R Textbook Companion for Mathematical Statistics And Data Analysis by John A. Rice[1]

Created by
Debatosh Chakraborty
BSMS (Integrated 5 year)
Mathematics
National Institute of Technology Agartala
Cross-Checked by
R TBC Team

September 23, 2022

[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: Mathematical Statistics And Data Analysis

Author: John A. Rice

Publisher: Cengage Learning India

Edition: 3

Year: 2007

ISBN: 978-81-315-1954-7

R numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means an R code whose theory is explained in Section 2.3 of the book.

Contents

List of R Codes

Chapter 1

Probability

R code Exa 1.3.A Probability of coin toss

```
#Page 6

P_A = 0.5
P_B = 0.5
P_int = 0.25

prob = P_A + P_B -P_int

print(prob)
```

R code Exa 1.3.B AIDS infection

```
#Page 6

library(MASS)

P_A1 = fractions(1/500)
P_A2 = fractions(1/500)
```

```


prob_sum = P_A1 + P_A2

print(prob_sum)
```

R code Exa 1.4.A Atleast one head in two toss

```
#Page 7

P_one = 0.25

P_one_head = P_one *3

print(P_one_head)
```

R code Exa 1.4.B Simpsons Paradox

```
#Page 7

p_blck1 = 5/(5+6)
p_white1 = 3/(3+4)

print(round(max(p_blck1,p_white1),3))

p_blck2 = 6/(6+3)
p_white2 = 9/(9+5)

print(round(max(p_blck2,p_white2),3))

p_blck_sum = (5+6)/(5+6+6+3)
p_white_sum = (3+9)/(3+4+9+5)

```

```
print(round(max(p_blck_sum,p_white_sum),3))
```

R code Exa 1.4.1.A Combinations

```
#Page 8

Face_val = 13
suit = 4

combination = Face_val * suit

print(combination)
```

R code Exa 1.4.1.B Class representatives selection

```
#Page 8

boy = 12
girl = 18

total_ways = boy * girl

print(total_ways)
```

R code Exa 1.4.1.C 8 bit words

```
#Page 8

choice_one_bit = 2

```

```
choice_all_bit = runif(8,2,2)

total_choices = prod(choice_all_bit)

print(total_choices)
```

R code Exa 1.4.2.A Children lineup

```
#Page 10

total_lineup = factorial(5)

print(total_lineup)
```

R code Exa 1.4.2.B Different lines for lineup

```
#Page 10

total_lineup = choose(10,5) * factorial(5)

print(total_lineup)
```

R code Exa 1.4.2.C Choosing license plates

```
#Page 10

total_letters = 26

ways_letter_sample = total_letters^3

```

```
total_numbers = 10

ways_number_sample = total_numbers^3

total_plates = ways_letter_sample * ways_number_
    sample

print(total_plates)
```

R code Exa 1.4.2.D Choosing license plates without duplicates

```
#Page 10

total_letters = 26

ways_letter_sample = total_letters^3

total_numbers = 10

ways_number_sample = total_numbers^3

total_plates = ways_letter_sample * ways_number_
    sample

print(total_plates)

letter_chosing = choose(total_letters,3)*factorial
    (3)

number_chosing = choose(total_numbers,3)*factorial
    (3)

chosing_without_replace = letter_chosing * number_
    chosing
print(chosing_without_replace)
```

```
prob_no_duplicate = round(chosing_without_replace /
   total_plates,2)

print(prob_no_duplicate)
```

R code Exa 1.4.2.E Birthday Problem

```
#Page 10

prob = function(n){

  A_comp = choose(365,n)*factorial(n)
  p_A_comp = A_comp / 365**n
  p_A = 1 - p_A_comp
  return(p_A)
}

age = c(4,16,23,32,40,56)
bthdy = data.frame(age,prob(age))

print(round(bthdy,3))
```

R code Exa 1.4.2.F Sharing birthday

```
#Page 10

prob = function(n){

  n_A_comp = 364**n
  p_A_comp = n_A_comp / 365**n
  p_A = 1 - p_A_comp
  return(p_A)
}

func_people = function(n) prob(n)-0.5

num_people = uniroot(func_people, lower = 0, upper =
    365)

print(num_people)
```

R code Exa 1.4.2.G California Lottery

```
#Page 12

tot_num = 49
num_choose = 6

tot_way = choose(tot_num,num_choose)
print(tot_way)

new_rul_tot = 53

tot_way_new_rule = choose(new_rul_tot,num_choose)
print(tot_way_new_rule)
```

R code Exa 1.4.2.I Capture Recapture Method

```
#Page 13

prob = function(n){

  total_grp = choose(n,20)
  no_evnt = choose(10,4)*choose(n-10,16)
  liklihood = no_evnt/total_grp
```

```r
  return(liklihood)
}

n = seq(20,100)

plot(n,prob(n))

print(max(prob(n)))
```

R code Exa 1.4.2.J Seven member committe

```r
#Page 15

library(iterpc)

way = multichoose(c(3,2,2))

print(way)
```

R code Exa 1.4.2.K Nucleotide sequence

```r
#Page 15

library(iterpc)

way = multichoose(c(2,4,3))

print(way)
```

R code Exa 1.5.A Urn both red

```r
#Page 17

library(MASS)

p_r1 = fractions(3/4)
p_r2_r1 = fractions(2/3)

p_intersect = p_r1*p_r2_r1

print(p_intersect)
```

R code Exa 1.5.B Cloudy and raining probability

```r
#Page 18

p_a_b = 0.3
p_b = 0.2

p_intersect = p_a_b*p_b

print(p_intersect)
```

R code Exa 1.5.C Red ball selection from urn

```r
#Page 19

library(MASS)

p_r1 = fractions(3/4)
p_r2_r1 = fractions(2/3)
p_r2_b1 = 1
p_b1 = 1 - p_r1

```

```
p_r2 = p_r2_r1*p_r1 + p_r2_b1*p_b1

print(p_r2)
```

R code Exa 1.5.D Occupation problem Glass and Hall

```
#Page 19

cname = c("u2", "m2", "l2")
rname = c('u1', 'm1', 'l1')

val = c(0.45, 0.48, 0.07, 0.05, 0.7, 0.25, 0.01,
   0.5, 0.49)

prob_matrix = matrix(val, nrow = 3, byrow = TRUE,
   dimnames = list(rname, cname))

print(prob_matrix)

rval = c(0.1, 0.4, 0.5)

prob_u2 = sum(prob_matrix[,'u2']*rval)

print(prob_u2)
```

R code Exa 1.5.E Coronary artery disease Diamond and Forrester

```
#Page 20

cname = c("d+", "d-")


val = c(0.42, 0.96, 0.24, 0.02, 0.2, 0.02, 0.15, 0)
```

```r

prob_matrix = matrix(val, ncol = 2, byrow = TRUE )
colnames(prob_matrix) = cname

print(prob_matrix)

cval1 = c(0.05, 0.95)

prob_t0 = sum(prob_matrix[1,]*cval1)
prob_t1 = sum(prob_matrix[2,]*cval1)

prob_dplus_t0 = prob_matrix[1,'d+']*cval1[1]/prob_t0
prob_dplus_t1 = prob_matrix[2,'d+']*cval1[1]/prob_t1


print(c(prob_dplus_t0, prob_dplus_t1))

cval2 = c(0.92, 0.08)

prob_t0_2 = sum(prob_matrix[1,]*cval2)
prob_t1_2 = sum(prob_matrix[2,]*cval2)

prob_dplus_t0_2 = prob_matrix[1,'d+']*cval2[1]/prob_
    t0_2
prob_dplus_t1_2 = prob_matrix[2,'d+']*cval2[1]/prob_
    t1_2

print(c(prob_dplus_t0_2, prob_dplus_t1_2))
```

R code Exa 1.5.F Polygraph Test

```r
#Page 21

p_t = 0.99
p_plus_t = 0.14
```

```

p_t_plus = (p_plus_t*p_t)/(p_plus_t*p_t + (1-p_plus_
   t)*(1- p_t))

print(round(p_t_plus ,2))
```

R code Exa 1.6.A Ace and Diamond probability

```
#Page 23

library(MASS)

p_a = fractions(4/52)
p_d = fractions(1/4)

p_intersect = p_a*p_d

print(p_intersect)
```

R code Exa 1.6.D No AIDS infection

```
#Page 24

p_transmit = 1/500

p_no_transmit = 1 - p_transmit

p_no_infection = round(p_no_transmit**500,2)

print(p_no_infection)

p_infection = round(1 - p_no_infection ,2)

```

```
print(p_infection)
```

R code Exa 1.6.F Circuit fail

```
#Page 25

circuit = function(n,p){
  (1-p)**n
}

n = 10
p = 0.05

circuit_work = round(circuit(n,p),2)
circuit_fail = 1 - circuit_work

print(c(circuit_work, circuit_fail))

system_fail = p**10

print(system_fail)
```

Chapter 2

Random Variables

R code Exa 2.1.2.A Tay Sachs disease

```
#Page 38

k = seq(0,4)

p_k = data.frame(k,p_k = round(dbinom(k,4,0.25),3))

print(p_k)
```

R code Exa 2.1.2.B Message receiving error

```
#Page 39

binom_approx = function(n,k,p){
  r = seq(0,k)
  sum(choose(n,r)*(p**r)*((1-p)**(n-r)))
}

p_2_error = binom_approx(5,2,0.1)
```

```

print(p_2_error)
```

R code Exa 2.1.3.A State lottery frequency function

```
#Page 40

p = 1/9

ticket = seq(1,50)

plot(ticket, dgeom(ticket,p), ylab = "p(x)")
```

R code Exa 2.1.3.B Distribution of ticket

```
#Page 41

neg_bino_2 = function(k,p){
  (k-1)*p**2*(1-p)**(k-2)
}

x = seq(1,50)
p = 1/9

plot(x,neg_bino_2(x,p))
```

R code Exa 2.1.4.A Lottery mass function

```
#Page 42

```

```r
k = seq(0,6)

data.frame(k, p_k = dhyper(k,6,47,6))
```

R code Exa 2.1.5.A Binomial and poisson approximation

```r
#page 44

k = seq(0,11)
prob = 1/36

p_binom = round(dbinom(k, size = 100, p = prob),4)

p_pois = round(dpois(k, lambda = 100*prob),4)

approximation = data.frame(k, binomial_probability =
    p_binom, poisson_probability = p_pois)

print(approximation)

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

R code Exa 2.1.5.B Horse kick fatalities

```r
#Page 45

death = seq(0,4)

freq = c(109,65,22,3,1)

rel_freq = freq/sum(freq)

```

```
p_pois = round(dpois(death, lambda = 0.61),3)

kicks = data.frame(death, freq, rel_freq,p_pois)

print(kicks)
```

R code Exa 2.1.5.C Telephone call poisson process

```
#Page 46

lambda_process = 0.5

parameter = 5*lambda_process

p_no_call = round(exp(-parameter),3)

p_one_call = round(dpois(1,lambda = parameter),3)

print(c(p_no_call, p_one_call))
```

R code Exa 2.1.5.D Poisson process simulation

```
#Page 46

 a1 = rpois(20, lambda = 25)
 a2 = rpois(20, lambda = 25)
 a3 = rpois(20, lambda = 25)
 a4 = rpois(20, lambda = 25)

 par(mfrow = c(2,2))
 plot(a1/mean(a1))
 plot(a2/mean(a2))
 plot(a3/mean(a3))
plot(a4/mean(a4))
```

R code Exa 2.2.C Quartile of distribution Function

```
#Page 48


F = function(n){
  if (n < 0) return(0) else if ( n <= 1) return(n^2)
      else return(1)
}

F_inv = function(n){
  if (n < 0) return(0) else if ( n <= 1) return(sqrt
     (n)) else return(1)
}

median = F_inv(0.5)
low_quart = F_inv(0.25)
up_quart = F_inv(0.75)

print(round(c(median, low_quart, up_quart),3))
```

R code Exa 2.3.A Standardized IQ scores

```
#Page 60

mean = 100
sd = 15

prob = round(pnorm(130, mean = mean, sd = sd) -
   pnorm(120, mean = mean, sd = sd),3)

```

```
print(prob)

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

R code Exa 2.3.B Normal probability

```
#Page 61


prob = round(pnorm(1) - pnorm(-1),2)

print(prob)
```

R code Exa 2.3.D CDF of uniform distribution

```
#Page 62

F = expression(1 - 1/v)

f = D(F, "v")

print(f)
```

Chapter 3

Joint Distributions

R code Exa 3.3.A Joint density plot

```
#Page 75

library(Ryacas)
bi_density = function(x,y) (x^2 + x*y)*12/7

x = yac_symbol("x")
y = yac_symbol("y")


integrand = integrate(bi_density(x,y), y, 0, "x")

prob = integrate(integrand, x, 0, 1)
print(prob)

x = y = seq(0,1, length = 10)
z = outer(x,y,bi_density)

persp(x,y,z,theta = 30, phi = 30,ticktype = "
    detailed")
```

R code Exa 3.3.B Marginal Disstribution

```r
#Page 76

library(Ryacas)
bi_density = function(x,y) (x^2 + x*y)*12/7

x = yac_symbol("x")
y = yac_symbol("y")

marginal_x = simplify(integrate(bi_density(x,y), y,
    0, 1))

marginal_y = simplify(integrate(bi_density(x,y), x,
    0, 1))

print(c(marginal_x, marginal_y))
```

R code Exa 3.3.C Farlie Morgenstern Family

```r
#Page 77

H_neg_1 = expression(x^2*y + x*y^2 - x^2*y^2)
H_1 = expression(2*x*y - x^2*y - x*y^2 + x^2*y^2)


h_neg_1 = D(D(H_neg_1, "x"),"y")
h_1 = D(D(H_1, "x"),"y")

print(c(h_neg_1, h_1))

x = y = seq(0,1, length = 30)
```

```r
z_neg_1 = outer(x,y,function(x,y) eval(h_neg_1))
z_1 = outer(x,y,function(x,y) eval(h_1))


persp(x,y,z_neg_1,theta = 30, phi = 30,ticktype = "
    detailed")
persp(x,y,z_1,theta = 30, phi = 30,ticktype = "
    detailed")
```

R code Exa 3.3.E Random point in a disk

```r
#Page 81

library(Ryacas)

disk = function(x) (sqrt(1 - x^2))
area = function(x,y) 1/pi + 0*x*y

x = yac_symbol("x")
y = yac_symbol("y")

marginal = integrate(area(x,y), y, as.character(-
    disk(x)), as.character(disk(x)))

print(marginal)

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

R code Exa 3.5.1.A Conditional Distribution frequency

```r
#Page 87

```

```
library(MASS)

disc = fractions(c(1/8, 2/8, 1/8, 0, 0, 1/8, 2/8, 1/
   8))

table = fractions(matrix(disc, byrow = TRUE, nrow =
   2))

p_y_1 = sum(table[,2])

p_0_1 = table[1,2]/p_y_1
p_1_1 = table[2,2]/p_y_1

print(fractions(c(p_0_1, p_1_1)))
```

R code Exa 3.6.2.B Jacobian finding

```
#Page 102

library(Ryacas)

r = function(x,y) (sqrt(x^2 + y^2))
theta = function(x,y) (atan(y/x))

x = yac_symbol("x")
y = yac_symbol("y")

drdx = deriv(r(x,y),"x")
drdy = deriv(r(x,y), "y")
dthetadx = deriv(theta(x,y), "x")
dthetady = deriv(theta(x,y), "y")

J = simplify(drdx*dthetady) - simplify(drdy*dthetadx
   )

```

```
print(J)

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

R code Exa 3.7.D Distribution of Range

```
#Page 106

library(Ryacas)

integrand = function(n,r,v) n*(n-1)*r^(n-2) +0*v

n = yac_symbol("n")
r = yac_symbol("r")
v = yac_symbol("v")

f_r = integrate(integrand(n,r,v), v, 0, "1-r")

print(f_r)

F_r = integrate(f_r, r, 0, "r")

print(F_r)

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

R code Exa 3.7.E Tolerance Interval

```
#Page 106

n = 100
```

```
alpha = 0.95

prob_q = round(1 - n*alpha^(n-1) + (n-1)*alpha^n,2)

print(prob_q)
```

Chapter 4

Expected Values

R code Exa 4.1.A Roulette

```
#Page 116

library(MASS)

freq = c(1,-1)
prob = c(18/38, 20/38)

expectation = fractions(sum(freq*prob))

print(expectation)
```

R code Exa 4.1.H Cauchy pseudorandom generator

```
#Page 119

set.seed(30)

n = seq(1,500)
```

```

x_n = rnorm(500)
x_c = rcauchy(500)

g_n = c()
c_n = c()

for (i in n) {
  g_i = mean(x_n[1:i])
  c_i = mean(x_c[1:i])

  g_n = c(g_n, g_i)
  c_n = c(c_n, c_i)
}

par(mfrow = c(2,1))
plot(n,abs(g_n), ylim = c(0,1))

plot(n,c_n)

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

R code Exa 4.1.2.B Coupon Collection

```
#Page 127

library(Ryacas)

e_xr = function(n,r) n/(n-r+1)

n = 10
r = yac_symbol("r")

e_x = sum(e_xr(n,r), r, 1, n)
```

```r

print(round(as_r(e_x),1))

e_x_appox = function(n) n*(log(n) -digamma(1))

print(round(e_x_appox(n), 1))
```

R code Exa 4.1.2.C Group Testing

```r
#Page 129

prop_n = function(k,p) 1 + 1/k -p^k

p = 0.99
k = seq(1,20)

prop = prop_n(k,p)

plot(k, prop, ylab = "Proportion")
```

R code Exa 4.2.C Uniform Distribution Variance

```r
#Page 132

library(MASS)

e_x = 1/2

e_x2 = integrate(function(x) x^2, 0, 1)

var_x = fractions( e_x2$value - (e_x)^2 )

print(var_x)
```

R code Exa 4.3.A Bivariate covariance

```
#Page 138

library(Ryacas)
library(MASS)

bi_f = function(x,y) 2*x + 2*y - 4*x*y

x = yac_symbol("x")
y = yac_symbol("y")

e_xy = integrate(bi_f(x,y)*x*y, x, 0, 1)
e_xy = integrate(e_xy, y, 0, 1)

print(e_xy)

e_x = e_y = 1/2

cov_xy = fractions(as_r(e_xy) - e_x*e_y)

print(cov_xy)
```

R code Exa 4.3.D Bivariate correlation coefficient

```
#Page 142

library(MASS)

var_x = var_y = 1/12
cov_xy = -1/36
```

```r

corr_coef = fractions(cov_xy/sqrt(var_x*var_y))

print(corr_coef)
```

R code Exa 4.4.1.E Random Sums

```r
#Page 151

E_x = 1000
var_n = 900

E_n = 900
var_x = 500

var_t = E_x^2 * var_n + E_n * var_x^2

sd = sqrt(var_t)

cat(sd, var_t)
```

R code Exa 4.6.B Accuracy of approximations

```r
#Page 163
library(MASS)

g_x = function(x) sqrt(x)
x = seq(0,2,0.001)
plot(x, g_x(x), type = "l")

e_y = fractions(integrate(g_x, 0,1)$value)
print(e_y)
e_y2 = 1/2
```

```

var_y = e_y2 - e_y^2
print(var_y)
sd_y = sqrt(var_y)
print(c("Exact results", round(e_y,3), round(var_y
   ,3), round(sd_y,3)))

g_1_x = D(expression(sqrt(p)), "p")
g_2_x = D(g_1_x, "p")


mu = 1/2
var = 1/12


app_e_y = g_x(mu) + 1/2*var*eval({p = mu; g_2_x})
app_var_y = var * eval({p = mu; g_1_x})^2
app_sd_y = sqrt(app_var_y)
print(c("Approximate Results", round(app_e_y,3),
   round(app_var_y,3), round(app_sd_y,3)))
```

Chapter 5

Limit Theorems

R code Exa 5.2.A Monte Carlo Integration

```r
#Page 179

set.seed(1)
I_f = round(pnorm(1) - pnorm(0),4)

x = runif(1000)

app_I_f = 1/1000*(1/sqrt(2*pi))*sum(exp(-x^2/2))

cat("Exact", I_f, "Approximation", app_I_f)

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

R code Exa 5.3.B Particle emission from poisson process

```r
#Page 183

```

```
prob_pois = 1 -ppois(950, lambda = 900)

prob_norm = 1 - pnorm(5/3)

cat("Actual", round(prob_pois,5), "Approx", round(
   prob_norm, 6))
```

R code Exa 5.3.C Approximating uniform distribution as Normal density

```
#Page 185

set.seed(39)
x = c()

for(i in 1:1000){
  x_i = runif(12, -1/2, 1/2)
  x = append(x, sum(x_i))
}

miu = mean(x)
sd = sqrt(var(x))

freq = hist(x, xlim = c(-4,6))
height = max(freq$counts)/dnorm(miu,miu,sd)
curve(dnorm(x,miu,sd)*height, add = TRUE, col = "
   dark blue")

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

R code Exa 5.3.E Measurement Error

```
#Page 186
```

```

c = 0.5
n = 16
sigma = 1

prob = pnorm(c*sqrt(n)/sigma) - pnorm(-c*sqrt(n)/
    sigma)

print(prob)
```

R code Exa 5.3.F Normal Approximation to Binomial Density

```
#Page 187

p = 0.5
n = 100

miu = n*p
sd = sqrt(n*p*(1-p))

x = 60

prob_approx = 1 - pnorm((x - miu)/sd)

print(round(prob_approx,4))
```

Chapter 7

Survey Sampling

R code Exa 7.3.1.A Simulation of sampling distribution

```

sample = function(n) replicate(500, mean(runif(n, 0,
    2000)))

par(mfrow = c(4,1))
hist(sample(8), xlim = c(0,2000))
hist(sample(16), xlim = c(0,2000))
hist(sample(32), xlim = c(0,2000))
hist(sample(64), xlim = c(0,2000))
```

R code Exa 7.3.1.B Sampling without replacement

```
#Page 209

var = 589.7
n = 32

N = 393
```

```

var_sample = var/sqrt(n)*sqrt(1 - (n-1)/(N-1))

print(round(var_sample,1))
```

R code Exa 7.3.1.C Sampling result applied to estimation

```
#Page 209

var = 589.7
n = 32
p = 0.654

N = 393

population_corr = sqrt(1 - (n-1)/(N-1))
std_error = sqrt(p*(1-p)/n)*population_corr

print(round(std_error,2))
```

R code Exa 7.3.2.A Standard Error of estimate

```
#Page 213

var = 589.7
n = 50

s = 614.53
X_bar = 938.5

N = 393

var_sample = s^2/n*(1 - n/N)
```

```
sd = sqrt(var_sample)

cat(var_sample,sd)
```

R code Exa 7.3.2.B Estimated standard error of true value

```
#Page 213

X_bar = 938.5

N = 393

s_x = 81.19

T = N*X_bar
s = N*s_x

cat(round(T),round(s))
```

R code Exa 7.3.2.C Standard error of Variance

```
#Page 213

p = 0.654
p_hat = 26/50

n = 50

N = 393

var_p_hat = p_hat*(1-p_hat)/(n-1)*(1 - n/N)

sd_p_hat = sqrt(var_p_hat)
```

```

error = 2*sd_p_hat

cat(round(sd_p_hat,3),round(error,3))
```

R code Exa 7.3.3.A CLT Approximation

```
#Page 215

var = 589.7
n = 64

X_bar = 938.5

N = 393

var_sample = var^2/n*(1 - n/N)
sd = round(sqrt(var_sample),1)

prob = round(1 - pnorm(100/sd),3)

cat(sd,prob)
```

R code Exa 7.3.3.B Standard error of sample mean

```
#Page 216

n = 50
sd_x = 78

X_bar = 938.35
c = 123.9

```

```r
prob = round(2 - 2*pnorm(c/sd_x),2)

print(prob)
```

R code Exa 7.3.3.C Hospital discharge problem estimate error

```r
#Page 216

p_hat = 0.52
p = 0.65

n = 50
N = 393

c = abs(p_hat - p)

sd_p = sqrt(p*(1-p)/n*(1-(n-1)/(N-1)))

prob_estimate = round(2*(1 - pnorm(2.03)),2)

print(prob_estimate)
```

R code Exa 7.3.3.D Error of condominium units

```r
#Page 219

N = 8000
n = 100

s = 0.8
X_bar = 1.6

s_x = round(s/sqrt(n)*sqrt(1 - n/N),2)
```

```
z = abs(qnorm(0.025))

cat("CI for X_bar (", round(X_bar - z*s_x,2), round(
    X_bar + z*s_x,2), ")")

T = round(N*X_bar)
s_t = round(N*s_x)

cat("CI for Total (", round(T - z*s_t), round(T + z*
    s_t), ")")

p_hat = 0.12
s_p = round(sqrt(p_hat*(1-p_hat)/(n-1)*(1 - n/N)),2)

cat("CI for population proportion (", round(p_hat -
    z*s_p,2), round(p_hat + z*s_p,2), ")")

T_p = round(N*p_hat)
s_tp = round(N*s_p)

cat("CI for number population planning (", round(T_p
     - z*s_tp), round(T_p + z*s_tp), ")")
```

R code Exa 7.3.3.E Error of owners selling

```
#Page 220

z = 1.96
N = 8000

p_hat = 0.12

correction = z*N*sqrt(p_hat*(1-p_hat))/200
n = 1 + round(correction**2)
```

```
print(n)
```

R code Exa 7.4.A Mortgage payment R standard error

```
#Page 223

n = 10
N = 100
X_bar = 3100
Y_bar = 868
s_y = 250
s_x = 1200
row = 0.85
R = 0.28

s_r = round(1/n*(1 - (n-1)/(N-1))/X_bar*
  sqrt(R^2*s_x^2 + s_y^2 - 2*R*row*s_x*s_y),3)

print(c("s_r",s_r))

cat("CI for r (", R - z*s_r, R + z*s_r, ")")

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

R code Exa 7.4.D Precision of ratio estimate

```
#Page 226

miu_x = 274.8
miu_y = 814.6
r = 2.96
```

```

sd_x = 213.2
sd_y = 589.7
row = 0.91

n = 64
N = 500

var_y_bar = (r^2*sd_x^2 + sd_y^2 - 2*r*row*sd_x*sd_y
   )/n

sd_y_bar = sqrt(var_y_bar)

print(sd_y_bar)

sd = 589.7
sd_y_bar_simple = sd*sqrt((1-(n-1)/(N-1))/n)

print(sd_y_bar_simple)

ratio = (var_y_bar*n) /sd^2

print(round(ratio,4))

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

R code Exa 7.5.2.A Size stratification of Hospitals

```
#Page 230

N = c(98,98,98,99)
W = c(0.249, 0.249, 0.249, 0.251)
miu = c(182.9,526.5,956.3,1591.2)
sd = c(103.4, 204.8, 243.5, 419.2)
```

```

hospital = data.frame(N,W,miu,sd)

var = 4*sum(W^2*sd^2)
sd_x_s = sqrt(var)

print(sd_x_s)
```

R code Exa 7.5.2.B CI for population mean of Hospital strata

```
#Page 231

N = c(98,98,98,99)
W = c(0.249, 0.249, 0.249, 0.251)
miu = c(182.9,526.5,956.3,1591.2)
sd = c(103.4, 204.8, 243.5, 419.2)

hospital = data.frame(N,W,miu,sd)

x_bar = c(240.6, 507.4, 865.1, 1716.5)
s_var = c(6827.6, 23790.7, 42573, 152099)

n = 10

X_s = mean(x_bar)
var_x = round(1/n*sum(W^2*(1 - (n-1)/(N-1))*s_var)
   ,1)

sd_x = sqrt(var_x)

cat(X_s, var_x, sd_x)
cat("CI for X_bar (", round(X_s - 1.96*sd_x,2),
   round(X_s + 1.96*sd_x,2), ")")

T_s = sum(N)*X_s
```

```
s_t = sum(N)*sd_x
cat("CI for T_s (", round(T_s - 1.96*s_t), round(T_s
    + 1.96*s_t), ")")

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

R code Exa 7.5.3.A Weight allocation of hospital strata

```
#Page 234

N = c(98,98,98,99)
W = c(0.249, 0.249, 0.249, 0.251)
miu = c(182.9,526.5,956.3,1591.2)
sd = c(103.4, 204.8, 243.5, 419.2)

hospital = data.frame(N,W,miu,sd)

weight = W*sd/sum(W*sd)

print(round(weight,3))
```

R code Exa 7.5.3.B Proportional allocation of Hospital strata

```
#Page 236

N = c(98,98,98,99)
W = c(0.249, 0.249, 0.249, 0.251)
miu = c(182.9,526.5,956.3,1591.2)
sd = c(103.4, 204.8, 243.5, 419.2)

hospital = data.frame(N,W,miu,sd)

var_ratio = 1 + sum(W*(sd-mean(sd))^2)/sum(W*sd)^2
```

```
print(round(var_ratio,3))

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

R code Exa 7.5.3.C Optimal allocation improvement

```
#Page 237

N = c(98,98,98,99)
W = c(0.249, 0.249, 0.249, 0.251)
miu = c(182.9,526.5,956.3,1591.2)
sd = c(103.4, 204.8, 243.5, 419.2)

hospital = data.frame(N,W,miu,sd)

var_ratio = 1 + sum(W*(miu-mean(miu))^2)/sum(W*sd^2)

print(round(var_ratio,3))

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

Chapter 8

Estimation of Parameters and Fitting of Probability Distributions

R code Exa 8.4.A Moments of Poisson Distribution

```
#Page 261

fibre = c(31, 29, 19, 18, 31, 28, 34, 27, 34, 30,
          16, 18, 26, 27, 27, 18, 24, 22, 28, 24,
             21, 17, 24)

lambda_hat = round(mean(fibre),1)

s_lamda = round(sqrt(lambda_hat/length(fibre)),2)

print(c(lambda_hat, s_lamda))
```

R code Exa 8.4.C Fitting of Gamma Distribution

```
#Page 263

X_bar = 0.224
sigma_hat_2 = 0.1338

lambda_hat = round(X_bar/sigma_hat_2,3)
alpha_hat = round(X_bar^2/sigma_hat_2,3)

print(c(lambda_hat,alpha_hat))
```

R code Exa 8.4.D Method of moments distribution of Angular Distribution

```
#Page 119

library(Ryacas)

f = function(x,a) x*(1 + a*x)/2

x = yac_symbol("x")
a = yac_symbol("a")

miu = integrate(f(x,a),"x",-1,1)

print(simplify(miu))
```

R code Exa 8.5.1.A Hardy Weinberg Equilibrium

```
#Page 273

x = c(342,500,187)

theta_hat = (2*x[3] + x[2])/(2*sum(x))
```

```

print(round(theta_hat,4))
```

R code Exa 8.5.3.A MLE simulation of mu and sigma squared

```
#Page 279

library(plotrix)
x_bar = c()
sd_hat = c()

for(i in 1:20){
  x = rnorm(11, mean = 10, sd = 9)
  print(mean(x))
  x_bar = c(x_bar,mean(x))
  sd_hat = c(sd_hat,sd(x))
}

n = 11
alpha = 0.9
lower_sd = n*sd_hat^2/qchisq(alpha/2, df = n-1)
upper_sd = n*sd_hat^2/qchisq(1-alpha/2, df = n-1)

plotCI(x = 1:20, y = sd_hat, li = upper_sd, ui =
   lower_sd, y_lim = c(7,12))
```

R code Exa 8.5.3.B Poisson distribution MLE

```
#Page 282

X_bar = 24.9
n = 23
alpha = 0.9
```

```

s_lamda = round(sqrt(X_bar/n),2)
print(s_lamda)

z = abs(qnorm((1-alpha)/2))

uplim = round(X_bar + z*s_lamda,2)
lowlim = round(X_bar - z*s_lamda,2)

cat("CI for lambda hat is (", lowlim, uplim, ")")
```

R code Exa 8.5.3.C Hardy Weinberg Equilibrium

```
#page 283

theta_hat = 0.4247
n = 1029

s_theta = sqrt(theta_hat*(1 - theta_hat)/(2*n))

cat("CI for theta (", round(theta_hat - 1.96*s_theta
    , 3), round(theta_hat + 1.96*s_theta, 3), ")")
```

R code Exa 8.5.3.D Bootstrap estimate of Hardy Weinberg equilibrium problem

```
#Page 284

quan_25 = 0.403
quan_975 = 0.446

theta_hat = 0.425

```

```r
d = quan_25 - theta_hat
d_bar = quan_975 - theta_hat

cat("CI for theta", theta_hat - d_bar, theta_hat - d
    )
```

R code Exa 8.5.3.E Bootstrap to find CI for Gamma Distribution

```r
#Page 285

quan_50 = 0.419
quan_950 = 0.538

alpha_bar = 0.471

d = quan_50 - alpha_bar
d_bar = quan_950 - alpha_bar

cat("CI for theta", alpha_bar - d_bar, alpha_bar - d
    )
```

R code Exa 8.7.A Muon decay

```r
#Page 299

alpha = seq(0.1,0.9,0.1)
alpha = append(alpha,0.95)

eff = 2*alpha^3/(3-alpha^2)*(1/(log((1+alpha)/(1-
    alpha)) - 2*alpha))

print(data.frame(alpha, round(eff,3)))
```

R code Exa 8.7.1.A Approximation to insect count problem

```r
# Page 304
x = 0:7
count = c(70, 38, 17,10,9,3,2,1)

lambda = sum(count*x)/150
pois_dist = round(dpois(x,lambda)*150,1)

neg_dist_prob = function(m,k,n){
  if(n ==0 ) p = (1 + (m/k))**(-k)
  else p = (k + n -1)/n*(m/(k+m))* neg_dist_prob(m,k
     ,(n-1))
}

m = 1.146
k = 1.025
neg_bin_dist = round(sapply(x, neg_dist_prob, m = m,
    k = k)*150,1)

data.frame(Number_per_leaf = x, Observed_Count =
   count, Poisson_Distribution = pois_dist, Negative
   _Binomial_Distribution = neg_bin_dist)

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

Chapter 9

Testing Hypotheses and Assessing Goodness of Fit

R code Exa 9.5.A Hardy Weinberg Equilibrium data fitting

```
#Page 343

blood = matrix(c(342,500,187,340.6,502.8, 185.6),
   nrow = 2, ncol = 3, byrow = TRUE,
                  dimnames = list(c("O","E"),c("M","MN"
                      , "N")))
print(blood)

X_2 = sum((blood["O",]-blood["E",])^2/blood["E",])

print(X_2)

log_lambda = round(2*sum(blood["O",]*log(blood["O",]
   /blood["E",])),4)
max_liklihood_ratio = round(exp(-log_lambda/2),2)

cat(log_lambda, max_liklihood_ratio)
```

R code Exa 9.5.B Chi squared statistic of Bacterial Clumps

```r
#Page 344

n = 400
no = c(0:10,19)
freq = c(56,104,80,62,42,27,9,9,5,3,2,1)

lambda = sum(no*freq)/n

print(lambda)

o = c(freq[1:7],sum(freq[8:length(freq)]))
e = round(dpois(no[1:7], lambda = lambda)*n,1)
e = append(e,round(400*(ppois(10,lambda)-ppois(6,
    lambda)),1))
x_2 = round((o-e)^2/e,2)

table = data.frame(o,e,x_2)
print(table)

chi_sq = sum(x_2)
print(chi_sq)
```

R code Exa 9.5.C Fishers Reexamination of Mendels Data

```r
#Page 345

o = c(315,108,102,31)
freq = c(9/16,3/16,3/16,1/16)
n = 556

```

```
e = freq*n

table = data.frame(o,e)

log_lambda = 2*sum(o*log(o/e))

print(round(log_lambda,3))
```

R code Exa 9.6.A Poisson Dispersion test of Asbestos Fibers

```
#Page 348

library(lmtest)
fibre = c(31, 29, 19, 18, 31, 28, 34, 27, 34, 30,
          16, 18, 26, 27, 27, 18, 24, 22, 28, 24,
             21, 17, 24)

x_bar = round(mean(fibre),1)

pois = 1/x_bar*sum((fibre-x_bar)^2)
print(pois)

liklihood = 2*sum(fibre*log(fibre/x_bar))
print(liklihood)
```

R code Exa 9.6.B Poisson Distribution fitting of Bacterial clumps

```
#Page 348

n = 400
no = c(0:10,19)
freq = c(56,104,80,62,42,27,9,9,5,3,2,1)

```

```
x_bar = sum(no*freq)/n

print(lambda)

var_hat = sum(no^2*freq)/n - x_bar^2

print(var_hat)

T = n*var_hat/x_bar
print(T)

df = n-1
P_val = (T - df)/sqrt(2*df)

pob = 1 - pnorm(P_val)
print(pob)
```

R code Exa 9.8.A Michelsons determinations of the velocity of light

```
#Page 355

data = c(850, 960, 880, 890, 890, 740,
         940, 880, 810, 840, 900, 960,
         880, 810, 780, 1070, 940, 860,
         820, 810, 930, 880, 720, 800,
         760, 850, 800, 720, 770, 810,
         950, 850, 620, 760, 790, 980,
         880, 860, 740, 810, 980, 900,
         970, 750, 820, 880, 840, 950,
         760, 850, 1000, 830, 880, 910,
         870, 980, 790, 910, 920, 870,
         930, 810, 850, 890, 810, 650,
         880, 870, 860, 740, 760, 880,
         840, 880, 810, 810, 830, 840,
         720, 940, 1000, 800, 850, 840,
```

```r
        950, 1000, 790, 840, 850, 800,
        960, 760, 840, 850, 810, 960,
        800, 840, 780, 870)
qqnorm(data)
```

R code Exa 9.8.B Normal probability plot of double exponential distribution

```r
#Page 356

library(nimble)

rand = rdexp(500)
qqnorm(rand)
```

R code Exa 9.8.C Gamma probability plot of rainfall distribution

```r
#Page 357

rand = rgamma(500,5)
qqnorm(rand)
```

Chapter 10

Summarizing Data

R code Exa 10.2.1.A Chemical properties of beeswax

```
#Page 378

data = c(63.78, 63.45, 63.58, 63.08, 63.40, 64.42,
   63.27, 63.10,
         63.34, 63.50, 63.83, 63.63, 63.27, 63.30,
            63.83, 63.50,
         63.36, 63.86, 63.34, 63.92, 63.88, 63.36,
            63.36, 63.51,
         63.51, 63.84, 64.27, 63.50, 63.56, 63.39,
            63.78, 63.92,
         63.92, 63.56, 63.43, 64.21, 64.24, 64.12,
            63.92, 63.53,
         63.50, 63.30, 63.86, 63.93, 63.43, 64.40,
            63.61, 63.03,
         63.68, 63.13, 63.41, 63.60, 63.13, 63.69,
            63.05, 62.85,
         63.31, 63.66, 63.60)
empirical = ecdf(data)
plot(empirical)
```

R code Exa 10.2.2.A Study of the lifetimes of guinea pigs infected with varying doses of tubercle bacilli

```
#Page 381

control_life = c(18, 36, 50, 52, 86, 87, 89, 91,
                 102, 105, 114, 114, 115, 118, 119,
                    120,
                 149, 160, 165, 166, 167, 167, 173,
                    178,
                 189, 209, 212, 216, 273, 278, 279,
                    292,
                 341, 355, 367, 380, 382, 421, 421,
                    432,
                 446, 455, 463, 474, 506, 515, 546,
                    559,
                 576, 590, 603, 607, 608, 621, 634,
                    634,
                 637, 638, 641, 650, 663, 665, 688,
                    725,
                 735)
dose_1 = c(76, 93, 97, 107, 108, 113, 114, 119,
           136, 137, 138, 139, 152, 154, 154, 160,
           164, 164, 166, 168, 178, 179, 181, 181,
           183, 185, 194, 198, 212, 213, 216, 220,
           225, 225, 244, 253, 256, 259, 265, 268,
           268, 270, 283, 289, 291, 311, 315, 326,
           326, 361, 373, 373, 376, 397, 398, 406,
           452, 466, 592, 598)

dose_2 = c(72, 72, 78, 83, 85, 99, 99, 110,
           113, 113, 114, 114, 118, 119, 123, 124,
           131, 133, 135, 137, 140, 142, 144, 145,
           154, 156, 157, 162, 162, 164, 165, 167,
               171, 176, 177, 181, 182, 187, 192, 196,
               211, 214, 216, 216, 218, 228, 238, 242,
               248, 256, 257, 262, 264, 267, 267, 270,
               286, 303, 309, 324, 326, 334, 335, 358,
               409, 473, 550)

dose_3 = c(10, 33, 44, 56, 59, 72, 74, 77,
               92, 93, 96, 100, 100, 102, 105, 107,
               107, 108, 108, 108, 109, 112, 113, 115,
               116, 120, 121, 122, 122, 124, 130, 134,
               136, 139, 144, 146, 153, 159, 160, 163,
               163, 168, 171, 172, 176, 183, 195, 196,
               197, 202, 213, 215, 216, 222, 230, 231,
               240, 245, 251, 253, 254, 254, 278, 293,
               327, 342, 347, 361, 402, 432, 458, 555)

dose_4 = c(43, 45, 53, 56, 56, 57, 58, 66,
               67, 73, 74, 79, 80, 80, 81, 81,
               81, 82, 83, 83, 84, 88, 89, 91,
               91, 92, 92, 97, 99, 99, 100, 100,
               101, 102, 102, 102, 103, 104, 107, 108,
               109, 113, 114, 118, 121, 123, 126, 128,
               137, 138, 139, 144, 145, 147, 156, 162,
               174, 178, 179, 184, 191, 198, 211, 214,
               243, 249, 329, 380, 403, 511, 522, 598)

dose_5 = c(12, 15, 22, 24, 24, 32, 32, 33,
               34, 38, 38, 43, 44, 48, 52, 53,
               54, 54, 55, 56, 57, 58, 58, 59,
               60, 60, 60, 60, 61, 62, 63, 65,
               65, 67, 68, 70, 70, 72, 73, 75,
               76, 76, 81, 83, 84, 85, 87, 91,
               95, 96, 98, 99, 109, 110, 121, 127,
               129, 131, 143, 146, 146, 175, 175, 211,
               233, 258, 258, 263, 297, 341, 341, 376)

emp_fn_cl = ecdf(control_life)
emp_fn_d1 = ecdf(dose_1)
```

```
emp_fn_d2 = ecdf(dose_2)
emp_fn_d3 = ecdf(dose_3)
emp_fn_d4 = ecdf(dose_4)
emp_fn_d5 = ecdf(dose_5)

emp_val_cl = (emp_fn_cl(control_life))
emp_val_d1 = emp_fn_d1(dose_1)
emp_val_d2 = emp_fn_d2(dose_2)
emp_val_d3 = emp_fn_d3(dose_3)
emp_val_d4 = emp_fn_d4(dose_4)
emp_val_d5 = emp_fn_d5(dose_5)

surv_val_cl = 1 - emp_val_cl
surv_val_d1 = 1 - emp_val_d1
surv_val_d2 = 1 - emp_val_d2
surv_val_d3 = 1 - emp_val_d3
surv_val_d4 = 1 - emp_val_d4
surv_val_d5 = 1 - emp_val_d5


plot(x = control_life, y =surv_val_cl, type = "l",
   xlim = c(0,800))
lines(x = dose_1, y =surv_val_d1, type = "l", xlim =
    c(0,800), lty = 2, add = TRUE)
lines(x = dose_2, y =surv_val_d2, type = "l", xlim =
    c(0,800), lty = 3, add = TRUE)
lines(x = dose_3, y =surv_val_d3, type = "l", xlim =
    c(0,800), lty = 4, add = TRUE)
lines(x = dose_4, y =surv_val_d4, type = "l", xlim =
    c(0,800), lty = 5, add = TRUE)
lines(x = dose_5, y =surv_val_d5, type = "l", xlim =
    c(0,800), lty = 6, add = TRUE)
```

R code Exa 10.2.2.B empirical survival functions of Guinea pig test

```r
#Page 384

control_life = c(18, 36, 50, 52, 86, 87, 89, 91,
                 102, 105, 114, 114, 115, 118, 119, 120,
                 149, 160, 165, 166, 167, 167, 173, 178,
                 189, 209, 212, 216, 273, 278, 279, 292,
                 341, 355, 367, 380, 382, 421, 421, 432,
                 446, 455, 463, 474, 506, 515, 546, 559,
                 576, 590, 603, 607, 608, 621, 634, 634,
                 637, 638, 641, 650, 663, 665, 688, 725,
                 735)
dose_1 = c(76, 93, 97, 107, 108, 113, 114, 119,
           136, 137, 138, 139, 152, 154, 154, 160,
           164, 164, 166, 168, 178, 179, 181, 181,
           183, 185, 194, 198, 212, 213, 216, 220,
           225, 225, 244, 253, 256, 259, 265, 268,
           268, 270, 283, 289, 291, 311, 315, 326,
           326, 361, 373, 373, 376, 397, 398, 406,
           452, 466, 592, 598)

dose_2 = c(72, 72, 78, 83, 85, 99, 99, 110,
           113, 113, 114, 114, 118, 119, 123, 124,
           131, 133, 135, 137, 140, 142, 144, 145,
           154, 156, 157, 162, 162, 164, 165, 167,
           171, 176, 177, 181, 182, 187, 192, 196,
           211, 214, 216, 216, 218, 228, 238, 242,
           248, 256, 257, 262, 264, 267, 267, 270,
           286, 303, 309, 324, 326, 334, 335, 358,
           409, 473, 550)

dose_3 = c(10, 33, 44, 56, 59, 72, 74, 77,
            92, 93, 96, 100, 100, 102, 105, 107,
            107, 108, 108, 108, 109, 112, 113, 115,
            116, 120, 121, 122, 122, 124, 130, 134,
            136, 139, 144, 146, 153, 159, 160, 163,
            163, 168, 171, 172, 176, 183, 195, 196,
            197, 202, 213, 215, 216, 222, 230, 231,
            240, 245, 251, 253, 254, 254, 278, 293,
            327, 342, 347, 361, 402, 432, 458, 555)

dose_4 = c(43, 45, 53, 56, 56, 57, 58, 66,
            67, 73, 74, 79, 80, 80, 81, 81,
            81, 82, 83, 83, 84, 88, 89, 91,
            91, 92, 92, 97, 99, 99, 100, 100,
            101, 102, 102, 102, 103, 104, 107, 108,
            109, 113, 114, 118, 121, 123, 126, 128,
            137, 138, 139, 144, 145, 147, 156, 162,
            174, 178, 179, 184, 191, 198, 211, 214,
            243, 249, 329, 380, 403, 511, 522, 598)

dose_5 = c(12, 15, 22, 24, 24, 32, 32, 33,
            34, 38, 38, 43, 44, 48, 52, 53,
            54, 54, 55, 56, 57, 58, 58, 59,
            60, 60, 60, 60, 61, 62, 63, 65,
            65, 67, 68, 70, 70, 72, 73, 75,
            76, 76, 81, 83, 84, 85, 87, 91,
            95, 96, 98, 99, 109, 110, 121, 127,
            129, 131, 143, 146, 146, 175, 175, 211,
            233, 258, 258, 263, 297, 341, 341, 376)

emp_fn_cl = ecdf(control_life)
emp_fn_d1 = ecdf(dose_1)
emp_fn_d2 = ecdf(dose_2)
emp_fn_d3 = ecdf(dose_3)
emp_fn_d4 = ecdf(dose_4)
emp_fn_d5 = ecdf(dose_5)

emp_val_cl = emp_fn_cl(control_life)
emp_val_d1 = emp_fn_d1(dose_1)
```

```
emp_val_d2 = emp_fn_d2(dose_2)
emp_val_d3 = emp_fn_d3(dose_3)
emp_val_d4 = emp_fn_d4(dose_4)
emp_val_d5 = emp_fn_d5(dose_5)

surv_val_cl = log10(1 - emp_val_cl)
surv_val_d1 = log10(1 - emp_val_d1)
surv_val_d2 = log10(1 - emp_val_d2)
surv_val_d3 = log10(1 - emp_val_d3)
surv_val_d4 = log10(1 - emp_val_d4)
surv_val_d5 = log10(1 - emp_val_d5)


plot(x = control_life, y =surv_val_cl, type = "l",
   xlim = c(0,800))
lines(x = dose_1, y =surv_val_d1, type = "l", xlim =
    c(0,800), lty = 2, add = TRUE)
lines(x = dose_2, y =surv_val_d2, type = "l", xlim =
    c(0,800), lty = 3, add = TRUE)
lines(x = dose_3, y =surv_val_d3, type = "l", xlim =
    c(0,800), lty = 4, add = TRUE)
lines(x = dose_4, y =surv_val_d4, type = "l", xlim =
    c(0,800), lty = 5, add = TRUE)
lines(x = dose_5, y =surv_val_d5, type = "l", xlim =
    c(0,800), lty = 6, add = TRUE)
```

R code Exa 10.4.2.A Cumulative binomial probabilities of Platinum data

```
#Page 396

k = 5:9
n = 26

p_binom = round(pbinom(k,26,0.5),4)
data.frame(k,p_binom)
```

```

r = 8

P_lessk = round(pbinom(r,26,0.5),4) - round(dbinom(r
    ,26,0.5),4)
print(P_lessk)

i = 19
P_great = 1 - round(pbinom(i-1,26,0.5),4)
print(P_great)
```

Chapter 11

Comparing two Samples

R code Exa 11.2.1.A Difference in Latent Heat of Fusion

```r
#Page 423

A = c(79.98, 80.04, 80.02, 80.04, 80.03,80.03,
    80.04, 79.97, 80.05, 80.03, 80.02, 80, 80.02)
B = c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03,
    79.95, 79.97)

heat = data.frame(A,B = c(B,rep(NA, length(A)-length
    (B))))

X_bar_A = round(mean(A),3)
X_bar_B = round(mean(B),3)

sd_A = round(sd(A),3)
sd_B = round(sd(B),3)

cat(X_bar_A, X_bar_B, sd_A, sd_B)

var_p = ((length(A)-1)*sd_A^2 + (length(B)-1)*sd_B
    ^2)/(length(A) + length(B) - 2)
sd_p = round(sqrt(var_p),3)
```

```
print(sd_p)

diff = round(X_bar_A - X_bar_B,2)
s_diff = round(sd_p*sqrt(1/length(A) + 1/length(B))
    ,3)

cat(diff, s_diff)

boxplot(heat, ylim = c(79.94, 80.06))

t_val = round(abs(qt(0.025, df = length(A) + length(
    B) -2)),3)

print(t_val)

cat("CI for mean diff is (", round(diff - t_val*s_
    diff,3), round(diff + t_val*s_diff,3), ")")
```

R code Exa 11.2.1.B Two sided alternative of mean test of two methods

```
#Page 425

A = c(79.98, 80.04, 80.02, 80.04, 80.03,80.03,
    80.04, 79.97, 80.05, 80.03, 80.02, 80, 80.02)
B = c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03,
    79.95, 79.97)

X_bar_A = round(mean(A),2)
X_bar_B = round(mean(B),2)
print(X_bar_A)

sd_A = round(sd(A),3)
sd_B = round(sd(B),3)

```

```
var_p = ((length(A)-1)*sd_A^2 + (length(B)-1)*sd_B
    ^2)/(length(A) + length(B) - 2)
sd_p = round(sqrt(var_p),3)

diff = round(X_bar_A - X_bar_B,2)
s_diff = round(sd_p*sqrt(1/length(A) + 1/length(B))
    ,3)

t_stat = round(diff/s_diff,3)

print(t_stat)

t_val = abs(qt(0.005, df = length(A) + length(B) -
    2))

print(t_val)
```

R code Exa 11.2.1.C Mean test without assumption of equal variance

```
#Page 428

A = c(79.98, 80.04, 80.02, 80.04, 80.03,80.03,
    80.04, 79.97, 80.05, 80.03, 80.02, 80, 80.02)
B = c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03,
    79.95, 79.97)

X_bar_A = round(mean(A),3)
X_bar_B = round(mean(B),3)

n = length(A)
m = length(B)

var_A = round(var(A),5)
var_B = round(var(B),5)

```

```
diff = round(X_bar_A - X_bar_B,2)
s_diff = round(sqrt(var_A/n + var_B/m),4)

t_stat = abs(diff)/sqrt(var_A/n + var_B/m)
print(t_stat)

df = round(((var_A/n) + (var_B/m))^2/(((var_A/n)^2/(
   n-1)) + ((var_B/m)^2/(m-1))),1)

print(df)

t_val = qt(0.995, df = df)

print(t_val)

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

R code Exa 11.2.2.A Power vs del plot for iron retention experiment

```
#Page 434

del = 1
sd = 5

z = qnorm(0.1)

n = round(((1.96 - z)*sd/del)^2 * 2)

print(n)
```

R code Exa 11.2.3.A Mann Whitney test of latent heat of fusion

```
#Page 437

A = c(79.98, 80.04, 80.02, 80.04, 80.03,80.03,
   80.04, 79.97, 80.05, 80.03, 80.02, 80, 80.02)
B = c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03,
   79.95, 79.97)

n = length(A)
m = length(B)

rn_A = rank(append(A,B), ties.method = "average")
A_rank = rn_A[1:n]
B_rank = rn_A[14:21]

print(list(A_rank, B_rank))

R = sum(B_rank)
R_dash = min(m,n)*(m+n+1) - R

print(c(R,R_dash))

R_star = min(R,R_dash)

print(R_star)
```

R code Exa 11.2.3.B Normal Distribution of Rank sum method

```
#Page 441

n = 13
m = 8

T = 51

E_T = min(m,n)*(m+n+1)/2
```

```
sigma_T = sqrt(m*n*(m+n+1)/12)

print(c(E_T, sigma_T))

t_test = round((T-E_T)/sigma_T,2)

p_val = round(2*pnorm(t_test),3)

print(c(t_test,p_val))
```

R code Exa 11.3.1.A Study of effect of cigarette smoking on platelet aggregation

```
#Page 446

before = c(25,25,27,44,30,67,53,53,52,60,28)
after = c(27,29,37,56,46,82,57,80,61,59,43)
n = length(before)

diff = after - before

print(data.frame(before,after,difference = diff))

D_bar = mean(diff)
sd_D_bar = sqrt(var(diff)/n)

print(c(D_bar,sd_D))

t = round(abs(qt(0.05, df = 10)),3)

cat("CI for D_bar (", round(D_bar - t*sd_D,3), round
   (D_bar + t*sd_D,3), ")")

plot(before,after)
```

Chapter 12

The Analysis of Variance

R code Exa 12.2.1.A F statistic applied to the tablet data

```
#Page 483

I = 7
J = 10
lab = matrix(c(4.13, 3.86, 4.00, 3.88, 4.02, 4.02,
   4.00,
               4.07, 3.85, 4.02, 3.88, 3.95, 3.86,
                  4.02,
               4.04, 4.08, 4.01, 3.91, 4.02, 3.96,
                  4.03,
               4.07, 4.11, 4.01, 3.95, 3.89, 3.97,
                  4.04,
               4.05, 4.08, 4.04, 3.92, 3.91, 4.00,
                  4.10,
               4.04, 4.01, 3.99, 3.97, 4.01, 3.82,
                  3.81,
               4.02, 4.02, 4.03, 3.92, 3.89, 3.98,
                  3.91,
               4.06, 4.04, 3.97, 3.90, 3.89, 3.99,
                  3.96,
               4.10, 3.97, 3.98, 3.97, 3.99, 4.02,
                  4.05,
               4.04, 3.95, 3.98, 3.90, 4.00, 3.93,
                  4.06), byrow = TRUE, nrow = J, ncol
                   = I)

mean = mean(lab)
mean_i = c()
ss_w = 0
residue = c()

for(i in 1:7){
  y_i = mean(lab[,i])
  mean_i = c(mean_i,y_i)
  residue = c(residue, lab[,i]-mean_i[i])
  ss_w = ss_w + sum((lab[,i]-mean_i[i])^2)
}

ss_b = round(10*sum((mean_i - mean)^2),3)

ss_total = round(sum((lab-mean)^2),3)

df = c(I-1,I*(J-1),I*J-1)
ss = c(ss_b,ss_w,ss_total)
ms = round(ss/df,4)
f = ss_b/(I-1)/(ss_w/(I*(J-1)))

var_tab = data.frame(df,ss,ms,f)

print(var_tab)

qqnorm(residue,
       ylab="Oredered Residuals", xlab="Normal
          Quantiles")
```

R code Exa 12.2.2.1.A Turkey method application to Tablet data

```r
#Page 486

library(dplyr)
I = 7
J = 10
lab = matrix(c(4.13, 3.86, 4.00, 3.88, 4.02, 4.02,
   4.00,
                    4.07, 3.85, 4.02, 3.88, 3.95, 3.86,
                       4.02,
                    4.04, 4.08, 4.01, 3.91, 4.02, 3.96,
                       4.03,
                    4.07, 4.11, 4.01, 3.95, 3.89, 3.97,
                       4.04,
                    4.05, 4.08, 4.04, 3.92, 3.91, 4.00,
                       4.10,
                    4.04, 4.01, 3.99, 3.97, 4.01, 3.82,
                       3.81,
                    4.02, 4.02, 4.03, 3.92, 3.89, 3.98,
                       3.91,
                    4.06, 4.04, 3.97, 3.90, 3.89, 3.99,
                       3.96,
                    4.10, 3.97, 3.98, 3.97, 3.99, 4.02,
                       4.05,
                    4.04, 3.95, 3.98, 3.90, 4.00, 3.93,
                       4.06), byrow = TRUE, nrow = J,
                       ncol = I)
mean_i = c()

for(i in 1:7){
  y_i = mean(lab[,i])
  mean_i = c(mean_i,y_i)
}

i = 1:7
labs_df = data.frame(labs =i, means = mean_i)
labs_df = arrange(labs_df, desc(means))
print(labs_df)

```

```r
s_err = 0.0037
sp = round(sqrt(s_err),3)

q_val = qtukey(0.95,7,63)

differ = round(q_val*sp/sqrt(J),3)

t_val = qt(1-0.025,63)
differ_t = round(t_val*sp*sqrt(2/J),3)

print(c(differ, differ_t))

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

R code Exa 12.3.3.A An experimental study of drugs

```r
#Page 501

I = 7
J = 10
itch_data = matrix(c(174, 263, 105, 199, 141, 108,
   141,
                    224, 213, 103, 143, 168, 341, 184,
                    260, 231, 145, 113, 78,159, 125,
                    255, 291, 103, 225, 164, 135, 227,
                    165, 168, 144, 176, 127, 239, 194,
                    237, 121, 94, 144, 114, 136, 155,
                    191, 137, 35, 87, 96, 140, 121,
                    100, 102, 133, 120, 222, 134, 129,
                    115, 89, 83, 100, 165, 185, 79,
                    189, 433, 237, 173, 168, 188, 317),
                       byrow = TRUE,ncol = 7)

itch = data.frame(no_drug = itch_data[,1], placebo =
    itch_data[,2], papaverine = itch_data[,3],
                   nmorphine = itch_data[,4], amino =
                       itch_data[,5], pentobarbital =
                       itch_data[,6],
                   tripelennamine = itch_data[,7],
                      row.names = c("BG","JF","BS","
                      SI","BW","TS","GM","SS","MU","
                      OS"))
boxplot(itch)

stack_data = stack(itch)
stack_data <- cbind(stack_data,subject = rep(
   rownames(itch),ncol(itch)))
one.way <- aov(values~ind+subject, data = stack_data
   )

summary(one.way)

y = unname(sort(residuals(one.way)))
qqnorm(y, pch = 20, xlim = c(-3,3), ylim = c
   (-100,150),
       xlab = "Normal quantiles", ylab = "Ordered
          Quantiles",
       main = "")

df = 54
nrange = 7
s = 3095
qt = qtukey(0.95, nrange,df)
error_var = qt*sqrt(s/J)
print(error_var)

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

R code Exa 12.3.4.A Friedman test on itching data

```
#Page 504

I = 7
J = 10
itch_data = matrix(c(174, 263, 105, 199, 141, 108,
   141,
                224, 213, 103, 143, 168, 341, 184,
                260, 231, 145, 113, 78,159, 125,
                255, 291, 103, 225, 164, 135, 227,
                165, 168, 144, 176, 127, 239, 194,
                237, 121, 94, 144, 114, 136, 155,
                191, 137, 35, 87, 96, 140, 121,
                100, 102, 133, 120, 222, 134, 129,
                115, 89, 83, 100, 165, 185, 79,
                189, 433, 237, 173, 168, 188, 317),
                   byrow = TRUE,ncol = 7)

itch_data = data.frame(no_drug = itch_data[,1],
   placebo = itch_data[,2], papaverine = itch_data
   [,3],
                       nmorphine = itch_data[,4],
                          amino = itch_data[,5],
                          pentobarbital = itch_data
                          [,6],
                       tripelennamine = itch_data
                          [,7], row.names = c("BG","
                          JF","BS","SI","BW","TS","
                          GM","SS","MU","OS"))

rank_df = lapply(as.data.frame(t(itch_data)), rank,
   ties.method = "average")
rank_df = as.data.frame(rank_df, row.names = names(
   itch_data))
itch_rank = as.data.frame(t(rank_df))
itch_rank = rbind(itch_rank,  Average = as.data.
   frame(lapply(itch_rank, mean)))
print(itch_rank)

```

```
R_bar = mean(unlist(itch_rank[-1,]))
R_sum = sum((itch_rank[11,]-R_bar)**2)
Q = 12*J*R_sum/(I*(I+1))

cat(R_bar, R_sum, Q)
```

Chapter 13

The Analysis of categorical data

R code Exa 13.5.B Cell Phones and Driving

```
# Page 526

table = matrix(data = c(13,24,157,505),nrow = 2,
   dimnames = list(c("On Phone", "Not on Phone"),c("
   On Phone", "Not on Phone")))
table = rbind(table, Total = colSums(table))
table = cbind(table, Total = rowSums(table))

table

X_2 = (table[1,2] - table[2,1])^2/(table[1,2] +
   table[2,1])

print(X_2)
```

Chapter 14

Linear Least squares

R code Exa 14.2.2.B Environmental impact study

```
# Page 551

env_table = data.frame(Depth = c(0.34, 0.29, 0.28,
   0.42, 0.29, 0.41, 0.76, 0.73, 0.46, 0.4),
                       Rate = c(0.636, 0.319, 0.734,
                          1.327, 0.487, 0.924,
                         7.35, 5.89, 1.979, 1.124))
plot(env_table)

env_reg = lm(Rate ~ Depth , data = env_table)
residue = resid(env_reg)

plot(env_table$Depth, residue, xlab = "Depth", ylab
   = "Residuals")

log_table = log(env_table, base = 10)
plot(log_table)

env_log_reg = lm(Rate ~ Depth, data = log_table)
log_residue = resid(env_log_reg)
plot(log_table$Depth, log_residue, xlab = "Depth",
   ylab = "Residuals")
```

R code Exa 14.4.5.B A quadratic model for stream flow data

```
# Page 579

env_table = data.frame(Depth = c(0.34, 0.29, 0.28,
   0.42, 0.29, 0.41, 0.76, 0.73, 0.46, 0.4),
                         Rate = c(0.636, 0.319, 0.734,
                              1.327, 0.487, 0.924,
                           7.35, 5.89, 1.979, 1.124))
env_table$Depth_2 = env_table$Depth**2

quadratic_model = lm(Rate ~ Depth + Depth_2, data =
   env_table)
summary(quadratic_model)

residuals = resid(quadratic_model)
plot(env_table$Depth, residuals, xlim = c(0.2,0.8),
   ylim = c(-0.6,0.4), xlab = "Depth")

x = matrix(c(rep(1,nrow(env_table)), env_table$Depth
   , env_table$Depth_2), ncol = 3)
y = matrix(env_table$Rate, ncol = 1)

sum_bb = solve(t(x)%*%x)

corr_matrix = diag(3)

for (i in 1:3){
  for(j in 1:3) if (i != j) corr_matrix[i,j] = sum_
     bb[i,j]/sqrt(sum_bb[i,i]*sum_bb[j,j])
}

print(round(corr_matrix,2))
```