var)/var

compare = qchisq((1-alpha),(n-1))

if(statistic>compare)
{
  print("Reject null hypothesis")
```

```
}
else
{
  print("there is no strong evidence of a problem
     with incorrectly filled bottles")
}
}
foo()
```

R code Exa 9.9 Automated Filling Sample Size

```
#Automated Filling Sample Size(Pg no. 343)

sd = 0.125
sd0 = 0.10
n1 = 20
alpha = 0.05

lambda = sd/sd0

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

```
#Automobile Engine Controller(Pg no. 346)

foo= function()
{
  x = 4
  n = 200
  p0 = 0.05
```

```
  statistic = (x - (n*p0))/sqrt(n*p0*(1-p0))
  statistic

  p_value = pnorm(statistic,0,1)
  p_value

  if(p_value < p0)
  {
    print("Null hypothesis is rejected")
  }
  else
  {
    print("Unable to reject the null hypothesis")
  }
}
foo()
```

R code Exa 9.11 Automobile Engine Controller Type II Error

```
#Automobile Engine Controller Type II Error(Pg no.
   347)

p0 = 0.05
n = 200
p = 0.03
z_alpha = qnorm(p0,0,1,lower.tail = FALSE)

x = (p0 - p - (z*sqrt((p0*(1-p0))/n)))/sqrt((p*(1-p)
   )/n)

beta = 1 - pnorm(x,0,1)

z_beta = qnorm(0.10,0,1,lower.tail = FALSE)
z_beta
```

```
n = (((z_alpha*sqrt(p0*(1-p0))) + (z_beta*sqrt(p*(1-
    p))))/(p-p0))^2
n = floor(n)
n
```

R code Exa 9.12 Printed Circuit Board Defects Poisson Distribution

```
#Printed Circuit Board Defects-Poisson Distribution(
    Pg no. 351)

foo = function()
{
  n = 60
  defects = c(0,1,2,3)
  obsv_freq = c(32,15,9,4)
  observed_mean = weighted.mean(defects,obsv_freq)

  p1 = dpois(0,lambda = observed_mean)
  p2 = dpois(1,lambda = observed_mean)
  p3 = dpois(2,lambda = observed_mean)
  p4 = 1 - (p1+p2+p3)

  obsv_freq = c(obsv_freq[1],obsv_freq[2],(obsv_freq
     [3]+obsv_freq[4]))

  expected_freq = c(n*p1,n*p2,(n*p3+n*p4))

  statistic = 0.0

  for(i in 1:3)
  {
    statistic = statistic + (((obsv_freq[i] -
       expected_freq[i])^2)/expected_freq[i])
  }
```

```
  statistic

  x1 = qchisq(0.9,1)
  x2 = qchisq(0.95,1)

  if(x1<statistic && statistic<x2)
  {
    print("Unable to reject the null hypothesis")
  }
  else
  {
    print("Null hypothesis is rejected")
  }
}
foo()
```

R code Exa 9.13 Power Supply Distribution Continuous Distribution

```
#Power Supply Distribution−Continuous Distribution(
    Pg no. 352)

foo = function()
{
  alpha = 0.05
  obsv_freq = c(12,14,12,13,12,11,12,14)
  ans = chisq.test(obsv_freq)
  statistic = ans$statistic

  x = qchisq((1-alpha),5)

  if(statistic<x)
  {
    print("Unable to reject the null hypothesis")
  }
  else
```

```
  {
    print("Null hypothesis is rejected")
  }
}
foo()
```

R code Exa 9.14 Health Insurance Plan Preference

```
#Health Insurance Plan Preference(Pg no. 355)

foo = function()
{
  alpha = 0.05
  n = 500
  u1 = (340/n)
  u2 = (160/n)
  v1 = (200/n)
  v2 = (200/n)
  v3 = (100/n)

  O11 = 160
  O12 = 140
  O13 = 40
  O21 = 40
  O22 = 60
  O23 = 60

  observed = c(O11,O12,O13,O21,O22,O23)
  observed

  E11 = n*u1*v1
  E12 = n*u1*v2
  E13 = n*u1*v3
  E21 = n*u2*v1
  E22 = n*u2*v2
```

```
  E23 = n*u2*v3

  expected = c(E11,E12,E13,E21,E22,E23)
  expected

  r=2
  c=3

  df = (r-1)*(c-1) #degrees of freedom

  statistic = 0.0
  for(i in 1:6)
  {
    statistic = statistic + ((observed[i]-expected[i])^2)/expected[i]
  }
  statistic

  x = qchisq(0.95,df)
  x

  if(statistic<x)
  {
    print("Unable to reject the null hypothesis")
  }
  else
  {
    print("Null hypothesis is rejected")
  }
}
foo()
```

R code Exa 9.15 Propellant Shear Strength Sign Test

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

```
library(BSDA)
#Propellant Shear Strength Sign Test(Pg no. 359)

foo = function()
{
  x = c
     (2158.7,1678.15,2316.00,2061.30,2207.50,1708.30,1784.70,2575.10

  ans = SIGN.test(x-2000)
  p_value = ans$p.value
  print(p_value)
  alpha = 0.05

  if(p_value<alpha)
  {
    print("Null hypothesis is rejected")
  }
  else
  {
    print("Unable to reject the null hypothesis")
  }
}
foo()
```

R code Exa 9.16 Propellant Shear Strength Wilcoxon Signed Rank Test

```
#Propellant Shear Strength—Wilcoxon Signed—Rank Test
   (Pg no. 363)
foo = function()
{
  obs = c
     (2158.7,1678.15,2316.00,2061.30,2207.50,1708.30,1784.70,2575.10

  alpha = 0.05
  ans = wilcox.test(obs,mu = 2000)
```

```
  p_value = ans$p.value

  if(p_value<alpha)
  {
    print("Null hypothesis rejected")
  }
  else
  {
    print("Unable to reject the null hypothesis")
  }
}
foo()
```

R code Exa 9.17 Two one sided tests

```
options(scipen=999)#For disabling scientific
   notation

#(Pg no. 366)

foo = function()
{
  n = 50
  sample_mean = 79.98
  sample_sd = 0.10
  x = 80 #ohms
  u1 = 80.05
  u2 = 79.95
  error = 0.01 #ohms

  t_statistic1 = (sample_mean-80.05)/(sample_sd/sqrt
     (n))
  p1 = pnorm(t_statistic1,0,1)

  t_statistic2 = (sample_mean-79.95)/(sample_sd/sqrt
```

```
(n))
  p2 = pnorm(t_statistic2,0,1,lower.tail = FALSE)

  if(p1<error)
  {
    print("the mean resistance is less than 80.05")
  }
  else
  {
    print("the mean resistance is not less than 80.05")
  }

  if(p2<0.025)
  {
    print("the mean resistance is significantly greater than 79.95")
  }
  else
  {
    print("the mean resistance is not greater than 79.95")
  }
}
foo()
```

Chapter 10

Statistical Inference for Two Samples

R code Exa 10.1 Paint Drying Time

```
#Paint Drying Time(Pg no. 377)

foo = function()
{
  X1bar = 121
  X2bar = 112
  std_dev1 = std_dev2 = 8
  n1 = n2 =10
  delta0 = 0

  statistic = (X1bar-X2bar-delta0)/sqrt((std_dev1^2/
     n1)+(std_dev2^2/n2))
  statistic

  p = 1 - pnorm(statistic,0,1,lower.tail = TRUE)
  p

  if(p<0.05)
  {
```

```
    print("Null hypothesis is rejected ")
  }
  else
  {
    print("Null hypothesis is accepted")
  }
}
foo()
```

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

```
#Paint Drying Time, Sample Size from OC Curves(Pg no
    . 378)

delta = 10
std_dev1 = std_dev2 = 8

d = (delta)/sqrt(std_dev1^2 + std_dev1^2)

round(d, digits = 2)
```

R code Exa 10.3 Paint Drying Time Sample Size

```
#Paint Drying Time Sample Size(Pg no. 379)

alpha = 0.05
beta = 0.10
delta = 10
delta0 = 0
std_dev1 = std_dev1 = 8

z_alpha = qnorm(alpha,0,1,lower.tail = FALSE)
z_alpha
```

```
z_beta = qnorm(beta,0,1,lower.tail = FALSE)
z_beta

n = ((z_alpha+z_beta)^2 * (std_dev1^2 + std_dev2^2))/(delta-delta0)^2
n = round(n)
n
```

R code Exa 10.4 Aluminum Tensile Strength

```
#Aluminum Tensile Strength(Pg no. 380)

X1bar = 87.6
X2bar = 74.5
std_dev1 = 1.0
std_dev2 = 1.5
n1 = 10
n2 =12
alpha = 0.1

z_alpha = qnorm(alpha/2,0,1,lower.tail = FALSE)

lim1 = X1bar - X2bar - (z_alpha * sqrt(((std_dev1^2)/n1)+((std_dev2^2)/n2)))
lim2 = X1bar - X2bar + (z_alpha * sqrt(((std_dev1^2)/n1)+((std_dev2^2)/n2)))

cat("90% CI is",round(lim1, digits = 2),"<= u - u0 <=",round(lim2, digits = 2))
```

R code Exa 10.5 Yield from a Catalyst

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

```
foo  = function()
{
  catalyst1 = c
     (91.50,94.18,92.18,95.39,91.79,89.07,94.72,89.21)

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

  ans = t.test(catalyst1,catalyst2)
  statistic = ans$statistic
  cat("t-statistic =",round(statistic, digits = 2),"
     \n")

  l1 = qt(0.40, df = 14, lower.tail = FALSE)
  l2 = qt(0.25, df = 14, lower.tail = FALSE)

  if(l1<abs(statistic) && abs(statistic)<l2)
  {
    print("Unable to reject the null hypothesis")

  }
  else
  {
    print("Null hypothesis rejected")
  }
}
foo()
```

R code Exa 10.6 Arsenic in Drinking Water

```
#Arsenic in Drinking Water(Pg no.387)

foo = function()
{
```

```
  X1bar = 12.5
  X2bar = 27.5
  std_dev1 = 7.63
  std_dev2 = 15.3
  n1 = 10
  n2 =10

  v = ((std_dev1^2/n1)+(std_dev2^2/n2))^2 / (((std_
     dev1^2/n1)^2/(n1-1))+((std_dev2^2/n2)^2/(n2-1))
     )
  v = round(v)
  v

  statistic = (X1bar - X2bar)/sqrt((std_dev1^2/n1)+(
     std_dev2^2/n2))
  cat("t-statistic = ",round(statistic, digits = 2),
     "\n")
  l1 = qt(0.025, df = 13)

  if(statistic<l1)
  {
    print("Null hypothesis rejected")
  }
  else
  {
    print("Unable to reject the null hypothesis")
  }
}
foo()
```

R code Exa 10.8 Yield from Catalyst Sample Size

```
#Yield from Catalyst Sample Size(Pg no. 390)

sp = 2.70
```

```
delta = 4.0
std_dev = (std_dev1+std_dev2)/2

d = delta/(2*std_dev)
d

n1 = 20

n = (n1+1)/2
n = ceiling(n)
n
```

R code Exa 10.9 Cement Hydration

```
#Cement Hydration(Pg no. 391)

X1bar = 90.0
X2bar = 87.0

s1 = 5.0
s2 = 4.0

n1 = 10
n2 = 15

sp = sqrt((((n1-1)*s1^2)+((n2-1)*s2^2))/(n1+n2-2))
sp

t = qt(0.025,23,lower.tail = FALSE)
t
lim1 = X1bar-X2bar-(t*sp*sqrt((1/n1)+(1/n2)))
lim2 = X1bar-X2bar+(t*sp*sqrt((1/n1)+(1/n2)))

cat("The 95% confidence interval is found as",round(lim1, digits = 2),"<= u-u0 <=",round(lim2,
```

```
    digits = 2))

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

R code Exa 10.10 Axial Stress

```
#Axial Stress(Pg no. 397)

n1 = n2 = 10
rank1 = c(2,3,4,8,9,11,13,15,16,18)

w1 = sum(rank1)

w2 = (((n1+n2)*(n1+n2+1))/2) - w1
w2
```

R code Exa 10.11 Shear Strength of Steel Girder

```
#Shear Strength of Steel Girder(Pg no. 401)

foo = function()
{
  km = c
     (1.186,1.151,1.322,1.339,1.200,1.402,1.365,1.537,1.559)

  lm = c
     (1.061,0.992,1.063,1.062,1.065,1.178,1.037,1.086,1.052)

  ans = t.test(km,lm, alternative = "two.sided",
     paired = TRUE)
  statistic = ans$statistic
  print(statistic)
```

```
  alpha = 0.05
  p_value = ans$p.value

  if(p_value<alpha)
  {
    print("the strength prediction methods yield
       different results")
  }
  else
  {
    print("the strength prediction methods yield
       similar results")
  }
}
foo()
```

R code Exa 10.12 Parallel Park Cars

```
#Parallel Park Cars(Pg no. 403)

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

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

ans = t.test(x1,x2,conf.level = 0.9, paired = TRUE)
ans
l1 = ans$conf.int[1]
l2 = ans$conf.int[2]

cat(l1,"<= uD <=",l2)

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

R code Exa 10.13 Semiconductor Etch Variability

```
#Semiconductor Etch Variability(Pg no. 410)

s1 = 1.96
s2 = 2.13
alpha = 0.05
n1 = n2 =16

statistic = s1^2/s2^2
statistic

f_max = qf(0.025,n1-1,n2-1,lower.tail = FALSE)
f_min = qf(0.975,15,15,lower.tail = FALSE)

if(f_min < statistic && statistic < f_max)
{
  cat("we cannot reject the null hypothesis at 0.05
     level of significance")
}
```

R code Exa 10.15 Surface Finish for Titanium Alloy

```
#Surface Finish for Titanium Alloy(Pg no. 412)

s1 = 5.1
s2 = 4.7
n1 = 11
n2 =16

statistic = s1^2/s2^2
```

```
statistic

f_max = qf(0.05,n2-1,n1-1,lower.tail = FALSE)
f_max
f_min = qf(0.95,n2-1,n1-1,lower.tail = FALSE)
f_min


cat("the interval is (",round(sqrt(statistic*f_min),
    digits = 3),",",round(sqrt(statistic*f_max),
   digits = 3),")")

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

R code Exa 10.16 St Johns Wort

```
#St. John's Wort(Pg no. 415)

x1 = 19
x2 = 27
n1 = n2 = 100
p1 = x2/n2
p2 = x1/n1

p = (x1+x2)/(n1+n2)
cat("p =",p)

statistic = (p1-p2)/sqrt(p*(1-p)*((1/n1)+(1/n2)))
statistic

p = 2*(1-pnorm(statistic,0,1,lower.tail = TRUE))
cat("P-value =",p)

if(p>=0.05)
```

```
{
  cat("we cannot reject the null hypothesis")
}
```

R code Exa 10.17 Defective Bearings

```
#Defective Bearings(Pg no. 419)

n1 = n2 = 85
p1 = 10/n1
p2 = 8/n2

z = qnorm(0.025,0,1,lower.tail = FALSE)

lim1 = p1 -p2 - z*sqrt((p1*(1-p1))/n1 + (p2*(1-p2))/n2)
lim2= p1 -p2 + z*sqrt((p1*(1-p1))/n1 + (p2*(1-p2))/n2)

cat("the required interval is (",round(lim1, digits = 4),",",round(lim2, digits = 4),")")
# The answer may slightly vary duet to rounding off values
```

Chapter 11

Simple Linear Regression and Correlation

R code Exa 11.1 Oxygen purity

```
#Oxygen purity(Pg no. 433)

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
y = c
   (90.01,89.05,91.43,93.74,96.73,94.45,87.59,91.77,99.42,93.65,93.5
df = data.frame('y' = y, 'x' = x)
model = lm(y~x, data = df)

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

```
#Oxygen Purity Tests of Coefficients(Pg no. 442)

foo = function()
{
  alpha = 0.01
  beta1 = 14.947
  S_xx = 0.68088
  sample_var = 1.18

  statistic = beta1/sqrt(sample_var/S_xx)
  round(statistic, digits = 2)

  t = qt(0.005,18,lower.tail = FALSE)
  round(t,digits = 2)

  if(t < statistic)
  {
    print("Null hypothesis is rejected as the test
       statistic is very far into the critical
       region")
  }
  else
  {
    print("Unable to reject the null hypothesis")
  }
}
foo()
```

R code Exa 11.3 Oxygen Purity ANOVA

```
#Oxygen Purity ANOVA(Pg no. 444)

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
```

```
y = c
   (90.01,89.05,91.43,93.74,96.73,94.45,87.59,91.77,99.42,93.65,93.5

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

R code Exa 11.4 Oxygen Purity Confidence Interval on the Slope

```
#Oxygen Purity Confidence Interval on the Slope(Pg
   no.447)

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

y = c
   (90.01,89.05,91.43,93.74,96.73,94.45,87.59,91.77,99.42,93.65,93.5

df = data.frame('y' = y, 'x' = x)
model = lm(y~x, data = df)
ans = confint(model)

lim1 = ans[2]
lim2 = ans[4]

cat(round(lim1, digits = 3),"<= Beta <=",round(lim2,
    digits = 3))
```

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

```
#Oxygen Purity Confidence Interval on the Mean Response(Pg no. 448)

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
y = c(90.01,89.05,91.43,93.74,96.73,94.45,87.59,91.77,99.42,93.65,93.5
df = data.frame('y' = y, 'x' = x)
model = lm(y~x, data = df)

ans = predict(model,data.frame(x = 1),interval = "confidence")
cat("prediction of mean oxygen purity when x0 = 1.00% is",ans[1])

min = ans[2]
max = ans[3]

cat(round(min, digits = 2),"<= u <=",round(max, digits = 2))

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

R code Exa 11.6 Oxygen Purity Prediction Interval

```
#Oxygen Purity Prediction Interval(Pg no. 450)

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
y = c(90.01,89.05,91.43,93.74,96.73,94.45,87.59,91.77,99.42,93.65,93.5
```

```
df = data.frame('y' = y, 'x' = x)
model = lm(y~x, data = df)

ans = predict(model,data.frame(x = 1),interval = "
   prediction")
cat("prediction of mean oxygen purity when x0 =
   1.00% is",ans[1])

min = ans[2]
max = ans[3]

cat(round(min, digits = 2),"<= y0 <=",round(max,
   digits = 2))
```

R code Exa 11.8 Wire Bond Pull Strength

```
#Wire Bond Pull Strength(Pg no. 460)

S_xx = 698.56
S_xy = 2027.7132
SSt = 6105.9
n = 25

r = S_xy/sqrt(S_xx*SSt)

statistic = (r*sqrt(n-2))/sqrt(1-(r^2))
statistic

compare = qt(0.025,n-2,lower.tail = FALSE)
compare

z = qnorm(0.025,0,1,lower.tail = FALSE)
z

```

```
lim1 = tanh(atanh(r) - (z/sqrt(n-3)))
lim2 = tanh(atanh(r) + (z/sqrt(n-3)))

cat(round(lim1, digits = 4),"<= P <= ",round(lim2,
   digits = 4))
```

Chapter 12

Multiple Linear Regression

R code Exa 12.1 Wire Bond Strength

```
#Wire Bond Strength(Pg no. 482)

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)
x2 = c
   (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,60
y = c
   (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,
df = data.frame('x1' = x1, 'x2'= x2, 'y' = y)
model = lm(y~x1+x2, data = df)

cat("the fitted regression equation is y =",round(
   model$coefficients[1], digits = 5),"+",round(
   model$coefficients[2], digits = 5),"x1 + ",round(
   model$coefficients[3], digits = 5),"x2")
```

R code Exa 12.2 Wire Bond Strength With Matrix Notation

```
#install.packages("matlib")
library(matlib)

#Wire Bond Strength With Matrix Notation(Pg no. 485)


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)

x2 = c
   (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,60

x3 = c(rep.int(1,25))
y = c
   (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,


X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow =
   FALSE)
Y = matrix(c(y), nrow = 25, ncol = 1, byrow = FALSE)

X_t = t(X)

XX_t = X_t %*% X

XX_t

Xt_y = X_t %*% Y

Xt_y

beta = inv(XX_t) %*% Xt_y

cat("the fitted regression model is y =",beta[1,1],"
   +",beta[2,1],"x1 +",beta[3,1],"x2")
y0 = beta[1,1] + beta[2,1]*2 + beta[3,1]*50
```

```
y0
y1 = 9.95

e = y1 - y0
e = round(e, digits = 2)
e
# The answer may slightly vary due to rounding off
    values
```

R code Exa 12.3 Wire Bond Strength ANOVA

```
#Wire Bond Strength ANOVA(Pg no. 498)

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)

die_ht = c
   (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,60

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,


df = data.frame('pull_strength' = pull_strgth, 'wire
   _length' = wire_lt, 'die_height' = die_ht)
model = lm(pull_strength~wire_length+die_height,
   data = df)
model
ans = aov(model)
summary(ans)
```

R code Exa 12.4 Wire Bond Strength Coefficient Test

```
#install.packages("matlib")
library(matlib)
#Wire Bond Strength Coefficient Test(Pg no. 500)

foo= function()
{
  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)

  x2 = c
     (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,

  x3 = c(rep.int(1,25))
  y = c
     (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08


  X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow
     = FALSE)
  Y = matrix(c(y), nrow = 25, ncol = 1, byrow =
     FALSE)
  X_t = t(X)
  XX_t = X_t %*% X
  Xt_y = X_t %*% Y
  beta = inv(XX_t) %*% Xt_y

  sample_var = 5.2352
  inv_XtX = inv(X_t%*%X)
  inv_XtX

  C22 = inv_XtX[3,3]

  statistic = beta[3,1]/sqrt(sample_var*C22)
  statistic

  compare = qt(0.025,22,lower.tail = FALSE)
  compare

```

```
  if(statistic>compare)
  {
    print("We reject the null hypothesis and
        conclude that the variable x2 (die height)
        contributes significantly to the model")
  }
  else
  {
    print("Unable to reject the null hypothesis")
  }
}
foo()
```

R code Exa 12.5 Wire Bond Strength One Sided Coefficient Test

```
#Wire Bond Strength One-Sided Coefficient Test(Pg no
    . 501)

foo = function()
{
  beta2 = 0.012528
  std_err = 0.002793

  statistic = (beta2 - 0.01)/std_err
  statistic

  t0 = qt(0.25,22,lower.tail = FALSE)
  t0
  t1 = qt(0.10,22,lower.tail = FALSE)
  t1

  p_max = 1 - pnorm(t0,0,1)
  p_min = 1 -pnorm(t1,0,1)

  p = 1 - pnorm(statistic,0,1)
```

```
  p

  if(p_min<p && p<p_max)
  {
    print("One cannot conclude that the coefficient
       exceeds 0.01 at common levels of significance
       ")
  }
  else
  {
    print("One can conclude that the coefficient
       exceeds 0.01 at common levels of significance
       ")
  }
}
foo()
```

R code Exa 12.6 Wire Bond Strength General Regression Test

```
#Wire Bond Strength General Regression Test(Pg no.
   502)

SSr1 = 6024.0 #4 degrees of freedom
SSr2 = SSr1 - 5990.8 #2 degrees of freedom
MSe = 4.1

statistic = (SSr2/2)/MSe
statistic = round(statistic, digits = 2)
statistic
```

R code Exa 12.7 Wire Bond Strength Confi dence Interval

```
#install.packages("matlib")
```

```r
library(matlib)

#Wire Bond Strength Confi dence Interval(Pg no. 507)

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)
x2 = c(50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,60
x3 = c(rep.int(1,25))
y = c(9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,

X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow = FALSE)
Y = matrix(c(y), nrow = 25, ncol = 1, byrow = FALSE)

X_t = t(X)
XX_t = X_t %*% X
Xt_y = X_t %*% Y
XtX = X_t%*%X
inv_XtX = inv(XtX)
beta = inv(XX_t) %*% Xt_y

C11 = inv_XtX[2,2]
sample_var = 5.2352
t = qt(0.025,22,lower.tail = FALSE)

min = beta[2,1] - (t*sqrt(sample_var*C11))
max = beta[2,1] + (t*sqrt(sample_var*C11))

cat("the interval is(",min,",",max,")")

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

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

```
#install.packages("matlib")
library(matlib)

#Wire Bond Strength Confidence Interval on the Mean
    Response(Pgno. 508)

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)

x2 = c
    (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,60

x3 = c(rep.int(1,25))
y = c
    (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,


X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow =
    FALSE)
Y = matrix(c(y), nrow = 25, ncol = 1, byrow = FALSE)

X_t = t(X)
XX_t = X_t %*% X
Xt_y = X_t %*% Y
beta = inv(XX_t) %*% Xt_y

sample_var = 5.2352
x0 = matrix(c(1,8,275), nrow = 3, ncol = 1, byrow =
    FALSE)
u = t(x0)%*%beta

```

```r
u_var = sample_var * ((t(x0))%*%inv(X_t%*%X)%*%x0)
u_var

t = qt(0.025,22,lower.tail = FALSE)

min = u - (t*sqrt(u_var))
max = u + (t*sqrt(u_var))

cat("the interval is(",min,",",max,")")

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

R code Exa 12.9 Wire Bond Strength Confidence Interval

```r
#install.packages("matlib")
library(matlib)
#Wire Bond Strength Confidence Interval

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)
x2 = c(50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,60
x3 = c(rep.int(1,25))
y = c(9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,

X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow = FALSE)
Y = matrix(c(y), nrow = 25, ncol = 1, byrow = FALSE)

X_t = t(X)
```

```
XX_t = X_t %*% X
Xt_y = X_t %*% Y
XtX = X_t%*%X
inv_XtX = inv(XtX)
beta = inv(XX_t) %*% Xt_y

sample_var = 5.2352
x0 = matrix(c(1,8,275), nrow = 3, ncol = 1, byrow = FALSE)
y0 = t(x0)%*%beta

y0_var = sample_var * (1+((t(x0))%*%inv(X_t%*%X)%*% x0))
y0_var

t = qt(0.025,22,lower.tail = FALSE)

min = y0 - (t*sqrt(y0_var))
max = y0 + (t*sqrt(y0_var))

cat("the interval is(",min,",",max,")")

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

R code Exa 12.11 Wire Bond Strength Confidence Interval

```
#install.packages("matlib")
library(matlib)
#Wire Bond Strength Confidence Interval(Pg no. 515)

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)

x2 = c
```

```
    (50,110,120,550,295,200,375,52,100,300,412,400,500,360,205,400,60

x3 = c(rep.int(1,25))
y = c
    (9.95,24.45,31.75,35.00,25.02,16.86,14.38,9.60,24.35,27.50,17.08,


X = matrix(c(x3,x1,x2),nrow = 25, ncol = 3, byrow =
    FALSE)
Y = matrix(c(y), nrow = 25, ncol = 1, byrow = FALSE)

X_t = t(X)

H = X%*%inv(X_t%*%X)%*%X_t
sample_var = 5.2352
e1 = 1.57
r1 = e1/sqrt(sample_var*(1-H[1,1]))
p = 3
MSe = 5.2352

D1 = ((r1^2)/p)*(H[1,1]/(1-H[1,1]))
D1 = round(D1, digits = 3)
D1
```

R code Exa 12.12 Airplane Sidewall Panels

```
#install.packages("matlib")
library(matlib)

#Airplane Sidewall Panels(Pg no. 518)

x1 = c(20,25,30,35,40,50,60,65,70,75,80,90)
x2 = c
    (400,625,900,1225,1600,2500,3600,4225,4900,5625,6400,8100)
```

```
x3 = c(rep.int(1,12))
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)

X = matrix(c(x3,x1,x2),nrow = 12, ncol = 3, byrow =
    FALSE)
Y = matrix(c(y), nrow = 12, ncol = 1, byrow = FALSE)

X_t = t(X)

XX_t = X_t %*% X

XX_t

Xt_y = X_t %*% Y

Xt_y

beta = inv(XX_t) %*% Xt_y
beta

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

```
#Surface Finish(Pg no. 520)

x2 = c
    (225,200,250,245,235,237,265,259,221,218,224,212,248,260,243,238,

y = c
    (45.44,42.03,50.10,48.75,47.92,47.79,52.26,50.52,45.58,44.78,33.5
```

```
tool = c(rep.int(302,10),rep.int(416,10))

df = data.frame("surface" = y, "rpm" = x2, "tool" =
    tool)
df$tool.f = factor(df$tool)
is.factor(df$tool.f)
df$tool.f[1:20]
model = lm(surface ~ rpm+tool.f, data = df)

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

```
#Tensile Strength ANOVA(Pg no. 546)

foo = function()
{
  obs1 = c(7,12,14,19)
  obs2 = c(8,17,18,25)
  obs3 = c(15,13,19,22)
  obs4 = c(11,18,17,23)
  obs5 = c(9,19,16,18)
  obs6 = c(10,15,18,20)

  N = 24
  n = 6

  M = matrix(c(obs1,obs2,obs3,obs4,obs5,obs6), nrow
     = 4, ncol = 6, byrow = FALSE)
  M
```

```
  y_total = sum(M)

  sum_y_sqr = sum(M^2)

  SSt = sum_y_sqr - y_total^2/N
  SSt

  sum_row1 = sum(M[1,])
  sum_row2 = sum(M[2,])
  sum_row3 = sum(M[3,])
  sum_row4 = sum(M[4,])

  SStreatments = ((sum_row1^2+sum_row2^2+sum_row3^2+
     sum_row4^2)/n) - y_total^2/N
  SStreatments

  SSe = SSt - SStreatments
  SSe

  f = qf(0.01, df1=3, df2=20, lower.tail = FALSE)
  f0 = (SStreatments/3)/(SSe/20)

  if(f<f0)
  {
    print("Null hypothesis is rejected and conclude
       that hardwood concentration in the pulp
       significantly affects the mean strength of
       the paper")
  }
  else
  {
    print("Unable to reject the null hypothesis")
  }
}
foo()

concn = c(5,10,15,20)
obs1 = c(7,12,14,19)
```

```
obs2 = c(8,17,18,25)
obs3 = c(15,13,19,22)
obs4 = c(11,18,17,23)
obs5 = c(9,19,16,18)
obs6 = c(10,15,18,20)

df = data.frame("concn" = concn, "obs1" = obs1, "
    obs2" = obs2, "obs3" = obs3, "obs4" = obs4, "obs5
    " = obs5, "obs6" = obs6)
df$concn = factor(df$concn)
```

R code Exa 13.2 Hardwood concentration experiment

```
#(Pg no. 549)

obs1 = c(7,12,14,19)
obs2 = c(8,17,18,25)
obs3 = c(15,13,19,22)
obs4 = c(11,18,17,23)
obs5 = c(9,19,16,18)
obs6 = c(10,15,18,20)

M = matrix(c(obs1,obs2,obs3,obs4,obs5,obs6), nrow =
    4, ncol = 6, byrow = FALSE)

sum_row1 = sum(M[1,])
sum_row2 = sum(M[2,])
sum_row3 = sum(M[3,])
sum_row4 = sum(M[4,])
a = 4
n = 6
MSe = 6.51
t = qt(0.025,20,lower.tail = FALSE)
t
y1 = sum_row1/n
```

```
y2 = sum_row2/n
y3 = sum_row3/n
y4 = sum_row4/n

LSD = t*(sqrt((2*MSe)/n))
LSD

cat("4 vs 1 =",y4-y1,">",LSD)
cat("4 vs 2 =",y4-y2,">",LSD)
cat("4 vs 3 =",y4-y3,">",LSD)
cat("3 vs 1 =",y3-y1,">",LSD)
cat("3 vs 2 =",y3-y2,">",LSD)
cat("2 vs 1 =",y2-y1,">",LSD)
```

R code Exa 13.4 Textile Manufacturing

```

#Textile Manufacturing(Pg no. 560)
obs1 = c(98,91,96,95)
obs2 = c(97,90,95,96)
obs3 = c(99,93,97,99)
obs4 = c(96,92,95,98)

data = rbind(obs1,obs2,obs3,obs4)

tag = c("obs1","obs2","obs3","obs4")

M = c(t(as.matrix(data)))
n_row = 4
n_col = 4

op <- gl(n_row,1,n_col*n_row,factor(tag))

m <- gl(n_col,n_row,n_row*n_col)

```

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

R code Exa 13.5 Fabric Strength

```
#Fabric Strength(Pg no. 565)

obs1 = c(1.3,2.2,1.8,3.9)
obs2 = c(1.6,2.4,1.7,4.4)
obs3 = c(0.5,0.4,0.6,2.0)
obs4 = c(1.2,2.0,1.5,4.1)
obs5 = c(1.1,1.8,1.3,3.4)

data = rbind(obs1,obs2,obs3,obs4,obs5)

tag = c("obs1","obs2","obs3","obs4","obs5")

M = c(t(as.matrix(data)))
n_row <- 4
n_col <- 5

op <- gl(n_row,1,n_col*n_row,factor(tag))

m <- gl(n_col,n_row,n_row*n_col)


summary(aov(M~op+m))

# 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

```
#Aircraft Primer Paint(pg no. 586)

foo = function()
{
  n = 3
  a = 3
  b = 2
  dripping = matrix(c
     (4.0,4.5,4.3,5.6,4.9,5.4,3.8,3.7,4.0),nrow = 3,
     ncol = 3,byrow = TRUE)
  spraying = matrix(c
     (5.4,4.9,5.6,5.8,6.1,6.3,5.5,5.0,5.0),nrow = 3,
     ncol = 3,byrow = TRUE)
  types = c(28.7,34.1,27.0)
  methods = c(40.2,49.6)
  interaction = c(12.8, 15.9, 15.9, 18.2, 11.5,
     15.5)

  total = sum(dripping) + sum(spraying)
```

```
Yijk = sum(dripping^2) + sum(spraying^2)
SSt = Yijk - (total^2/(a*b*n))
cat("SSt =",SSt,"\n")

Yi = sum(types^2)
SStypes = (Yi/(b*n)) - (total^2/(a*b*n))
cat("SStypes =",SStypes,"\n")

Yj = sum(methods^2)
SSmethods = (Yj/(a*n)) - (total^2/(a*b*n))
cat("SSmethods =",SSmethods,"\n")

Yij = sum(interaction^2)
SSinteraction = (Yij/n) - (total^2/(a*b*n)) - SStypes - SSmethods
cat("SSinteraction =",SSinteraction,"\n")

SSe = SSt - SStypes - SSmethods -SSinteraction
cat("SSe =",SSe,"\n")

f2 = qf(0.05, df1=2, df2=12, lower.tail = FALSE)
f1 = qf(0.05, df1=1, df2=12, lower.tail = FALSE)

f0_types = (SStypes/2)/(SSe/12)
f0_methods = (SSmethods/1)/(SSe/12)
f0_interaction = (SSinteraction/2)/(SSe/12)

if(f2<f0_types)
{
  print("the main effects of primer type affect adhesion force")
}
else
{
  print("the main effects of primer type do not affect adhesion force")
}
```

```
  if(f1<f0_methods)
  {
    print("the main effects of application method
       affect adhesion force")
  }
  else
  {
    print("the main effects of application method do
        not affect adhesion force")
  }

  if(f2<f0_interaction)
  {
    print("there is indication of interaction
       between these factors")
  }
  else
  {
    print("there is no indication of interaction
       between these factors")
  }
}
foo()
```

R code Exa 14.3 Epitaxial Process

```
#Epitaxial Process(Pg no. 596)

obs1 = c(14.037,14.165,13.972,13.907)
obs2 = c(14.821,14.757,14.843,14.878)
obs3 = c(13.880,13.860,14.032,13.914)
obs4 = c(14.888,14.921,14.415,14.932)

M = matrix(c(obs1,obs2,obs3,obs4), nrow = 4, ncol =
```

```
    4, byrow = TRUE)

sum_y_sqr = sum(M^2)

init = sum(M[1,])
a = sum(M[2,])
b = sum(M[3,])
ab = sum(M[4,])

n = 4
N = 16

A = (1/(2*n)) * (a+ab-b-init)
B = (1/(2*n)) * (b+ab-a-init)
AB = (1/(2*n)) * (ab+init-a-b)
SSa = ((a+ab-b-init)^2)/N
SSa
SSb = ((b+ab-a-init)^2)/N
SSb
SSab = ((ab+init-a-b)^2)/N
SSab
SSt = sum_y_sqr - ((init+a+b+ab)^2)/N
SSt
```

R code Exa 14.4 Surface Roughness

```
#Surface Roughness(Pg no. 604)

n=2
k=3
init=9+7
a=10+12
b=9+11
ab=12+15
c=11+10
```

```
ac=10+13
bc=10+8
abc=16+14

A = (1/(4*n)) * (a+ab+ac+abc-init-b-c-bc)
A
SSa = ((a+ab+ac+abc-init-b-c-bc)^2)/(n*(2^k))
SSa
```

R code Exa 14.5 Plasma Etch

```
#Plasma Etch(Pg no. 607)

A = (1/8) * (669 + 650 + 642 + 635 + 749 + 868 + 860
    + 729-550-604-633-601-1037-1052 - 1075 - 1063)
A

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

```
#Process Yield(Pg no. 612)

yc = c(40.3,40.5,40.7,40.2,40.6)
yf = c(40.0,41.5,39.3,40.9)

ycbar = sum(yc)/length(yc)
yfbar = sum(yf)/length(yf)
nf = length(yf)
nc = length(yc)
SSe = 0.0
```

```
for(i in 1:nc)
{
  SSe = SSe + (yc[i] - ycbar)^2
}
SSe
MSe = SSe/(nc-1)
MSe

SScurvature = (nc*nf*((yfbar-ycbar)^2))/(nc+nf)
SScurvature

statistic = ((yfbar-ycbar))/sqrt(MSe*((1/nc)+(1/nf))
   )
statistic = round(statistic, digits = 2)
statistic
```

R code Exa 14.8 Plasma Etch

```
#Plasma Etch(Pg no. 629)

init = 550
ad = 749
bd = 1052
ab = 650
cd = 1075
ac = 642
bc = 601
abcd = 729

lA = (1/4) * (-init+ad-bd+ab-cd+ac-bc+abcd)
lA
lB = (1/4) * (-init-ad+bd+ab-cd-ac+bc+abcd)
lB
lC = (1/4) * (-init-ad-bd-ab+cd+ac+bc+abcd)
lC
```

```
lD = (1/4) * (-init+ad+bd-ab+cd-ac-bc+abcd)
lD

lAB = (1/4) * (init-ad-bd+ab+cd-ac-bc+abcd)
lAB
lAC = (1/4) * (init-ad+bd-ab-cd+ac-bc+abcd)
lAC
lAD = (1/4) * (init+ad-bd-ab-cd-ac+bc+abcd)
lAD
```

Chapter 15

Statistical Quality Control

R code Exa 15.1 Vane Opening

```
#Vane Opening(Pg no. 677)

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
r = c(4,6,4,4,2,3,4,10,15,6,4,4,10,4,7,6,5,3,9,6)

Xbar = sum(xbar)/20
rbar = sum(r)/20
n = 5
A2 = 0.577
UCL = Xbar + A2*rbar
LCL = Xbar - A2*rbar
cat("the trial control limits for the Xbar chart are UCl = ",round(UCL,digits = 2),"LCl = ",round(LCL,digits = 2))

D4 = 2.115
D3 = 0

cat("the trial control limits for the rbar chart are
```

```
    UCl = ",round(D4*rbar, digits = 2),"LCl = ",(D3*
   rbar))

s = c
   (1.67332,2.60768,1.58114,1.64317,0.83666,1.14018,1.51658,4.38178,

sbar = sum(s)/20
c4 = 0.94

lim = ((3*sbar)/c4)*sqrt(1-(c4^2))

cat("the trial control limits for the sbar chart are
    UCl = ",(sbar+lim),"LCl = ",(sbar-lim))

lim4 = Xbar + (3*sbar)/(c4*sqrt(n))
lim3 = Xbar - (3*sbar)/(c4*sqrt(n))

cat("If sbar is used to determine the control limits
    for the Xbar chart then limits are(",lim3,",",
   lim4,")")

Xbar = 33.21
rbar = 5.0
UCL = Xbar + A2*rbar
LCL = Xbar - A2*rbar
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

```
#Chemical Process Concentration(Pg no. 685)

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,
```

```
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,

xbar = sum(x)/length(x)
xbar

mrbar = sum(mr)/length(mr)
mrbar

d2 = 1.128

UCL = xbar + (3*(mrbar/d2))
cat("UCL =",UCL)

LCL = xbar - (3*(mrbar/d2))
cat("LCL =",LCL)

CL = xbar
cat("CL =",CL)

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

R code Exa 15.3 Electrical Current

```
#Electrical Current(Pg no. 692)

x = 110
mean = 107.0
std_dev = 1.5

PCR = (x-90)/(6*std_dev)
cat("PCR =",PCR)
```

```
PCRk = (x-mean)/(3*std_dev)
cat("PCRk =",PCRk)

LSL = (90-mean)/std_dev
USL = (x - mean)/std_dev

#P(X<LSL)
z1 = pnorm(LSL,0,1)
z1 = round(z1)
z1

#P(X>USL)
z2 = pnorm(USL,0,1,lower.tail = FALSE)
z2 = round(z2 , digits = 3)
z2
```

R code Exa 15.4 Ceramic Substrate

```
#Ceramic Substrate(Pg no. 696)

defectives = c
   (44,48,32,50,29,31,46,52,44,48,36,52,35,41,42,30,46,38,26,30)
samples = 20
size = 100
total_size = samples*size

pbar = sum(defectives)/total_size
pbar

UCL = pbar + (3*sqrt((pbar*(1-pbar))/size))
cat("UCL =",UCL)

LCL = pbar - (3*sqrt((pbar*(1-pbar))/size))
cat("LCL =",LCL)
```

```

CL = pbar
cat("CL =",CL)

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

R code Exa 15.5 Printed Circuit Boards

```
#Printed Circuit Boards(Pg no. 697)

defects = c
    (6,4,8,10,9,12,16,2,3,10,9,15,8,10,8,2,7,1,7,13)
samples = 20
size = 5

defects_p_unit = defects/size
ubar = sum(defects_p_unit)/samples
ubar

UCL = ubar + 3*sqrt(ubar/size)
cat("UCL =",UCL)

LCL = ubar - 3*sqrt(ubar/size)
cat("LCL =",LCL,"< 0")
```

R code Exa 15.6 Chemical Process Concentration CUSUM Tabular Cusum

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

```

```
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,

Sh_0 = 0
Sl_0 = 0
u0 = 99
k = 1

Sh_1 = max(0, obs[1]-(u0+k)+Sh_0)
Sh_1
Sl_1 = max(0, (u0-k)-obs[1]+Sh_0)
Sl_1
```

R code Exa 15.7 Chemical Process Concentration EWMA

```
#Chemical Process Concentration EWMA(Pg no. 711)

lambda = 0.2
n = 1
xbar = 99.1
mrbar = 2.59
u0 = 99.1
sigma = 2.59/1.128

LCL = u0 - (3*(sigma/sqrt(n))*sqrt((lambda/(2-lambda
   ))*(1-(1-lambda)^(2))))
UCL = u0 + (3*(sigma/sqrt(n))*sqrt((lambda/(2-lambda
   ))*(1-(1-lambda)^(2))))
cat("The control limits for z1 are LCL=",LCL,"UCL ="
   ,UCL)

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