

R Textbook Companion for
Introductory Business Statistics
by Alexander Holmes, Barbara Illowsky, Susan
Dean¹

Created by
Radha R
B.Sc.

Computer Science and Engineering
Shrimathi Devkunvar Nanalal Bhatt Vaishnav College for Women, Chennai

Cross-Checked by
R TBC Team

April 28, 2022

¹Funded 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: Introductory Business Statistics

Author: Alexander Holmes, Barbara Illowsky, Susan Dean

Publisher: OpenStax

Edition: 2

Year: 2018

ISBN: 1-947172-47-6

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
1 Sampling and Data	5
2 Descriptive Statistics	10
3 Probability Topics	34
4 Discrete Random Variables	62
5 Continuous Random Variables	69
6 The Normal Distribution	77
7 The Central Limit Theorem	85
8 Confidence Intervals	87
9 Hypothesis Testing with One Sample	95
10 Hypothesis Testing with Two Samples	107
11 The Chi Square Distribution	122
12 F Distribution and one way ANOVA	134
13 Linear Regression and Correlation	142

List of R Codes

Exa 1.14	Finding percentage of heights	5
Exa 1.15	Finding percentage of heights within a range	6
Exa 1.16	Finding percentage of heights within a range	6
Exa 1.17	Frequency table generation	7
Exa 1.18	Frequency table	8
Exa 2.1	Stem and leaf graph example1	10
Exa 2.2	Stem and leaf graph example2	10
Exa 2.3	Stem and Leaf plot	11
Exa 2.4	Frequency curve example1	11
Exa 2.5	Bar graph example1	12
Exa 2.6	bar graph example2	12
Exa 2.7	Two way table	13
Exa 2.8	Histogram graph example1	13
Exa 2.9	Histogram graph example2	14
Exa 2.10	Histogram graph example3	15
Exa 2.11	Frequency curve example2	16
Exa 2.12	Frequency curve example3	16
Exa 2.13	Time series graph	17
Exa 2.14	Measures of Location of Data Example1	19
Exa 2.15	Measures of Location of Data example2	20
Exa 2.16	Measures of Location of Data example3	22
Exa 2.17	Measures of Location of Data example4	23
Exa 2.18	Measures of Location of Data example5	24
Exa 2.19	Measures of Location of Data example6	25
Exa 2.23	survey for buying gym	26
Exa 2.24	Measure of Centre	28
Exa 2.25	Measures of the Center of the Data example2	28
Exa 2.26	Measures of the Center of the Data example3	28

Exa 2.28	Calculating the Arithmetic Mean of Grouped Frequency Tables example1	29
Exa 2.29	Measures of the Spread of the Data Example1	30
Exa 2.30	Measures of the Spread of the Data example2	30
Exa 2.31	Standard deviation of grouped frequency tables	32
Exa 2.32	Comparing values from different data sets	33
Exa 3.1	Finding probabilities example1	34
Exa 3.2	Finding probabilities example2	35
Exa 3.3	Finding probabilities example3	36
Exa 3.6	Mutually Exclusive Events of coins	37
Exa 3.7	Flipping of two coins and finding the probability	38
Exa 3.8	Rolling of six sided dice	39
Exa 3.9	Mutually Exclusive Events example1	40
Exa 3.10	Mutually Exclusive Events example2	41
Exa 3.11	Drawing of red and blue cards from a box	42
Exa 3.12	Mutually Exclusive Events example3	43
Exa 3.13	Tossing of fair coin	44
Exa 3.14	Two basic rules of Probability example1	45
Exa 3.15	Two basic rules of Probability example2	46
Exa 3.16	Two basic rules of Probability example3	46
Exa 3.17	Finding probabilities example4	47
Exa 3.18	Finding probabilities example5	48
Exa 3.19	Finding probabilities example6	49
Exa 3.20	Contingency tables example1	49
Exa 3.21	Contingency tables example2	50
Exa 3.22	Contingency tables example4	51
Exa 3.23	Contingency tables example3	52
Exa 3.24	drawing of three colored balls from urn	53
Exa 3.25	drawing of marbles without replacement	54
Exa 3.26	adoption of Tabby kittens and black kittens	55
Exa 3.28	drawing venn diagram for flipping of two fair coins	56
Exa 3.30	Venn diagram for selection of dogs	57
Exa 3.31	venn diagram for selection of dogs2	58
Exa 3.32	probability of selecting student as fresher or earning B grade	59
Exa 3.33	Two basic rules of Probability example4	60
Exa 4.1	hypergeometric Distribution	62
Exa 4.8	Geometric probability distribution function example1	63

Exa 4.9	Geometric probability distribution function example2 .	63
Exa 4.10	Geometric probability distribution function example3 .	64
Exa 4.11	Geometric probability distribution function example4 .	64
Exa 4.14	Poison Probability distribution example1	65
Exa 4.15	Poison Probability distribution example2	65
Exa 4.16	Poison Probability distribution example3	66
Exa 4.17	Poison Probability distribution example4	67
Exa 4.18	Poison Probability distribution example5	68
Exa 5.1	Continuous Probability Density functions example1 . .	69
Exa 5.2	Uniform Distribution example1	70
Exa 5.3	The Exponential Distribution example1	70
Exa 5.4	The Exponential Distribution example2	71
Exa 5.5	The Exponential Distribution example3	72
Exa 5.6	The Exponential Distribution example4	73
Exa 5.7	The Exponential Distribution example5	74
Exa 5.8	Exponential probability	75
Exa 6.1	The Standard Normal Distribution example1	77
Exa 6.2	Finding z scores for given mean and standard deviation	78
Exa 6.3	Using the Normal Distribution example1	79
Exa 6.4	Using the Normal Distribution example2	80
Exa 6.5	Using the Normal Distribution example3	81
Exa 6.6	Using the Normal Distribution example4	82
Exa 6.7	Finding binomial and normal distribution	83
Exa 7.1	Finite Population Correction Factor example1	85
Exa 7.2	Finite Population Correction Factor example2	85
Exa 8.1	Calculating the Confidence Intervals example1	87
Exa 8.2	Calculating the Confidence Intervals example2	87
Exa 8.3	Calculating the Confidence Intervals example3	89
Exa 8.4	Calculating the Confidence Intervals example4	90
Exa 8.5	Confidence interval for a small sample example1	91
Exa 8.6	Confidence interval for a population proportion example1	92
Exa 8.7	Confidence interval for a population proportion example2	93
Exa 8.8	Confidence interval for a population proportion example3	93
Exa 8.9	Calculating the sample size n continuous and binary random variables	94
Exa 9.8	Full Hypothesis test example1	95
Exa 9.9	Full Hypothesis test example2	96
Exa 9.10	Full Hypothesis test example3	98

Exa 9.11	Hypothesis test for proportions example1	99
Exa 9.12	Hypothesis test for proportions example2	101
Exa 9.13	Hypothesis test for proportions example3	103
Exa 9.14	Hypothesis test for proportions example4	104
Exa 10.1	Comparing two independent population means example1	107
Exa 10.2	Comparing two independent population means example2	109
Exa 10.3	Comparing two independent population means example3	110
Exa 10.4	Cohens standards for small medium and large effect sizes	111
Exa 10.5	Test for differences in means	112
Exa 10.6	Comparing two independent population proportions ex- ample1	113
Exa 10.7	Two population means with known standard deviations example1	114
Exa 10.8	Two population means with known standard deviations example2	116
Exa 10.9	Matched or Paired Samples	117
Exa 10.10	Matched or Paired Samples example2	118
Exa 10.11	Matched or Paired Samples example3	119
Exa 11.2	Test of single variance example1	122
Exa 11.3	Test of single variance example2	123
Exa 11.4	Goodness of Fit test example1	124
Exa 11.5	Goodness of Fit test example2	125
Exa 11.6	Goodness of Fit test example3	126
Exa 11.7	Goodness of Fit test example4	127
Exa 11.8	Goodness of Fit test example5	128
Exa 11.9	Test of Independence example1	129
Exa 11.10	Test of Independence example2	130
Exa 11.11	Test of Homogeneity	132
Exa 12.1	Test of two variances	134
Exa 12.2	The F distribution and the F Ratio example1	135
Exa 12.3	The F distribution and the F Ratio example2	136
Exa 12.4	The F distribution and the F Ratio example3	138
Exa 12.5	The F distribution and the F Ratio example4	140
Exa 13.2	Linear Equations example1	142
Exa 13.5	The Regression Equations example1	142
Exa 13.6	The Regression Equations example2	142

Chapter 1

Sampling and Data

R code Exa 1.14 Finding percentage of heights

```
1 #page 24
2 library(dplyr)
3 s_h<-data.frame(
4     lbound=c
5         (59.95,61.95,63.95,65.95,67.95,69.95,71.95,73.95)
6     ,
7     ubound=c
8         (61.95,63.95,65.95,67.95,69.95,71.95,73.95,75.95)
9     ,
10    freq=c(5,3,15,40,17,12,7,1))
11 View(s_h)
12 total=sum(s_h$freq)
13 print(total)
14 s_h$r_freq=s_h$freq/total
15 View(s_h)
16 s_h$cum_freq=cumsum(s_h$r_freq)
17 percent=s_h %>%
18   select(lbound,ubound,freq)%>%
19   filter(ubound<=65.95)
20 p_6595=(sum(percent$freq)/total)*100
21 print(paste("the solution is =",p_6595,"%"))
```

R code Exa 1.15 Finding percentage of heights within a range

```
1 #page 25
2 library(dplyr)
3 s_h<-data.frame(
4     lbound=c
5         (59.95,61.95,63.95,65.95,67.95,69.95,71.95,73.95)
6     ,
7     ubound=c
8         (61.95,63.95,65.95,67.95,69.95,71.95,73.95,75.95)
9     ,
10    freq=c(5,3,15,40,17,12,7,1))
11 View(s_h)
12 total=sum(s_h$freq)
13 print(total)
14 s_h$r_freq=s_h$freq/total
15 View(s_h)
16 s_h$cum_freq=cumsum(s_h$r_freq)
17 percent=s_h %>%
18   select(lbound,ubound,freq,r_freq)%>%
19   filter(lbound>=61.95 & ubound<=65.95)
20 p_6165=sum(percent$r_freq)
21 print(paste("the solution is =",p_6165))
```

R code Exa 1.16 Finding percentage of heights within a range

```
1 #page 25-26
2 library(dplyr)
3 s_h<-data.frame(
4     lbound=c
5         (59.95,61.95,63.95,65.95,67.95,69.95,71.95,73.95)
6     ,
```

```

5             ubound=c
              (61.95,63.95,65.95,67.95,69.95,71.95,73.95,75.95)
              ,
6             freq=c(5,3,15,40,17,12,7,1))
7 View(s_h)
8 total=sum(s_h$freq)
9 print(total)
10 s_h$r_freq=s_h$freq/total
11 View(s_h)
12 s_h$cum_freq=cumsum(s_h$r_freq)
13 percent=s_h %>%
14   dplyr::select(lbound,ubound,freq,r_freq)%>%
15   filter(lbound>=67.95 & ubound<=71.95)
16 p_6771=(sum(percent$r_freq))*100
17 print(paste("solution of a is =",p_6771,"%"))
18 percent=s_h %>%
19   dplyr::select(lbound,ubound,freq,r_freq)%>%
20   filter(lbound>=67.95 & ubound<=73.95)
21 p_6773=(sum(percent$r_freq))*100
22 print(paste("solution of b is =",p_6773,"%"))
23 percent=s_h %>%
24   dplyr::select(lbound,ubound,freq,r_freq)%>%
25   filter(lbound>=65.95)
26 p_65=(sum(percent$r_freq))*100
27 print(paste("solution of c is =",p_65,"%"))
28 percent=s_h %>%
29   dplyr::select(lbound,ubound,freq,r_freq)%>%
30   filter(lbound>=61.95 & ubound<=71.95)
31 n_p=sum(percent$freq)
32 print(paste("solution of d is =",n_p))

```

R code Exa 1.17 Frequency table generation

```

1 #page 26–27
2 library(dplyr)

```

```

3 library(MASS)
4 miles<-data.frame(data=c
      (2,3,4,5,7,10,12,13,15,18,20),
5                      freq=c(2,1,1,3,2,3,2,1,1,2,1))
6 View(miles)
7 total=sum(miles$freq)
8 print(total)
9 miles$r_freq=miles$freq/total
10 View(miles)
11 miles$cum_freq=cumsum(miles$r_freq)
12 filterdata<-filter(miles,data>=5 & data<=7)
13 ans1=sum(filterdata$freq)/total
14 print(paste("the answer c=",fractions(ans1)))
15 filterdata<-filter(miles,data>=12)
16 ans2=sum(filterdata$freq)/total
17 print(paste("the answer d is=",fractions(ans2)))
18 filterdata<-filter(miles,data<12)
19 ans3=sum(filterdata$freq)/total
20 print(paste("the answer d=",fractions(ans3)))
21 filterdata<-filter(miles,data>5 & data<13)
22 ans4=sum(filterdata$freq)/total
23 print(paste("the answer d=",fractions(ans4)))

```

R code Exa 1.18 Frequency table

```

1 #page 28
2 library(dplyr)
3 earthquake<-data.frame(year=c
      (2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012)
4                      ,
      deaths=c
      (231,21357,11685,33819,228802,88003,6605,712,88
      )
5 View(earthquake)
6 total=sum(earthquake$deaths)

```

```

7 earthquake$r_freq=earthquake$deaths/total
8 View(earthquake)
9 earthquake$cum_freq=cumsum(earthquake$r_freq)
10 q_a=earthquake %>%
11   dplyr::select(year,r_freq) %>%
12   filter(year>=2006 & year<=2009)
13 View(q_a)
14 freq6_9=sum(q_a$r_freq)
15 p_6_9=freq6_9*100
16 cat("the answer (a) is = ",round(p_6_9,1),"%")
17 q_b=earthquake %>%
18   dplyr::select(year,r_freq) %>%
19   filter(year>2009)
20 View(q_b)
21 freq_9=sum(q_b$r_freq)
22 p_9=freq_9*100
23 cat("the answer % (b) is= ",round(p_9,1),"%")
24 q_c=earthquake %>%
25   dplyr::select(year,r_freq) %>%
26   filter(year<=2003)
27 View(q_c)
28 freq_3=sum(q_c$r_freq)
29 p_3=freq_3*100
30 cat("the answer (c) is=",round(p_3,1),"%")
31 q_d=earthquake %>%
32   dplyr::select(year,r_freq) %>%
33   filter(year==2004)
34 View(q_d)
35 freq_4=sum(q_d$r_freq)
36 p_4=freq_4*100
37 cat("the % (d) is= ",round(p_4,1),"%")

```

Chapter 2

Descriptive Statistics

R code Exa 2.1 Stem and leaf graph example1

```
1 # Page NO : 46
2 df<-data.frame(scores=c
  (33,42,49,49,53,55,55,61,63,67,68,68,69,69,72,
3     73,74,78,80,83,88,88,88,
4     90,92,94,94,94,94,96,100))
5 View(df)
6 stem(df$scores)
```

R code Exa 2.2 Stem and leaf graph example2

```
1 # Page NO : 47
2 df<-data.frame(dist=c
  (1.1,1.5,2.3,2.5,2.7,3.2,3.3,3.3,3.5,3.8,4.0,4.2,4.5,4.5,4.7,4.8,5
3     6.5,6.7,12.3))
4 View(df)
5 stem(df$dist,scale=3)
```

R code Exa 2.3 Stem and Leaf plot

```
1 # Page NO : 47-49
2 library("aplpack")
3 df<-data.frame(a_I=c
4     (57,61,57,57,58,57,61,54,68,51,49,64,50,48,65,
5     52,56,46,54,49,51,47,55,55,54,42,51,56,55,51,
6     54,51,60,62,43,55,56,61,52,69,64,47,54,47)
7     )
8 df1<-data.frame(a_D=c
9     (67,90,83,85,73,80,78,79,68,71,53,65,74,64,77,
10    56,66,63,70,49,56,71,67,71,58,60,72,67,57,60,
11    90,63,88,78,46,64,81,93,93))
12 View(df)
13 View(df1)
14 stem.leaf.backback(df$a_I,df1$a_D,m=1)
15 #The answer provided in the textbook is wrong.
```

R code Exa 2.4 Frequency curve example1

```
1 #page 49-50
2 library(dplyr)
3 remainder<-data.frame(no_times=c(0,1,2,3,4,5),
4     frequency=c(2,5,8,14,7,4))
5 View(remainder)
6 plot(remainder$no_times,remainder$frequency,xlab="no
7     of times",
8     ylab="frequency ",type="
9     o",col="blue")
```

R code Exa 2.5 Bar graph example1

```
1 #page 51
2 library(dplyr)
3 library(ggplot2)
4 dtabl<-data.frame(
5     agegroups=c("13-25", "26-44", "45-64"),
6     no_of_users=c(66082280, 53300200, 27885100)
7     ,
8     proportion=c(45,36, 19))
9 View(dtabl)
10 ggplot(dtabl, aes(x=agegroups, y=proportion, color=cut))
11     )+
12     geom_bar(color="blue", stat="identity", width=0.5)+
13     labs(x="agegroups", y="%")
```

R code Exa 2.6 bar graph example2

```
1 #page 52-53
2 library(dplyr)
3 library(ggplot2)
4 dtabl<-data.frame(
5     race=c("1", "2", "3", "4", "5", "6"),
6     examinee=c(10.3, 9.0, 17.0, 0.6, 57.1, 6.0),
7     overall=c(5.7, 14.7, 17.6, 1.1, 59.2, 1.7))
8 View(dtabl)
9 ggplot(dtabl, aes(x=race, y=examinee), fill=race)+
10     geom_bar(color="blue", stat="identity", width=0.5)+
11     theme(legend.position="bottom")+
12     theme(legend.direction="vertical")+
13     theme(axis.text.x=element_text(angle=0))+
14     geom_text(aes(label=examinee))
```

R code Exa 2.7 Two way table

```
1 #page 54
2 library(dplyr)
3 pets<-data.frame(dogs=c(4,4,8),
4                  cats=c(2,6,8),
5                  fish=c(2,2,4),
6                  total=c(8,12,20))
7 rownames(pets)<-c("men", "women", "total")
8 last=ncol(pets)
9 total=0
10 for(c in 1:(last-1)){
11   x=(pets[1,c]/pets[1,last])
12   print(paste("number owning ", colnames(pets)[c], x))
13   total=total+x
14 }
15 print(paste("sum = ", total))
```

R code Exa 2.8 Histogram graph example1

```
1 #page 55-56
2 library(ggplot2)
3 dtabl<-data.frame(height=c
4                   (59.95,60,60.5,61,61,61.5,63.5,63.5,63.5,64,64,64,64,64,64,
5                   64.5,64.5,64.5,64.5,64.5,64.5,64.5,64.5,
6                   66,66,66,66,66,66,66,66,66,66,
7                   66.5,66.5,66.5,66.5,66.5,66.5,66.5,66.5,66.5,66.5,66.5,66.5,
8                   67,67,67,67,67,67,67,67,67,67,67,
9                   67.5,67.5,67.5,67.5,67.5,67.5,67.5,
```

```

9         68,68,
10        69,69,69,69,69,69,69,69,69,69,
11        69.5,69.5,69.5,69.5,69.5,
12        70,70,70,70,70,70,
13        70.5,70.5,70.5,
14        71,71,71,
15        72,72,72,
16        72.5,72.5,
17        73,
18        73.5,
19        74,75.95))
20 View(dtbl)
21 breaks=seq(59.95,75.95,by=2)
22 dcut=cut(dtbl$height,breaks,right=FALSE)
23 dout=as.data.frame(table(dcut))
24 dout<-transform(dout,cumFreq=cumsum(Freq),relative=
      round(prop.table(Freq),2))
25 ggplot(dout,aes(dcut,relative))+
26   geom_col(fill="purple")+
27   geom_text(aes(label=relative))+
28   xlab("height")+ylab("r.freq")
29 #The answer may slightly vary due to rounding off
      values.

```

R code Exa 2.9 Histogram graph example2

```

1 #page 56-57
2 library(ggplot2)
3 dtbl<-data.frame(height=c(1,1,1,1,1,1,1,1,1,1,1,1,
4                           2,2,2,2,2,2,2,2,2,2,2,
5                           3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,
6                           4,4,4,4,4,4,4,
7                           5,5,5,5,5,5,
8                           6,6))

```

```

9 View(dtbl)
10 breaks=seq(0.5,6.5,by=1)
11 dcut=cut(dtbl$height,breaks,right=FALSE)
12 dout=as.data.frame(table(dcut))
13 dout<-transform(dout,cumFreq=cumsum(Freq),relative=
    round(prop.table(Freq),2))
14 nobars=(6.5-0.5)/1
15 print(paste("the number of bars =",nobars))
16 ggplot(dout,aes(dcut,Freq))+
17   geom_col(fill="purple")+
18   geom_text(aes(label=Freq))+
19     xlab("no. of books")+ylab("freq")

```

R code Exa 2.10 Histogram graph example3

```

1 #page 57-58
2 library(dplyr)
3 library(ggplot2)
4 dtbl<-data.frame(hours=c(9.95,10,2.25,16.75,0,
5                          19.5,22.5,7.5,15,12.75,
6                          5.5,11,10,20.75,17.5,
7                          23,21.9,24,23.75,18,
8                          20,15,22.9,18.8,20.5))
9 bins=seq(-0.01,25,by=5)
10 scores=cut(dtbl$hours,bins)
11 table(scores)
12 View(dtbl)
13 View(scores)
14 freq_table=transform(table(scores))
15 xx=transform(freq_table,Rel_freq=prop.table(Freq),
    cumfreq=cumsum(Freq))
16 View(xx)
17 ggplot(data=xx,
18        aes(x=scores,y=Freq))+
19   geom_col(color="red",fill="blue")+

```

```
20     labs(y="no. of students",x="hours spent")+
21     scale_y_continuous(breaks=seq(0,10,1))
```

R code Exa 2.11 Frequency curve example2

```
1 #page no 59
2 library(ggplot2)
3 df<-data.frame(lbound=c
4     (39.5,49.5,59.5,69.5,79.5,89.5,99.5),
5     ubound=c
6     (49.5,59.5,69.5,79.5,89.5,99.5,109.5)
7     ,
8     freq=c(0,5,10,30,40,15,0),
9     cfreq=c(0,5,15,45,85,100,0))
10 View(df)
11 df$mid=(df$lbound+df$ubound)/2
12
13 ggplot(df,aes(x=mid,y=freq))+
14   geom_line(aes(x=mid,y=freq),size=1)+
15   geom_point(aes(mid,size=0.5),shape=18,show.legend=
16     FALSE)+
17   scale_x_continuous(breaks=seq(from=44.5,to=104.5,
18     by=10))+
19   xlab("result")+ylab("number of students")
```

R code Exa 2.12 Frequency curve example3

```
1 #page no: 60-61
2 library(ggplot2)
3 df<-data.frame(lbound=c
4     (39.5,49.5,59.5,69.5,79.5,89.5,99.5),
```

```

4         ubound=c
           (49.5,59.5,69.5,79.5,89.5,99.5,109.5)
           ,
5         freq=c(0,5,10,30,40,15,0),
6         cfreq=c(0,5,15,45,85,100,0))
7 View(df)
8 df$mid=(df$lbound+df$ubound)/2
9 df2=data.frame(lbound=c
10               (39.5,49.5,59.5,69.5,79.5,89.5,99.5),
                ubound=c
11               (49.5,59.5,69.5,79.5,89.5,99.5,109.5)
                ,
12               freq=c(0,10,10,30,45,5,0),
                cfreq=c(0,10,20,50,95,100,0))
13 df2$mid=(df2$lbound+df2$ubound)/2
14 View(df2)
15 ggplot(df, aes(x=mid, y=freq))+
16   geom_line(aes(x=mid, y=freq), color="pink", size=1,
17             show.legend=FALSE)+
18   geom_point(aes(mid, size=0.5), shape=18, show.legend=
19              FALSE)+
20   geom_line(data=df2, aes(x=mid, y=freq), color="blue
21                    ", size=1, show.legend=FALSE)+
22   geom_point(aes(mid, size=0.5), shape=18, show.legend=
23              FALSE)+
24   scale_x_continuous(breaks=seq(44.5,104.5,10))+
25   scale_y_continuous(breaks=seq(0,50,5))+
26   xlab("result")+ylab("number of students")

```

R code Exa 2.13 Time series graph

```

1 #page 62-63
2 library(ggplot2)
3 consumer<-data.frame(year=c
4                       (2003,2004,2005,2006,2007,2008,2009,2010,2011,2012))

```

```

4      ,
      jan=c
        (181.7,185.2,190.7,198.3,202.416,211.080,211.143,216.687
5      ,
      feb=c
        (183.1,186.2,191.8,198.7,203.499,211.693,212.193,216.741
6      ,
      mar=c
        (184.2,187.4,193.3,199.8,205.352,213.528,212.709,217.631
7      ,
      apr=c
        (183.8,188,194.6,201.5,206.686,214.823,213.24,218.009,22
8      ,
      may=c
        (183.5,189.1,194.4,202.5,207.949,216.632,213.856,218.178
9      ,
      jun=c
        (183.7,189.7,194.5,202.9,208.352,218.815,215.693,217.965
10     ,
      jul=c
        (183.9,189.4,195.4,203.5,208.299,219.964,215.351,218.011
11     ,
      aug=c
        (184.6,189.5,196.4,203.9,207.917,219.086,215.834,218.312
12     ,
      sep=c
        (185.2,189.9,198.8,202.9,208.49,218.783,215.969,218.439,
13     ,
      oct=c
        (185,190.9,199.2,201.8,208.936,216.573,216.177,218.711,2
14     ,
      nov=c
        (184.5,191,197.6,201.5,210.177,212.425,216.33,218.803,22
15     ,
      dec=c
        (184.3,190.3,196.8,201.8,210.036,210.228,215.949,219.179
16     ,
      annual=c

```

```

        (184,188.9,195.3,201.6,207.342,215.303,214.537,218.056,2
        )
17 View(consumer)
18 min<-as.Date("2003-1-1")
19 max<-as.Date("2012-12-30")
20 yrs<-as.Date(ISOdate(consumer$year,1,1))
21 ggplot(data=consumer,aes(x=yrs,y=annual))+
22   geom_line(size=1)+
23   scale_x_date(limits=c(min,max))+
24   xlab("year")+ylab("annual price")

```

R code Exa 2.14 Measures of Location of Data Example1

```

1 #page no : 65-66
2 library(base)
3 library(dplyr)
4 price<-data.frame(
5   dollars=c
6     (389950,230500,158000,479000,639000,114950,550000,387000
7     575000,488800,1095000))
8 View(price)
9 M=median(price$dollars)
10 print(paste("median = ",M))
11 price1=price %>%
12   select(dollars) %>%
13   arrange(dollars)
14 View(price1)
15 rowcount=nrow(price1)
16 midindex=(rowcount+1)/2
17 if(floor(midindex)!=midindex)
18 {
19   l<-(midindex-1)/2
20   u<-(midindex+1)/2
21 } else

```

```

21 {
22   l<-(midindex-1)
23   u<-(midindex+1)
24 }
25 q1=median(price1$dollars[1:l])
26 print(paste("Q1 = ",q1))
27 q3=median(price1$dollars[u:rowcount])
28 print(paste("Q3 = ",q3))
29 IQR1=q3-q1
30 print(paste("IQR = ",IQR1))
31 o_IQR1=(1.5)*IQR1
32 loutlier=q1-o_IQR1
33 print(paste("lower outlier =",loutlier))
34 uoutlier=q3+o_IQR1
35 print(paste("upper outlier =",uoutlier))
36 if(min(price1$dollars)>loutlier)
37 {
38   print("No outliers in the lower end of price")
39 } else
40 {
41   print("Outliers in the lower end of price")
42   print(paste("the outlier value is = ",min(price1$dollars)))
43 }
44 if(max(price1$dollars)<uoutlier)
45 {
46   print("No outliers in the upper end of price")
47 } else
48 {
49   print("Outliers in the upper end of price")
50   print(paste("the outlier value is = ",max(price1$dollars)))
51 }

```

R code Exa 2.15 Measures of Location of Data example2


```

1 #page no : 66
2 library(dplyr)
3 t_s<-data.frame(
4     minimum=c(32,25.5),
5     q1=c(56,78),
6     median=c(74.5,81),
7     q3=c(82.5,89),
8     maximum=c(99,98)
9 )
10 rownames(t_s)<-c("day","night")
11 View(t_s)
12 t_s$IQR=t_s$q3-t_s$q1
13 t_s$outlierq1=t_s$q1-(t_s$IQR)*1.5
14 t_s$outlierq3=t_s$q3+(t_s$IQR)*1.5
15 result1=t_s %>% select(q1,q3,minimum,maximum,
16     outlierq1,outlierq3) %>%
17     filter(row.names(t_s)== "day")
18 if(result1$minimum>result1$outlierq1)
19 {
20   print("No outliers in the lower end of day")
21 } else
22 {
23   print("Outliers in the lower end of day")
24   print(paste("any value < than", result1$outlierq1,
25     " is outlier"))
26   print(paste("the value=",result1$minimum," is
27     outlier"))
28 }
29 if(result1$maximum<result1$outlierq3)
30 {
31   print("No outliers in the upper end of day")
32 } else
33 {
34   print("Outliers in the upper end of day")
35   print(paste("any value > than", result1$outlierq3,
36     " is outlier"))
37   print(paste("the value=",result1$maximum," is
38     outlier"))

```



```

5 dtabl1<-data.frame(
6     hours=c(4,5,6,7,8,9,10),
7     freq=c(2,5,7,12,14,7,3))
8 total=sum(dtabl1$freq)
9 dtabl1$rfreq=dtabl1$freq/total
10 dtabl1$cum_freq=cumsum(dtabl1$rfreq)
11 View(dtabl1)
12 print(paste("28% is",quantile(dtabl,0.28)))
13 print(paste("median is",quantile(dtabl,0.50)))
14 print(paste("75% is",quantile(dtabl,0.75)))
15 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 2.17 Measures of Location of Data example4

```

1 #page no : 68
2 library(base)
3 dtabl=c
4     (4,4,5,5,5,5,5,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7,7,7,7,7,7,8,8,8,8,8,8,8,
5     9,9,9,9,9,9,9,10,10,10)
6 dtabl1<-data.frame(
7     hours=c(4,5,6,7,8,9,10),
8     freq=c(2,5,7,12,14,7,3))
9 total=sum(dtabl1$freq)
10 dtabl1$rfreq=dtabl1$freq/total
11 dtabl1$cum_freq=cumsum(dtabl1$rfreq)
12 View(dtabl1)
13 dtabl_len=length(dtabl)
14 find_percentile<-function(val,dtabl_len)
15 {
16     i=(val/100)*(dtabl_len+1)
17     lvalue=floor(i)
18     uvalue=ceiling(i)
19     x=c(lvalue,uvalue)

```

```

19   return(x)
20 }
21 test1=find_percentile(80,dtabl_len)
22 per_80=(dtabl[test1[1]]+dtabl[test1[2]])/2
23 test2=find_percentile(90,dtabl_len)
24 per_90=(dtabl[test2[1]]+dtabl[test2[2]])/2
25 test3=find_percentile(25,dtabl_len)
26 per_25=(dtabl[test3[1]]+dtabl[test3[2]])/2
27 print(paste("80th percentile is",per_80))
28 print(paste("90th percentile is",per_90))
29 print(paste("first quartile is",per_25))

```

R code Exa 2.18 Measures of Location of Data example5

```

1 #page no : 68-69
2 library(base)
3 dtabl=c
4     (18,21,22,25,26,27,29,30,31,33,36,37,41,42,47,52,55,57,58,62,64,67,
5     71,72,73,74,76,77)
6 dtabl_len=length(dtabl)
7 find_percentile<-function(val,dtabl_len)
8 {
9   i=(val/100)*(dtabl_len+1)
10  lvalue=floor(i)
11  uvalue=ceiling(i)
12  x=c(lvalue,uvalue)
13  return(x)
14 }
15 test1=find_percentile(70,dtabl_len)
16 per_70=(dtabl[test1[1]]+dtabl[test1[2]])/2
17 test2=find_percentile(83,dtabl_len)
18 per_83=(dtabl[test2[1]]+dtabl[test2[2]])/2
19 print(paste("70th percentile is",per_70))
20 print(paste("83rd percentile is",per_83))

```

R code Exa 2.19 Measures of Location of Data example6

```
1 #page no : 69
2 library(base)
3 dtabl=c
   (18,21,22,25,26,27,29,30,31,33,36,37,41,42,47,52,55,57,58,62,64,67,
4     71,72,73,74,76,77)
5 dtabl_len=length(dtabl)
6 find_range<-function(val)
7 {
8   count=0
9   for(i in 1:dtabl_len)
10  {
11    if(dtabl[i]<val)
12    {
13      count=count+1
14    }
15    else
16    {
17      break
18    }
19  }
20  return(count)
21 }
22 find_percentile<-function(val)
23 {
24   x=find_range(val)
25   y=1
26   i=((x+0.5*y)/(dtabl_len))*100
27   return(round(i))
28 }
29 per_58=find_percentile(58)
30 per_25=find_percentile(25)
```

```
31 print(paste("58 is",per_58,"th percentile"))
32 print(paste("25 is",per_25,"th percentile "))
```

R code Exa 2.23 survey for buying gym

```
1 #page no : 71
2 library(dplyr)
3 minutes<-data.frame(
4     x=c
5         (0,40,60,30,60,10,45,30,300,90,30,120,60,0,20)
6     )
7 View(minutes)
8 M=median(minutes$x)
9 minutes1=minutes %>%
10   select(x) %>%
11   arrange(x)
12 View(minutes1)
13 rowcount=nrow(minutes1)
14 midindex=(rowcount+1)/2
15 if(floor(midindex)!=midindex)
16 {
17   l<-(midindex-1)/2
18   u<-(midindex+1)/2
19 } else
20 {
21   l<-(midindex-1)
22   u<-(midindex+1)
23 }
24 q1=median(minutes1$x[1:l])
25 print(paste("minimum=",min(minutes)))
26 print(paste(" (Q1) = ",q1," minutes"))
27 print(paste("median = ",M))
28 q3=median(minutes1$x[u:rowcount])
29 print(paste(" (Q3) = ",q3," minutes"))
30 IQR1=q3-q1
```

```

29 print(paste("maximum=", max(minutes)))
30 o_IQR1=(1.5)*IQR1
31 loutlier=q1-o_IQR1
32 uoutlier=q3+o_IQR1
33 minutes3<-data.frame(
34   x=c(0,40,60,30,60,10,45,30,90,30,120,60,0,20))
35 View(minutes3)
36 M=median(minutes3$x)
37 minutes4=minutes3 %>%
38   select(x) %>%
39   arrange(x)
40 View(minutes4)
41 rowcount=nrow(minutes4)
42 if(rowcount%%2==0)
43 {
44   midindex=rowcount/2
45 }else
46 {
47   midindex=(rowcount+1)/2
48 }
49 if(floor(midindex)!=midindex)
50 {
51   l<-(midindex-1)/2
52   u<-(midindex+1)/2
53 } else
54 {
55   l<-(midindex-1)
56   u<-(midindex+1)
57 }
58 q1=median(minutes4$x[1:l])
59 print(paste("minimum=", min(minutes3)))
60 print(paste(" (Q1) = ", q1, " minutes"))
61 q3=median(minutes4$x[u:rowcount])
62 print(paste(" (Q3) = ", q3, " minutes"))
63 IQR1=q3-q1
64 print(paste("maximum=", max(minutes3)))
65 #The answer provided in the textbook is wrong.

```

R code Exa 2.24 Measure of Centre

```
1 #page no : 72-73
2 library(dplyr)
3 aids<-c
   (3,4,8,8,10,11,12,13,14,15,15,16,16,17,17,18,21,22,22,24,24,25,26
4
   27,27,29,29,31,32,33,33,34,34,35,37,40,44,44,47)

5 aids_mean=mean(aids)
6 aids_median=median(aids)
7 print(paste("mean = ",round(aids_mean,1)))
8 print(paste("median = ",aids_median))
```

R code Exa 2.25 Measures of the Center of the Data example2

```
1 #page no : 73
2 earning<-c()
3 for(i in 1:49)
4 {
5   earning[i]=30000
6 }
7 earning[50]=5000000
8 earning_mean=mean(earning)
9 earning_median=median(earning)
10 print(paste("the mean is ",earning_mean))
11 print(paste("the median is ",earning_median))
```

R code Exa 2.26 Measures of the Center of the Data example3


```

1 #page no : 73
2 library(dplyr)
3 student<-c
      (50,53,59,59,63,63,72,72,72,72,7276,78,81,83,84,84,84,90,93)

4 mode<-function(x)
5 {
6   ux<-unique(x)
7   tab<-tabulate(match(x,ux))
8   return(ux[tab==max(tab)])
9 }
10 s_mode=mode(student)
11 print(paste("the MODE is ",s_mode))

```

R code Exa 2.28 Calculating the Arithmetic Mean of Grouped Frequency Tables example1

```

1 #page no : 74-75
2 s_test<-data.frame(
3     grade_l=c
      (50,56.5,62.5,68.5,74.5,80.5,86.5,92.5)
4     ,
      grade_u=c
      (56.5,62.5,68.5,74.5,80.5,86.5,92.5,98.5)
5     ,
      s_no=c(1,0,4,4,2,3,4,1))
6 s_test$midpoint=(s_test$grade_u+s_test$grade_l)/2
7 s_test$fm=s_test$midpoint*s_test$s_no
8 sum_fm=sum(s_test$fm)
9 sum_s_no=sum(s_test$s_no)
10 mean_value=sum_fm/sum_s_no
11 print(paste("the sum of product is ",round(sum_fm,2)
12 ))
12 print(paste("the mean is ",round(mean_value,2)))

```

R code Exa 2.29 Measures of the Spread of the Data Example1

```
1 #page no : 81–82
2 library(dplyr)
3 avg_ages<-data.frame(ages=c
4     (9,9.5,9.5,10,10,10,10,10.5,10.5,10.5,10.5,
5     11,11,11,11,11,11,11,11.5,11.5,11.5)
6     )
7 View(avg_ages)
8 ages1=avg_ages %>%
9     group_by(ages)%>%
10    mutate(frequency=n()) %>%
11    select(ages,frequency)%>%
12    distinct()
13 View(ages1)
14 total=sum(ages1$frequency)
15 ages1$val1=ages1$ages*ages1$frequency
16 xbar=sum(ages1$val1)/total
17 View(ages1)
18 ages1$dev1=ages1$ages-xbar
19 ages1$devsq=ages1$dev1*ages1$dev1
20 ages1$fd=ages1$frequency*ages1$devsq
21 View(ages1)
22 sv=sum(ages1$fd)/(total-1)
23 sd=sqrt(sv)
24 print(paste("the sample mean is =",round(xbar,3)))
25 print(paste("the sample variance is =",round(sv,4)))
26 print(paste("the sample sd is =",round(sd,2)))
```

R code Exa 2.30 Measures of the Spread of the Data example2

```
1 #page no : 83–84
```

```

2 library(dplyr)
3 calculus<-data.frame(marks=c
      (33,42,49,49,53,55,55,61,63,67,68,68,69,69,72,73,74,78,
4
          80,83,88,88,88,90,92,94,94,94,94,96,100)
          )
5 View(calculus)
6 calc1=calculus %>%
7   group_by(marks)%>%
8   mutate(frequency=n()) %>%
9   select(marks ,frequency)%>%
10  distinct()
11 View(calc1)
12 total=sum(calc1$frequency)
13 calc1$r_freq=round((calc1$frequency/total),3)
14 calc1$c_freq=round((cumsum(calc1$r_freq)),3)
15 View(calc1)
16 calc1$val1=calc1$marks*calc1$frequency
17 xbar=sum(calc1$val1)/total
18 View(calc1)
19 calc1$dev1=calc1$marks-xbar
20 calc1$devsq=calc1$dev1*calc1$dev1
21 calc1$fd=calc1$frequency*calc1$devsq
22 View(calc1)
23 sv=sum(calc1$fd)/(total-1)
24 sd=sqrt(sv)
25 rowcount=nrow(calculus)
26 midindex=(rowcount+1)/2
27 if(floor(midindex)!=midindex)
28 {
29   l<-(midindex-1)/2
30   u<-(midindex+1)/2
31 } else
32 {
33   l<-(midindex-1)
34   u<-(midindex+1)
35 }
36 q1=median(calculus$marks[1:l])

```

```

37 q3=median(calculus$marks[u:rowcount])
38 IQR1=q3-q1
39 mean1=mean(calculus$marks)
40 median1=median(calculus$marks)
41 print(paste("the sample mean is =",round(mean1,1)))
42 print(paste("the sample sd is =",round(sd,1)))
43 print(paste("the median is =",median1))
44 print(paste("the 1st quartile is =",q1))
45 print(paste("the 3rd quartile is =",q3))
46 print(paste("the IQR is =",IQR1))

```

R code Exa 2.31 Standard deviation of grouped frequency tables

```

1 #page no : 84-85
2 dtabl<-data.frame(lvalue=c(0,3,6,9,12),
3                   uvalue=c(2,5,8,11,14),
4                   f=c(1,6,10,7,0))
5
6 dtabl$m=(dtabl$lvalue+dtabl$uvalue)/2
7 total=sum(dtabl$f)
8 dtabl$fm=dtabl$f*dtabl$m
9 xbar=sum(dtabl$fm)/total
10 View(dtabl)
11 dtabl$dev1=dtabl$m-xbar
12 dtabl$devsq=dtabl$dev1*dtabl$dev1
13 dtabl$fmxbar=dtabl$f*dtabl$devsq
14 View(dtabl)
15 sv=sum(dtabl$fmxbar)/(total-1)
16 sd=sqrt(sv)
17 print(paste("the variance is =",round(sv,2)))
18 print(paste("the sample sd is =",round(sd,2)))
19 #The answer provided in the textbook is wrong.

```

R code Exa 2.32 Comparing values from different data sets

```
1 #page no : 85
2 dtabl<-data.frame(name=c(" John", " Ali"),
3                   GPA=c(2.85,77),
4                   meangpa=c(3.0,80),
5                   schoolsd=c(0.7,10))
6 dtabl$z1=(dtabl$GPA-dtabl$meangpa)/dtabl$schoolsd
7 View(dtabl)
8 no=nrow(dtabl)
9 for(i in 1:no)
10 {
11 print(paste("sd of = ",dtabl$name[i]," is",round(
12           dtabl$z1[i],2)))
13 }
14 if(dtabl$z1[1]>dtabl$z1[2])
15 {
16 print(paste(dtabl$name[1]," has more GPA"))
17 } else
18 {
19 print(paste(dtabl$name[2]," has more GPA"))
20 }
```

Chapter 3

Probability Topics

R code Exa 3.1 Finding probabilities example1

```
1 #page no :135
2 library(MASS)
3 S=c(1:19)
4 A=c(2,4,6,8,10,12,14,16,18)
5 B=c(14,15,16,17,18,19)
6 cat("a. S= ",S)
7 cat("b. A= ",A)
8 cat("b. B= ",B)
9 P_A=length(A)/length(S)
10 print(paste(" c. P(A) =",fractions(P_A)))
11 P_B=length(B)/length(S)
12 print(paste("c. P(B) =",fractions(P_B)))
13 ABint=intersect(A,B)
14 ABunion=union(A,B)
15 cat("d. AintB =",ABint)
16 cat("d. AUB=",ABunion)
17 P_AintB=length(ABint)/length(S)
18 print(paste("e. P(AintB) =",fractions(P_AintB)))
19 P_AUB=length(ABunion)/length(S)
20 print(paste("e. P(AUB) =",fractions(P_AUB)))
21 Acomp=setdiff(S,A)
```

```

22 cat("f. A' =", Acomp)
23 P_Acomp=length(Acomp)/length(S)
24 print(paste("f. P(A') =",fractions(P_Acomp)))
25 print(paste("g. P(A)+P(A') =",P_A+P_Acomp))
26 P_A_B=P_AintB/P_B
27 P_B_A=P_AintB/P_A
28 print(paste("h. P(A|B) = ",fractions(P_A_B)))
29 print(paste("h. P(B|A) = ",fractions(P_B_A)))
30 if(P_A_B==P_B_A){
31   print("h. P(A|B) is equal to P(B|A)")
32 }else {
33   print("h. P(A|B) not equal to P(B|A)")
34 }

```

R code Exa 3.2 Finding probabilities example2

```

1 #page no :136–137
2 library(MASS)
3 S=c(1:6)
4 T=c(2)
5 A=c(2,4,6)
6 B=c(1,2,3)
7 P_T=length(T)/length(S)
8 cat("a. T =",T)
9 print(paste("a. P(T) =",fractions(P_T)))
10 P_A=length(A)/length(S)
11 cat("b. A =",A)
12 print(paste("b. P(A) =",fractions(P_A)))
13 P_B=length(B)/length(S)
14 cat("c. B =",B)
15 print(paste("c. P(B) =",fractions(P_B)))
16 Acomp=setdiff(S,A)
17 cat("d. A' =", Acomp)
18 P_Acomp=length(Acomp)/length(S)
19 print(paste("d. P(A') =",fractions(P_Acomp)))

```

```

20 ABint=intersect(A,B)
21 ABunion=union(A,B)
22 P_AintB=length(ABint)/length(S)
23 P_A_B=P_AintB/P_B
24 P_B_A=P_AintB/P_A
25 print(paste("e.P(A|B) =",fractions(P_A_B)))
26 print(paste("f.P(B|A) =",fractions(P_B_A)))
27 cat("g. AintB =",ABint)
28 print(paste("g. P(AintB) =",fractions(P_AintB)))
29 P_AUB=length(ABunion)/length(S)
30 cat("h. AUB =",ABunion)
31 print(paste("h. P(AUB) =",fractions(P_AUB)))
32 Bcomp=setdiff(S,B)
33 ABunion=union(A,Bcomp)
34 cat("i. AUB' =",ABunion)
35 PABunion=length(ABunion)/length(S)
36 print(paste("i. P(AUB') =",fractions(PABunion)))
37 N=c(2,3,5)
38 PN=length(N)/length(S)
39 cat("j. N =",N)
40 print(paste("j. P(N) =",fractions(PN)))
41 print("k. P(7)=0")

```

R code Exa 3.3 Finding probabilities example3

```

1 #page no :137-138
2 library(MASS)
3 S=data.frame(sex=c('M','F'),
4              RH=c(43,44),
5              LH=c(9,4))
6 S$rtotal=S$RH+S$LH
7 View(S)
8 total=sum(S$rtotal)
9 P_M=S$rtotal[1]/total
10 P_F=S$rtotal[2]/total

```



```

11 P_R=sum(S$RH)/total
12 P_L=sum(S$LH)/total
13 table=round(prop.table(S[,2:3]),2)
14 table=as.matrix(table)
15 table
16 RintM=table[1]
17 FintL=table[4]
18 LintF=table[4]
19 MunionF=sum(table)
20 MunionR=table[1]+table[2]+table[3]
21 FunionL=table[3]+table[4]+table[2]
22 csum=apply(table[,1:2],2,sum)
23 rsum=apply(table[,1:2],1,sum)
24 P_Mdash=sum(table)-rsum[1]
25 P_RintM=round(RintM/P_M,4)
26 P_FintL=round(FintL/P_L,4)
27 P_LintF=round(LintF/P_F,4)
28 print(paste(" a. P(M) =",P_M))
29 print(paste(" b. P(F) =",P_F))
30 print(paste(" c. P(R) =",P_R))
31 print(paste(" d. P(L) =",P_L))
32 print(paste(" e. P(M int R) =",RintM))
33 print(paste(" f. P(F int L) =",FintL))
34 print(paste(" g. P(M U F) =",MunionF))
35 print(paste(" h. P(M U R) =",MunionR))
36 print(paste(" i. P(F U L) =",FunionL))
37 print(paste(" j. P(M') =",P_Mdash))
38 print(paste(" k. P(R|M) =",P_RintM))
39 print(paste(" l. P(F|L) =",P_FintL))
40 print(paste(" m. P(L|F) =",P_LintF))

```

R code Exa 3.6 Mutually Exclusive Events of coins

```

1 # page no : 140
2 library(prob)

```

```

3 library(MASS)
4 coin=tosscoin(2)
5 A=subset(coin,toss1=="H" | toss2=="H")
6 P_A=nrow(A)/nrow(coin)
7 print(paste(" one tail =",fractions(P_A)))
8 B=subset(coin,toss1=="T" & toss2=="T")
9 P_B=1-P_A
10 print(paste(" all tail =",fractions(P_B)))
11 C=subset(coin,toss1=="H" & toss2=="H")
12 P_C=nrow(C)/nrow(coin)
13 print(paste(" all heads =",fractions(P_C)))
14 BintC=intersect(B,C)
15 P_BintC=nrow(BintC)/nrow(coin)
16 D=subset(coin,toss1=="T" & toss2=="T")
17 P_D=nrow(D)/nrow(coin)
18 print(paste("more than one tail =",fractions(P_D)))
19 E=subset(coin,toss1=="H" & (toss1=="H" | toss2=="T"))
20 P_E=nrow(E)/nrow(coin)
21 print(paste("head-first roll =",fractions(P_E)))
22 F=subset(coin,toss1=="T" | toss2=="T")
23 P_F=nrow(F)/nrow(coin)
24 print(paste("one tail-two filps =",fractions(P_F)))

```

R code Exa 3.7 Flipping of two coins and finding the probability

```

1 # page no : 141
2 library(prob)
3 library(MASS)
4 coin=tosscoin(2)
5 F=subset(coin,toss1=="T" | toss2=="T")
6 P_F=nrow(F)/nrow(coin)
7 print(paste("a. P(F)=",fractions(P_F)))
8 G=subset(coin,(toss1=="T" & toss2=="T") |(toss1=="H"
    & toss2=="H"))
9 P_G=nrow(G)/nrow(coin)

```

```

10 print(paste("b. P(G) =", fractions(P_G)))
11 H=subset(coin, toss1=="H" & (toss1=="H" | toss2=="T"))
12 P_H=nrow(H)/nrow(coin)
13 print(paste("c.P(H) =", fractions(P_H)))
14 FintG=intersect(F,G)
15 P_FintG=nrow(FintG)/nrow(coin)
16 if(P_FintG==0)
17 {
18   print("d. F,G-mutually exclusive")
19 }else
20 {
21   print("d. F,G-not mutually exclusive")
22 }
23 J=subset(coin, toss1=="T" & toss2=="T")
24 JintH=intersect(J,H)
25 P_JintH=nrow(JintH)/nrow(coin)
26 if(P_JintH==0)
27 {
28   print("e. J,H-mutually exclusive")
29 }else
30 {
31   print("e. J,H-not mutually exclusive")
32 }

```

R code Exa 3.8 Rolling of six sided dice

```

1 # page no : 141-142
2 library(prob)
3 library(MASS)
4 library(sets)
5 S=rolldie(1)
6 A=subset(S, (X1 %% 2 != 0))
7 B=subset(S, (X1 %% 2 == 0))
8 P_A=nrow(A)/nrow(S)
9 print(paste("P(A) =", fractions(P_A)))

```

```

10 P_B=nrow(B)/nrow(S)
11 print(paste("P(B) =",fractions(P_B)))
12 C=subset(S,(X1 %% 2!=0) & (X1>2))
13 D=subset(S,(X1 %% 2==0) & (X1<5))
14 CintD=intersect(C,D)
15 P_CintD=length(CintD)/nrow(S)
16 if(P_CintD==0)
17 {
18   print("C,D-mutually exclusive")
19 }else
20 {
21   print("C,D-not mutually exclusive")
22 }
23 E=subset(S,(X1 < 5))
24 CintE=intersect(C,E)
25 P_CintE=length(CintE)/nrow(S)
26 if(P_CintE==0)
27 {
28   print("C,E-mutually exclusive")
29 }else
30 {
31   print("C,E-not mutually exclusive")
32 }
33 CintA=intersect(C,A)
34 P_CintA=length(CintA)/nrow(S)
35 P_C_A=P_CintA/P_B
36 print(paste(" P(C|A) = ",fractions(P_C_A)))

```

R code Exa 3.9 Mutually Exclusive Events example1

```

1 #page no:142-143
2 P_G=0.6
3 P_H=0.5
4 PGintH=0.3
5 P_G_H=PGintH/P_H

```

```

6  if(P_G_H==P_G)
7  {
8    print(paste("a. P(G|H) = P(G)"))
9  }else
10 {
11   print(paste("a. P(G|H) != P(G)"))
12 }
13 PGH=P_G*P_H
14 if(PGintH==PGH)
15 {
16   print(paste("b. P(G int H) = P(G)P(H)"))
17 }else
18 {
19   print(paste("b. P(G int H) != P(G)P(H)"))
20 }

```

R code Exa 3.10 Mutually Exclusive Events example2

```

1  #page no:143
2  P_C=0.75
3  P_D=0.3
4  P_C_D=0.75
5  PCintD=0.225
6  if(P_C_D==P_C)
7  {
8    print(paste('independent ,P(C|D)=P(C)'))
9  }else
10 {
11     print(paste('dependent P(C|D)!=P(C)'))
12 }
13 if(PCintD==0)
14 {
15   print(paste('mutually exclusive P(C int D)=0'))
16 }else
17 {

```

```

18   print(paste('not mutually exclusive P(C int D) !=0'
19             ))
19 }
20 P_D_C=PCintD/P_C
21 print(paste("c. P(D|C) =", P_D_C))

```

R code Exa 3.11 Drawing of red and blue cards from a box

```

1 # page no : 144
2 library(prob)
3 library(MASS)
4 S<-c('R1', 'R2', 'R3', 'B1', 'B2', 'B3', 'B4', 'B5')
5 R1<-c('R1', 'R2', 'R3')
6 B1<-c('B1', 'B2', 'B3', 'B4', 'B5')
7 E1<-c('R2', 'B2', 'B4')
8 G1<-c('B4', 'B5')
9 H1<-c('B1', 'B2', 'B3', 'B4')
10 P_R=length(R1)/length(S)
11 print(paste("P(R) =", fractions(P_R)))
12 P_B=length(B1)/length(S)
13 print(paste("P(B) =", fractions(P_B)))
14 P_E=length(E1)/length(S)
15 print(paste("P(E) =", fractions(P_E)))
16 EintB=intersect(E1, B1)
17 P_EintB=length(EintB)/length(S)
18 P_E_B=P_EintB/P_B
19 print(paste("The P(E|B) =", fractions(P_E_B)))
20 BintE=intersect(B1, E1)
21 P_BintE=length(BintE)/length(S)
22 P_B_E=P_BintE/P_E
23 print(paste("The P(B|E) =", fractions(P_B_E)))
24 RintB=intersect(R1, B1)
25 P_RintB=length(RintB)/length(S)
26 if(P_RintB==0)
27 {

```

```

28   print("R,B-mutually exclusive")
29 }else
30 {
31   print("R,B-not mutually exclusive")
32 }
33 P_G=length(G1)/length(S)
34 print(paste("P(G)=",fractions(P_G)))
35 P_H=length(H1)/length(S)
36 print(paste("P(H)=",fractions(P_H)))
37 GintH=intersect(G1,H1)
38 P_GintH=length(GintH)/length(S)
39 P_G_H=P_GintH/P_H
40 print(paste("The P(G|H) =",fractions(P_G_H)))
41 if(P_G==P_G_H)
42 {
43   print("G,H-independent")
44 }else
45 { print(" G,H-not independent")
46   }

```

R code Exa 3.12 Mutually Exclusive Events example3

```

1 #page no:145
2 P_F=0.60
3 P_L=0.5
4 PFintL=0.45
5 P_L_F=0.75
6 PFL=P_F*P_L
7 if(PFintL==PFL)
8 {
9   print(paste("a. female ,long hair-independent"))
10 }else
11 {
12   print(paste("a. female ,long hair-dependent"))
13 }

```

```

14 if(P_L_F==P_L)
15 {
16   print(paste("b. female ,long hair-independent"))
17 }else
18 {
19   print(paste("b. female ,long hair-dependent"))
20 }

```

R code Exa 3.13 Tossing of fair coin

```

1 # page no : 145-146
2 library(prob)
3 library(MASS)
4 S=tosscoin(1)
5 cat(paste("a. solution =",S$toss1[1] ,"and " ,S$toss1
      [2],";",nrow(S)))
6 S1=rolldie(1)
7 writeLines("\n b.")
8 print(S1$X1)
9 print(paste("b." ,nrow(S1)))
10 print(paste("c." , nrow(S1)*2))
11 cat(paste("d. ="))
12 count=0
13 for(val1 in 1:nrow(S1))
14 {
15   for(val in 1:nrow(S))
16   {
17     count=count+1
18     cat(paste(S$toss1[val] ,S1$X1[val1])," ")
19   }
20 }
21 A=c('H2' , 'H4' , 'H6')
22 P_A=length(A)/count
23 writeLines("\n")
24 print(paste("e. P(A) =", fractions(P_A)))

```



```

25 B=c( 'H3' )
26 P_B=length(B)/count
27 print(paste("f. P(B) =", fractions(P_B)))
28 AintB=intersect(A,B)
29 P_AintB=length(AintB)/count
30 if(P_AintB==0)
31 {
32   print("g. A,B-mutually exclusive")
33 }else
34 {
35   print("g. A,B-not mutually exclusive")
36 }
37 righteq=P_A*P_B
38 lefteq=P_AintB
39
40 if(lefteq==righteq)
41 {
42   print("h. A,B-independent")
43 }else
44 { print("h. A,B-dependent")
45 }

```

R code Exa 3.14 Two basic rules of Probability example1

```

1 #page no:147
2 P_A=0.60
3 P_B=0.35
4 PAintB=0
5 PUB=P_A+P_B
6 PUBcomp=1-PUB
7 print(paste("PUB=", PUB))
8 print(paste("1-PUB =", PUBcomp))

```

R code Exa 3.15 Two basic rules of Probability example2

```
1 #page no:147-148
2 P_A=0.65
3 P_B=0.65
4 P_B_A=0.90
5 PAintB=P_B_A*P_A
6 print(paste("a. solution =",PAintB))
7 PUB=P_A+P_B-PAintB
8 print(paste("b. solution = ",PUB))
9 PBA=P_A*P_B
10 if(PAintB==PBA)
11 {
12   print(paste("c. A,B-independent"))
13 }else
14 {
15   print(paste("c. A,B-not independent"))
16 }
17 if(PAintB==0)
18 {
19   print(paste("d. A,B-mutually exclusive"))
20 }else
21 {
22   print(paste("d. A,B-not mutually exclusive"))
23 }
```

R code Exa 3.16 Two basic rules of Probability example3

```
1 #page no:148-149
2 library(MASS)
3 N=150
4 A_swim=75
5 A_swim1=40
6 I_swim=47
7 I_swim1=30
```

```

8 N_swim=N-(A_swim+I_swim)
9 N_swim1=10
10 P_N_swim=N_swim/N
11 print(paste("a. solution = ",N_swim,"/",N))
12 P_swim_4=(A_swim1+I_swim1+N_swim1)/N
13 print(paste("b. solution = ",A_swim1+I_swim1+N_swim1
  ,"/",N))
14 P_A_swim1=A_swim1/N
15 print(paste("c. solution = ",A_swim1,"/",N))
16 PAintI=0
17 PNint4=0.0667
18 P=P_N_swim*P_swim_4
19 if(PAintI==0)
20 {
21   print(paste("d. A and B are mutually exclusive"))
22 }else
23 {
24   print(paste("d. A and B are not mutually exclusive
  "))
25 }
26 if(PNint4==P)
27 {
28   print(paste("e. A And B are independent"))
29 }else
30 {
31   print(paste("e.A And B are not independent"))
32 }

```

R code Exa 3.17 Finding probabilities example4

```

1 #page no: 149
2 P_M=0.2
3 P_S=0.65
4 P_M_S=0.25
5 PMintS=P_M_S*P_S

```

```

6 print(paste("a. solution=",PMintS))
7 PMunionS=P_M+P_S-PMintS
8 print(paste("b. solution=",PMunionS))
9 if(P_M_S==P_M){
10   print("c. M and S are independent")
11 }else
12 {
13   print("c. M and S are not independent")
14 }
15 if(PMintS==0){
16   print("d. M and S are mutually exclusive")
17 }else
18 {
19   print("d. M and S are not mutually exclusive")
20 }

```

R code Exa 3.18 Finding probabilities example5

```

1 #page no:150
2 P_B=14.3/100
3 P_N=85/100
4 P_N_B=2/100
5 PBintN=P_B*P_N_B
6 PBunionN=P_B+P_N-PBintN
7 print(paste(" a. P(B) =",P_B))
8 print(paste(" a. P(N) =",P_N))
9 print(paste(" b. P(N|B) =",P_N_B))
10 print(paste(" c. P(B int N) =",round(PBintN,4)))
11 print(paste(" d. P(B U N) =",round(PBunionN,4)))
12 if(P_N==P_N_B){
13   print("e. B and N are independent")
14 }else
15 {
16   print("e. B and N are not independent")
17 }

```

```

18 if(PBintN==0){
19   print("f. B and N are mutually exclusive")
20 }else
21 {
22   print("f. B and N are not mutually exclusive")
23 }

```

R code Exa 3.19 Finding probabilities example6

```

1 #page no:151
2 P_B=14.3/100
3 P_N=85/100
4 P_N_B=2/100
5 PBintN=P_B*P_N_B
6 PUnionN=P_B+P_N-PBintN
7 P_P_B=1-P_N_B
8 PBintP=P_P_B*P_B
9 P_Bcomp=1-P_B
10 P_P=1-P_N
11 print(paste(" a. P(P|B) =",P_P_B))
12 print(paste(" b. P(B int P) =",round(PBintP,4)))
13 print(paste(" c. P(B') =",P_Bcomp))
14 print(paste(" d. P(P) =",P_P))

```

R code Exa 3.20 Contingency tables example1

```

1 #page no:152-153
2 library(MASS)
3 df=data.frame(violation=c(25,45,70),
4               noviolation=c(280,405,685),
5               rtotal=c(305,450,755))
6 rownames(df)=c(" cell", " nocell", " cttotal")
7 View(df)

```

```

8 P_D_C=df$rtotal[1]/df$rtotal[3]
9 P_D_NV=df$noviolation[3]/df$rtotal[3]
10 PNvintC=df$noviolation[1]/df$rtotal[3]
11 e=df$violation[1]/df$violation[3]
12 f=df$noviolation[2]/df$noviolation[3]
13 term1=(df$rtotal[1]/df$rtotal[3])+(df$noviolation[3]
      /df$rtotal[3])
14 term2=(df$noviolation[1]/df$rtotal[3])
15 PC_U_NV=term1-term2
16 print(paste("a. solution=",df$rtotal[1],"/",df$rtotal
      [3]))
17 print(paste("b. solution=",df$noviolation[3],"/",df$
      rtotal[3]))
18 print(paste("c. solution=",df$noviolation[1],"/",df$
      rtotal[3]))
19 print(paste("d. solution=",fractions(PC_U_NV)))
20 print(paste("e. solution=",df$violation[1],"/",df$
      violation[3]))
21 print(paste("f. solution=",df$noviolation[2],"/",df$
      rtotal[2]))
22 #"The answer may vary due to difference in
      representation."

```

R code Exa 3.21 Contingency tables example2

```

1 #page no:153-155
2 library(MASS)
3 df=data.frame(coast=c(18,16,34),
4               lakes=c(16,25,41),
5               peaks=c(11,14,25),
6               rtotal=c(45,55,100))
7 rownames(df)=c("F","M","ctotal")
8 View(df)
9 PFintC=df$coast[1]/df$rtotal[3]
10 P_F=df$rtotal[1]/df$rtotal[3]

```

```

11 P_C=df$coast[3]/df$rtotal[3]
12 PFC=P_F*P_C
13 print(paste("b.1 P(FintC) =",PFintC))
14 print(paste("b.2 P(F)P(C) =",PFC))
15 if(PFintC==PFC){
16   print("b. F,C-independent")
17 }else
18 {
19   print("b. F,C-not independent")
20 }
21 P_M_L=df[[2]][2]/df$lakes[3]
22 print(paste("c.2 P(M|L)= ",fractions(P_M_L)))
23 P_P=df[[3]][3]/df$rtotal[3]
24 print(paste("d.1 P(F)= ",fractions(P_F)))
25 print(paste("d.2 P(P)= ",fractions(P_P)))
26 PFintP=df$peaks[1]/df$rtotal[3]
27 print(paste("d.3 P(F int P)= ",fractions(PFintP)))
28 FUP=(P_F+P_P)-PFintP
29 print(paste("d.4 P(FUP)= ",fractions(FUP)))
30 #The answer may vary due to difference in
    representation.

```

R code Exa 3.22 Contingency tables example4

```

1 #page no:155-156
2 library(MASS)
3 df=data.frame(D1=c(1/15,4/15),
4               D2=c(1/12,3/12),
5               D3=c(1/6,1/6))
6 rs <- rowSums(df)
7 df <- cbind(df,rs)
8 cs <- colSums(df)
9 df_s <- sum(df)
10 df <- rbind(df,c(cs,df_s))
11 rownames(df)=c("caught", "Ncought", "ctotal")

```

```

12 View(df)
13 b=fractions(df$rtotal[2])
14 print(paste("b. solution= ",fractions(b)))
15 P_CD1=df$D1[1]
16 P_CD2=df$D2[1]
17 c = P_CD1 + P_CD2
18 print(paste("c. solution= ",fractions(c)))
19 #The answer provided in the textbook is wrong.

```

R code Exa 3.23 Contingency tables example3

```

1 #page no:156-157
2 library(MASS)
3 df=data.frame(robbery=c(145.7,133.1,119.3,113.7),
4               burg=c(732.1,717.7,701,702.2),
5               rape=c(29.7,29.1,27.7,26.8),
6               vehicle=c(314.7,259.2,239.1,229.6))
7 rownames(df)=c("2008","2009","2010","2011")
8 View(df)
9 df$rtotal=rowSums(df)
10 cttotal<-c(colSums(df[]))
11 total=cttotal[5]
12 P_8=df$rtotal[1]/total
13 P_9=df$rtotal[2]/total
14 P_10=df$rtotal[3]/total
15 P_11=df$rtotal[4]/total
16 P_R=cttotal[1]/total
17 P_B=cttotal[2]/total
18 P_Ra=cttotal[3]/total
19 P_V=cttotal[4]/total
20 P9intR=df$robbery[2]/total
21 P10intB=df$burg[3]/total
22 P11intRa=df$rape[4]/total
23 P8intV=df$vehicle[1]/total
24 P10UB=(P_10+P_B)-P10intB

```



```

25 P11_Ra=P11intRa/P_Ra
26 PV_8=P8intV/P_8
27 print(paste("a. solution=",round(P9intR,4)))
28 print(paste("b. solution=",round(P10intB,4)))
29 print(paste("c. solution=",round(P10UB,4)))
30 print(paste("d. solution=",round(P11_Ra,4)))
31 print(paste("e. solution=",round(PV_8,4)))

```

R code Exa 3.24 drawing of three colored balls from urn

```

1 #page no:157-159
2 library("DiagrammeR")
3 library(data.tree)
4 library("prob")
5 library("dplyr")
6 library("MASS")
7 x<-Node$new(" ")
8   firstdraw_blue<-x$AddChild("8B")
9     seconddraw_blue<-firstdraw_blue$AddChild("8B")
10       seconddraw_red<-firstdraw_blue$AddChild("3R")
11         thirddraw_blue<-seconddraw_blue$AddChild("64BB")
12           thirddraw_red<-seconddraw_red$AddChild("24BR")
13     firstdraw_red<-x$AddChild("3R")
14       seconddraw_blue<-firstdraw_red$AddChild("8B")
15       seconddraw_red<-firstdraw_red$AddChild("3R")
16         thirddraw_blue<-seconddraw_blue$AddChild("24RB")
17         thirddraw_red<-seconddraw_red$AddChild("9RR")
18 print(x)
19 plot(x)
20 balls_R <- c(rep("R1",1),rep("R2",1),rep("R3",1))

```

```

21 balls_B <- c(rep("B1",1),rep("B2",1),rep("B3",1),rep
  ("B4",1),rep("B5",1),rep("B6",1),rep("B7",1),rep(
  "B8",1))
22 apply(as.data.frame(expand.grid(balls_B,balls_R)),1,
  function(x){paste(x,collapse = "")})
23 M<-unique(urnsamples(balls,size=2,replace=TRUE,
  ordered=TRUE))
24 N<-probspace(M)
25 P_RR=(3/11)*(3/11)
26 print(paste("b. P(RR)=",fractions(P_RR)))
27 P_RB_U_BR=((3/11)*(8/11))+((8/11)*(3/11))
28 print(paste("c. P(RR U BR)=",fractions(P_RB_U_BR)))
29 P_R1_int_B2=((3/11)*(8/11))
30 print(paste("d. P(R1 int B2)=",fractions(P_R1_int_B2
  )))
31 P_R2_int_B1=(24/(24+64))
32 print(paste("e. P(R2| B1)=",fractions(P_R2_int_B1)))
33 P_BB=64/(64+24+24+9)
34 print(paste("f. P(BB)=",fractions(P_BB)))
35 P_B2_int_R1=(24/(9+24))
36 print(paste("g.(B2 | R1)=",fractions(P_B2_int_R1)))

```

R code Exa 3.25 drawing of marbles without replacement

```

1 #page no:160–161
2 library(data.tree)
3 library("prob")
4 library("dplyr")
5 library("MASS")
6 x<-Node$new(" ")
7   firstdraw_blue<-x$AddChild("B \n 8/11")
8     seconddraw_blue<-firstdraw_blue$AddChild("B \
9       n 7/10")
10       seconddraw_red<-firstdraw_blue$AddChild("R \n
11         3/10")

```

```

10         thirddraw_blue<-seconddraw_blue$AddChild(
           "56/110 \n BB")
11         thirddraw_red<-seconddraw_red$AddChild("
           24/110 \n BR")
12     firstdraw_red<-x$AddChild("R \n 3/11")
13         seconddraw_blue<-firstdraw_red$AddChild("B \n
           8/10")
14         seconddraw_red<-firstdraw_red$AddChild("R \n
           2/10")
15         thirddraw_blue<-seconddraw_blue$AddChild
           ("24/110 \n RB")
16         thirddraw_red<-seconddraw_red$AddChild("
           6/110 \n RR")
17
18     print(x)
19     plot(x)
20     P_RR=(3/11)*(2/10)
21     print(paste("a. P(RR) =",fractions(P_RR)))
22     P_RB_U_BR=((3/11)*(8/10))+((8/11)*(3/10))
23     print(paste("b. P(RR U BR)=",fractions(P_RB_U_BR)))
24     P_R2_int_B1=(3/10)
25     print(paste("c.P(R2 | B1)=",fractions(P_R2_int_B1)))
26     P_R1_int_B2=(3/11)*(8/10)
27     print(paste("d.P(R1 int B2)=",fractions(P_R1_int_B2
           )))
28     P_BB=(8/11)*(7/10)
29     print(paste("e. P(BB)=",fractions(P_BB)))
30     P_B2_int_R1=(8/10)
31     print(paste("f. P(B on 2nd | R on 1st)=",fractions(P
           _B2_int_R1)))

```

R code Exa 3.26 adoptiong of Tabby kittens and black kittens

```

1 #pageno 162–163
2 library(data.tree)

```

```

3 library("prob")
4 library("dplyr")
5 library("MASS")
6 x<-Node$new(" ")
7   firstdraw_blue<-x$AddChild("T \n 4/9")
8     seconddraw_blue<-firstdraw_blue$AddChild("T \n
9       3/8")
10     seconddraw_red<-firstdraw_blue$AddChild("B \n
11       5/8")
12     thirddraw_blue<-seconddraw_blue$AddChild("
13       TT")
14     thirddraw_red<-seconddraw_red$AddChild("
15       TB")
16   firstdraw_red<-x$AddChild("B \n 5/9")
17     seconddraw_blue<-firstdraw_red$AddChild("T \n
18       4/8")
19     seconddraw_red<-firstdraw_red$AddChild("B \n
20       4/8")
21     thirddraw_blue<-seconddraw_blue$AddChild
22       ("BT")
23     thirddraw_red<-seconddraw_red$AddChild("
24       BB")
25
26 print(x)
27 plot(x)
28 P_TT=(4/9)*(3/8)
29 print(paste("a. =",fractions(P_TT)))
30 P_T_B=((4/9)*(5/8))+((5/9)*(4/8))
31 print(paste("b. =",fractions(P_T_B)))
32 P_B_T=(4/8)
33 print(paste("c. =",fractions(P_B_T)))
34 P_TT_BB=((4/9)*(3/8))+((5/9)*(4/8))
35 print(paste("d. =",fractions(P_TT_BB)))

```

R code Exa 3.28 drawing venn diagram for flipping of two fair coins

```

1 #page number : 164-165
2 library(RAM)
3 library(VennDiagram)
4 A=c("TT", "TH")
5 B=c("TT", "HT")
6 AintB=c("TT")
7 AUB=c("TH", "TT", "HT")
8 X=c("HH", "HT", "TH", "TT")
9 v<-venn.diagram(list(A=A,B=B),
10                 fill=c("orange", "blue"),
11                 alpha=c(0.5,0.5), cat.cex=1.5, cex
12                 =1.5,
13                 filename=NULL)
14 grid.newpage()
15 grid.draw(v)
16 lapply(v, names)
17 lapply(v, function(i) i$label)
18 v[[5]]$label<-paste(setdiff(A,B), collapse="\n")
19 v[[6]]$label<-paste(setdiff(B,A), collapse="\n")
20 v[[7]]$label<-paste(intersect(A,B), collapse="\n")
21 grid.newpage()
22 grid.text(x=c(0.5,0), y=c(0.1,0), label=paste(setdiff(
    X, c(A,B)), collapse=" "))
23 grid.draw(v)

```

R code Exa 3.30 Venn diagram for selection of dogs

```

1 #page no:166_167
2 library(eulerr)
3 library(grid)
4 VennDiag <- euler(c("C" = 0.40, "PT" = .50, "C&PT" =
    0.05))
5 p1<-plot(VennDiag, counts = TRUE, font=1, cex=1,
    alpha=0.5,
6         fill=c("grey", "blue", "red"))

```

```

7 grid.text("C int PT",x=0.5,y=0.9)
8 gridExtra::grid.arrange(p1)
9 P_C=0.40
10 P_PT=0.50
11 print(paste(" P(C)=",P_C))
12 print(paste(" P(PT)=",P_PT))
13 CintPT=0.05
14 print(paste(" P(C\u2229PT)=",CintPT))
15 C_PT=CintPT/P_PT
16 print(paste("P(C|PT)=",C_PT))
17 CUPT=P_C+P_PT-CintPT
18 print(paste("P(C U PT)=",CUPT))

```

R code Exa 3.31 venn diagram for selection of dogs2

```

1 #page no:168_169
2 library(eulerr)
3 library(grid)
4 VennDiag <- euler(c("Male" = 12, "Female" = 8, "Male
  &Female" = 0.00))
5 p1<-plot(VennDiag, counts = TRUE, font=1, cex=1,
  alpha=0.5,
6   fill=c("grey", "blue", "red"))
7 VennDiag <- euler(c("Male" = 12, "Brown" = 10, "Male
  &Brown" = 0.83))
8 p2<-plot(VennDiag, counts = TRUE, font=1, cex=1,
  alpha=0.5,
9   fill=c("grey", "blue", "red"))
10 grid.text("10",x=0.5,y=0.5)
11 VennDiag <- euler(c("Female" = 8, "White Fur" = 5, "
  Female&White Fur" = 0.625))
12 p3<-plot(VennDiag, counts = TRUE, font=1, cex=1,
  alpha=0.5,
13   fill=c("blue", "white"))
14 gridExtra::grid.arrange(p1,p2,p3)

```

R code Exa 3.32 probability of selecting student as fresher or earning B grade

```
1 #page no:170_172
2 library(eulerr)
3 library(grid)
4 library(multipanelfigure)
5 S=50
6 A=20
7 B=15
8 sopho=30
9 P_A=A/S
10 P_B=B/S
11 P_AintB=(5/S)
12 P_AUB=P_A+P_B-P_AintB
13 VennDiag <- euler(c("A" = A, "B" = B, "A&B" = 0.10))
14 p1<-plot(VennDiag, counts = TRUE, font=1, cex=1,
           alpha=0.5,
           fill=c("grey", "lightgray", "red"))
15 VennDiag <- euler(c("A" = A, "B" = B, "A&B" = 0.10))
16 p2<-plot(VennDiag, counts = TRUE, font=1, cex=1,
           alpha=0.5,
           fill=c("blue", "lightgray", "blue"))
17 VennDiag <- euler(c("A" = A, "B" = B, "A&B" = 0.10))
18 p3<-plot(VennDiag, counts = TRUE, font=1, cex=1,
           alpha=0.5,
           fill=c("lightgray", "blue", "blue"))
19 VennDiag <- euler(c("A" = A, "B" = B, "A&B" = 0.10))
20 p4<-plot(VennDiag, counts = TRUE, font=1, cex=1,
           alpha=0.5,
           fill=c("lightgray", "lightgray", "blue"))
21
22 gridExtra::grid.arrange(p1,p2,p3,p4, nrow=2)
23 grid.text("P(AUB)", x=0.25, y=.95)
```

```

28 grid.text("=", x=0.50, y=.75)
29 grid.text("P(A)", x=0.56, y=.95)
30 grid.text("+", x=0.99, y=.75)
31 grid.text("P(B)", x=0.35, y=.50)
32 grid.text("-", x=0.50, y=.25)
33 grid.text("P(A\u2229B)", x=0.75, y=.50)

```

R code Exa 3.33 Two basic rules of Probability example4

```

1 #page no:172-173
2 library(MASS)
3 df=data.frame(LHG=c(5,8),
4               HG=c(15,12),
5               C=c(40,30),
6               CG=c(60,30))
7 rownames(df)=c("M", "F")
8 View(df)
9 df$rtotal=rowSums(df)
10 cttotal<-c(colSums(df[]))
11 total=cttotal[5]
12 PCGintF=df$CG[2]/total
13 P_F=df$rtotal[2]/total
14 P_CG=cttotal[4]/total
15 PCGUF=(P_F+P_CG)-PCGintF
16 PHGintM=df$HG[1]/total
17 P_M=df$rtotal[1]/total
18 P_HG_M=PHGintM/P_M
19 P_HG=cttotal[2]/total
20 LHS1=PCGintF
21 RHS1=P_CG*P_F
22 print(paste("a. solution=", round(PCGintF, 2)))
23 print(paste("b. solution=", round(PCGUF, 2)))
24 print(paste("c. solution=", round(P_HG_M, 3)))
25 if(LHS1!=RHS1)
26 {

```



```
27     print("d. gender and education are not independent
        ")
28 }else
29 {
30     print("d. gender and education are independent")
31 }
32 LHS2=P_HG_M
33 RHS2=P_HG
34 if(LHS2!=RHS2)
35 {
36     print("d. gender and education are not independent
        ")
37 }else
38 {
39     print("d. gender and education are independent")
40 }
```

Chapter 4

Discrete Random Variables

R code Exa 4.1 hypergeometric Distribution

```
1 #page no: 206
2 A=30
3 gd=20
4 x=5
5 k=10
6 factorial=function(x)
7 {
8   fact=1
9   for(i in 1:x)
10  {
11    fact=fact*i
12  }
13  return(fact)
14 }
15 binomial=function(a,x)
16 {
17   b=factorial(a)/(factorial(x)*factorial(a-x))
18   return(b)
19 }
20 term1=binomial(A,x)
21 term2=binomial(gd,k-x)
```

```

22 term3=binomial(A+gd,k)
23 num=term1*term2
24 P=num/term3
25 print(paste(" 1. solution =",round(P,3)))
26 P1=dhyper(x,A,gd,k)
27 print(paste(" 2. solution (built-in function) =",round(
    P1,3)))

```

R code Exa 4.8 Geometric probability distribution function example1

```

1 #page no: 211-212
2 library(ggplot2)
3 p=0.02
4 x=7
5 P_7=((1-p)^(x-1))*p
6 print(paste(" a. P(x=7)= ",round(P_7,4)))
7 m=1/p
8 print(paste(" mean =",round(m,4)))
9 pp<-0
10 for(i in 1:51)
11 {
12   pp[i]=((1-p)^(i-1))*p
13 }
14 variance=(1/p)*((1/p)-1)
15 sd=sqrt(variance)
16 print(paste(" the variance=",variance))
17 print(paste(" the sd=",round(sd,1)))
18 x1=seq(1,51,by=1)
19 barplot(pp,names.arg=x1,xlab="x",ylab="P(X=x)")

```

R code Exa 4.9 Geometric probability distribution function example2

```

1 #page no: 212-213

```

```

2 p=1.28/100
3 x=9
4 P_9=((1-p)^x)*p
5 print(paste("a. P(x=9)= ",round(P_9,4)))
6 x=20
7 P_20=((1-p)^(x-1))*p
8 print(paste("b. P(x=20)= ",round(P_20,2)))
9 m=(1-p)/p
10 print(paste("c. mean= ",round(m,2)))
11 sd=sqrt((1-p)/(p^2))
12 print(paste("d sd= ",round(sd,2)))

```

R code Exa 4.10 Geometric probability distribution function example3

```

1 #page no: 213
2 p=0.320
3 x=3
4 P_3=((1-p)^(x-1))*p
5 print(paste("a. P(x=3)= ",round(P_3,4)))
6 m=(1/p)
7 print(paste("c. mean= ",round(m,2)))

```

R code Exa 4.11 Geometric probability distribution function example4

```

1 #page no: 213
2 p=0.80
3 x=3
4 P_3=((1-p)^(x))*p
5 print(paste("a. P(x=3)= ",round(P_3,4)))

```

R code Exa 4.14 Poisson Probability distribution example1

```
1 #page no :215–216
2 library(ggplot2)
3 mu=(1/8)*6
4 x=c(0,1,2,3,4,5)
5 xtot=0
6 p=mu
7 for(i in 1:length(x))
8 {
9   xtot=xtot+ppois(x[i],lambda=mu,lower=FALSE)
10 }
11 y=ppois(1,lambda=mu,lower.tail=FALSE)
12 print(paste('P(x>1)',round(y,4)))
13 pp<-0
14 for(i in 1:6)
15 {
16   pp[i]=dpois(x[i],lambda=mu)
17 }
18 x1=c(0:5)
19 barplot(pp, names.arg=x1, beside=TRUE, xlab="x", ylab="P
  (X=x)", space=0, ylim=range(pretty(c(0,pp))))
```

R code Exa 4.15 Poisson Probability distribution example2

```
1 #page no :216
2 mu=7
3 e=2.718
4 P_2=dpois(2,lambda=mu)
5 print(paste('P(x=2)',round(P_2,3)))
6 x<-c(0,1,2)
7 y=0
8 for(i in 1:length(x))
9 {
10   y=y+dpois(x[i],lambda=mu)
```

```

11 }
12 print(paste('P(x \u2264 2)', round(y,4)))
13 sd=sqrt(7)
14 print(paste('\u03c3 =', round(sd,2)))

```

R code Exa 4.16 Poisson Probability distribution example3

```

1 #page no :216-217
2 muday=41.5
3 muhour=muday/24
4 e=2.718
5 print(paste('a. solution ', round(muhour,4)))
6 factorial=function(v)
7 {
8   fact=1
9   if(v==0)
10  {
11    return(1)
12  }else{
13    for(i in 1:v)
14    {
15      fact=fact*i
16    }
17    return(fact)
18  }
19 }
20 x=2
21 nu=((muhour^x)*(e^(-muhour)))
22 p2=nu/factorial(x)
23 print(paste('b. P(x=2)', round(p2,3)))
24 x=0
25 nu1=((muhour^x)*(e^(-muhour)))
26 p0=nu1/factorial(x)
27 x=1
28 nu2=((muhour^x)*(e^(-muhour)))

```

```

29 p1=nu2/factorial(x)
30 x=2
31 nu3=((muhour^x)*(e^(-muhour)))
32 p2=nu3/factorial(x)
33 px2=1-(p0+p1+p2)
34 print(paste('c. P(x > 2) ',round(px2,3)))

```

R code Exa 4.17 Poison Probability distribution example4

```

1 #page no :217
2 mu=200*0.0102
3 e=2.718
4 N=200
5 r=10
6 a=dbinom(10,size=200,prob=.0102)
7 sprintf("a. binomial distribution = %.6f",a)
8 factorial=function(v)
9 {
10     fact=1
11     if(v==0)
12     {
13         return(1)
14     }else{
15         for(i in 1:v)
16         {
17             fact=fact*i
18         }
19         return(fact)
20     }
21 }
22
23 x=10
24 nu=((mu^x)*(e^(-mu)))
25 p2=nu/factorial(x)
26 sprintf('b. poison distribution = %.6f',p2)

```

R code Exa 4.18 Poisson Probability distribution example5

```
1 #page no :217-218
2 size=500
3 prob=0.01
4 mu=size*prob
5 x<-c(0,1,2)
6 binom1=0
7 for(i in 1:length(x))
8 {
9   v1=dbinom(x[i],size,prob)
10  binom1 = binom1 + v1
11  print(paste("binomial distribution P(x=",x[i],")=",
12            round(v1,5)))
13 }
14 print(paste("binomial distribution P(0,1,2)",round(
15   binom1,5)))
16 app1=1-binom1
17 print(paste("binomial solution =",round(app1,5)))
18 poisson1=0
19 for(i in 1:length(x))
20 {
21   v2=dpois(x[i],mu)
22   poisson1 = poisson1 +v2
23   print(paste("poisson approx P(x=",x[i],")=",round(
24     v2,4)))
25 }
26 print(paste("poisson approx P(0,1,2)",round(
27   poisson1,4)))
28 app2=1-poisson1
29 print(paste("poisson solution =",round(app2,4)))
```

Chapter 5

Continuous Random Variables

R code Exa 5.1 Continuous Probability Density functions example1

```
1 #page no: 243-246
2 library(visualize)
3 a=20
4 b=0
5 x<-seq(b,a,by=1)
6 fx1=dunif(x,min=0,max=20)
7 base=(2-0)
8 height=1/20
9 area=base*height
10 p_2=(2-0)*(1/20)
11 print(paste("a. p(0<x<2)=" ,round(p_2,4)))
12 par(mfcol=c(2,2))
13 visualize.unif(stat=c(0,2),a=0,b=20,section="bounded
    ")
14 base2=15-4
15 area2=base2*height
16 p_15=base2*(1/20)
17 print(paste("b. p(4 < x < 15)=" ,round(p_15,4)))
18 visualize.unif(stat=c(4,15),a=0,b=20,section="
    bounded")
19 base3=(12.7-2.3)
```

```

20 p_23_127=base3*height
21 print(paste("b. p(2.3 < x < 12.7)=",round(p_23_
    127,4)))
22 visualize.unif(stat=c(2.3,12.7),a=0,b=20,section="
    bounded")

```

R code Exa 5.2 Uniform Distribution example1

```

1 #page no: 247-248
2 library(visualize)
3 a=15
4 b=0
5 x<-seq(b,a,by=1)
6 fx1=dunif(x,min=0,max=12.5)
7 p_12=(12.5-0)*(1/15)
8 print(paste("a. p(x<12.5)=",round(p_12,4)))
9 mu=(a+b)/2
10 var=((b-a)^2)/12
11 sd=sqrt(((b-a)^2)/12)
12 print(paste("b. mean =",mu))
13 print(paste("b. sd =",round(sd,1)))
14 k=quantile(x,.90)
15 print(paste("c. 90th percentile =",k))
16 visualize.unif(stat=c(0,12.5),a=0,b=15,section="
    bounded")

```

R code Exa 5.3 The Exponential Distribution example1

```

1 #page no:250
2 library(ggplot2)
3 x<-seq(0,20,length.out=100)
4 mu=4
5 e=2.718

```

```

6 m=1/mu
7 px=dexp(x,rate=m)
8 df<-data.frame(x=x,y=px)
9 x=5
10 p_5=(m*(e^(-m*x)))
11 print(paste("a.P>5 =",round(p_5,3)))
12 qplot(x,y,data=df,geom="line")+
13   geom_ribbon(data=subset(df,x>=5 ),aes(ymax=y),ymin
14     =0,
15     fill="red",colour=NA,alpha=0.5)+
16   scale_x_continuous(labels=as.character(seq(0:20)),
17     breaks=seq(0:20))+
18   annotate(geom="text",x=4,y=0.25,label=paste("m
19     =0.25"))+
20   labs(x="mu=4", y="f(x)")

```

R code Exa 5.4 The Exponential Distribution example2

```

1 #page no: 251
2 library(ggplot2)
3 e=2.718
4 mu=0.25
5 x<-seq(0,20,length.out=100)
6 px=dexp(x,rate=mu)
7 df<-data.frame(x=x,y=px)
8 x1=5
9 p_5=(1-(e^(-mu*x1)))
10 x2=4
11 p_4=(1-(e^(-mu*x2)))
12 p_4_5=p_5-p_4
13 qplot(x,y,data=df,geom="line")+
14   geom_ribbon(data=subset(df,x>=4 & x<5),aes(ymax=y)
15     ,ymin=0,
16     fill="red",colour=NA,alpha=0.5)+
17   scale_x_continuous("x",labels=as.character(seq

```

```

      (0:20)), breaks=seq(0:20))+
17   annotate(geom="text", x=10, y=0.05, label=paste("P(4<
      x<5)"))+
18   labs( y="f(x)")
19   print(paste("P(x<5) =", round(p_5,4)))
20   print(paste("P(x<4) =", round(p_4,4)))
21   print(paste("P(4<x<5) =", round(p_4_5,4)))

```

R code Exa 5.5 The Exponential Distribution example3

```

1 #page no: 251-253
2 library(ggplot2)
3 require(gridExtra)
4 e=2.718
5 mu=10
6 m=1/10
7 x<-seq(0,15, length.out=200)
8 px=dexp(x, rate=m)
9 df<-data.frame(x=x, y=px)
10 x1=7
11 p_7=(1-(e^(-m*x1)))
12 p10=(1-p_7)
13 p1<-qplot(x,y, data=df, geom="line")+
14   geom_ribbon(data=subset(df, x>=7), aes(ymax=y), ymin
      =0.02,
15           fill="blue", colour=NA, alpha=0.5)+
16   scale_x_continuous("x", labels=as.character(seq
      (0:20)), breaks=seq(0:20))+
17   annotate(geom="text", x=7, y=0.06, label=paste("P(x
      >7)"))+
18   labs(x="mu=10", y="f(x)")
19   print(paste("P(x<7) =", round(p_7,4)))
20   print(paste("a. P(x > 7) =", round(p10,4)))
21   print(paste("b. solution =", 5*mu, " years"))
22   x2=9

```

```

23 p_9=(1-(e^(-m*x2)))
24 x3=11
25 p_11=(1-(e^(-m*x3)))
26 p_9_11=p_11-p_9
27 print(paste("d. P(9 < x < 11)=",round(p_9_11,4)))
28 p2<-qplot(x,y,data=df,geom="line")+
29   geom_ribbon(data=subset(df,x>=9 & x<=11),aes(ymax=
      y),ymin=0.02,
30             fill="blue",colour=NA,alpha=0.5)+
31   scale_x_continuous("x",labels=as.character(seq
      (0:20)),breaks=seq(0:20))+
32   annotate(geom="text",x=10,y=0.05,label=paste("P(9<
      x<11)"))+
33   labs(x="mu=10", y="f(x)")
34 grid.arrange(p1,p2,ncol=2)

```

R code Exa 5.6 The Exponential Distribution example4

```

1 #page no : 253
2 library(MASS)
3 m=1/12
4 e=2.718
5 mu=1/m
6 sd=12
7 x=5
8 p_5=(1-(e^(-m*x)))
9 p_gt_5=1-p_5
10 print(paste("a. m =",fractions(m)))
11 print(paste("b. mu =",fractions(mu)))
12 print(paste("c. sigma =",fractions(sd)))
13 print(paste("a. prob. more than 5=",round(p_gt_5,4))
      )
14 #The answer may slightly vary due to rounding off
      values.

```

R code Exa 5.7 The Exponential Distribution example5

```
1 #page no: 253-254
2 library(ggplot2)
3 require(gridExtra)
4 e=2.718
5 avg=30
6 hour=60
7 a=hour/avg
8 print(paste("a. solution=",a," minutes"))
9 b=a*3
10 print(paste("b. solution=",b," minutes"))
11 mu=2
12 m=1/mu
13 x<-seq(0,20,length.out=200)
14 px=dexp(x,rate=m)
15 df<-data.frame(x=x,y=px)
16 x1=1
17 p_1=(1-(e^(-m*x1)))
18 print(paste("c. P(x<1) =",round(p_1,4)))
19 p1<-qplot(x,y,data=df,geom="line")+
20   geom_ribbon(data=subset(df,x<1),aes(ymax=y),ymin
21     =0,
22     fill="red",colour=NA,alpha=0.5)+
23   scale_x_continuous("x",labels=as.character(seq
24     (0:20)),breaks=seq(0:20))+
25   annotate(geom="text",x=7,y=0.4,label=paste("P
26     =0.3935"))+
27   labs(y="f(x)")
28 x2=5
29 p_5=(1-(e^(-m*x2)))
30 p_gt_5=1-p_5
31 print(paste("d. P(x>5)=",round(p_gt_5,4)))
32 p2<-qplot(x,y,data=df,geom="line")+
```

```

30 geom_ribbon(data=subset(df,x>=5),aes(ymax=y),ymin
    =0,
31           fill="red",colour=NA,alpha=0.5)+
32 scale_x_continuous("x",labels=as.character(seq
    (0:20)),breaks=seq(0:20))+
33 annotate(geom="text",x=10,y=0.1,label=paste("P(x
    >5)=1-P(x<5)"))+
34 labs(y="f(x)")
35 grid.arrange(p1,p2,ncol=2)

```

R code Exa 5.8 Exponential probability

```

1 #page no:255
2 library(ggplot2)
3 require(gridExtra)
4 x<-seq(0,20,length.out=100)
5 e=2.718
6 mu=4
7 m=1/mu
8 px=dexp(x,rate=m)
9 df<-data.frame(x=x,y=px)
10 x1=3
11 p_3=1-(e^(-m*x1))
12 print(paste("P(x<3)=",round(p_3,4)))
13 p_gt_3=1-p_3
14 print(paste("Answer P(x>3)=",round(p_gt_3,4)))
15 p1<-qplot(x,y,data=df,geom="line")+
16   geom_ribbon(data=subset(df,x>3),aes(ymax=y),ymin
    =0,
17           fill="red",colour=NA,alpha=0.5)+
18 scale_x_continuous("x",labels=as.character(seq
    (0:20)),breaks=seq(0:20))+
19 annotate(geom="text",x=12,y=0.05,label=paste("P
    >3)=0.4724"))+
20 labs(y="f(x)")

```

21 `grid.arrange(p1, ncol=1)`

Chapter 6

The Normal Distribution

R code Exa 6.1 The Standard Normal Distribution example1

```
1 #page no: 280-281
2 mu=5
3 sd=6
4 x=17
5 z=(x-mu)/sd
6 if(z>0)
7 {
8   print(paste("x=",x," is to right of mean"))
9 }else{
10   print(paste("x=",x," is to left of mean"))
11 }
12 x=1
13 z=(x-mu)/sd
14 if(z>0)
15 {
16   print(paste("x=",x," is to right of mean"))
17 }else{
18   print(paste("x=",x," is to left of mean"))
19 }
```

R code Exa 6.2 Finding z scores for given mean and standard deviation

```
1 #page no: 281
2 mu=50
3 sd=6
4 x=68
5 x1=mu-sd
6 x2=mu+sd
7 z=(x-mu)/sd
8 z1=(x1-mu)/sd
9 z2=(x2-mu)/sd
10 print(paste("for the x value =",x1, "the z score is
    =",z1))
11 print(paste("for the x value =",x2, "the z score is
    =",z2))
12 sd1=2*sd
13 x3=mu-sd1
14 x4=mu+sd1
15 z3=(x3-mu)/sd
16 z4=(x4-mu)/sd
17 print(paste("for the x value =",x3, "the z score is
    =",z3))
18 print(paste("for the x value =",x4, "the z score is
    =",z4))
19 sd2=3*sd
20 x5=mu-sd2
21 x6=mu+sd2
22 z5=(x5-mu)/sd
23 z6=(x6-mu)/sd
24 print(paste("for the x value =",x5, "the z score is
    =",z5))
25 print(paste("for the x value =",x6, "the z score is
    =",z6))
```

R code Exa 6.3 Using the Normal Distribution example1

```
1 #page no:284-285
2 library(ggplot2)
3 require(gridExtra)
4 x=65
5 mu=63
6 sd=5
7 p65<-pnorm(x,mean=mu,sd=sd)
8 pgt65=1-p65
9 z=(x-mu)/sd
10 p0=pnorm(63,mean=mu,sd=5)
11 p63_65=p65-p0
12 print(paste("a. z1 left area=",round(p63_65,4)))
13 print(paste("a. P(x>65) area=",round(pgt65,4)))
14 xx=seq(0,100,length.out=1000)
15 px=dnorm(xx,mean=mu,sd=sd)
16 df<-data.frame(x=xx,y=px)
17 p1<-qplot(x,y,data=df,geom="line")+
18   geom_ribbon(data=subset(df, x>=65),aes(ymax=y),
19             ymin=0,
20             fill="blue",colour=NA,alpha=0.5)+
21   scale_x_continuous("x",labels=as.character(seq
22     (0:100)),breaks=seq(0:100))+
23   geom_vline(aes(xintercept=mu),color="blue",
24             linetype="dashed")+
25   annotate("text",x=64,y=0.01,size=3,label="0.1554")
26   +
27   annotate("text",x=80,y=0.01,size=3,label="0.3446")
28   +
29   annotate("text",x=63,y=0.00,size=3,label="mu=63")+
30   annotate("text",x=65,y=0.00,size=3,label="x=65")
31 grid.arrange(p1,ncol=1)
32 x1=85
```

```

28 z1=(x1-mu)/sd
29 if(z1>4.0)
30 {
31   print("b. solution = 1")
32 }

```

R code Exa 6.4 Using the Normal Distribution example2

```

1 #page no:285-287
2 library(ggplot2)
3 require(gridExtra)
4 mu=2
5 sd=0.5
6 x=1.8
7 p18<-pnorm(x,mean=mu,sd=sd)
8 x1=2.75
9 p275<-pnorm(x1,mean=mu,sd=sd)
10 p=p275-p18
11 print(paste("a. P(1.8<=x<=2.75)=",round(p,4)))
12 fz=sd-0.25
13 z=qnorm(0.25)
14 print(paste("a. z score =",round(z,3)))
15 xvalue=(z*sd)+mu
16 print(paste("b. max =",round(xvalue,2)," hours"))
17 xx=seq(0,10,length.out=100)
18 px=dnorm(xx,mean=mu,sd=sd)
19 df<-data.frame(x=xx,y=px)
20 z1=1.8
21 z2=2.75
22 p1<-qplot(x,y,data=df,geom="line")+
23   geom_ribbon(data=subset(df, x>=z1 & x<=z2 ),aes(
24     ymax=y),ymin=0,
25     fill="blue",colour=NA,alpha=0.5)+
26   scale_x_continuous("x",labels=as.character(seq
27     (0:10)),breaks=seq(0:10))+

```

```

26   geom_vline(aes(xintercept=mu), color="blue",
27             linetype="dashed")+
28   annotate("text",x=2,y=0.00,size=3,label="mu=2")+
29   annotate("text",x=1.8,y=-0.005,size=3,label="1.8")
30   +
31   annotate("text",x=2.75,y=0.00,size=3,label="2.75")
32   +
33   annotate("text",x=5,y=0.2,size=3,label="sd=0.5")
34 p2<-qplot(x,y,data=df,geom="line")+
35   geom_ribbon(data=subset(df, x<1.66 ),aes(ymax=y),
36             ymin=0,
37             fill="blue",colour=NA,alpha=0.5)+
38   scale_x_continuous("x",labels=as.character(seq
39                     (0:10)),breaks=seq(0:10))+
40   geom_vline(aes(xintercept=mu),color="blue",
41             linetype="dashed")+
42   annotate("text",x=0.5,y=0.4,size=3,label="P(x<k)
43           =0.25")+
44   annotate("text",x=4,y=0.2,size=3,label="P(x>k)
45           =0.75")
46 grid.arrange(p1,p2,nrow=2)

```

R code Exa 6.5 Using the Normal Distribution example3

```

1 #page no:287
2 library(ggplot2)
3 mu=36.9
4 sd=13.9
5 x1=23
6 p23<-pnorm(x1,mean=mu,sd=sd)
7 x2=64.7
8 p64<-pnorm(x2,mean=mu,sd=sd)
9 p=p64-p23
10 print(paste("a. solution =",round(p,4)))
11 x3=50.8

```

```

12 p508=pnorm(x3,mean=mu,sd=sd)
13 print(paste("b. solution =",round(p508,4)))

```

R code Exa 6.6 Using the Normal Distribution example4

```

1 #page no:288
2 library(ggplot2)
3 mu=5.85
4 sd=0.24
5 x1=6.0
6 p6<-round(pnorm(x1,mean=mu,sd=sd),4)
7 pgt6=1-p6
8 z1=(x1-mu)/sd
9 print(paste("a. P(>=6.0)=",round(pgt6,4)))
10 print(paste("a. Z1=",round(z1,4)))
11 f_z=0.20/2
12 fz1=0.25
13 fz2=-0.25
14 x2=(fz1*sd)+mu
15 x3=(fz2*sd)+mu
16 print(paste("b. solution = ",x3," and ",x2))
17 xx=seq(0,7,length.out=100)
18 px=dnorm(xx,mean=mu,sd=sd)
19 df<-data.frame(x=xx,y=px)
20 p1<-qplot(x,y,data=df,geom="line")+
21   geom_ribbon(data=subset(df, x>=6.0 & x<=7 ),aes(
22     ymax=y),ymin=0,
23     fill="blue",colour=NA,alpha=0.5)+
24   scale_x_continuous("x",labels=as.character(seq
25     (0:7)),breaks=seq(0:7))+
26   geom_vline(aes(xintercept=mu),color="blue",
27     linetype="dashed")+
28   geom_text(x=7,y=0.5,size=3.5,
29     label=paste0("sd= ",sd))+
30   annotate("text",x=5.85,y=-0.05,size=3,label="mu

```

```

    =5.85")+
28   annotate("text",x=6.0,y=-0.05,size=3,label="6.0")
29   grid.arrange(p1,ncol=1)

```

R code Exa 6.7 Finding binomial and normal distribution

```

1  #page no:290-291
2  library(ggplot2)
3  library(gridExtra)
4  mu=10
5  sd=3
6  x=16
7  z=(x-mu)/sd
8  xx=0:100
9  n=100
10 px=dbinom(xx,size=n,prob=0.0228)
11 df<-data.frame(x=xx,y=px)
12 z1=15
13 z2=100
14 p1<-plot(df$x,df$y,type='h',lty=1,lwd=5,xlab="number
    of sucesses",ylab="P(X)",col=ifelse(df$x<15,'
    grey','blue'))
15 x_norm=seq(0:100)
16 p_norm=dnorm(x_norm,mean=mu,sd=sd)
17 df_norm<-data.frame(x=x_norm,y=p_norm)
18 p2<-qplot(x,y,data=df_norm,geom="line")+
19   geom_ribbon(data=subset(df_norm, x>=15),aes(ymax=
    y),ymin=0,
20             fill="blue",colour=NA,alpha=0.5)+
21   scale_x_continuous("x",labels=as.character(seq
    (0:100)),breaks=seq(0:100))+
22   geom_vline(aes(xintercept=mu),color="blue",
    linetype="dashed")
23 p2
24 x1=16.0

```

```
25 p16<-round(pnorm(x1,mean=mu,sd=sd),4)
26 pgt16=1-p16
27 print(paste("P(X>16) =",pgt16))
```

Chapter 7

The Central Limit Theorem

R code Exa 7.1 Finite Population Correction Factor example1

```
1 #page no: 320-321
2 N=4000
3 n=100
4 sd=10.37
5 mu=75.45
6 differ1=2
7 differ2=-2
8 term1=sd/sqrt(n)
9 term2=sqrt((N-n)/(N-1))
10 denom=term1*term2
11 z=differ1/denom
12 Z1=differ2/denom
13 zscore_r=0.4744
14 fz=zscore_r*differ1
15 print(paste("f(Z) = ",fz))
```

R code Exa 7.2 Finite Population Correction Factor example2

```
1 #page no: 321
2 N=3000
3 n=360
4 p=0.06
5 term1=sqrt((p*(1-p))/n)
6 term2=sqrt((N-n)/(N-1))
7 sdp1=term1*term2
8 p10=10/n
9 p20=20/n
10 znum1=(p10-p)
11 znum2=(p20-p)
12 z10=znum1/sdp1
13 z20=znum2/sdp1
14 zscore10=0.4969
15 zscore20=0.1480
16 ans=zscore10-zscore20
17 print(paste("sigma p' =",round(sdp1,4)))
18 print(paste("P1' =",round(p10,4)))
19 print(paste("P2' =",round(p20,4)))
20 print(paste("Z1' =",round(z10,2)))
21 print(paste("Z2' =",round(z20,2)))
22 print(paste("the solution=",round(ans,4)))
```

Chapter 8

Confidence Intervals

R code Exa 8.1 Calculating the Confidence Intervals example1

```
1 #page no:338
2 xbar=68
3 sd=3
4 n=36
5 conf_l=0.90
6 alpha=1-conf_l
7 zalpha=alpha/2
8 arearight=zalpha
9 arealeft=1-zalpha
10 zscore=round(qnorm(arealeft),3)
11 al=(1+conf_l)/2
12 EBM=zscore*(sd/sqrt(n))
13 interval1=xbar-EBM
14 interval2=xbar+EBM
15 print(paste("z score is=",zscore))
16 print(paste("90% confidence interval= (",interval1,"
  , ",interval2,")"))
```

R code Exa 8.2 Calculating the Confidence Intervals example2

```

1 #page no:339-340
2 library(ggplot2)
3 xbar=68
4 sd=3
5 n=36
6 conf_l=0.95
7 alpha=1-conf_l
8 x=seq(67,69,length=n)
9 y=dnorm(x,xbar,sd)
10 df=data.frame(x,y)
11 zalpha=alpha/2
12 arearight=zalpha
13 arealeft=1-zalpha
14 zscore=round(qnorm(arealeft),3)
15 al=(1+conf_l)/2
16 EBM=zscore*(sd/sqrt(n))
17 mu1=xbar-EBM
18 mu2=xbar+EBM
19 print(paste("z =",zscore))
20 print(paste("90% interval= (",mu1," , ",mu2,")"))
21 par(mfcol=c(2,1))
22 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
23 abline(v=xbar)
24 abline(v=mu1,lwd=2,col='red')
25 abline(v=mu2,lwd=2,col='red')
26 polygon(c(x[x<=mu1],mu1),c(y[x<=mu1],0),col="red")
27 polygon(c(x[x>=mu2],mu2),c(y[x>=mu2],0),col="red")
28 text(mu1,0.126,"67.02")
29 text(mu2,0.126,"68.98")
30 text(68,0.126,"68")
31 x=seq(-3,3,length=100)
32 y <- dt(x,df=Inf)
33 df=data.frame(x,y)
34 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
35 abline(v=0)
36 abline(v=-1.96,lwd=2,col='red')
37 abline(v=1.96,lwd=2,col='red')
38 polygon(c(x[x<=-1.96],-1.96),c(y[x<=-1.96],0),col="red")

```

```

    red")
39 polygon(c(x[x>=abs(1.96)],abs(1.96) ),c(y[x>=abs
    (1.96)],0),col="red")
40 text(-1.96,0,-1.96)
41 text(1.96,0,1.96)
42 text(0,0,0)

```

R code Exa 8.3 Calculating the Confidence Intervals example3

```

1 #page no:340-341
2 xbar=68
3 sd=3
4 n=100
5 conf_l=0.90
6 alpha=1-conf_l
7 zalpha=alpha/2
8 arearight=zalpha
9 arealeft=1-zalpha
10 zscore=round(qnorm(arealeft),3)
11 al=(1+conf_l)/2
12 EBM=zscore*(sd/sqrt(n))
13 interval1=xbar-EBM
14 interval2=xbar+EBM
15 print(paste("z score is",zscore))
16 print(paste("solution A (",interval1," , ",interval2
    ,")"))
17 n=25
18 EBM=zscore*(sd/sqrt(n))
19 interval1=xbar-EBM
20 interval2=xbar+EBM
21 print(paste("Solution B (",interval1," , ",interval2
    ,")"))

```

R code Exa 8.4 Calculating the Confidence Intervals example4

```
1 #page no: 341-342
2 n=80
3 sd=369.34
4 xbar=593.84
5 conf_l=0.92
6 alpha=1-conf_l
7 zalpha=alpha/2
8 arearight=zalpha
9 arealeft=1-zalpha
10 zscore=round(qnorm(arealeft),2)
11 v1=round((zscore*(sd/sqrt(n))),2)
12 mu1=xbar-v1
13 mu2=xbar+v1
14 print(paste("z score =",zscore))
15 print(paste(" Answer :(",round(mu1,2)," , ",round(mu2
    ,2),")"))
16 xseq=seq(mu1,mu2,length=n)
17 d=dnorm(xseq,xbar,sd)
18 par(mfcol=c(2,1))
19 plot(xseq,d,type='l',lwd=3,col='blue',xlab='x')
20 abline(v=xbar)
21 abline(v=mu1,lwd=2,col='red')
22 abline(v=mu2,lwd=2,col='red')
23 text(mu1,0.001060,"521.58")
24 text(mu2,0.001060,"666.10")
25 text(xbar,0.001060,"593.84")
26 x=seq(-3,3,length=100)
27 y <- dt(x,df=Inf)
28 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
29 abline(v=0)
30 abline(v=-1.75,lwd=2,col='red')
31 abline(v=1.75,lwd=2,col='red')
32 polygon(c(x[x<=-1.75],-1.75),c(y[x<=-1.75],0),col="
    red")
33 polygon(c(x[x>=abs(1.75)],abs(1.75)),c(y[x>=abs
    (1.75)],0),col="red")
```

```

34 text(-1.75,0,-1.75)
35 text(1.75,0,1.75)
36 text(0,0,0)
37 text(-2.7,0.2,expression(frac(alpha,2)))
38 text(-2.5,0.2,zalpha)
39 text(2.3,0.2,expression(frac(alpha,2)))
40 text(2.5,0.2,zalpha)

```

R code Exa 8.5 Confidence interval for a small sample example1

```

1 #page no: 345-346
2 n=10
3 sd=0.395
4 xbar=1.851
5 conf_l=0.99
6 alpha=1-conf_l
7 zalpha=alpha/2
8 arearight=zalpha
9 arealeft=1-zalpha
10 zscore=round(qnorm(arealeft),2)
11 tscore=round(qt(arealeft,df=9),4)
12 v1=round((tscore*(sd/sqrt(n))),3)
13 mu1=xbar-v1
14 mu2=xbar+v1
15 print(paste("t score is=",tscore))
16 print(paste("Answer :",round(mu1,3)," \u2264 \u00B5
    \u2264" , round(mu2,3)))
17 x=seq(-4,4,length=100)
18 y <- dt(x,df=9)
19 prob1=-3.2498
20 par(mfcol=c(1,2))
21 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
22 abline(v=0)
23 abline(v=prob1,lwd=2,col='red')
24 abline(v=abs(prob1),lwd=2,col='red')

```

```

25 polygon(c(x[x<=prob1],prob1 ),c(y[x<=prob1],0),col="
    red")
26 polygon(c(x[x>=abs(prob1)],abs(prob1) ),c(y[x>=abs(
    prob1)],0),col="red")
27 text(prob1,0,prob1)
28 text(abs(prob1),0,abs(prob1))
29 text(0,0,round(0,2))
30 x=seq(mu1-2,mu2+2,length=100)
31 y=dnorm(x,xbar,sd)
32 df=data.frame(x,y)
33 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
34 abline(v=xbar)
35 abline(v=mu1,lwd=2,col='red')
36 abline(v=mu2,lwd=2,col='red')
37 polygon(c(x[x<=abs(mu1)],abs(mu1) ),c(y[x<=abs(mu1)
    ],0),col="red")
38 polygon(c(x[x>=mu2],mu2 ),c(y[x>=mu2],0),col="red")
39 text(mu1,0,round(mu1,2))
40 text(mu2,0,round(mu2,2))
41 text(xbar,0,round(xbar,2))

```

R code Exa 8.6 Confidence interval for a population proportion example1

```

1 #page no: 347-348
2 n=500
3 sample_x=421
4 p_dash=sample_x/n
5 q_dash=1-p_dash
6 conf_l=0.95
7 alpha=1-conf_l
8 zalpha=alpha/2
9 arearight=zalpha
10 arealeft=1-zalpha
11 zscore=round(qnorm(arealeft),2)
12 tscore=round(qt(arealeft,df=Inf),4)

```



```

13 v1=(zscore*(sqrt((p_dash*q_dash)/n)))
14 p1=p_dash-v1
15 p2=p_dash+v1
16 print(paste("p' =",p_dash))
17 print(paste("q' =",q_dash))
18 print(paste("z   =",zscore))
19 print(paste(round(p1,3)," \u2264 p \u2264" ,round(p2
    ,3)))
20 print(paste("people –cell phone ", round(p1,3)*100,"
    %", " to ", round(p2,3)*100,"%"))

```

R code Exa 8.7 Confidence interval for a population proportion example2

```

1 #page no: 348–349
2 n=150
3 l_limit=0.08
4 u_limit=0.16
5 p_dash=(l_limit+u_limit)/2
6 p=1-p_dash
7 c_interval=u_limit-p_dash
8 v1=sqrt((p_dash*(1-p_dash))/n)
9 zscore=round(c_interval*(1/v1),2)
10 p_zvalue=0.4345
11 p_zvalue1=2*p_zvalue
12 print(paste("p' is=",p_dash))
13 print(paste("z score is=",zscore))
14 print(paste("confidence is ", round(p_zvalue1,3)*
    100,"%"))

```

R code Exa 8.8 Confidence interval for a population proportion example3

```

1 #page no: 349
2 n=500

```

```

3 x=300
4 p_dash=(x/n)
5 q_dash=1-p_dash
6 cl=0.90
7 alpha=1-cl
8 zalpha=alpha/2
9 arearight=zalpha
10 arealeft=1-zalpha
11 zscore=round(qnorm(arealeft),3)
12 v1=round((zscore*(sqrt((p_dash*q_dash)/n))),3)
13 p1=p_dash-v1
14 p2=p_dash+v1
15 print(paste("z score is=",zscore))
16 print(paste("90% confidence :",round(p1,3)," , ",
              round(p2,3)))
17 print(paste("solution : ", round(p1,3)*100,"%", " to
              ", round(p2,3)*100,"%"))

```

R code Exa 8.9 Calculating the sample size n continuous and binary random variables

```

1 #page no 351
2 aer=0.03
3 cl=0.90
4 alpha=1-cl
5 zalpha=alpha/2
6 arearight=zalpha
7 arealeft=1-zalpha
8 zscore=round(qnorm(arealeft),3)
9 p_dash=0.5
10 q_dash=1-p_dash
11 n=((zscore^2)*p_dash*q_dash)/(aer^2)
12 print(paste(" the sample size is = ",round(n,0)))

```

Chapter 9

Hypothesis Testing with One Sample

R code Exa 9.8 Full Hypothesis test example1

```
1 #page no 392-394
2 mu0=16.43
3 xbar=16
4 sd=0.8
5 n=15
6 t_stat=(xbar-mu0)/(sd/sqrt(n))
7 alpha=0.05
8 t_c_left=qnorm(alpha)
9 p_value=pnorm(t_stat)
10 print(paste('t static =',round(t_stat,4)))
11 if(t_stat <= t_c_left)
12 {
13   print(paste("left tail : Reject H0"))
14 }else
15 {
16   print(paste("Left tail :Accept H0"))
17 }
18 t_c_right=qnorm(1-alpha)
19 p_value=1-pnorm(t_stat)
```

```

20   if(t_stat >= t_c_right)
21   {
22     print(paste("right tail : Reject H0"))
23   }else
24   {
25     print(paste("right tail :Accept H0"))
26   }
27 two_t_stat=abs((xbar-mu0))/(sd/sqrt(n))
28 two_t_critical=qnorm(1-alpha/2)
29 p_value=2*(1-pnorm(two_t_stat))
30 if(two_t_stat>=two_t_critical)
31 {
32   print("reject H0")
33 }else
34 {
35   print("accept H0")
36 }
37 dfs<-n-1
38 x<-seq(-3,3,0.1)
39 y<-dt(x,dfs)
40 t.val<-qt(0.95,df=dfs)
41 p1<-plot(x,y,type='l',lwd=3,col='blue',xlab='x')+
42   abline(v=0)+
43   abline(v=t.val,lwd=2,col="green")+
44   polygon(c(x[x<=-1.76],-2.08),c(y[x
45     <=-1.76],-2.08),col="white")+
46   polygon(c(x[x<=-2.08],-2.08),c(y[x<=-2.08],0),
47     col="blue")+
48   text(t_stat,0,round(t_stat,2))+
49   text(-t.val,0,"-1.76")+
50   text(-2.3,0.1,"alpha=")+
51   text(-2,0.1,alpha)

```

R code Exa 9.9 Full Hypothesis test example2

```

1 #page no 395
2 library(ggplot2)
3 mu0=100
4 xbar=108
5 sd=12
6 n=16
7 dfs=n-1
8 t_stat=(xbar-mu0)/(sd/sqrt(n))
9 alpha=0.05
10 t_c_right=abs(qt(1-alpha,df=dfs))
11 p_value=1-pt(t_stat,df=dfs)
12 print(paste("critical value=",round(t_c_right,3)))
13 print(paste("t-static =",round(t_stat,2)))
14 if(t_stat >= t_c_right)
15 {
16   print(paste("right tail : Reject H0"))
17 }else
18 {
19   print(paste("right tail : Accept H0"))
20 }
21 two_t_stat=abs((xbar-mu0))/(sd/sqrt(n))
22 two_t_critical=qnorm(1-alpha/2)
23 p_value=2*(1-pnorm(two_t_stat))
24 if(two_t_stat>=two_t_critical)
25 {
26   print("reject H0")
27 }else
28 {
29   print("accept H0")
30 }
31 dfs <- n-1
32 x <- seq(-4,4,0.1)
33 y <- dt(x,dfs)
34 t.val <- qt(0.95,df=dfs)
35 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
36 abline(v=0)
37 abline(v=t.val,lwd=2,col='red')
38 abline(v=t_stat,lwd=2,col='green')

```

```

39 polygon(c(x[x>=1.75],1.75 ),c(y[x>=1.75],0),col="red
    ")
40 text(t_stat,0,round(t_stat,2))
41 text(t.val,0,round(t.val,2))
42 text(3.0,0.1,"alpha=")
43 text(3.6,0.1,alpha)

```

R code Exa 9.10 Full Hypothesis test example3

```

1 #page no 396-397
2 library(ggplot2)
3 mu0=8
4 xbar=7.91
5 sv=.03
6 sd=sqrt(sv)
7 n=35
8 dfs=n-1
9 t_stat=(xbar-mu0)/(sd/sqrt(n))
10 alpha=(1-0.99)/2
11 t_c_left=(qt(alpha,df=Inf))
12 p_value=pt(t_stat,df=Inf)
13 print(paste("critical value=",round(t_c_left,3)))
14 print(paste("t-static =",round(t_stat,2)))
15 if(t_stat <= t_c_left)
16 {
17   print(paste("left tail : Reject H0"))
18 }else
19 {
20   print(paste("left tail :Accept H0"))
21 }
22 two_t_stat=abs((xbar-mu0))/(sd/sqrt(n))
23 two_t_critical=qnorm(1-alpha/2)
24 p_value=2*(1-pnorm(two_t_stat))
25 if(two_t_stat>=two_t_critical)
26 {

```

```

27   print("reject H0")
28 }else
29 {
30   print("accept H0")
31 }
32 x <- seq(-4,4,0.1)
33 y <- dt(x,df=Inf)
34 t.val <- qt(0.95,df=Inf)
35 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
36 abline(v=0)
37 abline(v=t_c_left,lwd=2,col='red')
38 abline(v=t_stat,lwd=2,col='green')
39 abline(v=abs(t_c_left),lwd=2,col='green')
40 polygon(c(x[x>=abs(t_c_left)],abs(t_c_left)),c(y[x
    >=abs(t_c_left)],0),col="red")
41 polygon(c(x[x<=t_c_left],t_c_left),c(y[x<=t_c_left
    ],0),col="red")
42 text(t_c_left,0,round(t_c_left,3))
43 text(t_stat,0,round(t_stat,2))
44 text(abs(t_c_left),0,round(abs(t_c_left),3))
45 text(3.0,0.1,"alpha/2=")
46 text(3.6,0.1,alpha)
47 text(-3.6,0.1,"alpha/2=")
48 text(-3.0,0.1,alpha)

```

R code Exa 9.11 Hypothesis test for proportions example1

```

1 #page no 398-399
2 library(ggplot2)
3 n=100
4 p_dash=53/n
5 p0=50/n
6 q0=1-p0
7 n=100
8 t_stat=(p_dash-p0)/(sqrt((p0*q0)/n))

```

```

 9 alpha=((5/n))/2
10 t_c_left=(qt(alpha,df=Inf))
11 p_value=pt(t_stat,df=Inf)
12 print(paste("critical value= ",round(t_c_left,2)))
13 print(paste("t-static =",round(t_stat,2)))
14 if(t_stat <= t_c_left)
15 {
16   print(paste("left tail : Reject H0"))
17 }else
18 {
19   print(paste("left tail :Accept H0"))
20 }
21 t_c_right=qnorm(1-alpha)
22 p_value=1-pnorm(t_stat)
23 if(t_stat >= t_c_right)
24 {
25   print(paste("right tail : Reject H0"))
26 }else
27 {
28   print(paste("right tail :Accept H0"))
29 }
30
31 two_t_stat=abs((p_dash-p0))/(sqrt((p0*q0)/n))
32 two_t_critical=qnorm(1-alpha/2)
33 p_value=2*(1-pnorm(two_t_stat))
34 if(two_t_stat>=two_t_critical)
35 {
36   print("reject H0")
37 }else
38 {
39   print("accept H0")
40 }
41 x <- seq(-4,4,0.1)
42 y <- dt(x,df=Inf)
43 t.val <- qt(0.95,df=Inf)
44 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
45 abline(v=0)
46 abline(v=t_c_left,lwd=2,col='red')

```



```

47 abline(v=t_stat, lwd=2, col='green')
48 abline(v=abs(t_c_left), lwd=2, col='green')
49 polygon(c(x[x>=abs(t_c_left)], abs(t_c_left)), c(y[x
    >=abs(t_c_left)], 0), col="red")
50 polygon(c(x[x<=t_c_left], t_c_left), c(y[x<=t_c_left
    ], 0), col="red")
51 text(t_c_left, 0, round(t_c_left, 2))
52 text(t_stat, 0, round(t_stat, 2))
53 text(abs(t_c_left), 0, round(abs(t_c_left), 2))
54 text(3.0, 0.1, "alpha/2=")
55 text(3.6, 0.1, alpha)
56 text(-3.6, 0.1, "alpha/2=")
57 text(-3.0, 0.1, alpha)

```

R code Exa 9.12 Hypothesis test for proportions example2

```

1 #page no 400
2 library(ggplot2)
3 n=150
4 x=43
5 p_dash=round(x/n, 3)
6 p0=30/100
7 q0=1-p0
8 t_stat=abs((p_dash-p0)/(sqrt((p0*q0)/n)))
9 alpha=(1-0.90)/2
10 t_c_left=(qt(alpha, df=Inf))
11 p_value=pt(t_stat, df=Inf)
12 print(paste("critical value=", round(t_c_left, 2)))
13 print(paste("t-static =", round(t_stat, 3)))
14 if(t_stat <= t_c_left)
15 {
16   print(paste("left tail : Reject H0"))
17 }else
18 {
19   print(paste("left tail : Accept H0"))

```

```

20 }
21 t_c_right=qnorm(1-alpha)
22 p_value=1-pnorm(t_stat)
23 if(t_stat >= t_c_right)
24 {
25   print(paste("right tail : Reject H0"))
26 }else
27 {
28   print(paste("right tail : Accept H0"))
29 }
30 two_t_stat=abs((p_dash-p0))/(sqrt((p0*q0)/n))
31 two_t_critical=qnorm(1-alpha/2)
32 p_value=2*(1-pnorm(two_t_stat))
33 if(two_t_stat>=two_t_critical)
34 {
35   print("reject H0")
36 }else
37 {
38   print("accept H0")
39 }
40 x <- seq(-4,4,0.1)
41 y <- dt(x,df=Inf)
42 t.val <- qt(0.95,df=Inf)
43 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
44 abline(v=0)
45 abline(v=t_c_left,lwd=2,col='red')
46 abline(v=t_stat,lwd=2,col='green')
47 abline(v=abs(t_c_left),lwd=2,col='green')
48 polygon(c(x[x>=abs(t_c_left)],abs(t_c_left)),c(y[x
  >=abs(t_c_left)],0),col="red")
49 polygon(c(x[x<=t_c_left],t_c_left),c(y[x<=t_c_left
  ],0),col="red")
50 text(t_c_left,0,round(t_c_left,2))
51 text(t_stat,0,round(t_stat,3))
52 text(abs(t_c_left),0,round(abs(t_c_left),2))
53 text(3.0,0.1,"alpha/2=")
54 text(3.6,0.1,alpha)
55 text(-3.6,0.1,"alpha/2=")

```

```
56 text(-3.0,0.1,alpha)
```

R code Exa 9.13 Hypothesis test for proportions example3

```
1 #page no 400-401
2 library(ggplot2)
3 glass=c
  (1.11,1.07,1.11,1.07,1.12,1.08,0.98,0.98,1.02,0.95,0.95)

4 n=length(glass)
5 xbar=mean(glass)
6 sd=sd(glass)
7 mu0=1
8 t_stat=abs((xbar-mu0)/(sd/sqrt(n)))
9 alpha=(1-0.90)/2
10 t_c_left=(qt(alpha,df=Inf))
11 p_value=pt(t_stat,df=Inf)
12 print(paste("critical value=",round(t_c_left,2)))
13 print(paste("t-static =",round(t_stat,2)))
14 t_c_right=qnorm(1-alpha)
15 p_value=1-pnorm(t_stat)
16 if(t_stat >= t_c_right)
17 {
18   print(paste("right tail : Reject H0"))
19 }else
20 {
21   print(paste("right tail : Accept H0"))
22 }
23 two_t_stat=abs((xbar-mu0)/(sd/sqrt(n)))
24 two_t_critical=qnorm(1-alpha/2)
25 p_value=2*(1-pnorm(two_t_stat))
26 if(two_t_stat>=two_t_critical)
27 {
28   print("reject H0")
29 }else
```

```

30 {
31   print("accept H0")
32 }
33 x <- seq(-4,4,0.1)
34 y <- dt(x,df=Inf)
35 t.val <- qt(0.95,df=Inf)
36 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
37 abline(v=0)
38 abline(v=t_c_left,lwd=2,col='red')
39 abline(v=t_stat,lwd=2,col='green')
40 abline(v=abs(t_c_left),lwd=2,col='green')
41 polygon(c(x[x>=abs(t_c_left)],abs(t_c_left)),c(y[x
    >=abs(t_c_left)],0),col="red")
42 polygon(c(x[x<=t_c_left],t_c_left),c(y[x<=t_c_left
    ],0),col="red")
43 text(t_c_left,0,round(t_c_left,2))
44 text(t_stat,-0.01,round(t_stat,3))
45 text(abs(t_c_left),0,round(abs(t_c_left),2))
46 text(3.0,0.1,"alpha/2=")
47 text(3.6,0.1,alpha)
48 text(-3.6,0.1,"alpha/2=")
49 text(-3.0,0.1,alpha)

```

R code Exa 9.14 Hypothesis test for proportions example4

```

1 #page no 401
2 library(ggplot2)
3 n=420019
4 x=172
5 p_dash=round(x/n,6)
6 p0=.0340/100
7 q0=1-p0
8 t_stat=abs((p_dash-p0)/(sqrt((p0*q0)/n)))
9 alpha=.005
10 t_c_left=(qt(alpha,df=Inf))

```

```

11 p_value=pt(t_stat,df=Inf)
12 print(paste("critical value=",round(t_c_left,2)))
13 print(paste("t-static =",round(t_stat,2)))
14 if(t_stat <= t_c_left)
15 {
16   print(paste("left tail : Reject H0"))
17 }else
18 {
19   print(paste("left tail :Accept H0"))
20 }
21 t_c_right=qnorm(1-alpha)
22 p_value=1-pnorm(t_stat)
23 if(t_stat >= t_c_right)
24 {
25   print(paste("right tail : Reject H0"))
26 }else
27 {
28   print(paste("right tail :Accept H0"))
29 }
30 two_t_stat=abs((p_dash-p0))/(sqrt((p0*q0)/n))
31 two_t_critical=qnorm(1-alpha/2)
32 p_value=2*(1-pnorm(two_t_stat))
33 if(two_t_stat>=two_t_critical)
34 {
35   print("reject H0")
36 }else
37 {
38   print("accept H0")
39 }
40 x <- seq(-3,3,0.1)
41 y <- dt(x,df=Inf)
42 t.val <- qt(0.95,df=Inf)
43 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
44 abline(v=0)
45 abline(v=t_c_left,lwd=2,col='red')
46 abline(v=t_stat,lwd=2,col='green')
47 abline(v=abs(t_c_left),lwd=2,col='green')
48 polygon(c(x[x>=abs(t_c_left)],abs(t_c_left)),c(y[x

```

```
    >=abs(t_c_left)],0),col="red")
49 polygon(c(x[x<=t_c_left],t_c_left ),c(y[x<=t_c_left
    ],0),col="red")
50 text(t_c_left,0,round(t_c_left,2))
51 text(t_stat,0.05,round(t_stat,3))
52 text(abs(t_c_left),0,round(abs(t_c_left),2))
53 text(-3.0,0.1,"alpha/2=")
54 text(-2.8,0.1,alpha)
55 text(2.6,0.1,"alpha/2=")
56 text(2.8,0.1,alpha)
```

Chapter 10

Hypothesis Testing with Two Samples

R code Exa 10.1 Comparing two independent population means example1

```
1 #page no 422-424
2 library(ggplot2)
3 library(MASS)
4 s1=0.866
5 s2=1.00
6 n1=9
7 n2=16
8 numerator=((s1)^2/n1)+((s2)^2/n2))^2
9 deno1=(1/(n1-1))*((s1^2)/n1))^2
10 deno2=(1/(n2-1))*((s2^2)/n2))^2
11 df=round(numerator/(deno1+deno2),0)
12 xbar1=2
13 xbar2=3.2
14 diffmu=0
15 num3=round((xbar1-xbar2)-diffmu,3)
16 term1=round((s1^2)/n1,3)
17 term2=round((s2^2)/n2,3)
18 deno3=round(sqrt(term1+term2),3)
19 t_stat=num3/deno3
```

```

20 alpha=.05
21 twotail_alpha=alpha/2
22 t_c_left=qt(twotail_alpha,df=df)
23 p_value=pnorm(t_stat)
24 if(t_stat <= t_c_left)
25 {
26   print(paste(" left tail : Reject H0"))
27 }else
28 {
29   print(paste(" Left tail : Accept H0"))
30 }
31 two_t_stat=abs(t_stat)
32 two_t_critical=qt(twotail_alpha,df=df)
33 p_value=2*(1-pnorm(two_t_stat))
34 if(two_t_stat>=two_t_critical)
35 {
36   print(" reject H0")
37 }else
38 {
39   print(" accept H0")
40 }
41 dfs <- df
42 x <- seq(-4,4,0.1)
43 y <- dt(x,dfs)
44 t.val <- qt(0.025,df=dfs)
45 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
46 abline(v=0)
47 abline(v=t.val,lwd=2,col='red')
48 abline(v=t_stat,lwd=2,col='green')
49 polygon(c(x[x>=abs(t_c_left)],abs(t_c_left)),c(y[x
  >=abs(t_c_left)],0),col="red")
50 polygon(c(x[x<=t_c_left],t_c_left),c(y[x<=t_c_left
  ],0),col="red")
51 text(t_c_left,0,round(t_c_left,2))
52 text(t_stat,0,round(t_stat,2))
53 text(abs(t_c_left),0,round(abs(t_c_left),2))
54 text(3.0,0.1,expression(frac(alpha,2)))
55 text(3.6,0.1,alpha/2)

```



```

56 text(-3.6,0.1,expression(frac(alpha,2)))
57 text(-3.0,0.1,alpha/2)
58 #The answer may slightly vary due to rounding off
    values

```

R code Exa 10.2 Comparing two independent population means example2

```

1 #page no 425-426
2 library(ggplot2)
3 s1=1.5
4 n1=11
5 xbar1=4
6 s2=1
7 n2=9
8 xbar2=3.5
9 diffmu=0
10 num3=round((xbar1-xbar2)-diffmu,3)
11 term1=round((s1^2)/n1,3)
12 term2=round((s2^2)/n2,3)
13 deno3=round(sqrt(term1+term2),3)
14 t_stat=num3/deno3
15 alpha=.01
16 df=10
17 t_c_right=qt(1-alpha,df=df)
18 p_value=1-pt(t_stat,df=df)
19 print(paste(" critical value= ",round(t_c_right,3)))
20 print(paste("g. t-static =",round(t_stat,2)))
21 if(t_stat >= t_c_right)
22 {
23   print(paste("right tail : Reject H0"))
24 }else
25 {
26   print(paste("right tail : Accept H0"))
27 }
28 dfs <- df

```

```

29 x <- seq(-4,4,0.1)
30 y <- dt(x,dfs)
31 t.val <- qt(1-alpha,df=dfs)
32 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
33 abline(v=0)
34 abline(v=t.val,lwd=2,col='red')
35 abline(v=t_stat,lwd=2,col='green')
36 polygon(c(x[x>=abs(t_c_right)],abs(t_c_right)),c(y[
      x>=abs(t_c_right)],0),col="red")
37 text(t_c_right,0,round(t_c_right,2))
38 text(t_stat,0,round(t_stat,2))
39 text(abs(t_c_right),0,round(abs(t_c_right),2))
40 text(3.0,0.1,expression(alpha))
41 text(3.6,0.1,alpha)
42 text(-3.6,0.1,expression(alpha))
43 text(-3.0,0.1,alpha)

```

R code Exa 10.3 Comparing two independent population means example3

```

1 #page no 426-427
2 library(ggplot2)
3 s1=16
4 n1=35
5 xbar1=74
6 s2=9
7 n2=40
8 xbar2=76
9 diffmu=0
10 num3=round((xbar1-xbar2)-diffmu,3)
11 term1=round((s1^2)/n1,3)
12 term2=round((s2^2)/n2,3)
13 deno3=round(sqrt(term1+term2),3)
14 t_stat=num3/deno3
15 print(paste("t-static =",round(t_stat,2)))
16 alpha=0.05

```

```

17 twotail_alpha=alpha/2
18 df=n1+n2-2
19 two_t_stat=abs(num3)/deno3
20 two_t_critical=qnorm(1-twotail_alpha)
21 p_value=2*(1-pnorm(t_stat))
22 if(two_t_stat>=two_t_critical)
23 {
24   print(" Reject H0")
25 }else
26 {
27   print(" Accept H0")
28 }
29 dfs <- df
30 x <- seq(-4,4,0.1)
31 y <- dt(x,dfs)
32 t.val <- qt(1-alpha,df=dfs)
33 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
34 abline(v=0)
35 abline(v=t.val,lwd=2,col='red')
36 abline(v=t_stat,lwd=2,col='green')
37 abline(v=two_t_critical,lwd=2,col='green')
38 polygon(c(x[x>=abs(two_t_critical)],abs(two_t_
          critical) ),c(y[x>=abs(two_t_critical)],0),col="
          red")
39 text(two_t_critical,0,round(two_t_critical,2))
40 text(t_stat,0,round(t_stat,2))
41 text(abs(two_t_critical),0,round(abs(two_t_critical)
          ,2))
42 text(3.0,0.1,expression(frac(alpha,2)))
43 text(3.6,0.1,twotail_alpha)
44 text(-3.6,0.1,expression(frac(alpha,2)))
45 text(-3.0,0.1,twotail_alpha)

```

R code Exa 10.4 Cohens standards for small medium and large effect sizes

```

1 #page no : 428
2 xbar1=4
3 s1=1.5
4 n1=11
5 xbar2=3.5
6 s2=1
7 n2=9
8 small=0.2
9 medium=0.5
10 large=0.8
11 num1=xbar1-xbar2
12 s_pool=sqrt(((n1-1)*s1^2)+((n2-1)*s2^2)/(n1+n2-2))
13 cohend=(num1/s_pool)
14 if((cohend>=small) & (cohend<=medium))
15 {
16   print(paste("small-size =",round(cohend,3)))
17 }
18 if((cohend>=medium) & (cohend<=large))
19 {
20   print(paste("medium-size =",round(cohend,3)))
21 }
22 if(cohend>=large)
23 {
24   print(paste("large-size is =",round(cohend,3)))
25 }

```

R code Exa 10.5 Test for differences in means

```

1 #page no : 428-429
2 xbar1=8
3 s1=5.4
4 n1=18
5 xbar2=4
6 s2=2.4
7 n2=11

```

```

8 alpha=0.05
9 num1=xbar1-xbar2
10 term1=((n1-1)*s1^2)+((n2-1)*s2^2)/(n1+n2-2)
11 term2=((1/n1)+(1/n2))
12 df=n1+n2-2
13 t_stat=num1/sqrt(term1*term2)
14 t_critical=qt(alpha,df)
15 print(paste("t-static =",round(t_stat,2)))
16 print(paste("t-alpha =",df))
17 if(t_stat>t_critical)
18 {
19     print("Reject H0")
20 }else
21 {
22     print("Accept H0")
23 }

```

R code Exa 10.6 Comparing two independent population proportions example1

```

1 #page no : 431-432
2 xa=20
3 xb=12
4 na=200
5 nb=200
6 pc=round(((xa+xb)/(na+nb)),3)
7 p=1-pc
8 print(paste('1-pc =',p))
9 p_dash_a=xa/na
10 p_dash_b=xb/nb
11 ediff=p_dash_a-p_dash_b
12 print(paste('P\ 'A =',p_dash_a))
13 print(paste('P\ 'B =',p_dash_b))
14 print(paste('P\ 'A - P\ 'B =',ediff))
15 deno1=(pc*p*((1/na)+(1/nb)))

```

```

16 t_stat=(ediff/sqrt(deno1))
17 print(paste('Zc =',round(t_stat,3)))
18 alpha=.1/2
19 df=na+nb-2
20 t_critical=qnorm(1-alpha)
21 if(t_stat>t_critical)
22 {
23     print(" Reject H0")
24 }else
25 {
26     print(" Accept H0")
27 }
28 dfs <- na+nb-2
29 x <- seq(-4,4,0.1)
30 y <- dt(x,dfs)
31 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
32 abline(v=0)
33 abline(v=t_critical,lwd=2,col='red')
34 abline(v=t_stat,lwd=2,col='green')
35 polygon(c(x[x>=abs(t_critical)],abs(t_critical)),c(
    y[x>=abs(t_critical)],0),col="red")
36 text(t_critical,0,round(t_critical,2))
37 text(t_stat,0,round(t_stat,2))
38 text(abs(t_critical),0,round(abs(t_critical),2))
39 text(3.0,0.1,expression(frac(alpha,2)))
40 text(3.6,0.1,alpha)
41 text(-3.6,0.1,expression(frac(alpha,2)))
42 text(-3.0,0.1,alpha)
43 #The answer provided in the textbook is wrong

```

R code Exa 10.7 Two population means with known standard deviations
example1

```

1 #page no : 433-434
2 x1=3

```

```

3 x2=2.9
4 psd1=0.33
5 psd2=0.36
6 n=20
7 mdiff=x1-x2
8 sd=sqrt((psd1^2/n)+(psd2^2/n))
9 t_stat=(mdiff/sd)
10 print(paste("mean diff= ",x1-x2))
11 print(paste("t stat= ",round(t_stat,1)))
12 alpha=.05
13 df=n+n-2
14 t_critical=round(qnorm(1-alpha),3)
15 print(paste("t critical= ",t_critical))
16 if(t_stat>t_critical)
17 {
18     print("Reject H0")
19 }else
20 {
21     print("Accept H0")
22 }
23 dfs <- n+n-2
24 x <- seq(-4,4,0.1)
25 y <- dt(x,dfs)
26 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
27 abline(v=0)
28 abline(v=t_critical,lwd=2,col='red')
29 abline(v=mdiff,lwd=2,col='green')
30 polygon(c(x[x>=abs(t_critical)],abs(t_critical)),c(
    y[x>=abs(t_critical)],0),col="red")
31 text(t_critical,0,round(t_critical,2))
32 text(mdiff,0,round(mdiff,2))
33 text(abs(t_critical),0,round(abs(t_critical),2))
34 text(3.0,0.1,expression(alpha))
35 text(3.6,0.1,alpha)
36 text(-3.6,0.1,expression(alpha))
37 text(-3.0,0.1,alpha)
38 #The answer provided in the textbook is wrong

```

R code Exa 10.8 Two population means with known standard deviations
example2

```
1 #page no : 434-435
2 x1=61.675
3 x2=61.704
4 psd1=10.17
5 psd2=9.55
6 n1=30
7 n2=30
8 mdiff=abs(x1-x2)
9 sd=sqrt((psd1^2/60)+(psd2^2/60))
10 t_stat=((mdiff/sd))
11 alpha=.05
12 t_critical=qnorm(1-alpha)
13 print(paste("t critical=",round(t_critical,2)))
14 p_value=pnorm(t_stat)
15 if(t_stat>t_critical)
16 {
17     print("Reject H0")
18 }else
19 {
20     print("Accept H0")
21 }
22 if(alpha>p_value)
23 {
24     print("Reject H0")
25 }else
26 {
27     print("Accept H0")
28 }
29 dfs <- n1+n2-2
30 x <- seq(-3,3,0.01)
31 y <- dt(x,dfs)
```



```

32 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
33 abline(v=0)
34 abline(v=round(t_stat,2),lwd=2,col='green')
35 polygon(c(x[x>=abs(t_critical)],abs(t_critical)),c(
    y[x>=abs(t_critical)],0),col="red")
36 text(t_stat,0,round(t_stat,2))
37 text(2.0,0.1,expression(alpha))
38 text(2.3,0.1,alpha)
39 text(-2.6,0.1,expression(alpha))
40 text(-2.0,0.1,alpha)
41 #The answer may slightly vary due to rounding off
    values.

```

R code Exa 10.9 Matched or Paired Samples

```

1 #page no : 436-437
2 xd=23.9
3 mud=20.4
4 sd=3.8
5 n=20
6 t_stat=round(((xd-mud)-0)/(sd/sqrt(n)),2)
7 alpha=0.1
8 t_critical1=qt(alpha/2,df=n-2)
9 t_critical2=qt(1-alpha/2,df=n-2)
10 p_value=pnorm(t_stat)
11 print(paste("t_stat=" ,t_stat))
12 print(paste("t_critical1=",round(t_critical1,3)))
13 print(paste("t_critical2=",round(t_critical2,3)))
14 if(t_stat>t_critical2)
15 {
16     print("Reject H0")
17 }else
18 {
19     print("Accept H0")
20 }

```

```

21 dfs <- n-2
22 x <- seq(-5,5,0.1)
23 y <- dt(x,dfs)
24 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
25 abline(v=0)
26 abline(v=t_critical2,lwd=2,col='red')
27 abline(v=t_stat,lwd=2,col='green')
28 polygon(c(x[x>=abs(t_critical2)],abs(t_critical2)),
          c(y[x>=abs(t_critical2)],0),col="red"),
29 text(t_critical2,0,round(t_critical2,2))
30 text(t_stat,0,round(t_stat,2))
31 text(abs(t_critical2),0,round(abs(t_critical2),2))
32 text(3.0,0.1,expression(alpha))
33 text(3.6,0.1,alpha)
34 text(-3.6,0.1,expression(alpha))
35 text(-3.0,0.1,alpha)

```

R code Exa 10.10 Matched or Paired Samples example2

```

1 #page no : 437-439
2 study<-data.frame(subject=c('A','B','C','D','E','F',
   'G','H'),
3                       before=c
4                           (6.6,6.5,9.0,10.3,11.3,8.1,6.3,11.6)
5                           ,
6                           after=c
7                               (6.8,2.4,7.4,8.5,8.1,6.1,3.4,2.0)
8                               )
9 study$diff=study$after-study$before
10 xbar=round(mean(study$diff),3)
11 study$dbarsq=(study$diff-xbar)^2
12 View(study)
13 n=8
14 sumxbar=round(sum(study$dbarsq),2)
15 s=round(sqrt(sumxbar/(n-1)),2)

```

```

12 SE=round(s/sqrt(n),2)
13 df=n-1
14 t_stat=round((xbar-0)/SE,2)
15 alpha=0.05
16 t_critical=qt(alpha,df=n-1)
17 p_value=pnorm(t_stat)
18 if(t_stat<t_critical)
19 {
20   print("Reject H0")
21 }else
22 {
23   print("Accept H0")
24 }
25 dfs <- n-1
26 x <- seq(-5,5,0.1)
27 y <- dt(x,dfs)
28 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
29 abline(v=0)
30 abline(v=t_critical,lwd=2,col='red')
31 abline(v=xbar,lwd=2,col='green')
32 polygon(c(x[x<=t_critical],t_critical),c(y[x<=t_
      critical],0),col="red")
33 text(t_critical,0,round(t_critical,2))
34 text(xbar,0,round(xbar,2))
35 text(3.0,0.1,expression(alpha))
36 text(3.6,0.1,alpha)
37 text(-3.6,0.1,expression(alpha))
38 text(-3.0,0.1,alpha)
39 #The answer may slightly vary due to rounding off
      values.

```

R code Exa 10.11 Matched or Paired Samples example3

```

1 #page no : 439-440
2 w_lift<-data.frame(weight=c("play1","play2","play3"),

```

```

    "play4"),
3         before=c(205,241,338,368),
4         after=c(295,252,330,360))
5 w_lift$diff=w_lift$after-w_lift$before
6 xbar=round(mean(w_lift$diff),3)
7 w_lift$dbarsq=(w_lift$diff-xbar)^2
8 View(w_lift)
9 n=4
10 sumxbar=round(sum(w_lift$dbarsq),2)
11 s=round(sqrt(sumxbar/(n-1)),2)
12 SE=round(s/sqrt(n),2)
13 df=n-1
14 t_stat=round((xbar-0)/SE,2)
15 alpha=0.05
16 t_critical=qt(1-alpha,df=n-1)
17 print(paste("t critical= ",round(t_critical,3)))
18 print(paste("t static= ",round(t_stat,3)))
19 p_value=pnorm(t_stat)
20 if(t_stat>t_critical)
21 {
22   print("Reject H0")
23 }else
24 {
25   print("Accept H0")
26 }
27 dfs <- n-1
28 x <- seq(-5,5,0.1)
29 y <- dt(x,dfs)
30 plot(x,y,type='l',lwd=3,col='blue',xlab='x')
31 abline(v=0)
32 abline(v=t_critical,lwd=2,col='red')
33 abline(v=t_stat,lwd=2,col='green')
34 polygon(c(x[x>=t_critical],t_critical),c(y[x>=t_
    critical],0),col="red")
35 text(t_critical,0,round(t_critical,2))
36 text(t_stat,0,round(t_stat,2))
37 text(3.0,0.1,expression(alpha))
38 text(3.6,0.1,alpha)

```

```
39 text(-3.6,0.1,expression(alpha))
40 text(-3.0,0.1,alpha)
```

Chapter 11

The Chi Square Distribution

R code Exa 11.2 Test of single variance example1

```
1 #page no : 467-468
2 library(ggplot2)
3 library(MASS)
4 options(scipen=999)
5 n=25
6 df=n-1
7 sigma=7.2
8 s_mean=3.5
9 t_stat=round(((n-1)*(s_mean)^2)/(sigma)^2,2)
10 alpha=0.05
11 p=round(pchisq(t_stat,df,lower.tail=TRUE),6)
12 print(paste0(" df=",df))
13 print(paste0(" tstatic=",t_stat))
14 if(alpha > p)
15 {
16   print(paste(" left tail : Reject H0"))
17 }else
18 {
19   print(paste(" Left tail : Accept H0"))
20 }
21 t_critical=13.85
```

```

22 curve(dchisq(x,df=df),from=0,to=100,
23       ylab="density",
24       lwd=2,
25       col='steelblue')
26 x<-seq(0,100)
27 y<-dchisq(x,df=df)
28 abline(v=0)
29 abline(v=t_stat,lwd=2,col='green')
30 polygon(c(x[x<=abs(t_critical)],abs(t_critical)),c(
31         y[x<=abs(t_critical)],0),col="red")
32 text(t_stat,0,round(t_stat,2))
33 text(abs(t_critical),0,round(abs(t_critical),2))
34 text(60,0.04,expression(alpha))
35 text(65,0.04,alpha)
36 mtext(expression(chi^2),side=1,at=5.67,cex=1)

```

R code Exa 11.3 Test of single variance example2

```

1 #page no : 469
2 library(ggplot2)
3 library(MASS)
4 options(scipen=999)
5 n=24
6 df=n-1
7 sigma=0.2
8 s_mean=.11
9 t_stat_num=(n-1)*(s_mean)^2
10 t_stat_deno=sigma*sigma
11 t_stat=round(t_stat_num/t_stat_deno,2)
12 alpha=0.025
13 p=round(pchisq(t_stat,df),6)
14 print(paste0(" tstatic=",t_stat))
15 print(paste0("P( tstatic)=",p))
16 if(alpha > p)
17 {

```

```

18   print(paste(" left tail : Reject H0"))
19 }else
20 {
21   print(paste(" Left tail :Accept H0"))
22 }
23 t_critical1=round(qchisq(alpha,df),2)
24 t_critical2=round(qchisq(1-alpha,df),2)
25 if((t_stat<=t_critical1) & (t_stat>=t_critical2))
26 {
27   print(" Accept H0")
28 }else
29 {
30   print(" Reject H0")
31 }
32 curve(dchisq(x,df=df),from=0,to=100,
33       ylab=" density",
34       lwd=2,
35       col='steelblue ')
36 x<-seq(0,100)
37 y<-dchisq(x,df=df)
38 abline(v=0)
39 abline(v=t_stat,lwd=2,col='red ')
40 abline(v=t_stat,lwd=2,col='green ')
41 polygon(c(x[x<=abs(t_critical1)],abs(t_critical1) ),
42        c(y[x<=abs(t_critical1)],0),col="red")
43 polygon(c(x[x>=abs(t_critical2)],abs(t_critical2) ),
44        c(y[x>=abs(t_critical2)],0),col="red")
45 text(t_stat,0,round(t_stat,2))
46 text(t_critical1,0,t_critical1)
47 text(t_critical2,0,t_critical2)
48 text(60,0.04,expression(frac(alpha, 2)))
49 text(65,0.04,alpha)
50 mtext(expression(chi^2),side=1,at=t_stat,cex=1)
51 #The answer provided in the textbook is wrong.

```

R code Exa 11.4 Goodness of Fit test example1

```
1 #apge no: 470-471
2 students<-data.frame(lterm=c(0,3,6,9),
3                       uterm=c(2,5,8,12),
4                       E=c(50,30,12,8),
5                       O=c(35,40,20,5))
6 View(students)
7 no=nrow(students)
8 df=no-1
9 print(paste("b. df =",df))
```

R code Exa 11.5 Goodness of Fit test example2

```
1 #apge no: 472-473
2 emp<-data.frame(E=c(12,12,12,12,12),
3                 O=c(15,12,9,9,15))
4 emp$OE=emp$O-emp$E
5 emp$OEsq=emp$OE^2
6 emp$OE2=emp$OEsq/emp$E
7 View(emp)
8 no=nrow(emp)
9 t_stat=sum(emp$OE2)
10 df=no-1
11 print(paste("df =",df))
12 print(paste("t stat =",t_stat))
13 alpha=0.05
14 t_critical=qchisq(1-alpha,df=no-1)
15 if(t_stat<=t_critical)
16 {
17   print("Accept H0")
18 }else
19 {
20   print("Reject H0")
21 }
```

```

22 curve(dchisq(x,df=df),from=0,to=15,
23       ylab="density",
24       lwd=2,
25       col='steelblue')
26 x<-seq(0,15,0.1)
27 y<-dchisq(x,df=df)
28 abline(v=0)
29 abline(v=t_stat,lwd=2,col='green')
30 polygon(c(x[x>=t_critical],t_critical),c(y[x>=t_
critical],0),col="red")
31 text(t_stat,0,round(t_stat,2))
32 text(t_critical,0,round(t_critical,2))
33 text(60,0.04,expression(alpha))
34 text(65,0.04,alpha)
35 mtext(expression(chi^2),side=1,at=3,cex=1)

```

R code Exa 11.6 Goodness of Fit test example3

```

1 #apge no: 474-476
2 tv<-data.frame(no_TV=c(0,1,2,3,4),
3                E=c(60,96,330,66,48),
4                O=c(66,119,340,60,15))
5 tv$OE=tv$O-tv$E
6 tv$OEsq=tv$OE^2
7 tv$OE2=tv$OEsq/tv$E
8 View(tv)
9 no=nrow(tv)
10 t_stat=sum(tv$OE2)
11 df=no-1
12 print(paste("df =",df))
13 print(paste("t statistic =",round(t_stat,2)))
14 alpha=0.01
15 t_critical=qchisq(1-alpha,df=no-1)
16 print(paste("critical value =",round(t_critical,3)))
17 if(t_stat<=t_critical)

```

```

18 {
19   print(" Accept H0")
20 }else
21 {
22   print(" Reject H0")
23 }
24 curve(dchisq(x,df=df),from=0,to=35,
25       ylab=" density",
26       lwd=2,
27       col='steelblue')
28 x<-seq(0,35,0.1)
29 y<-dchisq(x,df=df)
30 abline(v=0)
31 abline(v=t_critical,lwd=2,col='red')
32 abline(v=t_stat,lwd=2,col='green')
33 polygon(c(x[x>=t_critical],t_critical),c(y[x>=t_
      critical],0),col="blue")
34 text(20,0.05,expression(alpha ==0.01))
35 mtext(expression(chi^2),side=1,at=t_stat,cex=1)
36 mtext(round(t_stat,2),side=1,at=t_stat+2,cex=1)
37 mtext(round(t_critical,2),side=1,at=t_critical,cex
      =1)

```

R code Exa 11.7 Goodness of Fit test example4

```

1 #apge no: 476-477
2 library(GetoptLong)
3 coin<-data.frame(no_coin=c(0,1,2),
4                  E=c(25,50,25),
5                  O=c(20,57,23))
6 coin$OE=coin$O-coin$E
7 coin$OEsq=coin$OE^2
8 coin$OE2=coin$OEsq/coin$E
9 View(coin)
10 no=nrow(coin)

```

```

11 t_stat=sum(coin$OE2)
12 chisquare=t_stat
13 df=no-1
14 print(paste("df =",df))
15 print(paste("chi square =",chisquare))
16 alpha=0.05
17 t_critical=qchisq(1-alpha,df=no-1)
18 if(t_stat<=t_critical)
19 {
20   print("Accept H0")
21 }else
22 {
23   print("Reject H0")
24 }
25 x<-seq(0,15,length=100)
26 curve(dchisq(x,df=df),from=0,to=15,n=10000,
27       ylab="density",
28       lwd=2,
29       col='steelblue')
30 y<-dchisq(x,df=df)
31 abline(v=0)
32 abline(v=t_critical,lwd=2,col='red')
33 abline(v=t_stat,lwd=2,col='green')
34 polygon(c(x[x>=t_critical],t_critical),c(y[x>=t_
critical],0),col="blue")
35 text(10,0.3,expression(alpha ==0.05))
36 mtext((expression(chi^2==2.14)),side=1,at=t_stat,cex
=1)
37 mtext(round(t_critical,2),side=1,at=t_critical,cex
=1)

```

R code Exa 11.8 Goodness of Fit test example5

```

1 #page no: 478
2 library(GetoptLong)

```

```

3 A=70
4 B=305
5 total_pop=755
6 P_A=A/total_pop
7 P_B=B/total_pop
8 P_A_int_B=P_A*P_B
9 y=P_A_int_B*total_pop
10 print(paste("answer  =",round(y,digits=1)))

```

R code Exa 11.9 Test of Independence example1

```

1 #apge no: 479-480
2 library(MASS)
3 obs_matrix<-matrix(c(111,96,91,96,133,150,48,61,53),
4   ncol=3)
5 rownames(obs_matrix)<-c("comm_stud", "four_y_stud", "
6   non_stud")
7 colnames(obs_matrix)<-c('hours13 ', 'hours46 ', 'hours79
8   ')
9 exp_matrix<-matrix(c
10   (90.57,103.00,104.42,115.19,131.00,132.81,49.24,56.00,56.77)
11   ,ncol=3)
12 rownames(exp_matrix)<-c("comm_stud", "four_y_stud", "
13   non_stud")
14 colnames(exp_matrix)<-c('hours13 ', 'hours46 ', 'hours79
15   ')
16 no_r=nrow(obs_matrix)
17 no_c=ncol(obs_matrix)
18 df=(no_r-1)*(no_c-1)
19 t_stat=0
20 for(i in 1:dim(obs_matrix)[1])
21   {
22     for(j in 1:dim(obs_matrix)[2])
23       {
24         x=(obs_matrix[i,j]-exp_matrix[i,j])^2/exp_

```

```

        matrix[i,j]
18     t_stat=t_stat+x
19   }
20 }
21 print(paste(" critical value",round(t_stat,2)))
22 print(paste(" df =",df))
23 alpha=0.05
24 t_critical=qchisq(1-alpha,df=df)
25 if(t_stat<=t_critical)
26 {
27   print(" Accept H0")
28 }else
29 {
30   print(" Reject H0")
31 }
32 x<-seq(0,15,length=100)
33 curve(dchisq(x,df=df),from=0,to=15,
34       ylab=" density",
35       lwd=2,
36       col='steelblue')
37 y<-dchisq(x,df=df)
38 abline(v=0)
39 abline(v=t_critical,lwd=2,col='red')
40 abline(v=t_stat,lwd=2,col='green')
41 polygon(c(x[x>=t_critical],t_critical),c(y[x>=t_
42       critical],0),col="blue")
43 text(10,0.1,expression(alpha ==0.05))
44 mtext((expression(chi^2==12.99)),side=1,at=t_stat,
45       cex=1)
46 mtext(round(t_critical,2),side=1,at=t_critical,cex
47       =1)

```

R code Exa 11.10 Test of Independence example2

1 #apge no: 481-482

```

2 library(MASS)
3 obs_matrix<-matrix(c
      (35,18,4,42,48,5,53,63,11,15,33,15,10,31,17),ncol
      =5)
4 rownames(obs_matrix)<-c("high","medium","low")
5 colnames(obs_matrix)<-c('high_anx','med_hig_anx','
      med_anx','med_low_anx','low_anx')
6 View(obs_matrix)
7 H_Sum=0
8 for(i in 1:dim(obs_matrix)[1])
9 {
10   j=1
11   H_Sum=H_Sum+obs_matrix[i,j]
12 }
13 H_A_sum=0
14 for(j in 1:dim(obs_matrix)[2])
15 {
16   i=1
17   H_A_sum=H_A_sum+obs_matrix[i,j]
18 }
19 total=0
20 for(i in 1:dim(obs_matrix)[1])
21 {
22   for(j in 1:dim(obs_matrix)[2])
23   {
24     total=total+obs_matrix[i,j]
25   }
26 }
27 E1=(H_Sum*H_A_sum)/total
28 print(paste("a. solution : ", round(E1,2)))
29 L_Sum=0
30 for(i in 1:dim(obs_matrix)[1])
31 {
32   j=4
33   L_Sum=L_Sum+obs_matrix[i,j]
34 }
35 M_L_A_Sum=0
36 for(j in 1:dim(obs_matrix)[2])

```

```

37 {
38   i=3
39   M_L_A_Sum=M_L_A_Sum+obs_matrix[i,j]
40 }
41 E2=(L_Sum*M_L_A_Sum)/total
42 print(paste("b. solution : ", total))
43 print(paste("c. solution : ", round(E2,2)))
44 print(paste("d. solution : ", round(E2,0)))

```

R code Exa 11.11 Test of Homogeneity

```

1 #page no: 483-484
2 library(MASS)
3 obs_matrix<-matrix(c(72,91,84,86,49,88,45,35),ncol
   =4)
4 rownames(obs_matrix)<-c("males","females")
5 colnames(obs_matrix)<-c('dormitory','apart','
   withparents','other')
6 View(obs_matrix)
7 obs_table<-as.table(obs_matrix)
8 View(obs_table)
9 x=chisq.test(obs_table)
10 t_stat=x$statistic
11 df=x$parameter
12 print(paste("\u03C7 = ",round(t_stat,2)))
13 print(paste("df =",df))
14 alpha=0.05
15 t_critical=round(qchisq(1-alpha,df=df),3)
16 if(t_stat<=t_critical)
17 {
18   print("Accept H0")
19 }else
20 {
21   print("Reject H0")
22 }

```



```

23 print(paste(" critical value ",t_critical))
24 x<-seq(0,15,length=100)
25 curve(dchisq(x,df=df),from=0,to=15,
26       ylab="density",
27       lwd=2,
28       col='steelblue')
29 y<-dchisq(x,df=df)
30 abline(v=0)
31 abline(v=t_critical,lwd=2,col='red')
32 abline(v=t_stat,lwd=2,col='green')
33 polygon(c(x[x>=t_critical],t_critical),c(y[x>=t_
      critical],0),col="blue")
34 text(10,0.1,expression(alpha ==0.05))
35 mtext((expression(chi^2==10.13)),side=1,at=t_stat,
      cex=1)
36 mtext(t_critical,side=1,at=t_critical,cex=1)

```

Chapter 12

F Distribution and one way ANOVA

R code Exa 12.1 Test of two variances

```
1 #page no: 515-516
2 library(MASS)
3 n1=10
4 n2=10
5 s1sq=52.3
6 s2sq=89.9
7 f_stat=round((s2sq)/(s1sq),3)
8 alpha=0.01
9 df1=n1-1
10 df2=n2-1
11 t_critical=round(qf(1-alpha,df1=df1,df2=df2),3)
12 if(f_stat<=t_critical)
13 {
14   print("Accept H0")
15 }else
16 {
17   print("Reject H0")
18 }
19 print(paste("the critical value ",t_critical))
```

```

20 print(paste("the f-static ", f_stat))
21 x<-seq(0,10, length=100)
22 curve(df(x, df1=df1, df2=df2), from=0, to=10,
23       ylab="density",
24       lwd=2,
25       col='steelblue')
26 y<-df(x, df1=df1, df2=df2)
27 abline(v=0)
28 abline(v=t_critical, lwd=2, col='red')
29 abline(v=f_stat, lwd=2, col='green')
30 polygon(c(x[x>=t_critical], t_critical), c(y[x>=t_
      critical], 0), col="blue")
31 text(9, 0.1, expression(alpha == 0.01))
32 mtext((expression(F[c] == 1.719)), side=1, at=f_stat, cex
      =1)
33 mtext(round(t_critical, 2), side=1, at=t_critical, cex
      =1)

```

R code Exa 12.2 The F distribution and the F Ratio example1

```

1 #page no : 519-521
2 plan1<-c(5,4.5,4,3)
3 plan2<-c(3.5,7,4.5)
4 plan3<-c(8,4,3.5)
5 plan4<-c(plan1, plan2, plan3)
6 n<-c(length(plan1), length(plan2), length(plan3))
7 s<-c(sum(plan1), sum(plan2), sum(plan3))
8 term1=0
9 term2=0
10 n_sum=0
11 n_count=length(n)
12 for(i in 1:n_count)
13 {
14   term1=term1+(s[i]^2)/n[i]
15   term2=term2+s[i]

```

```

16   n_sum=n_sum+n[i]
17 }
18 ss_between=term1-((term2)^2/n_sum)
19 t_term1=0
20 t_term2=0
21 for(i in 1:length(plan4))
22 {
23 t_term1=t_term1+plan4[i]^2
24 t_term2=t_term2+plan4[i]
25 }
26 s_total=t_term1-((t_term2)^2/n_sum)
27 ss_within=s_total-ss_between
28 df_between=n_count-1
29 df_within=n_sum-n_count
30 df=n_sum-1
31 ms_between=ss_between/df_between
32 ms_within=ss_within/df_within
33 f_stat=ms_between/ms_within
34 print(paste("ss_between= ",round(ss_between,4)))
35 print(paste("ss_within= ",round(ss_within,4)))
36 print(paste("ss_total= ",round(s_total,1)))
37 print(paste("ms_between= ",round(ms_between,4)))
38 print(paste("ms_within= ",round(ms_within,4)))
39 print(paste("f_stat= ",round(f_stat,4)))

```

R code Exa 12.3 The F distribution and the F Ratio example2

```

1 #page no : 521-522
2 v1<-c(2625,2997,4915)
3 v2<-c(5348,5682,5482)
4 v3<-c(6583,8560,3830)
5 v4<-c(7285,6897,9230)
6 v5<-c( 6277,7818,8677)
7 v6<-c(v1,v2,v3,v4,v5)
8 n<-c(length(v1),length(v2),length(v3),length(v4),

```

```

    length(v5))
9  s<-c(sum(v1),sum(v2),sum(v3),sum(v4),sum(v5))
10 term1=0
11 term2=0
12 n_sum=0
13 n_count=length(n)
14 for(i in 1:n_count)
15 {
16   term1=term1+(s[i]^2)/n[i]
17   term2=term2+s[i]
18   n_sum=n_sum+n[i]
19 }
20 ss_between=term1-((term2)^2/n_sum)
21 t_term1=0
22 t_term2=0
23 for(i in 1:length(v6))
24 {
25   t_term1=t_term1+v6[i]^2
26   t_term2=t_term2+v6[i]
27 }
28 s_total=t_term1-((t_term2)^2/n_sum)
29 ss_within=s_total-ss_between
30 df_between=n_count-1
31 df_within=n_sum-n_count
32 df=n_sum-1
33 ms_between=ss_between/df_between
34 ms_within=ss_within/df_within
35 f_stat=ms_between/ms_within
36 print(paste("ss_b=",round(ss_between,0)))
37 print(paste("ss_w=",round(ss_within,0)))
38 print(paste("ss_t=",round(s_total,0)))
39 print(paste("ms_b=",round(ms_between,0)))
40 print(paste("ms_w=",round(ms_within,1)))
41 print(paste("f_stat=",round(f_stat,4)))
42 alpha=0.05
43 df1=df_between
44 df2=df_within
45 p_value=round(pf(f_stat,df1=df1,df2=df2,lower.tail=

```

```

FALSE),4)
46 print(paste("P(F>4.481)= ",p_value))
47 if(f_stat<=p_value)
48 {
49   print(" Accept H0")
50 }else
51 {
52   print(" Reject H0")
53 }
54 if(alpha<=p_value)
55 {
56   print(" Accept H0")
57 }else
58 {
59   print(" Reject H0")
60 }
61 x<-seq(0,5,length=100)
62 curve(df(x,df1=df1,df2=df2),from=0,to=5,
63       ylab="density",
64       lwd=2,
65       col='steelblue')
66 y<-df(x,df1=df1,df2=df2)
67 abline(v=0)
68 abline(v=f_stat,lwd=2,col='green')
69 text(4,0.1,expression(alpha ==0.05))
70 mtext((expression(F==4.481)),side=1,at=f_stat,cex=1)

```

R code Exa 12.4 The F distribution and the F Ratio example3

```

1 #page no : 523–525
2 v1<-c(2.17,1.85,2.83,1.69,3.33)
3 v2<-c(2.63,1.77,3.25,1.86,2.21)
4 v3<-c(2.63,3.78,4.00,2.55,2.45)
5 v4<-c(3.79,3.45,3.08,2.26,3.18)
6 v5<-c(v1,v2,v3,v4)

```

```

7 n<-c(length(v1),length(v2),length(v3),length(v4))
8 s<-c(sum(v1),sum(v2),sum(v3),sum(v4))
9 term1=0
10 term2=0
11 n_sum=0
12 n_count=length(n)
13 for(i in 1:n_count)
14 {
15   term1=term1+(s[i]^2)/n[i]
16   term2=term2+s[i]
17   n_sum=n_sum+n[i]
18 }
19 ss_between=term1-((term2)^2/n_sum)
20 t_term1=0
21 t_term2=0
22 for(i in 1:length(v5))
23 {
24   t_term1=t_term1+v5[i]^2
25   t_term2=t_term2+v5[i]
26 }
27 s_total=t_term1-((t_term2)^2/n_sum)
28 ss_within=s_total-ss_between
29 df_between=n_count-1
30 df_within=n_sum-n_count
31 df=n_sum-1
32 ms_between=ss_between/df_between
33 ms_within=ss_within/df_within
34 f_stat=ms_between/ms_within
35 print(paste("ss_b= ",round(ss_between,0)))
36 print(paste("ss_w= ",round(ss_within,0)))
37 print(paste("ss_t= ",round(s_total,0)))
38 print(paste("ms_b= ",round(ms_between,0)))
39 print(paste("ms_w= ",round(ms_within,1)))
40 print(paste("f_stat= ",round(f_stat,2)))
41 alpha=0.01
42 df1=df_between
43 df2=df_within
44 p_value=round(pf(f_stat,df1=df1,df2=df2,lower.tail=

```

```

FALSE), 4)
45 print(paste("P(F>2.23) = ", round(p_value, 4)))
46 if(alpha<=p_value)
47 {
48   print(" Accept H0")
49 }else
50 {
51   print(" Reject H0")
52 }
53 x<-seq(0, 5, length=100)
54 curve(df(x, df1=df1, df2=df2), from=0, to=5,
55       ylab=" density",
56       lwd=2,
57       col='steelblue ')
58 y<-df(x, df1=df1, df2=df2)
59 abline(v=0)
60 abline(v=f_stat, lwd=2, col='green ')
61 polygon(c(x[x>=f_stat], f_stat ), c(y[x>=f_stat], 0),
62         col="blue")
63 text(4, 0.1, expression(italic(p) ==0.1241))
64 mtext((expression(F==2.23)), side=1, at=f_stat, cex=1)

```

R code Exa 12.5 The F distribution and the F Ratio example4

```

1 #page no : 525–526
2 v1<-c(24, 21, 23, 30, 23)
3 v2<-c(25, 31, 23, 20, 28)
4 v3<-c(23, 27, 22, 30, 20)
5 v4<-c(v1, v2, v3)
6 n<-c(length(v1), length(v2), length(v3))
7 s<-c(sum(v1), sum(v2), sum(v3))
8 n_group=length(n)
9 SM_v1=mean(v1)
10 SM_v2=mean(v2)
11 SM_v3=mean(v3)

```



```

12 SV_v1=var(v1)
13 SV_v2=var(v2)
14 SV_v3=var(v3)
15 v5=c(SM_v1,SM_v2,SM_v3)
16 VG_Means=var(v5)
17 n=length(v1)
18 ms_between=n*VG_Means
19 v6=c(SV_v1,SV_v2,SV_v3)
20 s_pooled=mean(v6)
21 ms_within=s_pooled
22 f_stat=round(ms_between/ms_within,3)
23 df_num=n_group-1
24 df_deno=length(v4)-n_group
25 print(paste("variance= ",round(VG_Means,3)))
26 print(paste("mean = ",round(s_pooled,3)))
27 print(paste("ms_b= ",round(ms_between,3)))
28 print(paste("ms_w= ",round(ms_within,3)))
29 print(paste("f_stat= ",round(f_stat,4)))
30 alpha=0.03
31 df1=df_num
32 df2=df_deno
33 p_value=round(pf(f_stat,df1=df1,df2=df2,lower.tail=
      FALSE),4)
34 print(paste("P(F>0.134) = ",round(p_value,4)))
35 if(alpha<=p_value)
36 {
37   print("Accept H0")
38 }else
39 {
40   print("Reject H0")
41 }

```

Chapter 13

Linear Regression and Correlation

R code Exa 13.2 Linear Equations example1

```
1 #page no : 556
2 x<-seq(-5,20,length=100)
3 y=-1+2*x
4 plot(x,y,type="l",ylab="y=-1+2x")
```

R code Exa 13.5 The Regression Equations example1

```
1 #page no: 571
2 x<-c(65,67,71,71,66,75,67,70,71,69,69)
3 y<-c(175,133,185,163,126,198,153,163,159,151,159)
4 plot(x,y,xlab="3rd exam",ylab="final exam")
```

R code Exa 13.6 The Regression Equations example2

```
1 #page no: 576
2 x<-c(65,67,71,71,66,75,67,70,71,69,69)
3 y<-c(175,133,185,163,126,198,153,163,159,151,159)
4 ypredict=lm(y~x,data=faithful)
5 coeffs=coefficients(ypredict)
6 score1=66
7 eq1=round(coeffs[1],2)+round(coeffs[2],2)*score1
8 print(paste("a. solution =",eq1))
9 score2=90
10 eq2=round(coeffs[1],2)+round(coeffs[2],2)*score2
11 print(paste("b. solution =",eq2))
```
