R Textbook Companion for Business Statistics For Contemporary Decision Making by Ken Black[1]

Created by
Vibha Mehta
B.Tech.
Computer Science and Engineering
Techno India NJR Institute of Technology
Cross-Checked by
R TBC Team

May 29, 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: Business Statistics For Contemporary Decision Making

Author: Ken Black

Publisher: Wiley, USA

Edition: 6

Year: 2010

ISBN: 978-0470-40901-5

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

Contents

List of R Codes

Chapter 2

Charts and Graphs

R code Exa 2.1.a Class Midpoints

```
# Class midpoints.

Interest_rate <- c
   (7.29,7.23,7.11,6.78,7.47,6.69,6.77,6.57,6.80,6.88,6.98,7.16,
                        7.30,7.24,7.16,7.03,6.90,7.16,7.40,7.05,7.28,7.31
                        7.03,7.17,6.78,7.08,7.12,7.31,7.40,6.35,6.96,7.29
                        6.96,7.02,7.13,6.84)

summary(Interest_rate)

low_val<- 6.30
high_val <-7.70
step_val <- 0.20
x_breaks <- seq(low_val,high_val,step_val)
x_breaks
x_mid <- seq(low_val+step_val/2,high_val-step_val/2,
   step_val)
x_mid
```

```
x<-cut(Interest_rate,breaks = x_breaks,right=FALSE)
x
y<-table(x)
y

df <- data.frame(y)
df

# Class Mid point :
df$midpoint <- x_mid
View(df)
```

R code Exa 2.1.b Relative Frequency

```
# Relative Frequency.

Interest_rate <- c(7.29,7.23,7.11,6.78,7.47,6.69,6.77,6.57,6.80,6.88,6.98,7.16,
                    7.30,7.24,7.16,7.03,6.90,7.16,7.40,7.05,7.28,7.31
                    7.03,7.17,6.78,7.08,7.12,7.31,7.40,6.35,6.96,7.29
                    6.96,7.02,7.13,6.84)

summary(Interest_rate)

low_val<- 6.30
high_val <-7.70
step_val <- 0.20
x_breaks <- seq(low_val,high_val,step_val)
x_breaks
x_mid <- seq(low_val+step_val/2,high_val-step_val/2,
  step_val)
x_mid
```

```
x<-cut(Interest_rate,breaks = x_breaks,right=FALSE)
x
y<-table(x)
y

df <- data.frame(y)
df

# Class Mid point :
df$midpoint <- x_mid
df

# Relative Frequency :
rf <- df$Freq/sum(df$Freq)
rf
df$relative_frequency <- rf
View(df)
```

R code Exa 2.1.c Cumulative Frequency

```
# Cumulative Frequency.

Interest_rate <- c(7.29,7.23,7.11,6.78,7.47,6.69,6.77,6.57,6.80,6.88,6.98,7.16,
                    7.30,7.24,7.16,7.03,6.90,7.16,7.40,7.05,7.28,7.31
                    7.03,7.17,6.78,7.08,7.12,7.31,7.40,6.35,6.96,7.29
                    6.96,7.02,7.13,6.84)

summary(Interest_rate)

low_val<- 6.30
high_val <-7.70
```

```
step_val <- 0.20
x_breaks <- seq(low_val,high_val,step_val)
x_breaks
x_mid <- seq(low_val+step_val/2,high_val-step_val/2,
    step_val)
x_mid
x<-cut(Interest_rate,breaks = x_breaks,right=FALSE)
x
y<-table(x)
y

df <- data.frame(y)
df

# Class Mid point :
df$midpoint <- x_mid
df

# Relative Frequency :
rf <- df$Freq/sum(df$Freq)
rf
df$relative_frequency <- rf
View(df)

# Cumulative Frequency :
c<-cumsum(df$Freq)
df$cumulative_frequency <- c
n <- sum(df$Freq)
crf <- c/n
df$cumul <- crf
df$pie <- round(360*rf,1)
View(df)
```

R code Exa 2.2 Steam and leaf plot

```r
# Steam-and-leaf plot

costs <- c(3.67,2.75,9.15,5.11,3.32,2.09,1.83,10.94,1.93,3.89,
           7.20,2.78,6.72,7.80,5.47,4.15,3.55,3.53,3.34,4.95,
           5.42,8.64,4.84,4.10,5.10,6.45,4.65,1.97,2.84,3.21)


stem(costs, scale = 1, width = 80, atom = 1e-08)
```

R code Exa 2.3.a Bar Graph

```r
# Bar Graph :

Inventary_shrinkage <- c("Employee theft","Shoplifting","Administartive error","Vendor fraud")

Annual_amount <- c(17918.6, 15191.9,7617.6,2553.6)

data <- data.frame(Inventary_shrinkage,Annual_amount)

Proportion <- data$Annual_amount/sum(data$Annual_amount)

Percent <- Proportion*100

data <- cbind(data,Proportion,Percent)

Degree <- data$Proportion*360

```

```
data<-cbind(data,Degree)

library(ggplot2)

ggplot(data,aes(x=data$Inventary_shrinkage,y=data$Annual_amount))+geom_bar(stat = "identity")
```

R code Exa 2.3.b Bar Graph

```
# Pie Chart :

Inventary_shrinkage <- c("Employee theft","Shoplifting","Administartive error","Vendor fraud")

Annual_amount <- c(17918.6, 15191.9,7617.6,2553.6)

data <- data.frame(Inventary_shrinkage,Annual_amount)

Proportion <- data$Annual_amount/sum(data$Annual_amount)

Percent <- Proportion*100

data <- cbind(data,Proportion,Percent)

Degree <- data$Proportion*360

data<-cbind(data,Degree)

labls <- paste(data$Inventary_shrinkage,data$Percent,sep = " ")

labls <- paste(labls,"%",sep="")
```

```
pie(data$Percent, labels = labls)
```

R code Exa 2.4 Scatter Plot

```
# Scatter Plot :
Residential <- c(169635,155113,149410,175822,162706,134605,195028,231396,234955,
                 266481,267063,263385,252745,228943,197526,232134,24
                 251937,281229,280748,297886,315757)

Non_residential <- c(96497,115372,96407,129275,140569,145054,131289,155261,178925,
                     163740,160363,164191,169173,167896,135389,12092
                     139711,153866,166754,177639,175048)

home <- cbind(Residential,Non_residential)
View(home)

# Scatter plot :
plot(Residential, Non_residential,xlab="Residential",ylab="Non-Residential")
```

Chapter 3

Descriptive Statistics

R code Exa 3.1.a Mode

```r
# Mode Example :

getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

Company <- c("Enterprise","Hertz","Natioanl/Alamo","Avis", "Dollar", "Budget","Advantage",
             "U-save","Payless","ACE","Fox","Rent-A-Wreck","Traingle")

Number_of_Cars_in_Service <- c(643000,327000,233000,204000,167000,144000,20000,12000,10000,
                               9000,9000,7000,6000)

data1 <- data.frame(Company,Number_of_Cars_in_Service)

sort_data  <- data1[order(-Number_of_Cars_in_Service
```

```
),]

result <- getmode(sort_data$Number_of_Cars_in_Service)
print(result)
```

R code Exa 3.1.b Median

```
# Median :

Company <- c("Enterprise","Hertz","Natioanl/Alamo","Avis", "Dollar", "Budget","Advantage",
              "U-save","Payless","ACE","Fox","Rent-A-Wreck","Traingle")

Number_of_Cars_in_Service <- c(643000,327000,233000,204000,167000,144000,20000,12000,10000,
                                9000,9000,7000,6000)

data1 <- data.frame(Company,Number_of_Cars_in_Service)

sort_data  <- data1[order(-Number_of_Cars_in_Service),]

median(sort_data$Number_of_Cars_in_Service)
```

R code Exa 3.1.c Mean

```
# Mean Example :

```

```
Company <- c("Enterprise","Hertz","Natioanl/Alamo","Avis", "Dollar", "Budget","Advantage",
              "U-save","Payless","ACE","Fox","Rent-A-Wreck","Traingle")

Number_of_Cars_in_Service <- c(643000,327000,233000,204000,167000,144000,20000,12000,10000,
                              9000,9000,7000,6000)

data1 <- data.frame(Company,Number_of_Cars_in_Service)

sort_data  <- data1[order(-Number_of_Cars_in_Service),]

mean(sort_data$Number_of_Cars_in_Service)
```

R code Exa 3.2 Determine the 30th percentile of the following eight numbers

```
# Determine the 30th percentile of the following eight numbers :
data3 <- c(5,12,13,14,17,19,23,28)
N = 8
P = 30

# 30th percentile value is :
a <- quantile(data3,c(.30))
cat("30th percentile value is : ",a)
```

R code Exa 3.3 Quartiles

```
# Quartiles :

Category <- c("Automotive","Personal Care","
    Entertainment & Media",
                    "Food","Drugs", "Electronics","Soft
                        Drinks","Retail","Cleaners",
                    "Restaurants","Computers","Telephone",
                        "Financial",
                    "Beer Wine & Liquor","Candy","Toys")

Ad_spending <- c
    (22195,19526,9538,7793,7707,4023,3916,3576,3571,3553,3247,2488,

                        2433,2050,1137,699)

advertise_age <- cbind(Category,Ad_spending)
View(advertise_age)

N=16

# Q1 = P25 is found by :
i = 25/100*N
i

#Q3 = P75 is solved by :
i1 =75/100*(N)
i1

# Quantile :
quantile(Ad_spending)
```

R code Exa 3.5 Chebyshevs Theorem

```
# Chebyshev's Theorem :

```

```
avg_age = 28
sd = 6

# Chebyshev's theorem states that at least (1 - 1)/k
    ^2 proportion of the values are within
#(mean+k*sd). Because 80% of the values are within
    this range, let

#1- (1/k^2) = .80

k = sqrt(1/(1-0.80))
k

# now for :
mean = 28
sd = 6

# values are within
r1 = mean + k * sd
r1  #41.41
r2 = mean - k * sd
r2 # 14.58

# Years of age or between 14.6 and 41.4 years old.
```

R code Exa 3.6.a Mean Absolute Deviation

```
# Mean absolute deviation :

x<- c(55,100,125,140,60)
n = 5

# a = abs(x - x_bar), where x_bar = sum(x)/n
a <- c(41,4,29,44,36)

```

```
x <- cbind(x,a)
View(x)

# MAD :
mean_dev <- sum(a)/n
mean_dev
```

R code Exa 3.6.b Variance and Standard deviation

```
# Variance and stanadard deviation :

x<- c(55,100,125,140,60)
n = 5

# a = abs(x - x_bar), where x_bar = sum(x)/n
a <- c(41,4,29,44,36)

# b = (x - x_bar)^2
b <- c(1681,16,841,1936,1296)

y <- cbind(x,a,b)
View(y)

# Variance :
var(x)

# standard deviation :
sd(x)
```

R code Exa 3.7 Mean Median Mode Variance and Standard deviation

```
# Mean, Median, Mode, Variance, and Standard
    deviation :
```

```r

class <- c("10-under-15","15-under-20","20-under-25","25-under-30","30-under-35",
           "35-under-40","40-under-45","45-under-50")
freq <- c(6,22,35,29,16,8,4,2)
class <-data.frame(class,freq)
class

# Mean of each intervals :
a <- mean(10:15)
b<-mean(15:20)
c<-mean(20:25)
d<-mean(25:30)
e<-mean(30:35)
f<-mean(35:40)
g<-mean(40:45)
h<-mean(45:50)
Mean <- rbind(a,b,c,d,e,f,g,h)
Mean

# fM :
for(i in 1:8)
{
  fM <- freq * Mean
}
fM

# group mean :
Group_mean <- sum(fM)/sum(freq)
Group_mean

# Mean - group mean :
for(i in 1:8)
{
  Mean_grpmean <- Mean - Group_mean
}
Mean_grpmean
```

```r

# Square of Mean_grpmean :
Mean_grpmean_sq <- Mean_grpmean^2
Mean_grpmean_sq

# freq * Mean_grpmean_sq :
freq_Mean_grpmean_sq <- freq * Mean_grpmean_sq
freq_Mean_grpmean_sq


var <- sum(freq_Mean_grpmean_sq)/(sum(freq)-1)
var
sd <- sqrt(var)
sd
```

Chapter 4

Probability

R code Exa 4.1 Addition Law

```
# Addition Law : P(F and P) = P(F) + P(P) − P(F or P
    )

Type_of_position <- c("Managerial", "Professional","
    Technical","Clerical")
Sex_male <- c(8,31,52,9)
Sex_female <- c(3,13,17,22)
total_r <- c(11,44,69,31)
total_c <- c(" ",100,55,55)
Compny_HR_data <- cbind(Type_of_position,Sex_male,
    Sex_female,total_r)
Compny_HR_data <- rbind(Compny_HR_data,total_c)
View(Compny_HR_data)

#  F denote the event of female  and  P denote the
    event of professional worker

# Probability of event of female  :
Pb_F = sum(Sex_female)/sum(sum(Sex_female),sum(Sex_
    male))
Pb_F
```

```
# Probability of event of professional worker :
Pb_P = sum(Sex_male[2],Sex_female[2])/sum(sum(Sex_female),sum(Sex_male))
Pb_P

# Probability of female or Professional worker :
Pb_F_P = Sex_female[2]/sum(sum(Sex_female),sum(Sex_male))
Pb_F_P

# probability that the employee is female or a professional worker :
Pb_F_a_P <- Pb_F + Pb_P - Pb_F_P
Pb_F_a_P
```

R code Exa 4.3 Special Law of Addition

```
# Special Law of Addition : P (T and C) = P (T) + P (C)

Type_of_position <- c("Managerial", "Professional","Technical","Clerical")
Sex_male <- c(8,31,52,9)
Sex_female <- c(3,13,17,22)
total_r <- c(11,44,69,31)
total_c <- c(" ",100,55,55)
Compny_HR_data <- cbind(Type_of_position,Sex_male,Sex_female,total_r)
Compny_HR_data <- rbind(Compny_HR_data,total_c)
View(Compny_HR_data)

# T denote technical , C denote clerical , and P denote professional .

```

```
# Probability of Technical position  :
Pb_T = sum(Sex_male[3],Sex_female[3])/sum(sum(Sex_
    female),sum(Sex_male))
Pb_T

# Probability of Clerical position  :
Pb_C = sum(Sex_male[4],Sex_female[4])/sum(sum(Sex_
    female),sum(Sex_male))
Pb_C

# Probability of professional position :
Pb_P = sum(Sex_male[2],Sex_female[2])/sum(sum(Sex_
    female),sum(Sex_male))
Pb_P

#  probability that a worker is either technical or
    clerical is  :
Pb_T_C = Pb_T + Pb_C
Pb_T_C

#   probability that a worker is either professional
     or clerical is :
Pb_P_C = Pb_P + Pb_C
Pb_P_C
```

R code Exa 4.5 Multiplication Law

```
# General Law of Multiplication : P (X or Y) = P(X)*
    P(Y|X) = P(Y)*P(X|Y)

Total_emp = 140
supervisor = 30
Married_emp = 80
Pb_S_M = .20  # P(S|M) i.e. married employees are
    supervisors
```

```
# probability that the employee is married :
Pb_M = Married_emp/Total_emp
Pb_M

# probability that the employee is married and is a supervisor :
Pb_M_s <- Pb_M * Pb_S_M
Pb_M_s

# 11.43% of the 140 employees are married and are supervisors
```

R code Exa 4.6 General Law of Multiplication

```
# General Law of Multiplication :

Industry_type <- c("Finance_A", "Manufacturing_B","Communication_C")
Northeast_D <- c(.12,.15,.14)
Southeast_E <- c(.05,.03,.09)
Midwest_F <- c(.04,.11,.06)
West_G <- c(.07,.06,.08)
total_r <- c(.28,.35,.37)
total_c <- c(" ",.41,.17,.21,.21,1.00)
Industry_type <- cbind(Industry_type,Northeast_D,Southeast_E,Midwest_F,West_G,total_r)
Industry_type <- rbind(Industry_type,total_c)
View(Industry_type)

# a.) P(Manufacturing_B and Southeast_E) :
P_B_E <- total_r[2]*(Southeast_E[2]/total_r[2])
P_B_E

# b.) P(West_G and Finance_A) :
```

```
P_G_A <- sum(Midwest_F) *(West_G[1]/sum(Midwest_F))
P_G_A

# c.) P(Manufacturing_B and Communication_C) :
P_B_C <- .0
P_B_C  # The matrix shows no intersection for these two events.
# Thus B and C are mutually exclusive.
```

R code Exa 4.8 Special Law of Multiplication

```
# Special law of Mulyiplication : If X, Y are independent, P (X or Y) = P (X) * P (Y)


T1 <- c("A", "B","C")
D <- c(8,20,6)
E <- c(12,30,9)
total_r <- c(20,50,15)
total_c <- c(" ",34,51,85)
T1 <- cbind(T1,D,E,total_r)
T1 <- rbind(T1,total_c)
View(T1)

# Probability of B :
Pb_B = sum(D[2],E[2])/sum(total_r)
Pb_B

# Probability of D :
Pb_D = sum(D)/sum(total_r)
Pb_D

# Probability of B and D is :
Pb_B_D = Pb_B * Pb_D
Pb_B_D
```

R code Exa 4.9 Conditional Probability

```r
# Conditinal Probability : P (X|Y) = P(X or Y)/P(Y) = (P(X)*P(Y|X))/P(Y)

Industry_type <- c("Finance_A", "Manufacturing_B","Communication_C")
Northeast_D <- c(.12,.15,.14)
Southeast_E <- c(.05,.03,.09)
Midwest_F <- c(.04,.11,.06)
West_G <- c(.07,.06,.08)
total_r <- c(.28,.35,.37)
total_c <- c(" ",.41,.17,.21,.21,1.00)
Industry_type <- cbind(Industry_type,Northeast_D,Southeast_E,Midwest_F,West_G,total_r)
Industry_type <- rbind(Industry_type,total_c)
View(Industry_type)

#a.) P(Manufacturing_B | Midwest_F) = P(Manufacturing_B and Midwest_F)/P(Midwest_F)
Pb_B_F = Midwest_F[2]/sum(Midwest_F)
Pb_B_F

#b.) P(West_G | Communication_C) = P(West_G and Communication_C)/P(Communication_C)
Pb_G_C = West_G[3]/sum(Northeast_D[3],Southeast_E[3],Midwest_F[3],West_G[3])
Pb_G_C

#c.) P(Northeast_D | Midwest_F) = P(Northeast_D and Midwest_F)/P(Midwest_F)
Pb_D_F = .00/sum(Midwest_F)
Pb_D_F
```

R code Exa 4.11 Independent Event

```
# Independent Event : P(X|Y) = P(X) and P(Y|X) = P(Y)

T1 <- c("A", "B","C")
D <- c(8,20,6)
E <- c(12,30,9)
total_r <- c(20,50,15)
total_c <- c(" ",34,51,85)
T1 <- cbind(T1,D,E,total_r)
T1 <- rbind(T1,total_c)
View(T1)

# Check the ???rst cell in the matrix to ???nd whether P(A|D) = P(A)
Pb_A_D <- D[1]/sum(D)  # P(A|D)
Pb_A_D

P_A <- sum(D[1],E[1])/sum(total_r)
P_A # P(A)
```

R code Exa 4.12 Bayes Rule

```
# Bayes's Rule : P (Xi|Y) = P(Xi)*P(Y|Xi) / P(X1)*P(Y|X1)+P(X2)*P(Y|X2)+...+P(Xn)*P(Y|Xn)

Event <- c("A","B","C")
Prior <- c(.60,.30,.10)   # P(Ei)
Conditional <- c(.40,.50,.70)  # P(x|Ei)
Joint <- c(.24,.15,.07)  # P(X and Ei) = P(Ei)*P(x|Ei)
```

```
Posterior <- c(.52,.33,.15) # P(X and Ei)/sum(P(X and Ei))

machine <- cbind(Event,Prior,Conditional,Joint,Posterior)
machine

# Revised Probabilities :
machine_A <- Prior[1]* Conditional[1]/sum(Joint)
machine_A

machine_B <- Prior[2]* Conditional[2]/sum(Joint)
machine_B

machine_C <- Prior[3]* Conditional[3]/sum(Joint)
machine_C
```

Chapter 5

Discrete Distributions

R code Exa 5.1 Variance and standard deviation of a Discrete Distribution

```
# Variance and standard deviation of a Discrete
   Distribution :

Prize <- c(1000,100,20,10,4,2,1,0) # x
Probability <- c
   (.00002,.00063,.00400,.00601,.02403,.08877,.10479,.77175)
    # P(x)

# x * P(x) :
for(i in 1:8){
  x_Pb <- Prize*Probability  # x * P(x)
}
print(x_Pb)

# sum Of x * P(x) :
x_Pb_s <- sum(x_Pb)
x_Pb_s


# (x - x_Pb_s)^2
for(j in 1:8){
```

```
  x_mean_sq <- (Prize - x_Pb_s)^2
}
print(x_mean_sq)


# (x − x_Pb_s)^2 * P(x) :
for(j in 1:8){
  x_mean_sq_Pb <- (Prize - x_Pb_s)^2 * Probability
}
print(x_mean_sq_Pb)

# sum of (x − x_Pb_s)^2 * P(x) :
x_mean_sq_Pb_s <- sum(x_mean_sq_Pb)
x_mean_sq_Pb_s

Prize <- cbind(Prize,Probability,x_mean_sq,x_mean_sq
   _Pb)
View(Prize)

# Variance and Standard deviation :
var <- x_mean_sq_Pb_s
var
sd <- sqrt(var)
sd
```

R code Exa 5.2 Binomial Distribution

```
# Binomial Distribution : P(x) = nCx*p^x*q^n−x = n!/
   x!(n − x)! *p^x*q^n−x

p =.65
q = 1-p
n = 25
x = 19
x1 = 0:19
```

```
# Binomial Distribution through inbuild function in
    r :
bd <- dbinom(x,n,p)
bd

# Binomial Distribution through formula :
bd <- (factorial(n)/(factorial(x)*factorial(n-x))) *
    (p^x) * (q^(n-x))
bd
```

R code Exa 5.3 Binomial Distribution ex 2

```
# Binomial Disribution ex 2 :

p = .06
q = .94
n = 20

x <- c(0,1,2)
c<-choose(n,x)*(p^x) * (q^(n-x))
c
sum(c)
```

R code Exa 5.5 Using Binomial Table

```
# using Binomial Table :

n = 20
p = .10
q = 1-p

x <- c(0,1,2,3)
```

```
c<-choose(n,x)*(p^x) * (q^(n-x))
c

#  Probability that fewer than four purchasers
   choose Oreos i.e. x<4 :
sum(c) # about 86.7% of the time fewer than four of
   the 20 will select Oreos
```

R code Exa 5.6 Mean and standard deviation in Binomial distribution

```
# Mean and standard deviation in Binomial
   distribution :
# mean = n * p  and sd = sqrt(n*p*q)

n = 10
p<-c(.10,.20,.30,.40)
q = 1-p

# mean <- n*p
for(p1 in 1:4){
  mean = n*p
}
print(mean)

pd<-pbinom(2,n,p)


p<-cbind(p,mean,pd)
p
```

R code Exa 5.7 Poissions formula

```
# Poission formula : P(x) =lamda^x*e^-lamda/x!
```

```r

l <- 3.2 # lamda
# x>7 customers/4 minutes

# through in build function of poission in r:
dpois(8,lambda = 3.2) # x=8

# x = 8 through formula :
x = 8
pd_8 <- (l^x*exp(-l))/factorial(x)
pd_8

# x = 9 through formula :
x = 9
pd_9 <- (l^x*exp(-l))/factorial(x)
pd_9

# x = 10 through formula :
x = 10
pd_10 <- (l^x*exp(-l))/factorial(x)
pd_10

# x = 11 through formula :
x = 11
pd_11 <- (l^x*exp(-l))/factorial(x)
pd_11

# x = 12 through formula :
x = 12
pd_12 <- (l^x*exp(-l))/factorial(x)
pd_12

# x = 13 through formula :
x = 13
pd_13 <- (l^x*exp(-l))/factorial(x)
pd_13

# Poission distribution for x>=8
```

```
sum(pd_8,pd_9,pd_10,pd_11,pd_12,pd_13)
```

R code Exa 5.8 Poisson distribution Example

```
# Poisson distribution Example :
# Poission formula : P(x) =lamda^x*e^-lamda/x!

l=3.2
x = 10
pd <- dpois(x,l,log=FALSE)
pd

# probability of getting exactly 10 customers during
    an 8-minute interval
l1=6.4
x1 = 10
pd1 <- dpois(x1,l1,log=FALSE)
pd1
```

R code Exa 5.9 Using poissions table

```
# using poission table :

l <- 1.6
x<- c(6,7,8,9)


# Poission probability for x>5 :
p<-dpois(x,l)
p
sum(p)
```

R code Exa 5.10 Probability Example

```
# Probability Example :

p = .0003
n= 10000
l <- n*p
l
x<- c(7,8,9,10,11,12)

# Binomial probability for x>5 :
b<-dbinom(x,n,p)
b
sum(b)


# Poission probability for x>5 :
p<-dpois(x,l)
p
sum(p)
```

R code Exa 5.11 Hypergeometrics distribution

```
# Hypergeometric distribution : P(x) = ACx*(N-A)C(n-x)/NCn

# N = size of the population , n = sample size , A = number of successes in the population , x = number of successes in the sample; sampling is done without replacement

N = 18
```

```
n = 3
A = 12

# Using choose function :

1-((choose(A,0)*choose((N-A),n))/choose(N,n))
```

Chapter 6

Continuous Distributions

R code Exa 6.1 Uniform Distribution

```
# Probabilities in Uniform Distribution : P(x) = x2−
  x1 / b−a  where: a<=x1<=x2<=b

b = 39
a = 27

f_x = 1/(b-a) # f(x)
f_x

u <- (a+b)/2 #mean
u

sd <- (b-a)/sqrt(b-a)  # standard deviation
sd

# P(30 <= x <= 35) :
P = (35-30)/(39-27)
P

# P(x<30) :
P1 = (30-27)/(39-27)
```

```
P1
```

R code Exa 6.2 MEAN AND STANDARD DEVIATION OF A UNIFORM DISTRIBUTION

```
# MEAN AND STANDARD DEVIATION OF A UNIFORM
    DISTRIBUTION :

u = 691 # mean
a = 200
b = 1182
x1 = 410
x2 = 825
sd <- (b-a)/sqrt(12)  # standard deviation
sd

# height of distribution :
f_x = 1/(b-a) # f(x)
f_x

# probability that a randomly selected person pays
    between $410 and $825 annually for automobile
    insurance in the US:
p_x = (x2-x1)/(b-a)
p_x
```

R code Exa 6.3 Normal Curve distribution

```
# Normal Curve distribution :

mean = 494
sd=100
x =700
```

```

# probability of x greater than 700 :
pnorm(x, mean, sd, lower.tail=FALSE)
```

R code Exa 6.4 PROBABILITY OF A UNIFORM DISTRIBUTION

```
# PROBABILITY OF A UNIFORM DISTRIBUTION

x = 550
mean = 494
sd = 100
lb =.2123  # probability of values between 550 and
    the mean
ub =.5000  # probability of values less than the
    mean


# using r function :
pnorm(x, mean, sd)

# Or using normal formula :
z=(x-mean)/sd
z

ub+lb # probability of values 550
```

R code Exa 6.5 Probability of Normal Curve DISTRIBUTION

```
# Probability of Normal Curve DISTRIBUTION :

x = 600
mean = 494
sd = 100
```

```
x1 = 300

a <- pnorm(x1, mean, sd, lower.tail=FALSE)
a
b <- pnorm(x, mean, sd, lower.tail=FALSE)
b

# probability of a value between 300 and 600 :
a - b
```

R code Exa 6.6 PROBABILITY OF A UNIFORM DISTRIBUTION

```
# PROBABILITY OF A UNIFORM DISTRIBUTION

x = 350
mean = 494
sd = 100
x1 = 450

a <- pnorm(x, mean, sd, lower.tail=FALSE)
a
b <- pnorm(x1, mean, sd, lower.tail=FALSE)
b

# probability of a value between 350 and 450 :
a-b
```

R code Exa 6.7 MEAN OF A UNIFORM DISTRIBUTION

```
# MEAN OF A UNIFORM DISTRIBUTION

x = 449
z = 1.11 # value taken from z table
```

```
sd = 36
# z = (x − mean)/sd

mean = x - (z*sd)
mean
```

R code Exa 6.8 Normal distribution using z value

```
# Normal distribution using z value :

mean = 3.58
z = -0.46 # value taken from z table
sd = 1.04
# z = (x − mean)/sd

x = (z*sd) + mean
x

#  67.72% of the daily average amount of solid waste
    per person weighs more than 3.10 pound.
```

R code Exa 6.9 Binomial distribution problem by using the normal distribution

```
#  binomial distribution problem by using the normal
    distribution :

x = 12
n = 25
p = .40
q = 1-p

mean = n * p
```

```
mean

sd = sqrt(n*p*q)
sd

# test : mean +/- 3sd
test1 <- mean + 3*sd
test2 <- mean - 3*sd
test1
test2

# test :  2.65 to 17.35

# z value at x = 12.5
x = 12.5
z = (x-mean)/sd
z

# z value at x = 12.5
x = 11.5
z = (x-mean)/sd
z

#z = 1.02 produces a probability of .3461.
# z = 0.61 produces a probability of .2291.

# The difference in areas yields the following
    answer:
0.3461 - .2291
```

R code Exa 6.10 Binomial distribution by using the normal distribution

```
# Binomial distribution by using the normal
    distribution :

```

```
p = .37
n = 100
q=1-p
mean1 = n*p
mean1
sd = sqrt(n*p*q)
sd

#  range :
u = mean +3*(sd)
u
l = mean - 3*(sd)
l

x = 26.5
z=(x-mean)/sd
z

#  tail of the distribution :
.5000-.4850

x1 <- c(26:20)
b<-dbinom(x1,n,p)
b
sum(b)
```

R code Exa 6.11 Exponential Distribution

```
# Exponential Distribution : f(x) = lambda * e^-
    lambda*x

# Probability of right tail exponential distribution
     : P(x>=x0) = e^-lambda*x0

l = 1.38 # lambda
```

```
mean = 1/l
mean
x0 = .75

# P(x>=x0) :
P <- exp(-l*x0)
P

# for x0 = 0.75, Probability < x0 :
Prob = 1-P
Prob
```

Chapter 7

Sampling and Sampling Distributions

R code Exa 7.1 Z formula for sample means

```
# Z formula for sample means : z = (sample_mean − average)/(standard_dev/sqrt(sample_size))

mean = 448
sd = 21/sqrt(49)
n = 49 # sample size
# sample mean : 441 <= x_bar <= 446
samplemean_l = 441
samplemean_u = 446

a <-pnorm(samplemean_l, mean, sd, lower.tail=FALSE)
a
b <-pnorm(samplemean_u, mean, sd, lower.tail=FALSE)
b


#  probability of a value being between z = −2.33 and −0.67 is :
prob = a - b
```

```
prob

# The probability of a value being between z=2.33
    and −0.67 is .2426; that is,
# there is a 24.26% chance of randomly selecting 49
    hourly periods for
# which the sample mean is between 441 and 446
    shoppers.
```

R code Exa 7.2 Z formula for Sample mean of a finite population

```
# Z formula for Sample mean of a finite population :
# z = (samplemean − population_mean)/(sd/sqrt(n))*(
    sqrt((N−n)/(N−1)))

pop_mean = 37.6 # avg
pop_sd = 8.3 # sd
n = 45 # sample size
N = 360 # finite population
sample_mean = 40

sd = (pop_sd/sqrt(n))*(sqrt((N-n)/(N-1)))

pnorm(sample_mean,pop_mean,sd,lower.tail=TRUE)
```

R code Exa 7.3 Z formula for Sample Proportion

```
# Z formula for Sample Proportion :
# z = (sample_proportion − population_prop)/sqrt((
    population_prop*q)/sample size)

p = 0.10 # population_prop
sample_prop = 12/80
```

```
n = 80
q = 1-p

sd = sqrt(p*q/n)

# P(sample_prop >= .15) :
pnorm(sample_prop,p,sd,lower.tail=FALSE)
```

Chapter 8

Statistical Inference Estimation for Single Populations

R code Exa 8.1 Confidence interval to Estimate Population mean

```
# Confidence interval to Estimate Population mean :
# pop_mean +/- z*(sd/sqrt(n))

n = 44
sample_mean = 10.455
sd = 7.7
z = 1.645

pop_mean_1 = sample_mean - (z*(sd/sqrt(n)))
pop_mean_1

pop_mean_2 = sample_mean + (z*(sd/sqrt(n)))
pop_mean_2
```

R code Exa 8.2 Confidence interval to Estimate Population mean using Finite Correction

```
# Confidence interval to Estimate Population mean using finite correction :
# (pop_mean) +/- (z*(sd/sqrt(n))*sqrt(N-n/N-1))

n = 50
N = 800
sample_mean = 34.30
sd = 8
z = 2.33

pop_mean_1 = sample_mean - (z*(sd/sqrt(n))*sqrt((N-n)/(N-1)))
pop_mean_1

pop_mean_2 = sample_mean + (z*(sd/sqrt(n))*sqrt((N-n)/(N-1)))
pop_mean_2
```

R code Exa 8.3 Confidence Interval to Estimate population mean Population standard deviation unknown and population normally distributed

```
# Confidence Interval to Estimate population mean : Population standard devition unknown and population normally distributed
# pop_mean +/- t*(sd/sqrt(n)) , df = n-1
a<- c(3,1,3,2,5,1,2,1,4,2,1,3,1,1)
n = 14
df = n-1
t = 3.012
sd = 1.29
sample_mean = 2.14

pop_mean_1 = sample_mean - (t*(sd/sqrt(n)))
pop_mean_1

```

```
pop_mean_2 = sample_mean + (t*(sd/sqrt(n)))
pop_mean_2
```

R code Exa 8.4 Confidence Interval to estimate Population Proportion

```
# Confidence Interval to estimate Population
    Proportion :
# p = samp_prop +/- (z*sqrt(samp_prop*q/sample size)

samp_prop = 0.51
q = 1-samp_prop
z = 1.75
n = 210 # sample size

p_1 = samp_prop - (z*sqrt(samp_prop*q/n))
p_1

p_2 = samp_prop + (z*sqrt(samp_prop*q/n))
p_2
```

R code Exa 8.5 Confidence Interval to estimate Population Proportion

```
# Confidence Interval to estimate Population
    Proportion :
# p = samp_prop +/- (z*sqrt(samp_prop*q/sample size)

samp_prop = 34/212  # sample size =212 and no. of
    jeans = 34
q = 1-samp_prop
z = 1.645
n = 212 # sample size

p_1 = samp_prop - (z*sqrt(samp_prop*q/n))
```

```
p_1

p_2 = samp_prop + (z*sqrt(samp_prop*q/n))
p_2
```

R code Exa 8.6 Confidence to estimate the Population Variance

```

# Confidence to estimate the Population Variance :
# var = ((n-1)*s^2)/(X(a/2))^2 or ((n-1)*s^2)/(X(1-a
   /2))^2 , df = n-1

s = 1.12
n = 25
df = n-1

a = qchisq(0.975, df=24)
a
b = qchisq(.025, df=24)
b

var_1 = ((n-1)*s^2)/a
var_1

var_2 = ((n-1)*s^2)/b
var_2
```

R code Exa 8.7 Sample Size when Estimating Population mean

```
# Sample Size when Estimating Population mean :
# n = (z*sd/E)^2

E = 1 # error in estimating
```

```
z = 1.96
sd = 5

n = (z*sd/E)^2
n
```

R code Exa 8.8 Sample size when estimating population proportion

```
# Sample size when estimating population proportion
  :
# n = z^2*p*q/E^2

E = .03
p = .40
z = 2.33
q = 1-p

n = z^2*p*q/E^2
n
```

Chapter 9

Statistical Inference Hypothesis Testing for Single Populations

R code Exa 9.1 Test Hypothesis about population mean

```
# Formula to test Hypoyhesis about population mean
    :
# z = sample_mean - pop_mean/(sd/sqrt(n))

pop_mean = 4.30
sample_mean = 4.156
sd = .574
n = 32
a = .05 # alpha value

# Calcultaed value of test statistic :
z1 = (sample_mean - pop_mean)/(sd/sqrt(n))
z1

# Critical Z value associated with alpha = 0.05 :
z = qnorm(.05,lower.tail=TRUE)
z

# critical sample mean :
```

```
sample_mean_c = (z * (sd/sqrt(n))) + pop_mean
sample_mean_c
```

R code Exa 9.2 t test for population mean

```
# t test for population mean :
# t = (sample_mean − pop_mean) / (sd/sqrt(n)) , df = n−1

pop_mean = 471
sample_mean = 498.78
sd = 46.94
n = 23
alpha = 0.05
df = n-1

# t−distribution function to calculate critical t−value using alpha and df:
qt(alpha, df, lower.tail = FALSE, log.p = FALSE)

# Observed t valueusing sample mean and standard deviation :
t = (sample_mean - pop_mean) / (sd/sqrt(n))
t

# The observed t value of 2.84 is greater than the table t value of 1.717,
# so the business researcher rejects the null hypothesis.
```

R code Exa 9.3 z test of a population proportion

```
# z test of a population proportion :
```

```
# z = sample_prop - population_prop/sqrt(population_
   prop*q/n)

n = 550
x = 115
sample_prop = 115/550
population_prop = .17
q = 1- population_prop

# test statistic value of z :
z1 = (sample_prop - population_prop)/sqrt((
   population_prop*q)/n)
z1

# critical value of z :
z = qnorm(.05,lower.tail=FALSE)
z

# critical sample proportion :
sample_prop_c = z * sqrt(population_prop*q/n) +
   population_prop
sample_prop_c
```

R code Exa 9.4 Test Hypothesis about a population variance

```
# Test Hypothesis about a population variance :
# X^2 = (n-1)*s^2/var , df = n-1

var = 25
n = 16
s_sq = 28.0625 # sample variance
df = n-1

# Two tailed test and alpha = .10 it will be divided
    by 2 :
```

```
a <- .10/2

# we have two critical values of chi square :

# 1st chi-sq value is a :
qchisq(a, df=15)

# 2nd chi-sq is 1-a :
qchisq(1-a, df=15)

# The decision rule is to reject the null hypothesis if the observed value
# of the test statistic is less than 7.26093 or greater than 24.9958.

X_sq = ((n-1)*s_sq)/var
X_sq

# This observed chi-square value is in the nonrejection region because
# chi_sq(.05)=7.26 < chi_sq(observed) = 16.83 < chi_sq(.95) = 24.9958.
# The company fails to reject the null hypothesis. The population variance
# of overtime hours per week is 25.
```

R code Exa 9.5 Z value for Type II error

```
# Z value for Type II error : z = sample_mean_c - pop_mean_1/(sd/sqrt(n))

sample_mean_c = 11.979
pop_mean_1 = 11.96
sd = .10
n = 60
```

```
z = (sample_mean_c - pop_mean_1)/(sd/sqrt(n))
z
```

R code Exa 9.6 Z value for Type II error

```
# Z value for Type II error

z_c = 1.96
p = .40
q =1-p
n = 250
# z_c = (p_c-p)/sqrt(p*q/n)
p_c = z_c*sqrt((p*q)/n)+p
p_c
p_c1 = z_c*sqrt((p*q)/n)-p
p_c1

# z value on taking p_c = .46 and p = .36 :
p_c = .46
p = .36
z_c = (p_c-p)/sqrt(p*q/n)
z_c

# z value on taking p_c = .34 and p = .36 :
p_c = .34
p = .36
z_c = (p_c-p)/sqrt(p*q/n)
z_c

# The area associated with z = 3.29 is .4995.
    Combining this value with the .2454 obtained from
     the left side of the distribution in graph (b)
    yields the total probability of committing a Type
     II error:
```

```
.2454+.4994
```

Chapter 10

Statistical Inferences About Two Populations

R code Exa 10.1 Z formula for the difference in Two Sample Means

```
# z formula for the difference in two sample means :
# z = (samp_mean_1−samp_mean_2)−(pop_mean_1−pop_mean_2)/sqrt((sd1^2/n1)+(sd2^2/n2))

samp_mean_1 = 3352
samp_mean_2 = 5727
sd1 = 1100
sd2 = 1700
n1 = 87
n2 = 76

# Observed value of Z :
z1 = ((samp_mean_1-samp_mean_2)-(0))/sqrt((sd1^2/n1)+(sd2^2/n2))
z1

# Critical value of Z :
z = qnorm(.001, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
```

```
z

# sample critical :
s_c = (0)-(z*sqrt((sd1^2/n1)+(sd2^2/n2)))
s_c

# The difference in sample means would need to be at least 704.23
# to reject the null hypothesis.

#  The actual sample difference in this problem :
s_c = samp_mean_1-samp_mean_2
s_c # which is considerably larger than the critical value of difference

#  Thus, with the critical value method also, the null hypothesis is rejected.
```

R code Exa 10.2 Confidence Interval to estimate difference in two population means

```
# Confidence Interval to estimate difference in two population means :
# pop_mean_1-pop_mean_2 = (samp_mean_1-samp_mean_2) +/- (z*sqrt((sd1^2/n1)+(sd2^2/n2)))

n1 = 50
n2 = 50
samp_mean_1 = 21.45
samp_mean_2 = 24.6
sd1 = 3.46
sd2 = 2.99
z = 1.96
pmean_diff_1 = (samp_mean_1-samp_mean_2) + (z*sqrt((sd1^2/n1)+(sd2^2/n2)))
```

```
pmean_diff_1

pmean_diff_2 = (samp_mean_1-samp_mean_2) - (z*sqrt((
   sd1^2/n1)+(sd2^2/n2)))
pmean_diff_2
```

R code Exa 10.3 t formula to test the difference in means assuming the standard deviations are equal

```
# t formula to test the difference in means assuming
    sd1,sd2 are equal :
#t = (samp_mean_1-samp_mean_2)-(pop_mean_1-pop_mean_
   2)/(sqrt((s1^2(n1-1))+(s2^2(n2-1))/n1+n2-2))*sqrt
   ((1/n1)+(1/n2))

n1 = 46
n2 = 26
samp_mean_1 = 5.42
samp_mean_2 = 5.04
s1 = .58
s2 = .49
df = n1+n2-2

# Critical t value :
qt(.005, df, lower.tail = FALSE, log.p = FALSE)

# Observed t value :
t = ((samp_mean_1-samp_mean_2)-0)/(sqrt(((s1^2*(n1
   -1))+(s2^2*(n2-1)))/(n1+n2-2))*sqrt((1/n1)+(1/n2)
   ))
t

# Because the observed value of is greater than the
    critical table value of the decision is to reject
# the null hypothesis
```

R code Exa 10.4 CONFIDENCE INTERVAL TO ESTIMATE difference in means ASSUMING THE POPULATION VARIANCES ARE UNKNOWN AND EQUAL

```
# CONFIDENCE INTERVAL TO ESTIMATE difference in
    means ASSUMING THE POPULATION VARIANCES ARE
    UNKNOWN AND EQUAL :
n1 = 13
n2 = 15
samp_mean_1 = 4.35
samp_mean_2 = 6.84
s1 = 1.20
s2 = 1.42

alpha = .025
df = 26

t = qt(alpha, df, lower.tail = FALSE, log.p = FALSE)
t

# p_m_diff = pop_mean_1-pop_mean_2
s_diff = samp_mean_1-samp_mean_2
b = sqrt(((s1^2*(n1-1))+(s2^2*(n2-1)))/(n1+n2-2))
c = sqrt((1/n1)+(1/n2))


p_m_diff_1 = s_diff - (t*b*c)
p_m_diff_1

p_m_diff_2 = s_diff + (t*b*c)
p_m_diff_2
```

R code Exa 10.5 t formula to test the Difference in Two Dependent Population

```r
# t formula to test the Difference in Two Dependent Population :
# t = (mean_samp_diff − D  )/(sd/sqrt(n))
# df = n−1
# D = mean_pop_diff , sd = sd_samp_diff , n = num_of_pairs , d= samp_diff_pair


Individual <- c(1,2,3,4,5,6,7)
Before <- c(32,11,21,17,30,38,14)
After <- c(39,15,35,13,41,39,22)
n = 7

for(i in 1:7){
  d = Before - After
}
print(d)
Individual <- cbind(Individual,Before,After,d)
Individual

mean_samp_diff = sum(d)/n
mean_samp_diff
d1 = sum(d)/7

sd = sqrt((sum((d-mean_samp_diff)^2))/(n-1))
sd

D = 0
t = (mean_samp_diff - D  )/(sd/sqrt(n))
t

#  Because the observed value of −2.54 is less than the critical , table value of −1.943 and the
# p−value (0.022) is less than alpha (.05) , the decision is to reject the null hypothesis.
```

R code Exa 10.6 Z formula to test the difference in Population Proportion

```
# Z formula to test the difference in Population
    Proportion :
# z = ((p1_c - p2_c)-(p1-p2)) / sqrt((p_c*q_c)*((1/
    n1)+(1/n2)))
# p_c =((n1*p1_c)+(n2*p2_c))/(n1+n2)
# q_c = 1 - p_c

n1 = 100
n2 = 95
p1_c = .24
p2_c = .41

p_c =((n1*p1_c)+(n2*p2_c))/(n1+n2)
p_c
q_c = 1 - p_c
q_c
# p1 - p2 = 0

z = ( (p1_c - p2_c) - (0) ) / sqrt( (p_c*q_c) * ( (1
    /n1) + (1/n2) ) )
z

#  If a one-tailed test had been used, zc would have
    been z.01 = 2.33,
# and the null hypothesis would have been rejected.
    If alpha had been .05,
# zc would have been z. 025 = , and the null
    hypothesis would have been rejected.
```

R code Exa 10.7 F test for two Population Variance

```r
# F test for two Population Variance :
# F = s1^2/s2^2
# df_num = v1 = n1-1 and df_deno = v2 = n2-1

# from given table we computed :
s1_sq = 5961428.6
s2_sq = 737142.9
n1 = 7
n2 = 8

# critical F-value :
qf(.01, df1=n1-1, df2=n2-1, lower.tail = FALSE, log.p = FALSE)

# Obseved F- value :
F = s1_sq/s2_sq
F

# Because the observed value of F = 8.09 is greater than the table
# critical F value of 7.19, the decision is to reject the null hypothesis.
```

Chapter 11

Analysis of Variance and Design of Experiments

R code Exa 11.1 One Way ANOVA

```
# One Way ANOVA SSE, SSc, SST values :
# SSC = sum(nj*(xj_b-x_b)^2)
# SSE = sum(sum((xij-xj_b)^2))
# SST = sum(sum((xij-x_b)^2))

a <- c(29,27,30,27,28)
b <- c(32,33,31,34,30)
c <- c(25,24,24,25,26)
df <- data.frame(a,b,c)
df

r = c(t(as.matrix(df))) # response data
r
f = c("a", "b", "c")   # treatment levels
k = 3                      # number of treatment levels
n = 5

tm = gl(k, 1, n*k, factor(f))   # matching
```

```
    treatments
tm

av = aov(r ~ tm)
av
summary(av)
```

R code Exa 11.2 TUKEYs HSD Test

```
# TUKEYs HSD Test : HSD = q*sqrt(MSE/n)  # q =
    critical value

a <- c(2.46,2.41,2.43,2.47,2.46)
b <- c(2.38,2.34,2.31,2.40,2.32)
c <- c(2.51,2.48,2.46,2.49,2.44)
d <- c(2.49,2.47,2.48,2.46,2.44)
e <- c(2.56,2.57,2.53,2.55,2.55)
df <- data.frame(a,b,c,d,e)
df


r = c(t(as.matrix(df))) # response data
r
f = c("a", "b", "c","d","e")   # treatment levels
k = 5                         # number of treatment
    levels
n = 5

tm = gl(k, 1, n*k, factor(f))   # matching
    treatments
tm

av = aov(r ~ tm)
av
b <- summary(av)
```

```
b

# From above anova analysis we get MSE value :
MSE = 0.000618
q = 5.29
n = 5
HSD = q*sqrt(MSE/n)
HSD
```

R code Exa 11.3 Randomized Block Design

```
# Formula for computing Randomized Block Design for SSE, SSC, SSR, SST
# SSC = n*sum((xj_b-x_b)^2)
# SSR = C*sum((xi_b-x_b)^2)
# SSE = sum(sum((xij-xj_b-xi_b+x_b)^2))
# SST = sum(sum((xij-x_b)^2))

a <- c(3.47,3.43,3.44,3.46,3.46,3.44)
b <- c(3.40,3.41,3.41,3.45,3.40,3.43)
c <- c(3.38,3.42,3.43,3.40,3.39,3.42)
d <- c(3.32,3.35,3.36,3.30,3.39,3.39)
e <- c(3.50,3.44,3.45,3.45,3.48,3.49)
df <- data.frame(a,b,c,d,e)
df


r = c(t(as.matrix(df))) # response data
r
f = c("a", "b", "c","d","e")    # treatment levels
k = 5                           # number of treatment levels
n = 6

blk = gl(n, k, k*n)             # blocking factor
```

```
blk

tm = gl(k, 1, n*k, factor(f))   # matching treatments
tm

av = aov(r ~ tm + blk)
av
b <- summary(av)
b
```

R code Exa 11.4 Two Way ANOVA

```
#  Two-Way ANOVA :

Types_of_warehouses <- c("GM","GM","GM","GM","GM","GM","GM","GM","GM",
                         "Com","Com","Com","Com","Com","Com","Com","Com","Com",
                         "BS","BS","BS","BS","BS","BS","BS","BS","BS",
                         " CS"," CS"," CS"," CS"," CS"," CS"," CS"," CS"," CS")


Training_sessions <- c("A","A","A","B","B","B","C","C","C","A","A","A",
                       "B","B","B","C","C","C","A","A","A","B","B","B",
                       "C","C","C","A","A","A","B","B","B","C","C","C")

Values <- c(3,4.5,4,2,2.5,2,2.5,
```

```
    1,1.5,5,4.5,4,1,3,2.5,0,1.5,2,2.5,3,3.5,1,3, 1.5,
             3.5,3.5, 4,2,2,3,5, 4.5,2.5,4, 4.5, 5)

df <- data.frame(Types_of_warehouses,Training_
    sessions,Values)
df

av <- aov(Values~as.factor(Types_of_warehouses)*as.
    factor(Training_sessions),data= df)
av
summary(av)
```

Chapter 12

Simple Regression Analysis and Correlation

R code Exa 12.1 Slope of Regression line

```
# Slope of Regression line :

no_of_beds <- c(23,29,29,35,42,46,50,54,64,66,76,78)
FTEs <- c(69,95,102,118,126,125,138,178,156,184,176,225)
Hospitals<-data.frame(no_of_beds,FTEs)
Hospitals

# least squares equation of the regression line is :
lm( FTEs ~ no_of_beds, data=Hospitals)

# y_c = 30.91 + 2.23 * x
```

R code Exa 12.2 Residual Analysis

```
# Residual Analysis :
```

```r

Hospitals <- c(1,2,3,4,5,6,7,8,9,10,11,12)
x <- c(23,29,29,35,42,46,50,54,64,66,76,78)
y <- c(69,95,102,118,126,125,138,178,156,184,176,225)
for(i in 1:12){
  x_sq <- x*x
}
print(x_sq)

for(i in 1:12){
  xy <- x*y
}
print(xy)

x1 <- cbind(x,y,x_sq,xy)

n = 12

b1 = ((sum(x*y))-((sum(x)*sum(y))/n))/((sum(x^2))-(sum(x)^2/n))
b1

b0 = (sum(y)/n)-b1*(sum(x)/n)
b0

# y_c = 30.91 + 2.23 * x
y_c = b0 + b1*x
y_c
x1 <- cbind(x1,y_c)

Residual <- y-y_c
Residual

x1 <- cbind(x1,Residual)
View(x1)

sum(Residual)
```

```
hist(Residual)
```

R code Exa 12.3 Standard Error of Estimation

```
# Standard Error of Estimation : Se = sqrt(SSE/(n-2)
    )
# SSE = sum((y-y_c)^2)

Hospitals <- c(1,2,3,4,5,6,7,8,9,10,11,12)
x <- c(23,29,29,35,42,46,50,54,64,66,76,78)
y <- c
    (69,95,102,118,126,125,138,178,156,184,176,225)
for(i in 1:12){
  x_sq <- x*x
}
print(x_sq)

for(i in 1:12){
  xy <- x*y
}
print(xy)

x1 <- cbind(x,y,x_sq,xy)

n = 12

b1 = ((sum(x*y))-((sum(x)*sum(y))/n))/((sum(x^2))-(
    sum(x)^2/n))
b1

b0 = (sum(y)/n)-b1*(sum(x)/n)
b0

# y_c = 30.91 + 2.23 * x
```

```
y_c = b0 + b1*x
y_c
x1 <- cbind(x1,y_c)

Residual <- y-y_c
Residual

x1 <- cbind(x1,Residual)

for(i in 1:12){
  Residual_sq = Residual^2
}
print(Residual_sq)

x1 <- cbind(x1,Residual_sq)
View(x1)

SSE = sum(Residual_sq)
SSE

Se = sqrt(SSE/(n-2))
Se
```

R code Exa 12.4 Coeficient of Determination

```
# Coeficient of Determination : r_sq = 1 - (SSE/SS_
    yy)
# SS_yy = sum(y_sq)-(sum(y)^2/n)

Hospitals <- c(1,2,3,4,5,6,7,8,9,10,11,12)
x <- c(23,29,29,35,42,46,50,54,64,66,76,78)
y <- c
    (69,95,102,118,126,125,138,178,156,184,176,225)
for(i in 1:12){
  x_sq <- x*x
```

```
}
print(x_sq)

for(i in 1:12){
  xy <- x*y
}
print(xy)

x1 <- cbind(x,y,x_sq,xy)

n = 12

b1 = ((sum(x*y))-((sum(x)*sum(y))/n))/((sum(x^2))-(sum(x)^2/n))
b1

b0 = (sum(y)/n)-b1*(sum(x)/n)
b0

# y_c = 30.91 + 2.23 * x
y_c = b0 + b1*x
y_c
x1 <- cbind(x1,y_c)

Residual <- y-y_c
Residual

x1 <- cbind(x1,Residual)

for(i in 1:12){
  Residual_sq = Residual^2
}
print(Residual_sq)

x1 <- cbind(x1,Residual_sq)
View(x1)

SSE = sum(Residual_sq)
```

```
SSE

SS_yy = sum(y^2)-(sum(y)^2/n)
SS_yy

r_sq = 1-(SSE/SS_yy)
r_sq

# Or r_sq = (b1^2 * SS_xx)/SS_yy
```

R code Exa 12.5 t test for slope

```
# t test for slope :

no_of_beds <- c(23,29,29,35,42,46,50,54,64,66,76,78)
FTEs <- c(69,95,102,118,126,125,138,178,156,184,176,225)
Hospitals<-data.frame(no_of_beds,FTEs)
Hospitals

# critical t value :
qchisq(.01,df = 10)

# least squares equation of the regression line is :
a <- lm( FTEs ~ no_of_beds, data=Hospitals)
a                    # y_c = 30.91 + 2.23 * x
b <- summary(a)
b

# observed t value :
b$coefficients[6]
```

R code Exa 12.6 CONFIDENCE INTERVAL TO ESTIMATE THE SINGAL VALUE FOR A GIVEN VALUE OF x

```
# CONFIDENCE INTERVAL TO ESTIMATE E (yx) FOR A GIVEN
    VALUE OF x :
# y_c +/- t*Se*sqrt((1/n)+((x0-x_b)^2)/SS_xx)
# SS_xx = sum(x^2)-(sum(x)^2/n)

no_of_beds <- c(23,29,29,35,42,46,50,54,64,66,76,78)
FTEs <- c
   (69,95,102,118,126,125,138,178,156,184,176,225)
Hospitals<-data.frame(no_of_beds,FTEs)
Hospitals

a <- lm( FTEs ~ no_of_beds, data=Hospitals)
a

data = data.frame(no_of_beds=40)
data

predict(a, data, interval="confidence")

predict(a, data, interval="predict")
```

R code Exa 12.7 Regression Analysis Example

```
# Regression Analysis Example :

Month <- c("January","Feburary","March","April","May
   ","June","July","August")
Sales <- c
   (32569,32274,32583,32304,32149,32077,31989,31977)
Month_number <- c(1,2,3,4,5,6,7,8)
df <- data.frame(Month,Sales,Month_number)
df
```

```r

library("ggplot2")
ggplot(df, aes(x=Month, y=Sales)) + geom_point(size
   =1)

# Regression Analysis: Sales versus Month
a <- lm(Sales~Month_number, data= df)
a
summary(a)

# y_cap = 32,628.2 - 86.21*x :
x =10
y_cap = 32628.2 - 86.21*x
y_cap
```

Chapter 13

Multiple Regression Analysis

R code Exa 13.1 Multiple Regression Model

```
# Multiple Regression Model:

Year <- c(1980,1982,1984,1986,1988,1990,1992,1994,1996,1998,2000,2002,2004
Prime_Interest_rate <- c(15.26,14.85,12.04,8.33,9.32,10.01,6.25,7.15,8.27,8.35,9.23,4.67,
Unemp_rate <- c(7.1,9.7,7.5,7.0,5.5,5.6,7.5,6.1,5.4,4.5,4.0,5.8,5.5,4.6,5.8)
Personal_saving <- c(10.0,11.2,10.8,8.2,7.3,7.0,7.7,4.8,4.0,4.3,2.3,2.4,2.1,0.7,1.8)
df <- data.frame(Year,Prime_Interest_rate,Unemp_rate,Personal_saving)
df

a <-lm(Prime_Interest_rate ~ Unemp_rate+Personal_saving,data=df)
a
```

```
summary(a)
anova(a)

# y_cap = 7.4904 − 0.6725x1 + 0.9500x2
# If the unemployment rate is 6.5 and the personal saving rate is 5.0,
# the predicted prime interest rate is 7.869%:
x1 = 6.5
x2 = 5.0
y_cap = 7.4904 - (0.6725)*(x1) + (0.9500)*(x2)
y_cap
```

R code Exa 13.2 Multiple Regression Analysis Model

```
#  Multiple Regression Model:

Year <- c(1980,1982,1984,1986,1988,1990,1992,1994,1996,1998,2000,2002,2004
Prime_Interest_rate <- c(15.26,14.85,12.04,8.33,9.32,10.01,6.25,7.15,8.27,8.35,9.23,4.67,
Unemp_rate <- c(7.1,9.7,7.5,7.0,5.5,5.6,7.5,6.1,5.4,4.5,4.0,5.8,5.5,4.6,5.8)
Personal_saving <- c(10.0,11.2,10.8,8.2,7.3,7.0,7.7,4.8,4.0,4.3,2.3,2.4,2.1,0.7,1.8)
df <- data.frame(Year,Prime_Interest_rate,Unemp_rate,Personal_saving)
View(df)

a <-lm(Prime_Interest_rate ~ Unemp_rate+Personal_saving,data=df)
a
```

```
s <-summary(a)
s
anova(a)

pred <- predict(a)
resd <- s$residuals
data <- data.frame(pred,resd)
View(data)
```

Chapter 14

Building Multiple Regression Models

R code Exa 14.1 Model Transformation

```
# Model Transformation : y = B0*x_B1 + E

y_cost <- c(1.2,9.0,4.5,3.2,13.0,0.6,1.8,2.7)
x_weight <- c
    (450,20200,9060,3500,75600,175,800,2100)
y_cost <- data.frame(y_cost,x_weight)
y_cost

# logy = log_B0 + B1*logx + E
log_xy <- log10(y_cost)
log_xy

a <-lm(y_cost~x_weight,data=log_xy)
a
b <-summary(a)
b

b0 <- b$coefficients[1]
b0
```

```
b1 <- b$coefficients[2]
b1

logy_c = b0 + b1 * (sum(log_xy$x_weight)/8)
logy_c

# antilog = 2.9644
# y = (.055857)*x^.49606
```

Chapter 15

Time Series Forecasting and Index Numbers

R code Exa 15.1.a Moving average

```
# Moving average :

Month <- c("January","February","March","April","May","June","July","August","September","October","November","December")
Shipments <- c(1056,1345,1381,1191,1259,1361,1110,1334,1416,1282,1341,1382)
Month <- cbind(Month,Shipments)
Month

# The ???rst moving average is
first_four_Month_Moving_Average = sum(Shipments[1],Shipments[2],Shipments[3],Shipments[4])/4
first_four_Month_Moving_Average
Second_four_Month_Moving_Average = sum(Shipments[5],Shipments[2],Shipments[3],Shipments[4])/4
Second_four_Month_Moving_Average
Third_four_Month_Moving_Average = sum(Shipments[5],
```

```
    Shipments[6],Shipments[3],Shipments[4])/4
Third_four_Month_Moving_Average
fourth_four_Month_Moving_Average = sum(Shipments[5],
    Shipments[6],Shipments[7],Shipments[4])/4
fourth_four_Month_Moving_Average
fifth_four_Month_Moving_Average = sum(Shipments[5],
    Shipments[6],Shipments[7],Shipments[8])/4
fifth_four_Month_Moving_Average
sixth_four_Month_Moving_Average = sum(Shipments[9],
    Shipments[6],Shipments[7],Shipments[8])/4
sixth_four_Month_Moving_Average
seventh_four_Month_Moving_Average = sum(Shipments
    [9],Shipments[10],Shipments[7],Shipments[8])/4
seventh_four_Month_Moving_Average
eight_four_Month_Moving_Average = sum(Shipments[9],
    Shipments[10],Shipments[11],Shipments[8])/4
eight_four_Month_Moving_Average

a = " "
b= " "
c = " "
d =  " "
Average = rbind(a,b,c,d,first_four_Month_Moving_
    Average,Second_four_Month_Moving_Average,Third_
    four_Month_Moving_Average,
           fourth_four_Month_Moving_Average,fifth_
              four_Month_Moving_Average,sixth_four_
              Month_Moving_Average,
           seventh_four_Month_Moving_Average,eight_
              four_Month_Moving_Average)
Average
```

R code Exa 15.1.b Moving average

```
# Error in Moving Average :
```

```
# Moving average :

Month <- c("January","February","March","April","May","June","July","August","September","October","November","December")
Shipments <- c(1056,1345,1381,1191,1259,1361,1110,1334,1416,1282,1341,1382)
Month <- cbind(Month,Shipments)
Month

# The ???rst moving average is
first_four_Month_Moving_Average = sum(Shipments[1],Shipments[2],Shipments[3],Shipments[4])/4
first_four_Month_Moving_Average
Second_four_Month_Moving_Average = sum(Shipments[5],Shipments[2],Shipments[3],Shipments[4])/4
Second_four_Month_Moving_Average
Third_four_Month_Moving_Average = sum(Shipments[5],Shipments[6],Shipments[3],Shipments[4])/4
Third_four_Month_Moving_Average
fourth_four_Month_Moving_Average = sum(Shipments[5],Shipments[6],Shipments[7],Shipments[4])/4
fourth_four_Month_Moving_Average
fifth_four_Month_Moving_Average = sum(Shipments[5],Shipments[6],Shipments[7],Shipments[8])/4
fifth_four_Month_Moving_Average
sixth_four_Month_Moving_Average = sum(Shipments[9],Shipments[6],Shipments[7],Shipments[8])/4
sixth_four_Month_Moving_Average
seventh_four_Month_Moving_Average = sum(Shipments[9],Shipments[10],Shipments[7],Shipments[8])/4
seventh_four_Month_Moving_Average
eight_four_Month_Moving_Average = sum(Shipments[9],Shipments[10],Shipments[11],Shipments[8])/4
eight_four_Month_Moving_Average

a = " "
```

```
b= ” ”
c = ” ”
d = ” ”
Average = rbind(a,b,c,d,first_four_Month_Moving_Average,Second_four_Month_Moving_Average,Third_four_Month_Moving_Average,
                fourth_four_Month_Moving_Average,fifth_four_Month_Moving_Average,sixth_four_Month_Moving_Average,
                seventh_four_Month_Moving_Average,eight_four_Month_Moving_Average)
Average

a = ” ”
b= ” ”
c = ” ”
d = ” ”
Error_May = Shipments[5]-first_four_Month_Moving_Average
Error_June = Shipments[6]-Second_four_Month_Moving_Average
Error_July = Shipments[7]-Third_four_Month_Moving_Average
Error_Aug = Shipments[8]-fourth_four_Month_Moving_Average
Error_sep = Shipments[9]-fifth_four_Month_Moving_Average
Error_oct = Shipments[10]-sixth_four_Month_Moving_Average
Error_nov = Shipments[11]-seventh_four_Month_Moving_Average
Error_dec = Shipments[12]-eight_four_Month_Moving_Average
Error <- rbind(a,b,c,d,Error_May,Error_June,Error_July,Error_Aug,Error_sep,Error_oct,Error_nov,Error_dec)
Error
Month <- cbind(Month,Average,Error)
```

```
View(Month)
```

R code Exa 15.2 Weighted Moving Average

```
# Weighted MOving Average : 3*l + 3*p + 3*b_p/6

Month <- c("January","February","March","April","May
    ","June","July","August","September","October","
    November","December")
Shipments <- c
    (1056,1345,1381,1191,1259,1361,1110,1334,1416,1282,1341,1382)

Month <- data.frame(Month,Shipments)
Month
weights1 <- c(4,2,1,1)

# install.packages("stats")
library(stats)

f_weight_may <- weighted.mean(Shipments[4:1],
    weights1)
f_weight_june <- weighted.mean(Shipments[5:2],
    weights1)
f_weight_july <- weighted.mean(Shipments[6:3],
    weights1)
f_weight_aug <- weighted.mean(Shipments[7:4],
    weights1)
f_weight_sep <- weighted.mean(Shipments[8:5],
    weights1)
f_weight_oct <- weighted.mean(Shipments[9:6],
    weights1)
f_weight_nov <- weighted.mean(Shipments[10:7],
    weights1)
f_weight_dec <- weighted.mean(Shipments[11:8],
    weights1)
```

```r
f_weights <- data.frame(f_weight_may,f_weight_june,f_weight_july,f_weight_aug,
                          f_weight_sep,f_weight_oct,f_weight_nov,f_weight_dec)
f_weights

Shipments[5:12] - f_weights

# We noticed that in this problem the errors obtained by using the 4-month weighted moving average
# were greater than most of the errors obtained by using an  unweighted 4-month moving average
# in Ex15_1.
```

R code Exa 15.3 EXPONENTIAL SMOOTHING

```r
# EXPONENTIAL SMOOTHING :
Year <- c(1:16)
Total_units <- c(1193,1014,1200,1288,1457,1354,1477,1474,1617,1641,1569,
                 1603,1705,1848,1956,2068)
data <- data.frame(Year,Total_units)
data

library(ggplot2)
ggplot(data=data, aes(x=data$Year, y=data$Total_units, group=1)) +
  geom_line(linetype = "dashed")+
  geom_point()

# using  exponential smoothing function i.e. ses() :
# install.package("forecast")
library(forecast)
```

```r
# Forecast and error values for alpha = 0.2 :
f_a <- ses(Total_units, h = 8, alpha = 0.2, initial = "simple")[["fitted"]]
error_a <- ses(Total_units, h = 8, alpha = 0.2, initial = "simple")[["residuals"]]

# Forecast and error values for alpha = 0.2 :
f_b <- ses(Total_units, h = 8, alpha = 0.5, initial = "simple")[["fitted"]]
error_b <- ses(Total_units, h = 8, alpha = 0.5, initial = "simple")[["residuals"]]

# Forecast and error values for alpha = 0.2 :
f_c <- ses(Total_units, h = 8, alpha = 0.8, initial = "simple")[["fitted"]]
error_c <- ses(Total_units, h = 8, alpha = 0.8, initial = "simple")[["residuals"]]

f_data <- data.frame(data,f_a,error_a,f_b,error_b,f_c,error_c)
View(f_data)

# MAD and MSE values of alpha = 0.2, 0.5, 0.8 :
MAD_a <- sum(abs(error_a))/15
MSE_a <- sum(abs(error_a^2))/15

MAD_b <- sum(abs(error_b))/15
MSE_b <- sum(abs(error_b^2))/15

MAD_c <- sum(abs(error_c))/15
MSE_c <- sum(abs(error_c^2))/15

val <- rbind(MAD_a,MSE_a,MAD_b,MSE_b,MAD_c,MSE_c)
val
```

R code Exa 15.4 Regression Trend Analysis Using Quadratic Models

```
# Regression Trend Analysis Using Quadratic Models

Year <- c(1991:2007)
Labour_force <- c
   (117.72,118.49,120.26,123.06,124.90,126.71,129.56,131.46,133.49,13
Year_sq <- Year^2
Year <- data.frame(Year,Labour_force,Year_sq)
Year
a <-lm(Labour_force~Year,data=Year)
a
anova(a)
ggplot(data = Year,aes(x=Year,y=Labour_force))+geom_
   point()+geom_smooth(method = "lm")

b <-lm(Labour_force~.,data=Year)
b
anova(b)
ggplot(data = Year,aes(x=Year_sq,y=Labour_force))+
   geom_point()+geom_smooth(method = "lm")
```

R code Exa 15.5 LASPEYRES PRICE INDEX and PAASCHE PRICE INDEX

```
# LASPEYRES PRICE INDEX and PAASCHE PRICE INDEX :
year <- c(2008,2009)
p.Syrings <- c(6.70,6.95)
q.Syrings <- c(150,135)
p.Cotton <- c(1.35,1.45)
q.Cotton <- c(60,65)
p.Patient <- c(5.10,6.25)
q.Patient <- c(8,12)
p.ChildrenTylenol <- c(4.50,4.95)
```

```r
q.ChildrenTylenol <- c(25,30)
p.Computerpaper <- c(11.95,13.20)
q.Computerpaper <- c(6,8)
p.Thermometer <- c(7.90,9.00)
q.Thermometer <- c(4,2)

data <- data.frame(year,p.Syrings,q.Syrings,p.Cotton,q.Cotton,p.Patient,q.Patient,
                   p.ChildrenTylenol,q.ChildrenTylenol,p.Computerpaper,q.Computerpaper,
                   p.Thermometer,q.Thermometer)
data

# Unweighted Aggregate Index for 2009 :
p_2009 <- sum(p.Syrings[2],p.Cotton[2],p.Patient[2],p.ChildrenTylenol[2],p.Computerpaper[2],
         p.Thermometer[2])
p_2008 <- sum(p.Syrings[1],p.Cotton[1],p.Patient[1],p.ChildrenTylenol[1],p.Computerpaper[1],
              p.Thermometer[1])
I = (p_2009/p_2008)*100
I

# Laspeyres Price Indices
# install.packages("micEcon")
# install.packages("micEconIndex")
library(micEconIndex)
library(micEcon)
a <- priceIndex(c("p.Syrings","p.Cotton","p.Patient","p.ChildrenTylenol","p.Computerpaper",
             "p.Thermometer"), c("q.Syrings","q.Cotton","q.Patient","q.ChildrenTylenol",
                               "q.Computerpaper","q.Thermometer"),1,data)
a
```

```
I_2009_Laspeyres <- a[2]*100
I_2009_Laspeyres

# Paasche Price Indices
b <- priceIndex(c("p.Syrings","p.Cotton","p.Patient","p.ChildrenTylenol","p.Computerpaper",
              "p.Thermometer"), c("q.Syrings","q.Cotton","q.Patient","q.ChildrenTylenol",
                                   "q.Computerpaper","q.Thermometer"),1,data,"Paasche")
b
I_2009_Passache <- b[2]*100
I_2009_Passache
```

Chapter 16

Analysis of Categorical Data

R code Exa 16.1 CHI SQUARE GOODNESS OF FIT TEST

```
# CHI-SQUARE GOODNESS OF-FIT TEST : X_sq = sum((fo-fe)^2/fe)
# df = k-1-c
Month <- c("January","February","March","April","May","June","July","August","September","October","November","December")
fo <- c(1610,1585,1649,1590,1540,1397,1410,1350,1495,1564,1602,1655)

# critical value of chi-square when alpha is 0.01 :
qchisq(.99, df=11)


fe <- sum(fo)/12
for(i in 1:12){
  X = (fo-fe)^2/fe
}
print(X)

# Observed chi-square value :
```

```
X_sq = sum(X)
X_sq

Month <- cbind(Month,fo,fe,X)
Month

# The observed value of chi-square is 74.37, greater
    than the critical table value i.e. 24.725,
# so the decision is to reject the null hypothesis.
   This problem provides enough
# evidence to indicate that the distribution of milk
    sales is not uniform.
```

R code Exa 16.2 Test data is whether in Poisson distributed

```
# Test data is whether in Poisson distributed :

no_of_arrival <- c(0,1,2,3,4,5)
obs_freq <- c(7,18,25,17,12,5)

# chi square value when alpha = 0.05 :
qchisq(.95,4)

for(i in 1:6){
  arr_obs <- no_of_arrival*obs_freq
}
print(arr_obs)

l = sum(arr_obs)/sum(obs_freq)
l # lambda

# Expected probability using lamba and no_of_arrival
    :
exp_pb <- c(.1003,.2306,.2652,.2033,.1169,.0837)

```

```
for(i in 0:5){
  exp_freq = sum(obs_freq)*exp_pb
}
print(exp_freq)

no_of_arrival <- cbind(no_of_arrival,obs_freq,arr_
   obs,exp_pb,exp_freq)
no_of_arrival

for(i in 0:5){
  X = (obs_freq-exp_freq)^2/exp_freq
}
print(X)
sum(X)
```

R code Exa 16.3 CHI SQUARE GOODNESS OF FIT TEST example 2

```
# CHI SQUARE GOODNESS OF FIT TEST example 2 :
p <- c("Milk","non-Milk")
fo <- c(115,435)
fe <- c(93.5,456.5)

# critical value of chi-square :
qchisq(.95, df=1)

X_1 = (fo[1]-fe[1])^2/fe[1]
X_1

X_2 = (fo[2]-fe[2])^2/fe[2]
X_2

# Observed value of chi-square :
X_sq = X_1 + X_2
X_sq

```

```
# This observed chi-square, 5.95, is greater than the critical chi-square value of 3.8415.
# The decision is to reject the null hypothesis.
```

R code Exa 16.4 CHI SQUARE TEST OF INDEPENDENCE

```
# CHI-SQUARE TEST OF INDEPENDENCE :

Age = matrix(c(26,95,18,41,40,20,24,13,32),nrow=3,
    ncol=3,byrow = TRUE)
dimnames(Age) = list(c("21-34","35-55",">55"),c("
    Coffee_tea ", "Soft_Drink", "Other"))
Age

# chi-square expected value when alpha =.01 :
qchisq(.99,df=4)

# The degrees of freedom are (3 - 1)(3 - 1) = 4, and
     the critical value is 13.2767.
# The decision rule is to reject the null hypothesis
     if the observed value of chisquare
# is greater than 13.2767.


# chi-square observed value :
# installed.pacakges("stats")
library(stats)
chisq.test(Age)

#  The observed value of chi-square, 59.41,is
    greater than the critical value, 13.2767,
# so the null hypothesis is rejected.
```

Chapter 17

Nonparametric Statistics

R code Exa 17.1 Mann Whitney U test

```
# Mann-Whitney U test :

Total_emp_comp <- c
   (18.75,19.80,20.10,20.75,21.64,21.90,22.36,22.96,23.45,23.88,24.1
Rank <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
Group <- c("H","H","H","H","E","H","H","H","E","E","
   E","E","E","E","E")
Total_emp_comp <- data.frame(Total_emp_comp,Rank,
   Group)
Total_emp_comp

W1 = 1+2+3+4+6+7+8
W1
W2 = 5+9+10+11+12+13+14+15
W2
U1 = (7)*(8) + ((7)*(8))/2 - W1
U1
U2 = (7)*(8) + ((8)*(9))/2 - W2
U2

```

```
# Using Wilcox test :
wilcox.test(Total_emp_comp ~ Group, data = Total_emp_comp,exact = FALSE)

#Because U2 is the smaller value of U, we use U=3 as the test statistic for Table A.13.
# Because it is the smallest size, let n1=7; n2=8.

# Because the p -value is less than a = .05, the null hypothesis is rejected.
```

R code Exa 17.2 LARGE SAMPLE FORMULAS MANN WHITNEY U TEST

```
#LARGE-SAMPLE FORMULAS MANN-WHITNEY U TEST :

Value <- c(2.25,2.70,2.75,2.97,2.97,3.10,3.15,3.29,3.50,3.60,3.61,3.65,3.68
            4.01,4.05,4.10,4.10,4.25,4.29,4.53,4.75,4.80,4.80,4.98,5.
Rank <- c(1,2,3,4.5,4.5,6,7,8,9,10,11,12,13,14,15,16,17,18.5,18.5,20,21,22
Group <- c('V','R','V','V','V','V','V','V','R','V','V','R','V','R','R',
            'V','V','R','R','R','V','V','R','R','R','R','R','R','R','R')
Value <- data.frame(Value,Rank,Group)
Value

W1 =  1 + 3 + 4.5 + 4.5 + 6 + 7 + 8 + 10 + 11 + 13 + 16 + 17 + 21 + 22
W1

```

```r
U = (14)*(16) + ((14)*(15))/2 - W1
U

U_u = ((14)*(16))/2
U_u

sd_u = sqrt(((14)*(16)*(31))/12)
sd_u

# observed value
z = (U-U_u)/sd_u
z

# Wilcox test :
wilcox.test(Value ~ Group, data = Value,exact = FALSE)
```

R code Exa 17.3 WILCOXON MATCHED PAIRS SIGNED RANK TEST

```r
# WILCOXON MATCHEDPAIRS SIGNED RANK TEST :

Worker <- c(1:20)
Before <- c(5,4,9,6,3,8,7,10,3,7,2,5,4,5,8,7,9,5,4,3)
After <- c(11,9,9,8,5,7,9,9,7,9,6,10,9,7,9,6,10,8,5,6)
d <- c(-6,-5,0,-2,-2,1,-2,1,-4,-2,-4,-5,-5,-2,-1,1,-1,-3,-1,-3)
Rank <- c(-19,-17,0,-9,-9,3.5,-9,3.5,-14.5,-9,-14.5,-17,-17,-9,-3.5,3.5,-3
Worker <- data.frame(Worker,Before,After,d,Rank)
Worker

```

```
# test statistic z value :
qnorm(.99,lower.tail = FALSE)

# T positive and negative using wilcox test function :
wilcox.test(Worker$Before, Worker$After, paired=TRUE)
wilcox.test(Worker$d, Worker$Rank, paired=TRUE)

# T positive and negative using formula :
T_p <- 3.5+3.5+3.5
T_p
T_n <- 19 + 17 + 9 + 9 + 9 + 14.5 + 9 + 14.5 + 17 + 17 + 9 + 3.5 + 3.5 + 12.5 + 3.5 + 12.5
T_n

T_min = min(T_p,T_n)
T_min

n = 19
T_mean = (n*(n+1))/4
T_mean

T_sd = sqrt((n*(n+1)*(2*n+1))/24)
T_sd

# observed z value :
z = (T_min - T_mean)/T_sd
z

# The observed z value (−3.41) is in the rejection region, so the analyst rejects the null hypothesis.
# The productivity is signi???cantly greater after the implementation of quality control
# at this company.
```

R code Exa 17.4 KRUSKAL WALLIS TEST

```
# KRUSKAL-WALLIS TEST  :

Group_native <- c(8,5,7,11,9,6)
Group_water <- c(10,12,11,9,13,12)
Group_fertilizer <- c(11,14,10,16,17,12)
Group_water_fertilizer <- c(18,20,16,15,14,22)
Group <-data.frame(Group_native,Group_water,Group_fertilizer,Group_water_fertilizer)
Group

# alpha = .01, critical value :
qchisq(.99,df=3)

native<- Group$Group_native
water<- Group$Group_water
fertilizer<- Group$Group_fertilizer
water_fertilizer<- Group$Group_water_fertilizer
x1<-c(native,water,fertilizer,water_fertilizer)
x1
g<- factor(rep(1:4, c(6,6,6,6)),
           labels = c("native",
                      "water",
                      "fertilizer",
                      "water_fertilizer"))
kruskal.test(x1, g)


#  The observed K value is 16.77 and the critical is 11.3449.
# Because the observed value is greater than the table value, the null hypothesis
# is rejected. There is a signi???cant difference in
```

```
the way the trees grow
```

R code Exa 17.5 FRIEDMAN TEST

```
# FRIEDMAN TEST :

Brand <- matrix(c
   (3,5,2,4,1,1,3,2,4,5,3,4,5,2,1,2,3,1,4,5,5,4,2,1,3,1,5,3,4,2,4,1,3
              2,3,4,5,1,2,4,5,3,1,3,5,4,2,1),
            nrow=10,ncol=5,byrow = TRUE)
Brand

# Chi-square value , alpha =0.01 :
qchisq(.99,df=4)

# observed value :
friedman.test(Brand)

# Because the observed value of = 3.68 is not
   greater than the critical value , 13.2767,
# the researchers fail to reject the null hypothesis
   .
```

R code Exa 17.6 SPEARMANS RANK CORRELATION

```
# SPEARMAN'S RANK CORRELATION :

Crude_oil <- c
   (14.60,10.50,12.30,15.10,18.35,22.60,28.90,31.40,26.75)
Gasoline <- c
   (3.25,3.26,3.28,3.26,3.32,3.44,3.56,3.60,3.54)
```

```
Crude_rank <- c(3,1,2,4,5,6,8,9,7)
Gasoline_rank <- c(1,2.5,4,2.5,5,6,8,9,7)
d <- c(2,-1.5,-2,1.5,0,0,0,0,0)
d_sq <- c(4,2.25,4,2.25,0,0,0,0,0)
oil <- data.frame(Crude_oil,Gasoline,Crude_rank,
    Gasoline_rank,d,d_sq)
oil
d_sq_sum <- sum(d_sq)
d_sq_sum

# Using cor.test :
# install.packages("stats")
library(stats)
cor.test(oil$Crude_oil, oil$Gasoline, method = "
    spearman")

# using formula :
n = 9
r_s <- 1 - ((6*d_sq_sum)/(n*(n^2-1)))
r_s


# A high positive correlation is computed between
    the price of a barrel of
# West Texas intermediate crude and a gallon of
    regular unleaded gasoline.
```