R Textbook Companion for An Introduction to Statistical Methods and Data Analysis by R Lyman Ott and Michael Longnecker[1]

Created by
Himanshu Gupta
B.E.
Information Technology
SGSITS INDORE
Cross-Checked by
R TBC Team

May 26, 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: An Introduction to Statistical Methods and Data Analysis

Author: R Lyman Ott and Michael Longnecker

Publisher: Cengage Learning, Canada

Edition: 6

Year: 2010

ISBN: 0-495-01758-2

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 3

Data Description

R code Exa 3.1 Mode

```
# Page No. 78

selling_price<-c( 26.6, 25.3 ,23.8 ,24.0, 27.5,
        21.1 ,25.9 ,22.6, 23.8, 25.1,
        22.6 ,27.5 ,26.8 ,23.4, 27.5,
        20.8 ,20.4, 22.4 ,27.5, 23.7,
        22.2 ,23.8, 23.2, 28.7,27.5)
modal_selling_price<-table(selling_price)

print("Modal SP")
names(modal_selling_price)[which(modal_selling_price
   ==max(modal_selling_price))]
```

R code Exa 3.2 Median

```
# Page No. 79

percentage<-c(95 ,86 ,78 ,90, 62, 73 ,89 ,92 ,84
   ,76)
```

```
new_list<-sort(percentage)
n<-length(percentage)

if(n%%2==0) {
  a<-new_list[n/2]
  print(a)
  b<-new_list[(n/2)+1]
  print(b)
  print("median is")
  print((a+b)/2)
} else {
  m<-(n+1)/2
  print("median is")
  print(new_list[m])
}
```

R code Exa 3.3 Mode and median

```
# Page No. 80

productivity<-c(4.4 ,4.9, 4.2 ,4.4, 4.8 ,4.9, 4.8,
   4.5, 4.3 ,4.8 ,4.7 ,4.4, 4.2)
new_list<-sort(productivity)
n<-length(productivity)

if(n%%2==0) {
  a<-new_list[n/2]
  print(a)
  b<-new_list[(n/2)+1]
  print(b)
  print("median is")
  print((a+b)/2)
} else {
  m<-(n+1)/2
  print("output is")
```

```
  print(new_list[m])
}
print("mode is ")
modal_productivity<-table(productivity)
names(modal_productivity)[which(modal_productivity==max(modal_productivity))]
```

R code Exa 3.4 Median for interval data

```
# Page No. 80

Median_calculate <- function(frequencies, intervals, sep = NULL, trim = NULL) {
  if (!is.null(sep)) {
    if (is.null(trim)) pattern <- ""
    else if (trim == "cut") pattern <- "\\[|\\]|\\(|\\)"
    else pattern <- trim
    intervals <- sapply(strsplit(gsub(pattern, "", intervals), sep), as.numeric)
  }

  Midpoints <- rowMeans(intervals)
  cf <- cumsum(frequencies)
  Midrow <- findInterval(max(cf)/2, cf) + 1
  L <- intervals[1, Midrow]
  h <- diff(intervals[, Midrow])
  f <- frequencies[Midrow]
  cf2 <- cf[Midrow - 1]
  n_2 <- max(cf)/2
  unname(L + (n_2 - cf2)/f * h)
}

mydataframe <- structure(list(Class_interval = c("16.25-18.75", "18.75-21.25", "21.25-23.75", "
```

```
    23.75-26.25", "26.25-28.75", " 28.75-31.25", "
    31.25-33.75", "33.75-36.25", "36.25-38.75", "
    38.75- 41.25","41.25- 43.75"), freq = c(2L, 7L, 7
    L, 14L, 17L, 24L, 11L, 11L, 3L, 3L,1L)), .Names =
     c("class_interval", "freq"), class = "data.frame
    ", row.names = c(NA, -11L))
print(mydataframe)

Median_calculate(mydataframe$freq, mydataframe$class
    _interval, sep = "-")
```

R code Exa 3.5 Mean

```
# Page No. 82

accounts_due<-c(55.20,  4.88 ,271.95,
                 18.06, 180.29, 365.29,
                 28.16, 399.11 ,807.80,
                 44.14, 97.47, 9.98,
                 61.61, 56.89 ,82.73)
n<-sum(accounts_due)
d<-length(accounts_due)

mean_accountsdue<-n/d
print(mean_accountsdue)
total_overdueamount=150*mean_accountsdue
print(total_overdueamount)
```

R code Exa 3.6 Mean for interval data

```
# Page No. 83

```

```
ClassInterval <- c("16.25-18.75", "18.75-21.25", "
   21.25-23.75","23.75-26.25", "26.25-28.75", "
   28.75-31.25", " 31.25-33.75", "33.75-36.25","
   36.25-38.75", "38.75- 41.25", "41.25- 43.75")
freq <- c( 2,7,7,14,17,24,11,11,3,3,1)
mid_interval<- c
   (17.5,20.0,22.5,25.0,27.5,30.0,32.5,35.0,37.5,40.0,42.5)

fmi<-freq*mid_interval
List<- data.frame(ClassInterval, freq, mid_interval,
   fmi)
print(List)
print("mean is")
print(sum(fmi)/sum(freq))
```

R code Exa 3.7 Range

```
# Page No. 86

accounts_due<-c(55.20,  4.88 ,271.95,
                18.06, 180.29, 365.29,
                28.16, 399.11 ,807.80,
                44.14, 97.47, 9.98,
                61.61, 56.89 ,82.73)

range(accounts_due)
diff=max(accounts_due)-min(accounts_due)
print(diff)
```

R code Exa 3.8 Percentile

```
# Page No. 90
```

```
L <- 33.75
n <- 100
cfb <- 82
f90 <- 11
w <-2.5

P<-L+(w/f90)*(0.9*n-cfb)
print(P)
```

R code Exa 3.9 Sample varaince

```
# Page No. 92

y <- c(5,4,3,1,3)

mean_y<-sum(y)/length(y)
sample_variance <-(sum((y-mean_y)^2/(length(y)-1)))
print(sample_variance)
```

R code Exa 3.10 Variance and standard deviation

```
# Page No. 93

ClassInterval <- c("16.25-18.75", "18.75-21.25", "
    21.25-23.75","23.75-26.25", "26.25-28.75", "
    28.75-31.25", " 31.25-33.75", "33.75-36.25","
    36.25-38.75", " 38.75- 41.25","41.25- 43.75")
freq <- c( 2,7,7,14,17,24,11,11,3,3,1)
mid_interval<- c
    (17.5,20.0,22.5,25.0,27.5,30.0,32.5,35.0,37.5,40.0,42.5)

fmi<-freq*mid_interval
```

```
mean_y<-sum(fmi)/sum(freq)
sample_variance <-(sum(freq*((mid_interval-mean_y)
   ^2)/(sum(freq)-1)))
print(sample_variance)
standard_deviation <-sqrt(sample_variance)
print(standard_deviation)
```

R code Exa 3.12 Approximate value

```
# Page No. 95

y<-c(26 ,28 ,30, 37 ,33 ,30,
    29 ,39 ,49 ,31, 38 ,36,
    33 ,24, 34, 40 ,29, 41,
    40, 29, 35 ,44 ,32, 45,
    35 ,26, 42, 36 ,37 ,35)

mean_y<-sum(y)/length(y)
sample_variance <-(sum((y-mean_y)^2/(length(y)-1)))
standard_deviation <-sqrt(sum((y-mean_y)^2/(length(y
   )-1)))
s=(max(y)-min(y))/4;

print(mean_y)
print(sample_variance)
print(standard_deviation)
print(s)
```

R code Exa 3.13 Crime study

```
# Page No. 97

```

```r
crime_rate=c(876,578,718,388,562,971,698,298,673,537,642,856,376,508,529,393,

median(crime_rate)
lower_quartile=quantile(crime_rate,0.25)
lower_quartile
upper_quartile=quantile(crime_rate,0.75)
upper_quartile
IQR(crime_rate)

# The answer provided in the textbook is wrong.
```

R code Exa 3.14 Outliers

```r
# Page No. 100

crime_rate=c(876,578,718,388,562,971,698,298,673,537,642,856,376,508,529,393,

lower_quartile= 464.5
upper_quartile=718.5

iqr=IQR(crime_rate)
lower_inner_fence= lower_quartile - (1.5*iqr)
upper_inner_fence= upper_quartile + (1.5*iqr)
lower_outer_fence= lower_quartile - (3*iqr)
upper_outer_fence= upper_quartile +(3*iqr)
print(lower_inner_fence)
print(upper_inner_fence)
print(lower_outer_fence)
print(upper_outer_fence)

# The answer provided in the textbook is wrong.
```

R code Exa 3.15 Boxplot

```
# Page No. 101

crime_rate=c(876,578,718,388,562,971,698,298,673,537,642,856,376,508,529,393,

boxplot(crime_rate, horizontal = TRUE, axes = FALSE, staplewex = 1)
text(x=fivenum(crime_rate), labels =fivenum(crime_rate), y=1.25)
```

R code Exa 3.16 Correlation coefficient value

```
# Page No. 107

x<-c(20,23,29,27,30,34,35,37,40,43)
y<-c(1.32,1.67,2.17,2.70,2.75,2.87,3.65,2.86,3.61,4.25)
z<-(x-mean(x))*(y-mean(y))

A<-sum(z)
p<-sum((x-mean(x))*(x-mean(x)))
q<-sum((y-mean(y))*(y-mean(y)))
B<-sqrt(p*q)
cofficient<-A/B
print((cofficient))
```

Chapter 4

Probability and Probability Distributions

R code Exa 4.1 Venn diagram

```
# Page No. 149

p_A<-0.5
p_B<-0.2
p_A_inter_B<-0.05

p_A_Comp<-1 - p_A
print(p_A_Comp)
p_B_Comp<-1 - p_B
print(p_B_Comp)
print(p_A_inter_B)
p_A_un_B<-p_A+p_B-p_A_inter_B
print(p_A_un_B)
```

R code Exa 4.2 Intersection probability

```
# Page No. 151

p_A<-0.6
p_B_by_A<-5/9
p_A_inter_B<-p_A*p_B_by_A
print(p_A_inter_B)
```

R code Exa 4.3 Book club

```
# Page No. 153

p_l=0.50
p_m=0.30
p_h=0.20
p_0_l=0.60
p_0_m=0.15
p_0_h=0.05

p=(p_l*p_0_l)/((p_l*p_0_l)+(p_m*p_0_m)+(p_h*p_0_h))
print(p)
```

R code Exa 4.4 Circuit boards

```
# Page No. 154

p_d1=0.028
p_d2=0.012
p_d3=0.032
p_d4=0.928
p_a4_d1=0.02
p_a4_d2=0.09
p_a4_d3=0.10
p_a4_d4=0.95
```

```
p_nd_or_d1=(p_d1*p_a4_d1)/((p_d1*p_a4_d1)+(p_d2*p_a4
    _d2)+(p_d3*p_a4_d3)+(p_d4*p_a4_d4))
p_nd_or_d2=(p_d2*p_a4_d2)/((p_d1*p_a4_d1)+(p_d2*p_a4
    _d2)+(p_d3*p_a4_d3)+(p_d4*p_a4_d4))
p_nd_or_d3=(p_d3*p_a4_d3)/((p_d1*p_a4_d1)+(p_d2*p_a4
    _d2)+(p_d3*p_a4_d3)+(p_d4*p_a4_d4))
p_nd_or_d4=(p_d4*p_a4_d4)/((p_d1*p_a4_d1)+(p_d2*p_a4
    _d2)+(p_d3*p_a4_d3)+(p_d4*p_a4_d4))

print(p_nd_or_d1)
print(p_nd_or_d2)
print(p_nd_or_d3)
print(p_nd_or_d4)
```

R code Exa 4.7 Probability numerical

```
# Page No. 162

n<-20
z<-0.85

y<-18
p_18sds<-(factorial(n))/(factorial(y)*factorial(n-y)
    )*(z^y)*(1-z)^(n-y)
y<-19
p_19sds<-(factorial(n))/(factorial(y)*factorial(n-y)
    )*(z^y)*(1-z)^(n-y)
y<-20
p_20sds<-(factorial(n))/(factorial(y)*factorial(n-y)
    )*(z^y)*(1-z)^(n-y)
t_p<-p_18sds+p_19sds+p_20sds
print(t_p)
```

R code Exa 4.8 Number of trials

```
# Page No. 162

n<-5
z<-0.9
y<-5

p_15<-(factorial(n))/(factorial(y)*factorial(n-y))*(
   z^y)*(1-z)^(n-y)
print(p_15)
```

R code Exa 4.9 Probability of unemployed

```
# Page No. 163

n<-5
z<-0.9
y<-4

p_1_unemp<-(factorial(n))/(factorial(y)*factorial(n-
   y))*(z^y)*(1-z)^(n-y)
p_few_unemp=((factorial(n))/(factorial(4)*factorial(
   n-4))*(z^4)*(1-z)^(n-4))+((factorial(n))/(
   factorial(5)*factorial(n-5))*(z^5)*(1-z)^(n-5))

print(p_1_unemp)
print(p_few_unemp)
```

R code Exa 4.10 Mean and standard deviation

```
# Page No. 164

n<-20
z<-0.85

m<-n*z
s_d=sqrt(n*z*(1-z))
print(m)
print(s_d)
```

R code Exa 4.11 Economic estimate

```
# Page No. 165

n<-1218
z<-0.5

m<-n*z
print(m)
s_d=sqrt(n*z*(1-z))
print(s_d)

o_v_y=516
o_v_y>3*s_d
o_v_y<m
```

R code Exa 4.12 Mice in trap

```
# Page No. 167

U<-2.3
```

```
y<-4

p_4<-((U^y)*(exp(1)^-U))/factorial(y)
p_most_4=(((U^0)*(exp(1)^-U))/factorial(0))+(((U^1)*
    (exp(1)^-U))/factorial(1))+(((U^2)*(exp(1)^-U))/
    factorial(2))+(((U^3)*(exp(1)^-U))/factorial(3))
    +(((U^4)*(exp(1)^-U))/factorial(4))
p_more_4=1-p_most_4

print(p_4)
print(p_most_4)
print(p_more_4)
```

R code Exa 4.13 Drug effect

```
# Page No. 167

n<-1000
z<-0.001
U<-1
y<-0

m<-n*z
p_se<-((U^y)*(exp(1)^-U))/factorial(y)

print(m)
print(p_se)
```

R code Exa 4.15 Probability in normal distribution

```
# Page No. 174

pnorm(23,mean =20,sd=2)
```

R code Exa 4.16 Probability in normal distribution

```
# Page No. 174

pnorm(16,mean =20,sd=2)
```

R code Exa 4.17 Probability in normal distribution

```
# Page No. 175

pnorm(60,mean =70,sd=13)
pnorm(90,mean =70,sd=13,lower.tail = FALSE)
pnorm(90,mean =70,sd=13)-pnorm(60,mean =70,sd=13)
```

R code Exa 4.18 Percentile of normal distribution

```
# Page No. 177

qnorm(0.10,70,13)
```

R code Exa 4.19 Income tax

```
# Page No. 178

mu=530
sigma=205
```

```
z_75=0.67
y=mu+sigma*z_75
print(y)
```

R code Exa 4.20 Random sample

```
# Page No. 179

ct<-c("c1","c2","c3","c4","c5","c6","c7","c8","c9","c10")
t(combn(ct, 2))
t_p<-nrow(t(combn(ct, 2)))
p_s2c<-1/t_p
print(p_s2c)
```

R code Exa 4.21 Sample function

```
# Page No. 180

st<-c(0:849)
sample(st, 20)
```

R code Exa 4.22 Probability numerical

```
# Page No. 182

pop<-c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11)

combn(pop, 2)
samps<-combn(pop, 2)
```

```
xbars <- colMeans(samps)
table(xbars)
prop.table(table(xbars))
```

R code Exa 4.24 Blood pressure test

```
# Page No. 189

pnorm(150,mean =160,sd=20)
sd <-20/sqrt(5)
pnorm(150,160,sd=8.94)
st_dv=20
n=((-2.326*st_dv)/(150-160))^2
print(n)
```

R code Exa 4.25 Finding probability

```
# Page No. 192

n<-1000
z<-0.5

m<-n*z
s<-sqrt(n*z*(1-z))
pnorm(460,mean=m,sd=s)
```

R code Exa 4.26 License probability

```
# Page No. 194

```

```
n<-100
z<-0.2

m<-n*z
s<-sqrt(n*z*(1-z))
pnorm(14.5, mean=m, sd=s, lower.tail=FALSE)
```

R code Exa 4.27 Normal quantile

```
# Page No. 195

Ch_Rd=c
   (133,137,148,149,152,167,174,179,189,192,201,209,210,211,218,238,
n_q=c
   (-1.96,-1.44,-1.15,-.935,-.755,-.598,-.454,-.319,-.189,-.063,.063

plot(n_q,Ch_Rd)
md=lm(Ch_Rd~n_q)
summary(md)
```

R code Exa 4.28 Correlation coefficient

```
# Page No. 197

y=c
   (133,137,148,149,152,167,174,179,189,192,201,209,210,211,218,238,
x=c
   (-1.868,-1.403,-1.128,-.919,-.744,-.589,-.448,-.315,-.187,-.062,.

```

```
cor(y,x)
```

Chapter 5

Inferences about Population Central Values

R code Exa 5.1 Calculating a Confidence Interval From a Normal Distribution

```
# Calculating a Confidence Interval From a Normal
    Distribution
n<-50
 a<-2.8
 s<-0.6
 # we will use a 95% confidence level and wish to
    find the confidence interval
 margin <- qnorm(0.975)*s/sqrt(n)
 left_i <- a-margin
   right_i <- a+margin
   print("Confidence interval is")
   print(left_i)
   print(right_i)
```

R code Exa 5.2 Calculating a Confidence Interval From a Normal Distribution

```r
#  Calculating a Confidence Interval From a Normal
   Distribution
n<-50
a<-27.3
s<-12.1
#  we will use a 99% confidence level and wish to
   find the confidence interval
margin <- qnorm(0.995)*s/sqrt(n)
left_i <- a-margin
right_i <- a+margin
print("Confidence interval is")
print(left_i)
print(right_i)
```

R code Exa 5.3 cost of textbooks

```r
#  the 95% confidence level would imply the 97.5th
   percentile of the normal distribution at the
   upper tail
zstar <-qnorm(.975)
# standard deviation
 sd <- 125
 #  level of accuracy
 E <- 25
sample_size<- zstar^2 * sd * sd/ E^2
 print(ceiling(sample_size))
 # A sample size of 97 or larger is recommended to
    obtain an estimate of the mean textbook
```

R code Exa 5.4 federal agency

```r
#  the 99% confidence level would imply the 99.5th percentile of the normal distribution at the upper tail
zstar <- qnorm(0.995)
width_interval<-0.50
E<-width_interval/2
sd<-0.75
sample_size<- zstar^2 * sd * sd/ E^2
print(ceiling(sample_size))
# the federal agency must obtain a random sample of 60 cereal cartons to estimate .
```

R code Exa 5.5 Hypothesis Testing

```r
# Hypothesis Testing or  one-tailed test

ybar = 573              # sample mean
 mu0 = 520                # hypothesized value
 sigma = 124             # population standard deviation
 n = 36                   # sample size
 z = (ybar- mu0)/(sigma/sqrt(n))
 print(z) # test statistic

 # We then compute the critical value at .025 significance level.
 #  For alpha= .025, reject the null hypothesis if lies more than 1.96
  alpha = .025
 z.alpha = qnorm(1-alpha)
  print(z.alpha)
```

R code Exa 5.6 Cholesterol levels

```
# two tailed test
ybar = 178.2              # sample mean
mu0 = 190               # hypothesized value
sigma = 45.3            # population standard
   deviation
n = 100                   # sample size
# We compute the critical value at .025
   significance level.
alpha = .05
z.half.alpha = qnorm(1-alpha/2)
# critical values
lr=mu0-(z.half.alpha*sigma)/sqrt(n)
ur=mu0+(z.half.alpha*sigma)/sqrt(n)
paste0(" lower rejection = ",lr)
paste0("upper rejection = ",ur)
z = (ybar- mu0)/(sigma/sqrt(n))
print(z) # test statistic

 print("The test statistic doesnot lies between the
    critical values(i.e. |z|>critical value). Hence,
     at .025 significance level, we reject the null
     hypothesis")
```

R code Exa 5.7 municipal employees

```


ybar = 390            # sample mean
mu0 = 380                # hypothesized value
sigma = 35.2             # population standard
   deviation
n = 50                   # sample size
z = (ybar- mu0)/(sigma/sqrt(n))
print(z) # test statistic

```

```
# We then compute the critical value at .01 significance level.
#  For alpha= .01, reject the null hypothesis if lies more than 2.33
alpha = .01
# critical value
z.alpha = qnorm(1-alpha)
print(z.alpha)
print("the observed value of z < critical value,  so we might be tempted to accept the null hypothesis")
#  but Beta is not computed so  there is insufficient evidence to reject the null hypothesis.
# To reach a conclusion about whether to accept or reject  H0, beta should be calculated.
```

R code Exa 5.8 power for test

```


ybar = 380           # sample mean
mu0 = 395              # hypothesized value
sigma = 35.2            # population standard deviation
n = 50                  # sample size
z = abs((ybar- mu0)/(sigma/sqrt(n)))
 # test statistic
print(z)
# We then compute the critical value at .01 significance level.
alpha = .01
# critical value
z.alpha = qnorm(1-alpha)
print(z.alpha)
```

```
# computing Beta for hypothesized value
Beta_onetailedtest<-pnorm(z.alpha-z)
print(Beta_onetailedtest)
# power for test
powerfortest<-1-Beta_onetailedtest
print(powerfortest)
```

R code Exa 5.9 power for test

```


ybar = 31.2            # sample mean
mu0 = 33              # hypothesized value
sigma = 8.4           # population standard deviation
n = 35                 # sample size
z = (ybar- mu0)/(sigma/sqrt(n))
print(z) # test statistic

# We then compute the critical value at .05
    significance level.
# For alpha = .05, we will reject the null
    hypothesis if z <= -1.645
alpha = .05
z.alpha = qnorm(1-alpha)
#  the observed value of z  is not less than -z.
    alpha, the test statistic does not fall in the
    rejection region.
```

R code Exa 5.10 Suppose that the consumer testing agency thinks

```
ybar=33
sigma=8.4
n=35
```

```
alpha = .05
# critical value
z.alpha = qnorm(1-alpha)
z1=function(mu0){
  z=abs((ybar- mu0)/(sigma/sqrt(n)))
   b=pnorm(z.alpha-z)
   return(b)
}
muo=c(33,32,31,30,29,28,27,26,25)
beta=c(z1(33),z1(32),z1(31),z1(30),z1(29),z1(28),z1
   (27),z1(26),z1(25))
pwr=1-beta
rbind(muo,beta,pwr)
```

R code Exa 5.11 cereal manufacturer produces cereal

```
sd<-.225
z_foralpha=qnorm(1-0.05)
z_forbeta=qnorm(1-0.01)
zstar<-z_foralpha+z_forbeta
E<-16.37 - 16.27
sample_size<- zstar^2 * sd * sd/ E^2
print(ceiling(sample_size))
#  the manufacturer must obtain a random sample of n
    = 80 boxes to conduct this test
```

R code Exa 5.12 research hypothesis validity

```

ybar = 390           # sample mean
mu0 = 380            # hypothesized value
sigma = 35.2        # population standard deviation
n = 50                  # sample size
```

```
z = (ybar- mu0)/(sigma/sqrt(n))
 # test statistic

p_value=1-pnorm(z)
print(p_value)
alpha=0.01
if(p_value>alpha){
  print("we fail to reject H0")
print(" data do not support the research hypothesis.")
 }else{
   print("reject H0")
     }
```

R code Exa 5.13 research hypothesis validity

```

ybar = 31.2            # sample mean
mu0 = 33             # hypothesized value
sigma = 8.4        # population standard deviation
n = 35               # sample size
z = (ybar- mu0)/(sigma/sqrt(n))
# test statistic

p_value=pnorm(z)
print(p_value)
alpha=0.05
if(p_value>alpha){
  print("we fail to reject H0")
  print(" data do not support the research hypothesis(insufficient evidence).")
}else{
  print("reject H0")
}
```

R code Exa 5.14 support the research hypothesis

```

ybar = 178.2            # sample mean
mu0 = 190            # hypothesized value
sigma = 45.3        # population standard deviation
n = 100              # sample size
z = (ybar- mu0)/(sigma/sqrt(n))
print(z)
k=abs(z)
# test statistic
 # formula based on level of significance
p_value=2*(1-pnorm(k))
print(p_value)
# mentioned p value in book is wrong
alpha=0.01
if(p_value>alpha){
  print("we fail to reject H0")
  print(" data do not support the research
     hypothesis(insufficient evidence).")
}else{
  print("  there is very little evidence in the data
      to support the research hypothesis  hence we
     will reject H0")
}
```

R code Exa 5.15 research hypothesis validity

```
y<-c(.593 ,.142, .329, .691 ,.231 ,.793 ,.519 ,.392,
    .418 )
ybar = mean(y)
mu0 = 0.3
```

```
sigma = sd(y)
n = 9                  # sample size
z = abs((ybar- mu0)/(sigma/sqrt(n)))
print(z)
p_value=2*(1-pnorm(z))
print(p_value)
alpha=0.01
if(p_value>alpha){
  print("we fail to reject H0")
  print(" data do not support the research hypothesis(insufficient evidence).")
}else{
  print("reject H0")
}
```

R code Exa 5.17 confidence interval

```
confidence_interval <- function(vector, interval) {
  # Standard deviation of sample
  vec_sd <- sd(vector)
  # Sample size
  n <- length(vector)
  # Mean of sample
  vec_mean <- mean(vector)
  # Error according to t distribution
  error <- qt((interval + 1)/2, df = n - 1) * vec_sd / sqrt(n)
  # Confidence interval as a vector
  ans <- c("lower" = vec_mean - error, "upper" = vec_mean + error)
  return(ans)
}
vector <- c(2.7, 2.4, 1.9, 2.6, 2.4, 1.9, 2.3,
            2.2, 2.5 ,2.3 ,1.8, 2.5, 2.0 ,2.2 )
confidence_interval(vector, 0.95)
```

R code Exa 5.18 confidence interval

```
confidence_interval <- function(vector, interval) {
  # Standard deviation of sample
  vec_sd <- sd(vector)
  # Sample size
  n <- length(vector)
  # Mean of sample
  vec_mean <- mean(vector)
  # Error according to t distribution
  error <- qt((interval + 1)/2, df = n - 1) * vec_sd
     / sqrt(n)
  # Confidence interval as a vector
  ans <- c("lower" = vec_mean - error, "upper" = vec
     _mean + error)
  return(ans)
}
vector <- c( 29, 30, 53, 75, 89, 34, 21, 12, 58, 84,
    92, 117, 115, 119, 109, 115, 134, 253, 289, 287
   )
confidence_interval(vector, 0.95)
```

R code Exa 5.19 CPS personnel

```

ybar = 105.75           # sample mean
mu0 = 75            # hypothesized value
sigma = 82.429      # population standard deviation
n = 20               # sample size
# test statistic
t = abs((ybar- mu0)/(sigma/sqrt(n)))
```

```
print(t)

# formula based on level of significance
m=33
B=1000
p_value= m/B
print(p_value)
alpha=0.05
# our p value< alpha , therfore
print("we conclude that there is sufficient evidence
    that the mean cotanine level exceeds 75 in the
    population of children under CPS supervision")
```

R code Exa 5.20 confidence interval

```
x<-c(14.2, 5.3 ,2.9, 4.2,1.2, 4.3, 1.1, 2.6 ,6.7
    ,7.8 ,25.9, 43.8, 2.7,
      5.6, 7.8 ,3.9, 4.7, 6.5, 29.5 ,2.1 ,34.8 ,3.6
        ,5.8, 4.5, 6.7 )
bootmed = apply(matrix(sample(x, rep=TRUE, 10^4*
    length(x)), nrow=10^4), 1, median)
# The 95% confidence interval for the population
    median is given by
print("Confidence interval is")
quantile(bootmed, c(.025, 0.975))
```

R code Exa 5.21 large scale approximation

```
# large scale approximation
n<-25
alpha<-0.05
z.half.alpha=qnorm(1-alpha/2)
C_alpha2_n=(n/2)-z.half.alpha*sqrt(n/4)
```

```
print(C_alpha2_n)
```

R code Exa 5.23 large sample approximation to the sign test

```
# Large−Sample Approximation
n=25
B=13
# test statstic
Bst=(B-(n/2))/(sqrt(n/4))
print(Bst)
# critical value
alpha=0.05
z.alpha=qnorm(1-alpha/2)
print(z.alpha)
print("e BST is not greater than z.alpha, we fail to
    reject H0")
pvalue=1-pnorm(Bst)
print(pvalue)
```

Chapter 6

Inferences Comparing Two Population Central Values

R code Exa 6.1 confidence interval for indpendent sample

```
# confidence interval for indpendent sample
fresh=c(10.2, 10.6,10.5 ,10.7,10.3, 10.2,10.8,
   10.0,9.8 ,10.6 )
stored=c( 9.8, 9.7,
          9.6, 9.5,
          10.1, 9.6,
          10.2, 9.8,
          10.1 ,9.9)
n1=length(fresh)
n2= length(stored)
y1bar = mean(fresh)
y2bar=mean(stored)
s1=sd(fresh)
s2=sd(stored)
# common standard deviation
sp=sqrt(((n1-1)*s1*s1+(n2-1)*s2*s2)/(n1+n2-2))

# the t-percentile based on df for 95% confidence
   interval
```

```
tstar=qt( .975, df=18)
 margin=tstar*sp*sqrt((1/n1)+(1/n2))
 left_i=(y1bar-y2bar)-margin
 right_i=(y1bar-y2bar)+margin
 print("confidence interval is")
 print(left_i)
 print(right_i)
```

R code Exa 6.2 confidence interval for indpendent sample

```
# confidence interval for indpendent sample

n1= 10
n2= 9
y1bar = 8.27
y2bar=6.78
s1=2.956
s2=2.565
# common standard deviation
sp=sqrt(((n1-1)*s1*s1+(n2-1)*s2*s2)/(n1+n2-2))

# the t-percentile based on df for 95% confidence
   interval
tstar=qt( .975, df=18)
margin=tstar*sp*sqrt((1/n1)+(1/n2))
left_i=(y1bar-y2bar)-margin
right_i=(y1bar-y2bar)+margin
print("confidence interval is")
print(left_i)
print(right_i)
```

R code Exa 6.3 research hypothesis

```
# confidence interval for indpendent sample

n1= 12
n2= 12
y1bar = 26.58
y2bar=39.67
s1=14.36
s2=13.86
# solving part c
# common standard deviation
sp=sqrt(((n1-1)*s1*s1+(n2-1)*s2*s2)/(n1+n2-2))

# the t-percentile based on df for 95% confidence
    interval
tstar=qt( .975, df=18)
margin=tstar*sp*sqrt((1/n1)+(1/n2))
left_i=(y1bar-y2bar)-margin
right_i=(y1bar-y2bar)+margin
print("confidence interval is")
print(left_i)
print(right_i)

# solving part  a and b
t=(y1bar-y2bar)/((sp)*sqrt((1/n1)+(1/n2)))
print(t)
# crtitical value
alpha= 0.05
df=n1+n2-2
t.alpha=qt(0.05, df=22)
if(t<=t.alpha){
  print(" We will reject H0")
}else{
  print("we will fail to reject H0 (no significant
      evidence")
}
```

R code Exa 6.4 research hypothesis

```
n1= 33
n2= 12
y1bar = 25.2
y2bar=33.9
s1=8.6
s2=17.4


t=(y1bar-y2bar)/(sqrt((s1*s1/n1)+(s2*s2/n2)))
print(t)
# To compute the rejection and p-value , we need to
    compute the approximate df
c=((s1*s1)/n1)/(((s1*s1)/n1)+((s2*s2)/n2))
print(c)
df=((n1-1)*(n2-1))/((1-c)^2*(n1-1)+(c*c)*(n2-1))
print(df)
# crtitical value
alpha= 0.05

t.alpha=qt(0.05, df=13)
if(t<=t.alpha){
  print(" We will reject H0")
}else{
  print("we fail to reject H0 (no significant
      evidence")
}
```

R code Exa 6.5 Many states are considering lowering the blood alcohol

```
library(DescTools)
```

```
x = c(0.90, 0.37, 1.63, 0.83, 0.95, 0.78, 0.86, 0.61, 0.38, 1.97)
y = c(1.46, 1.45, 1.76 ,1.44, 1.11 ,3.07 ,0.98 ,1.27 ,2.56 ,1.32)

cbind(c(x,y),rank(c(x,y)))

a <- wilcox.test(x,y,correct=FALSE,conf.int = TRUE)
n1 <- length(x)
a$statistic <- a$statistic + n1*(n1+1)/2
names(a$statistic) <- "T.W"
a
# T<83 so  we reject H0 and conclude there is significant evidence that the placebo population has smaller reaction times than the population of alcohol consumers
# p value calculated in book is wrong
# confidence interval for delta (-1.08, -0.25)

#  95% confidence interval   for the placebo population median
MedianCI(x,conf.level = 0.95,na.rm = FALSE, method = "exact",R = 10000)
# #  95% confidence interval   for the alcohol population median
MedianCI(y,conf.level = 0.95,na.rm = FALSE, method = "exact",R = 10000)
```

R code Exa 6.6 Environmental engineers

```
x=c(11.0,11.2,11.2,11.2,11.4,11.5,11.6,11.7,11.8,11.9,11.9,12.1,10.2
y=rank(x)
cbind(x,y)
```

```
rep(table(y), table(y))
n1=12
n2=12
mut=n1*(n1+n2+1)/2
s=((n1*n2)/12)*((n1+n2+1)-(48/((n1+n2)*(n1+n2-1))))
sigmat=sqrt(s)
T=216 # sum of ranks of before clean up values
Z=(T-mut)/sigmat
Z
# This value exceeds 1.645, so we reject H0 and conclude that the distribution of before-cleanup measurements is shifted to the right of the corresponding distribution of after-cleanup measurements
# part b
si=(n1*n2*(n1+n2+1))/n1
sqrt(si)
```

R code Exa 6.7 Insurance adjusters are concerned about the high

```
garage1=c(17.6,20.2,19.5,11.3,13.0,16.3,15.3,16.2,12.2,14.8,21.3,22.1,16.9
garage2=c(17.3,19.1,18.4,11.5,12.7,15.8,14.9,15.3,12.0,14.2,21.0,21.0,16.1
t.test(garage1,garage2)
```

R code Exa 6.8 perform a paired t test

```
garage1=c(17.6,20.2,19.5,11.3,13.0,16.3,15.3,16.2,12.2,14.8,21.3,22.1,16.9
```

```
garage2=c(17.3,19.1,18.4,11.5,12.7,15.8,14.9,15.3,12.0,14.2,21.0,21.0,16.1
t.test(garage1,garage2,paired = TRUE)
tvalue=qt(1-0.05,14)
# t>tvale we reject H0 and conclude that mean repair estimate for garage I is greater than that for garage II
```

R code Exa 6.9 A city park department compared a new formulation

```
library(DescTools)
brandA=c(211.4,204.4,202.0,201.9,202.4,202.0,202.4,207.1,203.6,216.0,208.
brandB=c(186.3,205.7,184.4,203.6,180.4,202.0,181.5,186.7,205.7,189.1,183.
difference=brandA-brandB

y=rank(replace(abs(difference),abs(difference)==0,NA),na='keep');
cbind(difference,y)
# sum of positive and negative ranks are
Tminus=1+2+3+4+5+6
Tplus= 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 +15 + 16 +17.5 +18 + 19
T=min(Tminus,Tplus)
T
# T<53, we reject H0 and conclude that brand A fertilizer tends to produce more grass than brand B
difference=difference[-6]


```

```
MedianCI(difference,conf.level = 0.95,na.rm = FALSE,
    method = "exact",R = 999)
```

R code Exa 6.10 one sided test

```
# one sided test
sigma=2.4
delta=1.5
z_alpha=qnorm(0.05)
z_beta=qnorm(0.10)
sample_size=2*(sigma^2)*((z_alpha+z_beta)^2)/(delta
    ^2)
print(sample_size)
```

R code Exa 6.11 Sample Size

```

sigma=2.4
delta=1.5
z_alpha=qnorm(0.05)
z_beta=qnorm(0.10)
m=3
# replace 2 with (m+1)/m i.e 4/3
sample_size=(4/3)*(sigma^2)*((z_alpha+z_beta)^2)/(
    delta^2)
print(sample_size)
```

Chapter 7

Inferences about Population Variances

R code Exa 7.1 The 99 confidence interval for mean

```
weights=c(501.4, 498.0, 498.6 ,499.2, 495.2 ,501.4 ,509.5 ,494.9 ,498.6, 497.6,
          505.5 ,505.1 ,499.8 ,502.4, 497.0 ,504.3 ,499.7 ,497.9 ,496.5, 498.9,
          504.9 ,503.2 ,503.0 ,502.6 ,496.8 ,498.2, 500.1 ,497.9 ,502.2, 503.2)
n=length(weights)
ybar=mean(weights)
s=sd(weights)
# The upper-tail chi-square value
XU=qchisq(.995, df=29)
# The lower-tail chi-square value
XL=qchisq(.005, df=29)
# The 99% confidence interval for standard deviation
right_i=sqrt((n-1)*(s^2)/(XL))
left_i=sqrt((n-1)*(s^2)/(XU))
print(left_i)
print(right_i)
# The 99% confidence interval for mean
```

```
margin <- qnorm(0.995)*s/sqrt(n)
left_interval_mean=ybar-margin
right_interval_mean=ybar+margin
print(left_interval_mean)
print(right_interval_mean)
```

R code Exa 7.2 The 95 confidence interval for standard deviation

```
readings=c(203.1, 184.5, 206.8 ,211.0 ,218.3, 174.2, 193.2 ,201.9 ,199.9 ,194.3,
           199.4, 193.6, 194.6 ,187.2 ,197.8 ,184.3, 196.1, 196.4 ,197.5 ,187.9)
n=length(readings)
ybar=mean(readings)
s=sd(readings)
muo=5
# test static
X=(n-1)*(s^2)/(muo^2)
print(X)
#critical value
alpha=0.05
X.alpha=qchisq(1-0.05,df=19)

# the null hypothesis, H0 is rejected if the value of the X is greater than X.alpha
#Since the computed value of the X., 74.61, is greater than the
# critical value 30.14, there is sufficient evidence to reject H0
# The upper-tail chi-square value
XU=qchisq(.975, df=19)
# The lower-tail chi-square value
XL=qchisq(1-.975, df=19)
# The 95% confidence interval for standard deviation
right_i=sqrt((n-1)*(s^2)/(XL))
```

```
left_i=sqrt((n-1)*(s^2)/(XU))
print(left_i)
print(right_i)
```

R code Exa 7.4 the upper percentile for the F distribution

```
#the upper .025 percentile for the F distribution
   with df1 = 10 and df2 =7 is

upper_percentile=qf(1-0.025,10,7)
lower_percentile=1/upper_percentile
print(lower_percentile)
```

R code Exa 7.5 test the equality of the population variances

```

y1bar=38.48
s1=16.37
n1=40
y2bar=26.93
s2=9.88
n2=40
# test statistic
F=s1^2/s2^2
print(F)
#critical value
alpha=0.05
f.alpha=qf(1-alpha/2,39,39)
# we reject H0 if F>=f.alpha
```

R code Exa 7.6 confidence interval

```
#  confidence interval for the ratio of the two
   variances

y1bar=38.48
s1=16.37
n1=40
y2bar=26.93
s2=9.88
n2=40
alpha=0.05
FU=qf(1-alpha/2,39,39)
FL=1/FU

# confidence interval for sigma1^2/sigma2^2
left_i=(s1^2/s2^2)*FL
right_i=(s1^2/s2^2)*FU
print(left_i)
print(right_i)
```

R code Exa 7.7 confidence interval

```

y1bar=20.04
s1=0.474
n1=10
y2bar=9.99
s2=0.233
n2=16
alpha=0.10
FU=qf(1-alpha/2,15,9)
FL=1/FU
# confidence interval for sigma1/sigma2
left_i=sqrt((s1^2/s2^2)*FL)
```

```
right_i=sqrt((s1^2/s2^2)*FU)
print(left_i)
print(right_i)
```

R code Exa 7.8 comparing the variability in power

```
s=c(8.69, 6.89, 80.22)
s_min=min(s)
s_max=max(s)
#test statistic
F=s_max/s_min
print(F)

# critical value
alpha=0.05
df=8
F.alpha=qnorm(alpha/2,8)
print(F.alpha)

# Reject H0 if F >=F.alpha
# conclusion : Thus, we reject H0 and conclude that
  the variances are not all equa
```

R code Exa 7.9 Three different additives that are marketed

```
# install car package by writing install.packages("
  car") command in console
library(car)
y=c
  (4.2,2.9,0.2,25.7,6.3,7.2,2.3,9.9,5.3,6.5,0.2,11.3,0.3,17.1,51.0,
```

```
additive=c
   (1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3)
leveneTest(y,additive)
critical_value=qf(1-0.05,2,27)
critical_value
# L < critical value  we fail to reject and conclude
    that there is insufficient evidence of a
   difference in the population variances of the
   percentage increase in mpg for the three
   additives.
```

Chapter 8

Inferences about More Than Two Population Central Values

R code Exa 8.1 A large body of evidence

```

Group1 <- c(5,17,12,10,4)
Group2 <- c(19,10,9,7,5)
Group3 <- c(25,15,12,9,8)

Combined_Groups <- data.frame(cbind(Group1, Group2,
    Group3)) # combines the data into a single data
    set.
Combined_Groups # shows spreadsheet like results
#summary(Combined_Groups)  # min, median, mean, max

Stacked_Groups <- stack(Combined_Groups)
Stacked_Groups #shows the table Stacked_Groups

Anova_Results <- aov(values ~ ind, data = Stacked_
    Groups)
summary(Anova_Results) # shows Anova_Results


```

```
# answer given in book is wrong because sample
    varaince calcaulated for group 1 column in book
    is 33.7 which is wrong
# correct sample varaince is 28.3
```

R code Exa 8.2 clinical psychologist

```

Group1 <- c(96,79,91,85,83,91,82,87)
Group2 <- c(77,76,74,73,78,71,80)
Group3 <- c(66,73,69,66,77,73,71,70,74)


cols <- list(m=Group1, y=Group2,z=Group3)
as.data.frame(lapply(cols, `length<-`, max(sapply(
    cols, length))))
cols
Stacked_Groups <- stack(cols)
Stacked_Groups #shows the table Stacked_Groups

Anova_Results <- aov(values ~ ind, data = Stacked_
    Groups)
summary(Anova_Results) # shows Anova_Results
```

R code Exa 8.3 clerics knowledge of mental illness

```
M=c(62,60,60,25,24,23,20,13,12,6)
C=c(62,62,24,24,22,20,19,10,8,8)
P=c(37,31,15,15,14,14,14,5,3,2)

Group1=abs(M-median(M))
Group2=abs(C-median(C))
Group3=abs(P-median(P))
```

```
Combined_Groups <- data.frame(cbind(Group1, Group2, Group3)) # combines the data into a single data set.
Combined_Groups # shows spreadsheet like results
#summary(Combined_Groups) # min, median, mean, max

Stacked_Groups <- stack(Combined_Groups)
Stacked_Groups #shows the table Stacked_Groups

Anova_Results <- aov(values ~ ind, data = Stacked_Groups)
summary(Anova_Results) # shows Anova_Results
```

R code Exa 8.4 dissolved oxygen contents at four distances from mouth

```
Mean=c(2.2, 4.6 ,21.2 ,31.4)
Standard_deviation=c(1.476,2.119,4.733,5.52)
s_min=min(Standard_deviation)
s_max=max(Standard_deviation)
# test statistic
F=s_max^2/s_min^2
print(F)
# The critical value of F > F.alpha
#  we reject the hypothesis of homogeneity
#of the population variances.

distance_1km=c(1,5,2,1,2,2,4,3,0,2)
distance_5km=c(4,8,2,3,8,5,6,4,3,3)
distance_10km=c(20,26,24,11,28,20,19,19,21,24)
distance_20km=c(37,30,26,24,41,25,36,31,31,33)
print(Standard_deviation[1]^2/Mean[1])
print(Standard_deviation[2]^2/Mean[2])
print(Standard_deviation[3]^2/Mean[3])
print(Standard_deviation[4]^2/Mean[4])
```

```
i=1
while(i<11){
  distance_1km[i]=sqrt(distance_1km[i]+0.375)
  i=i+1
}
i=1
while(i<11){
  distance_5km[i]=sqrt(distance_5km[i]+0.375)
  i=i+1
}
i=1
while(i<11){
  distance_10km[i]=sqrt(distance_10km[i]+0.375)
  i=i+1
}
i=1
while(i<11){
  distance_20km[i]=sqrt(distance_20km[i]+0.375)
  i=i+1
}
combined_group=data.frame(cbind(distance_1km,
   distance_5km,distance_10km,distance_20km))
combined_group
```

R code Exa 8.7 rank sum test

```
#  the rank sum test
Methodist=c(62,60,60,25,24,23,20,13,12,6)
Catholic=c( 62,62,24,24,22,20,19,10,8,8)
Pentecostal=c(37,31,15,15,14,14,14,5,3,2 )
n=30
data.value <- c( Methodist,Catholic,Pentecostal)

data.rank <- rank(data.value)
data <- data.frame(data.value, data.rank)
```

```
print(data)
 Sumofranks=c(182.5,167.5,115)
 #test statistic
 H=(12/(n*(n+1)))*((Sumofranks[1]^2+Sumofranks[2]^2+
    Sumofranks[3]^2)/10)-3*(n+1)
 print(H)
```

Chapter 9

Multiple Comparisons

R code Exa 9.1 contrasts orthogonal

```
sample_size=c(5,4,6,5)
#l1=y1bar-y3bar
#l2=y2bar-y4bar
# We can rewrite the contrasts in the following form :
#l1=y1bar+0*y2bar-y3bar+0*y4bar
#l2=0*y1bar+y2bar+0*y3bar-y4bar
# thus we identify a1 = 1, a2 = 0, a3 =-1, a4 = 0 and b1 =0, b2 = 1, b3 =0, b4 =-1
a=c(1,0,-1,0)
b=c(0,1,0,-1)
test=0
i=1
while(i<=length(sample_size)){
  test=test+(a[i]*b[1])/sample_size[i]
  i=i+1
}
print(test)
if(test==0){
  print("hence the contrasts are orthogonal.")
}else{
```

```
  print("hence the contrasts are not orthogonal")
}
```

R code Exa 9.2 contrasts orthogonal

```
sample_size=c(5,4,6,5)
#l1=y1bar-y3bar
#l2=y1bar+y2bar+y3bar-3*y4bar
# We can rewrite the contrasts in the following form
    :
#l1=y1bar+0*y2bar-y3bar+0*y4bar
#l2=l2=y1bar+y2bar+y3bar-3*y4bar
# thus we identify a1 = 1, a2 = 0, a3 =-1, a4 = 0
    and b1 =0, b2 = 1, b3 =0, b4 =-1
a=c(1,0,-1,0)
b=c(1,1,1,-3)
test=0
i=1
while(i<=length(sample_size)){
  test=test+(a[i]*b[1])/sample_size[i]
  i=i+1
}
print(test)
if(test==0){
  print("hence the contrasts are orthogonal.")
}else{
  print("hence the contrasts are not orthogonal")
}
# part b
sample_size=5
a=c(1,0,-1,0)
b=c(1,1,1,-3)
test=0
i=1
while(i<=4){
```

```
  test=test+(a[i]*b[1])/sample_size
  i=i+1
}
print(test)
if(test==0){
  print("hence the contrasts are orthogonal.")
}else{
  print("hence the contrasts are not orthogonal")
}
```

R code Exa 9.3 control weeds in crops

```
sample_size=c(6,6,6,6,6)

# l=y1bar-(y2bar+y3bar+y4bar+y5bar)/4

# thus we identify a1 = 4, a2 = -1, a3 =-1, a4 = -1
  , a5=-1
a=c(4,-1,-1,-1,-1)

test=0
i=1
while(i<=length(sample_size)){
  test=test+a[i]^2
  i=i+1
}
print(test)
y1bar=1.175
y2bar=1.293
y3bar=1.328
y4bar=1.415
y5bar=1.500
l=4*y1bar-y2bar-y3bar-y4bar-y5bar
print(l)
# we can obtain the sum of squares associated with
```

```
   the contrast from
SSC1=(sample_size[1]*(l^2))/test
print(SSC1)

a=c(0,1,1,-1,-1)
test=0
i=1
while(i<=length(sample_size)){
  test=test+a[i]^2
  i=i+1
}
l=0*y1bar+y2bar+y3bar-y4bar-y5bar
# we can obtain the sum of squares associated with
   the contrast from
SSC2=(sample_size[2]*(l^2))/test
print(SSC2)

a=c(0,1,-1,0,0)
test=0
i=1
while(i<=length(sample_size)){
  test=test+a[i]^2
  i=i+1
}
l=0*y1bar+y2bar-y3bar+0*y4bar+0*y5bar
# we can obtain the sum of squares associated with
   the contrast from
SSC3=(sample_size[2]*(l^2))/test
print(SSC3)

a=c(0,0,0,1,-1)
test=0
i=1
while(i<=length(sample_size)){
  test=test+a[i]^2
  i=i+1
}
l=0*y1bar+0*y2bar+0*y3bar+y4bar-y5bar
```

```
# we can obtain the sum of squares associated with the contrast from
SSC4=(sample_size[2]*(l^2))/test
print(SSC4)
```

R code Exa 9.5 Test each of the four contrasts for significance

```
si=c(.1204,.1269,.1196,.1249,.1265)
# data fom example 9.3
Fmax=max(si)^2/min(si)^2
print(Fmax)
# four test statistic
SSC1=.2097 # these value are computed in 9.3
SSC2=.1297
SSC3=.0037
SSC4=.0217
MSError=.0153
F1=SSC1/MSError
F2=SSC2/MSError
F3=SSC3/MSError
F4=SSC4/MSError

print(F1)
print(F2)
print(F3)
print(F4)

alpha=0.05
df1=1
df2=25
F_0.05_1_25=qf(1-alpha,df1,df2)
print(F_0.05_1_25)

# we conclude that contrasts l1 and l2 were significantly
```

```
#different from zero but contrasts l3 and l4 were not significantly different from zero.
```

R code Exa 9.6 control the experimentwise error rate

```
alpha=0.05
m=4 #comparisons
alpha_l=alpha/m
F_aplha_l_1_25=qf(1-alpha_l,df1 = 1,df2 = 25)
print(F_aplha_l_1_25)
# We would then reject H0 if SSCi/MSError >=F_alpha_l_1_25
F1 = 13.71 # computed in 9.5
F2 = 8.48
F3 = 0.24
F4 = 1.42
#  we would declare contrast l1 and l2 significantly  different from 0 because their F ratios are greater than 7.24.
```

R code Exa 9.7 five different weed agents

```
F=5.96 # computed from 9.3
alpha=0.05
MSError=.0153
F_value=qf(1-alpha,df1=4,df2 = 25)
# as the F >F_value
print(" we reject H0 and conclude that at least one of the population means differs from the rest")
t.alpha=qt(1-alpha/2,df=25)
LSD=t.alpha*(sqrt((2*MSError)/6))
print(LSD)
```

R code Exa 9.9 confidence interval for mean

```
y1bar=1.175
y2bar=1.293
y3bar=1.328
y4bar=1.415
y5bar=1.500

alpha=0.05
tstar=qt(1-alpha/2,df=25)
MSError=0.0153
LSD=tstar*sqrt((2*MSError)/6)
print(LSD)

#  the 95% confidence interval for y3bar-y1bar
left_i=(y3bar-y1bar)-LSD
right_i=(y3bar-y1bar)+LSD
print("Confidence interval is")
print(left_i)
print(right_i)
```

R code Exa 9.11 confidence interval

```
y1bar=1.175
y2bar=1.293
y3bar=1.328
y4bar=1.415
y5bar=1.500

alpha=0.05
q.alpha=4.158
MSError=0.0153
```

```
LSD=q.alpha*sqrt((MSError)/6)
print(LSD)

#  the 95% confidence interval for y3bar−y1bar
left_i=(y3bar-y1bar)-LSD
right_i=(y3bar-y1bar)+LSD
print("Confidence interval is")
print(left_i)
print(right_i)
```

R code Exa 9.13 Compare the two biological treatments

```
# install package by  writing install.packages("
    DunnettTests") command in console
# install package by writing install.packages(
    mvtnorm") command in console
library(DunnettTests)
library(mvtnorm)
alpha=0.05
k=4
v=25
n=6

 cvSUDT(k=4,alpha=0.05,alternative="U",df = 25,corr
    = .5)
 # max value of critical value is taken
critical_value=2.28 # approx
sw2=0.0153
# test statistic
D=critical_value*sqrt((2*sw2)/n)
print(D)
# conclusion
yi=c(1.293,1.328,1.415,1.500)
yc=1.175
i=1
```

```
while (i<5) {
  if((yi[i]-yc)<D){
    print("Not greater than control")
  }
  else{
    print("greater than control")
  }
  i=i+1
}
```

R code Exa 9.14 Scheffs procedure to determine

```
sample_size=6
sw2=0.0153
y1bar=1.175
y2bar=1.293
y3bar=1.328
y4bar=1.415
y5bar=1.500
t=5
alpha=0.05
F_value=qf(1-alpha,t-1,25)

a=c(4,-1,-1,-1,-1)# for control vs agents
# test statistic
vl=sw2*((a[1]^2+a[2]^2+a[3]^2+a[4]^2+a[5]^2)/sample_
   size)
print(vl)
S=sqrt(vl)*sqrt((t-1)*F_value)
print(S)
# critcal value
l=abs(4*y1bar-y2bar-y3bar-y4bar-y5bar)
print(l)
if(S<l){
  print("  contrasts are significantly different
```

```
from zero")

}else{
  print(" contrasts are not significantly different from zero")
}

a=c(0,1,1,-1,-1)# Biological vs. chemical
# test statistic
vl=sw2*((a[1]^2+a[2]^2+a[3]^2+a[4]^2+a[5]^2)/sample_size)
print(vl)
S=sqrt(vl)*sqrt((t-1)*F_value)
print(S)
# critcal value
l=abs(0*y1bar+y2bar+y3bar-y4bar-y5bar)
print(l)
if(S<l){
  print("  contrasts are significantly different from zero")

}else{
  print(" contrasts are not significantly different from zero")
}

a=c(0,1,-1,0,0)# Bio1 vs. Bio2
# test statistic
vl=sw2*((a[1]^2+a[2]^2+a[3]^2+a[4]^2+a[5]^2)/sample_size)
print(vl)
S=sqrt(vl)*sqrt((t-1)*F_value)
print(S)
# critcal value
l=abs(0*y1bar+y2bar-y3bar+0*y4bar+0*y5bar)
print(l)
if(S<l){
  print("  contrasts are significantly different
```

```
from zero")

}else{
  print(" contrasts are not significantly different from zero")
}

a=c(0,0,0,1,-1)# Chm1 vs. Chm2
# test statistic
vl=sw2*((a[1]^2+a[2]^2+a[3]^2+a[4]^2+a[5]^2)/sample_size)
print(vl)
S=sqrt(vl)*sqrt((t-1)*F_value)
print(S)
# critcal value
l=abs(0*y1bar+0*y2bar+0*y3bar+y4bar-y5bar)
print(l)
if(S<l){
  print("  contrasts are significantly different from zero")

}else{
  print(" contrasts are not significantly different from zero")
}
```

Chapter 10

Categorical Data

R code Exa 10.1 Estimate the proportion of all patients with the specified type of cancer

```
# y denote the number of successes in the n sample trials ,
# sample proportion
y=330
n=870
pie=y/n
sigma=sqrt((pie*(1-pie))/n)

alpha=0.05
z.alpha=qnorm(1-alpha)
error=z.alpha*sigma
#  the 90% confidence interval on the proportion of cancer
#patients who will survive at least 5 years
left_i=pie-error
right_i=pie+error
print(left_i)
print(right_i)
```

R code Exa 10.2 water department

```
# y denote the number of successes in the n sample
    trials ,
# sample proportion
y=43
n=50
pie=y/n
sigma=sqrt((pie*(1-pie))/n)
alpha=0.025
z.alpha=qnorm(1-alpha)
error=z.alpha*sigma
#  95% confidence interval
left_i=pie-error
right_i=pie+error
print(" Wald  95 % confidence interval")
print(left_i)
print(right_i)

# Using the WAC confidence interval , we need to
    compute:
ybar = y + 0.5*(z.alpha^2)
nbar = n + (z.alpha^2)
pie_bar=ybar/nbar


sigma_bar=sqrt((pie_bar*(1-pie_bar))/nbar)
error_bar=z.alpha*sigma_bar
left=pie_bar-error_bar
right=pie_bar+error_bar
print(" WAC 95% confidence interval")
print(left)
print(right)
```

R code Exa 10.3 confidence interval for pie would be

```
y=50
n=50
alpha=.05
# If we used the standard estimator of pie
pie=y/n
#  The point estimator would be given by
pie_adj=(n+(3/8))/(n+(3/4))
print(pie_adj)
# A 95% confidence interval for pie would be
left_i=(alpha/2)^(1/n)
right_i=1
print(left_i)
print(right_i)
```

R code Exa 10.4 designer of the new operating system

```
alpha=0.025
pie=0.5
E=0.03
z.alpha=qnorm(1-alpha)
#  sample size necessary to achieve this accuracy
n=((z.alpha^2)*pie*(1-pie))/E^2
print(ceiling(n))
# 1,068 programs would need to be tested in order to
    be 95% confident that
#the estimate of pie is within .03 of the actual
   value of pie

pie=0.8
n=((z.alpha^2)*pie*(1-pie))/E^2
```

```
print(n)
#  if the designer was fairly certain that the
   actual value of pie was at least .80,
#then the required sample size can be greatly
   reduced.
```

R code Exa 10.5 percentage of binge drinkers at the university

```
y=1200
n=2500
pie=y/n
sigma=sqrt((pie*(1-pie))/n)
pie0=0.44
#test statistic
z=(pie-pie0)/sigma
print(z)
# critical value
alpha=0.05
z.alpha=qnorm(1-alpha)
#Because the observed value of z exceeds the
   critical value 1.645, we conclude that the
#percentage of students that participate in binge
   drinking exceeds the national percentage of 44%
nbar = n + (z.alpha^2)
pie_bar=(y+z.alpha)/nbar

sigma_bar=sqrt((pie_bar*(1-pie_bar))/nbar)
error_bar=z.alpha*sigma_bar
left=pie_bar-error_bar
right=pie_bar+error_bar
print(left)
print(right)
# the percentage of binge drinkers at the university
   is, with 95% confidence, between 46% and 50%
```

R code Exa 10.6 confidence interval

```
pie1=413/527
pie2=392/608
# The sample awareness proportion is higher in
   Wichita, so let's make Wichita region 1.
#The estimated standard error is
sigma=sqrt(((pie1*(1-pie1))/527)+((pie2*(1-pie2))/
   608))
print(sigma)
alpha=0.025
z.alpha=qnorm(1-alpha)
error=z.alpha*sigma
#  95% confidence interval
left_i=(pie1-pie2)-error
right_i=(pie1-pie2)+error
print(left_i)
print(right_i)
```

R code Exa 10.7 confidence interval

```
pie1=94/125
pie2=113/175
# The sample awareness proportion is higher in
   Wichita, so let's make Wichita region 1.
#The estimated standard error is
sigma=sqrt(((pie1*(1-pie1))/125)+((pie2*(1-pie2))/
   175))

#test statistic
z=(pie1-pie2)/sigma
print(z)
```

```
alpha=0.05
z.alpha=qnorm(1-alpha)
zstar=qnorm(1-alpha/2)
# Since z   is greater than z.alpha, we reject H0
    and conclude that the observations
#support the hypothesis
error=zstar*sigma
#  95% confidence interval
left_i=(pie1-pie2)-error
right_i=(pie1-pie2)+error
print(left_i)
print(right_i)
```

R code Exa 10.8 clinical trial is conducted to compare two drug therapies

```

data =  rbind(c(38,4), c(14,7) )

print(data)
# fisher test
fisher.test(data, alternative="greater")
alpha=0.025
# as pvalue > alpha then  we conclude that there is
    not
#significant evidence that the proportion of
    patients obtaining a successful outcome
#is higher for drug PV than for drug P.
```

R code Exa 10.9 Previous experience with the breeding of a particular herd of cattle

```
p_1calf=0.83
p_2calf=0.02
```

```
p_0calf=0.15

# event A=  dams gives birth to no healthy progeny ,1 healthy progeny ,2 healthy progeny(n1=1,n2=1,n3=1)

P_eventA=(factorial(3)*(p_0calf^1*p_1calf^1*p_2calf^1))/(factorial(1)*factorial(1)*factorial(1))
print(P_eventA)
 # event B= 3 dams give birth to 1 healthy progeny(n1=0,n2=3,n3=0)
P_eventB=(factorial(3)*(p_0calf^0*p_1calf^3*p_2calf^0))/(factorial(0)*factorial(3)*factorial(0))
print(P_eventB)
#  the probability of obtaining exactly three healthy progeny from three dams
p=P_eventA+P_eventB
print(p)
```

R code Exa 10.10 null hypothesis

```
obsereved_cell_number = c(120,60,10,10)
expected_cell_number=c(100,50,20,30)
# test statistic
Xsquare = 0
i=1
while(i<=length(obsereved_cell_number)){
  Xsquare=Xsquare+(((obsereved_cell_number[i]-expected_cell_number[i])^2)/expected_cell_number[i])
  i=i+1
}
print(Xsquare)
# critical value
alpha = 0.05
```

```
X.alpha=qchisq(1-alpha,df=3)
# The computed value of xsquare is greater than x.
    alpha, so we reject the null hypothes
```

R code Exa 10.11 Environmental engineers

```
yi=c(0,1,2,3,4,5,6,7)
ni=c(6,23,29,31,27,13,8,13)
n=sum(ni)
ybar=sum(yi*ni)/sum(ni)
print(ybar)
# ybar value in book is calculated wrong
# The Poisson probabilities
Pyi=c(dpois(0,ybar),dpois(1,ybar),dpois(2,ybar),
    dpois(3,ybar),dpois(4,ybar),dpois(5,ybar),dpois
    (6,ybar),dpois(7,ybar))
print(Pyi)
# Expected cell count
Ei=n*Pyi
print(Ei)

# test statistic
i=1
X2=0
while(i<9){
  X2=X2+(((ni[i]-Ei[i])^2)/Ei[i])
  i=i+1
}
print(X2) # ans in book is calculate wrong
df=6

pvalue=pchisq(X2,df,lower.tail = FALSE)
print(pvalue)
# as p-value <=.01 model is   Poisson model provides
     an Unacceptable fit to data
```

R code Exa 10.12 random sample of 216 patients

```
moderate=c(15,32,18,5)
mildly=c(8,29,23,18)
severe=c(1,20,25,22)
all_ages=c(sum(moderate),sum(mildly),sum(severe))
all_servetiles=c(24,81,66,45)
grand_total=216
# For row 1,the estimated expected number of occurrences
E11=(sum(moderate)*all_servetiles[1])/grand_total
print(E11)
E12=(sum(moderate)*all_servetiles[2])/grand_total
print(E12)
E13=(sum(moderate)*all_servetiles[3])/grand_total
print(E13)
E14=(sum(moderate)*all_servetiles[4])/grand_total
print(E14)

# For row 2,the estimated expected number of occurrences
E21=(sum(mildly)*all_servetiles[1])/grand_total
print(E21)
E22=(sum(mildly)*all_servetiles[2])/grand_total
print(E22)
E23=(sum(mildly)*all_servetiles[3])/grand_total
print(E23)
E24=(sum(mildly)*all_servetiles[4])/grand_total
print(E24)

# For row 3,the estimated expected number of occurrences
E31=(sum(severe)*all_servetiles[1])/grand_total
print(E31)
```

```
E32=(sum(severe)*all_servetiles[2])/grand_total
print(E32)
E33=(sum(severe)*all_servetiles[3])/grand_total
print(E33)
E34=(sum(severe)*all_servetiles[4])/grand_total
print(E34)

dt=data.frame(cbind(E11,E12,E13,E14),cbind(E21,E22,E23,E24),cbind(E31,E32,E33,E34))
dt
```

R code Exa 10.13 test to determine the severity of the disease

```
n = c(15,32,18,5,8,29,23,18,1,20,25,22)
E=c(7.78,26.25,21.39,14.58,8.67,29.25,23.83,16.25,7.56,25.50,20.78,14
# test statistic
Xsquare = 0
i=1
while(i<=length( n)){
  Xsquare=Xsquare+(((n[i]-E[i])^2)/E[i])
  i=i+1
}
print(Xsquare)
# critical value
alpha = 0.05
X.alpha=qchisq(1-alpha,df=6)
# The computed value of xsquare is greater than x.alpha, so we reject the null hypothes
```

R code Exa 10.14 test of homogeneity of distributions

```
n = c(50,59,161,88,20,40,56,52,188,4,3,5,2,66,6)
E=c(67.5,67.5,135,37,37,74,74,74,148,3,3,6,18.5,18.5,37)
# test statistic
Xsquare = 0
i=1
while(i<=length( n)){
  Xsquare=Xsquare+(((n[i]-E[i])^2)/E[i])
  i=i+1
}
print(Xsquare)
# critical value
alpha = 0.001
X.alpha=qchisq(1-alpha,df=8)
# The computed value of xsquare is greater than x.alpha, so we reject the null hypothes
```

R code Exa 10.15 Consider both a population in which

```
 #  odds of a randomly chosen person carrying HIV
p=.001/.009
#  occurrence of a positive test result causes the odds to change to
p_hiv=.001
p_positive_hiv=.95
p_nothiv=.999
p_positive_nothiv=.02
P_occurencepositive=(p_hiv*p_positive_hiv)/(p_nothiv*p_positive_nothiv)
print(P_occurencepositive)

# The odds of carrying HIV do go up given a positive test result, from about
#.001 (to 1) to about .0475 (to 1).
```

R code Exa 10.16 level of stress

```
low=c(250,750)
high=c(400,1600)
odds_ratio=(250/750)/(400/1600)
print(odds_ratio)
# We will next compute a 95% confidence interval for
    the odds ratio
a= log(odds_ratio)
sigma=sqrt(1/low[1]+1/low[2]+1/high[1]+1/high[2])
print(sigma)
# The 95% confidence interval for the odds ratio is
   obtained by first computing
error=1.96*sigma
left_i=a-error
right_i=a+error

print(left_i)
print(right_i)
print("confidence interval")
print(exp(left_i))
print(exp(right_i))
```

R code Exa 10.17 The pharmaceutical study

```
 # for clinic 1
r1=c(50,50)
c1=c(55,45)

# for clinic 2
r2=c(50,50)
```

```r
c2=c(55,45)

# for clinic 3
r3=c(50,50)
c3=c(74,26)
nh=100
# The numerator of the CMH statistic
N=((40-((r1[1]*c1[1])/nh))+(35-((r2[1]*c2[1])/nh))+(43-((r3[1]*c3[1])/nh)))^2
print(N)

D=((r1[1]*r1[2]*c1[1]*c1[2])/(nh^2*(nh-1)))+((r2[1]*r2[2]*c2[1]*c2[2])/(nh^2*(nh-1)))+((r3[1]*r3[2]*c3[1]*c3[2])/(nh^2*(nh-1)))
print(D)

X2=N/D
print(X2)
# For df = 1, this result is significant at the p < .001 level. the drug-treated groups have consistently higher improvement rates than the placebo groups.
```

Chapter 11

Linear Regression and Correlation

R code Exa 11.2 least squares estimates of slope and intercept

```
# Sales Volume
y=c(25,55,50,75,110,138,90,60,10,100)
# % of Ingredients Purchased Directly ,
x=c(10,18,25,40,50,63,42,30,5,55)
# Sxy is the sum of x deviations times y deviations
    and Sxx is the sum of x deviations squared.

Sxx=(x-mean(x))^2
Sxy=(x-mean(x))*(y-mean(y))

# least-squares estimates of slope and intercept
Beta1=sum(Sxy)/sum(Sxx)
Beta0=mean(y)-Beta1*mean(x)
 print(Beta1)
 print(Beta0)
```

R code Exa 11.3 the least squares estimates of the intercept

```
xbar=31.80
ybar=2.785
Sxx=485.60
Syy=7.36
Sxy=55.810
Beta1=Sxy/Sxx
Beta0=ybar-Beta1*xbar
print(Beta1)
print(Beta0)
```

R code Exa 11.4 Forest growth retardation

```
soil_ph <- c
    (3.3,3.4,3.4,3.5,3.6,3.6,3.7,3.7,3.8,3.8,3.9,4.0,4.1,4.2,4.3,4.4,
grow_ret <- c
    (17.78,21.59,23.84,15.13,23.45,20.87,17.78,20.09,17.78,12.46,14.9

# Apply the lm() function.
relation <- lm(grow_ret~soil_ph)

print(summary(relation))
anova(relation)
# The regression equation  : GrowthRet = 47.475 -
    7.86 SoilpH
 # The estimated intercept (constant)
beta0=47.475
# estimated slope (Soil pH)
beta1=-7.859
# least square prediction
y=47.475-7.859*4
print(y)
```

R code Exa 11.7 confidence interval for the slope

```
beta1=-7.859 # caclulated in 11.4
error=1.090
alpha=0.05/2
df=18
t.alpha=qt(1-alpha,df)
print(t.alpha)
# corresponding confidence interval for the true
    value of beta1
left_i=beta1-t.alpha*error
right_i=beta1+t.alpha*error
print("confidence interval")
print(left_i)
print(right_i)
```

R code Exa 11.8 use the F test for testing

```
F_statistic=52.01 # computed in example 11.4
F_statistic
t_statistic=-7.212
# both are calculated in 11.4
p=t_statistic^2
# t^2 equals the F value, to within round-off error
    .
p
```

R code Exa 11.10 The manufacturer of a new brand of thermal panes

```
heat_loss <- c(86,80,77 , 78,84,75 ,33,38,43)
temperature <- c(20,20,20,40,40,40,60,60,60)
plot(temperature,heat_loss)
# Apply the lm() function.
relation <- lm(heat_loss~temperature)

print(summary(relation))
 # linear regression model : y= 109-1.07*x
anova(relation)
#  y and y-ycap for the nine observations
cbind( temperature,heat_loss,predict(relation),resid
    (relation))
plot(predict(relation),resid(relation))
```

R code Exa 11.11 Conduct a test for lack of fit

```
heat_loss <- c(86,80,77 , 78,84,75 ,33,38,43)
temperature <- c(81,81,81,79,79,79,38,38,38)

# Apply the lm() function.
relation <- lm(heat_loss~temperature)

aov(relation)

SSPexp=134
SSresidual=894.5 #calculated in 11.10
SSlack=SSresidual-SSPexp

MSPexp=SSPexp/6
MSlack=SSlack/1
# test statistic
F=MSlack/MSPexp
print(F)
```

```
alpha=0.05

fvalue=qf(1-alpha,df1 = 1,df2 = 6)
if(F>fvalue){
  print(" we reject H0 and conclude that there is
     significant lack of fit for a linear regression
      model")
}else{
  print("we  do not reject H0 and conclude that
     there is no significant lack of fit for a
     linear regression model")
}
```

R code Exa 11.12 An engineer is interested in calibrating a flow meter

```

x <- c(1,2,3,4,5,6,7,8,9,10)
y <- c(1.4,2.3,3.1,4.2,5.1,5.8,6.8,7.6,8.7,9.5)
xbar=mean(x)
# Apply the lm() function.
relation <- lm(y~x)
print(summary(relation))

# linear regression model : y=.9012*x+.4934
 mod=lm(x~y)
 predict(mod,data.frame(y=4),interval = "prediction"
    ,level = 0.95)
 #  the 95% prediction limits for x are 3.65 to 4.13
```

R code Exa 11.13 In a study of the reproductive success of grasshoppers

```
eggs=c
  (27,32,39,48,59,67,71,65,73,67,78,72,81,74,83,75,84,77,83,76,82,7
weight=c
  (2.1,2.3,2.4,2.5,2.9,3.1,3.2,3.3,3.4,3.4,3.5,3.5,3.5,3.6,3.6,3.6,
 plot(weight,eggs)
 # correlation coefficient
 cor(eggs,weight)
```

R code Exa 11.14 compute SSTotal

```
# SS(Total) = Syy
SStotal=6066.1667
rxy=0.606
SSregression = (rxy^2)*SStotal
print(SSregression)

SSresidual=SStotal-SSregression
print(SSresidual)
```

R code Exa 11.15 The personnel director of a small company

```

y <- c(41, 39, 47, 51, 43, 40, 57, 46, 50, 59, 61,
  52)
x<- c( 24 ,30 ,33 ,35 ,36 ,36 ,37 ,37 ,38 ,40 ,43
  ,49)

# Apply the lm() function.
relation <- lm(y~x)

print(summary(relation))
```

```
cor(y,x)
# For all 12 observations, the output shows a correlation coefficient of .646
# after subseting x>=37

y1 <- c(  57, 46, 50, 59, 61, 52)
x1<- c(  37 ,37 ,38 ,40 ,43 ,49)
# Apply the lm() function.
relation <- lm(y1~x1)
print(summary(relation))
 cor(y1,x1)
# For  subset having x greater than or equal to 37, the output shows a correlation coefficient of .188
```

R code Exa 11.16 Perform t tests for the null hypothesis

```
y <- c(41, 39, 47, 51, 43, 40, 57, 46, 50, 59, 61, 52)

x<- c( 24 ,30 ,33 ,35 ,36 ,36 ,37 ,37 ,38 ,40 ,43 ,49)
cor.test(y,x)
```

Chapter 12

Multiple Regression and the General Linear Model

R code Exa 12.2 An industrial engineer is designing a simulation model to generate

```
# model : y = b0 + b1x1 + b2x2 + b3x3 +e
#x1 = 1 if system 2 is used , x1 = 0 otherwise
#x2 = 1 if system 3 is used , x2 = 0 otherwise
#x3 = 1 if system 4 is used , x3 = 0 otherwise
u1=7
u2=9
u3=6
u4=15
b0=u1
b1=u2-u1
b2=u3-u1
b3=u4-u1
print(b0)
print(b1)
print(b2)
print(b3)
```

R code Exa 12.3 Check your findings by substituting values

```r
u1=7
u2=9
u3=6
u4=15
x=(u3-u1)-(u2-u1)
y=(u3-u1)-(u4-u1)
print(x)
print(y)
```

R code Exa 12.5 An experiment was conducted to investigate

```r
y=c(4.3,5.5,6.8,8.0,4.0,5.2,6.6,7.5,2.0,4.0,5.7,6.5)
x1=c(4,5,6,7,4,5,6,7,4,5,6,7)
x2=c(.20,.20,.20,.20,.30,.30,.30,.30,.40,.40,.40,.40)

cbind(sum(y),sum(x1),sum(x2),sum(y*x1),sum(y*x2),sum(x1*x2),sum(x1^2),sum(x2^2))
#  three normal equations for this model
# 66.1=12*beta0+66*beta1+3.6*beta2
# 383.3=66*beta0+378*beta1+19.8*beta2
#  19.19=3.6*beta0+19.8*beta1+1.16*beta2
relation = lm(y~x1+x2)
print(summary(relation))
anova(relation)
# linear regression model : y=0.667+1.316*x1-8.000*x2
x1=6.5
x2=.35
y=0.667+1.316*x1-8.000*x2
```

```
print(y)
```

R code Exa 12.6 A kinesiologist is investigating measures of the physical fitness

```
y=c(1.5 ,2.1, 1.8, 2.2, 2.2, 2.0, 2.1, 1.9, 2.8,
   1.9, 2.0, 2.7, 2.4, 2.3, 2.0, 1.7, 2.3, 0.9, 1.2,
    1.9, 0.8, 2.2, 2.3, 1.7,  1.6, 1.6, 2.8, 2.7,
   1.3 ,2.1, 2.5, 1.5, 2.4, 2.3, 1.9, 1.5,  2.4,
   2.3, 1.7 ,2.0 ,1.9, 2.3, 2.1, 2.2, 1.8, 2.1, 2.2,
    1.3, 2.5, 2.2, 1.4, 2.2 ,2.5 ,1.8)
x1=c(139.8, 143.3, 154.2 ,176.6 ,154.3, 185.4,
   177.9, 158.8, 159.8, 123.9, 164.2, 146.3, 172.6
   ,147.5 ,163.0, 159.8 ,162.7, 133.3 ,142.8 ,146.6
   ,141.6, 158.9, 151.9, 153.3 ,144.6, 133.3, 153.6,
    158.6, 108.4, 157.4 ,141.7, 151.1, 149.5, 144.3
   ,166.6 ,153.6  ,144.1, 148.7, 159.9, 162.8 ,145.7
    ,156.7, 162.3, 164.7, 134.4, 160.1, 143.0,
   141.6,  152.0, 187.1, 122.9, 157.1 ,155.1 ,133.6)
x2=c(19.1, 21.1, 21.2, 23.2, 22.4 ,22.1, 21.6, 19.0
   ,20.9 ,22.0 ,19.5 ,19.8  ,20.7 ,21.0 ,21.2 ,20.4
   ,20.0 ,21.1 ,22.6 ,23.0 ,22.1, 22.8 ,21.8 ,20.0
   ,22.9 ,22.9 ,19.4 ,21.0 ,21.1 ,20.1 ,19.8 ,21.8
   ,20.5 ,21.0, 21.4 ,20.8 ,20.3 ,19.1 ,19.6 ,21.3
   ,20.0, 19.2 ,22.1 ,19.1, 20.9 ,21.1 ,20.5 ,21.7
   ,20.8 ,21.5, 22.6 ,23.4 ,20.8, 22.5)
x3=c(18.1, 15.3, 15.3, 17.7, 17.1, 16.4, 17.3, 16.8,
    15.5 ,13.8, 17.0, 13.8, 16.8, 15.3, 14.2, 16.8,
   16.6, 17.5, 18.0 ,15.7 ,19.1, 13.4, 13.6, 16.1,
   15.8 ,18.2 ,13.3, 14.9, 16.7 ,15.7, 13.5, 18.8
   ,14.9 ,17.2 ,17.4, 16.4, 13.3, 15.4, 17.4, 16.2,
   18.6, 16.4, 19.0, 17.1, 15.6, 14.2, 17.1, 14.5
   ,17.3, 14.6, 18.6, 14.2, 16.0, 15.4)
x4=c(133.6 ,144.6, 164.6, 139.4 ,127.3, 137.3, 144.0
    ,141.4, 127.7 ,124.2 ,135.7 ,116.1 ,109.0,
```

```
    131.0, 143.3 ,156.6 ,120.1, 131.8, 149.4 ,106.9,
    135.6 ,164.6 ,162.6 ,134.8  ,154.0 ,120.7 ,151.9,
     133.6 ,142.8 ,168.2 ,120.5, 135.6, 119.5 ,119.0
    ,150.8, 144.0 , 124.7 ,154.4 ,136.7 ,152.4
    ,133.6, 113.2 ,81.6 ,134.8, 130.4 ,162.1, 144.7
    ,163.1, 137.1 ,156.0 ,127.2 ,121.4 ,155.3, 140.4)

relation = lm(y~x1+x2+x3+x4)
print(summary(relation))
```

R code Exa 12.8 The following SPSS computer output is obtained from the data

```
y=c(1.5 ,2.1, 1.8, 2.2, 2.2, 2.0, 2.1, 1.9, 2.8,
    1.9, 2.0, 2.7, 2.4, 2.3, 2.0, 1.7, 2.3, 0.9, 1.2,
     1.9, 0.8, 2.2, 2.3, 1.7,  1.6, 1.6, 2.8, 2.7,
    1.3 ,2.1, 2.5, 1.5, 2.4, 2.3, 1.9, 1.5,  2.4,
    2.3, 1.7 ,2.0 ,1.9, 2.3, 2.1, 2.2, 1.8, 2.1, 2.2,
     1.3, 2.5, 2.2, 1.4, 2.2 ,2.5 ,1.8)
x1=c(139.8, 143.3, 154.2 ,176.6 ,154.3, 185.4,
    177.9, 158.8, 159.8, 123.9, 164.2, 146.3, 172.6
    ,147.5 ,163.0, 159.8 ,162.7, 133.3 ,142.8 ,146.6
    ,141.6, 158.9, 151.9, 153.3 ,144.6, 133.3, 153.6,
     158.6, 108.4, 157.4 ,141.7, 151.1, 149.5, 144.3
    ,166.6 ,153.6  ,144.1, 148.7, 159.9, 162.8 ,145.7
     ,156.7, 162.3, 164.7, 134.4, 160.1, 143.0,
    141.6,  152.0, 187.1, 122.9, 157.1 ,155.1 ,133.6)
x2=c(19.1, 21.1, 21.2, 23.2, 22.4 ,22.1, 21.6, 19.0
    ,20.9 ,22.0 ,19.5 ,19.8  ,20.7 ,21.0 ,21.2 ,20.4
    ,20.0 ,21.1 ,22.6 ,23.0 ,22.1, 22.8 ,21.8 ,20.0
    ,22.9 ,22.9 ,19.4 ,21.0 ,21.1 ,20.1 ,19.8 ,21.8
    ,20.5 ,21.0, 21.4 ,20.8 ,20.3 ,19.1 ,19.6 ,21.3
    ,20.0, 19.2 ,22.1 ,19.1, 20.9 ,21.1 ,20.5 ,21.7
    ,20.8 ,21.5, 22.6 ,23.4 ,20.8, 22.5)
x3=c(18.1, 15.3, 15.3, 17.7, 17.1, 16.4, 17.3, 16.8,
```

```
    15.5 ,13.8, 17.0, 13.8, 16.8, 15.3, 14.2, 16.8,
   16.6, 17.5, 18.0 ,15.7 ,19.1, 13.4, 13.6, 16.1,
   15.8 ,18.2 ,13.3, 14.9, 16.7 ,15.7, 13.5, 18.8
   ,14.9 ,17.2 ,17.4, 16.4, 13.3, 15.4, 17.4, 16.2,
   18.6, 16.4, 19.0, 17.1, 15.6, 14.2, 17.1, 14.5
   ,17.3, 14.6, 18.6, 14.2, 16.0, 15.4)
x4=c(133.6 ,144.6, 164.6, 139.4 ,127.3, 137.3, 144.0
    ,141.4, 127.7 ,124.2 ,135.7 ,116.1 ,109.0,
   131.0, 143.3 ,156.6 ,120.1, 131.8, 149.4 ,106.9,
   135.6 ,164.6 ,162.6 ,134.8  ,154.0 ,120.7 ,151.9,
    133.6 ,142.8 ,168.2 ,120.5, 135.6, 119.5 ,119.0
   ,150.8, 144.0 , 124.7 ,154.4 ,136.7 ,152.4
   ,133.6, 113.2 ,81.6 ,134.8, 130.4 ,162.1, 144.7
   ,163.1, 137.1 ,156.0 ,127.2 ,121.4 ,155.3, 140.4)

relation = lm(y~x1+x2+x3+x4)

anova(relation)
SSresidual=4.3938
df=49 # by looking at table given in question
# Std. Error of the Estimate
SE=sqrt(SSresidual/df)
print(SE)
```

R code Exa 12.9 Using the sum of squares in the ANOVA table

```
y=c(1.5 ,2.1, 1.8, 2.2, 2.2, 2.0, 2.1, 1.9, 2.8,
   1.9, 2.0, 2.7, 2.4, 2.3, 2.0, 1.7, 2.3, 0.9, 1.2,
    1.9, 0.8, 2.2, 2.3, 1.7,  1.6, 1.6, 2.8, 2.7,
   1.3 ,2.1, 2.5, 1.5, 2.4, 2.3, 1.9, 1.5,  2.4,
   2.3, 1.7 ,2.0 ,1.9, 2.3, 2.1, 2.2, 1.8, 2.1, 2.2,
    1.3, 2.5, 2.2, 1.4, 2.2 ,2.5 ,1.8)
x1=c(139.8, 143.3, 154.2 ,176.6 ,154.3, 185.4,
   177.9, 158.8, 159.8, 123.9, 164.2, 146.3, 172.6
   ,147.5 ,163.0, 159.8 ,162.7, 133.3 ,142.8 ,146.6
```

```
,141.6, 158.9, 151.9, 153.3 ,144.6, 133.3, 153.6,
 158.6, 108.4, 157.4 ,141.7, 151.1, 149.5, 144.3
,166.6 ,153.6  ,144.1, 148.7, 159.9, 162.8 ,145.7
 ,156.7, 162.3, 164.7, 134.4, 160.1, 143.0,
141.6,  152.0, 187.1, 122.9, 157.1 ,155.1 ,133.6)
x2=c(19.1, 21.1, 21.2, 23.2, 22.4 ,22.1, 21.6, 19.0
,20.9 ,22.0 ,19.5 ,19.8  ,20.7 ,21.0 ,21.2 ,20.4
,20.0 ,21.1 ,22.6 ,23.0 ,22.1, 22.8 ,21.8 ,20.0
,22.9 ,22.9 ,19.4 ,21.0 ,21.1 ,20.1 ,19.8 ,21.8
,20.5 ,21.0, 21.4 ,20.8 ,20.3 ,19.1 ,19.6 ,21.3
,20.0, 19.2 ,22.1 ,19.1, 20.9 ,21.1 ,20.5 ,21.7
,20.8 ,21.5, 22.6 ,23.4 ,20.8, 22.5)
x3=c(18.1, 15.3, 15.3, 17.7, 17.1, 16.4, 17.3, 16.8,
 15.5 ,13.8, 17.0, 13.8, 16.8, 15.3, 14.2, 16.8,
16.6, 17.5, 18.0 ,15.7 ,19.1, 13.4, 13.6, 16.1,
15.8 ,18.2 ,13.3, 14.9, 16.7 ,15.7, 13.5, 18.8
,14.9 ,17.2 ,17.4, 16.4, 13.3, 15.4, 17.4, 16.2,
18.6, 16.4, 19.0, 17.1, 15.6, 14.2, 17.1, 14.5
,17.3, 14.6, 18.6, 14.2, 16.0, 15.4)
x4=c(133.6 ,144.6, 164.6, 139.4 ,127.3, 137.3, 144.0
 ,141.4, 127.7 ,124.2 ,135.7 ,116.1 ,109.0,
131.0, 143.3 ,156.6 ,120.1, 131.8, 149.4 ,106.9,
135.6 ,164.6 ,162.6 ,134.8  ,154.0 ,120.7 ,151.9,
 133.6 ,142.8 ,168.2 ,120.5, 135.6, 119.5 ,119.0
,150.8, 144.0 , 124.7 ,154.4 ,136.7 ,152.4
,133.6, 113.2 ,81.6 ,134.8, 130.4 ,162.1, 144.7
,163.1, 137.1 ,156.0 ,127.2 ,121.4 ,155.3, 140.4)

relation = lm(y~x1+x2+x3+x4)

anova(relation)

# cofficient of determination
summary(relation)$r.squared
```

R code Exa 12.11 The following SAS output is provided for fitting the model

```
y=c(1.5 ,2.1, 1.8, 2.2, 2.2, 2.0, 2.1, 1.9, 2.8,
   1.9, 2.0, 2.7, 2.4, 2.3, 2.0, 1.7, 2.3, 0.9, 1.2,
    1.9, 0.8, 2.2, 2.3, 1.7,  1.6, 1.6, 2.8, 2.7,
   1.3 ,2.1, 2.5, 1.5, 2.4, 2.3, 1.9, 1.5,  2.4,
   2.3, 1.7 ,2.0 ,1.9, 2.3, 2.1, 2.2, 1.8, 2.1, 2.2,
    1.3, 2.5, 2.2, 1.4, 2.2 ,2.5 ,1.8)
x1=c(139.8, 143.3, 154.2 ,176.6 ,154.3, 185.4,
   177.9, 158.8, 159.8, 123.9, 164.2, 146.3, 172.6
   ,147.5 ,163.0, 159.8 ,162.7, 133.3 ,142.8 ,146.6
   ,141.6, 158.9, 151.9, 153.3 ,144.6, 133.3, 153.6,
    158.6, 108.4, 157.4 ,141.7, 151.1, 149.5, 144.3
   ,166.6 ,153.6  ,144.1, 148.7, 159.9, 162.8 ,145.7
    ,156.7, 162.3, 164.7, 134.4, 160.1, 143.0,
   141.6,  152.0, 187.1, 122.9, 157.1 ,155.1 ,133.6)
x2=c(19.1, 21.1, 21.2, 23.2, 22.4 ,22.1, 21.6, 19.0
   ,20.9 ,22.0 ,19.5 ,19.8  ,20.7 ,21.0 ,21.2 ,20.4
   ,20.0 ,21.1 ,22.6 ,23.0 ,22.1, 22.8 ,21.8 ,20.0
   ,22.9 ,22.9 ,19.4 ,21.0 ,21.1 ,20.1 ,19.8 ,21.8
   ,20.5 ,21.0, 21.4 ,20.8 ,20.3 ,19.1 ,19.6 ,21.3
   ,20.0, 19.2 ,22.1 ,19.1, 20.9 ,21.1 ,20.5 ,21.7
   ,20.8 ,21.5, 22.6 ,23.4 ,20.8, 22.5)
x3=c(18.1, 15.3, 15.3, 17.7, 17.1, 16.4, 17.3, 16.8,
    15.5 ,13.8, 17.0, 13.8, 16.8, 15.3, 14.2, 16.8,
   16.6, 17.5, 18.0 ,15.7 ,19.1, 13.4, 13.6, 16.1,
   15.8 ,18.2 ,13.3, 14.9, 16.7 ,15.7, 13.5, 18.8
   ,14.9 ,17.2 ,17.4, 16.4, 13.3, 15.4, 17.4, 16.2,
   18.6, 16.4, 19.0, 17.1, 15.6, 14.2, 17.1, 14.5
   ,17.3, 14.6, 18.6, 14.2, 16.0, 15.4)
x4=c(133.6 ,144.6, 164.6, 139.4 ,127.3, 137.3, 144.0
    ,141.4, 127.7 ,124.2 ,135.7 ,116.1 ,109.0,
   131.0, 143.3 ,156.6 ,120.1, 131.8, 149.4 ,106.9,
   135.6 ,164.6 ,162.6 ,134.8  ,154.0 ,120.7 ,151.9,
    133.6 ,142.8 ,168.2 ,120.5, 135.6, 119.5 ,119.0
   ,150.8, 144.0 , 124.7 ,154.4 ,136.7 ,152.4
   ,133.6, 113.2 ,81.6 ,134.8, 130.4 ,162.1, 144.7
```

```
,163.1, 137.1 ,156.0 ,127.2 ,121.4 ,155.3, 140.4)

relation = lm(y~x1+x2+x3+x4)
print(summary(relation))
anova(relation)
SSregression=1.8028+0.6973+2.4205+1.1856
print(SSregression)
df=4
MSregression=SSregression/df
print(MSregression)
MSresidual=0.08967

# test statistic
F=MSregression/MSresidual
print(F)
fvalue=qf(1-0.01,4,49)
# F>fvalue, therefore there is strong evidence in
  the data to reject the null hypothesis to reject
  the null hypothesis
```

R code Exa 12.12 A large city bank studies the relation of average account size

```
# linear regression model : y= 0.15085 - 0.00288x1 -
   0.00759x2 + 0.26528x3
SStotal=3.328
SSresidual=0.674
n=21
R2=(SStotal-SSresidual)/SStotal
print(R2)

#test statistic
F=(R2/3)/((1-R2)/(n-4))
print(F)
alpha=0.05
```

```
fvalue=qf(1-alpha,df1=3,df2 = 17)
print(fvalue)
# Because the computed F statistic , , is greater
   than fvalue , we reject H0 and
#conclude that one or more of the x values has some
   predictive power
```

R code Exa 12.13 confidence interval

```
beta1=.2652
se=.1012
n=21
k=3
df=n-(k+1)
alpha=0.025
tvalue=qt(1-alpha,df)
print(tvalue)
left_i=beta1-tvalue*se
right_i=beta1+tvalue*se
print(" 95% confidence interval")
print(left_i)
print(right_i)
```

R code Exa 12.14 Locate the estimated partial slope

```
beta1=.01291
se=.00283
n=54
k=4
df=n-(k+1)
alpha=0.05
tvalue=qt(1-alpha,df)
print(tvalue)
```

```
left_i=beta1-tvalue*se
right_i=beta1+tvalue*se
print(" 95% confidence interval")
print(left_i)
print(right_i)
```

R code Exa 12.15 conclusion of the test compatible with the confidence interval

```
beta1=.01291
se=.00283
# test statistic
t=beta1/se
print(t)
n=54
k=4
df=n-(k+1)
alpha=0.05
tvalue=qt(1-alpha,df)
 #the computed value of the test statistic > tvalue
    . we conclude there is significant evidence to
    reject H0
```

R code Exa 12.16 Locate the t statistic

```
# test statistic
t=.26528/.10127
print(t)
df=17
t1value=qt(1-0.01,df)
t2value=qt(1-0.005,df)
print(t1value)
print(t2value)
```

```
  #  Thus, H0 would be rejected at the alpha= .01 level but not at the alpha=  .005 level
pvalue =pt(-t, df)
print(pvalue)
```

R code Exa 12.17 A state fisheries commission wants to estimate the number of bass

```
catch=c(3.6,.8,2.5,2.9,1.4,.9,3.2,2.7,2.2,5.9,3.3,2.9,3.6,2.4,.9,2.0,1.9
residence=c(92.2,86.7,80.2,87.2,64.9,90.1,60.7,50.9,86.1,90.0,80.4,75.0,70.0
size=c(.21,.30,.31,.40,.44,.56,.78,1.21,.34,.40,.52,.66,.78,.91,1.10,1.
access=c(0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1)
structures=c(81,26,52,64,40,22,80,60,30,90,74,50,61,40,22,50,37,61,39,53)
relation = lm(catch~residence+size+access+structures)
print(summary(relation))
anova(relation)

# for reduced model
print(" for reduced model")
relation = lm(catch~residence+size)
print(summary(relation))
anova(relation)
# complete linear regression model :  -2.78 + .0268 x1 + .504x2 + .743x3 + .0511x4
# reduced model : -.87 + .0394x1 + .828x2
# test statistic
```

```
SSregression_complete=24.0624
SSregression_reduced=2.913
Ssresidual_complete=2.2756
n=20
a=(SSregression_complete-SSregression_reduced)/2
b=(Ssresidual_complete)/(n-5)
F=a/b
print(F)
fvalue=qf(1-.01,2,15)
print(fvalue)
#  The value of the test statistic is much larger
   than the tabled value, so we have conclusive
   evidence that the access and structure variables
   add predictive value
```

R code Exa 12.22 logistic regression

```

CK <- c(20, 60, 100, 140, 180, 220, 260, 300, 340,
   380, 420, 460, 500)
y <- cbind( c(2, 13, 30, 30, 21, 19, 18, 13, 19, 15,
    7, 8, 35) , c(88, 26, 8, 5, 0, 1, 1, 1, 0, 0, 0,
    0, 0))
model1=glm(y~CK, family = binomial)
summary(model1)
cbind(CK,PRED=predict(model1,type = "response"))
 #  probability that a patient had a heart attack
    when the CK level in the patient was 140 is .868
```

R code Exa 12.26 matrix operations

```
C <- matrix(c(-2,4,6,2,4,1),2,3)
D <- matrix(c(3,9,7,2,5,1,4,-2,8),3,3)
E <- matrix(c(4,8,1,-1,6,-6,0,4,7),3,3)
# C+D cannot be computed because of different
    dimensions.

x=D+E
y=D-E
z=C%*%D
p=t(E)
x
y
z
p
```

R code Exa 12.27 inverse of matrix

```
B <- matrix(c(7,9,3,5),2,2)
C <- matrix(c(3,2,3,2,8,1,4,-2,8),3,3)
determinant_B=det(B)
determinant_C=det(C)
determinant_B
determinant_C
solve(B)
solve(C)
# B* inverse B
x=zapsmall(solve(B)%*%B)
x
# C* inverse C
y=zapsmall(solve(C)%*%C)
y
```

R code Exa 12.28 Obtain the least squares estimates for the prediction equation

```
Y <- matrix(c(25,19,33,23),4,1)
X <- matrix(c(1,1,1,1,-2,-2,2,2,5,-5,5,-5),4,3)
transpose_X=t(X)
transpose_X_X=transpose_X%*%X
transpose_X_Y=transpose_X%*%Y
inverse_transpose_X_X=solve(transpose_X_X)

beta=inverse_transpose_X_X%*%transpose_X_Y
beta
# prediction equation is
y=25.0+1.5*x1+0.8*x2
```

R code Exa 12.29 Compute SSResidual for the data

```
Y <- matrix(c(25,19,33,23),4,1)
X <- matrix(c(1,1,1,1,-2,-2,2,2,5,-5,5,-5),4,3)
transpose_X=t(X)
transpose_Y=t(Y)
transpose_X_X=transpose_X%*%X
transpose_X_Y=transpose_X%*%Y
inverse_transpose_X_X=solve(transpose_X_X)

beta=inverse_transpose_X_X%*%transpose_X_Y
transpose_Y_Y=transpose_Y%*%Y
transpose_Y_Y
SSresidual=transpose_Y_Y-t(beta)%*%transpose_X_Y
SSresidual
```

R code Exa 12.30 Calculate SSRegression and SSTotal for the data

```
Y <- matrix(c(25,19,33,23),4,1)
X <- matrix(c(1,1,1,1,-2,-2,2,2,5,-5,5,-5),4,3)
transpose_X=t(X)
transpose_Y=t(Y)
transpose_X_X=transpose_X%*%X
transpose_X_Y=transpose_X%*%Y
inverse_transpose_X_X=solve(transpose_X_X)

beta=inverse_transpose_X_X%*%transpose_X_Y
transpose_Y_Y=transpose_Y%*%Y
SSresidual=transpose_Y_Y-t(beta)%*%transpose_X_Y
SSregression=t(beta)%*%transpose_X_Y-(sum(Y)^2/4)
SSregression
SStotal=SSregression+SSresidual
SStotal
```

R code Exa 12.31 Calculate the estimated standard error

```
 ## prediction equation is
 #y=25.0+1.5*x1+0.8*x2
Y <- matrix(c(25,19,33,23),4,1)
X <- matrix(c(1,1,1,1,-2,-2,2,2,5,-5,5,-5),4,3)
transpose_X=t(X)
transpose_X_X=transpose_X%*%X
inverse_transpose_X_X=solve(transpose_X_X)
inverse_transpose_X_X
# for  estimated standard error use inverse_
   trasnpose_X_X matrix
s_beta0=2*sqrt(0.25)
s_beta0
s_beta1=2*sqrt(0.0625)
s_beta2=2*sqrt(0.01)
s_beta1
s_beta2
```

Chapter 13

Further Regression Topics

Chapter 14

Analysis of Variance for Completely Randomized Designs

R code Exa 14.1 assignment of paints to the highway sections

```
#  we obtain a random permutation of the numbers 1
   to 16.
x=sample(1:16)
# We thus obtain the assignment of paints to the
   highway sections
values=c("P1","P1","P1","P1","P2","P2","P2","P2","P3
   ","P3","P3","P3","P4","P4","P4","P4")
cbind(x,values)
```

R code Exa 14.2 sum of squares for error

```
paint_data=c(28, 35, 27, 21,21, 36, 25 ,18,26, 38,
   27, 17,16, 25, 22, 18)
ybar=sum(paint_data)/length(paint_data)
```

```
print(ybar)
# total sum of squares
TSS=0
i=1
while(i<=length(paint_data)){
  TSS=TSS+(paint_data[i]-ybar)^2
  i=i+1
}
print(TSS)
# between treatment sum of squares

yi=c(mean(paint_data[1:4]),mean(paint_data[5:8]),
   mean(paint_data[9:12]),mean(paint_data[13:16]))

SST=0
j=1
while(j<=length(paint_data)/4){
  SST=SST+4*((yi[j]-ybar)^2)
  j=j+1
}
print(SST)
#  sum of squares for error
SSE=TSS-SST
print(SSE)
```

R code Exa 14.11 confidence interval

```
y1bar=25.1
y2bar=23.5
```

```
y3bar=37.8

MSE=10.278
sigma=sqrt(MSE)

# crtical value
alpha = 0.025
z.alpha=qt(1-alpha,df=15)

# For panels 2 and 3, we have nt = 10 observations
   per panel, thus confidence interval will be
n=10
error=sigma*z.alpha*sqrt(2/n)
#  thus confidence interval will be
left_i=(y3bar-y2bar)-error
right_i=(y3bar-y2bar)+error
print("confidence interval is")
print(left_i)
print(right_i)
```

R code Exa 14.12 locate significant differences among display panels

```
alpha=0.05
q.alpha=qtukey(1-alpha,3,15)
n=10
sw2=10.28 # calculated in 14.10
W=q.alpha*(sqrt(sw2/n))
W
sample_means=c(25.1,23.5,37.8)
# by ordering sample mean from lowest to highest ,
   we rank display panels by 2 1 3
# if diffrence between means > W then we declare
   them to be significantly different from each
   other
```

R code Exa 14.13 number of replications is

```
sigma=(70-40)/4

alpha=0.025
z.alpha=qnorm(1-alpha)

E=4 #given
#  number of replications is
n=((sigma^2)*(z.alpha^2))/E^2
print(n)
```

Chapter 15

Analysis of Variance for Blocked Designs

R code Exa 15.7 Assess whether taking into account the two extraneous sources of variation

```
MSR=11128.14
MSC=544.44
t=5
MSE=2887.29
# relative efficiency of this Latin square design
    relative to a completely randomized design is
re=(MSR+MSC+(t-1)*MSE)/((t+1)*MSE)
re
```

R code Exa 15.11 determine which pairs of treatments have significantly different means

```
t=9
v=16
r=3
sw2=2.847
qvalue=qtukey(1-0.05,t,v)
W=qvalue*sqrt(sw2/r)
W
 # any pair of treatment means having a difference
    between corresponding
#sample means exceeding 4.9 would be declared
   significantly different
```