

R Textbook Companion for  
Business Statistics For Contemporary Decision  
Making  
by Ken Black<sup>1</sup>

Created by  
Vibha Mehta  
B.Tech.

Computer Science and Engineering  
Techno India NJR Institute of Technology

Cross-Checked by  
R TBC Team

May 14, 2020

<sup>1</sup>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   | 4  |
| 2 Charts and Graphs   | 5  |
| 3 Descriptive Statistics  | 12 |
| 4 Probability   | 20 |
| 5 Discrete Distributions  | 28 |
| 6 Continuous Distributions  | 36 |
| 7 Sampling and Sampling Distributions                             | 44 |
| 8 Statistical Inference Estimation for Single Populations         | 47 |
| 9 Statistical Inference Hypothesis Testing for Single Populations | 52 |
| 10 Statistical Inferences About Two Populations                   | 58 |
| 11 Analysis of Variance and Design of Experiments                 | 65 |
| 12 Simple Regression Analysis and Correlation                     | 70 |
| 13 Multiple Regression Analysis                                   | 78 |
| 14 Building Multiple Regression Models                            | 81 |
| 15 Time Series Forecasting and Index Numbers                      | 83 |

|           |                                     |           |
|-----------|-------------------------------------|-----------|
| <b>16</b> | <b>Analysis of Categorical Data</b> | <b>93</b> |
| <b>17</b> | <b>Nonparametric Statistics</b>     | <b>97</b> |

# List of R Codes

|           |  |    |
|-----------|--|----|
| Exa 2.1.a | Class Midpoints . . . . .  | 5  |
| Exa 2.1.b | Relative Frequency . . . . .   | 6  |
| Exa 2.1.c | Cumulative Frequency . . . . .   | 7  |
| Exa 2.2   | Steam and leaf plot . . . . .  | 8  |
| Exa 2.3.a | Bar Graph . . . . .  | 9  |
| Exa 2.3.b | Bar Graph . . . . .  | 10 |
| Exa 2.4   | Scatter Plot . . . . .   | 11 |
| Exa 3.1.a | Mode . . . . .   | 12 |
| Exa 3.1.b | Median . . . . .   | 13 |
| Exa 3.1.c | Mean . . . . .   | 13 |
| Exa 3.2   | Determine the 30th percentile of the following eight numbers . . . . . | 14 |
| Exa 3.3   | Quartiles . . . . .  | 14 |
| Exa 3.5   | Chebyshevs Theorem . . . . .   | 15 |
| Exa 3.6.a | Mean Absolute Deviation . . . . .                                      | 16 |
| Exa 3.6.b | Variance and Standard deviation . . . . .                              | 17 |
| Exa 3.7   | Mean Median Mode Variance and Standard deviation . . . . .             | 17 |
| Exa 4.1   | Addition Law . . . . .   | 20 |
| Exa 4.3   | Special Law of Addition . . . . .                                      | 21 |
| Exa 4.5   | Multiplication Law . . . . .   | 22 |
| Exa 4.6   | General Law of Multiplication . . . . .                                | 23 |
| Exa 4.8   | Special Law of Multiplication . . . . .                                | 24 |
| Exa 4.9   | Conditional Probability . . . . .                                      | 25 |
| Exa 4.11  | Independent Event . . . . .  | 26 |
| Exa 4.12  | Bayes Rule . . . . .   | 26 |
| Exa 5.1   | Variance and standard deviation of a Discrete Distribution . . . . .   | 28 |
| Exa 5.2   | Binomial Distribution . . . . .  | 29 |

|          |   |    |
|----------|---|----|
| Exa 5.3  | Binomial Distribution ex 2 . . . . .  | 30 |
| Exa 5.5  | Using Binomial Table . . . . .  | 30 |
| Exa 5.6  | Mean and standard deviation in Binomial distribution  | 31 |
| Exa 5.7  | Poissions formula . . . . .   | 31 |
| Exa 5.8  | Poisson distribution Example . . . . .  | 33 |
| Exa 5.9  | Using poissions table . . . . .   | 33 |
| Exa 5.10 | Probability Example . . . . .   | 34 |
| Exa 5.11 | Hypergeometrics distribution . . . . .  | 34 |
| Exa 6.1  | Uniform Distribution . . . . .  | 36 |
| Exa 6.2  | MEAN AND STANDARD DEVIATION OF A UNIFORM DISTRIBUTION . . . . .   | 37 |
| Exa 6.3  | Normal Curve distribution . . . . .   | 37 |
| Exa 6.4  | PROBABILITY OF A UNIFORM DISTRIBUTION .   | 38 |
| Exa 6.5  | Probability of Normal Curve DISTRIBUTION . . . .  | 38 |
| Exa 6.6  | PROBABILITY OF A UNIFORM DISTRIBUTION .   | 39 |
| Exa 6.7  | MEAN OF A UNIFORM DISTRIBUTION . . . . .  | 39 |
| Exa 6.8  | Normal distribution using z value . . . . .   | 40 |
| Exa 6.9  | Binomial distribution problem by using the normal distribution . . . . .  | 40 |
| Exa 6.10 | Binomial distribution by using the normal distribution  | 41 |
| Exa 6.11 | Exponential Distribution . . . . .  | 42 |
| Exa 7.1  | Z formula for sample means . . . . .  | 44 |
| Exa 7.2  | Z formula for Sample mean of a finite population . . .  | 45 |
| Exa 7.3  | Z formula for Sample Proportion . . . . .   | 45 |
| Exa 8.1  | Confidence interval to Estimate Population mean . . .   | 47 |
| Exa 8.2  | Confidence interval to Estimate Population mean using Finite Correction . . . . .   | 47 |
| Exa 8.3  | Confidence Interval to Estimate population mean Population standard deviation unknown and population normally distributed . . . . . | 48 |
| Exa 8.4  | Confidence Interval to estimate Population Proportion   | 49 |
| Exa 8.5  | Confidence Interval to estimate Population Proportion   | 49 |
| Exa 8.6  | Confidence to estimate the Population Variance . . . .  | 50 |
| Exa 8.7  | Sample Size when Estimating Population mean . . . .   | 50 |
| Exa 8.8  | Sample size when estimating population proportion . .   | 51 |
| Exa 9.1  | Test Hypothesis about population mean . . . . .   | 52 |
| Exa 9.2  | t test for population mean . . . . .  | 53 |
| Exa 9.3  | z test of a population proportion . . . . .   | 53 |

|            |   |    |
|------------|---|----|
| Exa 9.4    | Test Hypothesis about a population variance . . . . .   | 54 |
| Exa 9.5    | Z value for Type II error . . . . .   | 55 |
| Exa 9.6    | Z value for Type II error . . . . .   | 56 |
| Exa 10.1   | Z formula for the difference in Two Sample Means . .  | 58 |
| Exa 10.2   | Confidence Interval to estimate difference in two popu-<br>lation means . . . . .   | 59 |
| Exa 10.3   | t formula to test the difference in means assuming the<br>standard deviations are equal . . . . .                           | 60 |
| Exa 10.4   | CONFIDENCE INTERVAL TO ESTIMATE difference<br>in means ASSUMING THE POPULATION VARIANCES<br>ARE UNKNOWN AND EQUAL . . . . . | 61 |
| Exa 10.5   | t formula to test the Difference in Two Dependent Pop-<br>ulation . . . . .   | 62 |
| Exa 10.6   | Z formula to test the difference in Population Proportion   | 63 |
| Exa 10.7   | F test for two Population Variance . . . . .  | 63 |
| Exa 11.1   | One Way ANOVA . . . . .   | 65 |
| Exa 11.2   | TUKEYs HSD Test . . . . .   | 66 |
| Exa 11.3   | Randomized Block Design . . . . .   | 67 |
| Exa 11.4   | Two Way ANOVA . . . . .   | 68 |
| Exa 12.1   | Slope of Regression line . . . . .  | 70 |
| Exa 12.2   | Residual Analysis . . . . .   | 70 |
| Exa 12.3   | Standard Error of Estimation . . . . .  | 72 |
| Exa 12.4   | Coefficient of Determination . . . . .  | 73 |
| Exa 12.5   | t test for slope . . . . .  | 75 |
| Exa 12.6   | CONFIDENCE INTERVAL TO ESTIMATE THE SIN-<br>GAL VALUE FOR A GIVEN VALUE OF x . . . . .                                      | 75 |
| Exa 12.7   | Regression Analysis Example . . . . .   | 76 |
| Exa 13.1   | Multiple Regression Model . . . . .   | 78 |
| Exa 13.2   | Multiple Regression Analysis Model . . . . .  | 79 |
| Exa 14.1   | Model Transformation . . . . .  | 81 |
| Exa 15.1.a | Moving average . . . . .  | 83 |
| Exa 15.1.b | Moving average . . . . .  | 84 |
| Exa 15.2   | Weighted Moving Average . . . . .   | 87 |
| Exa 15.3   | EXPONENTIAL SMOOTHING . . . . .   | 88 |
| Exa 15.4   | Regression Trend Analysis Using Quadratic Models . .  | 89 |
| Exa 15.5   | LASPEYRES PRICE INDEX and PAASCHE PRICE<br>INDEX . . . . .  | 90 |
| Exa 16.1   | CHI SQUARE GOODNESS OF FIT TEST . . . . .   | 93 |



|          |  |     |
|----------|--|-----|
| Exa 16.2 | Test data is whether in Poisson distributed . . . . .  | 94  |
| Exa 16.3 | CHI SQUARE GOODNESS OF FIT TEST example 2              | 95  |
| Exa 16.4 | CHI SQUARE TEST OF INDEPENDENCE . . . . .              | 96  |
| Exa 17.1 | Mann Whitney U test . . . . .                          | 97  |
| Exa 17.2 | LARGE SAMPLE FORMULAS MANN WHITNEY U<br>TEST . . . . . | 98  |
| Exa 17.3 | WILCOXON MATCHED PAIRS SIGNED RANK TEST                | 99  |
| Exa 17.4 | KRUSKAL WALLIS TEST . . . . .                          | 101 |
| Exa 17.5 | FRIEDMAN TEST . . . . .                                | 102 |
| Exa 17.6 | SPEARMANS RANK CORRELATION . . . . .                   | 102 |

# Chapter 2

## Charts and Graphs

R code Exa 2.1.a Class Midpoints

```
1 # Class midpoints.
2
3 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,
4
  7.30,7.24,7.16,7.03,6.90,7.16,7.40,7.05,7.28,7.31
5
  7.03,7.17,6.78,7.08,7.12,7.31,7.40,6.35,6.96,7.29
6
  6.96,7.02,7.13,6.84)
7
8 summary(Interest_rate)
9
10 low_val<- 6.30
11 high_val <-7.70
12 step_val <- 0.20
13 x_breaks <- seq(low_val,high_val,step_val)
14 x_breaks
15 x_mid <- seq(low_val+step_val/2,high_val-step_val/2,
  step_val)
16 x_mid
```

```

17 x<-cut(Interest_rate,breaks = x_breaks,right=FALSE)
18 x
19 y<-table(x)
20 y
21
22 df <- data.frame(y)
23 df
24
25 # Class Mid point :
26 df$midpoint <- x_mid
27 View(df)

```

---

### R code Exa 2.1.b Relative Frequency

```

1 # Relative Frequency.
2
3 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,
4
      7.30,7.24,7.16,7.03,6.90,7.16,7.40,7.05,7.28,7.31
5
      7.03,7.17,6.78,7.08,7.12,7.31,7.40,6.35,6.96,7.29
6
      6.96,7.02,7.13,6.84)
7
8 summary(Interest_rate)
9
10 low_val<- 6.30
11 high_val <-7.70
12 step_val <- 0.20
13 x_breaks <- seq(low_val,high_val,step_val)
14 x_breaks
15 x_mid <- seq(low_val+step_val/2,high_val-step_val/2,
      step_val)
16 x_mid

```

```

17 x<-cut(Interest_rate,breaks = x_breaks,right=FALSE)
18 x
19 y<-table(x)
20 y
21
22 df <- data.frame(y)
23 df
24
25 # Class Mid point :
26 df$midpoint <- x_mid
27 df
28
29 # Relative Frequency :
30 rf <- df$Freq/sum(df$Freq)
31 rf
32 df$relative_frequency <- rf
33 View(df)

```

---

### R code Exa 2.1.c Cumulative Frequency

```

1 # Cumulative Frequency.
2
3 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,
4
5      7.30,7.24,7.16,7.03,6.90,7.16,7.40,7.05,7.28,7.31
6
7      7.03,7.17,6.78,7.08,7.12,7.31,7.40,6.35,6.96,7.29
8
9      6.96,7.02,7.13,6.84)
8 summary(Interest_rate)
9
10 low_val<- 6.30
11 high_val <-7.70

```

```

12 step_val <- 0.20
13 x_breaks <- seq(low_val,high_val,step_val)
14 x_breaks
15 x_mid <- seq(low_val+step_val/2,high_val-step_val/2,
               step_val)
16 x_mid
17 x<-cut(Interest_rate,breaks = x_breaks,right=FALSE)
18 x
19 y<-table(x)
20 y
21
22 df <- data.frame(y)
23 df
24
25 # Class Mid point :
26 df$midpoint <- x_mid
27 df
28
29 # Relative Frequency :
30 rf <- df$Freq/sum(df$Freq)
31 rf
32 df$relative_frequency <- rf
33 View(df)
34
35 # Cumulative Frequency :
36 c<-cumsum(df$Freq)
37 df$cumulative_frequency <- c
38 n <- sum(df$Freq)
39 crf <- c/n
40 df$cumul <- crf
41 df$pie <- round(360*rf,1)
42 View(df)

```

---

**R code Exa 2.2** Steam and leaf plot

```

1 # Steam-and-leaf plot
2
3 costs <- c
      (3.67,2.75,9.15,5.11,3.32,2.09,1.83,10.94,1.93,3.89,
4
      7.20,2.78,6.72,7.80,5.47,4.15,3.55,3.53,3.34,4.95,
5
      5.42,8.64,4.84,4.10,5.10,6.45,4.65,1.97,2.84,3.21
      )
6
7
8 stem(costs, scale = 1, width = 80, atom = 1e-08)

```

---

### R code Exa 2.3.a Bar Graph

```

1 # Bar Graph :
2
3 Inventory_shrinkage <- c("Employee theft","
      Shoplifting","Administartive error","Vendor fraud
      ")
4
5 Annual_amount <- c(17918.6, 15191.9,7617.6,2553.6)
6
7 data <- data.frame(Inventory_shrinkage,Annual_amount
      )
8
9 Proportion <- data$Annual_amount/sum(data$Annual_
      amount)
10
11 Percent <- Proportion*100
12
13 data <- cbind(data,Proportion,Percent)
14
15 Degree <- data$Proportion*360
16

```

```

17 data<-cbind(data,Degree)
18
19 library(ggplot2)
20
21 ggplot(data,aes(x=data$Inventory_shrinkage,y=data$
    Annual_amount))+geom_bar(stat = "identity")

```

---

### R code Exa 2.3.b Bar Graph

```

1 # Pie Chart :
2
3 Inventory_shrinkage <- c("Employee theft","
    Shoplifting","Administartive error","Vendor fraud
    ")
4
5 Annual_amount <- c(17918.6, 15191.9,7617.6,2553.6)
6
7 data <- data.frame(Inventory_shrinkage,Annual_amount
    )
8
9 Proportion <- data$Annual_amount/sum(data$Annual_
    amount)
10
11 Percent <- Proportion*100
12
13 data <- cbind(data,Proportion,Percent)
14
15 Degree <- data$Proportion*360
16
17 data<-cbind(data,Degree)
18
19 labls <- paste(data$Inventory_shrinkage,data$Percent
    ,sep = " ")
20
21 labls <- paste(labls,"%",sep="")

```

```
22
23 pie(data$Percent, labels = labls)
```

---

### R code Exa 2.4 Scatter Plot

```
1 # Scatter Plot :
2 Residential <- c
   (169635,155113,149410,175822,162706,134605,195028,231396,234955,
3
   266481,267063,263385,252745,228943,197526,232134,24
4
   251937,281229,280748,297886,315757)
5
6 Non_residential <- c
   (96497,115372,96407,129275,140569,145054,131289,155261,178925,
7
   163740,160363,164191,169173,167896,135389,12092
8
   139711,153866,166754,177639,175048)
9
10 home <- cbind(Residential,Non_residential)
11 View(home)
12
13 # Scatter plot :
14 plot(Residential, Non_residential,xlab=" Residential"
   ,ylab="Non-Residential")
```

---



# Chapter 3

## Descriptive Statistics

R code Exa 3.1.a Mode

```
1 # Mode Example :
2
3 getmode <- function(v) {
4   uniqv <- unique(v)
5   uniqv[which.max(tabulate(match(v, uniqv)))]
6 }
7
8 Company <- c("Enterprise", "Hertz", "Natioanl/Alamo", "
9             "Avis", "Dollar", "Budget", "Advantage",
10             "U-save", "Payless", "ACE", "Fox", "Rent-A-
11             Wreck", "Traingle")
12
13 Number_of_Cars_in_Service <- c
14   (643000, 327000, 233000, 204000, 167000, 144000, 20000, 12000, 10000,
15   9000, 9000, 7000, 6000)
16
17 data1 <- data.frame(Company, Number_of_Cars_in_
18   Service)
19
20 sort_data <- data1[order(-Number_of_Cars_in_Service
```

```

    ),]
17
18 result <- getmode(sort_data$Number_of_Cars_in_
    Service)
19 print(result)

```

---

### R code Exa 3.1.b Median

```

1 # Median :
2
3 Company <- c("Enterprise", "Hertz", "Natioanl/Alamo", "
    Avis", "Dollar", "Budget", "Advantage",
4             "U-save", "Payless", "ACE", "Fox", "Rent-A-
    Wreck", "Traingle")
5
6 Number_of_Cars_in_Service <- c
    (643000, 327000, 233000, 204000, 167000, 144000, 20000, 12000, 10000,
7
8             9000, 9000, 7000, 6000)
9
10
11 data1 <- data.frame(Company, Number_of_Cars_in_
    Service)
12
13 sort_data <- data1[order(-Number_of_Cars_in_Service
    ),]
14
15
16
17
18
19 median(sort_data$Number_of_Cars_in_Service)

```

---

### R code Exa 3.1.c Mean

```

1 # Mean Example :
2

```

```

3 Company <- c("Enterprise","Hertz","Natioanl/Alamo","
  Avis", "Dollar", "Budget","Advantage",
4           "U-save","Payless","ACE","Fox","Rent-A-
  Wreck","Traingle")
5
6 Number_of_Cars_in_Service <- c
  (643000,327000,233000,204000,167000,144000,20000,12000,10000,
7           9000,9000,7000,6000)
8
9 data1 <- data.frame(Company,Number_of_Cars_in_
  Service)
10
11 sort_data <- data1[order(-Number_of_Cars_in_Service
  ),]
12
13 mean(sort_data$Number_of_Cars_in_Service)

```

---

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

```

1 # Determine the 30th percentile of the following
  eight numbers :
2 data3 <- c(5,12,13,14,17,19,23,28)
3 N = 8
4 P = 30
5
6 # 30th percentile value is :
7 a <- quantile(data3,c(.30))
8 cat("30th percentile value is : ",a)

```

---

**R code Exa 3.3** Quartiles

```

1 # Quartiles :
2
3 Category <- c("Automotive","Personal Care","
      Entertainment & Media",
4              "Food","Drugs", "Electronics","Soft
              Drinks","Retail","Cleaners",
5              "Restaurants","Computers","Telephone",
              "Financial",
6              "Beer Wine & Liquor","Candy","Toys")
7
8 Ad_spending <- c
      (22195,19526,9538,7793,7707,4023,3916,3576,3571,3553,3247,2488,
9
10              2433,2050,1137,699)
11
12 advertise_age <- cbind(Category,Ad_spending)
13 View(advertise_age)
14
15 N=16
16
17 # Q1 = P25 is found by :
18 i = 25/100*N
19 i
20
21 #Q3 = P75 is solved by :
22 i1 =75/100*(N)
23 i1
24
25 # Quantile :
26 quantile(Ad_spending)

```

---

### R code Exa 3.5 Chebyshevs Theorem

```

1 # Chebyshev 's Theorem :
2

```

```

3 avg_age = 28
4 sd = 6
5
6 # Chebyshev's theorem states that at least  $(1 - 1/k^2)$  proportion of the values are within
7  $\#(mean + k * sd)$ . Because 80% of the values are within
   this range, let
8
9  $\#1 - (1/k^2) = .80$ 
10
11 k = sqrt(1/(1-0.80))
12 k
13
14 # now for :
15 mean = 28
16 sd = 6
17
18 # values are within
19 r1 = mean + k * sd
20 r1 #41.41
21 r2 = mean - k * sd
22 r2 # 14.58
23
24 # Years of age or between 14.6 and 41.4 years old.

```

---

### R code Exa 3.6.a Mean Absolute Deviation

```

1 # Mean absolute deviation :
2
3 x <- c(55, 100, 125, 140, 60)
4 n = 5
5
6 # a = abs(x - x_bar), where x_bar = sum(x)/n
7 a <- c(41, 4, 29, 44, 36)
8

```

```

9 x <- cbind(x,a)
10 View(x)
11
12 # MAD :
13 mean_dev <- sum(a)/n
14 mean_dev

```

---

### R code Exa 3.6.b Variance and Standard deviation

```

1 # Variance and stanadard deviation :
2
3 x<- c(55,100,125,140,60)
4 n = 5
5
6 # a = abs(x - x_bar), where x_bar = sum(x)/n
7 a <- c(41,4,29,44,36)
8
9 # b = (x - x_bar)^2
10 b <- c(1681,16,841,1936,1296)
11
12 y <- cbind(x,a,b)
13 View(y)
14
15 # Variance :
16 var(x)
17
18 # standard deviation :
19 sd(x)

```

---

### R code Exa 3.7 Mean Median Mode Variance and Standard deviation

```

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

```

```

2
3 class <- c("10-under-15", "15-under-20", "20-under-25"
4           , "25-under-30", "30-under-35",
5           "35-under-40", "40-under-45", "45-under-50"
6           )
7 freq <- c(6,22,35,29,16,8,4,2)
8 class <- data.frame(class, freq)
9 class
10
11 # Mean of each intervals :
12 a <- mean(10:15)
13 b<-mean(15:20)
14 c<-mean(20:25)
15 d<-mean(25:30)
16 e<-mean(30:35)
17 f<-mean(35:40)
18 g<-mean(40:45)
19 h<-mean(45:50)
20 Mean <- rbind(a,b,c,d,e,f,g,h)
21 Mean
22
23 # fM :
24 for(i in 1:8)
25 {
26   fM <- freq * Mean
27 }
28 fM
29
30 # group mean :
31 Group_mean <- sum(fM)/sum(freq)
32 Group_mean
33
34 # Mean - group mean :
35 for(i in 1:8)
36 {
37   Mean_grpmean <- Mean - Group_mean
38 }
39 Mean_grpmean

```

```
38
39 # Square of Mean_grpmean :
40 Mean_grpmean_sq <- Mean_grpmean^2
41 Mean_grpmean_sq
42
43 # freq * Mean_grpmean_sq :
44 freq_Mean_grpmean_sq <- freq * Mean_grpmean_sq
45 freq_Mean_grpmean_sq
46
47
48 var <- sum(freq_Mean_grpmean_sq)/(sum(freq)-1)
49 var
50 sd <- sqrt(var)
51 sd
```

---



# Chapter 4

## Probability

R code Exa 4.1 Addition Law

```
1 # Addition Law :  $P(F \text{ and } P) = P(F) + P(P) - P(F \text{ or } P)$ 
  )
2
3 Type_of_position <- c("Managerial", "Professional", "
  Technical", "Clerical")
4 Sex_male <- c(8,31,52,9)
5 Sex_female <- c(3,13,17,22)
6 total_r <- c(11,44,69,31)
7 total_c <- c(" ",100,55,55)
8 Compny_HR_data <- cbind(Type_of_position,Sex_male,
  Sex_female,total_r)
9 Compny_HR_data <- rbind(Compny_HR_data,total_c)
10 View(Compny_HR_data)
11
12 # F denote the event of female and P denote the
  event of professional worker
13
14 # Probability of event of female :
15 Pb_F = sum(Sex_female)/sum(sum(Sex_female),sum(Sex_
  male))
16 Pb_F
```

```

17
18 # Probability of event of professional worker :
19 Pb_P = sum(Sex_male[2],Sex_female[2])/sum(sum(Sex_
      female),sum(Sex_male))
20 Pb_P
21
22 # Probability of female or Professional worker :
23 Pb_F_P = Sex_female[2]/sum(sum(Sex_female),sum(Sex_
      male))
24 Pb_F_P
25
26 # probability that the employee is female or a
      professional worker :
27 Pb_F_a_P <- Pb_F + Pb_P - Pb_F_P
28 Pb_F_a_P

```

---

#### R code Exa 4.3 Special Law of Addition

```

1 # Special Law of Addition :  $P(T \text{ and } C) = P(T) + P(C)$ 
2
3 Type_of_position <- c("Managerial", "Professional", "
      Technical", "Clerical")
4 Sex_male <- c(8,31,52,9)
5 Sex_female <- c(3,13,17,22)
6 total_r <- c(11,44,69,31)
7 total_c <- c(" ",100,55,55)
8 Compny_HR_data <- cbind(Type_of_position,Sex_male,
      Sex_female,total_r)
9 Compny_HR_data <- rbind(Compny_HR_data,total_c)
10 View(Compny_HR_data)
11
12 # T denote technical, C denote clerical, and P
      denote professional.
13

```

```

14 # Probability of Technical position :
15 Pb_T = sum(Sex_male[3],Sex_female[3])/sum(sum(Sex_
      female),sum(Sex_male))
16 Pb_T
17
18 # Probability of Clerical position :
19 Pb_C = sum(Sex_male[4],Sex_female[4])/sum(sum(Sex_
      female),sum(Sex_male))
20 Pb_C
21
22 # Probability of professional position :
23 Pb_P = sum(Sex_male[2],Sex_female[2])/sum(sum(Sex_
      female),sum(Sex_male))
24 Pb_P
25
26 # probability that a worker is either technical or
      clerical is :
27 Pb_T_C = Pb_T + Pb_C
28 Pb_T_C
29
30 # probability that a worker is either professional
      or clerical is :
31 Pb_P_C = Pb_P + Pb_C
32 Pb_P_C

```

---

#### R code Exa 4.5 Multiplication Law

```

1 # General Law of Multiplication :  $P(X \text{ or } Y) = P(X) * P(Y|X)$ 
       $= P(Y) * P(X|Y)$ 
2
3 Total_emp = 140
4 supervisor = 30
5 Married_emp = 80
6 Pb_S_M = .20 #  $P(S|M)$  i.e. married employees are
      supervisors

```

```

7
8 # probability that the employee is married :
9 Pb_M = Married_emp/Total_emp
10 Pb_M
11
12 # probability that the employee is married and is a
    supervisor :
13 Pb_M_s <- Pb_M * Pb_S_M
14 Pb_M_s
15
16 # 11.43% of the 140 employees are married and are
    supervisors

```

---

#### R code Exa 4.6 General Law of Multiplication

```

1 # General Law of Multiplication :
2
3 Industry_type <- c("Finance_A", "Manufacturing_B", "
    Communication_C")
4 Northeast_D <- c(.12,.15,.14)
5 Southeast_E <- c(.05,.03,.09)
6 Midwest_F <- c(.04,.11,.06)
7 West_G <- c(.07,.06,.08)
8 total_r <- c(.28,.35,.37)
9 total_c <- c(" ",.41,.17,.21,.21,1.00)
10 Industry_type <- cbind(Industry_type,Northeast_D,
    Southeast_E,Midwest_F,West_G,total_r)
11 Industry_type <- rbind(Industry_type,total_c)
12 View(Industry_type)
13
14 # a.) P(Manufacturing_B and Southeast_E) :
15 P_B_E <- total_r[2]*(Southeast_E[2]/total_r[2])
16 P_B_E
17
18 # b.) P(West_G and Finance_A) :

```

```

19 P_G_A <- sum(Midwest_F) *(West_G[1]/sum(Midwest_F))
20 P_G_A
21
22 # c.) P(Manufacturing_B and Communication_C) :
23 P_B_C <- .0
24 P_B_C # The matrix shows no intersection for these
        two events.
25 # Thus B and C are mutually exclusive.

```

---

#### R code Exa 4.8 Special Law of Multiplication

```

1 # Special law of Mulyiplication : If X, Y are
  independent ,  $P(X \text{ or } Y) = P(X) * P(Y)$ 
2
3
4 T1 <- c("A", "B", "C")
5 D <- c(8,20,6)
6 E <- c(12,30,9)
7 total_r <- c(20,50,15)
8 total_c <- c(" ",34,51,85)
9 T1 <- cbind(T1,D,E,total_r)
10 T1 <- rbind(T1,total_c)
11 View(T1)
12
13 # Probability of B :
14 Pb_B = sum(D[2],E[2])/sum(total_r)
15 Pb_B
16
17 # Probability of D :
18 Pb_D = sum(D)/sum(total_r)
19 Pb_D
20
21 # Probability of B and D is :
22 Pb_B_D = Pb_B * Pb_D
23 Pb_B_D

```

---

**R code Exa 4.9** Conditional Probability

```
1 # Conditinal Probability :  $P(X|Y) = P(X \text{ or } Y)/P(Y)$   
  =  $(P(X)*P(Y|X))/P(Y)$   
2  
3 Industry_type <- c("Finance_A", "Manufacturing_B", "  
  Communication_C")  
4 Northeast_D <- c(.12,.15,.14)  
5 Southeast_E <- c(.05,.03,.09)  
6 Midwest_F <- c(.04,.11,.06)  
7 West_G <- c(.07,.06,.08)  
8 total_r <- c(.28,.35,.37)  
9 total_c <- c(" ",.41,.17,.21,.21,1.00)  
10 Industry_type <- cbind(Industry_type,Northeast_D,  
  Southeast_E,Midwest_F,West_G,total_r)  
11 Industry_type <- rbind(Industry_type,total_c)  
12 View(Industry_type)  
13  
14 #a.)  $P(\text{Manufacturing}_B \mid \text{Midwest}_F) = P(\text{Manufacturing}_B \text{ and } \text{Midwest}_F)/P(\text{Midwest}_F)$   
15 Pb_B_F = Midwest_F[2]/sum(Midwest_F)  
16 Pb_B_F  
17  
18 #b.)  $P(\text{West}_G \mid \text{Communication}_C) = P(\text{West}_G \text{ and } \text{Communication}_C)/P(\text{Communication}_C)$   
19 Pb_G_C = West_G[3]/sum(Northeast_D[3],Southeast_E  
  [3],Midwest_F[3],West_G[3])  
20 Pb_G_C  
21  
22 #c.)  $P(\text{Northeast}_D \mid \text{Midwest}_F) = P(\text{Northeast}_D \text{ and } \text{Midwest}_F)/P(\text{Midwest}_F)$   
23 Pb_D_F = .00/sum(Midwest_F)  
24 Pb_D_F
```

---

### R code Exa 4.11 Independent Event

```
1 # Independent Event :  $P(X|Y) = P(X)$  and  $P(Y|X) = P(Y)$ 
2
3 T1 <- c("A", "B", "C")
4 D <- c(8, 20, 6)
5 E <- c(12, 30, 9)
6 total_r <- c(20, 50, 15)
7 total_c <- c(" ", 34, 51, 85)
8 T1 <- cbind(T1, D, E, total_r)
9 T1 <- rbind(T1, total_c)
10 View(T1)
11
12 # Check the first cell in the matrix to find
    whether  $P(A|D) = P(A)$ 
13 Pb_A_D <- D[1]/sum(D) #  $P(A|D)$ 
14 Pb_A_D
15
16 P_A <- sum(D[1], E[1])/sum(total_r)
17 P_A #  $P(A)$ 
```

---

### R code Exa 4.12 Bayes Rule

```
1 # Bayes's Rule :  $P(X_i|Y) = \frac{P(X_i)*P(Y|X_i)}{P(X_1)*P(Y|X_1)+P(X_2)*P(Y|X_2)+\dots+P(X_n)*P(Y|X_n)}$ 
2
3 Event <- c("A", "B", "C")
4 Prior <- c(.60, .30, .10) #  $P(E_i)$ 
5 Conditional <- c(.40, .50, .70) #  $P(x|E_i)$ 
6 Joint <- c(.24, .15, .07) #  $P(X \text{ and } E_i) = P(E_i)*P(x|E_i)$ 
```

```

7 Posterior <- c(.52,.33,.15) # P(X and Ei)/sum(P(X
  and Ei))
8
9 machine <- cbind(Event,Prior,Conditional,Joint,
  Posterior)
10 machine
11
12 # Revised Probabilities :
13 machine_A <- Prior[1]* Conditional[1]/sum(Joint)
14 machine_A
15
16 machine_B <- Prior[2]* Conditional[2]/sum(Joint)
17 machine_B
18
19 machine_C <- Prior[3]* Conditional[3]/sum(Joint)
20 machine_C

```

---



# Chapter 5

## Discrete Distributions

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

```
1 # Variance and standard deviation of a Discrete
  Distribution :
2
3 Prize <- c(1000,100,20,10,4,2,1,0) # x
4 Probability <- c
  (.00002,.00063,.00400,.00601,.02403,.08877,.10479,.77175)
  # P(x)
5
6 # x * P(x) :
7 for(i in 1:8){
8   x_Pb <- Prize*Probability # x * P(x)
9 }
10 print(x_Pb)
11
12 # sum Of x * P(x) :
13 x_Pb_s <- sum(x_Pb)
14 x_Pb_s
15
16
17 # (x - x_Pb_s)^2
18 for(j in 1:8){
```

```

19   x_mean_sq <- (Prize - x_Pb_s)^2
20 }
21 print(x_mean_sq)
22
23
24 # (x - x_Pb_s)^2 * P(x) :
25 for(j in 1:8){
26   x_mean_sq_Pb <- (Prize - x_Pb_s)^2 * Probability
27 }
28 print(x_mean_sq_Pb)
29
30 # sum of (x - x_Pb_s)^2 * P(x) :
31 x_mean_sq_Pb_s <- sum(x_mean_sq_Pb)
32 x_mean_sq_Pb_s
33
34 Prize <- cbind(Prize,Probability,x_mean_sq,x_mean_sq
35               _Pb)
36 View(Prize)
37
38 # Variance and Standard deviation :
39 var <- x_mean_sq_Pb_s
40 sd <- sqrt(var)
41 sd

```

---

### R code Exa 5.2 Binomial Distribution

```

1 # Binomial Distribution :  $P(x) = nC_x * p^x * q^{n-x} = \frac{n!}{x!(n-x)!} * p^x * q^{n-x}$ 
2
3 p = .65
4 q = 1-p
5 n = 25
6 x = 19
7 x1 = 0:19

```

```

8
9 # Binomial Distribution through inbuild function in
   r :
10 bd <- dbinom(x,n,p)
11 bd
12
13 # Binomial Distribution through formula :
14 bd <- (factorial(n)/(factorial(x)*factorial(n-x))) *
      (p^x) * (q^(n-x))
15 bd

```

---

### R code Exa 5.3 Binomial Distribution ex 2

```

1 # Binomial Distribution ex 2 :
2
3 p = .06
4 q = .94
5 n = 20
6
7 x <- c(0,1,2)
8 c<-choose(n,x)*(p^x) * (q^(n-x))
9 c
10 sum(c)

```

---

### R code Exa 5.5 Using Binomial Table

```

1 # using Binomial Table :
2
3 n = 20
4 p = .10
5 q = 1-p
6
7 x <- c(0,1,2,3)

```

```

8 c<-choose(n,x)*(p^x) * (q^(n-x))
9 c
10
11 # Probability that fewer than four purchasers
    choose Oreos i.e. x<4 :
12 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

```

1 # Mean and standard deviation in Binomial
    distribution :
2 # mean = n * p   and sd = sqrt(n*p*q)
3
4 n = 10
5 p<-c(.10,.20,.30,.40)
6 q = 1-p
7
8 # mean <- n*p
9 for(p1 in 1:4){
10     mean = n*p
11 }
12 print(mean)
13
14 pd<-pbinom(2,n,p)
15
16
17 p<-cbind(p,mean,pd)
18 p

```

---

**R code Exa 5.7** Poissons formula

```

1 # Poission formula :  $P(x) = \text{lamda}^x * e^{-\text{lamda}} / x!$ 

```

```

2
3 l <- 3.2 # lamda
4 # x>7 customers/4 minutes
5
6 # through in build function of poission in r:
7 dpois(8,lambda = 3.2) # x=8
8
9 # x = 8 through formula :
10 x = 8
11 pd_8 <- (1^x*exp(-l))/factorial(x)
12 pd_8
13
14 # x = 9 through formula :
15 x = 9
16 pd_9 <- (1^x*exp(-l))/factorial(x)
17 pd_9
18
19 # x = 10 through formula :
20 x = 10
21 pd_10 <- (1^x*exp(-l))/factorial(x)
22 pd_10
23
24 # x = 11 through formula :
25 x = 11
26 pd_11 <- (1^x*exp(-l))/factorial(x)
27 pd_11
28
29 # x = 12 through formula :
30 x = 12
31 pd_12 <- (1^x*exp(-l))/factorial(x)
32 pd_12
33
34 # x = 13 through formula :
35 x = 13
36 pd_13 <- (1^x*exp(-l))/factorial(x)
37 pd_13
38
39 # Poission distribution for x>=8

```

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

---

### R code Exa 5.8 Poisson distribution Example

```
1 # Poisson distribution Example :
2 # Poisson formula :  $P(x) = \text{lamda}^x * e^{-\text{lamda}} / x!$ 
3
4 l=3.2
5 x = 10
6 pd <- dpois(x,l,log=FALSE)
7 pd
8
9 # probability of getting exactly 10 customers during
   an 8-minute interval
10 l1=6.4
11 x1 = 10
12 pd1 <- dpois(x1,l1,log=FALSE)
13 pd1
```

---

### R code Exa 5.9 Using poissions table

```
1 # using poisson table :
2
3 l <- 1.6
4 x <- c(6,7,8,9)
5
6
7 # Poisson probability for  $x > 5$  :
8 p <- dpois(x,l)
9 p
10 sum(p)
```

---

### R code Exa 5.10 Probability Example

```
1 # Probability Example :
2
3 p = .0003
4 n= 10000
5 l <- n*p
6 l
7 x<- c(7,8,9,10,11,12)
8
9 # Binomial probability for x>5 :
10 b<-dbinom(x,n,p)
11 b
12 sum(b)
13
14
15 # Poisson probability for x>5 :
16 p<-dpois(x,l)
17 p
18 sum(p)
```

---

### R code Exa 5.11 Hypergeometrics distribution

```
1 # Hypergeometric distribution :  $P(x) = \frac{A C_x (N-A) C(n-x)}{N C_n}$ 
2
3 # 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
4
5 N = 18
```

```
6 n = 3
7 A = 12
8
9 # Using choose function :
10
11 1 - ((choose(A,0)*choose((N-A),n))/choose(N,n))
```

---



# Chapter 6

## Continuous Distributions

**R code Exa 6.1** Uniform Distribution

```
1 # Probabilities in Uniform Distribution :  $P(x) = \frac{x_2 - x_1}{b - a}$  where:  $a \leq x_1 \leq x_2 \leq b$ 
2
3 b = 39
4 a = 27
5
6 f_x = 1 / (b - a) # f(x)
7 f_x
8
9 u <- (a + b) / 2 #mean
10 u
11
12 sd <- (b - a) / sqrt(b - a) # standard deviation
13 sd
14
15 #  $P(30 \leq x \leq 35)$  :
16 P = (35 - 30) / (39 - 27)
17 P
18
19 #  $P(x < 30)$  :
20 P1 = (30 - 27) / (39 - 27)
```

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

```
1 # MEAN AND STANDARD DEVIATION OF A UNIFORM
  DISTRIBUTION :
2
3 u = 691 # mean
4 a = 200
5 b = 1182
6 x1 = 410
7 x2 = 825
8 sd <- (b-a)/sqrt(12) # standard deviation
9 sd
10
11 # height of distribution :
12 f_x = 1/(b-a) # f(x)
13 f_x
14
15 # probability that a randomly selected person pays
  between $410 and $825 annually for automobile
  insurance in the US:
16 p_x = (x2-x1)/(b-a)
17 p_x
```

---

**R code Exa 6.3** Normal Curve distribution

```
1 # Normal Curve distribution :
2
3 mean = 494
4 sd=100
5 x =700
```

```
6
7 # probability of x greater than 700 :
8 pnorm(x, mean, sd, lower.tail=FALSE)
```

---

#### **R code Exa 6.4** PROBABILITY OF A UNIFORM DISTRIBUTION

```
1 # PROBABILITY OF A UNIFORM DISTRIBUTION
2
3 x = 550
4 mean = 494
5 sd = 100
6 lb = .2123 # probability of values between 550 and
              the mean
7 ub = .5000 # probability of values less than the
              mean
8
9
10 # using r function :
11 pnorm(x, mean, sd)
12
13 # Or using normal formula :
14 z = (x - mean) / sd
15 z
16
17 ub + lb # probability of values 550
```

---

#### **R code Exa 6.5** Probability of Normal Curve DISTRIBUTION

```
1 # Probability of Normal Curve DISTRIBUTION :
2
3 x = 600
4 mean = 494
5 sd = 100
```

```

6 x1 = 300
7
8 a <- pnorm(x1, mean, sd, lower.tail=FALSE)
9 a
10 b <- pnorm(x, mean, sd, lower.tail=FALSE)
11 b
12
13 # probability of a value between 300 and 600 :
14 a - b

```

---

#### R code Exa 6.6 PROBABILITY OF A UNIFORM DISTRIBUTION

```

1 # PROBABILITY OF A UNIFORM DISTRIBUTION
2
3 x = 350
4 mean = 494
5 sd = 100
6 x1 = 450
7
8 a <- pnorm(x, mean, sd, lower.tail=FALSE)
9 a
10 b <- pnorm(x1, mean, sd, lower.tail=FALSE)
11 b
12
13 # probability of a value between 350 and 450 :
14 a-b

```

---

#### R code Exa 6.7 MEAN OF A UNIFORM DISTRIBUTION

```

1 # MEAN OF A UNIFORM DISTRIBUTION
2
3 x = 449
4 z = 1.11 # value taken from z table

```

```

5 sd = 36
6 # z = (x - mean)/sd
7
8 mean = x - (z*sd)
9 mean

```

---

**R code Exa 6.8** Normal distribution using z value

```

1 # Normal distribution using z value :
2
3 mean = 3.58
4 z = -0.46 # value taken from z table
5 sd = 1.04
6 # z = (x - mean)/sd
7
8 x = (z*sd) + mean
9 x
10
11 # 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

```

1 # binomial distribution problem by using the normal
    distribution :
2
3 x = 12
4 n = 25
5 p = .40
6 q = 1-p
7
8 mean = n * p

```

```

9  mean
10
11 sd = sqrt(n*p*q)
12 sd
13
14 # test : mean +/- 3sd
15 test1 <- mean + 3*sd
16 test2 <- mean - 3*sd
17 test1
18 test2
19
20 # test : 2.65 to 17.35
21
22 # z value at x = 12.5
23 x = 12.5
24 z = (x-mean)/sd
25 z
26
27 # z value at x = 12.5
28 x = 11.5
29 z = (x-mean)/sd
30 z
31
32 #z = 1.02 produces a probability of .3461.
33 # z = 0.61 produces a probability of .2291.
34
35 # The difference in areas yields the following
   answer :
36 0.3461 - .2291

```

---

**R code Exa 6.10** Binomial distribution by using the normal distribution

```

1 # Binomial distribution by using the normal
   distribution :
2

```

```

3 p = .37
4 n = 100
5 q=1-p
6 mean1 = n*p
7 mean1
8 sd = sqrt(n*p*q)
9 sd
10
11 # range :
12 u = mean +3*(sd)
13 u
14 l = mean - 3*(sd)
15 l
16
17 x = 26.5
18 z=(x-mean)/sd
19 z
20
21 # tail of the distribution :
22 .5000-.4850
23
24 x1 <- c(26:20)
25 b<-dbinom(x1,n,p)
26 b
27 sum(b)

```

---

#### R code Exa 6.11 Exponential Distribution

```

1 # Exponential Distribution :  $f(x) = \lambda * e^{-\lambda * x}$ 
2
3 # Probability of right tail exponential distribution
  :  $P(x \geq x_0) = e^{-\lambda * x_0}$ 
4
5 l = 1.38 # lambda

```

```
6 mean = 1/1
7 mean
8 x0 = .75
9
10 # P(x>=x0) :
11 P <- exp(-1*x0)
12 P
13
14 # for x0 = 0.75, Probability < x0 :
15 Prob = 1-P
16 Prob
```

---



# Chapter 7

## Sampling and Sampling Distributions

R code Exa 7.1 Z formula for sample means

```
1 # Z formula for sample means :  $z = (\text{sample\_mean} -$   
     $\text{average}) / (\text{standard\_dev} / \text{sqrt}(\text{sample\_size}))$   
2  
3 mean = 448  
4 sd = 21/sqrt(49)  
5 n = 49 # sample size  
6 # sample mean :  $441 \leq \bar{x} \leq 446$   
7 samplemean_l = 441  
8 samplemean_u = 446  
9  
10 a <-pnorm(samplemean_l, mean, sd, lower.tail=FALSE)  
11 a  
12 b <-pnorm(samplemean_u, mean, sd, lower.tail=FALSE)  
13 b  
14  
15  
16 # probability of a value being between  $z = -2.33$   
    and  $-0.67$  is :  
17 prob = a - b
```

```

18 prob
19
20 # The probability of a value being between z=2.33
    and -0.67 is .2426; that is ,
21 # there is a 24.26% chance of randomly selecting 49
    hourly periods for
22 # which the sample mean is between 441 and 446
    shoppers.

```

---

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

```

1 # Z formula for Sample mean of a finite population :
2 #  $z = (\text{samplemean} - \text{population\_mean}) / (\text{sd} / \sqrt{n}) * (\sqrt{(N-n)/(N-1)})$ 
3
4 pop_mean = 37.6 # avg
5 pop_sd = 8.3 # sd
6 n = 45 # sample size
7 N = 360 # finite population
8 sample_mean = 40
9
10 sd = (pop_sd/sqrt(n))*(sqrt((N-n)/(N-1)))
11
12 pnorm(sample_mean, pop_mean, sd, lower.tail=TRUE)

```

---

#### R code Exa 7.3 Z formula for Sample Proportion

```

1 # Z formula for Sample Proportion :
2 #  $z = (\text{sample\_proportion} - \text{population\_prop}) / \sqrt{(\text{population\_prop} * q) / \text{sample size}}$ 
3
4 p = 0.10 # population_prop
5 sample_prop = 12/80

```

```
6 n = 80
7 q = 1-p
8
9 sd = sqrt(p*q/n)
10
11 # P(sample_prop >= .15) :
12 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

```
1 # Confidence interval to Estimate Population mean :
2 # pop_mean +/- z*(sd/sqrt(n))
3
4 n = 44
5 sample_mean = 10.455
6 sd = 7.7
7 z = 1.645
8
9 pop_mean_1 = sample_mean - (z*(sd/sqrt(n)))
10 pop_mean_1
11
12 pop_mean_2 = sample_mean + (z*(sd/sqrt(n)))
13 pop_mean_2
```

---

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

```

1 # Confidence interval to Estimate Population mean
  using finite correction :
2 # (pop_mean) +/- (z*(sd/sqrt(n))*sqrt((N-n)/(N-1)))
3
4 n = 50
5 N = 800
6 sample_mean = 34.30
7 sd = 8
8 z = 2.33
9
10 pop_mean_1 = sample_mean - (z*(sd/sqrt(n))*sqrt((N-n)/(N-1)))
11 pop_mean_1
12
13 pop_mean_2 = sample_mean + (z*(sd/sqrt(n))*sqrt((N-n)/(N-1)))
14 pop_mean_2

```

---

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

```

1 # Confidence Interval to Estimate population mean :
  Population standard deviation unknown and
  population normally distributed
2 # pop_mean +/- t*(sd/sqrt(n)) , df = n-1
3 a<- c(3,1,3,2,5,1,2,1,4,2,1,3,1,1)
4 n = 14
5 df = n-1
6 t = 3.012
7 sd = 1.29
8 sample_mean = 2.14
9
10 pop_mean_1 = sample_mean - (t*(sd/sqrt(n)))
11 pop_mean_1
12

```

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

---

**R code Exa 8.4** Confidence Interval to estimate Population Proportion

```
1 # Confidence Interval to estimate Population
  Proportion :
2 # p = samp_prop +/- (z*sqrt(samp_prop*q/sample size)
3
4 samp_prop = 0.51
5 q = 1-samp_prop
6 z = 1.75
7 n = 210 # sample size
8
9 p_1 = samp_prop - (z*sqrt(samp_prop*q/n))
10 p_1
11
12 p_2 = samp_prop + (z*sqrt(samp_prop*q/n))
13 p_2
```

---

**R code Exa 8.5** Confidence Interval to estimate Population Proportion

```
1 # Confidence Interval to estimate Population
  Proportion :
2 # p = samp_prop +/- (z*sqrt(samp_prop*q/sample size)
3
4 samp_prop = 34/212 # sample size =212 and no. of
  jeans = 34
5 q = 1-samp_prop
6 z = 1.645
7 n = 212 # sample size
8
9 p_1 = samp_prop - (z*sqrt(samp_prop*q/n))
```

```

10 p_1
11
12 p_2 = samp_prop + (z*sqrt(samp_prop*q/n))
13 p_2

```

---

#### R code Exa 8.6 Confidence to estimate the Population Variance

```

1
2 # Confidence to estimate the Population Variance :
3 # var = ((n-1)*s^2)/(X(a/2))^2 or ((n-1)*s^2)/(X(1-a
  /2))^2 , df = n-1
4
5 s = 1.12
6 n = 25
7 df = n-1
8
9 a = qchisq(0.975, df=24)
10 a
11 b = qchisq(.025, df=24)
12 b
13
14 var_1 = ((n-1)*s^2)/a
15 var_1
16
17 var_2 = ((n-1)*s^2)/b
18 var_2

```

---

#### R code Exa 8.7 Sample Size when Estimating Population mean

```

1 # Sample Size when Estimating Population mean :
2 # n = (z*sd/E)^2
3
4 E = 1 # error in estimating

```

```
5 z = 1.96
6 sd = 5
7
8 n = (z*sd/E)^2
9 n
```

---

**R code Exa 8.8** Sample size when estimating population proportion

```
1 # Sample size when estimating population proportion
  :
2 # n = z^2*p*q/E^2
3
4 E = .03
5 p = .40
6 z = 2.33
7 q = 1-p
8
9 n = z^2*p*q/E^2
10 n
```

---



## Chapter 9

# Statistical Inference Hypothesis Testing for Single Populations

**R code Exa 9.1** Test Hypothesis about population mean

```
1 # Formula to test Hypoythesis about population mean
  :
2 #  $z = \frac{\text{sample\_mean} - \text{pop\_mean}}{\text{sd}/\sqrt{n}}$ 
3
4 pop_mean = 4.30
5 sample_mean = 4.156
6 sd = .574
7 n = 32
8 a = .05 # alpha value
9
10 # Calcultaed value of test statistic :
11 z1 = (sample_mean - pop_mean)/(sd/sqrt(n))
12 z1
13
14 # Critical Z value associated with alpha = 0.05 :
15 z = qnorm(.05, lower.tail=TRUE)
16 z
17
18 # critical sample mean :
```

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

---

**R code Exa 9.2** t test for population mean

```
1 # t test for population mean :
2 # t = (sample_mean - pop_mean) / (sd/sqrt(n)) , df =
   n-1
3
4 pop_mean = 471
5 sample_mean = 498.78
6 sd = 46.94
7 n = 23
8 alpha = 0.05
9 df = n-1
10
11 # t-distribution function to calculate critical t-
   value using alpha and df:
12 qt(alpha, df, lower.tail = FALSE, log.p = FALSE)
13
14 # Observed t value using sample mean and standard
   deviation :
15 t = (sample_mean - pop_mean) / (sd/sqrt(n))
16 t
17
18 # The observed t value of 2.84 is greater than the
   table t value of 1.717,
19 # so the business researcher rejects the null
   hypothesis.
```

---

**R code Exa 9.3** z test of a population proportion

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

```

2 # z = sample_prop - population_prop/sqrt(population_
   prop*q/n)
3
4 n = 550
5 x = 115
6 sample_prop = 115/550
7 population_prop = .17
8 q = 1- population_prop
9
10 # test statistic value of z :
11 z1 = (sample_prop - population_prop)/sqrt((
   population_prop*q)/n)
12 z1
13
14 # critical value of z :
15 z = qnorm(.05,lower.tail=FALSE)
16 z
17
18 # critical sample proportion :
19 sample_prop_c = z * sqrt(population_prop*q/n) +
   population_prop
20 sample_prop_c

```

---

#### **R code Exa 9.4** Test Hypothesis about a population variance

```

1 # Test Hypothesis about a population variance :
2 #  $X^2 = (n-1)*s^2/var$  , df = n-1
3
4 var = 25
5 n = 16
6 s_sq = 28.0625 # sample variance
7 df = n-1
8
9 # Two tailed test and alpha = .10 it will be divided
   by 2 :

```

```

10 a <- .10/2
11
12 # we have two critical values of chi square :
13
14 # 1st chi-sq value is a :
15 qchisq(a, df=15)
16
17 # 2nd chi-sq is 1-a :
18 qchisq(1-a, df=15)
19
20 # The decision rule is to reject the null hypothesis
    if the observed value
21 # of the test statistic is less than 7.26093 or
    greater than 24.9958.
22
23 X_sq = ((n-1)*s_sq)/var
24 X_sq
25
26 # This observed chi-square value is in the
    nonrejection region because
27 # chi_sq(.05)=7.26 < chi_sq(observed) = 16.83 < chi_
    sq(.95) = 24.9958.
28 # The company fails to reject the null hypothesis.
    The population variance
29 # of overtime hours per week is 25.

```

---

#### R code Exa 9.5 Z value for Type II error

```

1 # Z value for Type II error : z = sample_mean_c -
    pop_mean_1/(sd/sqrt(n))
2
3 sample_mean_c = 11.979
4 pop_mean_1 = 11.96
5 sd = .10
6 n = 60

```

```

7
8 z = (sample_mean_c - pop_mean_1)/(sd/sqrt(n))
9 z

```

---

**R code Exa 9.6** Z value for Type II error

```

1 # Z value for Type II error
2
3 z_c = 1.96
4 p = .40
5 q = 1-p
6 n = 250
7 # z_c = (p_c-p)/sqrt(p*q/n)
8 p_c = z_c*sqrt((p*q)/n)+p
9 p_c
10 p_c1 = z_c*sqrt((p*q)/n)-p
11 p_c1
12
13 # z value on taking p_c = .46 and p = .36 :
14 p_c = .46
15 p = .36
16 z_c = (p_c-p)/sqrt(p*q/n)
17 z_c
18
19 # z value on taking p_c = .34 and p = .36 :
20 p_c = .34
21 p = .36
22 z_c = (p_c-p)/sqrt(p*q/n)
23 z_c
24
25 # 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:

```

26    .2454+.4994

---

## Chapter 10

# Statistical Inferences About Two Populations

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

```
1 # z formula for the difference in two sample means :
2 #  $z = \frac{(\text{samp\_mean\_1} - \text{samp\_mean\_2}) - (\text{pop\_mean\_1} - \text{pop\_mean\_2})}{\sqrt{(\text{sd1}^2/n1) + (\text{sd2}^2/n2)}}$ 
3
4 samp_mean_1 = 3352
5 samp_mean_2 = 5727
6 sd1 = 1100
7 sd2 = 1700
8 n1 = 87
9 n2 = 76
10
11 # Observed value of Z :
12 z1 = ((samp_mean_1 - samp_mean_2) - (0)) / sqrt((sd1^2/n1)
13       + (sd2^2/n2))
14
15 # Critical value of Z :
16 z = qnorm(.001, mean = 0, sd = 1, lower.tail = TRUE,
17          log.p = FALSE)
```

```

17 z
18
19 # sample critical :
20 s_c = (0)-(z*sqrt((sd1^2/n1)+(sd2^2/n2)))
21 s_c
22
23 # The difference in sample means would need to be at
    least 704.23
24 # to reject the null hypothesis.
25
26 # The actual sample difference in this problem :
27 s_c = samp_mean_1-samp_mean_2
28 s_c # which is considerably larger than the critical
    value of difference
29
30 # 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

```

1 # Confidence Interval to estimate difference in two
    population means :
2 # pop_mean_1-pop_mean_2 = (samp_mean_1-samp_mean_2)
    +/- (z*sqrt((sd1^2/n1)+(sd2^2/n2)))
3
4 n1 = 50
5 n2 = 50
6 samp_mean_1 = 21.45
7 samp_mean_2 = 24.6
8 sd1 = 3.46
9 sd2 = 2.99
10 z = 1.96
11 pmean_diff_1 = (samp_mean_1-samp_mean_2) + (z*sqrt((
    sd1^2/n1)+(sd2^2/n2)))

```



```

12 pmean_diff_1
13
14 pmean_diff_2 = (samp_mean_1-samp_mean_2) - (z*sqrt((
    sd1^2/n1)+(sd2^2/n2)))
15 pmean_diff_2

```

---

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

```

1 # t formula to test the difference in means assuming
    sd1, sd2 are equal :
2 #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)))
3
4 n1 = 46
5 n2 = 26
6 samp_mean_1 = 5.42
7 samp_mean_2 = 5.04
8 s1 = .58
9 s2 = .49
10 df = n1+n2-2
11
12 # Critical t value :
13 qt(.005, df, lower.tail = FALSE, log.p = FALSE)
14
15 # Observed t value :
16 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)
    ))
17 t
18
19 # Because the observed value of is greater than the
    critical table value of the decision is to reject
20 # the null hypothesis

```

---

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

```
1 # CONFIDENCE INTERVAL TO ESTIMATE difference in
  means ASSUMING THE POPULATION VARIANCES ARE
    UNKNOWN AND EQUAL :
2 n1 = 13
3 n2 = 15
4 samp_mean_1 = 4.35
5 samp_mean_2 = 6.84
6 s1 = 1.20
7 s2 = 1.42
8
9 alpha = .025
10 df = 26
11
12 t = qt(alpha, df, lower.tail = FALSE, log.p = FALSE)
13 t
14
15 # p_m_diff = pop_mean_1-pop_mean_2
16 s_diff = samp_mean_1-samp_mean_2
17 b = sqrt(((s1^2*(n1-1))+(s2^2*(n2-1)))/(n1+n2-2))
18 c = sqrt((1/n1)+(1/n2))
19
20
21 p_m_diff_1 = s_diff - (t*b*c)
22 p_m_diff_1
23
24 p_m_diff_2 = s_diff + (t*b*c)
25 p_m_diff_2
```

---

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

```
1 # t formula to test the Difference in Two Dependent
  Population :
2 #  $t = (\text{mean\_samp\_diff} - D) / (\text{sd} / \sqrt{n})$ 
3 #  $df = n-1$ 
4 #  $D = \text{mean\_pop\_diff}$ ,  $\text{sd} = \text{sd\_samp\_diff}$ ,  $n = \text{num\_of\_}$ 
  pairs,  $d = \text{samp\_diff\_pair}$ 
5
6
7 Individual <- c(1,2,3,4,5,6,7)
8 Before <- c(32,11,21,17,30,38,14)
9 After <- c(39,15,35,13,41,39,22)
10 n = 7
11
12 for(i in 1:7){
13   d = Before - After
14 }
15 print(d)
16 Individual <- cbind(Individual,Before,After,d)
17 Individual
18
19 mean_samp_diff = sum(d)/n
20 mean_samp_diff
21 d1 = sum(d)/7
22
23 sd = sqrt((sum((d-mean_samp_diff)^2))/(n-1))
24 sd
25
26 D = 0
27 t = (mean_samp_diff - D) / (sd/sqrt(n))
28 t
29
30 # Because the observed value of -2.54 is less than
  the critical, table value of -1.943 and the
31 # 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

```
1 # Z formula to test the difference in Population
  Proportion :
2 # z = ((p1_c - p2_c)-(p1-p2)) / sqrt((p_c*q_c)*((1/
  n1)+(1/n2)))
3 # p_c =((n1*p1_c)+(n2*p2_c))/(n1+n2)
4 # q_c = 1 - p_c
5
6 n1 = 100
7 n2 = 95
8 p1_c = .24
9 p2_c = .41
10
11 p_c =((n1*p1_c)+(n2*p2_c))/(n1+n2)
12 p_c
13 q_c = 1 - p_c
14 q_c
15 # p1 - p2 = 0
16
17 z = ( (p1_c - p2_c) - (0) ) / sqrt( (p_c*q_c) * ( (1
  /n1) + (1/n2) ) )
18 z
19
20 # If a one-tailed test had been used, zc would have
  been z.01 = 2.33,
21 # and the null hypothesis would have been rejected.
  If alpha had been .05,
22 # 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

```

1 # F test for two Population Variance :
2 #  $F = s1^2/s2^2$ 
3 #  $df\_num = v1 = n1-1$  and  $df\_deno = v2 = n2-1$ 
4
5 # from given table we computed :
6 s1_sq = 5961428.6
7 s2_sq = 737142.9
8 n1 = 7
9 n2 = 8
10
11 # critical F-value :
12 qf(.01, df1=n1-1, df2=n2-1, lower.tail = FALSE, log.
    p = FALSE)
13
14 # Obseved F- value :
15 F = s1_sq/s2_sq
16 F
17
18 # Because the observed value of  $F = 8.09$  is greater
    than the table
19 # 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

```
1 # One Way ANOVA SSE, SSc, SST values :
2 # SSC = sum( nj*( xj_b-x_b)^2)
3 # SSE = sum(sum(( xij-xj_b)^2))
4 # SST = sum(sum(( xij-x_b)^2))
5
6 a <- c(29,27,30,27,28)
7 b <- c(32,33,31,34,30)
8 c <- c(25,24,24,25,26)
9 df <- data.frame(a,b,c)
10 df
11
12 r = c(t(as.matrix(df))) # response data
13 r
14 f = c("a", "b", "c")    # treatment levels
15 k = 3                    # number of treatment
    levels
16 n = 5
17
18 tm = gl(k, 1, n*k, factor(f)) # matching
```

```

      treatments
19 tm
20
21 av = aov(r ~ tm)
22 av
23 summary(av)

```

---

### R code Exa 11.2 TUKEYs HSD Test

```

1 # TUKEYs HSD Test :  $HSD = q \cdot \sqrt{MSE/n}$  # q =
  critical value
2
3 a <- c(2.46, 2.41, 2.43, 2.47, 2.46)
4 b <- c(2.38, 2.34, 2.31, 2.40, 2.32)
5 c <- c(2.51, 2.48, 2.46, 2.49, 2.44)
6 d <- c(2.49, 2.47, 2.48, 2.46, 2.44)
7 e <- c(2.56, 2.57, 2.53, 2.55, 2.55)
8 df <- data.frame(a, b, c, d, e)
9 df
10
11
12 r = c(t(as.matrix(df))) # response data
13 r
14 f = c("a", "b", "c", "d", "e") # treatment levels
15 k = 5 # number of treatment
  levels
16 n = 5
17
18 tm = gl(k, 1, n*k, factor(f)) # matching
  treatments
19 tm
20
21 av = aov(r ~ tm)
22 av
23 b <- summary(av)

```

```

24 b
25
26 # From above anova analysis we get MSE value :
27 MSE = 0.000618
28 q = 5.29
29 n = 5
30 HSD = q*sqrt(MSE/n)
31 HSD

```

---

### R code Exa 11.3 Randomized Block Design

```

1 # Formula for computing Randomized Block Design for
  SSE, SSC, SSR, SST
2 # SSC = n*sum(( xj_b-x_b)^2)
3 # SSR = C*sum(( xi_b-x_b)^2)
4 # SSE = sum(sum(( xij-xj_b-xi_b+x_b)^2))
5 # SST = sum(sum(( xij-x_b)^2))
6
7 a <- c(3.47,3.43,3.44,3.46,3.46,3.44)
8 b <- c(3.40,3.41,3.41,3.45,3.40,3.43)
9 c <- c(3.38,3.42,3.43,3.40,3.39,3.42)
10 d <- c(3.32,3.35,3.36,3.30,3.39,3.39)
11 e <- c(3.50,3.44,3.45,3.45,3.48,3.49)
12 df <- data.frame(a,b,c,d,e)
13 df
14
15
16 r = c(t(as.matrix(df))) # response data
17 r
18 f = c("a", "b", "c", "d", "e") # treatment levels
19 k = 5 # number of treatment
    levels
20 n = 6
21
22 blk = gl(n, k, k*n) # blocking factor

```



```

23 blk
24
25 tm = gl(k, 1, n*k, factor(f)) # matching
    treatments
26 tm
27
28 av = aov(r ~ tm + blk)
29 av
30 b <- summary(av)
31 b

```

---

#### R code Exa 11.4 Two Way ANOVA

```

1 # Two-Way ANOVA :
2
3 Types_of_warehouses <- c("GM", "GM", "GM", "GM", "GM", "
    GM", "GM", "GM", "GM",
4
5     "Com", "Com", "Com", "Com", "
        Com", "Com", "Com", "Com", "
        Com",
6
7     "BS", "BS", "BS", "BS", "BS", "
        BS", "BS", "BS", "BS",
8
9     "CS", "CS", "CS", "CS", "
        CS", "CS", "CS", "CS", "
        CS")
10
11
12
13 Training_sessions <- c("A", "A", "A", "B", "B", "B", "C", "
    C", "C", "A", "A", "A",
14
15     "B", "B", "B", "C", "C", "C", "A", "
        A", "A", "B", "B", "B",
16
17     "C", "C", "C", "A", "A", "A", "B", "
        B", "B", "C", "C", "C")
18
19
20 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,
14      3.5,3.5, 4,2,2,3,5, 4.5,2.5,4, 4.5, 5)
15
16 df <- data.frame(Types_of_warehouses,Training_
      sessions,Values)
17 df
18
19 av <- aov(Values~as.factor(Types_of_warehouses)*as.
      factor(Training_sessions),data= df)
20 av
21 summary(av)

```

---

## Chapter 12

# Simple Regression Analysis and Correlation

**R code Exa 12.1** Slope of Regression line

```
1 # Slope of Regression line :  
2  
3 no_of_beds <- c(23,29,29,35,42,46,50,54,64,66,76,78)  
4 FTEs <- c  
    (69,95,102,118,126,125,138,178,156,184,176,225)  
5 Hospitals<-data.frame(no_of_beds,FTEs)  
6 Hospitals  
7  
8 # least squares equation of the regression line is :  
9 lm( FTEs ~ no_of_beds, data=Hospitals)  
10  
11 # y_c = 30.91 + 2.23 * x
```

---

**R code Exa 12.2** Residual Analysis

```
1 # Residual Analysis :
```

```

2
3 Hospitals <- c(1,2,3,4,5,6,7,8,9,10,11,12)
4 x <- c(23,29,29,35,42,46,50,54,64,66,76,78)
5 y <- c
    (69,95,102,118,126,125,138,178,156,184,176,225)
6 for(i in 1:12){
7   x_sq <- x*x
8 }
9 print(x_sq)
10
11 for(i in 1:12){
12   xy <- x*y
13 }
14 print(xy)
15
16 x1 <- cbind(x,y,x_sq,xy)
17
18 n = 12
19
20 b1 = ((sum(x*y))-((sum(x)*sum(y))/n))/((sum(x^2))-
    sum(x)^2/n))
21 b1
22
23 b0 = (sum(y)/n)-b1*(sum(x)/n)
24 b0
25
26 # y_c = 30.91 + 2.23 * x
27 y_c = b0 + b1*x
28 y_c
29 x1 <- cbind(x1,y_c)
30
31 Residual <- y-y_c
32 Residual
33
34 x1 <- cbind(x1,Residual)
35 View(x1)
36
37 sum(Residual)

```

```
38
39 hist(Residual)
```

---

### R code Exa 12.3 Standard Error of Estimation

```
1 # Standard Error of Estimation : Se = sqrt(SSE/(n-2)
  )
2 # SSE = sum((y-y_c)^2)
3
4 Hospitals <- c(1,2,3,4,5,6,7,8,9,10,11,12)
5 x <- c(23,29,29,35,42,46,50,54,64,66,76,78)
6 y <- c
  (69,95,102,118,126,125,138,178,156,184,176,225)
7 for(i in 1:12){
8   x_sq <- x*x
9 }
10 print(x_sq)
11
12 for(i in 1:12){
13   xy <- x*y
14 }
15 print(xy)
16
17 x1 <- cbind(x,y,x_sq,xy)
18
19 n = 12
20
21 b1 = ((sum(x*y))-((sum(x)*sum(y))/n))/((sum(x^2))-
  sum(x)^2/n))
22 b1
23
24 b0 = (sum(y)/n)-b1*(sum(x)/n)
25 b0
26
27 # y_c = 30.91 + 2.23 * x
```

```

28 y_c = b0 + b1*x
29 y_c
30 x1 <- cbind(x1,y_c)
31
32 Residual <- y-y_c
33 Residual
34
35 x1 <- cbind(x1,Residual)
36
37 for(i in 1:12){
38   Residual_sq = Residual^2
39 }
40 print(Residual_sq)
41
42 x1 <- cbind(x1,Residual_sq)
43 View(x1)
44
45 SSE = sum(Residual_sq)
46 SSE
47
48 Se = sqrt(SSE/(n-2))
49 Se

```

---

#### R code Exa 12.4 Coefficient of Determination

```

1 # Coefficient of Determination : r_sq = 1 - (SSE/SS_
  yy)
2 # SS_yy = sum(y_sq)-(sum(y)^2/n)
3
4 Hospitals <- c(1,2,3,4,5,6,7,8,9,10,11,12)
5 x <- c(23,29,29,35,42,46,50,54,64,66,76,78)
6 y <- c
  (69,95,102,118,126,125,138,178,156,184,176,225)
7 for(i in 1:12){
8   x_sq <- x*x

```

```

9  }
10 print(x_sq)
11
12 for(i in 1:12){
13   xy <- x*y
14 }
15 print(xy)
16
17 x1 <- cbind(x,y,x_sq,xy)
18
19 n = 12
20
21 b1 = ((sum(x*y))-((sum(x)*sum(y))/n))/((sum(x^2))-
      sum(x)^2/n))
22 b1
23
24 b0 = (sum(y)/n)-b1*(sum(x)/n)
25 b0
26
27 # y_c = 30.91 + 2.23 * x
28 y_c = b0 + b1*x
29 y_c
30 x1 <- cbind(x1,y_c)
31
32 Residual <- y-y_c
33 Residual
34
35 x1 <- cbind(x1,Residual)
36
37 for(i in 1:12){
38   Residual_sq = Residual^2
39 }
40 print(Residual_sq)
41
42 x1 <- cbind(x1,Residual_sq)
43 View(x1)
44
45 SSE = sum(Residual_sq)

```

```

46 SSE
47
48 SS_yy = sum(y^2)-(sum(y)^2/n)
49 SS_yy
50
51 r_sq = 1-(SSE/SS_yy)
52 r_sq
53
54 # Or r_sq = (b1^2 * SS_xx)/SS_yy

```

---

#### R code Exa 12.5 t test for slope

```

1 # t test for slope :
2
3 no_of_beds <- c(23,29,29,35,42,46,50,54,64,66,76,78)
4 FTEs <- c
      (69,95,102,118,126,125,138,178,156,184,176,225)
5 Hospitals<-data.frame(no_of_beds,FTEs)
6 Hospitals
7
8 # critical t value :
9 qchisq(.01,df = 10)
10
11 # least squares equation of the regression line is :
12 a <- lm( FTEs ~ no_of_beds, data=Hospitals)
13 a          # y_c = 30.91 + 2.23 * x
14 b <- summary(a)
15 b
16
17 # observed t value :
18 b$coefficients[6]

```

---



**R code Exa 12.6** CONFIDENCE INTERVAL TO ESTIMATE THE SINGLE VALUE FOR A GIVEN VALUE OF  $x$

```

1 # CONFIDENCE INTERVAL TO ESTIMATE E (yx) FOR A GIVEN
  VALUE OF x :
2 # y_c +/- t*Se*sqrt((1/n)+((x0-x_b)^2)/SS_xx)
3 # SS_xx = sum(x^2)-(sum(x)^2/n)
4
5 no_of_beds <- c(23,29,29,35,42,46,50,54,64,66,76,78)
6 FTEs <- c
  (69,95,102,118,126,125,138,178,156,184,176,225)
7 Hospitals<-data.frame(no_of_beds,FTEs)
8 Hospitals
9
10 a <- lm( FTEs ~ no_of_beds, data=Hospitals)
11 a
12
13 data = data.frame(no_of_beds=40)
14 data
15
16 predict(a, data, interval="confidence")
17
18 predict(a, data, interval="predict")

```

---

**R code Exa 12.7** Regression Analysis Example

```

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

```

```

8
9 library("ggplot2")
10 ggplot(df, aes(x=Month, y=Sales)) + geom_point(size
    =1)
11
12 # Regression Analysis: Sales versus Month
13 a <- lm(Sales~Month_number, data= df)
14 a
15 summary(a)
16
17 #  $y_{\text{cap}} = 32,628.2 - 86.21 * x$  :
18 x =10
19 y_cap = 32628.2 - 86.21*x
20 y_cap

```

---

# Chapter 13

## Multiple Regression Analysis

**R code Exa 13.1** Multiple Regression Model

```
1 # Multiple Regression Model:
2
3 Year <- c
  (1980,1982,1984,1986,1988,1990,1992,1994,1996,1998,2000,2002,2004)
4 Prime_Interest_rate <- c
  (15.26,14.85,12.04,8.33,9.32,10.01,6.25,7.15,8.27,8.35,9.23,4.67,4.67)
5 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)
6 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)
7 df <- data.frame(Year,Prime_Interest_rate,Unemp_rate
  ,Personal_saving)
8 df
9
10 a <-lm(Prime_Interest_rate ~ Unemp_rate+Personal_
  saving,data=df)
11 a
```

```

12 summary(a)
13 anova(a)
14
15 #  $y_{\text{cap}} = 7.4904 - 0.6725x_1 + 0.9500x_2$ 
16 # If the unemployment rate is 6.5 and the personal
    saving rate is 5.0,
17 # the predicted prime interest rate is 7.869%:
18 x1 = 6.5
19 x2 = 5.0
20 y_cap = 7.4904 - (0.6725)*(x1) + (0.9500)*(x2)
21 y_cap

```

---

### R code Exa 13.2 Multiple Regression Analysis Model

```

1 # Multiple Regression Model:
2
3 Year <- c
    (1980,1982,1984,1986,1988,1990,1992,1994,1996,1998,2000,2002,2004,
4 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,
5 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)
6 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)
7 df <- data.frame(Year,Prime_Interest_rate,Unemp_rate
    ,Personal_saving)
8 View(df)
9
10 a <-lm(Prime_Interest_rate ~ Unemp_rate+Personal_
    saving,data=df)
11 a

```

```
12 s <-summary(a)
13 s
14 anova(a)
15
16 pred <- predict(a)
17 resd <- s$residuals
18 data <- data.frame(pred,resd)
19 View(data)
```

---

# Chapter 14

## Building Multiple Regression Models

R code Exa 14.1 Model Transformation

```
1 # Model Transformation :  $y = B_0 * x_{B1} + E$ 
2
3 y_cost <- c(1.2,9.0,4.5,3.2,13.0,0.6,1.8,2.7)
4 x_weight <- c
      (450,20200,9060,3500,75600,175,800,2100)
5 y_cost <- data.frame(y_cost,x_weight)
6 y_cost
7
8 #  $\log y = \log B_0 + B_1 * \log x + E$ 
9 log_xy <- log10(y_cost)
10 log_xy
11
12 a <- lm(y_cost ~ x_weight, data=log_xy)
13 a
14 b <- summary(a)
15 b
16
17 b0 <- b$coefficients[1]
18 b0
```

```
19 b1 <- b$coefficients[2]
20 b1
21
22 logy_c = b0 + b1 * (sum(log_xy$x_weight)/8)
23 logy_c
24
25 # antilog = 2.9644
26 # y = (.055857)*x^.49606
```

---

# Chapter 15

## Time Series Forecasting and Index Numbers

R code Exa 15.1.a Moving average

```
1 # Moving average :
2
3 Month <- c("January", "February", "March", "April", "May",
4           ", "June", "July", "August", "September", "October", "
5           November", "December")
6 Shipments <- c
7           (1056, 1345, 1381, 1191, 1259, 1361, 1110, 1334, 1416, 1282, 1341, 1382)
8
9 Month <- cbind(Month, Shipments)
10 Month
11
12 # The first moving average is
13 first_four_Month_Moving_Average = sum(Shipments[1],
14           Shipments[2], Shipments[3], Shipments[4])/4
15 first_four_Month_Moving_Average
16 Second_four_Month_Moving_Average = sum(Shipments[5],
17           Shipments[2], Shipments[3], Shipments[4])/4
18 Second_four_Month_Moving_Average
19 Third_four_Month_Moving_Average = sum(Shipments[5],
```



```

    Shipments[6],Shipments[3],Shipments[4])/4
14 Third_four_Month_Moving_Average
15 fourth_four_Month_Moving_Average = sum(Shipments[5],
    Shipments[6],Shipments[7],Shipments[4])/4
16 fourth_four_Month_Moving_Average
17 fifth_four_Month_Moving_Average = sum(Shipments[5],
    Shipments[6],Shipments[7],Shipments[8])/4
18 fifth_four_Month_Moving_Average
19 sixth_four_Month_Moving_Average = sum(Shipments[9],
    Shipments[6],Shipments[7],Shipments[8])/4
20 sixth_four_Month_Moving_Average
21 seventh_four_Month_Moving_Average = sum(Shipments
    [9],Shipments[10],Shipments[7],Shipments[8])/4
22 seventh_four_Month_Moving_Average
23 eight_four_Month_Moving_Average = sum(Shipments[9],
    Shipments[10],Shipments[11],Shipments[8])/4
24 eight_four_Month_Moving_Average
25
26 a = " "
27 b= " "
28 c = " "
29 d = " "
30 Average = rbind(a,b,c,d,first_four_Month_Moving_
    Average,Second_four_Month_Moving_Average,Third_
    four_Month_Moving_Average ,
31         fourth_four_Month_Moving_Average,fifth_
    four_Month_Moving_Average,sixth_four_
    Month_Moving_Average ,
32         seventh_four_Month_Moving_Average,eight_
    four_Month_Moving_Average)
33 Average

```

---

#### R code Exa 15.1.b Moving average

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

```

2 # Moving average :
3
4 Month <- c("January", "February", "March", "April", "May",
            ", "June", "July", "August", "September", "October", "
            November", "December")
5 Shipments <- c
            (1056, 1345, 1381, 1191, 1259, 1361, 1110, 1334, 1416, 1282, 1341, 1382)

6 Month <- cbind(Month, Shipments)
7 Month
8
9 # The ???rst moving average is
10 first_four_Month_Moving_Average = sum(Shipments[1],
            Shipments[2], Shipments[3], Shipments[4])/4
11 first_four_Month_Moving_Average
12 Second_four_Month_Moving_Average = sum(Shipments[5],
            Shipments[2], Shipments[3], Shipments[4])/4
13 Second_four_Month_Moving_Average
14 Third_four_Month_Moving_Average = sum(Shipments[5],
            Shipments[6], Shipments[3], Shipments[4])/4
15 Third_four_Month_Moving_Average
16 fourth_four_Month_Moving_Average = sum(Shipments[5],
            Shipments[6], Shipments[7], Shipments[4])/4
17 fourth_four_Month_Moving_Average
18 fifth_four_Month_Moving_Average = sum(Shipments[5],
            Shipments[6], Shipments[7], Shipments[8])/4
19 fifth_four_Month_Moving_Average
20 sixth_four_Month_Moving_Average = sum(Shipments[9],
            Shipments[6], Shipments[7], Shipments[8])/4
21 sixth_four_Month_Moving_Average
22 seventh_four_Month_Moving_Average = sum(Shipments
            [9], Shipments[10], Shipments[7], Shipments[8])/4
23 seventh_four_Month_Moving_Average
24 eight_four_Month_Moving_Average = sum(Shipments[9],
            Shipments[10], Shipments[11], Shipments[8])/4
25 eight_four_Month_Moving_Average
26
27 a = " "

```

```

28 b= " "
29 c = " "
30 d = " "
31 Average = rbind(a,b,c,d,first_four_Month_Moving_
    Average,Second_four_Month_Moving_Average,Third_
    four_Month_Moving_Average,
32         fourth_four_Month_Moving_Average,
            fifth_four_Month_Moving_Average,
            sixth_four_Month_Moving_Average,
33         seventh_four_Month_Moving_Average,
            eight_four_Month_Moving_Average)

34 Average
35
36 a = " "
37 b= " "
38 c = " "
39 d = " "
40 Error_May = Shipments[5]-first_four_Month_Moving_
    Average
41 Error_June = Shipments[6]-Second_four_Month_Moving_
    Average
42 Error_July = Shipments[7]-Third_four_Month_Moving_
    Average
43 Error_Aug = Shipments[8]-fourth_four_Month_Moving_
    Average
44 Error_sep = Shipments[9]-fifth_four_Month_Moving_
    Average
45 Error_oct = Shipments[10]-sixth_four_Month_Moving_
    Average
46 Error_nov = Shipments[11]-seventh_four_Month_Moving_
    Average
47 Error_dec = Shipments[12]-eight_four_Month_Moving_
    Average
48 Error <- rbind(a,b,c,d>Error_May>Error_June>Error_
    July>Error_Aug>Error_sep>Error_oct>Error_nov,
    Error_dec)
49 Error
50 Month <- cbind(Month,Average>Error)

```

51 View(Month)

---

### R code Exa 15.2 Weighted Moving Average

```
1 # Weighted MOving Average :  $3 \cdot l + 3 \cdot p + 3 \cdot b_p / 6$ 
2
3 Month <- c("January", "February", "March", "April", "May",
4           ", "June", "July", "August", "September", "October", "November", "December")
5 Shipments <- c(1056, 1345, 1381, 1191, 1259, 1361, 1110, 1334, 1416, 1282, 1341, 1382)
6
7 Month <- data.frame(Month, Shipments)
8 Month
9 weights1 <- c(4, 2, 1, 1)
10
11 # install.packages("stats")
12 library(stats)
13
14 f_weight_may <- weighted.mean(Shipments[4:1],
15                               weights1)
16 f_weight_june <- weighted.mean(Shipments[5:2],
17                                weights1)
18 f_weight_july <- weighted.mean(Shipments[6:3],
19                                weights1)
20 f_weight_aug <- weighted.mean(Shipments[7:4],
21                               weights1)
22 f_weight_sep <- weighted.mean(Shipments[8:5],
23                               weights1)
24 f_weight_oct <- weighted.mean(Shipments[9:6],
25                               weights1)
26 f_weight_nov <- weighted.mean(Shipments[10:7],
27                               weights1)
28 f_weight_dec <- weighted.mean(Shipments[11:8],
29                               weights1)
```

```

20 f_weights <- data.frame(f_weight_may,f_weight_june,f
    _weight_july,f_weight_aug,
21                          f_weight_sep,f_weight_oct,f_
    weight_nov,f_weight_dec)
22 f_weights
23
24 Shipments[5:12] - f_weights
25
26 # We noticed that in this problem the errors
    obtained by using the 4-month weighted moving
    average
27 # were greater than most of the errors obtained by
    using an unweighted 4-month moving average
28 # in Ex15_1.

```

---

### R code Exa 15.3 EXPONENTIAL SMOOTHING

```

1 # EXPONENTIAL SMOOTHING :
2 Year <- c(1:16)
3 Total_units <- c
    (1193,1014,1200,1288,1457,1354,1477,1474,1617,1641,1569,
4
    1603,1705,1848,1956,2068)
5 data <- data.frame(Year,Total_units)
6 data
7
8 library(ggplot2)
9 ggplot(data=data, aes(x=data$Year, y=data$Total_
    units, group=1)) +
10   geom_line(linetype = "dashed")+
11   geom_point()
12
13 # using exponential smoothing function i.e. ses() :
14 # install.package("forecast")
15 library(forecast)

```

```

16 # Forecast and error values for alpha = 0.2 :
17 f_a <- ses(Total_units, h = 8, alpha = 0.2, initial
    = "simple")[[ "fitted" ]]
18 error_a <- ses(Total_units, h = 8, alpha = 0.2,
    initial = "simple")[[ "residuals" ]]
19
20 # Forecast and error values for alpha = 0.2 :
21 f_b <- ses(Total_units, h = 8, alpha = 0.5, initial
    = "simple")[[ "fitted" ]]
22 error_b <- ses(Total_units, h = 8, alpha = 0.5,
    initial = "simple")[[ "residuals" ]]
23
24 # Forecast and error values for alpha = 0.2 :
25 f_c <- ses(Total_units, h = 8, alpha = 0.8, initial
    = "simple")[[ "fitted" ]]
26 error_c <- ses(Total_units, h = 8, alpha = 0.8,
    initial = "simple")[[ "residuals" ]]
27
28 f_data <- data.frame(data, f_a, error_a, f_b, error_b, f_
    c, error_c)
29 View(f_data)
30
31 # MAD and MSE values of alpha = 0.2, 0.5, 0.8 :
32 MAD_a <- sum(abs(error_a))/15
33 MSE_a <- sum(abs(error_a^2))/15
34
35 MAD_b <- sum(abs(error_b))/15
36 MSE_b <- sum(abs(error_b^2))/15
37
38 MAD_c <- sum(abs(error_c))/15
39 MSE_c <- sum(abs(error_c^2))/15
40
41 val <- rbind(MAD_a, MSE_a, MAD_b, MSE_b, MAD_c, MSE_c)
42 val

```

---

### R code Exa 15.4 Regression Trend Analysis Using Quadratic Models

```
1 # Regression Trend Analysis Using Quadratic Models
2
3 Year <- c(1991:2007)
4 Labour_force <- c
  (117.72,118.49,120.26,123.06,124.90,126.71,129.56,131.46,133.49,135.49,137.49,139.49,141.49,143.49,145.49,147.49,149.49,151.49,153.49,155.49,157.49,159.49,161.49,163.49,165.49,167.49,169.49,171.49,173.49,175.49,177.49,179.49,181.49,183.49,185.49,187.49,189.49,191.49,193.49,195.49,197.49,199.49,201.49,203.49,205.49,207.49,209.49,211.49,213.49,215.49,217.49,219.49,221.49,223.49,225.49,227.49,229.49,231.49,233.49,235.49,237.49,239.49,241.49,243.49,245.49,247.49,249.49,251.49,253.49,255.49,257.49,259.49,261.49,263.49,265.49,267.49,269.49,271.49,273.49,275.49,277.49,279.49,281.49,283.49,285.49,287.49,289.49,291.49,293.49,295.49,297.49,299.49,301.49,303.49,305.49,307.49,309.49,311.49,313.49,315.49,317.49,319.49,321.49,323.49,325.49,327.49,329.49,331.49,333.49,335.49,337.49,339.49,341.49,343.49,345.49,347.49,349.49,351.49,353.49,355.49,357.49,359.49,361.49,363.49,365.49,367.49,369.49,371.49,373.49,375.49,377.49,379.49,381.49,383.49,385.49,387.49,389.49,391.49,393.49,395.49,397.49,399.49,401.49,403.49,405.49,407.49,409.49,411.49,413.49,415.49,417.49,419.49,421.49,423.49,425.49,427.49,429.49,431.49,433.49,435.49,437.49,439.49,441.49,443.49,445.49,447.49,449.49,451.49,453.49,455.49,457.49,459.49,461.49,463.49,465.49,467.49,469.49,471.49,473.49,475.49,477.49,479.49,481.49,483.49,485.49,487.49,489.49,491.49,493.49,495.49,497.49,499.49,501.49,503.49,505.49,507.49,509.49,511.49,513.49,515.49,517.49,519.49,521.49,523.49,525.49,527.49,529.49,531.49,533.49,535.49,537.49,539.49,541.49,543.49,545.49,547.49,549.49,551.49,553.49,555.49,557.49,559.49,561.49,563.49,565.49,567.49,569.49,571.49,573.49,575.49,577.49,579.49,581.49,583.49,585.49,587.49,589.49,591.49,593.49,595.49,597.49,599.49,601.49,603.49,605.49,607.49,609.49,611.49,613.49,615.49,617.49,619.49,621.49,623.49,625.49,627.49,629.49,631.49,633.49,635.49,637.49,639.49,641.49,643.49,645.49,647.49,649.49,651.49,653.49,655.49,657.49,659.49,661.49,663.49,665.49,667.49,669.49,671.49,673.49,675.49,677.49,679.49,681.49,683.49,685.49,687.49,689.49,691.49,693.49,695.49,697.49,699.49,701.49,703.49,705.49,707.49,709.49,711.49,713.49,715.49,717.49,719.49,721.49,723.49,725.49,727.49,729.49,731.49,733.49,735.49,737.49,739.49,741.49,743.49,745.49,747.49,749.49,751.49,753.49,755.49,757.49,759.49,761.49,763.49,765.49,767.49,769.49,771.49,773.49,775.49,777.49,779.49,781.49,783.49,785.49,787.49,789.49,791.49,793.49,795.49,797.49,799.49,801.49,803.49,805.49,807.49,809.49,811.49,813.49,815.49,817.49,819.49,821.49,823.49,825.49,827.49,829.49,831.49,833.49,835.49,837.49,839.49,841.49,843.49,845.49,847.49,849.49,851.49,853.49,855.49,857.49,859.49,861.49,863.49,865.49,867.49,869.49,871.49,873.49,875.49,877.49,879.49,881.49,883.49,885.49,887.49,889.49,891.49,893.49,895.49,897.49,899.49,901.49,903.49,905.49,907.49,909.49,911.49,913.49,915.49,917.49,919.49,921.49,923.49,925.49,927.49,929.49,931.49,933.49,935.49,937.49,939.49,941.49,943.49,945.49,947.49,949.49,951.49,953.49,955.49,957.49,959.49,961.49,963.49,965.49,967.49,969.49,971.49,973.49,975.49,977.49,979.49,981.49,983.49,985.49,987.49,989.49,991.49,993.49,995.49,997.49,999.49,1001.49,1003.49,1005.49,1007.49,1009.49,1011.49,1013.49,1015.49,1017.49,1019.49,1021.49,1023.49,1025.49,1027.49,1029.49,1031.49,1033.49,1035.49,1037.49,1039.49,1041.49,1043.49,1045.49,1047.49,1049.49,1051.49,1053.49,1055.49,1057.49,1059.49,1061.49,1063.49,1065.49,1067.49,1069.49,1071.49,1073.49,1075.49,1077.49,1079.49,1081.49,1083.49,1085.49,1087.49,1089.49,1091.49,1093.49,1095.49,1097.49,1099.49,1101.49,1103.49,1105.49,1107.49,1109.49,1111.49,1113.49,1115.49,1117.49,1119.49,1121.49,1123.49,1125.49,1127.49,1129.49,1131.49,1133.49,1135.49,1137.49,1139.49,1141.49,1143.49,1145.49,1147.49,1149.49,1151.49,1153.49,1155.49,1157.49,1159.49,1161.49,1163.49,1165.49,1167.49,1169.49,1171.49,1173.49,1175.49,1177.49,1179.49,1181.49,1183.49,1185.49,1187.49,1189.49,1191.49,1193.49,1195.49,1197.49,1199.49,1201.49,1203.49,1205.49,1207.49,1209.49,1211.49,1213.49,1215.49,1217.49,1219.49,1221.49,1223.49,1225.49,1227.49,1229.49,1231.49,1233.49,1235.49,1237.49,1239.49,1241.49,1243.49,1245.49,1247.49,1249.49,1251.49,1253.49,1255.49,1257.49,1259.49,1261.49,1263.49,1265.49,1267.49,1269.49,1271.49,1273.49,1275.49,1277.49,1279.49,1281.49,1283.49,1285.49,1287.49,1289.49,1291.49,1293.49,1295.49,1297.49,1299.49,1301.49,1303.49,1305.49,1307.49,1309.49,1311.49,1313.49,1315.49,1317.49,1319.49,1321.49,1323.49,1325.49,1327.49,1329.49,1331.49,1333.49,1335.49,1337.49,1339.49,1341.49,1343.49,1345.49,1347.49,1349.49,1351.49,1353.49,1355.49,1357.49,1359.49,1361.49,1363.49,1365.49,1367.49,1369.49,1371.49,1373.49,1375.49,1377.49,1379.49,1381.49,1383.49,1385.49,1387.49,1389.49,1391.49,1393.49,1395.49,1397.49,1399.49,1401.49,1403.49,1405.49,1407.49,1409.49,1411.49,1413.49,1415.49,1417.49,1419.49,1421.49,1423.49,1425.49,1427.49,1429.49,1431.49,1433.49,1435.49,1437.49,1439.49,1441.49,1443.49,1445.49,1447.49,1449.49,1451.49,1453.49,1455.49,1457.49,1459.49,1461.49,1463.49,1465.49,1467.49,1469.49,1471.49,1473.49,1475.49,1477.49,1479.49,1481.49,1483.49,1485.49,1487.49,1489.49,1491.49,1493.49,1495.49,1497.49,1499.49,1501.49,1503.49,1505.49,1507.49,1509.49,1511.49,1513.49,1515.49,1517.49,1519.49,1521.49,1523.49,1525.49,1527.49,1529.49,1531.49,1533.49,1535.49,1537.49,1539.49,1541.49,1543.49,1545.49,1547.49,1549.49,1551.49,1553.49,1555.49,1557.49,1559.49,1561.49,1563.49,1565.49,1567.49,1569.49,1571.49,1573.49,1575.49,1577.49,1579.49,1581.49,1583.49,1585.49,1587.49,1589.49,1591.49,1593.49,1595.49,1597.49,1599.49,1601.49,1603.49,1605.49,1607.49,1609.49,1611.49,1613.49,1615.49,1617.49,1619.49,1621.49,1623.49,1625.49,1627.49,1629.49,1631.49,1633.49,1635.49,1637.49,1639.49,1641.49,1643.49,1645.49,1647.49,1649.49,1651.49,1653.49,1655.49,1657.49,1659.49,1661.49,1663.49,1665.49,1667.49,1669.49,1671.49,1673.49,1675.49,1677.49,1679.49,1681.49,1683.49,1685.49,1687.49,1689.49,1691.49,1693.49,1695.49,1697.49,1699.49,1701.49,1703.49,1705.49,1707.49,1709.49,1711.49,1713.49,1715.49,1717.49,1719.49,1721.49,1723.49,1725.49,1727.49,1729.49,1731.49,1733.49,1735.49,1737.49,1739.49,1741.49,1743.49,1745.49,1747.49,1749.49,1751.49,1753.49,1755.49,1757.49,1759.49,1761.49,1763.49,1765.49,1767.49,1769.49,1771.49,1773.49,1775.49,1777.49,1779.49,1781.49,1783.49,1785.49,1787.49,1789.49,1791.49,1793.49,1795.49,1797.49,1799.49,1801.49,1803.49,1805.49,1807.49,1809.49,1811.49,1813.49,1815.49,1817.49,1819.49,1821.49,1823.49,1825.49,1827.49,1829.49,1831.49,1833.49,1835.49,1837.49,1839.49,1841.49,1843.49,1845.49,1847.49,1849.49,1851.49,1853.49,1855.49,1857.49,1859.49,1861.49,1863.49,1865.49,1867.49,1869.49,1871.49,1873.49,1875.49,1877.49,1879.49,1881.49,1883.49,1885.49,1887.49,1889.49,1891.49,1893.49,1895.49,1897.49,1899.49,1901.49,1903.49,1905.49,1907.49,1909.49,1911.49,1913.49,1915.49,1917.49,1919.49,1921.49,1923.49,1925.49,1927.49,1929.49,1931.49,1933.49,1935.49,1937.49,1939.49,1941.49,1943.49,1945.49,1947.49,1949.49,1951.49,1953.49,1955.49,1957.49,1959.49,1961.49,1963.49,1965.49,1967.49,1969.49,1971.49,1973.49,1975.49,1977.49,1979.49,1981.49,1983.49,1985.49,1987.49,1989.49,1991.49,1993.49,1995.49,1997.49,1999.49,2001.49,2003.49,2005.49,2007.49)
5 Year_sq <- Year^2
6 Year <- data.frame(Year,Labour_force,Year_sq)
7 Year
8 a <-lm(Labour_force~Year,data=Year)
9 a
10 anova(a)
11 ggplot(data = Year,aes(x=Year,y=Labour_force))+geom_
  point()+geom_smooth(method = "lm")
12
13 b <-lm(Labour_force~.,data=Year)
14 b
15 anova(b)
16 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

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

```

10 q.ChildrenTylenol <- c(25,30)
11 p.Computerpaper <- c(11.95,13.20)
12 q.Computerpaper <- c(6,8)
13 p.Thermometer <- c(7.90,9.00)
14 q.Thermometer <- c(4,2)
15
16 data <- data.frame(year,p.Syrings,q.Syrings,p.Cotton
    ,q.Cotton,p.Patient,q.Patient,
17                      p.ChildrenTylenol,q.
                        ChildrenTylenol,p.
                        Computerpaper,q.Computerpaper,
18                      p.Thermometer,q.Thermometer)
19 data
20
21 # Unweighted Aggregate Index for 2009 :
22 p_2009 <- sum(p.Syrings[2],p.Cotton[2],p.Patient[2],
    p.ChildrenTylenol[2],p.Computerpaper[2],
23              p.Thermometer[2])
24 p_2008 <- sum(p.Syrings[1],p.Cotton[1],p.Patient[1],
    p.ChildrenTylenol[1],p.Computerpaper[1],
25              p.Thermometer[1])
26 I = (p_2009/p_2008)*100
27 I
28
29 # Laspeyres Price Indices
30 # install.packages("micEcon")
31 # install.packages("micEconIndex")
32 library(micEconIndex)
33 library(micEcon)
34 a <- priceIndex(c("p.Syrings","p.Cotton","p.Patient"
    ,"p.ChildrenTylenol","p.Computerpaper",
35                  "p.Thermometer"), c("q.Syrings","q.
    Cotton","q.Patient","q.
    ChildrenTylenol",
36                                     "q.Computerpaper","
    q.Thermometer")
    ,1,data)
37 a

```



```

38 I_2009_Laspeyres <- a[2]*100
39 I_2009_Laspeyres
40
41 # Paasche Price Indices
42 b <- priceIndex(c("p.Syrings","p.Cotton","p.Patient"
43                 ,"p.ChildrenTylenol","p.Computerpaper",
44                 "p.Thermometer"), c("q.Syrings","q.
45                                     Cotton","q.Patient","q.
46                                     ChildrenTylenol",
47                                     "q.Computerpaper","
48                                     q.Thermometer")
49                                     ,1,data,"Paasche
50                                     ")
51
52 b
53 I_2009_Passache <- b[2]*100
54 I_2009_Passache

```

---

# Chapter 16

## Analysis of Categorical Data

**R code Exa 16.1** CHI SQUARE GOODNESS OF FIT TEST

```
1 # CHI-SQUARE GOODNESS OF-FIT TEST :  $X_{sq} = \sum((fo - fe)^2 / fe)$ 
2 #  $df = k - 1 - c$ 
3 Month <- c("January", "February", "March", "April", "May",
4            ", "June", "July", "August", "September", "October", "November", "December")
5 fo <- c(1610, 1585, 1649, 1590, 1540, 1397, 1410, 1350, 1495, 1564, 1602, 1655)
6
7 # critical value of chi-square when alpha is 0.01 :
8 qchisq(.99, df=11)
9
10 fe <- sum(fo)/12
11 for(i in 1:12){
12   X = (fo-fe)^2/fe
13 }
14 print(X)
15
16 # Observed chi-square value :
```

```

17 X_sq = sum(X)
18 X_sq
19
20 Month <- cbind(Month,fo,fe,X)
21 Month
22
23 # The observed value of chi-square is 74.37, greater
    than the critical table value i.e. 24.725,
24 # so the decision is to reject the null hypothesis.
    This problem provides enough
25 # evidence to indicate that the distribution of milk
    sales is not uniform.

```

---

**R code Exa 16.2** Test data is whether in Poisson distributed

```

1 # Test data is whether in Poisson distributed :
2
3 no_of_arrival <- c(0,1,2,3,4,5)
4 obs_freq <- c(7,18,25,17,12,5)
5
6 # chi square value when alpha = 0.05 :
7 qchisq(.95,4)
8
9 for(i in 1:6){
10   arr_obs <- no_of_arrival*obs_freq
11 }
12 print(arr_obs)
13
14 l = sum(arr_obs)/sum(obs_freq)
15 l # lambda
16
17 # Expected probability using lambda and no_of_arrival
    :
18 exp_pb <- c(.1003,.2306,.2652,.2033,.1169,.0837)
19

```

```

20 for(i in 0:5){
21   exp_freq = sum(obs_freq)*exp_pb
22 }
23 print(exp_freq)
24
25 no_of_arrival <- cbind(no_of_arrival,obs_freq,arr_
   obs,exp_pb,exp_freq)
26 no_of_arrival
27
28 for(i in 0:5){
29   X = (obs_freq-exp_freq)^2/exp_freq
30 }
31 print(X)
32 sum(X)

```

---

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

```

1 # CHI SQUARE GOODNESS OF FIT TEST example 2 :
2 p <- c("Milk","non-Milk")
3 fo <- c(115,435)
4 fe <- c(93.5,456.5)
5
6 # critical value of chi-square :
7 qchisq(.95, df=1)
8
9 X_1 = (fo[1]-fe[1])^2/fe[1]
10 X_1
11
12 X_2 = (fo[2]-fe[2])^2/fe[2]
13 X_2
14
15 # Observed value of chi-square :
16 X_sq = X_1 + X_2
17 X_sq
18

```

```

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

```

---

#### R code Exa 16.4 CHI SQUARE TEST OF INDEPENDENCE

```

1 # CHI-SQUARE TEST OF INDEPENDENCE :
2
3 Age = matrix(c(26,95,18,41,40,20,24,13,32),nrow=3,
               ncol=3,byrow = TRUE)
4 dimnames(Age) = list(c("21-34","35-55",">55"),c("
    Coffee_tea ", "Soft_Drink", "Other"))
5 Age
6
7 # chi-square expected value when alpha =.01 :
8 qchisq(.99,df=4)
9
10 # The degrees of freedom are  $(3 - 1)(3 - 1) = 4$ , and
    the critical value is 13.2767.
11 # The decision rule is to reject the null hypothesis
    if the observed value of chisquare
12 # is greater than 13.2767.
13
14
15 # chi-square observed value :
16 # installed.pacakges("stats")
17 library(stats)
18 chisq.test(Age)
19
20 # The observed value of chi-square, 59.41, is
    greater than the critical value, 13.2767,
21 # so the null hypothesis is rejected.

```

---

# Chapter 17

## Nonparametric Statistics

R code Exa 17.1 Mann Whitney U test

```
1 # Mann-Whitney U test :
2
3 Total_emp_comp <- c
  (18.75,19.80,20.10,20.75,21.64,21.90,22.36,22.96,23.45,23.88,24.11)
4 Rank <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
5 Group <- c("H","H","H","H","E","H","H","H","E","E","E",
  "E","E","E","E","E")
6 Total_emp_comp <- data.frame(Total_emp_comp,Rank,
  Group)
7 Total_emp_comp
8
9 W1 = 1+2+3+4+6+7+8
10 W1
11 W2 = 5+9+10+11+12+13+14+15
12 W2
13 U1 = (7)*(8) + ((7)*(8))/2 - W1
14 U1
15 U2 = (7)*(8) + ((8)*(9))/2 - W2
16 U2
17
```

```

18 # Using Wilcox test :
19 wilcox.test(Total_emp_comp ~ Group, data = Total_emp
    _comp, exact = FALSE)
20
21 #Because U2 is the smaller value of U, we use U=3 as
    the test statistic for Table A.13.
22 # Because it is the smallest size , let n1=7; n2=8.
23
24 # Because the p-value is less than  $\alpha = .05$ , the
    null hypothesis is rejected.

```

---

#### R code Exa 17.2 LARGE SAMPLE FORMULAS MANN WHITNEY U TEST

```

1 #LARGE-SAMPLE FORMULAS MANN-WHITNEY U TEST :
2
3 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
    4.01, 4.05, 4.10, 4.10, 4.25, 4.29, 4.53, 4.75, 4.80, 4.80, 4.98, 5.
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
6 Group <- c('V', 'R', 'V', 'V', 'V', 'V', 'V', 'V', 'R', 'V', '
    V', 'R', 'V', 'R', 'R',
7
    'V', 'V', 'R', 'R', 'R', 'V', 'V', 'R', 'R', 'R', '
    R', 'R', 'R', 'R', 'R')
8 Value <- data.frame(Value, Rank, Group)
9 Value
10
11 W1 = 1 + 3 + 4.5 + 4.5 + 6 + 7 + 8 + 10 + 11 + 13 +
    16 + 17 + 21 + 22
12 W1
13

```

```

14 U = (14)*(16) + ((14)*(15))/2 - W1
15 U
16
17 U_u = ((14)*(16))/2
18 U_u
19
20 sd_u = sqrt(((14)*(16)*(31))/12)
21 sd_u
22
23 # observed value
24 z = (U-U_u)/sd_u
25 z
26
27 # Wilcox test :
28 wilcox.test(Value ~ Group, data = Value, exact =
    FALSE)

```

---

### R code Exa 17.3 WILCOXON MATCHED PAIRS SIGNED RANK TEST

```

1 # WILCOXON MATCHEDPAIRS SIGNED RANK TEST :
2
3 Worker <- c(1:20)
4 Before <- c
    (5,4,9,6,3,8,7,10,3,7,2,5,4,5,8,7,9,5,4,3)
5 After <- c
    (11,9,9,8,5,7,9,9,7,9,6,10,9,7,9,6,10,8,5,6)
6 d <- c
    (-6,-5,0,-2,-2,1,-2,1,-4,-2,-4,-5,-5,-2,-1,1,-1,-3,-1,-3)

7 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

8 Worker <- data.frame(Worker,Before,After,d,Rank)
9 Worker
10

```



```

11 # test statistic z value :
12 qnorm(.99,lower.tail = FALSE)
13
14 # T positive and negative using wilcox test function
15 :
16 wilcox.test(Worker$Before, Worker$After, paired=TRUE)
17
18 # T positive and negative using formula :
19 T_p <- 3.5+3.5+3.5
20 T_p
21 T_n <- 19 + 17 + 9 + 9 + 9 + 14.5 + 9 + 14.5 + 17 +
22      17 + 9 + 3.5 + 3.5 + 12.5 + 3.5 + 12.5
23
24 T_min = min(T_p,T_n)
25 T_min
26
27 n = 19
28 T_mean = (n*(n+1))/4
29 T_mean
30
31 T_sd = sqrt((n*(n+1)*(2*n+1))/24)
32 T_sd
33
34 # observed z value :
35 z = (T_min - T_mean)/T_sd
36 z
37
38 # The observed z value (-3.41) is in the rejection
39   region, so the analyst rejects the null
40   hypothesis.
41 # The productivity is signi???cantly greater after
42   the implementation of quality control
43 # at this company.

```

---

## R code Exa 17.4 KRUSKAL WALLIS TEST

```
1 # KRUSKAL-WALLIS TEST :
2
3 Group_native <- c(8,5,7,11,9,6)
4 Group_water <- c(10,12,11,9,13,12)
5 Group_fertilizer <- c(11,14,10,16,17,12)
6 Group_water_fertilizer <- c(18,20,16,15,14,22)
7 Group <- data.frame(Group_native, Group_water, Group_
   fertilizer, Group_water_fertilizer)
8 Group
9
10 # alpha = .01, critical value :
11 qchisq(.99, df=3)
12
13 native <- Group$Group_native
14 water <- Group$Group_water
15 fertilizer <- Group$Group_fertilizer
16 water_fertilizer <- Group$Group_water_fertilizer
17 x1 <- c(native, water, fertilizer, water_fertilizer)
18 x1
19 g <- factor(rep(1:4, c(6,6,6,6)),
20             labels = c("native",
21                        "water",
22                        "fertilizer",
23                        "water_fertilizer"))
24 kruskal.test(x1, g)
25
26
27 # The observed K value is 16.77 and the critical is
   11.3449.
28 # Because the observed value is greater than the
   table value, the null hypothesis
29 # is rejected. There is a significant difference in
```

the way the trees grow

---

### R code Exa 17.5 FRIEDMAN TEST

```
1 # FRIEDMAN TEST :
2
3 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)
4 Brand
5
6 # Chi-square value , alpha =0.01 :
7 qchisq(.99, df=4)
8
9 # observed value :
10 friedman.test(Brand)
11
12 # Because the observed value of = 3.68 is not
13 greater than the critical value, 13.2767,
14 # the researchers fail to reject the null hypothesis
15 .
```

---

### R code Exa 17.6 SPEARMANS RANK CORRELATION

```
1 # SPEARMAN'S RANK CORRELATION :
2
3 Crude_oil <- c
  (14.60,10.50,12.30,15.10,18.35,22.60,28.90,31.40,26.75)
4 Gasoline <- c
  (3.25,3.26,3.28,3.26,3.32,3.44,3.56,3.60,3.54)
```

```

5 Crude_rank <- c(3,1,2,4,5,6,8,9,7)
6 Gasoline_rank <- c(1,2.5,4,2.5,5,6,8,9,7)
7 d <- c(2,-1.5,-2,1.5,0,0,0,0,0)
8 d_sq <- c(4,2.25,4,2.25,0,0,0,0,0)
9 oil <- data.frame(Crude_oil, Gasoline, Crude_rank,
  Gasoline_rank, d, d_sq)
10 oil
11 d_sq_sum <- sum(d_sq)
12 d_sq_sum
13
14 # Using cor.test :
15 # install.packages("stats")
16 library(stats)
17 cor.test(oil$Crude_oil, oil$Gasoline, method = "
  spearman")
18
19 # using formula :
20 n = 9
21 r_s <- 1 - ((6*d_sq_sum)/(n*(n^2-1)))
22 r_s
23
24
25 # A high positive correlation is computed between
  the price of a barrel of
26 # West Texas intermediate crude and a gallon of
  regular unleaded gasoline.

```

---