

R Textbook Companion for
Probability and Statistics for Engineering and
the Sciences
by Jay L Devore¹

Created by
Angela Paul
B.Tech.

Computer Science and Engineering
Government Engineering College, Thrissur

Cross-Checked by
R TBC Team

May 31, 2020

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: Probability and Statistics for Engineering and the Sciences

Author: Jay L Devore

Publisher: Richard Stratton, Boston, USA

Edition: 8

Year: 2012

ISBN: 9780538733526

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 Overview and descriptive statistics	5
2 Probability	14
3 Discrete random variables and probability distributions	23
4 Continuous random variables and probability distributions	42
5 Joint probability distributions and random samples	62
6 Point estimation	77
7 Statistical intervals based on a single sample	84
8 Test of hypotheses based on a single sample	95
9 Inferences based on two samples	111
10 The analysis of variance	124
11 Multifactor analysis of variance	133
12 Simple linear regression and correlation	146
13 Nonlinear and multiple regression	162
14 Goodness of fit tests and categorical data analysis	176

15	Distribution free procedures	188
16	Quality control methods	197

List of R Codes

Exa 1.1	Branches of statistics	5
Exa 1.5	Collecting data	5
Exa 1.8	Dotplots	6
Exa 1.9	Histograms	7
Exa 1.10	Histograms	8
Exa 1.11	Histograms	9
Exa 1.13	Qualitative data	9
Exa 1.14	Mean	10
Exa 1.15	Median	11
Exa 1.17	Measures of variability for sample data	11
Exa 1.18	Measures of variability	12
Exa 1.19	Boxplots	12
Exa 1.20	Box plots that show outliers	13
Exa 2.8	Relations from set theory	14
Exa 2.11	Properties of probability	14
Exa 2.13	Properties of probability	15
Exa 2.14	Properties of probability	15
Exa 2.19	Product rule	16
Exa 2.21	Permutations and combinations	17
Exa 2.22	Permutations and combinations	17
Exa 2.23	Permutations and combinations	18
Exa 2.25	Conditional probability	19
Exa 2.26	Conditional probability	20
Exa 2.30	Bayes Theorem	21
Exa 2.36	Independence of more than two events	22
Exa 3.7	Probability distributions for discrete random variables	23
Exa 3.8	Probability distributions for discrete random variables	24
Exa 3.9	Probability distributions for discrete random variables	25

Exa 3.10	Probability distributions for discrete random variables	26
Exa 3.13	Cumulative distribution function	27
Exa 3.15	Cumulative distribution function of discrete random variable	28
Exa 3.16	Expected value of X	29
Exa 3.17	Expected value of X	29
Exa 3.23	Expected value of a function	29
Exa 3.24	The variance of X	30
Exa 3.25	The variance of X	31
Exa 3.26	Rules of variance	31
Exa 3.29	Binomial experiment	32
Exa 3.30	Binomial experiment	33
Exa 3.31	Binomial random variable and distribution	33
Exa 3.32.1	Cumulative distribution function	34
Exa 3.32.2	Using binomial tables	34
Exa 3.32.3	Using binomial tables	35
Exa 3.32.4	Using binomial tables	35
Exa 3.33	Using binomial tables	35
Exa 3.34	Mean and variance of X	36
Exa 3.35	Hypergeometric distribution	37
Exa 3.36	Hypergeometric distribution	37
Exa 3.37	Mean and variance of hypergeometric distribution	38
Exa 3.38	Negative binomial distribution	38
Exa 3.39	Poisson probability distribution	39
Exa 3.40	Poisson distribution as a limit	39
Exa 3.41	Mean and variance in Poisson distribution	40
Exa 3.42	Poisson process	40
Exa 4.4	Probability density function	42
Exa 4.5	Uniform distribution	43
Exa 4.7	Computing probabilities	43
Exa 4.10	Expected values	44
Exa 4.11	Expected value of a function	44
Exa 4.12	Variance and standard deviation	45
Exa 4.13.a	Standard normal distribution	45
Exa 4.13.b	Standard normal distribution	46
Exa 4.13.c	Standard normal distribution	47
Exa 4.13.d	Standard normal distribution	48
Exa 4.14	Percentiles of standard normal distribution	48

Exa 4.15	z critical values	49
Exa 4.16	Nonstandard normal distributions	50
Exa 4.17	Nonstandard normal distributions	51
Exa 4.18	Percentiles of arbitrary normal distribution	52
Exa 4.20	Approximating binomial distribution	53
Exa 4.21	Exponential distribution	53
Exa 4.22	Exponential distribution	54
Exa 4.23	Gamma distribution	54
Exa 4.24	Gamma distribution	55
Exa 4.25	Weibull distribution	56
Exa 4.26	Weibull distribution	57
Exa 4.27	Lognormal distribution	57
Exa 4.28	Beta distribution	58
Exa 4.29	Probability plots	59
Exa 4.30	Probability plots	60
Exa 4.31	Beyond normality	60
Exa 5.1	Joint probability mass function	62
Exa 5.2	Marginal probability mass function	62
Exa 5.3	Joint probability density function	63
Exa 5.4	Marginal probability density function	64
Exa 5.5	Marginal probability density function	65
Exa 5.8	Independent random variables	65
Exa 5.9	Multinomial distribution	66
Exa 5.12	Conditional distributions	66
Exa 5.14	Expected values	67
Exa 5.15	Covariance	68
Exa 5.18	Correlation	69
Exa 5.19	Statistics and their distributions	70
Exa 5.24	Standard error of the mean	71
Exa 5.25	Normal population distribution	72
Exa 5.26	Central limit theorem	73
Exa 5.27	Central limit theorem	74
Exa 5.29	Linear combination	74
Exa 5.30	Difference between two random variables	75
Exa 5.31	Normal random variables	76
Exa 6.1	Some general concepts of point estimation	77
Exa 6.2	Some general concepts of point estimation	77
Exa 6.3	Some general concepts of point estimation	78

Exa 6.9	Standard error of an estimator	79
Exa 6.10	Standard error of an estimator	80
Exa 6.11	Standard error of an estimator	80
Exa 6.13	Method of moments	81
Exa 6.14	Method of moments	82
Exa 6.21	The invariance principle	82
Exa 7.2	Basic properties of confidence intervals	84
Exa 7.3	Other levels of confidence	85
Exa 7.4	Confidence level and precision and sample size	85
Exa 7.6	Large sample CI for mean	86
Exa 7.7	Large sample CI for mean	87
Exa 7.8	CI for population proportion	87
Exa 7.10	Confidence bounds	88
Exa 7.11	One sample t confidence interval	88
Exa 7.12	A prediction interval for a single future value	90
Exa 7.13	Prediction interval	91
Exa 7.14	Tolerance intervals	91
Exa 7.15	CI for variance and sd of normal population	93
Exa 8.1	Hypotheses and test procedures	95
Exa 8.2	Hypotheses and test procedures	96
Exa 8.3	Hypotheses and test procedures	97
Exa 8.4	Hypotheses and test procedures	98
Exa 8.5	Hypotheses and test procedures	99
Exa 8.6	Tests about a population mean	100
Exa 8.7	Tests about a population mean	100
Exa 8.8	Tests about a population mean	101
Exa 8.9	Tests about a population mean	102
Exa 8.11	Tests concerning a population proportion	103
Exa 8.12	Tests concerning a population proportion	104
Exa 8.13	Tests concerning a population proportion	105
Exa 8.14	P values	106
Exa 8.15	P values	107
Exa 8.17	P values for z tests	108
Exa 8.18	P values for t tests	109
Exa 8.19	Interpreting P values	109
Exa 9.1	Test procedures for normal population with known vari- ances	111
Exa 9.2	Using a comparison to identify causality	112

Exa 9.3	Beta and the choice of sample size	113
Exa 9.4	Large sample tests	113
Exa 9.5	Confidence intervals for difference of means	114
Exa 9.6	Two sample t test and confidence interval	115
Exa 9.7	Two sample t test and confidence interval	116
Exa 9.9	Paired t test	117
Exa 9.10	Paired t confidence interval	118
Exa 9.11	Inferences concerning difference between population proportions	119
Exa 9.12	Type II error probabilities and sample sizes	120
Exa 9.13	A large sample confidence interval	121
Exa 9.14	F test for equality of variances	122
Exa 10.1	Single factor ANOVA	124
Exa 10.2	F distribution and the F test	126
Exa 10.3	F distribution and the F test	127
Exa 10.4	Sums of squares	128
Exa 10.6	Multiple comparisons in ANOVA	129
Exa 10.9	Unequal sample sizes	130
Exa 10.10	Unequal sample sizes	130
Exa 10.11	A random effects model	131
Exa 11.1	Two factor ANOVA	133
Exa 11.3	Fixed effects model	133
Exa 11.5	Randomized block experiments	134
Exa 11.6	Randomized block experiments	135
Exa 11.7	Two factor ANOVA with Kij greater than 1	136
Exa 11.9	Models with mixed and random effects	137
Exa 11.10	Three factor ANOVA	138
Exa 11.11	Latin square designs	139
Exa 11.12	Factorial experiments with p factors at two levels	140
Exa 11.13	Factorial experiments with p factors at two levels	142
Exa 12.1	The simple linear regression model	146
Exa 12.2	The simple linear regression model	147
Exa 12.3	A linear probabilistic model	147
Exa 12.4	Estimating model parameters	148
Exa 12.5	Estimating model parameters	149
Exa 12.6	Estimating model parameters	149
Exa 12.7	Estimating model parameters	150
Exa 12.8	Estimating model parameters	151

Exa 12.9	Coefficient of determination	152
Exa 12.11	Confidence interval for β_1	153
Exa 12.12	Hypothesis testing procedures	155
Exa 12.13	Inferences concerning mean	156
Exa 12.14	Prediction interval for a future value of Y	157
Exa 12.15	Correlation	158
Exa 12.16	Inferences about the population correlation coefficient	158
Exa 12.17	Inferences about the population correlation coefficient	158
Exa 12.18	Other inferences concerning ρ	159
Exa 12.19	Other inferences concerning ρ	160
Exa 13.2	Diagnostic plots	162
Exa 13.4	Regression with transformed variables	163
Exa 13.7	Polynomial regression	164
Exa 13.8	Estimating parameters	165
Exa 13.9	Statistical intervals and test procedures	166
Exa 13.10	Centering x values	167
Exa 13.12	Estimating parameters	168
Exa 13.13	Estimating parameters	169
Exa 13.14	Model utility test	171
Exa 13.15	Inferences in multiple regression	172
Exa 13.16	Inferences in multiple regression	172
Exa 13.18	Transformations	174
Exa 14.1	Goodness of fit tests when category probabilities are completely specified	176
Exa 14.2	P values for chi squared tests	177
Exa 14.3	Chi square when underlying distribution is continuous	178
Exa 14.4	Chi square when underlying distribution is continuous	179
Exa 14.5	Chi square when parameters are estimated	180
Exa 14.6	Chi square when parameters are estimated	181
Exa 14.9	Goodness of fit for discrete distributions	182
Exa 14.10	Goodness of fit for continuous distributions	183
Exa 14.12	A special test for normality	184
Exa 14.13	Testing for homogeneity	185
Exa 14.14	Testing for independence	186
Exa 15.1	The Wilcoxon signed rank test	188
Exa 15.2	Paired observations	188
Exa 15.3	Large sample approximation	189
Exa 15.4	The Wilcoxon rank sum test	190

Exa 15.5	A normal approximation for W	190
Exa 15.6	The Wilcoxon signed rank interval	191
Exa 15.8	The Wilcoxon rank sum interval	192
Exa 15.9	The Kruskal Wallis test	193
Exa 15.10	Friedman test for a randomized block experiment . . .	194
Exa 16.1	X bar chart based on known parameter values	197
Exa 16.2	X bar charts based on estimated parameters	198
Exa 16.3	X bar charts based on estimated parameters	198
Exa 16.4	The S chart	199
Exa 16.5	R chart	200
Exa 16.6	The p chart	202
Exa 16.7	c chart for number of defectives	203
Exa 16.8	CUSUM procedures	204
Exa 16.9	Computational version	205
Exa 16.10	Designing a CUSUM procedure	206
Exa 16.11	Acceptance sampling	206
Exa 16.12	Designing a single sample plan	207

Chapter 1

Overview and descriptive statistics

R code Exa 1.1 Branches of statistics

```
1 #Ex1.1, Page 4
2
3 library(lattice)
4
5 data<-c
      (6.1,12.6,34.7,1.6,18.8,2.2,3.0,2.2,5.6,3.8,2.2,3.1,1.3,1.1,14.1,
6
7 stem(data,scale=2)
8
9 hist(data,main="Histogram for charity fundraising
      percentage data",xlab="FundRaising",col="grey",
      xlim=c(0,100),ylim=c(0,40))
```

R code Exa 1.5 Collecting data

```

1 #Ex1.5, Page 11
2
3 Adhesive_Type<-c(1,1,2,2)
4 Conductor_material<-c(1,2,1,2)
5 Observed_bond_strength<-c(82,75,84,78,77,87,80,90)
6 mat1<-matrix(Observed_bond_strength,nrow=4,ncol=2)
7 Average<-c(sum(mat1[1,])/2,sum(mat1[2,])/2,sum(mat1[3,])/2,sum(mat1[4,])/2)
8 mat1<-cbind(Adhesive_Type,Conductor_material,mat1,
              Average)
9 colnames(mat1)[3]<-"Observed bond strength"
10 print(mat1)
11
12 n<-length(Adhesive_Type)/2
13 plot(Conductor_material,Average,main="Average bond
        strengths",xlab="Conducting material",ylab="
        Average strength")
14 segments(Conductor_material[1],Average[1],Conductor_
        material[2],Average[2])
15 segments(Conductor_material[3],Average[3],Conductor_
        material[4],Average[4])

```

R code Exa 1.8 Dotplots

```

1 #Ex1.8, Page 15
2
3 library(graphics)
4
5 data<-c
      (10.8,6.9,8.0,8.8,7.3,3.6,4.1,6.0,4.4,8.3,8.1,8.0,5.9,5.9,7.6,8.9
6
7 #To create a stacked dotplot
8 stripchart(data,method="stack",offset=0.5,pch=20,at
      =0,main="Dotplot of given data")

```

R code Exa 1.9 Histograms

```
1 #Ex1.9 , Page 17
2
3 Hits_per_game<-0:27
4 No_of_games<-c
    (20,72,209,527,1048,1457,1988,2256,2403,2256,1967,1509,1230,834,56

5 sum1<-sum(No_of_games)
6 Relative_frequency<-round(No_of_games/sum1,digits=4)
7 df1<-data.frame(Hits_per_game,No_of_games,Relative_
    frequency)
8 print(df1)
9
10 #Proportion of games with atmost 2 hits
11 f1<-0
12 for(i in Hits_per_game){
13     if(i<=2){
14         f1<-f1+df1[i+1,3]
15     }
16 }
17 print(paste("Proportion of games with atmost two
    hits:",f1))
18
19 #Proportion of games with between 5 and 10 hits
20 f2<-0
21 for(j in Hits_per_game){
22     if(j>=5 && j<=10){
23         f2<-f2+df1[j+1,3]
24     }
25 }
26 print(paste("Proportion of games with between 5 and
    10 hits:",f2))
27
```

```

28 par(mfrow=c(1,2))
29 plot(Hits_per_game,Relative_frequency,type="h",xlim=
      c(0,20),main="Histogram of hits/game")
30 barplot(Relative_frequency,width=1,space=0,xlim=c
      (0,25),main="Histogram of hits/game",xlab="Hits/
      game",ylab="Relative frequency")

```

R code Exa 1.10 Histograms

```

1 #Ex1.10, Page 18
2 #Answers may vary since different class intervals
  have been taken
3
4 #Please install and load the following packages: 1)
  HistogramTools 2)ash
5 library(HistogramTools)
6 library(ash)
7
8 data<-c
  (2.97,4.00,5.20,5.56,5.94,5.98,6.35,6.62,6.72,6.78,6.80,6.85,6.94
9
10 #Class intervals can also be created using
  classIntervals() from ClassInt library:
11 #classIntervals(data,style="pretty",intervalClosure
  ="left") which creates different intervals as
  compared to those of textbook
12
13 #To create class intervals
14 ci<-seq(1,19,2)
15
16 freq<-table(cut(data,ci,right=FALSE))
17 rel_freq<-freq/length(data)
18 print(cbind(freq,rel_freq))
19

```



```

20 #To display histogram
21 PlotRelativeFrequency(hist(data,breaks = ci,right=
    FALSE,plot=FALSE),xlab="BTUIN",main="Histogram of
    energy consumption data",ylim=c(0,0.30))

```

R code Exa 1.11 Histograms

```

1 #Ex1.11, Page 20
2
3 data<-c
    (11.5,12.1,9.9,9.3,7.8,6.2,6.6,7.0,13.4,17.1,9.3,5.6,5.7,5.4,5.2,
4 #Unequal class widths
5 ci<-c(2,4,6,8,12,20,30)
6
7 #To find frequency, relative frequency and density
    of data manually
8 freq<-table(cut(data,ci,right=FALSE))
9 rel_freq<-freq/length(data)
10 density<-rel_freq/diff(ci)
11
12 print(cbind(freq,rel_freq,density))
13
14 #To create histogram of the bond strength data
15 hist(data,breaks=ci,freq=FALSE,right=FALSE,main="
    Histogram for bond strength data",xlab="Bond
    Strength")

```

R code Exa 1.13 Qualitative data

```

1 #Ex1.13, Page 23
2 #Answers may vary slightly due to rounding off of
    values

```

```

3
4 Rating<-c("A","B","C","D","F","Don't know")
5 Frequency<-c(478,893,680,178,100,172)
6 Relative_frequency<-Frequency/sum(Frequency)
7 df1<-data.frame(Rating,Frequency,Relative_frequency)
8 print(df1)
9
10 print(paste("Total frequency:",sum(Frequency)))
11 print(paste("Total relative frequency:",sum(Relative
    _frequency)))
12
13 barplot(df1$Relative_frequency,space=1,names.arg=df1
    $Rating,ylim=c(0,0.4),xlab="Rating",ylab="
    Relative Frequency",main="Chart of relative
    frequency vs. rating")

```

R code Exa 1.14 Mean

```

1 #Ex1.14, Page 29
2 #Steam and leaf plot differs since number of stem
    parts=1
3
4 library(aplpack)
5
6 x<-c
    (16.1,9.6,24.9,20.4,12.7,21.2,30.2,25.8,18.5,10.3,25.3,14.0,27.1,
7
8 sl<-stem.leaf(x,unit=0.1,m=1,style="bare")
9
10 s<-sum(x)
11 mean<-s/length(x)
12 print(paste("Mean value:",mean))
13 print(paste("The value",sl$upper,"is an outlier"))
14 print(paste("Mean value when",sl$upper,"is excluded="

```

```
”,(s-45)/(length(x)-1)))
```

R code Exa 1.15 Median

```
1 #Ex1.15, Page 30
2 #Answers may vary slightly due to rounding off of
  values
3
4 data<-c
  (62.3,62.8,63.6,65.2,65.7,66.4,67.4,68.4,68.8,70.8,75.7,79.0)

5
6 dotplot(data,xlab="Duration")
7
8 n<-length(data)
9 n1<-data[n/2]
10 n2<-data[(n/2)+1]
11 med<-(n1+n2)/2
12 print(paste("Median:",med))
13
14 max<-max(data)
15 #If maximum value is omitted
16 data1<-data[1:length(data)-1]
17 mean1<-mean(data1)
18 print(paste("Mean when maximum value ",max," is
  omitted:",mean1))
```

R code Exa 1.17 Measures of variability for sample data

```
1 #Ex1.17, Page 36
2
3 Car<-1:11
```

```

4  xi<-c
    (27.3,27.9,32.9,35.2,44.9,39.9,30.0,29.7,28.5,32.0,37.6)

5
6  xibar<-mean(xi)
7  s<-sd(xi)
8
9  print(paste("The size of the representative
    deviation from the mean,",round(xibar,digits=2),"
    is roughly",round(s,digits=2),"mpg"))

```

R code Exa 1.18 Measures of variability

```

1  #Ex1.18, Page 38
2  #Answers may vary slightly due to rounding off of
    values
3
4  data<-c
    (154,142,137,133,122,126,135,135,108,120,127,134,122)

5
6  print(paste("s:",sd(data)))

```

R code Exa 1.19 Boxplots

```

1  #Ex1.19, Page 40
2
3  obs<-c
    (40,52,55,60,70,75,85,85,90,90,92,94,94,95,98,100,115,125,125)

4
5  #To find five-number summary
6  print(summary(fivenum(obs)))

```

```
7
8 boxplot(obs, horizontal=TRUE, xlab="Depth", main="
  Boxplot of the corrosion data")
```

R code Exa 1.20 Box plots that show outliers

```
1 #Ex1.20, Page 41
2
3 data<-c
  (9.69,13.16,17.09,18.12,23.70,24.07,24.29,26.43,30.75,31.54,35.07
4
5 #To find relevant summary quantities
6 print(summary(fivenum(data)))
7
8 c<-seq(0,1600,by=200)
9
10 boxplot(data, horizontal=TRUE, xlab="Daily nitrogen
  load", main="Boxplot of daily nitrogen load data
  showing mild and extreme outliers")
```

Chapter 2

Probability

R code Exa 2.8 Relations from set theory

```
1 #Ex2.8, Page 53
2
3 A<-c(0,1,2,3,4)
4 B<-c(3,4,5,6)
5 C<-c(1,3,5)
6 U<-union(A,B)
7
8 cat("A'=",setdiff(U,A),"\\n")
9 cat("A U B =",union(A,B),"\\n")
10 cat("A U C =",union(A,C),"\\n")
11 cat("A n B =",intersect(A,B),"\\n")
12 cat("A n C =",intersect(A,C),"\\n")
13 cat("(A n C)'=",setdiff(U,intersect(A,C)))
```

R code Exa 2.11 Properties of probability

```
1 #Ex2.11, Page 56
2
```

```

3 #S={U,D}
4
5 p<-seq(0,1,by=0.25)
6
7 for(i in p){
8   pu<-i
9   pd<-1-i
10  cat("P(U) :",pu,"\n")
11  cat("P(D) :",pd,"\n\n")
12 }
13 cat("p value can represent any fixed value between 0
      and 1")

```

R code Exa 2.13 Properties of probability

```

1 #Ex2.13, Page 59
2 #Answers may vary slightly due to rounding off of
  values
3
4 #For A to occur, at least one of the individual
  components must fail
5 #To find P(A')=SSSSS
6 p<-90/100
7 pa_dash<-p^5
8 cat("P(A') :",pa_dash,"\n")
9
10 pa<-1-pa_dash
11 cat("P(A) :",1-pa_dash,"\n\n")
12 cat("Roughly",round(pa*100,digits=0),"% of the
      systems will fail")

```

R code Exa 2.14 Properties of probability

```

1 #Ex2.14, Page 60
2
3 #P(A)=>probability that a household gets internet
  service
4 pa<-60/100
5
6 #P(B)=>probability that a household gets TV service
7 pb<-80/100
8
9 #P(A n B)=>probability that a household gets both
  services
10 pab<-50/100
11
12 #To find probability that a household gets at least
  one of the services
13 p1<-pa+pb-pab
14 cat("P(at least one service)=P(A U B) =",p1,"\n")
15
16 pa_b<-p1-pa
17 pb_a<-p1-pb
18 #To find probability that a household gets exactly
  one service
19 p2<-pa_b+pb_a
20 cat("P(exactly one service)=",p2)

```

R code Exa 2.19 Product rule

```

1 #Ex2.19, Page 67
2
3 #No of appliance dealers
4 n1<-5
5
6 #No of plumbing contractors
7 n2<-12
8

```



```

9 #No of electrical contractors
10 n3<-9
11
12 cat("There are",n1*n2*n3,"ways to first choose an
    appliance dealer, then a plumbing contractor and
    finally an electrical contractor")

```

R code Exa 2.21 Permutations and combinations

```

1 #Ex2.21, Page 68
2
3 #No of teaching assistants
4 n<-10
5 #No of questions in the first exam
6 k<-4
7
8 #To find number of permutations
9 p<-factorial(n)/factorial(n-k)
10 cat("The professor could give",p,"different four-
    question exam papers without using the same
    graders")

```

R code Exa 2.22 Permutations and combinations

```

1 #Ex2.22, Page 69
2 #Answers may vary slightly due to rounding off of
    values
3
4 #Total no of songs
5 n<-100
6
7 #No of songs by the Beatles
8 m<-10

```

```

9
10 #Probability that the 1st Beatles song heard is the
    5th song played
11 t<-5
12 p1<-(factorial(n-m)/factorial(n-m-(t-1))*m/(
    factorial(n)/factorial(n-t)))
13 cat("P(1st B is 5th song played):",p1,"\n")
14
15 #Probability that the 1st Beatles song heard is the
    1st,2nd,3rd,4th or 5th song played
16 p2<-0
17 for(i in 1:5){
18   p2<-p2+(choose(n-i,m-1))/(choose(n,m))
19 }
20 cat("P(1st B is 1st,2nd,3rd,4th or 5th song played):
    ",p2,"\n")

```

R code Exa 2.23 Permutations and combinations

```

1 #Ex2.23, Page 70
2 #Answers may vary slightly due to rounding off of
    values
3
4 #Total no of printers
5 n<-25
6 l<-10
7 i<-15
8
9 #Probability that 6 random printers selected will
    have 3 laser printers
10 r1<-6
11 r<-3
12 ND3<-(choose(l,r)*choose(i,r))/choose(n,r1)
13 cat("P(3 laser printers are selected among the 6
    selected):",ND3,"\n")

```

```

14
15
16 #To find the probability that at least 3 inkjet
    printers are selected
17 prob<-0
18 for(j in 3:6){
19     num<-choose(i,j)*choose(l,r1-j)
20     denom<-choose(n,r1)
21     prob<-prob+(num/denom)
22 }
23 cat("P(at least 3 inkjet printers are selected):",
    prob,"\n")

```

R code Exa 2.25 Conditional probability

```

1 #Ex2.25, Page 74
2
3 #Probability that memory card is purchased
4 pa<-0.6
5
6 #Probability that battery is purchased
7 pb<-0.4
8
9 #Probability that both memory card and battery is
    purchased
10 pab<-0.3
11
12 #Probability that an optional card was purchased
    given that individuals purchased extra battery
13 pa_b<-pab/pb
14 cat("P(A|B)=",pa_b,"\n")
15 cat("Of all individuals who purchased an extra
    battery,",pa_b*100,"% purchased an optional
    memory card\n")
16

```

```

17 #Probability that an additional battery was
    purchased given that individuals purchased an
    optional memory card
18 pb_a<-pab/pa
19 cat("P(B|A)=" ,pb_a,"\\n")
20
21 if(pa_b!=pb_a)  print(paste("P(A|B) is not equal to
    P(B|A)"))

```

R code Exa 2.26 Conditional probability

```

1 #Ex2.26, Page 75
2 #Answers may vary slightly due to rounding off of
    values
3
4 #Package to be installed: VennDiagram
5 library(VennDiagram)
6
7 #A=>Arts , B=>Books , C=>Cinema
8 pa<-0.14
9 pb<-0.23
10 pc<-0.37
11 pab<-0.08
12 pbc<-0.13
13 pac<-0.09
14 pabc<-0.05
15
16 #To illustrate the Venn diagram using the given
    probabilities
17 draw.triple.venn(area1=pa,area2=pb,area3=pc,n12=pab,
    n23=pbc,n13=pac,n123=pabc,category=c("Arts","
    Books","Cinema"))
18
19 cat("With the help of Venn diagram:\\n")
20 #P(A|B)

```

```

21 pa_b<-pab/pb
22 print(paste("P(A|B)=",pa_b))
23
24 #P(A|B U C)
25 pa_bUc<-(pab+pac-pabc)/(pb+pc-pbc)
26 print(paste("P(A|B U C)=",pa_bUc))
27
28 #P(A|reads at least once)=P(A|A U B U C)
29 pa_aUbUc<-pa/(pa+pb+pc-pab-pac-pbc+pabc)
30 print(paste("P(A|reads at least once)=",pa_aUbUc))
31
32 #P(A U B|C)
33 paUb_c<-(pac+pbc-pabc)/pc
34 print(paste("P(A U B|C)=",paUb_c))

```

R code Exa 2.30 Bayes Theorem

```

1 #Ex2.30 , Page 78
2
3 # % of messages which come into account 1
4 pa1<-70/100
5
6 # % of messages which come into account 2
7 pa2<-20/100
8
9 # % of messages which come into account 3
10 pa3<-10/100
11
12 # % of messages which are spam in account 1
13 pb_a1<-1/100
14
15 # % of messages which are spam in account 2
16 pb_a2<-2/100
17
18 # % of messages which are spam in account 3

```

```

19 pb_a3<-5/100
20
21 #To find the probability that a randomly selected
    message is spam
22 #Using law of total probability
23 pb<-pa1*pb_a1+pa2*pb_a2+pa3*pb_a3
24 print(paste("P(Randomly selected message is spam):",
    pb,"and thus",pb*100,"% messages will be spam"))

```

R code Exa 2.36 Independence of more than two events

```

1 #Ex2.36, Page 85
2 #Answers may vary slightly due to rounding off of
    values
3
4 pai<-rep(0.9,times=6)
5
6 #To find probability that system lifetime exceeds t0
7 prob1<-pai[1]*pai[2]*pai[3]+pai[4]*pai[5]*pai[6]-
    prod(pai)
8 print(paste("P(system lifetime exceeds t0):",prob1))
9
10 #To find probability that system lifetime is at
    least t0
11 prob2<-(1-(1-pai[1])*(1-pai[2]))^3
12 print(paste("P(system lifetime is at least t0):",
    prob2))

```

Chapter 3

Discrete random variables and probability distributions

R code Exa 3.7 Probability distributions for discrete random variables

```
1 #Example 3.7, Page 97
2 #Storing all probabilities in an array
3 prob<-c(0.05,0.1,0.15,0.25,0.20,0.15,0.10)
4 P<-array(c(prob),dim=c(1,7,1))
5
6 #P(X<=2)=P(X=0)+P(X=1)+P(X=2)
7 p1<-P[1,1,1]+P[1,2,1]+P[1,3,1]
8 print(paste("P(X<=2) =",p1))
9
10 #P(X>=3)=1-P(X<=2)
11 p2<-1-(P[1,1,1]+P[1,2,1]+P[1,3,1])
12 print(paste("P(X>=3) =",p2))
13
14 #P(2<=X<=5)=P(X=2,3,4 or 5)
15 p3<-P[1,3,1]+P[1,4,1]+P[1,5,1]+P[1,6,1]
16 print(paste("P(2<=X<=5) =",p3))
17
18 #P(2<X<5)=P(X=3,4)
19 p4<-P[1,4,1]+P[1,5,1]
```

```
20 print(paste("P(2<X<5) =",p4))
```

R code Exa 3.8 Probability distributions for discrete random variables

```
1 #Example 3.8, Page 97
2 #The answers may slightly vary due to rounding off
  of values
3 p1<-c(1,0,2,2,3,0,4,1,5,2,6,0)
4 row.names<-c("Lot","Number of defectives")
5 column.names<-c("","","","","","","")
6 matrix.names<-c("Number of defectives in each lot")
7 p<-array(c(p1),dim=c(2,6,1),dimnames=list(row.names,
  column.names,matrix.names))
8 print(p)
9 #p(0)=P(X=0)
10 c<-0
11 for(i in 1:6) {
12   d<-p[2,i,1]
13   if(d==0){
14     c<-c+1
15   }
16 }
17 d<-c/6
18 print(paste("p(0)=",d))
19
20 #p(1)=P(X=1)
21 c<-0
22 for(i in 1:6){
23   d<-p[2,i,1]
24   if(d==1){
25     c<-c+1
26   }
27 }
28 d<-c/6
29 print(paste("p(1)=",d))
```



```

30
31 #p(2)=P(X=2)
32 c<-0
33 for(i in 1:6){
34     d<-p[2,i,1]
35     if(d==2){
36         c<-c+1
37     }
38 }
39 d<-c/6
40 print(paste("p(2)=",d))

```

R code Exa 3.9 Probability distributions for discrete random variables

```

1 #Ex3.9, Page 98
2 #X=0 if customer purchases laptop computer
3 #X=1 if customer purchases desktop computer
4 #p(0)
5 p0<-80/100
6 print(paste("Probability that next customer
  purchases laptop model:",p0))
7 p1<-20/100
8 print(paste("Probability that next customer
  purchases desktop model:",p1))
9 p1<-c(0.8,0.2,0)
10 row.names<-c("if x=0","if x=1","if (x!=0 or 1)")
11 column.names<-c("")
12 matrix.names<-c("p(x)")
13 p<-array(c(p1),dim=c(3,1,1),dimnames=list(row.names,
  column.names,matrix.names))
14 print(p)
15 a<- 1
16 p<- dbinom(a, size = 1, prob = 0.2)
17 plot(1, p, type = 'h', xlab = 'x', ylab = 'p(x)')

```

R code Exa 3.10 Probability distributions for discrete random variables

```
1 #Ex3.10, Page 98
2 #Let Y=>O+ blood type and N=>Other blood types
3 p1<-c('a','Y','b','Y','c','N','d','N','e','N')
4 row.names<-c("Blood donors","O+ blood donors")
5 column.names<-c("","","","","")
6 matrix.names<-c("Blood donation table")
7 p<-array(c(p1),dim=c(2,5,1),dimnames=list(row.names,
      column.names,matrix.names))
8 print(p)
9 #p(1)=P(X=1)
10 c<-0
11 for(i in 1:5) {
12   d<-p[2,i,1]
13   if(d=="Y"){
14     c<-c+1
15   }
16 }
17 d1<-c/5
18 print(paste("p(1)=",d1))
19
20 #p(2)=P(X=2)
21 g<-0
22 e<-0
23 for(i in 1:5) {
24   h<-p[2,i,1]
25   if(h=="Y"){
26     g<-g+1
27   }
28   else{
29     e<-e+1
30   }
31 }
```

```

32 h<-g/5
33 f<-e/4
34 d2<-h*f
35 print(paste("p(2)=",d2))
36
37 #p(3)=P(X=3)
38 d3<-(e/5)*((e-1)/4)*(c/3)
39 print(paste("p(3)=",d3))
40
41 #p(4)=P(X=4)
42 d4<-(e/5)*((e-1)/4)*((e-2)/3)
43 print(paste("p(4)=",d4))
44
45 p2<-c(1,d1,2,d2,3,d3,4,d4)
46 row.names<-c("y","p(y)")
47 column.names<-c("","","","")
48 matrix.names<-c("PMF")
49 q<-array(c(p2),dim=c(2,4,1),dimnames=list(row.names,
        column.names,matrix.names))
50 print(q)
51
52 py<- 1:4
53 y<- c(0.4,0.3,0.2,0.1)
54 plot(py,y,type="h",main="Line graph for pmf",xlab="y
    ",ylab="p(y)")

```

R code Exa 3.13 Cumulative distribution function

```

1 #Ex3.13, Page 101
2
3 y<-c(1,2,4,8,16)
4 p<-c(0.05,0.1,0.35,0.4,0.1)
5
6 #F(1)=P(Y<=1)=p(1)
7 print(paste("F(1)=",cumsum(p[1])))

```

```

8
9 #F(2)=P(Y<=2)=p(2)
10 print(paste("F(2)=",max(cumsum(p[1:2]))))
11
12 #F(4)=P(Y<=4)=p(4)
13 print(paste("F(4)=",max(cumsum(p[1:3]))))
14
15 #F(8)=P(Y<=8)=p(8)
16 print(paste("F(8)=",max(cumsum(p[1:4]))))
17
18 #F(16)=P(X<=16)=p(16)
19 print(paste("F(16)=",max(cumsum(p[1:5]))))

```

R code Exa 3.15 Cumulative distribution function of discrete random variable

```

1 #Ex3.15, Page 104
2 a<-c(0,0.58,1,0.72,2,0.76,3,0.81,4,0.88,5,0.94)
3 row.names<-c("X","F(X)")
4 column.names<-c("","","","","","")
5 matrix.names<-c("Number of days of sick leave taken
   by a randomly selected employee")
6 b<-array(c(a),dim=c(2,6,1),dimnames=list(row.names,
   column.names,matrix.names))
7 print(b)
8
9 #P(2<=X<=5)=P(X=2,3,4,5)
10 p<-b[2,6,1]-b[2,2,1]
11 print(paste("P(2<=X<=5)=",p))
12
13 #P(X=3)
14 q<-b[2,4,1]-b[2,3,1]
15 print(paste("P(X=3)=",q))

```

R code Exa 3.16 Expected value of X

```
1 #Ex3.16, Page 107
2
3 x<-1:7
4 w<-c(0.01,0.03,0.13,0.25,0.39,0.17,0.02)
5
6 #Mean, m=sum of(xp(x))
7 m<-weighted.mean(x,w)
8 print(paste("Population mean =",m))
```

R code Exa 3.17 Expected value of X

```
1 #Ex3.17, Page 107
2
3 x<-0:10
4 w<-c
   (0.002,0.001,0.002,0.005,0.02,0.04,0.18,0.37,0.25,0.12,0.01)
5
6 #Mean, m= sum of(xp(x))
7 m<-weighted.mean(x,w)
8 print(paste("Population mean, E(X) =",m))
```

R code Exa 3.23 Expected value of a function

```
1 #Ex3.23, Page 110
2
3 comp<-c(0,0.1,1,0.2,2,0.3,3,0.4)
```

```

4 row.names<-c("X", "p(X)")
5 column.names<-c("", "", "", "")
6 matrix.names<-c("X->Number of computers sold")
7 p<-array(c(comp), dim=c(2,4,1), dimnames=list(row.
    names, column.names, matrix.names))
8 print(p)
9
10 x<-0:3
11 w<-c(0.1,0.2,0.3,0.4)
12 #h(x)<-profit associated with selling x units<-
    revenue-cost<-800x-900
13 print(paste("Expected profit:$", weighted.mean(800*x
    -900, w)))

```

R code Exa 3.24 The variance of X

```

1 #Ex3.24, Page 111
2 #Answers may slightly vary due to rounding off of
    values
3
4 w<-c(0.3,0.25,0.15,0.05,0.1,0.15)
5 x<-c(1,2,3,4,5,6)
6
7 #Mean, m= sum of(xp(x))
8 m<-weighted.mean(x, w)
9 print(paste("Mean =", m))
10
11 #Variance, v= sum of(x-m)p(x)
12 v<-weighted.mean((x-m)^2, w)
13 print(paste("Variance =", v))
14
15 #Standard deviation, sd= square root(variance)
16 print(paste("Standard deviation=", sqrt(v)))

```

R code Exa 3.25 The variance of X

```
1 #Ex3.25, Page 112
2
3 #Alternative formula for variance,  $v = E(X^2) - (m^2)$ 
4 x<-1:6
5 w<-c(0.3,0.25,0.15,0.05,0.1,0.15)
6
7 # $E(X^2) = \text{sum of } (X^2)p(X)$ 
8 e<-weighted.mean(x^2,w)
9 print(paste("E(X^2)=" ,e))
10
11 #Variance
12 mean<-weighted.mean(x,w)
13 print(paste("Variance through alternative formula ="
  ,e-(mean^2)))
```

R code Exa 3.26 Rules of variance

```
1 #Ex3.26, Page 113
2
3 x<-0:3
4 w<-c(0.1,0.2,0.3,0.4)
5
6 #Mean,  $m = \text{sum of } (xp(x))$ 
7 m<-weighted.mean(x,w)
8 print(paste("Mean, E(X)=" ,m))
9
10 #To find  $E(X^2)$ 
11 e<-weighted.mean(x^2,w)
12 print(paste("E(X^2)=" ,e))
13
```

```

14 #To find V(X)
15 v<-e-(m^2)
16 print(paste("V(X)=",v))
17
18 #Profit function , h(X)=800X-900
19 #Variance
20 print(paste("Variance=", (800^2)*v))
21
22 #Standard deviation
23 print(paste("Standard deviation =",sqrt((800^2)*v)))

```

R code Exa 3.29 Binomial experiment

```

1 #Ex3.29 , Page 115
2 #Number of licensed restaurants
3 n<-50
4 #Number of restaurants having at least one serious
  health code violation
5 f<-15
6 #Number of restaurants having no serious health code
  violations
7 s<-35
8 #P(s on first trial)
9 p1<-s/n;
10 print(paste("P(s on first trial)=",p1))
11
12 #P(s on second trial)=P(ss)+P(fs)
13 p2<-(s/n)*((s-1)/(n-1))+(f/n)*(s/(n-1))
14 print(paste("P(s on second trial)=",p2))
15 print(paste("Similarly , P(s on ith trial) = 0.7
  for i=3,4,5"))
16
17 #P(s on fifth trial|ssss)
18 p3<-(s-4)/(n-4)
19 print(paste("P(s on fifth trial|ssss)=",p3))

```



```

20
21 #P(s on fifth trial|ffff)
22 p4<-s/(n-4)
23 print(paste("P(s on fifth trial|ffff)=",p4))

```

R code Exa 3.30 Binomial experiment

```

1 #Ex3.30, Page 116
2 #Number of licensed drivers
3 n<-500000
4 #Number of licensed drivers who are insured
5 s<-400000
6 #P(s on second trial after s on first trial)
7 p1<-(s-1)/(n-1)
8 print(paste("P(s on 2|s on 1)=",p1))
9
10 #P(s on tenth trial after s on ninth trial)
11 p2<-(s-9)/(n-9)
12 print(paste("P(s on 10|s on 9)=",p2))
13
14 print(paste("The experiment is binomial with n=10
    and p=0.8"))

```

R code Exa 3.31 Binomial random variable and distribution

```

1 #Ex3.31, Page 118
2 #Answers may slightly vary due to rounding off of
  values
3
4 #n=6, p=0.5
5
6 #P(X=3)
7 print(paste("P(X=3)=",dbinom(3,size=6,prob=0.5)))

```

```

8
9 #Probability that at least 3 three prefer S is
10 print(paste("P(X>=3)=", pbinom(3, size=6, prob=0.5)))
11
12 #Probability that at most one prefers S is
13 print(paste("P(X<=1)=", pbinom(1, size=6, prob=0.5)))

```

R code Exa 3.32.1 Cumulative distribution function

```

1 #Example 3.32.1, Page 118
2 #Binomial distribution: n=15, p=2
3 #The answers may slightly vary due to rounding off
  of values
4
5 #Probability that at most 8 copies fail the test
6 #P(X<=8)
7 a=pbinom(8, size=15, prob=0.2)
8 print(paste("The probability that at most 8 copies
  fail the test is", a))

```

R code Exa 3.32.2 Using binomial tables

```

1 #Example 3.32.2, Page 118
2 #Binomial distribution: n=15, p=0.2
3 #The answers may slightly vary due to rounding off
  of values
4
5 #Probability that exactly 8 copies fail
6 #P(X=8)=P(X<=8)-P(X<=7)
7 a<-0.999
8 b=pbinom(7, size=15, prob=0.2)
9 print(paste("The probability that exactly 8 copies
  fail the test", a-b))

```

R code Exa 3.32.3 Using binomial tables

```
1 #Example 3.32.3, Page 119
2 #Binomial distribution: n=15, p=0.2
3 #The answers may slightly vary due to rounding off
  of values
4
5 #Probability that at least 8 copies fail
6 #P(X>=8)=1-P(X<=7)
7 b<-0.996
8 c=1-b
9 print(paste("The probability that at least 8 copies
  fail", c))
```

R code Exa 3.32.4 Using binomial tables

```
1 #Example 3.32.4, Page 119
2 #Binomial distribution: n=15, p=0.2
3 #The answers may slightly vary due to rounding off
  of values
4
5 #Probability that between 4 to 7 copies fail
6 #P(4<=X<=7)=P(X=4)+P(X=5)+P(X=6)+P(X=7)
7 d=pbinom(7,15,0.2)-pbinom(3,15,0.2)
8 print(paste("The probability that between 4 to 7
  copies fail is", d))
```

R code Exa 3.33 Using binomial tables

```

1 #Ex3.33, Page 119
2 #Answers may vary slightly due to rounding off of
  values
3
4 #n=20
5 #P(X>=5) when p=0.10
6 p1<-1-pbinom(4,size=20,prob=0.1)
7 print(paste("P(X>=5 when p=.1)=",p1))
8
9 #P(X<=4) when p=0.2
10 p2<-pbinom(4,size=20,prob=0.2)
11 print(paste("P(X<=4 when p=.2)=",p2))

```

R code Exa 3.34 Mean and variance of X

```

1 #Ex3.34, Page 120
2 n<-10
3 p<-0.75
4
5 #Mean, E(x)=np
6 e<-n*p
7 print(paste("Mean=",e))
8
9 #Variance, V(X)=npq
10 v<-n*p*(1-p)
11 print(paste("Variance=",v))
12
13 #Standard deviation=sqrt(V(X))
14 s<-sqrt(v)
15 print(paste("Standard deviation=",s))
16
17 #P(X= 7 or 8)=P(X=7)+P(X=8)
18 prob1=dbinom(0:7,size=10,prob=0.75)
19 prob2=dbinom(0:8,size=10,prob=0.75)
20 c<-max(prob1)+max(prob2)

```

```
21 print(paste("P(X= 7 or 8)=",c))
```

R code Exa 3.35 Hypergeometric distribution

```
1 #Ex3.35, Page 123
2 #Answers may slightly vary due to rounding off of
  values
3
4 N<-20
5 n<-5
6 M<-12
7
8 #P(X=2)=h(2,5,12,20)
9 x<-2
10 h<-(choose(M,x)*choose(N-M,n-x))/choose(N,n)
11 print(paste("h(2;5,12,20)=",h))
```

R code Exa 3.36 Hypergeometric distribution

```
1 #Ex3.36, Page 124
2 #Answers may slightly vary due to rounding off of
  values
3
4 N<-25
5 n<-10
6 M<-5
7
8 #P(X=2)=h(2,10,5,25)
9 x<-2
10 h1<-(choose(M,X)*choose(N-M,n-x))/choose(N,n)
11 print(paste("P(X=2)=",h1))
12
13 #P(X<=2)=P(X=0)+P(X=1)+P(X=2)
```

```

14 s<-0
15 for(i in 0:2){
16   h2<-(choose(M,i)*choose(N-M,n-i))/choose(N,n)
17   s<-s+h2
18 }
19 print(paste("P(X<=2)=" , s))

```

R code Exa 3.37 Mean and variance of hypergeometric distribution

```

1 #Ex3.37 , Page 125
2
3 n<-10
4 M<-5
5 N<-25
6 p<-M/N
7
8 #E(X)=np
9 e<-n*p
10 print(paste("Mean=" , e))
11
12 #V(X)=npq
13 v<-n*p*(1-p)
14 print(paste("Variance without replacement=" , v))
15
16 #Estimate of N when M=5, n=10 and x=16
17 M<-100
18 n<-40
19 x<-16
20 ne<-(M*n)/x
21 print(paste("Estimate is" , ne))

```

R code Exa 3.38 Negative binomial distribution

```

1 #Ex3.38, Page 126
2 #Answers may slightly vary due to rounding off of
  values
3
4 #P(X=10)=nb(10;5,0.2)
5 prob1=dnbinom(0:10,size=5,prob=0.2)
6 print(paste("P(X=10)=",max(prob1)))
7
8 #P(X<=10)
9 s<-pnbinom(0:10,size=5,prob=0.2)
10 print(paste("P(X<=10)=",max(s)))

```

R code Exa 3.39 Poisson probability distribution

```

1 #Ex3.39, Page 129
2 #Answers may slightly vary due to rounding off of
  values
3
4 u<-4.5
5 #Poisson distribution
6 #P(X=5)
7 p1<-dpois(5,lambda=u)
8 print(paste("P(X=5)=",p1))
9
10 #P(X<=5)
11 p2<-ppois(5,lambda=u)
12 print(paste("P(X<=5)=",p2))

```

R code Exa 3.40 Poisson distribution as a limit

```

1 #Ex3.40, Page 129
2 #Answers may vary slightly due to rounding off of
  values

```

```

3
4 n<-400
5 p<-0.005
6
7 #P(X=1):b(x;n,p)->p(x;u)
8 u<-n*p
9 p1<-dpois(1,lambda=u)
10 print(paste("P(X=1)=",p1))
11
12 #P(X<=3)
13 p2<-ppois(3,lambda=u)
14 print(paste("P(X<=3)=",p2))

```

R code Exa 3.41 Mean and variance in Poisson distribution

```

1 #Ex3.41, Page 130
2 #Answer may slightly vary due to rounding off of
  value
3
4 u<-4.5
5 #Standard deviation
6 sd<-sqrt(u)
7 print(paste("Standard deviation=",sd))

```

R code Exa 3.42 Poisson process

```

1 #Ex3.42, Page 131
2 #Answer may slightly vary due to rounding off of
  value
3
4 alpha<-6
5 t<-0.5
6 l<-alpha*t

```



```
7
8 #P(X>=1)=1-P(X=0)
9 p<-1-dpois(0,lambda=1)
10 print(paste("P(X>=1)=",p))
```

Chapter 4

Continuous random variables and probability distributions

R code Exa 4.4 Probability density function

```
1 #Example 4.4 , Page 139*
2 #n=10,M=5,N=25
3
4 #f(x)>=0
5 #Area under density curve
6 a<-(1/360)*360
7
8 #P(90<=X<=180)
9 f1<-function(x) {(x^0)/360}
10 p1<-integrate(f1,lower=90,upper=180)
11 print(paste("Probability that angle is between 90
    and 180 is",p1$value))
12
13 #P(0<=X<=90)+P(270<=X<360)
14 f2<-function(x) {(x^0)/360}
15 p2<-integrate(f2,lower=0,upper=90)
16 f3<-function(x) {(x^0)/360}
17 p3<-integrate(f3,lower=270,upper=360)
18 p4<-p2$value+p3$value
```

```
19 print(paste("P(angle of occurrence is within 90
    degrees of reference line)",p4))
```

R code Exa 4.5 Uniform distribution

```
1 #Example 4.5, Page 141
2 #The answers may slightly vary due to rounding off
  of values
3
4 #Taking upper limit as 1000 instead of infinity
5 integrand<-function(x) {.15*exp(-.15*(x-.5))}
6 #f(x)>=0
7 #To show that integral from negative infinity to
  infinity is 1
8 c<-integrate(integrand,lower=0.5,upper=Inf)
9 #To access the list of values of integrate by name
10 print(paste("Value of integral is",c$value))
11
12 #To find P(X<5)
13 d<-integrate(integrand,lower=0.5,upper=5)
14 print(paste("Thus, probability that headway time is
    atmost 5 sec:",d$value))
```

R code Exa 4.7 Computing probabilities

```
1 #Ex4.7, Page 145*
2
3 #if x<0,          F(X)=0
4 #if 0<=x<=2,      F(X)=(x/8)+(3/16)*(x^2)
5 #if x>2,          F(X)=1
6
7 load<-function(x){
8   f<-(x/8)+(3/16)*(x^2)
```

```

9   return(f)
10 }
11
12 #P(1<=X<=1.5)=P(X=1.5)-P(X=1)
13 p1<-load(1.5)-load(1)
14 print(paste("P(1<=X<=1.5)=",p1))
15
16 #P(X>1)=1-P(X<=1)
17 p2<-1-load(1)
18 print(paste("P(load exceeds 1)=",p2))

```

R code Exa 4.10 Expected values

```

1 #Ex4.10 , Page 148
2
3 #if 0<=x<=1      f(x)= (3/2)(1-x^2)
4 #otherwise       f(x)= 0
5
6 f1<-function(x) {(3/2)*x*(1-x^2)}
7 p1<-integrate(f1,lower=0,upper=1)
8 print(paste("E(X)=",p1$value))

```

R code Exa 4.11 Expected value of a function

```

1 #Ex4.11 , Page 149
2 #if 0<=x<=1      f(x)=1
3 #otherwise       f(x)=0
4
5 #h(X)=max(1-X,X)
6 #if 0<=X<1/2     h(X)=1-X
7 #if 1/2<=X<=1    h(X)=X
8
9 #f<-function(x) {max(x,1-x)*f(x)}

```

```

10 #integrate(f, lower=0, upper=1)
11
12 f1<-function(x) {(1-x)*1}
13 p1<-integrate(f1, lower=0, upper=1/2)
14
15 f2<-function(x) {x*1}
16 p2<-integrate(f2, lower=0.5, upper=1)
17
18 p3<-p1$value+p2$value
19 print(paste("E[h(X)]=", p3))

```

R code Exa 4.12 Variance and standard deviation

```

1 #Ex4.12, Page 150
2
3 #E(X)
4 e<-3/8
5
6 #E(X^2)
7 e2<-function(x) {(3/2)*(x^2)*(1-(x^2))}
8 p1<-integrate(e2, lower=0, upper=1)
9 print(paste("E(X^2)=", p1$value))
10
11 #V(X)
12 v<-(p1$value)-(e^2)
13 print(paste("V(X)=", v))
14
15 #Standard deviation
16 sd<-sqrt(v)
17 print(paste("Standard deviation =", sd))

```

R code Exa 4.13.a Standard normal distribution

```

1 #Example 4.13.a, Page 154
2 #The answers may slightly vary due to rounding off
  of values
3
4 #P(Z<=1.25)
5 a<-pnorm(1.25,0,1)
6 print(paste("The standard normal probability is",a))
7
8 #To plot the standard normal distribution bell curve
9 curve(dnorm, xlim = c(-3, 3), ylim = c(0, 0.4), xlab
    = "z", ylab="f(z)",lwd=2, col="black")
10 z = 1.25
11 pnorm(z)
12 x = c(-3, seq(-3, z, by=.001), z)
13 #plot(x)
14 y = c(0, dnorm(seq(-3, z, by=.001)), 0)
15 #plot(y)
16 polygon(x, y, col="red")

```

R code Exa 4.13.b Standard normal distribution

```

1 #Example 4.13.b, Page 154
2 #The answers may slightly vary due to rounding off
  of values
3
4 #P(Z>1.25)
5 #Variable to store result(1-P(Z<=1.25))
6 a=1-pnorm(1.25,0,1)
7 print(paste("The standard normal probability is", a)
  )
8
9
10 #To plot the standard normal distribution bell curve
11 curve(dnorm, xlim = c(-3, 3), ylim = c(0, 0.4), xlab
    = "z", ylab="f(z)",lwd=2, col="black")

```

```

12 z = 1.25
13 pnorm(z)
14 x = c(z, seq(z, 3, by=.001), 3)
15 #plot(x)
16 y = c(0, dnorm(seq(z, 3, by=.001)), 0)
17 #plot(y)
18 polygon(x, y, col="red")

```

R code Exa 4.13.c Standard normal distribution

```

1 #Example 4.13.C, Page 154
2 #The answers may slightly vary due to rounding off
  of values
3
4 #P(Z<=-1.25)=P(Z>=1.25)
5 #Variable to store result(1-P(Z<=1.25))
6 a=1-pnorm(1.25,0,1)
7 print(paste("The standard normal probability is", a)
  )
8
9 #To plot the standard normal distribution bell curve
10 x=seq(-3,3,length=200)
11 y=dnorm(x)
12 plot(x,y,type="l", lwd=2, col="black")
13 x=seq(-3,-1.25,length=200)
14 y=dnorm(x)
15 polygon(c(-3,x,-1.25),c(0,y,0),col="red")
16
17 #To plot the standard normal distribution bell curve
18 curve(dnorm, xlim = c(-3, 3), ylim = c(0, 0.4), xlab
  = "z", ylab="f(z)",lwd=2, col="black")
19 z = -1.25
20 pnorm(z)
21 x = c(-3, seq(-3, z, by=.001), z)
22 #plot(x)

```

```

23 y = c(0, dnorm(seq(-3, z, by=.001)), 0)
24 #plot(y)
25 polygon(x, y, col="red")

```

R code Exa 4.13.d Standard normal distribution

```

1 #Example 4.13.d, Page 154
2 #The answers may slightly vary due to rounding off
  of values
3
4 #P(-0.38<=Z<=1.25)
5 #Variable to store result(1-P(Z<=1.25))
6 a<-pnorm(1.25,0,1)-pnorm(-0.38,0,1)
7 print(paste("The standard normal probability is", a)
  )
8
9 #To plot the standard normal distribution bell curve
10 curve(dnorm, xlim = c(-3, 3), ylim = c(0, 0.4), xlab
  = "z", ylab="f(z)",lwd=2, col="black")
11 z1 = 1.25
12 z2 = -0.38
13 pnorm(z)
14 x = c(z2, seq(z2, z1, by=.001), z1)
15 #plot(x)
16 y = c(0, dnorm(seq(z2, z1, by=.001)), 0)
17 #plot(y)
18 polygon(x, y, col="red")

```

R code Exa 4.14 Percentiles of standard normal distribution

```

1 #Ex4.14, Page 155
2 #Answers may slightly vary due to rounding off of
  values

```



```

3
4 print(paste("99th percentile:", qnorm(0.99)))
5 print(paste("1st percentile:", qnorm(0.01)))
6
7 #Blue=> 99th percentile
8 #Red=> 1st percentile
9 plot.new()
10 title(main="Blue: 99th percentile
11 Red: 1st percentile")
12 curve(dnorm, xlim = c(-3, 3), ylim = c(0, 0.5), xlab
      = "z", ylab="f(z)")
13 prob1 = 0.99
14 x1= c(-3, seq(-3, qnorm(prob1), by=.001), qnorm(
      prob1))
15 y1= c(0, dnorm(seq(-3, qnorm(prob1), by=.001)), 0)
16 polygon(x1, y1, col="blue")
17
18 par(new=TRUE)
19
20 prob2 = 0.01
21 x2= c(-3, seq(-3, qnorm(prob2), by=.0001), qnorm(
      prob2))
22 y2= c(0, dnorm(seq(-3, qnorm(prob2), by=.0001)), 0)
23 polygon(x2, y2, col="red")

```

R code Exa 4.15 z critical values

```

1 #Ex4.15, Page 156
2 #Answers may slightly vary due to rounding off of
  values
3
4 #z(.05)
5 x<-0.05
6 z<-100*(1-x)
7

```

```

8 plot.new()
9 title(main="Red: z(0.05)
10 Blue: -z(0.05)")
11
12 #z(.05)
13 print(paste(z,"th percentile:",qnorm(z/100)))
14 curve(dnorm, xlim = c(-3, 3), ylim = c(0, 0.5), xlab
      = "z", ylab="f(z)")
15 prob = z/100
16 x = c(-3, seq(-3, qnorm(prob), by=.001), qnorm(prob)
      )
17 y = c(0, dnorm(seq(-3, qnorm(prob), by=.001)), 0)
18 polygon(x, y, col="red")
19
20 #-z(.05)
21 print(paste("-z(.05)",-qnorm(z/100)))
22 x = c(-3, seq(-3, -qnorm(prob), by=.001), -qnorm(
      prob))
23 y = c(0, dnorm(seq(-3, -qnorm(prob), by=.001)), 0)
24 polygon(x, y, col="blue")

```

R code Exa 4.16 Nonstandard normal distributions

```

1 #Ex4.16, Page 158
2 #Answers may slightly vary due to rounding off of
  values
3
4 #P(1.00<=X<=1.75)
5 #Mean=>1.25, Standard deviation=>0.46
6
7 title(main="P(1.00<=X<=1.75)")
8 a<-pnorm(1.00,mean=1.25,sd=0.46)
9 b<-pnorm(1.75,mean=1.25,sd=0.46)
10 c<-b-a
11 print(paste("P(1.00<=X<=1.75)=",c))

```

```

12
13 #To plot the standard normal distribution bell curve
14 curve(dnorm, xlim = c(-3, 3), ylim = c(0, 0.4), xlab
      = "z", ylab="f(z)",lwd=2, col="black")
15 z<-1.75
16 # Cumulative density function of the standard normal
      distribution
17 pnorm(z)
18 x<-c(1.00, seq(1.00, z, by=.001), z)
19 #plot(x)
20 y<-c(0, dnorm(seq(1.00, z, by=.001)), 0)
21 #plot(y)
22 polygon(x, y, col="red")
23
24 #P(X>2)
25 d<-1-pnorm(2,mean=1.25,sd=0.46)
26 print(paste("P(X>2)=",d))

```

R code Exa 4.17 Nonstandard normal distributions

```

1 #Ex4.17, Page 158
2 #Answers may slightly vary due to rounding off of
      values
3
4 #P(-1.00<=Z<=1.00)
5 a<-pnorm(1.00,0,1)-pnorm(-1.00,0,1)
6 print(paste("P(X is within 1 standard deviation of
      its mean):",a))
7
8 #P(-2.00<=Z<=2.00)
9 b<-pnorm(2.00,0,1)-pnorm(-2.00,0,1)
10 print(paste("P(X is within 2 standard deviation of
      its mean):",b))
11
12 #P(-3.00<=Z<=3.00)

```

```

13 c<-pnorm(3.00,0,1)-pnorm(-3.00,0,1)
14 print(paste("P(X is within 3 standard deviation of
    its mean):",c))

```

R code Exa 4.18 Percentiles of arbitrary normal distribution

```

1 #Ex4.18, Page 159
2 #Answers may slightly vary due to rounding off of
  values
3
4 m<-64
5 sd<-0.78
6 #P(X>c)=0.05
7 p<-0.005
8 #P(X<=c)=1-P(X>c)
9 s<-(1-p)*100
10 print(paste("c is the",s,"th percentile of the
    normal distribution with mean=64 and sd=0.78"))
11 q<-qnorm(s/100)
12 print(paste("Therefore",s,"th percentile:",q))
13
14 c<-m+q*sd
15 print(paste("c=",c,"oz"))
16
17 #Plotting 99.5th percentile
18 plot.new()
19 title(main="99.5th percentile",sub="Shaded area
    =>0.995")
20 curve(dnorm, xlim = c(-3, 3), ylim = c(0, 0.5), xlab
    = "z", ylab="f(z)")
21 prob = 0.995
22 x1= c(-3, seq(-3, qnorm(prob), by=.001), qnorm(prob)
    )
23 y1= c(0, dnorm(seq(-3, qnorm(prob), by=.001)), 0)
24 polygon(x1, y1, col="blue")

```

R code Exa 4.20 Approximating binomial distribution

```
1 #Ex4.20 , Page 161
2 #Answers may vary slightly due to rounding off of
  values
3
4 n<-50
5 p<-25/100
6 q<-1-p
7 m<-12.5
8 sd<-3.06
9 x<-n*p
10 y<-sqrt(n*p*q)
11 z<-n*q
12 if(x>=10 && z>=10){
13   #P(X<=10)
14   a<-pnorm(10+0.5, mean=x, sd=y)
15   print(paste("P(X<=10)=", a))
16
17   #P(5<=X<=15)
18   b<-pnorm(15+0.5, mean=x, sd=y)-pnorm(5-0.5, mean=x, sd
     =y)
19   print(paste("P(5<=X<=15)=", b))
20 }
```

R code Exa 4.21 Exponential distribution

```
1 #Ex4.21 , Page 166
2 #Answers may slightly vary due to rounding off of
  values
3
```

```

4 #E(X)=6MPa=1/lambda
5 e<-6
6 lambda<-1/e
7 #Probability that stress range is at most 10MPa
8 p<-pexp(10,lambda)
9 print(paste("P(stress range is at most 10MPa)=",p))
10
11 #Probability that stress range is between 5 and 10
    MPa
12 q<-pexp(10,lambda)-pexp(5,lambda)
13 print(paste("P(stress range is between 5 and 10MPa)=
    ",q))

```

R code Exa 4.22 Exponential distribution

```

1 #Ex4.22, Page 167
2 #Answer may slightly vary due to rounding off of
    value
3
4 alpha<-0.5
5
6 #P(X>2)=1-P(X<=2)
7 x<-2
8 p<-1-pexp(x,alpha)
9 print(paste("P(X>2)=",p))
10
11 t<-1/alpha
12 print(paste("The expected time between successive
    calls is",t," days"))

```

R code Exa 4.23 Gamma distribution

```

1 #Ex4.23, Page 169

```

```

2 #Answers may slightly vary due to rounding off of
  values
3
4 alpha<-2
5
6 #P(3<=X<=5)
7 a<-pgamma(5,shape=alpha)-pgamma(3,shape=alpha)
8 print(paste("P(3<=X<=5)=",a))
9
10 #P(X>4)=1-P(X<=4)
11 b<-1-pgamma(4,shape=alpha)
12 print(paste("P(reaction time is more than 4 sec)=",b
  ))

```

R code Exa 4.24 Gamma distribution

```

1 #Ex4.24, Page 169
2 #Answers may slightly vary due to rounding off of
  values
3
4 #alpha
5 a<-8
6 #beta
7 b<-15
8
9 #E(X)
10 e<-a*b
11 print(paste("E(X)=",e))
12
13 #V(X)
14 v<-a*(b^2)
15 print(paste("V(X)=",v))
16
17 #SD
18 sd<-sqrt(v)

```

```

19 print(paste("Standard deviation =",sd))
20
21 #P(60<=X<=120)=P(X<=120)-P(X<=60)
22 p1<-pgamma(120/b,shape=a)-pgamma(60/b,a)
23 print(paste("P(mouse survives between 60 and 120
    weeks)=" ,p1))
24
25 #P(X>=30)=1-P(X<30)
26 p2<-1-pgamma(30/b,a)
27 print(paste("P(mouse survives at least 30 weeks)=" ,
    p2))

```

R code Exa 4.25 Weibull distribution

```

1 #Ex4.25, Page 173
2 #Answers may slightly vary due to rounding off of
  values
3
4 alpha<-2
5 beta<-10
6
7 #P(X<=10)
8 p1<-pweibull(10,alpha,beta)
9 print(paste("P(X<=10)=" ,p1))
10
11 #P(X<=25)
12 p2<-pweibull(25,alpha,beta)
13 print(paste("P(X<=25)=" ,p2))
14
15 #To find c value which separates 5% of all engines
    having largest NOx emissions from remaining 95%
16 per<-5/100
17 c<-sqrt((-1)*(10^2)*log(per))
18 print(paste(c,"is the 95th percentile of the
    emission distribution"))

```

R code Exa 4.26 Weibull distribution

```
1 #Ex4.26 , Page 173
2 #Answers may slightly vary due to rounding off of
  values
3
4 #alpha
5 a<-1.3
6 #beta
7 b<-0.8
8 #gamma
9 g<-4
10
11 #P(5<=X<=6)
12 p1<-pweibull(6-g,a,b)-pweibull(5-g,a,b)
13 print(paste("P(air void volume of a specimen is
  between 5% and 6%)=",p1))
```

R code Exa 4.27 Lognormal distribution

```
1 #Ex4.27 , Page 175
2 #Answers may slightly vary due to rounding off of
  values
3
4 m<-0.353
5 s<-0.754
6
7 e<-exp(m+(s^2)/2)
8 print(paste("E(X)=",e))
9
10 v<-exp(2*m+(s^2))*(exp(s^2)-1)
```

```

11 print(paste("V(X)=" ,v))
12
13 #P(1<=X<=2)
14 l1<-log(1,base=exp(1))
15 l2<-log(2,base=exp(1))
16 p<-pnorm(l2,m,s)-pnorm(l1,m,s)
17 print(paste("P(maximum pit depth is between 1 and 2
      mm)=" ,p))
18
19 #To find c such that only 1% of all specimens have a
      maximum pit depth excceding c
20 per<-1/100
21 c<-exp(1)^(qnorm(1-per)*s+m)
22 print(paste(c," is the 99th percentile of the maximum
      pit depth distribution"))

```

R code Exa 4.28 Beta distribution

```

1 #Ex4.28, Page 177
2
3 #Optimistic time
4 A<-2
5
6 #Pessimistic time
7 B<-5
8
9 #alpha
10 a<-2
11 #beta
12 b<-3
13
14 #E(X)
15 e<-A+(B-A)*(a/(a+b))
16 print(paste("Mean=" ,e))
17

```

```

18 c<-(1/(B-A))*((gamma(a+b))/(gamma(a)*gamma(b)))
19 f1<-function(x) {c*(((x-A)/(B-A))^(a-1))*(((B-x)/(B-
    A))^(b-1))}
20 p<-integrate(f1,lower=2,upper=3)
21 print(paste("P(it take atmost 3 days to lay the
    foundation)=",p$value))

```

R code Exa 4.29 Probability plots

```

1 #Ex4.29 , Page 181
2
3 old.par<-par(mfrow=c(1,2))
4
5 #To plot first sample
6 Percentage1<-c(5, 15, 25, 35, 45)
7 z_percentile1<-c(qnorm(5/100), qnorm(15/100), qnorm
    (25/100), qnorm(35/100), qnorm(45/100))
8 Sample_observation1<-c(-1.91, -1.25, -0.75, -0.53,
    0.2)
9 df1<-data.frame(Percentage1, z_percentile1, Sample_
    observation1)
10 print(df1)
11 plot(z_percentile1,Sample_observation1)
12
13 #To plot second sample
14 Percentage2<-c(55, 65, 75, 85, 95)
15 z_percentile2<-c(qnorm(55/100), qnorm(65/100), qnorm
    (75/100), qnorm(85/100), qnorm(95/100))
16 Sample_observation2<-c(0.35, 0.72, 0.87, 1.40, 1.56)
17 df2<-data.frame(Percentage2, z_percentile2, Sample_
    observation2)
18 print(df2)
19 plot(z_percentile2,Sample_observation2)
20
21 par(old.par)

```

R code Exa 4.30 Probability plots

```
1 #Ex4.30 , Page 182
2
3 old.par<-par(mfrow=c(1,1))
4 n<-20
5 z_percentile<-c(qnorm((1-0.5)/n), qnorm((2-0.5)/n),
6                 qnorm((3-0.5)/n), qnorm((4-0.5)/n), qnorm((5-0.5)/n),
7                 qnorm((6-0.5)/n), qnorm((7-0.5)/n),
8                 qnorm((8-0.5)/n), qnorm((9-0.5)/n),
9                 qnorm((10-0.5)/n), qnorm((11-0.5)/n), qnorm((12-0.5)/n),
10                qnorm((13-0.5)/n), qnorm((14-0.5)/n), qnorm((15-0.5)/n),
11                qnorm((16-0.5)/n), qnorm((17-0.5)/n), qnorm((18-0.5)/n), qnorm((19-0.5)/n), qnorm((20-0.5)/n))
12 Sample_observation<-c(24.46, 25.61, 26.25, 26.42,
13                       26.66, 27.15, 27.31, 27.54, 27.74, 27.94, 27.98,
14                       28.04, 28.28, 28.49, 28.50, 28.87, 29.11, 29.13,
15                       29.50, 30.88)
16 df<-data.frame(z_percentile, Sample_observation)
17 print(df)
18 plot(z_percentile, Sample_observation)
19
20 par(old.par)
```

R code Exa 4.31 Beyond normality

```
1 #Ex4.31 , Page 186
```

```

2
3 old.par<-par(mfrow=c(1,1))
4
5 Percentile<-c(-2.97, -1.82, -1.25, -0.84, -0.51,
               -0.23, 0.05, 0.33, 0.64, 1.10)
6 x<-c(282, 501, 741, 851, 1072, 1122, 1202, 1585,
       1905, 2138)
7 lnx<-c(log(282,base=exp(1)),log(501,base=exp(1)),
         log(741,base=exp(1)), log(851,base=exp(1)), log
         (1072,base=exp(1)),
8         log(1122,base=exp(1)), log(1202,base=exp(1)
         ), log(1585,base=exp(1)), log(1905,base=
         exp(1)), log(2138,base=exp(1)))
9 df<-data.frame(Percentile,x,lnx)
10 print(df)
11 plot(Percentile,lnx)
12
13 par(old.par)

```

Chapter 5

Joint probability distributions and random samples

R code Exa 5.1 Joint probability mass function

```
1 #Ex5.1, Page 194
2
3 m<-matrix(c(0.20,0.05,0.10,0.15,0.20,0.30),nrow=2,
4           ncol=3)
5 rownames(m)<-c("100","250")
6 colnames(m)<-c("0","100","200")
7 print(m)
8
9 #To find P(Y>=100)
10 prob<-m[1,2]+m[1,3]+m[2,2]+m[2,3]
11 print(paste("P(Y>=100)=",prob))
```

R code Exa 5.2 Marginal probability mass function

```
1 #Ex5.2, Page 195
2
```

```

3 m<-matrix(c(0.20,0.05,0.10,0.15,0.20,0.30),nrow=2,
            ncol=3)
4 rownames(m)<-c("100","250")
5 colnames(m)<-c("0","100","200")
6 print(m)
7 cat("\n")
8
9 px_100<-apply(m,1,sum)[1]
10 px_250<-apply(m,1,sum)[2]
11 cat("px(100)=",px_100,"\n")
12 cat("px(250)=",px_250,"\n\n")
13
14 cat("Marginal pmf of X:\n")
15 cat("p(x)=",px_100,"when x=100,250\n")
16 cat("p(x)=0 otherwise\n\n")
17
18 py_0<-apply(m,2,sum)[1]
19 py_100<-apply(m,2,sum)[2]
20 py_200<-apply(m,2,sum)[3]
21 cat("py(0)=",py_0,"\n")
22 cat("py(100)=",py_100,"\n")
23 cat("py(200)=",py_200,"\n\n")
24
25 cat("Marginal pmf of Y:\n")
26 cat("p(y)=",py_0,"when y=0,100\n")
27 cat("p(y)=",py_200,"when y=200\n")
28 cat("p(y)=0 otherwise\n\n")
29
30 #To find P(Y>=100)
31 prob<-py_100+py_200
32 cat("P(Y>=100)=",prob)

```

R code Exa 5.3 Joint probability density function

1 #Ex5.3, Page 196

```

2 #Answers may ary slightly due to rounding off of
  values
3
4 f<-function(x,y) {(6/5)*(x+y^2)}
5
6 #To check whether pdf is legitimate
7 check_val<-integrate(function(y) {sapply(y,function(
  y) {integrate(function(x) f(x,y),0,1)$value}
  },0,1)
8 if(check_val$value==1) cat("PDF is legitimate\n\n")
9
10 #P(0<=X<=1/4,0<=Y<=1/4)
11 low1<-0
12 high1<-1/4
13 low2<-0
14 high2<-1/4
15 integral<-integrate(function(y) {sapply(y,function(y
  ) {integrate(function(x) f(x,y),low1,high1)$value
  })},low2,high2)
16 print(paste("P(0<=X<=1/4,0<=Y<=1/4)=",integral$value
  ))

```

R code Exa 5.4 Marginal probability density function

```

1 #Ex5.4 , Page 197
2
3 f<-function(x,y) {(6/5)*(x+y^2)}
4
5 #Finding P(1/4<=Y<=3/4) using marginal pdf of Y
6 xlow<-0
7 xhigh<-1
8 ylow<-1/4
9 yhigh<-3/4
10 prob<-integrate(function(y) {sapply(y,function(y) {
  integrate(function(x) f(x,y),xlow,xhigh)$value}

```



```

    }, ylow, yhigh)
11 print(paste("P(1/4<=Y<=3/4)=", prob$value))

```

R code Exa 5.5 Marginal probability density function

```

1 #Ex5.5, Page 198
2
3 #Joint pdf
4 f<-function(x,y) {24*x*y}
5
6 val<-Vectorize(function(x) {sapply(x,function(z) {
7   integrate(function(y) f(x,y),0,1-z)$value})})
8 check_val<-integrate(val,0,1)$value
9 if(check_val==1) print(paste("PDF is legitimate"))
10
11 #Probability that two types of nuts together make up
12   50% of the can
13 #Taking A={(x,y):0<=x<=1,0<=y<=1,x+y<=0.5}
14 xlow<-0
15 xhigh<-0.5
16 integral<-Vectorize(function(x) {sapply(x,function(z)
17   ) {integrate(function(y) f(x,y),0,0.5-z)$value})
18   })
19 prob<-integrate(integral,xlow,xhigh)
20 print(paste("P((X,Y)EA) =", prob$value))

```

R code Exa 5.8 Independent random variables

```

1 #Ex5.8, Page 200
2 #Answers may vary slightly due to rounding off of
3   values
4 #Expected lifetimes are 1000 and 1200 hours

```

```

5 lambda1<-(1/1000)
6 lambda2<-(1/1200)
7
8 #P(X1>=1500)
9 p1<-exp(-(lambda1)*1500)
10
11 #P(X2>=1500)
12 p2<-exp(-(lambda2)*1500)
13
14 #Joint PDF
15 #P(X1>=1500,X2>=1500)
16 p<-p1*p2
17 print(paste("P(both component lifetimes are at least
    1500 hours)=",p))

```

R code Exa 5.9 Multinomial distribution

```

1 #Ex5.9, Page 201
2 #Answer may slightly vary due to rounding off of
  value
3
4 #Respective number and probabilities
5 n<-c(2,5,3)
6 p<-c(0.25,0.5,0.25)
7
8 #Multinomial PMF
9 m<-dmultinom(n,prob=p)
10 print(paste("Multinomial PMF:",m))

```

R code Exa 5.12 Conditional distributions

```

1 #Ex5.12, Page 203

```

```

2 #Answers may slightly vary due to rounding off of
  values
3
4 #Function when X=0.8: (1/34)(24+30(y^2))
5
6 #P(Y<=0.5) given X=0.8
7 f1<-function(y) {(1/34)*(24+30*(y^2))}
8 f<-integrate(f1,lower=0,upper=0.5)
9 print(paste("P(walk-up facility is busy atmost half
  the time)=",f$value))
10
11 #E(Y) given that X=0.8
12 f2<-function(y) {y*(1/34)*(24+30*(y^2))}
13 p<-integrate(f2,lower=0,upper=1)
14 print(paste("Expected proportion of time that walk-
  up facility is busy:",p$value))

```

R code Exa 5.14 Expected values

```

1 #Ex5.14, Page 207
2
3 #Joint pdf
4 f<-function(x,y) {24*x*y}
5
6 almonds<-1
7 cashews<-1.5
8 peanuts<-0.5
9
10 #Total cost of the contents of a can
11 h<-function(x,y) {almonds*x+cashews*y+(1-x-y)*
  peanuts}
12
13 #Expected total cost
14 integral<-Vectorize(function(x) {supply(x,function(z
  ) {integrate(function(y) h(x,y)*f(x,y),0,1-z)$

```

```

    value}}))
15 prob<-integrate(integral,0,1)
16 print(paste("Expected total cost:$",prob$value))

```

R code Exa 5.15 Covariance

```

1 #Ex5.15, Page 208
2
3 x<-c(100,250)
4 y<-c(0,100,200)
5
6 m1<-matrix(c(0.20,0.05,0.10,0.15,0.20,0.30),nrow=2,
7           ncol=3)
8 rownames(m1)<-c("100","250")
9 colnames(m1)<-c("0","100","200")
10 print(m1)
11 cat("\n")
12
13 m2<-matrix(c(0.5,0.5),nrow=1,ncol=2)
14 rownames(m2)<-c("p(x)")
15 colnames(m2)<-c("100","250")
16 print(m2)
17 cat("\n")
18
19 m3<-matrix(c(0.25,0.25,0.5),nrow=1,ncol=3)
20 rownames(m3)<-c("p(y)")
21 colnames(m3)<-c("0","100","200")
22 print(m3)
23 cat("\n")
24
25 x1<-c(0.5,0.5)
26 j<-1
27 mu_x<-0
28 for(i in x){

```

```

29   if(j<=length(x1)){
30       mu_x<-mu_x+i*x1[j]
31       j=j+1
32   }
33 }
34 cat("mu_X=",mu_x,"\n\n")
35
36 y1<-c(0.25,0.25,0.5)
37 k<-1
38 mu_y<-0
39
40 for(i in y){
41     if(k<=length(y1)){
42         mu_y<-mu_y+i*y1[k]
43         k=k+1
44     }
45 }
46 cat("mu_Y=",mu_y,"\n\n")
47
48 p<-c(0.2,0.1,0.20,0.05,0.15,0.30)
49 sum<-0
50 k<-1
51 for(i in x){
52     for(j in y){
53         if(k<=length(p)){
54             sum=sum+(i-175)*(j-125)*p[k]
55             k=k+1
56         }
57     }
58 }
59 print(paste("Cov(X,Y):",sum))

```

R code Exa 5.18 Correlation

1 [#Ex5.18](#), Page 210

```

2
3 x<-c(-4,4,2,-2)
4 y<-c(1,-1,2,-2)
5 xy<-x*y
6 w<-c(1/4,1/4,1/4,1/4)
7 m<-weighted.mean(xy,w)
8 print(paste("E(XY)=",m))
9
10 #Cov(XY)=E(XY)-ux.uy
11 ux<-0
12 uy<-0
13 cov<-m-(ux*uy)
14 print(paste("Cov(XY)=",cov))
15
16 print(paste("Complete absence of any linear
      relationship"))
17
18 #To scatterplot
19 df<-data.frame(x,y)
20 plot(x,y)

```

R code Exa 5.19 Statistics and their distributions

```

1 #Ex5.19, Page 212
2 #Answers may vary slightly due to rounding off of
  values
3
4 #alpha
5 a<-2
6
7 #beta
8 b<-5
9
10 #Mean
11 u<-b*gamma(1+(1/a))

```

```

12 print(paste("Mean, E(X)=" ,u))
13
14 #Variance
15 v<-(b^2)*{gamma(1+(2/a))-(gamma(1+(1/a)))^2}
16 print(paste(" Variance=" ,v))
17
18 #Standard deviation
19 s<-sqrt(v)
20 print(paste(" Standard deviation=" ,s))
21
22 #Median
23 med<-b*(log(2,base=exp(1)))^(1/a)
24 print(paste(" Median=" ,med))
25
26 q<-qweibull(0.99,shape=a,scale=b)
27 curve(dweibull(x,a,b),from=0,to=q,ylab='density')
28
29 if(u>med){
30     print(paste("Mean exceeds median because of the
        distribution 's positive skew"))
31 }

```

R code Exa 5.24 Standard error of the mean

```

1 #Ex5.24, Page 223
2
3 n<-25
4 u<-28000
5 s<-5000
6
7 #E(Mean)
8 print(paste("E(Mean)=" ,u))
9
10 #E(T0)
11 t<-n*u

```

```

12 print(paste("E(T0)=" ,t))
13
14 #Standard deviation of mean
15 sm<-s/sqrt(n)
16 print(paste("SD of mean:" ,sm))
17
18 #Standard deviation of T0
19 st<-s*sqrt(n)
20 print(paste("SD of T0:" ,st))
21
22 #When sample size increases to n=100
23 print(paste("When n=100:") )
24 n<-100
25 #E(Mean)
26 print(paste("E(Mean)=" ,u))
27
28 #E(T0)
29 t<-n*u
30 print(paste("E(T0)=" ,t))
31
32 #Standard deviation of mean
33 sm<-s/sqrt(n)
34 print(paste("SD of mean:" ,sm))
35
36 #Standard deviation of T0
37 st<-s*sqrt(n)
38 print(paste("SD of T0:" ,st))

```

R code Exa 5.25 Normal population distribution

```

1 #Ex5.25 , Page 224
2 #Answers may slightly vary due to rounding off of
  values
3
4 n<-5

```



```

5
6 #Mean(min)
7 m<-1.5
8
9 #Standard deviation(min)
10 sd<-0.35
11
12 ut<-n*m
13 print(paste("uT0:",ut))
14
15 vt<-n*(sd^2)
16 print(paste("vT0:",vt))
17
18 st<-sqrt(vt)
19 print(paste("sT0:",st))
20
21 #P(6<=T0<=8)
22 p1<-pnorm(8,mean=ut,sd=st)-pnorm(6,mean=ut,sd=st)
23 print(paste("P(6<=T0<=8)=",p1))
24
25 #Probability that the sample average time is atmost
    2 min
26 u<-1.5
27 s<-sd/sqrt(n)
28
29 #P(Mean<=2.0)
30 p2<-pnorm(2,mean=u,sd=s)
31 print(paste("P(Mean<=2)=",p2))

```

R code Exa 5.26 Central limit theorem

```

1 #Ex5.26, Page 225
2 #Answer may slightly vary due to rounding off of
    value
3

```

```

4  n<-50
5  m<-4
6  sd<-1.5
7
8  ux<-m
9  sx<-sd/sqrt(n)
10
11 #P(3.5<=Mean<=3.8)
12 p<-pnorm(3.8,mean=ux,sd=sx)-pnorm(3.5,mean=ux,sd=sx)
13 print(paste("P(3.5<=Mean<=3.8)=",p))

```

R code Exa 5.27 Central limit theorem

```

1  #Ex5.27, Page 226
2  #Answer may slightly vary due to rounding off of
   value
3
4  n<-100
5  m<-3.2
6  sd<-2.4
7
8  ux<-m
9  sx<-sd/sqrt(n)
10
11 #P(Mean>4)=1-P(Mean<=4)
12 p<-1-pnorm(4,mean=ux,sd=sx)
13 print(paste("P(Mean>4)=",p))

```

R code Exa 5.29 Linear combination

```

1  #Ex5.29, Page 231
2  #Answers may vary slightly due to rounding off of
   values

```

```

3
4 a1<-3.00
5 a2<-3.20
6 a3<-3.40
7
8 #Mean
9 u1<-1000
10 u2<-500
11 u3<-300
12
13 #Standard deviation
14 s1<-100
15 s2<-80
16 s3<-50
17
18 #Y=3.0X1+3.20X2+3.40X3
19 EY<-a1*u1+a2*u2+a3*u3
20 VY<-(a1^2)*(s1^2)+(a2^2)*(s2^2)+(a3^2)*(s3^2)
21 sY<-sqrt(VY)
22 print(paste("E(Y)=$",EY))
23 print(paste("V(Y)=$",VY))
24 print(paste("sY=$",sY))

```

R code Exa 5.30 Difference between two random variables

```

1 #Ex5.30, Page 232
2 #Answers may slightly vary due to rounding off of
  values
3
4 u1<-22
5 u2<-26
6 s1<-1.2
7 s2<-1.5
8
9 #Mean

```

```

10 e<-u1-u2
11 print(paste("E(X1-X2)=" ,e))
12
13 #Variance
14 v<-(s1^2)+(s2^2)
15 print(paste("V(X1-X2)=" ,v))
16
17 #Standard deviation
18 sd<-sqrt(v)
19 print(paste("SD=" ,sd))

```

R code Exa 5.31 Normal random variables

```

1 #Ex5.31 , Page 232
2
3 #Y=3.0X1+3.2X2+3.4X3
4 u<-5620
5 s<-429.46
6
7 #P(Y>4500)
8 p<-1-pnorm(4500 ,mean=u ,sd=s)
9 print(paste("P(revenue exceeds 4500)=" ,p))

```

Chapter 6

Point estimation

R code Exa 6.1 Some general concepts of point estimation

```
1 #Ex6.1 , Page 241
2
3 x<-15
4 n<-25
5 p<-x/n
6 print(paste("Most reasonable estimate is",p))
```

R code Exa 6.2 Some general concepts of point estimation

```
1 #Ex6.2 , Page 241
2 #Answers may vary slightly due to rounding off of
  values
3
4 volt<-c
  (24.46 , 25.61 , 26.25 , 26.42 , 26.66 , 27.15 , 27.31 , 27.54 , 27.74 , 27.94 , 27.98)
5
6 #Estimator1
```

```

7 x1<-mean(volt)
8 print(paste(" Estimator (1) :",x1))
9
10 #Estimator2
11 x2<-median(volt)
12 print(paste(" Estimator (2) :",x2))
13
14 #Estimator3
15 x3<-(min(volt)+max(volt))/2
16 print(paste(" Estimator (3) :",x3))
17
18 #Estimator4
19 volt<-setdiff(volt,c(volt[1],volt[2],volt[19],volt
    [20]))
20 x4<-mean(volt)
21 print(paste(" Estimator (4) :",x4))

```

R code Exa 6.3 Some general concepts of point estimation

```

1 #Ex6.3, Page 242
2 #Answers may vary slightly due to rounding off of
  values
3
4 obs<-c
    (74.33,71.07,73.82,77.42,79.35,82.27,77.75,78.65,77.19,74.69,
5
    77.25,74.84,60.90,60.75,74.09,65.36,67.84,69.97,68.83,75.09,
6
    62.54,67.47,72.00,66.51,68.21,64.46,64.34,64.93,67.33,66.08,
7
    67.31,74.87,69.40,70.83,81.73,82.50,79.87,81.96,79.51,84.12,
8
    80.61,79.89,79.70,78.74,77.28,79.97,75.09,74.38,77.67,83.73,
9
    80.39,76.90)

```

```

10
11 #Mean
12 print(paste("Mean=",mean(obs)))
13
14 #Variance
15 print(paste("Point estimate of the population
    variance:",var(obs)))
16
17 #Alternative formula
18 #To find numerator of s^2
19 s1<-sum(obs)
20 s2<-sum(obs^2)
21 Sxx<-s2-(s1^2)/length(obs)
22
23 #Standard deviation
24 print(paste("Point estimate of the population
    standard deviation:",sd(obs)))
25
26 #Alterative estimator
27 print(paste("Alternative estimator of population
    variance:",Sxx/length(obs)))

```

R code Exa 6.9 Standard error of an estimator

```

1 #Ex6.9, Page 251
2 #Answers may slightly vary due to rounding off of
    values
3
4 n<-20
5 sigma<-1.5
6
7 #Standard error of mean
8 sigx<-sigma/sqrt(n)
9 print(paste("Standard error of mean:",sigx))
10

```

```

11 #Estimate of standard deviation
12 s<-1.462
13
14 sx<-s/sqrt(n)
15 print(paste("Estimated standard error:",sx))

```

R code Exa 6.10 Standard error of an estimator

```

1 #Ex6.10, Page 251
2 #Answers may vary slightly due to rounding off of
  values
3
4 n<-25
5 p<-0.6
6 q<-1-p
7
8 #To find estimated standard error
9 sp1<-sqrt((p*q)/n)
10 print(paste("Estimated standard error:",sp1))
11
12 #pq is largest when
13 p<-q<-0.5
14 sp2<-sqrt((p*q)/n)
15 print(paste("Estimated standard error when pq is
  largest:",sp2))

```

R code Exa 6.11 Standard error of an estimator

```

1 #Ex6.11, Page 252
2 #Answers may slightly vary due to rounding off of
  values
3

```



```

4  t<-c
    (41.53,18.73,2.99,30.34,12.33,117.52,73.02,223.63,4.00,26.78)

5
6  #Mean
7  m<-mean(t)
8  print(paste("Mean:",m))
9
10 #lambda(l)
11 l<-1/m
12 print(paste("Reasonable estimate of lambda:",l))

```

R code Exa 6.13 Method of moments

```

1  #Ex6.13, Page 257
2  #Answers may vary slightly due to rounding off of
    values
3
4  data<-c
    (152,115,109,94,88,137,152,77,160,165,125,40,128,123,136,101,62,1

5  n<-length(data)
6
7  #Mean
8  m<-mean(data)
9  print(paste("Mean:",m))
10
11 #Sum of x^2
12 s<-sum(data^2)/n
13
14 #alpha
15 alpha<-(m^2)/(s-(m^2))
16 print(paste("Alpha:",alpha))
17
18 #beta

```

```
19 beta<-(s-(m^2))/m
20 print(paste("Beta:",beta))
```

R code Exa 6.14 Method of moments

```
1 #Ex6.14, Page 257
2 #Answers may vary slightly due to rounding off of
  values
3
4 goals<-c(0,1,2,3,4,5,6,7,8,9,10)
5 freq<-c(29,71,82,89,65,45,24,7,4,1,3)
6
7 #Mean
8 m<-weighted.mean(goals,freq)
9 print(paste("Mean=",m))
10
11 #To find sum of x^2
12 ms<-weighted.mean(goals^2,freq)
13 print(paste("Sum of x^2/frequency=",ms))
14
15 #To find alpha
16 p_cap<-m/(ms-(m^2))
17 print(paste("Alpha:",p_cap))
18
19 #To find beta
20 r_cap<-(m^2)/(ms-(m^2)-m)
21 print(paste("Beta:",r_cap))
```

R code Exa 6.21 The invariance principle

```
1 #Ex6.21, Page 262
2 #Answers may slightly vary due to rounding off of
  values
```

```
3
4 alpha<-11.9731
5 beta<-77.0153
6 m<-beta*gamma(1+(1/alpha))
7 print(paste("Mean estimate:",m))
```

Chapter 7

Statistical intervals based on a single sample

R code Exa 7.2 Basic properties of confidence intervals

```
1 #Ex7.2, Page 270
2 #Answers may slightly vary due to rounding off of
  values
3
4 mean<-80
5 n<-31
6 sd<-2
7 error<-qt(0.95,df=n-1)*sd/sqrt(n)
8
9 #Left endpoint
10 l<-mean-error
11 print(paste("Left endpoint of interval:",l))
12
13 #Right endpoint
14 r<-mean+error
15 print(paste("Right endpoint of interval:",r))
```

R code Exa 7.3 Other levels of confidence

```
1 #Ex7.3, Page 272
2 #Answers may slightly vary due to rounding off of
  values
3
4 mean<-5.426
5 n<-40
6 sd<-0.1
7
8 #To find alpha for (100(1-alpha)=90)
9 alpha<-solve(100,10)
10
11 error<-qnorm(1-(alpha/2))*(sd/sqrt(n))
12
13 #Left endpoint
14 l<-mean-error
15 print(paste("Left endpoint of interval:",l))
16
17 #Right endpoint
18 r<-mean+error
19 print(paste("Right endpoint of interval:",r))
```

R code Exa 7.4 Confidence level and precision and sample size

```
1 #Ex7.4, Page 273
2 #Answers may slightly vary due to rounding off of
  values
3
4 #For 95% CI
5 z<-qnorm(1-(0.05/2))
6
7 sd<-25
8 w<-10
9
```

```

10 n<-(2*z*sd/w)^2
11 print(paste("Sample size necessary for CI to have
    width 10 is ",n,"and thus approx. 97"))

```

R code Exa 7.6 Large sample CI for mean

```

1 #Ex7.6, Page 277–278
2 #Answers may slightly vary due to rounding off of
  values
3
4 data<-c
    (2948,2996,7197,8338,8500,8759,12710,12925,15767,20000,23247,24863
5 26210,30552,30600,35700,36466,40316,40596,41021,41234,43000,44607,45
6 45027,45442,46963,47978,49518,52000,53334,54208,56062,57000,57365,60
7 60265,60803,62851,64404,72140,74594,79308,79500,80000,80000,84000,11
8 boxplot(data)
9 print(data)
10
11 n<-50
12
13 #To find mean
14 mean<-mean(data)
15 print(paste("Mean:",mean))
16
17 #To find median
18 med<-median(data)
19 print(paste("Median:",med))
20
21 #To find standard deviation
22 s<-sd(data)
23 print(paste("Standard deviation:",s))

```

```

24
25 #To find confidence intervals
26 z<-qnorm(1-(0.05/2))
27 error<-z*(s/sqrt(n))
28
29 #Left endpoint
30 l<-mean-error
31 print(paste("Left endpoint of interval:",l))
32
33 #Right endpoint
34 r<-mean+error
35 print(paste("Right endpoint of interval:",r))

```

R code Exa 7.7 Large sample CI for mean

```

1 #Ex7.7, Page 279
2
3 t1<-320
4 t2<-440
5
6 #To find value of s
7 s<-(t2-t1)/4
8
9 w<-5
10
11 z<-qnorm(1-(0.05/2))
12
13 n<-(z*s/w)^2
14 print(paste("Sample size:",n))
15 print(paste("Integer value of sample size:",ceiling(
    n)))

```

R code Exa 7.8 CI for population proportion

```

1 #Ex7.8, Page 282
2 #Answer may vary slightly compared to textbook value
3
4 n<-48
5 n1<-16
6 p<-n1/n
7
8 alpha<-0.05
9 b<-binom.test(16,48,p,conf.level=0.95)
10 print(b)

```

R code Exa 7.10 Confidence bounds

```

1 #Ex7.10, Page 283
2 #Answer may slightly vary due to rounding off of
  value
3
4 mean<-17.17
5 z<-qnorm(0.95)
6 n<-48
7 sd<-3.28
8
9 #To find lower bound
10 c<-mean-(z*sd/sqrt(n))
11 print(paste("Lower confidence bound for mean:",c))

```

R code Exa 7.11 One sample t confidence interval

```

1 #Ex7.11, Page 288
2 #Answers may slightly vary due to rounding off of
  values

```



```

3  data<-c
    (6807.99,7637.06,6663.28,6165.03,6991.41,6992.23,6981.46,7569.75,
4      7437.88,6872.39,7663.18,6032.28,6906.04,6617.17,6984.12,7093
5      7659.50,7378.61,7295.54,6702.76,7440.17,8053.26,8284.75,7347
6      7422.69,7886.87,6316.67,7713.65,7503.33,7674.99)

7
8  n<-30
9
10 #To find mean
11 mean<-mean(data)
12 print(paste("Mean:",mean))
13
14 #To find standard deviation
15 s<-sd(data)
16 print(paste("Standard deviation:",s))
17
18 #To find confidence intervals
19 t<-qt(1-(0.05/2),df=n-1)
20 print(paste("Critical value,t:",t))
21 error<-t*(s/sqrt(n))
22
23 #Left endpoint
24 l<-mean-error
25 print(paste("Left endpoint of interval:",l))
26
27 #Right endpoint
28 r<-mean+error
29 print(paste("Right endpoint of interval:",r))
30
31 #To plot the normal probability plot
32 qqnorm(data,main="Normal Probability of MOR")

```

R code Exa 7.12 A prediction interval for a single future value

```
1 #Ex7.12, Page 288
2 #Answers may slightly vary due to rounding off of
  values
3
4 data<-c
  (25.2,21.3,22.8,17.0,29.8,21.0,25.5,16.0,20.9,19.5)

5 print(data)
6
7 n<-10
8
9 #To find mean
10 mean<-mean(data)
11 print(paste("Mean:",mean))
12
13 #To find standard deviation
14 s<-sd(data)
15 print(paste("Standard deviation:",s))
16
17 #To find confidence intervals
18 t<-qt(1-(0.05/2),df=n-1)
19 print(paste("Critical value,t:",t))
20 error<-t*(s/sqrt(n))
21
22 #Left endpoint
23 l<-mean-error
24 print(paste("Left endpoint of interval:",l))
25
26 #Right endpoint
27 r<-mean+error
28 print(paste("Right endpoint of interval:",r))
```

R code Exa 7.13 Prediction interval

```
1 #Ex7.13, Page 290
2 #Answers may slightly vary due to rounding off of
  values
3
4 mean<-21.9
5 s<-4.134
6 n<-10
7
8 #To find critical value
9 t<-qt(1-(0.05/2),df=n-1)
10 print(paste("Critical value ,t:",t))
11
12 error<-t*s*sqrt(1+(1/n))
13
14 #Left endpoint
15 l<-mean-error
16 print(paste("Left endpoint of prediction interval:",
  l))
17
18 #Right endpoint
19 r<-mean+error
20 print(paste("Right endpoint of prediction interval:"
  ,r))
```

R code Exa 7.14 Tolerance intervals

```
1 #Ex7.14, Page 291
2 #Answers may vary slightly due to rounding off of
  values
3
```

```

4  data<-c
    (10490,16620,17300,15480,12970,17260,13400,13900,
5      13630,13260,14370,11700,15470,17840,14070,14760)

6  print(data)
7
8  n<-16
9
10 #To find mean
11 mean<-mean(data)
12 print(paste("Mean:",mean))
13
14 #To find standard deviation
15 s<-sd(data)
16 print(paste("Standard deviation:",s))
17
18 #To find tolerance intervals
19 t1<-2.903
20 error1<-t1*s
21 #Left endpoint
22 l1<-mean-error1
23 print(paste("Left endpoint of tolerance interval:",
    l1))
24 #Right endpoint
25 r1<-mean+error1
26 print(paste("Right endpoint of tolerance interval:",
    r1))
27
28 #To find confidence intervals
29 t2<-qt(1-(0.05/2),df=n-1)
30 error2<-t2*(s/sqrt(n))
31 #Left endpoint
32 l2<-mean-error2
33 print(paste("Left endpoint of confidence interval:",
    l2))
34 #Right endpoint
35 r2<-mean+error2
36 print(paste("Right endpoint of confidence interval:")

```

```

        ,r2))
37
38 #To find prediction intervals
39 t3<-qt(1-(0.05/2),df=n-1)
40 error3<-t3*s*sqrt(1+(1/n))
41 #Left endpoint
42 l3<-mean-error3
43 print(paste("Left endpoint of prediction interval:",
44             l3))
44 #Right endpoint
45 r3<-mean+error3
46 print(paste("Right endpoint of prediction interval:",
47             r3))

```

R code Exa 7.15 CI for variance and sd of normal population

```

1 #Ex7.15, Page 296
2 #Answers may slightly vary due to rounding off of
  values
3
4 data<-c
  (1470,1510,1690,1740,1900,2000,2030,2100,2190,
5   2200,2290,2380,2390,2480,2500,2580,2700)
6
7 interval<- function(data,conf.level=0.95){
8   df<-length(data)-1
9   l<-qchisq((1 - conf.level)/2, df)
10  r<-qchisq((1 - conf.level)/2, df, lower.tail =
    FALSE)
11  v<-var(data)
12  c(df* v/r,df*v/l)
13 }
14 print(paste("Confidence interval for variance of
    normal population:"))
15 print(interval(data))

```


Chapter 8

Test of hypotheses based on a single sample

R code Exa 8.1 Hypotheses and test procedures

```
1 #Ex8.1, Page 304
2 #Answers may vary slightly due to rounding off of
  values
3
4 #X<-number of crashes with no visible damage
5 #Reject H0 if x>=8 (x<-observed value of test
  statistic)
6
7 #When H0 is true
8 n<-20
9 p1<-0.25
10
11 #alpha=P(type I error)
12 alpha<-1-pbinom(7,n,p1)
13 print(paste("Alpha value:",alpha))
14
15 print(paste("When H0 is true, roughly",trunc(100*
  alpha,digits=2),"% of all experiments consisting
  of 20 crashes would result into a Type I error"))
```

```

16
17 #beta=P(type II error)
18 #Different beta for each different p that exceeds
    0.25, thus taking p=0.3
19 p2<-0.3
20 beta<-pbinom(7,n,p2)
21 print(paste("Beta value:",beta))
22 print(paste("When H0 is false , roughly",trunc(100*
    beta,digits=2),"% of all experiments would result
    into a Type II error"))
23
24 #Table showing variation of beta values with
    increase in value of p
25 for(i in 3:8){
26     p<-0.1*i
27     a<-pbinom(7,n,p)
28     print(paste("p-value:",p))
29     print(paste("Beta value:",a))
30 }

```

R code Exa 8.2 Hypotheses and test procedures

```

1 #Ex8.2 , Page 305
2 #Answers may vary slightly due to rounding off of
    values
3
4 m<-75
5 sd<-9
6 #Hypotheses H0: mean=75, H1: mean<75
7
8 #Experimental data
9 n<-25
10 sd<-9
11 #Standard deviation of normal distribution
12 sdx<-sd/sqrt(n)

```



```

13 print(paste("SD of normal distribution:",sdx))
14
15 #When H0 is true
16 mx<-75
17
18 #Consider rejection region x<=70.8
19 #To find alpha and beta
20
21 #P(type I error)
22 alpha<-pnorm(70.8,mx,sdx)
23 print(paste("Alpha value:",alpha))
24
25 #P(type II error)
26 mx<-c(72,70,67) #H0 is not rejected
    when it is false because mean= eg:72
27 for(i in mx){
28     beta<-1-pnorm(70.8,i,sdx)
29     print(paste("Beta value:",beta,"when mean is",i))
30 }
31
32 print(paste("Only",round(alpha,digits=3)*100,"% of
    all experiments will result into Type I error"))
33 print(paste("Possibility of Type II error decreases
    as mean value increases"))

```

R code Exa 8.3 Hypotheses and test procedures

```

1 #Ex8.3, Page 306
2 #Answers may vary slightly due to rounding off of
    values
3
4 #R9={9,10,.....,20}
5 n<-20
6
7 #To find alpha=P(H0 rejected when p=.25)=P(X>=9 when

```

```

      X follows binomial dist.)
8  p<-0.25
9  alpha<-1-pbinom(8,n,p)
10 print(paste("Alpha value:",alpha))
11 print(paste("Type I error probability has reduced
      considerably when using R9 rather than R8
      rejection region"))
12
13 #To find beta value=P(H0 not rejected)=P(X<=8 when X
      follows binomial dist.)
14 p1<-0.3      #1st case
15 beta1<-pbinom(8,n,p1)
16 print(paste("Beta(0.3)=",beta1))
17
18 p2<-0.5      #2nd case
19 beta2<-pbinom(8,n,p2)
20 print(paste("Beta(0.5)=",beta2))
21 print(paste("P(Type II error in R9)>P(Type II error
      in R8)"))

```

R code Exa 8.4 Hypotheses and test procedures

```

1 #Ex8.4, Page 307
2 #Answers may vary slightly due to rounding off of
      values
3
4 #New rejection region  $x \leq 72$ 
5 sdx<-1.8
6
7 #To find alpha=P(H0 rejected when true)=P(X<=72 when
      X follows normal dist)
8 mx<-75
9 alpha<-pnorm(72,mx,sdx)
10 print(paste("Alpha value:",alpha))
11 print(paste("Type I error probability is greater in

```

```

    the new rejection region"))
12
13 #To find beta=P(H0 not rejected)
14 mx<-c(72,70,67) #H0 is not rejected
    when it is false because mean= eg:72
15 for(i in mx){
16   beta<-1-pnorm(72,i,sdx)
17   print(paste("Beta(",i,"):",beta))
18 }
19 print(paste("Type II error probability is lower in
    the new rejection region"))

```

R code Exa 8.5 Hypotheses and test procedures

```

1 #Ex8.5 , Page 308
2
3 #H0: mu=1.5
4 #Ha: mu>1.5
5
6 mu0<-1.5
7 n<-32
8 sd<-0.2
9 sdx<-sd/sqrt(n)
10 print(paste("Rejection region: Z>=c"))
11
12 alpha<-0.05
13 #where alpha=P(type I error)=P(Z>=c when Z follows N
    (0,1))
14 c<-qnorm(1-alpha)
15 print(paste("C value:",c))
16
17 #Z>=c
18 x<-round(mu0+sdx*c,digits=2)
19 print(paste("xbar>=",x))
20 print(paste("Beta refers to the prob. that xbar<",x,

```

"and can be calculated for any $\mu > 1.5$ ")

R code Exa 8.6 Tests about a population mean

```
1 #Ex8.6, Page 312
2 #Answers may vary slightly due to rounding off of
  values
3
4 mu0<-130
5 n<-9
6 sd<-1.5
7 xbar<-131.08
8
9 z<-(xbar-mu0)/(sd/sqrt(n))           #Test statistic
  value
10 print(paste("Test statistic value:",z))
11
12 #To determine rejection region
13 alpha<-0.01
14 a<-qnorm(1-alpha/2)
15 l<-(-a)
16 r<-a
17 print(paste("Rejection region: z<=",l," ,z>=",r))
18 if(z<=l || z>=r)print(paste(z,"falls in the
  rejection region and H0 is rejected"))else print(
  paste(z,"does not fall in the rejection region so
  H0 cannot be rejected at significance level",
  alpha))
```

R code Exa 8.7 Tests about a population mean

```
1 #Ex8.7, Page 314
```

```

2 #Answers may vary slightly due to rounding off of
  values
3
4 n<-16
5 sd<-1500
6 alpha1<-0.01
7 xbar<-30000
8 mu0<-31000
9
10 zalpha1<-qnorm(1-alpha1)
11
12 z<-(xbar-mu0)/(sd/sqrt(n))
13
14 #To find beta(31000) when alpha=0.01
15 beta1<-pnorm(zalpha1+z)
16 print(paste("Beta(31000):",beta1,"when alpha=",
  alpha1))
17
18 alpha2<-0.1
19 zalpha2<-qnorm(1-alpha2)
20
21 #To find beta(31000) when alpha=0.1
22 beta2<-pnorm(zalpha2+z)
23 print(paste("Beta(31000):",round(beta2,digits=1),"
  when alpha=",alpha2))
24
25 #To find n
26 n<-(sd*(zalpha1+zalpha2)/(xbar-mu0))^2
27 print(paste("n value:",n))
28 print(paste("Sample size=",ceiling(n)))

```

R code Exa 8.8 Tests about a population mean

```

1 #Ex8.8, Page 315
2 #Answers may vary slightly due to rounding off of

```

```

        values
3
4  n<-52
5  mu0<-30
6  data<-c
      (14.1,14.5,15.5,16.0,16.0,16.7,16.9,17.1,17.5,17.8,17.8,18.1,18.2
7
      19.0,19.2,19.4,20.0,20.0,20.8,20.8,21.0,21.5,23.5,27.5,27.5,
8
      30.0,31.6,31.7,31.7,32.5,33.5,33.9,35.0,35.0,35.0,36.7,40.0,
9
      47.5,50.0,51.0,51.8,54.4,55.0,57.0)
10 xbar<-mean(data)
11 sd<-sd(data)
12 print(paste("Mean:",xbar))
13 print(paste("Standard deviation:",sd))
14
15 alpha<-0.05
16 #To find zalpha
17 z1<-qnorm(alpha)
18
19 #To find z
20 z<-(xbar-mu0)/(sd/sqrt(n))
21 print(paste("z value:",z))
22
23 if(z>z1) print(paste("H0 cannot be rejected since",z
      ,">",z1)) else print(paste("H0 is rejected since"
      ,z,"<",z1))

```

R code Exa 8.9 Tests about a population mean

```

1 #Ex8.9, Page 317
2 #Answers may vary slightly due to rounding off of
  values
3

```

```

4 data<-c(2.67,4.62,4.14,3.81,3.83)
5 n<-length(data)
6
7 alpha<-0.05
8 mu0<-4
9
10 #To determine rejection region
11 t1<-qt(1-alpha/2,df=n-1)
12 l<-(-t1)
13 r<-t1
14 print(paste("Rejection region: t<=",l," ,t>=",r))
15
16 #To find mean
17 print(paste("Sum of sample:",sum(data)))
18 print(paste("Mean:",mean(data)))
19
20 #To find standard deviation
21 print(paste("Standard deviation:",sd(data)))
22 print(paste("Estimated standard error of the mean:",
23             sd(data)/sqrt(n)))
24
25 #Test statistic value
26 t1<-t.test(data,alternative="two.sided",mu=mu0)
27 print(t1)

```

R code Exa 8.11 Tests concerning a population proportion

```

1 #Ex8.11, Page 324
2 #Answers may vary slightly due to rounding off of
3   values
4
5 #H0:p=0.15
6 #Ha:p>0.15
7
8 p0<-0.15

```

```

8 q0<-1-p0
9 n<-91
10 n1<-16
11
12 #Check validity of test procedures
13 if(n*p0>10 && n*q0>10) print(paste("Large sample z
    test can be used")) else print(paste("z test
    cannot be used"))
14
15 alpha<-0.1
16 zalpha<-qnorm(1-alpha)
17 print(paste("Rejection region: z>=",zalpha))
18
19 p<-n1/n
20 print(paste("p-cap:",p))
21
22 #To find z
23 z<-(p-p0)/(sqrt(p0*q0/n))
24 print(paste("Test statistic value:",z))
25
26 if(z<zalpha)print(paste("z=",z,"is not in the
    rejection region and hence, H0 cannot be rejected
    ")) else print(paste("z is in the rejection
    region"))

```

R code Exa 8.12 Tests concerning a population proportion

```

1 #Ex8.12, Page 326
2 #Answers may vary slightly due to rounding off of
  values
3
4 #H0: p=0.9
5 #Ha: p<0.9
6
7 n<-225

```



```

8  p0<-0.9
9  pdash<-0.8
10 alpha<-0.01
11
12 zalpha<-qnorm(1-alpha)
13
14 #To find beta(0.8)
15 num<-p0-pdash-(zalpha*sqrt((p0*(1-p0))/n))
16 denom<-sqrt(pdash*(1-pdash)/n)
17 beta<-1-pnorm(num/denom)
18 print(paste("Beta(0.8):",beta))
19
20 #P(H0 rejected)
21 prob<-1-beta
22 print(paste("P(H0 rejected)=",prob,"and thus",
              ceiling(prob*100),"% of samples will result in
              correct H0 rejection"))
23
24 #To find n
25 n<-((zalpha*sqrt(p0*(1-p0))+zalpha*sqrt(pdash*(1-
      pdash)))/(pdash-p0))^2
26 print(paste("Sample size:",ceiling(n)))

```

R code Exa 8.13 Tests concerning a population proportion

```

1  #Ex8.13, Page 327
2
3  #H0: p=0.9
4  #Ha: p<0.9
5  #Rejection region: x<=15
6
7  n<-20
8  pdash<-0.8
9  alpha<-0.05
10

```

```

11 #To find beta=P(H0 not rejected when X follows
    binomial dist)
12 beta<-1-pbinom(15,n,pdash)
13 print(paste("Beta(",pdash,")=",round(beta,digits=2))
    )
14
15 print(paste(ceiling(beta*100),"% of all samples
    causes H0 to be rejected incorrectly when p=",
    pdash))

```

R code Exa 8.14 P values

```

1 #Ex8.14, Page 329
2 #Answers may vary slightly due to rounding off of
    values
3
4 #H0: mu=2.0
5 #Ha: mu>2.0
6
7 mu0<-2
8 xbar<-2.06
9 sd<-0.141
10 n<-51
11
12 #To find z value
13 z<-(xbar-mu0)/(sd/sqrt(n))
14 print(paste("Test statistic value:",z))
15
16 #To find P-value=P(Z>=z when mu=2.0)
17 p<-1-pnorm(z)
18 print(paste("P-value:",p))
19
20 alpha<-c(0.01,0.001)
21 for(i in alpha){
22     if(p<=i)

```

```

23         print(paste("H0 rejected when significance
                        level is at",i))
24     else
25         print(paste("H0 not rejected when
                        significance level is at",i))
26 }

```

R code Exa 8.15 P values

```

1  #Ex8.15, Page 330
2  #Answers may vary slightly due to rounding off of
   values
3
4  #H0: mu=15
5  #Ha: mu>1.5
6
7  z<-2.10
8
9  #To determine p-value
10 p<-1-pnorm(z)
11 print(paste("P-value:",p))
12
13 alpha<-c(0.1,0.05,0.01)
14 for(i in alpha){
15     z1<-qnorm(1-i)
16     if(z>=z1)
17         print(paste("Reject H0 at significance level",i)
                )
18     else
19         print(paste("Do not reject H0 at significance
                        level",i))
20 }
21
22 #Using p-value approach
23 print(paste("Using p-value approach:"))

```

```

24 for(j in alpha){
25   if(j>=p)
26     print(paste("Reject H0 at significance level",j,
27               "since",j,">=",p))
27   else
28     print(paste("Do not reject H0 at significance
29               level",j,"since",j,"<",p))
29 }

```

R code Exa 8.17 P values for z tests

```

1 #Ex8.17, Page 333
2 #Answers may vary slightly due to rounding off of
  values
3
4 #H0: mu=245
5 #Ha: mu!=245
6
7 xbar<-246.18
8 mu0<-245
9 n<-50
10 sd<-3.6
11 alpha<-0.01
12
13 #To find z value
14 z<-(xbar-mu0)/(sd/sqrt(n))
15 print(paste("Test statistic value:",z))
16
17 #To find p value in two tailed experiment
18 p<-2*(1-pnorm(z))
19 print(paste("P-value:",p))
20
21 if(p>alpha) print(paste("H0 not rejected at
  significance level",alpha)) else print(paste("H0
  rejected at significance level",alpha))

```

R code Exa 8.18 P values for t tests

```
1 #Ex8.18 , Page 335
2
3 #H0: mu=4
4 #Ha: mu!=4
5
6 n<-5
7 t<-round(-0.594,digits=1)
8 df<-n-1
9
10 alpha<-0.05
11
12 #To find critical values
13 a<-qt(1-alpha/2,df=n-1)
14
15 #Since test is two tailed
16 p<-2*pt(t,df=n-1)
17 print(paste("P-value:",round(p,digits=2)))
18
19 b<-c(0.01,0.05,0.1)
20 for(i in b){
21   if(p>i)
22     print(paste("H0 is not rejected at significance
23                 level",i))
24   else
25     print(paste("H0 rejected at significance level",
26                 i))
27 }
```

R code Exa 8.19 Interpreting P values

```

1 #Ex8.19, Page 335
2 #Answers may vary slightly due to rounding off of
   values
3 #Tests for mean=20 have only been done....
4
5 #H0: mu=20
6 #Ha: mu>20
7
8 n<-4
9 df<-n-1
10 mu0<-20
11 sd<-2
12 x<-c(20.83,22.232,20.276,17.718)
13
14 xbar<-mean(x)
15 print(paste("Mean:",xbar))
16 s<-sd(x)
17 print(paste("Standard deviation:",s))
18
19 #To find t
20 t<-(xbar-mu0)/(s/sqrt(n))
21 print(paste("Test statistic value:",t))
22
23 #To find p value
24 p<-pt(t,df=n-1,lower.tail = FALSE)
25
26 alpha<-0.05
27 if(p>alpha) print(paste("H0 is not rejected at
   significance level",alpha)) else print(paste("H0
   is rejected at significance level",alpha))

```

Chapter 9

Inferences based on two samples

R code Exa 9.1 Test procedures for normal population with known variances

```
1 #Ex9.1 , Page 348
2 #Answers may slightly vary due to rounding off of
  values
3
4 #H0:  $\mu_1 - \mu_2 = 0$ 
5 #Ha:  $\mu_1 - \mu_2 \neq 0$ 
6
7 xbar <- 29.8
8 ybar <- 34.7
9 s1 <- 4
10 s2 <- 5
11 m <- 20
12 n <- 25
13
14 #To find percentile
15 alpha <- 0.01
16 H0 <- qnorm(1-alpha/2)
17
```

```

18 #To find test statistic value
19 z<-(xbar-ybar)/sqrt((s1^2)/m+(s2^2)/n)
20 print(paste("Test statistic value:",z))
21
22 if(z>=H0||z<=H0)print(paste("H0 is rejected at level
    ",alpha)) else print(paste("H0 is accepted at
    level",alpha))

```

R code Exa 9.2 Using a comparison to identify causality

```

1 #Ex9.2, Page 349
2 #Answers may slightly vary due to rounding off of
  values
3
4 #H0:  $\mu_1 - \mu_2 = 0$ 
5 #Ha:  $\mu_1 - \mu_2 > 0$ 
6
7 xbar<-48.9
8 ybar<-43.2
9 s1<-14.6
10 s2<-14.4
11 m<-125
12 n<-90
13 alpha<-0.01
14
15 #To find test statistic value
16 z<-(xbar-ybar)/sqrt((s1^2)/m+(s2^2)/n)
17 print(paste("Test statistic value:",z))
18
19 #To find p-value
20 p<-pnorm(-abs(z))
21 print(paste("p-value:",p))
22
23 if(alpha>p) print(paste("H0 is rejected at level",
    alpha)) else print(paste("H0 is accepted at

```



```
level",alpha))
```

R code Exa 9.3 Beta and the choice of sample size

```
1 #Ex9.3, Page 350
2 #Answers may slightly vary due to rounding off of
  values
3
4 t<-5
5 t1<-0
6 s<-1.34
7 n<-qnorm(1-0.01/2)
8 beta<-pnorm(n-((t-t1)/s))-pnorm(-n-((t-t1)/s))
9 print(paste("Beta(5):",beta))
```

R code Exa 9.4 Large sample tests

```
1 #Ex9.4, Page 352
2 #Answers may vary slightly due to rounding off of
  values
3
4 mu1<-2258
5 mu2<-2637
6 s1<-1519
7 s2<-1138
8 m<-663
9 n<-413
10 l<-0.05
11 H0<-(-200)
12
13 #To find z0.05
14 z1<-qnorm(1-l)
15
```

```

16 #To find test statistic value
17 z<-(mu1-mu2-H0)/sqrt(((s1^2)/m+(s2^2)/n))
18 print(paste("Test statistic value:",z))
19
20 if(z<=-z1) print(paste("H0 is rejected since",z,"<="
    ,-z1))
21
22 #To find p-value
23 p<-pnorm(-abs(z))
24 print(paste("P-value:",p))
25 if(p<=1) print(paste("H0 is rejected since",p,"<=",1
    ))

```

R code Exa 9.5 Confidence intervals for difference of means

```

1 #Ex9.5, Page 354
2 #Answers may vary slightly due to rounding off
  values
3
4 c1<-c('Variable_diam', 'N', 'Mean', 'Median', '
    TrMean', 'StDev', 'SEMean', 'Min', 'Max', 'Q1', '
    Q3')
5 c2<-c('3/8'
    ,78,4.250,4.230,4.238,1.300,0.147,1.634,7.327,3.389,5.075)
6 c3<-c('1/2'
    ,88,7.140,7.113,7.150,1.680,0.179,2.450,11.343,5.965,8.447)
7 df<-data.frame(c1,c2,c3)
8 print(df)
9
10 mu1<-4.25
11 mu2<-7.14
12 z<-qnorm(1-(0.05/2))
13 s1<-1.3

```

```

14 s2<-1.68
15 m<-78
16 n<-88
17
18 #To find confidence intervals
19 l<-(mu1-mu2)-z*sqrt((s1^2)/m+(s2^2)/n)
20 r<-(mu1-mu2)+z*sqrt((s1^2)/m+(s2^2)/n)
21 print(paste("Left endpoint of interval:",l))
22 print(paste("Right endpoint of interval:",r))

```

R code Exa 9.6 Two sample t test and confidence interval

```

1 #Ex9.6 , Page 358
2 #Answers may vary slightly due to rounding off of
  values
3
4 Fabric_type<-c('Cotton','Triacetate')
5 Sample_size<-c(10,10)
6 Sample_mean<-c(51.71,136.14)
7 Sample_standard_deviation<-c(.79,3.59)
8 data<-data.frame(Sample_size,Sample_mean,Sample_
  standard_deviation,row.names=Fabric_type)
9 print(data)
10
11 #Assigning values to variables
12 s1<-Sample_standard_deviation[1]
13 s2<-Sample_standard_deviation[2]
14
15 xbar<-Sample_mean[1]
16 ybar<-Sample_mean[2]
17
18 m<-Sample_size[1]
19 n<-Sample_size[2]
20
21 #To find df

```

```

22 num<-((s1^2/m)+(s2^2/n))^2
23 denom<-((s1^2/m)^2)/(m-1)+((s2^2/n)^2)/(n-1)
24 df<-num/denom
25 print(paste("df:",df))
26
27 #To find t value
28 alpha<-0.05
29 t<-qt(1-alpha/2,df=m-1)
30 print(paste("t:",t))
31
32 #To find intervals
33 l<-xbar-ybar-t*sqrt((s1^2/m)+(s2^2/n))
34 r<-xbar-ybar+t*sqrt((s1^2/m)+(s2^2/n))
35 print(paste("Left endpoint of interval:",l))
36 print(paste("Right endpoint of interval:",r))
37
38 print(paste("The true average porosity of the
    triacetate fabric is greater than that of the
    cotton fabric by between",round(-r,digits=2),"and
    ",round(-l,digits=2),"cm^3/cm^2/sec"))

```

R code Exa 9.7 Two sample t test and confidence interval

```

1 #Ex9.7, Page 359
2 #Answers may vary slightly due to rounding off of
  values
3
4 No_fusion<-c
  (2748,2700,2655,2822,2511,3149,3257,3213,3220,2753)
5
6 Fused<-c(3027,3356,3359,3297,3125,2910,2889,2902)
7
8 #H0: mu1-mu2=0
9 #Ha: mu1-mu2<0

```

```

10 #To find test staistic value
11 t<-t.test(No_fusion,Fused)
12 print(t)
13 print(paste("t statistic value:",t$statistic))
14 print(paste("df:",t$parameter))
15
16 #To plot normal probability plots
17 par(mfrow=c(2,1))
18 qqnorm(No_fusion,datax=TRUE,ylab="Not fused",xlab="
    Probability",main="Normal probability plot for
    not fused data")
19 qqline(No_fusion,datax=TRUE)
20 qqnorm(Fused,datax=TRUE,ylab="Fused",xlab="
    Probability",main="Normal probability plot for
    fused data")
21 qqline(Fused,datax=TRUE)

```

R code Exa 9.9 Paired t test

```

1 #Ex9.9, Page 367
2 #Answers may vary slightly due to rounding off of
  values
3
4 Subject<-c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)
5 Before<-c
  (81,87,86,82,90,86,96,73,74,75,72,80,66,72,56,82)
6 After<-c
  (78,91,78,78,84,67,92,70,58,62,70,58,66,60,65,73)
7 Difference<-Before-After
8 data<-data.frame(Before,After,Difference,row.names=
  Subject)
9 print(data)
10
11 #Normal probability plot
12 par(mfrow=c(1,2))

```

```

13 qqnorm(data$Difference,main="Normal probability plot
    of the differences",xlab="diff",ylab="
    Probability")
14 qqline(data$Difference)
15 boxplot(Difference,horizontal=TRUE,xlab="Difference"
    )
16
17 #H0: mu=0
18 #Ha: mu!=0
19 t1<-t.test(Difference,mu=0)
20 print(paste("t statistic value:",t1$statistic))
21 print(paste("p value:",t1$p.value))
22
23 alpha<-0.01
24 if(t1$p.value<alpha) print(paste("H0 rejected"))
    else print(paste("H0 not rejected"))

```

R code Exa 9.10 Paired t confidence interval

```

1 #Ex9.10, Page 369
2 #Answers may vary slightly due to rounding off of
    values
3
4 Subject<-c(1,2,3,4,5,6,7,8,9,10,11,12,13)
5 Slide<-c(30,35,40,25,20,30,35,62,40,51,25,42,33)
6 Digital<-c(25,16,15,15,10,20,7,16,15,13,11,19,19)
7 Difference<-Slide-Digital
8 data<-data.frame(Slide,Digital,Difference,row.names=
    Subject)
9 print(data)
10
11 #Normal probability plot
12 qqnorm(data$Difference,main="Normal probability plot
    of the differences",ylab="diff",xlab="
    Probability")

```

```

13 qqline(data$Difference)
14
15 #To find t critical value
16 alpha<-0.05
17 t<-qt(1-alpha/2,df=12)
18 print(paste("t statistic value:",t))
19
20 #To find confidence intervals
21 t1<-t.test(Slide,Digital,paired=TRUE)
22 print(t1)

```

R code Exa 9.11 Inferences concerning difference between population proportions

```

1 #Ex9.11, Page 376
2 #Answers may slightly vary due to rounding off of
  values
3
4 alpha<-0.05
5
6 #H0:  $p_1 - p_2 = 0$ 
7 #Ha:  $p_1 - p_2 < 0$ 
8
9 a<-81
10 tot1<-549
11 p1cap<-a/tot1
12 q1cap<-1-p1cap
13
14 b<-141
15 tot2<-730
16 p2cap<-b/tot2
17 q2cap<-1-p2cap
18
19 pcap<-(a+b)/(tot1+tot2)
20 qcap<-1-pcap

```

```

21
22 #To find test statistic value
23 if((tot1*p1cap)>=10 && (tot1*q1cap)>=10 && (tot2*
    p2cap)>=10 && (tot2*q2cap)>=10)
24     z<-(p1cap-p2cap)/sqrt(pcap*qcap*(1/tot1+1/tot2))
25 print(paste("Test statistic value:",z))
26
27 #To find P-value
28 p<-pnorm(z)
29 print(paste("P-value:",p))
30
31 if(p<=alpha) print(paste("H0 can be rejected at
    significance level",alpha)) else print(paste("H0
    cannot be rejected"))

```

R code Exa 9.12 Type II error probabilities and sample sizes

```

1 #Ex9.12, Page 378
2 #Answers may vary slightly due to rounding off of
  values
3
4 #H0: p1-p2=0
5 #Ha: p1-p2>0
6
7 a<-30
8 tot1<-100000
9 p1<-a/tot1
10 print(paste("p1:",p1))
11 q1<-1-p1
12
13 b<-a/2
14 p2<-b/tot1
15 print(paste("p2:",p2))
16 q2<-1-p2
17

```



```

18 alpha<-0.05
19 beta<-0.1
20
21 zalpha<-qnorm(1-alpha)
22 zbeta<-qnorm(1-beta)
23 d<-p1-p2
24
25 #To find n value
26 n<-(zalpha*sqrt((p1+p2)*(q1+q2)/2)+zbeta*sqrt(p1*q1+
    p2*q2))^2/d^2
27 print(paste("Value of n:",n))

```

R code Exa 9.13 A large sample confidence interval

```

1 #Ex9.13, Page 380
2 #Answers may vary slightly due to rounding off of
  values
3
4 alpha<-0.01
5 a<-76      #number of people who survived through
  chemotherapy treatment
6 m<-154     #number of people who received
  chemotherapy treatment
7 p1cap<-a/m
8 q1cap<-1-p1cap
9 print(paste("p1cap:",p1cap))
10
11 b<-98      #number of people who survived through
  hybrid treatment
12 n<-164     #number of people who received hybrid
  treatment
13 p2cap<-b/n
14 q2cap<-1-p2cap
15 print(paste("p2cap:",p2cap))
16

```

```

17 #To find z
18 zalpha<-qnorm(1-alpha/2)
19
20 if(m*p1cap>=10 && m*q1cap>=10 && n*p2cap>=10 && n*
    q2cap>=10)
21   l<-p1cap-p2cap-zalpha*sqrt((p1cap*q1cap/m)+(p2cap*
    q2cap/n))
22   r<-p1cap-p2cap+zalpha*sqrt((p1cap*q1cap/m)+(p2cap*
    q2cap/n))
23
24 print(paste("Left endpoint of confidence interval:",
    l))
25 print(paste("Right endpoint of confidence interval:"
    ,r))
26 print(paste("Hence",round(l,digits=3),"<p1-p2<" ,
    round(r,digits=3)))

```

R code Exa 9.14 F test for equality of variances

```

1 #Ex9.14, Page 385
2 #Answers may vary slightly due to rounding off of
  values
3
4 s1<-52.6
5 s2<-84.2
6 m<-28
7 n<-26
8
9 #H0:  $s_1^2=s_2^2$ 
10 #Ha:  $s_1^2<s_2^2$ 
11
12 alpha<-0.01
13 #To find F value
14 Falpha<-qf(alpha,df1=m-1,df2=n-1)
15 print(paste("F value:",Falpha))

```

```
16
17 f<-(s1^2)/(s2^2)
18 print(paste("f value:",f))
19
20 if(f<=Falpha) print(paste("H0 is rejected at
    significance level",alpha)) else print(paste("H0
    is not rejected"))
```

Chapter 10

The analysis of variance

R code Exa 10.1 Single factor ANOVA

```
1 #Ex10.1 , Page 392
2 #Answers may vary slightly due to rounding off of
  values
3
4 Type_of_box<-c(1,2,3,4)
5 CS1<-c(655.5,789.2,737.1,535.1)
6 CS2<-c(788.3,772.5,639.0,628.7)
7 CS3<-c(734.3,786.9,696.3,542.4)
8 CS4<-c(721.4,686.1,671.7,559.0)
9 CS5<-c(679.1,732.1,717.2,586.9)
10 CS6<-c(699.4,774.8,727.1,520.0)
11 Mean<-c(NULL)
12 SD<-c(NULL)
13
14 #To find mean of observations
15 for(i in 1:4){
16   sum<-CS1[i]+CS2[i]+CS3[i]+CS4[i]+CS5[i]+CS6[i]
17   mean<-sum/6
18   Mean[i]<-mean
19 }
20 #To find standard deviation of values
```

```

21 for(i in 1:4){
22   s<-(CS1[i]-Mean[i])^2+(CS2[i]-Mean[i])^2+(CS3[i]-
      Mean[i])^2+(CS4[i]-Mean[i])^2+(CS5[i]-Mean[i])
      ^2+(CS6[i]-Mean[i])^2
23   sd<-sqrt(s/5)
24   SD[i]<-sd
25 }
26
27 data<-data.frame(Type_of_box,CS1,CS2,CS3,CS4,CS5,CS6
      ,Mean,SD)
28 print(data)
29
30 #To find grand mean of mean values
31 tot_mean<-0
32 for(i in 1:4){
33   tot_mean<-tot_mean+Mean[i]
34 }
35 tot_mean<-tot_mean/4
36 print(paste("Grand mean:",tot_mean))
37
38 par(mfrow=c(1,2))
39 #To create boxplots of the data
40 #First element of each CS column gives us the
      compression strength of one box
41 a<-c(CS1[1],CS2[1],CS3[1],CS4[1],CS5[1],CS6[1])
42 b<-c(CS1[2],CS2[2],CS3[2],CS4[2],CS5[2],CS6[2])
43 c<-c(CS1[3],CS2[3],CS3[3],CS4[3],CS5[3],CS6[3])
44 d<-c(CS1[4],CS2[4],CS3[4],CS4[4],CS5[4],CS6[4])
45 boxplot(a,b,c,d)
46 title(main="Boxplots for original data")
47
48 #Altering data in sample 2 by adding 120 to every
      value in Sample 4
49 e<-c(NULL)
50 for(i in 1:6){
51   e[i]<-d[i]+120
52 }
53 boxplot(a,b,c,e)

```

```
54 title(main="Boxplots for altered data")
```

R code Exa 10.2 F distribution and the F test

```
1 #Ex10.2, Page 397
2 #Answers may vary slightly due to rounding off of
  values
3
4 #From Ex10.1
5 Mean<-c(713.00,756.93,698.07,562.02)
6 SD<-c(46.55,40.34,37.20,39.87)
7 tot_mean<-682.50
8
9 I<-4
10 J<-6
11
12 #Numerator
13 v1<-I-1
14
15 #Denominator
16 v2<-I*(J-1)
17
18 alpha<-0.05
19
20 #H0:mu1=mu2=mu3=mu4
21 #Ha:at least two mean values will be different
22
23 #To find F critical value
24 F<-qf(p=1-alpha,df1=v1,df2=v2)
25 print(paste("Critical value F:",F))
26
27 #To find mean square for treatment
28 MStr<-(J/(I-1))*sum((Mean-tot_mean)^2)
29 print(paste("MStr:",MStr))
30
```

```

31 #To find mean square for error
32 MSE<-(1/I)*sum(SD^2)
33 print(paste("MSE:",MSE))
34
35 #To find f value
36 f<-MSTr/MSE
37 print(paste("f:",f))
38
39 if(f>=F) print(paste("H0 is rejected at
    significance level",alpha,"since f>=F")) else
    print(paste("H0 not rejected"))

```

R code Exa 10.3 F distribution and the F test

```

1 #Ex10.3, Page 397
2 #Answers may vary slightly due to rounding off of
  values
3
4 Treatment<-c(1,2,3,4,5)
5 Sample_mean<-c(10.5,14.8,15.7,16.0,21.6)
6 Sample_sd<-c(4.5,6.8,6.5,6.7,6.0)
7
8 #To find total mean of mean values
9 tot_mean<-mean(Sample_mean)
10 print(paste("Grand mean:",tot_mean))
11
12 #H0: mu1=mu2=mu3=mu4=mu5
13 #Ha: at least two of the mean values are different
14
15 I<-5
16 J<-10
17 alpha<-0.01
18
19 #Numerator
20 v1<-I-1

```

```

21
22 #Denominator
23 v2<-I*(J-1)
24
25 #To find F value
26 F<-qf(1-alpha,v1,v2)
27 print(paste("Critical value F:",F))
28
29 #To find mean square for treatment
30 s<-Sample_mean-tot_mean
31 MStr<-(J/(I-1))*sum(s^2)
32 print(paste("MStr:",MStr))
33
34 #To find mean square for error
35 MSE<-mean(Sample_sd^2)
36 print(paste("MSE:",MSE))
37
38 #To find f value
39 f<-MStr/MSE
40 print(paste("f:",f))
41
42 if(f>=F) print(paste("H0 is rejected at
    significance level",alpha,"since f>=F")) else
    print(paste("H0 not rejected"))

```

R code Exa 10.4 Sums of squares

```

1 #Ex10.4, Page 400
2 #Answers may vary slightly due to rounding off of
  values
3
4 Mixture1<-c(0.56,1.12,0.90,1.07,0.94)
5 Mixture2<-c(0.72,0.69,0.87,0.78,0.91)
6 Mixture3<-c(0.62,1.08,1.07,0.99,0.93)
7

```



```

8 data1<-data.frame(Mixture1,Mixture2,Mixture3)
9
10 I<-3
11 J<-5
12
13 #Numerator
14 v1<-I-1
15
16 #Denominator
17 v2<-I*(J-1)
18
19 alpha<-0.01
20 #To find F value
21 F<-qf(1-alpha,v1,v2)
22 print(paste("Critical value , F:",F))
23
24
25 #To create ANOVA table
26 mix<-stack(data1)
27 names(mix)<-c("Degree","Treatment")
28
29 a<-aov(Degree~Treatment,data=mix)
30 print(paste("ANOVA table:"))
31 a1<-summary(a)
32 print(a1)
33 print(paste("f value:",a1[[1]]$F[1]))

```

R code Exa 10.6 Multiple comparisons in ANOVA

```

1 #Ex10.6 , Page 404
2
3 Treatment1<-c(88.6,73.2,91.4,68.0,75.2)
4 Treatment2<-c(63.0,53.9,69.2,50.1,71.5)
5 Treatment3<-c(44.9,59.5,40.2,56.3,38.7)
6 Treatment4<-c(31.0,39.6,45.3,25.2,22.7)

```

```

7 data1<-data.frame(Treatment1,Treatment2,Treatment3,
  Treatment4)
8
9 Treatment<-stack(data1)
10 names(Treatment)<-c("Time","Model")
11
12 #To display ANOVA table
13 a<-aov(Time~Model,data=Treatment)
14 print(summary(a))

```

R code Exa 10.9 Unequal sample sizes

```

1 #Ex10.9, Page 412
2
3 Permanent_molding<-c
  (45.5,45.3,45.4,44.4,44.6,43.9,44.6,44.0)
4 Die_casting<-c
  (44.2,43.9,44.7,44.2,44.0,43.8,44.6,43.1)
5 Plaster_molding<-c(46.0,45.9,44.8,46.2,45.1,45.5)
6
7 #To display ANOVA table
8 data1<-data.frame(Mold=c(Permanent_molding,Die_
  casting,Plaster_molding),Treatment=factor(rep(c("
  Mold1","Mold2","Mold3"),times=c(length(Permanent_
  molding),length(Die_casting),length(Plaster_
  molding)))))
9
10 cat("ANOVA table\n")
11 a<-aov(Mold~Treatment,data=data1)
12 print(summary(a))

```

R code Exa 10.10 Unequal sample sizes

```

1 #Ex10.10 , Page 413
2
3 alpha<-0.95
4 n<-22
5 J1<-8
6 J2<-8
7 J3<-6
8 I<-3
9 J<-n-I
10 MSE<-0.316
11 Q<-qtukey(alpha,I,J)
12 xibar<-c(44.71,44.06,45.58)
13
14 #To find w12
15 w12<-Q*sqrt((MSE/2)*(1/J1+1/J2))
16 print(paste("w12:",w12))
17 w23<-Q*sqrt((MSE/2)*(1/J2+1/J3))
18 print(paste("w23:",w23))
19 w31<-Q*sqrt((MSE/2)*(1/J3+1/J1))
20 print(paste("w31:",w31))
21
22 if(xibar[1]-xibar[2]< w12){
23   print(paste("mu1 and mu2 are not significantly
                different"))
24 }
25 if(xibar[2]-xibar[3]< w23){
26   print(paste("mu2 and mu3 are not significantly
                different"))
27 }
28 if(xibar[3]-xibar[1]< w31){
29   print(paste("mu3 and mu1 are not significantly
                different"))
30 }

```

R code Exa 10.11 A random effects model

```

1 #Ex.10.11, Page 415
2
3 #H0:  $\sigma_A^2=0$ 
4
5 C1<-c(55,53,54)
6 C2<-c(26,37,32)
7 C3<-c(78,91,85)
8 C4<-c(92,100,96)
9 C5<-c(49,51,50)
10 C6<-c(80,85,83)
11 data1<-data.frame(C1,C2,C3,C4,C5,C6)
12 print(paste("x..=",sum(data1)))
13
14
15 #To create ANOVA table
16 col<-stack(data1)
17 names(col)<-c("Type","Treatments")
18 a<-aov(Type~Treatments,data=col)
19 print(paste("ANOVA table:"))
20 print(summary(a))

```

Chapter 11

Multifactor analysis of variance

R code Exa 11.1 Two factor ANOVA

```
1 #Ex11.1, Page 420
2
3 dat<-c
   (0.97,0.48,0.48,0.46,0.77,0.14,0.22,0.25,0.67,0.39,0.57,0.19)

4 mat1<-matrix(dat,nrow=3,ncol=4,byrow=TRUE)
5 rownames(mat1)<-c("Brand1","Brand2","Brand3")
6 colnames(mat1)<-c("WT1","WT2","WT3","WT4")
7 print(mat1)
8
9 print(paste("Number of levels of factor A(Brand of
   pen), I:",nrow(mat1)))
10 print(paste("Number of levels of factor B(Washing
   treatment), J:",ncol(mat1)))
```

R code Exa 11.3 Fixed effects model

```
1 #Ex11.3, Page 424
```

```

2 #Answers may vary slightly due to rounding off of
  values
3
4 #To create data frame
5 brand<-rep(c(" I", " II", " III"),4)
6 WT<-rep(c(" 1", " 2", " 3", " 4"),each=3) #WT<-Washing
  treatment
7 col_change<-c
  (0.97,0.77,0.67,0.48,0.14,0.39,0.48,0.22,0.57,0.46,0.25,0.19)

8
9 df1<-data.frame(brand,WT,col_change)
10 print(df1)
11
12 #To display ANOVA table
13 a<-aov(col_change~brand+WT,data=df1)
14 print(a)
15 cat("\nSummarized table:\n")
16 print(summary(a))
17
18 #Diagnostic plots
19 model<-lm(col_change~brand+WT,data=df1)
20 par(mfrow=c(1,2))
21 qqnorm(model$residuals,datax=TRUE,main="Normal
  probability plot",ylab="Residual",xlab="Percent")
22 qqline(model$residuals,datax=TRUE)
23
24 plot(df1$col_change,resid(model),main="Residual plot
  ",xlab="Fitted value",ylab="Residual")
25 abline(0,0)

```

R code Exa 11.5 Randomized block experiments

```

1 #Ex11.5, Page 427
2 #Answers may vary slightly due to rounding off of

```

```

      values
3
4 #To create data frame
5 Treatment<-rep(c(" I", " II", " III", " IV", " V"), 4)
6 Block<-rep(c(" 1", " 2", " 3", " 4"), each=5)
7 Power<-c
      (685, 722, 733, 811, 828, 792, 806, 802, 888, 920, 838, 893, 880, 952, 978, 875,
8
9 df1<-data.frame(Treatment, Block, Power)
10 print(df1)
11
12 #To display ANOVA table
13 a<-aov(Power~Treatment+Block, data=df1)
14 print(a)
15 cat("\nSummarized table:\n")
16 print(summary(a))

```

R code Exa 11.6 Randomized block experiments

```

1 #Ex11.6, Page 428
2 #Answers may vary slightly due to rounding off of
      values
3
4 #To create data frame
5 Tension<-rep(c(" 210", " 235", " 260", " 285"), 6)
6 Player<-rep(c(" 1", " 2", " 3", " 4", " 5", " 6"), each=4)
7 Speed<-c
      (105.7, 113.3, 117.2, 110.0, 116.6, 119.9, 124.4, 106.8, 106.6, 120.5, 122.1,
8
9 df1<-data.frame(Tension, Player, Speed)
10 print(df1)
11
12 #To find mean for corresponding tension data

```

```

13 cat("\nTension mean values:\n")
14 m1<-tapply(df1$Speed,list(Tension),mean)
15 print(m1)
16
17 #To find mean of corresponding player data
18 cat("\nPlayer mean values:\n")
19 m2<-tapply(df1$Speed,list(Player),mean)
20 print(m2)
21
22 #To display ANOVA table
23 a<-aov(Speed~Tension+Player,data=df1)
24 print(a)
25 print(summary(a))

```

R code Exa 11.7 Two factor ANOVA with Kij greater than 1

```

1 #Ex11.7, Page 435
2
3 AggCont<-rep(c("38","41","44"),each=6)
4 AsphGr<-rep(c("PG58","PG64","PG70"),6)
5 data<-c
      (0.835,0.855,0.815,0.845,0.865,0.825,0.822,0.832,0.800,0.826,0.83
6 df1<-data.frame(AggCont,AsphGr,data)
7 print(df1)
8
9 #To find mean for corresponding content data
10 cat("\nCoarse aggregate content(%) mean values:\n")
11 m1<-tapply(df1$data,list(AggCont),mean)
12 print(m1)
13
14 #To find mean of corresponding Asphalt binder grade
    data
15 cat("\nAsphalt binder grade mean values:\n")
16 m2<-tapply(df1$data,list(AsphGr),mean)

```



```

17 print(m2)
18
19
20 #To display ANOVA table
21 model<-lm(data~AsphGr+AggCont+AsphGr:AggCont)
22 a<-aov(data~AsphGr+AggCont+AsphGr:AggCont)
23 print(a)
24 print(summary(a))
25
26 #To display interaction plots(Thermal diffusivity
    data is not given)
27 interaction.plot(AggCont,AsphGr,response=data)
28
29 #Plots for checking normality and constant variance
    assumptions
30 par(mfrow=c(1,2))
31 qqnorm(model$residuals,datax=TRUE,main="Normal
    probability plot",ylab="Residual",xlab="Percent")
32 qqline(model$residuals,datax=TRUE)
33
34 plot(df1$data,resid(model),main="Residual plot",xlab
    ="Fitted value",ylab="Residual")
35 abline(0,0)

```

R code Exa 11.9 Models with mixed and random effects

```

1 #Ex11.9, Page 439
2 #Answers may vary slightly from textbook values
3
4 #To create data frame
5 Casmater<-rep(c("Steel","Aluminium","Plastic"),10)
6 Source<-rep(c("1","2","3","4","5"),each=6)
7 Power<-c
    (13.1,15.0,14.0,13.2,14.8,14.3,16.3,15.7,17.2,15.8,16.4,16.7,13.7

```

```

8
9 df1<-data.frame(Casmater,Source,Power)
10 print(df1)
11
12 #To create ANOVA table
13 a<-aov(Power~Casmater+Source+Casmater:Source,data=
      df1)
14 print(a)
15 cat("\nSummarized table:\n")
16 print(summary(a))

```

R code Exa 11.10 Three factor ANOVA

```

1 #Ex11.10, Page 444
2 #Answers may vary slightly due to rounding off of
  values
3
4 #To create data frame
5 A<-rep(c(rep(c("A1","A2","A3","A4"),each=3)),8)
6 B<-rep(c("B1","B2"),each=48)
7 C<-rep(c("C1","C2","C3","C4","C1","C2","C3","C4"),
      each=12)
8 data<-c
      (3.6,3.8,3.9,3.8,3.6,4.0,3.7,3.9,4.2,3.6,3.5,3.8,3.4,3.7,3.9,3.8,
9
10 df1<-data.frame(A,B,C,data)
11 print(df1)
12
13 #To find cell totals for all combinations of the
    three factors
14 cat("\nCell totals:\n")
15 m1<-tapply(df1$data,list(A,B,C),sum)
16 print(m1)
17

```

```

18 #To create ANOVA table
19 a<-aov(data~A+B+C+A:B+A:C+B:C+A:B:C,data=df1)
20 print(a)
21 cat("\nSummarized table:\n")
22 print(summary(a))

```

R code Exa 11.11 Latin square designs

```

1 #Ex11.11, Page 447
2 #Answers may vary from textbook values
3
4 library(nlme)
5 library(multcomp)
6
7 #Dependent variable
8 data<-c
   (7.38,7.15,6.75,8.05,5.65,6.00,5.39,8.16,5.64,6.45,5.44,6.55,5.03
9
10 #Repeated measures
11 ##B(Columns)
12 B<-factor(rep(c("1","2","3","4","5","6"),each=6))
13
14 ##A(Rows)
15 A<-factor(rep(c("I","II","III","IV","V","VI"),6))
16
17 df1<-data.frame(A,B,data)
18 print(df1)
19
20 #To create ANOVA table
21 #using aov() function
22 cat("\nUsing aov():\n")
23 a1<-aov(data~A+Error(B/A),data=df1)
24 print(summary(a1))
25

```

```

26 #using lme()
27 cat("\nUsing lme():\n")
28 model<-lme(data~A,random=~1|B/A,data=df1)
29 a2<-anova(model)
30 print(a2)
31
32 #Tukey test
33 print(summary(glht(model,linfct=mcp(A="Tukey"))))

```

R code Exa 11.12 Factorial experiments with p factors at two levels

```

1 #Ex11.12, Page 452
2
3 Age<-rep(c("1","2"),each=2,4)
4 Temp<-rep(c("1","2"),8)
5 Soil<-rep(c("1","2"),each=8)
6 Comp_Stre<-c
   (471,485,712,712,413,552,637,789,385,530,770,741,434,593,705,806)
7 df1<-data.frame(Age,Temp,Soil,Comp_Stre)
8
9 #To compute cell totals
10 s1<-s2<-s3<-s4<-s5<-s6<-s7<-s8<-c(NULL)
11 for(i in 1:length(Comp_Stre)){
12   if(df1$Age[i]=="1" && df1$Temp[i]=="1" && df1$Soil
      [i]=="1"){
13     s1[i]<-df1$Comp_Stre[i]
14     x111<-sum(s1,na.rm=TRUE)
15   }
16   else if(df1$Age[i]=="1" && df1$Temp[i]=="2" && df1
      $Soil[i]=="1"){
17     s2[i]<-df1$Comp_Stre[i]
18     x121<-sum(s2,na.rm=TRUE)
19   }
20   else if(df1$Age[i]=="1" && df1$Temp[i]=="1" && df1

```

```

    $Soil[i]=="2"){
21     s3[i]<-df1$Comp_Stre[i]
22     x112<-sum(s3,na.rm=TRUE)
23   }
24   else if(df1$Age[i]=="2" && df1$Temp[i]=="1" && df1
    $Soil[i]=="1"){
25     s4[i]<-df1$Comp_Stre[i]
26     x211<-sum(s4,na.rm=TRUE)
27   }
28   else if(df1$Age[i]=="2" && df1$Temp[i]=="1" && df1
    $Soil[i]=="2"){
29     s5[i]<-df1$Comp_Stre[i]
30     x212<-sum(s5,na.rm=TRUE)
31   }
32   else if(df1$Age[i]=="1" && df1$Temp[i]=="2" && df1
    $Soil[i]=="2"){
33     s6[i]<-df1$Comp_Stre[i]
34     x122<-sum(s6,na.rm=TRUE)
35   }
36   else if(df1$Age[i]=="2" && df1$Temp[i]=="2" && df1
    $Soil[i]=="1"){
37     s7[i]<-df1$Comp_Stre[i]
38     x221<-sum(s7,na.rm=TRUE)
39   }
40   else if(df1$Age[i]=="2" && df1$Temp[i]=="2" && df1
    $Soil[i]=="2"){
41     s8[i]<-df1$Comp_Stre[i]
42     x222<-sum(s8,na.rm=TRUE)
43   }
44
45 }
46 #Parameters of the model
47 n<-2
48 alpha1_cap<-(x111+x121+x112+x122-x211-x212-x221-x222
  )/(8*n)
49 alpha2_cap<-(-1)*alpha1_cap
50
51 gamma11_cap<-(x111-x121-x211+x221+x112-x122-x212+

```

```

      x222)/(8*n)
52 gamma12_cap<-(-1)*gamma11_cap
53 gamma21_cap<-(-1)*gamma11_cap
54 gamma22_cap<-gamma11_cap
55
56 print(paste("alpha1_cap:",alpha1_cap))
57 print(paste("alpha2_cap:",alpha2_cap))
58 print(paste("gamma11_cap:",gamma11_cap))
59 print(paste("gamma12_cap:",gamma12_cap))
60 print(paste("gamma21_cap:",gamma21_cap))
61 print(paste("gamma22_cap:",gamma22_cap))

```

R code Exa 11.13 Factorial experiments with p factors at two levels

```

1 #Ex11.13, Page 453
2 #Answers may vary slightly due to rounding off of
  values
3
4 Age<-factor(rep(c("1","2"),each=2,4))
5 Temp<-factor(rep(c("1","2"),8))
6 Soil<-factor(rep(c("1","2"),each=8))
7 Comp_Stre<-c
  (471,485,712,712,413,552,637,789,385,530,770,741,434,593,705,806)
8 df1<-data.frame(Age,Temp,Soil,Comp_Stre)
9
10 #To compute cell totals
11 s1<-s2<-s3<-s4<-s5<-s6<-s7<-s8<-c(NULL)
12 for(i in 1:length(Comp_Stre)){
13   if(df1$Age[i]=="1" && df1$Temp[i]=="1" && df1$Soil
      [i]=="1"){
14     s1[i]<-df1$Comp_Stre[i]
15     x111<-sum(s1,na.rm=TRUE)
16   }
17   else if(df1$Age[i]=="1" && df1$Temp[i]=="2" && df1

```

```

    $Soil[i]=="1"){
18     s2[i]<-df1$Comp_Stre[i]
19     x121<-sum(s2,na.rm=TRUE)
20   }
21   else if(df1$Age[i]=="1" && df1$Temp[i]=="1" && df1
    $Soil[i]=="2"){
22     s3[i]<-df1$Comp_Stre[i]
23     x112<-sum(s3,na.rm=TRUE)
24   }
25   else if(df1$Age[i]=="2" && df1$Temp[i]=="1" && df1
    $Soil[i]=="1"){
26     s4[i]<-df1$Comp_Stre[i]
27     x211<-sum(s4,na.rm=TRUE)
28   }
29   else if(df1$Age[i]=="2" && df1$Temp[i]=="1" && df1
    $Soil[i]=="2"){
30     s5[i]<-df1$Comp_Stre[i]
31     x212<-sum(s5,na.rm=TRUE)
32   }
33   else if(df1$Age[i]=="1" && df1$Temp[i]=="2" && df1
    $Soil[i]=="2"){
34     s6[i]<-df1$Comp_Stre[i]
35     x122<-sum(s6,na.rm=TRUE)
36   }
37   else if(df1$Age[i]=="2" && df1$Temp[i]=="2" && df1
    $Soil[i]=="1"){
38     s7[i]<-df1$Comp_Stre[i]
39     x221<-sum(s7,na.rm=TRUE)
40   }
41   else if(df1$Age[i]=="2" && df1$Temp[i]=="2" && df1
    $Soil[i]=="2"){
42     s8[i]<-df1$Comp_Stre[i]
43     x222<-sum(s8,na.rm=TRUE)
44   }
45 }
46 }
47
48 Effect_contrasts<-c(NULL)

```

```

49 #To compute effect contrasts
50 for(j in 1:7){
51   if(j==1){
52     Effect_contrasts[j]=-x111+x211-x121+x221-x112+
      x212-x122+x222
53   }
54   else if(j==2){
55     Effect_contrasts[j]=-x111-x211+x121+x221-x112-
      x212+x122+x222
56   }
57   else if(j==3){
58     Effect_contrasts[j]=x111-x211-x121+x221+x112-
      x212-x122+x222
59   }
60   else if(j==4){
61     Effect_contrasts[j]=-x111-x211-x121-x221+x112+
      x212+x122+x222
62   }
63   else if(j==5){
64     Effect_contrasts[j]=x111-x211+x121-x221-x112+
      x212-x122+x222
65   }
66   else if(j==6){
67     Effect_contrasts[j]=x111+x211-x121-x221-x112-
      x212+x122+x222
68   }
69   else if(j==7){
70     Effect_contrasts[j]=-x111+x211+x121-x221+x112-
      x212-x122+x222
71   }
72 }
73
74 SS<-(Effect_contrasts^2)/(8*n)
75 df2<-data.frame(Effect_contrasts,SS)
76 cat("\nEffect contrast and SS values:\n")
77 print(df2)
78 SSsum<-sum(SS)
79

```



```

80 #To find sum of squares value
81 x2_ijkl<-0
82 X2<-0
83 for(k in 1:length(Comp_Stre)){
84     x2_ijkl<-x2_ijkl+(Comp_Stre[k]^2)
85 }
86 x2<-(x111+x112+x211+x121+x212+x221+x122+x222)^2/16
87 print(paste("Sum of square of data:",x2_ijkl))
88 print(paste("x...^2/16:",x2))
89
90 #To find SST value
91 SST<-x2_ijkl-x2
92 print(paste("SST:",SST))
93
94 #To find SSE value
95 SSE<-SST-SSsum
96 print(paste("SSE:",SSE))
97
98 #To display corresponding ANOVA tables
99 model1<-lm(Comp_Stre~Age:Temp:Soil)
100 model2<-lm(Comp_Stre~Age*Temp*Soil)
101 a<-aov(model1,data=df1)
102 b<-aov(model2,data=df1)
103 print(summary(a))
104 print(summary(b))

```

Chapter 12

Simple linear regression and correlation

R code Exa 12.1 The simple linear regression model

```
1 #Ex12.1 , Page 470
2
3 i<-c
   (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,2
4 xi<-c
   (0.40,0.42,0.48,0.51,0.57,0.60,0.70,0.75,0.75,0.78,0.84,0.95,0.99
5 yi<-c
   (1.02,1.21,0.88,0.98,1.52,1.83,1.50,1.80,1.74,1.63,2.00,2.80,2.48
6 data<-data.frame(xi,yi,row.names=i)
7 print(data)
8
9 par(mfrow=c(1,3))
10 plot(xi,yi,main="Scatterplot",xlab="Width of
    palprebal fissure",ylab="OSA")
11 dotchart(xi,labels=row.names(xi),main="Dot plot of x
    values")
```

```
12 dotchart(yi, labels=row.names(yi), main="Dot plot of y  
values")
```

R code Exa 12.2 The simple linear regression model

```
1 #Ex12.2, Page 471  
2  
3 x<-c  
    (7.01,7.11,7.12,7.24,7.94,7.94,8.04,8.05,8.07,8.90,8.94,8.95,8.97  
4 y<-c  
    (60,67,66,52,50,45,52,48,40,23,20,40,31,26,9,22,13,7)  
5 data1<-data.frame(x,y)  
6 print(data)  
7  
8 plot(x,y,main="Scatterplot",xlab="pH",ylab="Arsenic  
removed(%)")
```

R code Exa 12.3 A linear probabilistic model

```
1 #Ex12.3, Page 474  
2 #Answers may slightly vary due to rounding off of  
  probabilities  
3  
4 y<- function(x) (65-1.2*x)  
5 sd<-8  
6  
7 #P(time-to-failure exceeds 50 when applied stress is  
  20)  
8 x1<-20  
9 mu1<-y(x1)  
10 q<-50
```

```

11 P1<-1-pnorm(q,mu1,sd)
12 print(paste("Probability that the time-to-failure >50
    when applied stress=20):",P1))
13
14 #P(time-to-failure exceeds 50 when applied stress is
    25)
15 x2<-25
16 mu2<-y(x2)
17 q<-50
18 P2<-1-pnorm(q,mu2,sd)
19 print(paste("Probability that the time-to-failure >50
    when applied stress=25):",P2))
20
21 #Let Y1<-observation made with x=25 and Y2<-
    observation made with x=24
22 #Mean value
23 X1<-25
24 X2<-24
25 E<-y(X1)-y(X2)
26 #Variance
27 V<-sd^2+sd^2
28 SD<-sqrt(V)
29 #Probability that Y1 exceeds Y2
30 P3<-1-pnorm(0,E,V)
31 print(paste("P(Y1>Y2):",P3))

```

R code Exa 12.4 Estimating model parameters

```

1 #Ex12.4, Page 479
2 #Answers may slightly vary due to rounding off of
    values
3
4 x<-c
    (132.0,129.0,120.0,113.2,105.0,92.0,84.0,83.2,88.4,59.0,80.0,81.5

```

```

5 y<-c
  (46.0,48.0,51.0,52.1,54.0,52.0,59.0,58.7,61.6,64.0,61.4,54.6,58.8

6 data1<-data.frame(x,y)
7
8 #To display regression model
9 model<-lm(y~x,data=data1)
10 cat("Regression model of data:\n")
11 a<-summary(model)
12 print(a)
13
14 #To display the scatterplot
15 plot(x,y,main="Scatterplot with least square line
  superimposed",xlab="iod val",ylab="cet num")
16 abline(a$coefficients[1],a$coefficients[2])

```

R code Exa 12.5 Estimating model parameters

```

1 #Ex12.5, Page 480
2
3 y<- function(x) {75.212-0.2094*x}
4 #If iodine value 100 is selected...
5 x<-100
6 print(paste("Point prediction for the resulting
  cetane number:",round(y(x),digits=2)))

```

R code Exa 12.6 Estimating model parameters

```

1 #Ex12.6, Page 482
2 #Answers may slightly vary due to rounding off of
  values
3

```

```

4  x<-c
    (125.3,98.2,201.4,147.3,145.9,124.7,112.2,120.2,161.2,178.9,159.5
5  y<-c
    (77.9,76.8,81.5,79.8,78.2,78.3,77.5,77.0,80.1,80.2,79.9,79.0,76.7

6  data1<-data.frame(x,y)
7
8  model<-lm(y~x)
9
10 #To display final table of predicted values and
    residuals
11 yb<-predict(model,data1)
12 Residual<-resid(model)
13 data1<-data.frame(x,y,yb,Residual)
14 colnames(data1)[colnames(data1)=="x"]<-"Filtrate"
15 colnames(data1)[colnames(data1)=="y"]<-"Moistcon"
16 colnames(data1)[colnames(data1)=="yb"]<-"Fit"
17 print(data1)

```

R code Exa 12.7 Estimating model parameters

```

1 #Ex12.7, Page 483
2 #Answers may vary slightly due to rounding off of
    values
3
4 #From Ex12.6
5 x<-c
    (125.3,98.2,201.4,147.3,145.9,124.7,112.2,120.2,161.2,178.9,159.5
6 y<-c
    (77.9,76.8,81.5,79.8,78.2,78.3,77.5,77.0,80.1,80.2,79.9,79.0,76.7

7 data1<-data.frame(x,y)
8

```

```

9  model<-lm(y~x)
10 Residual<-resid(model)
11 n<-length(Residual)
12
13 SSE<-sum(Residual^2)
14 print(paste("SSE:",SSE))
15
16 #To find variance(sigma^2)
17 s2<-SSE/(n-2)
18 print(paste("Estimate of variance:",s2))
19
20 #To find standard deviation(sigma)
21 sd<-sqrt(s2)
22 print(paste("Estimate of standard deviation:",sd))
23
24 #To display ANOVA table
25 a<-aov(model,data=data1)
26 cat("ANOVA table:\n")
27 print(summary(a))

```

R code Exa 12.8 Estimating model parameters

```

1  #Ex12.8, Page 484
2  #Answers may vary slightly due to rounding off of
   values
3
4  x<-c(12,30,36,40,45,57,62,67,71,78,93,94,100,105)
5  y<-c
   (3.3,3.2,3.4,3.0,2.8,2.9,2.7,2.6,2.5,2.6,2.2,2.0,2.3,2.1)
6  data1<-data.frame(x,y)
7
8  #To display ANOVA table
9  model<-lm(y~x)
10 print(summary(model))

```

```

11 a<-aov(model,data=data1)
12 cat("ANOVA table:\n")
13 print(summary(a))

```

R code Exa 12.9 Coefficient of determination

```

1 #Ex12.9, Page 486
2 #Answers may vary slightly due to rounding off of
  values
3
4 #From Ex12.4
5 x<-c
  (132.0,129.0,120.0,113.2,105.0,92.0,84.0,83.2,88.4,59.0,80.0,81.5
6
7 y<-c
  (46.0,48.0,51.0,52.1,54.0,52.0,59.0,58.7,61.6,64.0,61.4,54.6,58.8
8
9 data1<-data.frame(x,y)
10
11 n<-14
12 beta0<-75.212432
13 beta1<-(-0.20938742)
14 yi<-779.2
15 xiyi<-71347.30
16 yi2<-43745.22
17
18 #To find SST and SSE
19 SST<-yi2-(yi^2)/n
20 print(paste("SST:",SST))
21 SSE<-yi2-beta0*yi-beta1*xiyi
22 print(paste("SSE:",SSE))
23
24 #Coefficient of determination
25 r2<-1-(SSE/SST)
26 print(paste("Coefficient of determination:",r2))

```



```

25
26 #To create regression model using ANOVA table
27 a<-lm(y~x)
28 print(summary(a))
29 a1<-aov(a,data=data1)
30 print(summary(a1))

```

R code Exa 12.11 Confidence interval for beta1

```

1 #Ex12.11, Page 494
2 #Answers may vary slightly due to rounding off of
  values
3
4 x<-c
  (5.7,6.8,9.6,10.0,10.7,12.6,14.4,15.0,15.3,16.2,17.8,18.7,19.7,20.7,21.6,22.5,23.4,24.3,25.2,26.1,27.0,27.9,28.8,29.7,30.6,31.5,32.4,33.3,34.2,35.1,36.0,36.9,37.8,38.7,39.6,40.5,41.4,42.3,43.2,44.1,45.0,45.9,46.8,47.7,48.6,49.5,50.4,51.3,52.2,53.1,54.0,54.9,55.8,56.7,57.6,58.5,59.4,60.3,61.2,62.1,63.0,63.9,64.8,65.7,66.6,67.5,68.4,69.3,70.2,71.1,72.0,72.9,73.8,74.7,75.6,76.5,77.4,78.3,79.2,80.1,81.0,81.9,82.8,83.7,84.6,85.5,86.4,87.3,88.2,89.1,90.0,90.9,91.8,92.7,93.6,94.5,95.4,96.3,97.2,98.1,99.0,100.0)
5 y<-c
  (119.0,121.3,118.2,124.0,112.3,114.1,112.2,115.1,111.3,107.2,108.9,110.5,112.7,114.9,117.1,119.3,121.5,123.7,125.9,128.1,130.3,132.5,134.7,136.9,139.1,141.3,143.5,145.7,147.9,150.1,152.3,154.5,156.7,158.9,161.1,163.3,165.5,167.7,169.9,172.1,174.3,176.5,178.7,180.9,183.1,185.3,187.5,189.7,191.9,194.1,196.3,198.5,200.7,202.9,205.1,207.3,209.5,211.7,213.9,216.1,218.3,220.5,222.7,224.9,227.1,229.3,231.5,233.7,235.9,238.1,240.3,242.5,244.7,246.9,249.1,251.3,253.5,255.7,257.9,260.1,262.3,264.5,266.7,268.9,271.1,273.3,275.5,277.7,279.9,282.1,284.3,286.5,288.7,290.9,293.1,295.3,297.5,299.7,301.9,304.1,306.3,308.5,310.7,312.9,315.1,317.3,319.5,321.7,323.9,326.1,328.3,330.5,332.7,334.9,337.1,339.3,341.5,343.7,345.9,348.1,350.3,352.5,354.7,356.9,359.1,361.3,363.5,365.7,367.9,370.1,372.3,374.5,376.7,378.9,381.1,383.3,385.5,387.7,389.9,392.1,394.3,396.5,398.7,400.9,403.1,405.3,407.5,409.7,411.9,414.1,416.3,418.5,420.7,422.9,425.1,427.3,429.5,431.7,433.9,436.1,438.3,440.5,442.7,444.9,447.1,449.3,451.5,453.7,455.9,458.1,460.3,462.5,464.7,466.9,469.1,471.3,473.5,475.7,477.9,480.1,482.3,484.5,486.7,488.9,491.1,493.3,495.5,497.7,499.9,502.1,504.3,506.5,508.7,510.9,513.1,515.3,517.5,519.7,521.9,524.1,526.3,528.5,530.7,532.9,535.1,537.3,539.5,541.7,543.9,546.1,548.3,550.5,552.7,554.9,557.1,559.3,561.5,563.7,565.9,568.1,570.3,572.5,574.7,576.9,579.1,581.3,583.5,585.7,587.9,590.1,592.3,594.5,596.7,598.9,601.1,603.3,605.5,607.7,609.9,612.1,614.3,616.5,618.7,620.9,623.1,625.3,627.5,629.7,631.9,634.1,636.3,638.5,640.7,642.9,645.1,647.3,649.5,651.7,653.9,656.1,658.3,660.5,662.7,664.9,667.1,669.3,671.5,673.7,675.9,678.1,680.3,682.5,684.7,686.9,689.1,691.3,693.5,695.7,697.9,700.1,702.3,704.5,706.7,708.9,711.1,713.3,715.5,717.7,719.9,722.1,724.3,726.5,728.7,730.9,733.1,735.3,737.5,739.7,741.9,744.1,746.3,748.5,750.7,752.9,755.1,757.3,759.5,761.7,763.9,766.1,768.3,770.5,772.7,774.9,777.1,779.3,781.5,783.7,785.9,788.1,790.3,792.5,794.7,796.9,799.1,801.3,803.5,805.7,807.9,810.1,812.3,814.5,816.7,818.9,821.1,823.3,825.5,827.7,829.9,832.1,834.3,836.5,838.7,840.9,843.1,845.3,847.5,849.7,851.9,854.1,856.3,858.5,860.7,862.9,865.1,867.3,869.5,871.7,873.9,876.1,878.3,880.5,882.7,884.9,887.1,889.3,891.5,893.7,895.9,898.1,900.3,902.5,904.7,906.9,909.1,911.3,913.5,915.7,917.9,920.1,922.3,924.5,926.7,928.9,931.1,933.3,935.5,937.7,939.9,942.1,944.3,946.5,948.7,950.9,953.1,955.3,957.5,959.7,961.9,964.1,966.3,968.5,970.7,972.9,975.1,977.3,979.5,981.7,983.9,986.1,988.3,990.5,992.7,994.9,997.1,999.3,1001.5,1003.7,1005.9,1008.1,1010.3,1012.5,1014.7,1016.9,1019.1,1021.3,1023.5,1025.7,1027.9,1030.1,1032.3,1034.5,1036.7,1038.9,1041.1,1043.3,1045.5,1047.7,1049.9,1052.1,1054.3,1056.5,1058.7,1060.9,1063.1,1065.3,1067.5,1069.7,1071.9,1074.1,1076.3,1078.5,1080.7,1082.9,1085.1,1087.3,1089.5,1091.7,1093.9,1096.1,1098.3,1100.5,1102.7,1104.9,1107.1,1109.3,1111.5,1113.7,1115.9,1118.1,1120.3,1122.5,1124.7,1126.9,1129.1,1131.3,1133.5,1135.7,1137.9,1140.1,1142.3,1144.5,1146.7,1148.9,1151.1,1153.3,1155.5,1157.7,1159.9,1162.1,1164.3,1166.5,1168.7,1170.9,1173.1,1175.3,1177.5,1179.7,1181.9,1184.1,1186.3,1188.5,1190.7,1192.9,1195.1,1197.3,1199.5,1201.7,1203.9,1206.1,1208.3,1210.5,1212.7,1214.9,1217.1,1219.3,1221.5,1223.7,1225.9,1228.1,1230.3,1232.5,1234.7,1236.9,1239.1,1241.3,1243.5,1245.7,1247.9,1250.1,1252.3,1254.5,1256.7,1258.9,1261.1,1263.3,1265.5,1267.7,1269.9,1272.1,1274.3,1276.5,1278.7,1280.9,1283.1,1285.3,1287.5,1289.7,1291.9,1294.1,1296.3,1298.5,1300.7,1302.9,1305.1,1307.3,1309.5,1311.7,1313.9,1316.1,1318.3,1320.5,1322.7,1324.9,1327.1,1329.3,1331.5,1333.7,1335.9,1338.1,1340.3,1342.5,1344.7,1346.9,1349.1,1351.3,1353.5,1355.7,1357.9,1360.1,1362.3,1364.5,1366.7,1368.9,1371.1,1373.3,1375.5,1377.7,1379.9,1382.1,1384.3,1386.5,1388.7,1390.9,1393.1,1395.3,1397.5,1399.7,1401.9,1404.1,1406.3,1408.5,1410.7,1412.9,1415.1,1417.3,1419.5,1421.7,1423.9,1426.1,1428.3,1430.5,1432.7,1434.9,1437.1,1439.3,1441.5,1443.7,1445.9,1448.1,1450.3,1452.5,1454.7,1456.9,1459.1,1461.3,1463.5,1465.7,1467.9,1470.1,1472.3,1474.5,1476.7,1478.9,1481.1,1483.3,1485.5,1487.7,1489.9,1492.1,1494.3,1496.5,1498.7,1500.9,1503.1,1505.3,1507.5,1509.7,1511.9,1514.1,1516.3,1518.5,1520.7,1522.9,1525.1,1527.3,1529.5,1531.7,1533.9,1536.1,1538.3,1540.5,1542.7,1544.9,1547.1,1549.3,1551.5,1553.7,1555.9,1558.1,1560.3,1562.5,1564.7,1566.9,1569.1,1571.3,1573.5,1575.7,1577.9,1580.1,1582.3,1584.5,1586.7,1588.9,1591.1,1593.3,1595.5,1597.7,1599.9,1602.1,1604.3,1606.5,1608.7,1610.9,1613.1,1615.3,1617.5,1619.7,1621.9,1624.1,1626.3,1628.5,1630.7,1632.9,1635.1,1637.3,1639.5,1641.7,1643.9,1646.1,1648.3,1650.5,1652.7,1654.9,1657.1,1659.3,1661.5,1663.7,1665.9,1668.1,1670.3,1672.5,1674.7,1676.9,1679.1,1681.3,1683.5,1685.7,1687.9,1690.1,1692.3,1694.5,1696.7,1698.9,1701.1,1703.3,1705.5,1707.7,1709.9,1712.1,1714.3,1716.5,1718.7,1720.9,1723.1,1725.3,1727.5,1729.7,1731.9,1734.1,1736.3,1738.5,1740.7,1742.9,1745.1,1747.3,1749.5,1751.7,1753.9,1756.1,1758.3,1760.5,1762.7,1764.9,1767.1,1769.3,1771.5,1773.7,1775.9,1778.1,1780.3,1782.5,1784.7,1786.9,1789.1,1791.3,1793.5,1795.7,1797.9,1800.1,1802.3,1804.5,1806.7,1808.9,1811.1,1813.3,1815.5,1817.7,1819.9,1822.1,1824.3,1826.5,1828.7,1830.9,1833.1,1835.3,1837.5,1839.7,1841.9,1844.1,1846.3,1848.5,1850.7,1852.9,1855.1,1857.3,1859.5,1861.7,1863.9,1866.1,1868.3,1870.5,1872.7,1874.9,1877.1,1879.3,1881.5,1883.7,1885.9,1888.1,1890.3,1892.5,1894.7,1896.9,1899.1,1901.3,1903.5,1905.7,1907.9,1910.1,1912.3,1914.5,1916.7,1918.9,1921.1,1923.3,1925.5,1927.7,1929.9,1932.1,1934.3,1936.5,1938.7,1940.9,1943.1,1945.3,1947.5,1949.7,1951.9,1954.1,1956.3,1958.5,1960.7,1962.9,1965.1,1967.3,1969.5,1971.7,1973.9,1976.1,1978.3,1980.5,1982.7,1984.9,1987.1,1989.3,1991.5,1993.7,1995.9,1998.1,2000.3,2002.5,2004.7,2006.9,2009.1,2011.3,2013.5,2015.7,2017.9,2020.1,2022.3,2024.5,2026.7,2028.9,2031.1,2033.3,2035.5,2037.7,2039.9,2042.1,2044.3,2046.5,2048.7,2050.9,2053.1,2055.3,2057.5,2059.7,2061.9,2064.1,2066.3,2068.5,2070.7,2072.9,2075.1,2077.3,2079.5,2081.7,2083.9,2086.1,2088.3,2090.5,2092.7,2094.9,2097.1,2099.3,2101.5,2103.7,2105.9,2108.1,2110.3,2112.5,2114.7,2116.9,2119.1,2121.3,2123.5,2125.7,2127.9,2130.1,2132.3,2134.5,2136.7,2138.9,2141.1,2143.3,2145.5,2147.7,2149.9,2152.1,2154.3,2156.5,2158.7,2160.9,2163.1,2165.3,2167.5,2169.7,2171.9,2174.1,2176.3,2178.5,2180.7,2182.9,2185.1,2187.3,2189.5,2191.7,2193.9,2196.1,2198.3,2200.5,2202.7,2204.9,2207.1,2209.3,2211.5,2213.7,2215.9,2218.1,2220.3,2222.5,2224.7,2226.9,2229.1,2231.3,2233.5,2235.7,2237.9,2240.1,2242.3,2244.5,2246.7,2248.9,2251.1,2253.3,2255.5,2257.7,2259.9,2262.1,2264.3,2266.5,2268.7,2270.9,2273.1,2275.3,2277.5,2279.7,2281.9,2284.1,2286.3,2288.5,2290.7,2292.9,2295.1,2297.3,2299.5,2301.7,2303.9,2306.1,2308.3,2310.5,2312.7,2314.9,2317.1,2319.3,2321.5,2323.7,2325.9,2328.1,2330.3,2332.5,2334.7,2336.9,2339.1,2341.3,2343.5,2345.7,2347.9,2350.1,2352.3,2354.5,2356.7,2358.9,2361.1,2363.3,2365.5,2367.7,2369.9,2372.1,2374.3,2376.5,2378.7,2380.9,2383.1,2385.3,2387.5,2389.7,2391.9,2394.1,2396.3,2398.5,2400.7,2402.9,2405.1,2407.3,2409.5,2411.7,2413.9,2416.1,2418.3,2420.5,2422.7,2424.9,2427.1,2429.3,2431.5,2433.7,2435.9,2438.1,2440.3,2442.5,2444.7,2446.9,2449.1,2451.3,2453.5,2455.7,2457.9,2460.1,2462.3,2464.5,2466.7,2468.9,2471.1,2473.3,2475.5,2477.7,2479.9,2482.1,2484.3,2486.5,2488.7,2490.9,2493.1,2495.3,2497.5,2499.7,2501.9,2504.1,2506.3,2508.5,2510.7,2512.9,2515.1,2517.3,2519.5,2521.7,2523.9,2526.1,2528.3,2530.5,2532.7,2534.9,2537.1,2539.3,2541.5,2543.7,2545.9,2548.1,2550.3,2552.5,2554.7,2556.9,2559.1,2561.3,2563.5,2565.7,2567.9,2570.1,2572.3,2574.5,2576.7,2578.9,2581.1,2583.3,2585.5,2587.7,2589.9,2592.1,2594.3,2596.5,2598.7,2600.9,2603.1,2605.3,2607.5,2609.7,2611.9,2614.1,2616.3,2618.5,2620.7,2622.9,2625.1,2627.3,2629.5,2631.7,2633.9,2636.1,2638.3,2640.5,2642.7,2644.9,2647.1,2649.3,2651.5,2653.7,2655.9,2658.1,2660.3,2662.5,2664.7,2666.9,2669.1,2671.3,2673.5,2675.7,2677.9,2680.1,2682.3,2684.5,2686.7,2688.9,2691.1,2693.3,2695.5,2697.7,2699.9,2702.1,2704.3,2706.5,2708.7,2710.9,2713.1,2715.3,2717.5,2719.7,2721.9,2724.1,2726.3,2728.5,2730.7,2732.9,2735.1,2737.3,2739.5,2741.7,2743.9,2746.1,2748.3,2750.5,2752.7,2754.9,2757.1,2759.3,2761.5,2763.7,2765.9,2768.1,2770.3,2772.5,2774.7,2776.9,2779.1,2781.3,2783.5,2785.7,2787.9,2790.1,2792.3,2794.5,2796.7,2798.9,2801.1,2803.3,2805.5,2807.7,2809.9,2812.1,2814.3,2816.5,2818.7,2820.9,2823.1,2825.3,2827.5,2829.7,2831.9,2834.1,2836.3,2838.5,2840.7,2842.9,2845.1,2847.3,2849.5,2851.7,2853.9,2856.1,2858.3,2860.5,2862.7,2864.9,2867.1,2869.3,2871.5,2873.7,2875.9,2878.1,2880.3,2882.5,2884.7,2886.9,2889.1,2891.3,2893.5,2895.7,2897.9,2900.1,2902.3,2904.5,2906.7,2908.9,2911.1,2913.3,2915.5,2917.7,2919.9,2922.1,2924.3,2926.5,2928.7,2930.9,2933.1,2935.3,2937.5,2939.7,2941.9,2944.1,2946.3,2948.5,2950.7,2952.9,2955.1,2957.3,2959.5,2961.7,2963.9,2966.1,2968.3,2970.5,2972.7,2974.9,2977.1,2979.3,2981.5,2983.7,2985.9,2988.1,2990.3,2992.5,2994.7,2996.9,2999.1,3001.3,3003.5,3005.7,3007.9,3010.1,3012.3,3014.5,3016.7,3018.9,3021.1,3023.3,3025.5,3027.7,3029.9,3032.1,3034.3,3036.5,3038.7,3040.9,3043.1,3045.3,3047.5,3049.7,3051.9,3054.1,3056.3,3058.5,3060.7,3062.9,3065.1,3067.3,3069.5,3071.7,3073.9,3076.1,3078.3,3080.5,3082.7,3084.9,3087.1,3089.3,3091.5,3093.7,3095.9,3098.1,3100.3,3102.5,3104.7,3106.9,3109.1,3111.3,3113.5,3115.7,3117.9,3120.1,3122.3,3124.5,3126.7,3128.9,3131.1,3133.3,3135.5,3137.7,3139.9,3142.1,3144.3,3146.5,3148.7,3150.9,3153.1,3155.3,3157.5,3159.7,3161.9,3164.1,3166.3,3168.5,3170.7,3172.9,3175.1,3177.3,3179.5,3181.7,3183.9,3186.1,3188.3,3190.5,3192.7,3194.9,3197.1,3199.3,3201.5,3203.7,3205.9,3208.1,3210.3,3212.5,3214.7,3216.9,3219.1,3221.3,3223.5,3225.7,3227.9,3230.1,3232.3,3234.5,3236.7,3238.9,3241.1,3243.3,3245.5,3247.7,3249.9,3252.1,3254.3,3256.5,3258.7,3260.9,3263.1,3265.3,3267.5,3269.7,3271.9,3274.1,3276.3,3278.5,3280.7,3282.9,3285.1,3287.3,3289.5,3291.7,3293.9,3296.1,3298.3,3300.5,3302.7,3304.9,3307.1,3309.3,3311.5,3313.7,3315.9,3318.1,3320.3,3322.5,3324.7,3326.9,3329.1,3331.3,3333.5,3335.7,3337.9,3340.1,3342.3,3344.5,3346.7,3348.9,3351.1,3353.3,3355.5,3357.7,3359.9,3362.1,3364.3,3366.5,3368.7,3370.9,3373.1,3375.3,3377.5,3379.7,3381.9,3384.1,3386.3,3388.5,3390.7,3392.9,3395.1,3397.3,3399.5,3401.7,3403.9,3406.1,3408.3,3410.5,3412.7,3414.9,3417.1,3419.3,3421.5,3423.7,3425.9,3428.1,3430.3,3432.5,3434.7,3436.9,3439.1,3441.3,3443.5,3445.7,3447.9,3450.1,3452.3,3454.5,3456.7,3458.9,3461.1,3463.3,3465.5,3467.7,3469.9,3472.1,3474.3,3476.5,3478.7,3480.9,3483.1,3485.3,3487.5,3489.7,3491.9,3494.1,3496.3,3498.5,3500.7,3502.9,3505.1,3507.3,3509.5,3511.7,3513.9,3516.1,3518.3,3520.5,3522.7,3524.9,3527.1,3529.3,3531.5,3533.7,3535.9,3538.1,3540.3,3542.5,3544.7,3546.9,3549.1,3551.3,3553.5,3555.7,3557.9,3560.1,3562.3,3564.5,3566.7,3568.9,3571.1,3573.3,3575.5,3577.7,3579.9,3582.1,3584.3,3586.5,3588.7,3590.9,3593.1,3595.3,3597.5,3599.7,3601.9,3604.1,3606.3,3608.5,3610.7,3612.9,3615.1,3617.3,3619.5,3621.7,3623.9,3626.1,3628.3,3630.5,3632.7,3634.9,3637.1,3639.3,3641.5,3643.7,3645.9,3648.1,3650.3,3652.5,3654.7,3656.9,3659.1,3661.3,3663.5,3665.7,3667.9,3670.1,3672.3,3674.5,3676.7,3678.9,3681.1,3683.3,3685.5,3687.7,3689.9,3692.1,3694.3,3696.5,3698.7,3700.9,3703.1,3705.3,3707.5,3709.7,3711.9,3714.1,3716.3,3718.5,3720.7,3722.9,3725.1,3727.3,3729.5,3731.7,3733.9,3736.1,3738.3,3740.5,3742.7,3744.9,3747.1,3749.3,3751.5,3753.7,3755.9,3758.1,3760.3,3762.5,3764.7,3766.9,3769.1,3771.3,3773.5,3775.7,3777.9,3780.1,3782.3,3784.5,3786.7,3788.9,3791.1,3793.3,3795.5,3797.7,3799.9,3802.1,3804.3,3806.5,3808.7,3810.9,3813.1,3815.3,3817.5,3819.7,3821.9,3824.1,3826.3,3828.5,3830.7,3832.9,3835.1,3837.3,3839.5,3841.7,3843.9,3846.1,3848.3,3850.5,3852.7,3854.9,3857.1,3859.3,3861.5,3863.7,3865.9,3868.1,3870.3,3872.5,3874.7,
```

```

21 print(paste("Sxy:", Sxy))
22
23 #To find beta1 and beta0
24 xbar<-mean(x)
25 ybar<-mean(y)
26 beta1<-Sxy/Sxx
27 beta0<-ybar-beta1*xbar
28 print(paste("Beta1:", beta1))
29 print(paste("Beta0:", beta0))
30
31 #To find SSE
32 SSE<-yi2-beta0*yi-beta1*xiyi
33 print(paste("SSE:", SSE))
34
35 #To find SST
36 SST<-yi2-yi^2/n
37 print(paste("SST:", SST))
38
39 #To find r^2
40 r2<-1-(SSE/SST)
41 print(paste("r^2:", r2))
42
43 I<-15
44 J<-2
45 df<-I-J
46
47 #s^2
48 s2<-SSE/df
49 s<-sqrt(s2)
50
51 #Estimated standard deviation of beta1
52 sb<-s/sqrt(Sxx)
53 print(paste("Estimated standard deviation of beta1:",
54             ,sb))
54
55 alpha<-0.05
56 t<-qt(1-alpha/2, df)
57

```

```

58 #To find confidence intervals
59 l<-beta1-t*sb
60 r<-beta1+t*sb
61 print(paste("Left endpoint of confidence interval:",
62             1))
63 print(paste("Right endpoint of confidence interval:",
64             ,r))
65
66 #To display analysis of variance and parameter
67 estimates
68 model<-lm(y~x)
69 cat("\nParameter estimates\n")
70 print(summary(model))
71 cat("\nANOVA table:\n")
72 print(summary(aov(model)))
73 a<-predict(model)
74 print(paste("Predict values:"))
75 print(a)
76 print(paste("Residuals:"))
77 print(resid(model))

```

R code Exa 12.12 Hypothesis testing procedures

```

1 #Ex12.12, Page 496
2 #Answers may vary slightly due to rounding off of
3 values
4
5 x<-c
6   (42.2,42.6,43.3,43.5,43.7,44.1,44.9,45.3,45.7,45.7,45.9,46.0,46.2
7
8 y<-c
9   (44,44,44,45,45,46,46,46,47,48,48,48,47,48,48,49,49,49)
10
11 data1<-data.frame(x,y)
12

```

```

8 model<-lm(y~x,data=data1)
9 cat("Regression model:\n")
10 print(summary(model))
11
12 #To display ANOVA table
13 a<-aov(model)
14 cat("ANOVA table:\n")
15 print(summary(a))

```

R code Exa 12.13 Inferences concerning mean

```

1 #Ex12.13, Page 501
2 #Answers may vary slightly due to rounding off of
  values
3
4 library(ggplot2)
5
6 x<-c
  (8.0,15.0,16.5,20.0,20.0,27.5,30.0,30.0,35.0,38.0,40.0,45.0,50.0,
7
7 y<-c
  (22.8,27.2,23.7,17.1,21.5,18.6,16.1,23.4,13.4,19.5,12.4,13.2,11.4
8
8 data1<-data.frame(x,y)
9
10 #To plot data using ggplot
11 ggplot(data=data1,mapping=aes(x,y))+geom_point(size
  =2)+geom_smooth(method=lm)
12
13 #To display linear regression model
14 model<-lm(y~x)
15 print(summary(model))
16
17 #To display analysis of variance
18 cat("ANOVA table:\n")

```

```

19 a<-aov(model,data=data1)
20 print(summary(a))
21
22 #To display confidence and prediction intervals
23 d<-data.frame(x=c(35,45))
24 cat("Confidence intervals:\n")
25 print(predict(model,newdata=d,interval="confidence",
    level=0.95))
26 cat("Prediction intervals:\n")
27 print(predict(model,newdata=d,interval="predict",
    level=0.95))

```

R code Exa 12.14 Prediction interval for a future value of Y

```

1 #Ex12.14, Page 505
2 #Answers may vary slightly due to rounding off of
  values
3
4 ycap<-13.79
5 sy<-0.7582
6 s<-2.8640
7 n<-18
8 df<-n-2
9 t<-2.120
10
11 #To find confidence intervals
12 l<-ycap-t*sqrt(s^2+sy^2)
13 r<-ycap+t*sqrt(s^2+sy^2)
14 print(paste("Left endpoint of confidence interval:",
    l))
15 print(paste("Right endpoint of confidence interval:",
    r))

```

R code Exa 12.15 Correlation

```
1 #Ex12.15, Page 509
2 #Answers may vary slightly due to rounding off of
  values
3
4 x<-c(2.4,3.4,4.6,3.7,2.2,3.3,4.0,2.1)
5 y<-c(1.33,2.12,1.80,1.65,2.00,1.76,2.11,1.63)
6
7 #To find sample correlation coefficient
8 r<-cor(x,y)
9 print(paste("Sample correlation coefficient ,r:",r))
```

R code Exa 12.16 Inferences about the population correlation coefficient

```
1 #Ex12.16, Page 511
2 #Answers may vary slightly due to rounding off of
  values
3
4 x<-c
  (0.066,0.088,0.120,0.050,0.162,0.186,0.057,0.100,0.112,0.055,0.154)
5 y<-c
  (4.6,11.6,9.5,6.3,13.8,15.4,2.5,11.8,8.0,7.0,20.6,16.6,9.2,17.9,21.5)
6
7 #To find r(point estimate of population correlation
  coefficient)
8 r<-cor(x,y)
9 print(paste("r:",r))
```

R code Exa 12.17 Inferences about the population correlation coefficient

```

1 #Ex12.17, Page 513
2 #Answers may vary slightly due to rounding off of
  values
3
4 r<-0.29
5 n<-45
6 #To find the test statistic
7 t<-r*sqrt(n-2)/sqrt(1-r^2)
8 print(paste("t:",round(t,digits=0)))
9
10 #P value for two tailed test
11 p<-2*pt(-abs(t),df=n-1)
12 print(p)

```

R code Exa 12.18 Other inferences concerning rho

```

1 #Ex12.18, Page 515
2 #Answers may vary slightly due to rounding off of
  values
3
4 x<-c
  (55.10,44.83,46.32,51.10,49.89,45.20,48.18,46.70,54.31,41.50,47.50)
5 y<-c
  (49.10,31.20,32.80,42.60,42.50,32.70,36.21,40.40,37.42,30.80,35.30)
6
7 #To find r
8 r<-cor(x,y)
9 print(paste("Correlation coefficient , r:",r))
10
11 #H0: rho=0.5
12 #Ha: rho>0.5
13 rho<-0.5
14

```

```

15 #To find v
16 v<-0.5*log((1+r)/(1-r))
17 print(paste("v:",v))
18
19 muv<-0.5*log((1+rho)/(1-rho))
20 print(paste("muv:",muv))
21
22 z<-(v-muv)*sqrt(length(x)-3)
23 print(paste("z:",z))
24
25 #To find P value
26 p<-pnorm(z,lower.tail=FALSE)
27 print(paste("p value:",p))
28
29 print(paste("H0 is rejected..."))

```

R code Exa 12.19 Other inferences concerning rho

```

1 #Ex12.19, Page 516
2 #Answers may vary slightly due to rounding off of
  values
3
4 xi<-285.90
5 xi2<-4409.55
6 yi<-690.30
7 yi2<-29040.29
8 xiyi<-10818.56
9 r<-0.733
10 v<-0.935
11 n<-20
12 alpha<-0.05
13
14 #To find z
15 zalp2<-qnorm(1-alpha/2)
16

```



```

17 #95% interval for muv
18 c1<-v-zalpha2/sqrt(n-3)
19 c2<-v+zalpha2/sqrt(n-3)
20 print(paste("Left endpoint of CI for muv:",c1))
21 print(paste("Right endpoint of CI for muv:",c2))
22
23 #95% interval for rho
24 l<-(exp(1)^(2*c1)-1)/(exp(1)^(2*c1)+1)
25 r<-(exp(1)^(2*c2)-1)/(exp(1)^(2*c2)+1)
26 print(paste("Left endpoint of CI for rho:",l))
27 print(paste("Right endpoint of CI for rho:",r))

```

Chapter 13

Nonlinear and multiple regression

R code Exa 13.2 Diagnostic plots

```
1 #Ex13.2 , Page 526
2
3 x<-c
   (100,125,125,150,150,200,200,250,250,300,300,350,400,400)
4 y<-c
   (150,140,180,210,190,320,280,400,430,440,390,600,610,670)
5 y_cap<-(-45.55)+(1.71*x)
6 m1<-lm(y~x)
7 ei<-round(resid(m1),digits=1)
8 ei_std<-round(rstandard(m1),digits=2)
9 df<-data.frame(x,y,y_cap,ei,ei_std)
10 print(df)
11
12 par(mfrow=c(3,2))
13 plot(x,y,main="y vs. x")
14 abline(m1)
15 plot(y_cap,ei_std,main="Standardized residuals vs. y")
```

```

    _cap")
16 abline(0,0)
17 plot(y,y_cap,main="y_cap vs. y")
18 abline(lm(y_cap~y))
19 plot(x,ei_std,main="Standardized residuals vs. x")
20 abline(0,0)
21 qqnorm(ei_std,xlab="z percentile",ylab="e*",main="
    Normal probability plot")

```

R code Exa 13.4 Regression with transformed variables

```

1 #Ex13.4, Page 534
2 #Answers may vary slightly due to rounding off of
  values
3
4 x<-c(2,10,20,30,40,50,60,70,80,90,100)
5 y<-c(408,274,196,137,90,78,51,40,30,22,15)
6 y_dash<-log(y,base=exp(1))
7 m1<-lm(y~x)
8 m2<-lm(y_dash~x)
9 ei_std1<-rstandard(m1)
10 ei_std2<-rstandard(m2)
11
12 par(mfrow=c(2,2))
13 plot(x,y,main="Scatterplot")
14 abline(m1)
15 plot(x,ei_std1,xlab="x",ylab="e*",main="Residual
    plot from linear regression for the data")
16 abline(0,0)
17
18 #To find the coefficients
19 beta0<-m2$coefficients[1]
20 beta1<-round(m2$coefficients[2],digits=4)
21
22 y_cap<-(exp(1)^(beta0))*(exp(1)^(beta1*x))

```

```

23
24 df<-data.frame(x,y,y_dash,y_cap)
25 print(df)
26
27 plot(x,ei_std2,ylab="e*",main="Standardized
    residuals(after transforming) vs. x")
28 abline(0,0)
29 plot(y,y_cap,main="ycap vs. y")
30 abline(lm(y_cap~y))

```

R code Exa 13.7 Polynomial regression

```

1 #Ex13.7, Page 544
2
3 Thickness<-c
    (220,220,220,220,370,370,370,370,440,440,440,440,680,680,680,680,
4
5 Strength<-c
    (24.0,22.0,19.1,15.5,26.3,24.6,23.1,21.2,25.2,24.0,21.7,19.2,17.0
6
7 df<-data.frame(Thickness,Strength)
8 cat("Regression model:\n")
9 relation<-lm(Strength~Thickness+I(Thickness^2))
10 print(summary(relation))
11
12 cat("Analysis of variance:\n")
13 model<-lm(Strength~poly(Thickness,2))
14 a<-aov(model,data=df)
15 print(summary(a))
16
17 #To find confidence and prediction intervals
18 cat("\nWhen thickness=500\n")
19 cat("95% CI:\n")
20 print(predict(relation,newdata=data.frame(Thickness

```

```

    =500), interval="confidence"))
20 cat("95% PI:\n")
21 print(predict(relation, newdata=data.frame(Thickness
    =500), interval="prediction"))
22
23 cat("\nWhen thickness=800\n")
24 cat("95% CI:\n")
25 print(predict(relation, newdata=data.frame(Thickness
    =800), interval="confidence"))
26 cat("95% PI:\n")
27 print(predict(relation, newdata=data.frame(Thickness
    =800), interval="prediction"))

```

R code Exa 13.8 Estimating parameters

```

1 #Ex13.8, Page 546
2 #Answers may vary slightly due to rounding off of
  values
3
4 #From Ex13.7
5 Thickness<-c
  (220,220,220,220,370,370,370,370,440,440,440,440,680,680,680,680,
6
7 Strength<-c
  (24.0,22.0,19.1,15.5,26.3,24.6,23.1,21.2,25.2,24.0,21.7,19.2,17.0
8
9 df<-data.frame(Thickness,Strength)
10 relation<-lm(df$Strength~poly(df$Thickness,2),data=
  df)
11
12 a<-summary(aov(relation,data=df))
13 print(a)
14
15 n<-20
16 k<-2
17 SSE<-a[[1]][, 'Sum Sq'][2]

```

```

15 SST<-a[[1]][, 'Sum Sq'] [1]+a[[1]][, 'Sum Sq'] [2]
16 R2<-1-(SSE/SST)
17 print(paste("R-squared value:",R2))
18
19 #To find variance
20 sigma2<-SSE/(n-(k+1))
21 print(paste("sigma^2:",sigma2))
22 print(paste("sigma:",sqrt(sigma2)))

```

R code Exa 13.9 Statistical intervals and test procedures

```

1 #Ex13.9, Page 548
2 #Answers may vary slightly due to rounding off of
  values
3
4 Thickness<-c
  (220,220,220,220,370,370,370,370,440,440,440,440,680,680,680,680,
5
6 Strength<-c
  (24.0,22.0,19.1,15.5,26.3,24.6,23.1,21.2,25.2,24.0,21.7,19.2,17.0
7
8
9 new<-data.frame(Thickness=500)
10 y_cap<-predict(model,new)
11 print(paste("ycap when x=500:",y_cap))
12
13 #To find confidence and prediction intervals
14 cat("95% CI:\n")
15 print(predict(model,newdata=data.frame(Thickness
  =500),interval="confidence"))
16 cat("95% PI:\n")
17 print(predict(model,newdata=data.frame(Thickness
  =500),interval="prediction"))

```

R code Exa 13.10 Centering x values

```
1 #Ex13.10, Page 549
2 #Answers may vary slightly due to rounding off of
  values
3
4 x<-c(280,284,292,295,298,305,308,315)
5 x_dash<-c
  (-17.13,-13.13,-5.13,-2.13,0.87,7.87,10.87,17.87)
6 y<-c(770,800,840,810,735,640,590,560)
7 df<-data.frame(x,x_dash,y)
8
9 #To find estimated coefficients and standard
  deviations using original model
10 m1<-lm(y~x+I(x^2))
11 print(summary(m1))
12 Estimate1<-c(NULL)
13 Estimated_SD1<-c(NULL)
14 for(i in 1:3){
15   Estimate1[i]<-m1$coefficients[i]
16   Estimated_SD1[i]<-coef(summary(m1))[, "Std. Error"
    ][i]
17 }
18 cat("Estimated coefficients and standard deviations
  using original model\n")
19 Parameter1<-c("beta0","beta1","beta2")
20 df1<-data.frame(Parameter1,Estimate1,Estimated_SD1)
21 print(df1)
22
23 #To find estimated coefficients and standard
  deviations using centered model
24 Estimate2<-c(NULL)
25 Estimated_SD2<-c(NULL)
26 m2<-lm(y~x_dash+I(x_dash^2))
```

```

27 print(summary(m2))
28 for(i in 1:3){
29   Estimate2[i]<-m2$coefficients[i]
30   Estimated_SD2[i]<-coef(summary(m2))[, "Std. Error"
    ][i]
31 }
32 cat("Estimated coefficients and standard deviations
    using centered model\n")
33 Parameter2<-c("beta0*", "beta1*", "beta2*")
34 df2<-data.frame(Parameter2, Estimate2, Estimated_SD2)
35 print(df2)

```

R code Exa 13.12 Estimating parameters

```

1 #Ex13.12, Page 558
2 #Answers may vary slightly due to rounding off of
  values
3
4 Observation<-1:30
5 Force<-c
  (30,40,30,40,30,40,30,40,30,40,30,40,30,40,30,40,25,45,35,35,35,30)
6 Power<-c
  (60,60,90,90,60,60,90,90,60,60,90,90,60,60,90,90,75,75,45,105,75,75)
7 Temperature<-c
  (175,175,175,175,225,225,225,225,175,175,175,175,225,225,225,225,175,175,175,175,225,225,225)
8 Time<-c
  (15,15,15,15,15,15,15,15,25,25,25,25,25,25,25,25,20,20,20,20,20,20,20)
9 Strength<-c
  (26.2,26.3,39.8,39.7,38.6,35.5,48.8,37.8,26.6,23.4,38.6,52.1,39.5,39.5,39.5,39.5,39.5,39.5,39.5,39.5,39.5,39.5)
10

```



```

11 m1<-lm(Strength~(Force+Power+Temperature+Time))
12
13 #To form estimated regression equation
14 Force<-35 #Force
15 Power<-75 #Power
16 Temperature<-200 #Temperature
17 Time<-20 #Time
18
19 new<-data.frame(Force,Power,Temperature,Time)
20 y_cap<-predict(m1,new)
21 print(paste("Point prediction of strength at
    (35,75,200,20):",y_cap,"gm"))

```

R code Exa 13.13 Estimating parameters

```

1 #Ex13.13, Page 560
2 #Answers may vary slightly due to rounding off of
  values
3
4 temp1<-x1<-c(21,21,7,7,28,0,14,14,14)
5 temp2<-x2<-c
    (0.65,0.55,0.65,0.55,0.60,0.60,0.70,0.50,0.60)
6 x1x2<-x1*x2
7 Comp_str<-c
    (33.55,47.55,35.00,35.90,40.90,39.10,31.55,48.00,42.30)

8 Adsorbability<-c
    (8.42,6.26,6.74,6.59,7.28,6.90,10.80,5.63,7.43)
9 df1<-data.frame(x1,x2,x1x2,Comp_str,Adsorbability)
10
11 #To find mean and SST values
12 ybar_compstr<-mean(Comp_str)
13 ybar_Adsorb<-mean(Adsorbability)
14 SST_CompStr<-sum((Comp_str-ybar_compstr)^2)
15 SST_Adsorb<-sum((Adsorbability-ybar_Adsorb)^2)

```

```

16 cat("Mean of compression strength values:", ybar_
    compstr, "\n")
17 cat("Mean of adsorbability values:", ybar_Adsorb, "\n"
    )
18 cat("SST of compression strength:", SST_CompStr, "\n")
19 cat("SST of adsorbability:", SST_Adsorb, "\n\n")
20
21 #First-order regression model
22 m1<-lm(Comp_str~(x1+x2))
23 SSE1<-sum(resid(m1)^2)
24 cat("SSE for first-order model:", SSE1, "\n")
25 cat("R squared value:", summary(m1)$r.squared, "\n\n")
26
27 #Model including interaction predictor
28 m2<-lm(Comp_str~(x1*x2))
29 SSE2<-sum(resid(m2)^2)
30 cat("SSE for first-order model including interaction
    predictor:", SSE2, "\n")
31 cat("R squared value:", summary(m2)$r.squared, "\n\n")
32
33 #To create estimated regression function
34 x1<-14 #% limestone
35 x2<-0.6 #Water-cement ratio
36 new<-data.frame(x1,x2)
37 y_cap<-predict(m2,new)
38 cat("Prediction of compression strength at (14,0.6):
    ", y_cap, "\n")
39
40 #When adsorbability is taken as the dependent
    variable
41 #First-order regression model
42 m3<-lm(Adsorbability~(temp1+temp2))
43 cat("R squared value for first-order model:", summary
    (m3)$r.squared, "\n")
44
45 #Model including interaction predictor
46 m4<-lm(Adsorbability~(temp1*temp2))
47 cat("R squared value when using interaction

```



```
22 print(paste("F value:",a[[1]]$F[1]))
```

R code Exa 13.15 Inferences in multiple regression

```
1 #Ex13.15, Page 564
2 #Answers may vary slightly due to rounding off of
  values
3
4 x1<-c
  (61,175,111,124,130,173,169,169,160,244,257,333,199)

5 x2<-c(13,21,24,23,64,38,33,61,39,71,112,88,54)
6 y<-c(4,18,14,18,26,26,21,30,28,36,65,62,40)
7 data1<-data.frame(x1,x2,y)
8
9 model<-lm(y~(x1+x2))
10 print(summary(model))
11
12 new<-data.frame(x1=160,x2=39)
13 y_cap<-predict(model,new)
14 print(paste("y_cap:",y_cap))
15
16 cat("\n99% CI:\n")
17 print(confint(model,level=0.99))
18
19 #To find confidence and prediction intervals
20 cat("95% CI:\n")
21 print(predict(model,new,interval="confidence"))
22 cat("95% PI:\n")
23 print(predict(model,new,interval="prediction"))
```

R code Exa 13.16 Inferences in multiple regression

```

1 #Ex13.16, Page 566
2 #Answers may slightly vary due to rounding off of
  values
3
4 Observation<-1:30
5 x1<-c
  (8,2,7,10,7,7,7,5,4,5,8,2,4,6,10,4,4,10,5,8,10,2,6,7,5,8,4,6,4,7)
6 x2<-c
  (4,4,4,7,4,7,13,4,7,1,10,4,10,7,13,10,13,10,4,13,1,13,13,1,13,1,1
7 x3<-c
  (100,180,180,120,180,180,140,160,140,100,140,100,180,120,180,160,
8 x4<-c
  (1,7,1,5,5,1,1,7,3,7,3,3,3,7,3,5,7,7,1,1,1,1,7,7,1,7,7,1,1,7)
9 y<-c
  (1.4,2.2,4.6,4.9,4.6,4.7,4.6,4.5,4.8,1.4,4.7,1.6,4.5,4.7,4.8,4.6,
10 x12<-x1^2
11 x22<-x2^2
12 x32<-x3^2
13 x42<-x4^2
14 x1x2<-x1*x2
15 x1x3<-x1*x3
16 x1x4<-x1*x4
17 x2x3<-x2*x3
18 x2x4<-x2*x4
19 x3x4<-x3*x4
20 df1<-data.frame(Observation,x1,x2,x3,x4,y)
21 print(df1)
22
23 #To find estimate for reduced model
24 cat("\nReduced model:\n")
25 m1<-lm(y~x1+x2+x3+x4)
26 print(summary(m1))
27

```

```

28 #To find estimate for full model
29 cat(" Full model:\n")
30 m2<-lm(y~(x1+x2+x3+x4+x12+x22+x32+x42+x1x2+x1x3+x1x4
      +x2x3+x2x4+x3x4))
31 print(summary(m2))
32
33 #H0: beta5=beta6=.....=beta14=0
34 #Ha: at least one among beta5 ,....beta14 is not 0
35 n<-30
36 k<-14
37 l<-4
38 alpha<-0.01
39 #To find F value
40 Fval<-qf(1-alpha,df1=k-1,df2=n-(k+1))
41 cat("F value:",Fval,"\n")
42
43 #Unexplained variation for the full model
44 SSEk<-sum(resid(m2)^2)
45
46 #Unexplained variation for the reduced model
47 SSEl<-sum(resid(m1)^2)
48
49 #f value
50 f<-((SSEl-SSEk)/(k-1))/(SSEk/(n-(k+1)))
51 cat("f value:",f,"\n")
52
53 if(f>=Fval) cat("H0 is rejected since",f,">=",Fval)

```

R code Exa 13.18 Transformations

```

1 #Ex13.18, Page 575
2
3 s<-c
      (20,20,20,20,20,20,20,20,20,60,60,60,60,60,60,60,60,60,60,100,100,10

```

```

4 l_1000s<-c
  (3,3,3,6,6,6,10,10,10,3,3,3,6,6,6,10,10,10,3,3,3,6,6,6,10,10,10)

5 w<-c
  (300.2,310.8,333.0,99.6,136.2,142.4,20.2,28.2,102.7,67.3,77.9,93.9)

6
7 si<-log(s,base=exp(1))
8 li<-log(l_1000s,base=exp(1))
9 wi<-log(w,base=exp(1))
10
11 #To display estimated coefficients and t ratios
12 df1<-data.frame(si,li,wi)
13 model<-lm(wi~(si+li),data=df1)
14 print(summary(model))

```

Chapter 14

Goodness of fit tests and categorical data analysis

R code Exa 14.1 Goodness of fit tests when category probabilities are completely specified

```
1 #Ex14.1, Page 597
2 #Answers may vary slightly due to rounding off of
  value
3
4 p<-c(9/16,3/16,3/16,1/16)
5 ni<-c(926,288,293,104)
6 n<-1611
7 k<-4
8 df<-k-1
9
10 npi<-round(n*p,digits=1)
11 df1<-data.frame(ni,npi)
12
13 #To find contribution to chi square from each cell
14 chi<-c(NULL)
15 for(i in 1:length(ni)){
16   s<-((ni[i]-npi[i])^2)/npi[i]
17   chi[i]<-s
```



```

18 }
19 print(paste("Chi-square values:"))
20 print(chi)
21
22 sum<-sum(chi)
23 print(paste("Chi square value:",sum))
24
25 alpha<-0.1
26 #To find chi square value
27 chival<-qchisq(1-alpha,df)
28 print(paste("Critical value:",chival))
29
30 if(sum < chival) print(paste("H0 cannot be rejected"
    )) else print(paste("H0 can be rejected"))

```

R code Exa 14.2 P values for chi squared tests

```

1 #Ex14.2, Page 599
2 #Answers may vary slightly due to rounding off of
  values
3
4 p<-9/16
5 n<-4
6 k<-5
7 df<-k-1
8 b<-c(NULL)
9 prob<-dbinom(0:4,n,p)
10 print(prob)
11
12 npi<-269*prob
13 print(npi)
14
15 Observed<-c(16,45,100,82,26)
16 Expected<-npi
17

```

```

18 df1<-data.frame(Observed,Expected)
19
20 #To find contribution to chi square from each cell
21 chi<-c(NULL)
22 for(i in 1:length(prob)){
23   s<-((Observed[i]-Expected[i])^2)/Expected[i]
24   chi[i]<-round(s,digits=3)
25 }
26 print(paste("Chi-square values:"))
27 print(chi)
28
29 sum<-sum(chi)
30 print(paste("Chi square value:",sum))
31
32 alpha<-0.01
33 #To find chi square value
34 chival<-qchisq(1-alpha,df)
35 print(paste("Critical value:",chival))
36
37 if(sum < chival) print(paste("H0 cannot be rejected"
  )) else print(paste("H0 can be rejected"))

```

R code Exa 14.3 Chi square when underlying distribution is continuous

```

1 #Ex14.3, Page 600
2 #Answers may vary slightly due to rounding off of
  values
3
4 k<-24
5 df<-k-1
6 p<-1/24
7
8 #Cell counts
9 Observed<-c
  (52,73,89,88,68,47,58,47,48,53,47,34,21,31,40,24,37,31,47,34,36,4

```

```

10 npi<-1186*p
11 Expected<-c(NULL)
12 for(i in 1:length(Observed)){
13   Expected[i]<-npi
14 }
15 chi<-c(NULL)
16 for(i in 1:length(Observed)){
17   s<-((Observed[i]-Expected[i])^2)/Expected[i]
18   chi[i]<-round(s,digits=3)
19 }
20 sum<-sum(chi)
21 print(paste("Chi square value:",sum))
22
23 alpha<-0.01
24 #To find chi square value
25 chival<-qchisq(1-alpha,df)
26 print(paste("Critical value:",chival))
27
28 if(sum < chival) print(paste("H0 cannot be rejected"
  )) else print(paste("H0 can be rejected"))

```

R code Exa 14.4 Chi square when underlying distribution is continuous

```

1 #Ex14.4, Page 600
2 #Answers may vary slightly due to rounding off fo
  values
3
4 #Cell counts
5 Observed<-c(21,17,12,16,10,15,19,10)
6 p<-1/8
7 n<-120
8 k<-8
9 df<-k-1
10 npi<-n*p

```

```

11
12 Expected<-c(NULL)
13 for(i in 1:length(Observed)){
14   Expected[i]<-npi
15 }
16 chi<-c(NULL)
17 for(i in 1:length(Observed)){
18   s<-((Observed[i]-Expected[i])^2)/Expected[i]
19   chi[i]<-round(s,digits=3)
20 }
21 sum<-sum(chi)
22 print(paste("Chi square value:",sum))
23
24 alpha<-0.1
25 #To find chi square value
26 chival<-qchisq(1-alpha,df)
27 print(paste("Critical value:",chival))
28
29 if(sum < chival) print(paste("H0 cannot be rejected"
    )) else print(paste("H0 can be rejected"))

```

R code Exa 14.5 Chi square when parameters are estimated

```

1 #Ex14.5, Page 603
2
3 Type<-c("Observed")
4 M<-c(125)
5 MN<-c(225)
6 N<-c(150)
7 print(paste("Observed counts:"))
8 df<-data.frame(Type,M,MN,N)
9 print(df)
10
11 n<-500
12

```

```

13 n1<-M
14 n2<-MN
15 theta<-(2*n1+n2)/(2*n)
16 print(paste("Resulting estimator:",theta))

```

R code Exa 14.6 Chi square when parameters are estimated

```

1 #Ex14.6 , Page 604
2 #Answers may slightly due to rounding off of values
3
4 k<-3
5 m<-1
6 theta<-0.475
7 n<-500
8 np1<-500*theta^2
9 np2<-n*2*theta*(1-theta)
10 np3<-n-np1-np2
11
12 Observed<-c(125,225,150)
13 Estimated_expected<-c(np1,np2,np3)
14
15 chi<-c(NULL)
16 for(i in 1:length(Observed)){
17   s<-((Observed[i]-Estimated_expected[i])^2)/
18     Estimated_expected[i]
19   chi[i]<-round(s,digits=3)
20 }
21 sum<-sum(chi)
22
23 alpha<-0.05
24 df<-k-m-1
25 #To find chi square value
26 chival<-qchisq(1-alpha,df)
27 print(paste("Critical value:",chival))

```

```

28
29 if(sum < chival) print(paste("H0 is rejected")) else
    print(paste("H0 can be rejected"))

```

R code Exa 14.9 Goodness of fit for discrete distributions

```

1 #Ex14.9, Page 608
2 #Answers may vary slightly due to rounding off of
  values
3
4 k<-5
5 m<-1
6 mu<-2.10
7 n<-48
8 df<-k-m-1
9
10 Frequency<-c(9,9,10,14,6)
11
12 np1<-c(NULL)
13 for(i in 1:(k-1)){
14   num<-(exp(1)^((-1)*mu))*(mu^(i-1))
15   np1[i]<-n*(num/factorial(i-1))
16
17 }
18 np1[5]<-n-np1[1]-np1[2]-np1[3]-np1[4]
19 print(np1)
20
21 chi<-c(NULL)
22 for(i in 1:length(Frequency)){
23   s<-((Frequency[i]-np1[i])^2)/np1[i]
24   chi[i]<-round(s,digits=3)
25 }
26 sum<-sum(chi)
27 print(paste("Chi square value:",sum))
28

```

```

29 alpha<-0.05
30 #To find chi square value
31 chival<-qchisq(1-alpha,df)
32 print(paste("Critical value:",chival))
33
34 if(sum < chival) print(paste("H0 is not rejected
    since",sum,"<",chival)) else print(paste("H0 can
    be rejected"))

```

R code Exa 14.10 Goodness of fit for continuous distributions

```

1 #Ex14.10, Page 608
2 #Answers may vary slightly from textbook values
3
4 data<-c
    (204,108,140,152,158,129,175,146,157,174,192,194,144,152,135,223,
5
6 n<-length(data)
7 mu_cap<-round(mean(data),digits=2)
8 sigma_cap<-round(sqrt((n-1)*(sd(data)^2)/n),digits
    =2)
9 print(paste("mu_cap:",mu_cap))
10 print(paste("sigma_cap:",sigma_cap))
11
12 #To find estimated expected counts
13 interval<-c(-Inf
    ,117.9,132.9,144.6,155.4,167.1,182.1,Inf)
14 p<-c(NULL)
15 np_i<-c(NULL)
16 for(i in 1:length(interval)-1){
17     p[i]<-pnorm(interval[i+1],mean=mu_cap,sd=sigma_cap
    )-pnorm(interval[i],mean=mu_cap,sd=sigma_cap)
18     np_i[i]<-n*p[i]
19 }

```

```

20
21 Cell<-c("(-Inf,117.9)","(117.9,132.9)","
      (132.9,144.6)","(144.6,155.4)","(155.4,167.1)","
      (167.1,182.1)","(182.1,Inf)")
22 Observed<-c(5,5,11,6,6,7,9)
23 Estimated_expected<-npi
24 data1<-data.frame(Cell,Observed,Estimated_expected)
25 print(data1)
26
27 #To find chi-squared values
28 alpha<-0.95
29 df1<-length(npi)-1
30 m<-2
31 q1<-qchisq(alpha,df=df1)
32 print(paste("X^2(0.05,k-1):",q1))
33
34 df2<-df1-m
35 q2<-qchisq(alpha,df=df2)
36 print(paste("X^2(0.05,k-1-m):",q2))

```

R code Exa 14.12 A special test for normality

```

1 #Ex14.12, Page 611
2
3 #Package to be installed: DescTools
4 library(DescTools)
5
6 yi<-c
      (-1.871,-1.404,-1.127,-0.917,-.742,-0.587,-0.446,-0.313,-0.186,-0
7
8 xi<-c
      (24.46,25.61,26.25,26.42,26.66,27.15,27.31,27.54,27.74,27.94,27.9
8
9 z<-c(NULL)

```



```

10 #To find z percentiles
11 for(i in 1:length(xi)){
12   z[i]<-round(qnorm(1-((i-0.5)/length(xi)),lower.
      tail=FALSE),digits=2)
13 }
14
15 #Shapiro test is similar to Ryan-Joiner test which
    is not available in R
16 t<-shapiro.test(xi)
17 print(t)
18
19 #To create normal probability plot
20 qqnorm(xi,datax=TRUE,ylab="Voltage",xlab="
    Probability",main="Normal probability plot")
21 qqline(xi,datax=TRUE)

```

R code Exa 14.13 Testing for homogeneity

```

1 #Ex14.13, Page 615
2
3 print(paste("Reason for nonconformity:"))
4 Blemish<-c(34,23,32)
5 Crack<-c(65,52,28)
6 Location<-c(17,25,16)
7 Missing<-c(21,19,14)
8 Other<-c(13,6,10)
9 Sample_size<-c(150,125,100)
10 df1<-data.frame(Blemish,Crack,Location,Missing,Other
    ,Sample_size)
11 print(df1)
12
13
14 #Total values
15 Blemish_tot<-sum(Blemish)
16 Crack_tot<-sum(Crack)

```

```

17 Location_tot<-sum(Location)
18 Missing_tot<-sum(Missing)
19 Other_tot<-sum(Other)
20 Sample_tot<-sum(Sample_size)
21
22 #H0: production lines are homogeneous
23 #Ha: production lines are not not homogeneous
24
25 tbl<-cbind.data.frame(df1$Blemish,df1$Crack,df1$
      Location,df1$Missing,df1$Other)
26 c<-chisq.test(tbl,df1$Sample_size)
27 print(c)
28
29 print(paste("P value:",c$p.value))
30
31 print(paste("H0 should not be rejected at levels
      0.05 or 0.01"))

```

R code Exa 14.14 Testing for independence

```

1 #Ex14.14, Page 617
2 #Answers may vary slightly due to rounding off of
  values
3
4 O<-c(24,15,17,52,73,80,58,86,36)
5 E<-c
      (17.02,22.10,16.89,62.29,80.88,61.83,54.69,71.02,54.29)
6
7 print(paste("Observed pricing policy"))
8 Observed<-matrix(data=0,nrow=3,ncol=3,byrow=TRUE)
9 rownames(Observed)<-c("Substandard","Standard","
      Modern")
10 colnames(Observed)<-c("Aggressive","Neutral","
      Nonaggressive")

```

```

11 print(Observed)
12
13 print(paste("Expected pricing policy"))
14 Expected<-matrix(data=E,nrow=3,ncol=3,byrow=TRUE)
15 rownames(Expected)<-c("Substandard","Standard","
    Modern")
16 colnames(Expected)<-c("Aggressive","Neutral","
    Nonaggressive")
17 print(Expected)
18
19 chi<-c(NULL)
20 for(i in 1:length(Observed)){
21     chi[i]<-(Observed[i]-Expected[i])^2/(Expected[i
        ])
22 }
23 sum<-sum(chi)
24 print(paste("Chi square value:",sum))
25
26 alpha<-0.01
27 #To find chi square value
28 chival<-qchisq(1-alpha,df=4)
29 print(paste("Critical value (Chi square):",chival))

```

Chapter 15

Distribution free procedures

R code Exa 15.1 The Wilcoxon signed rank test

```
1 #Ex15.1, Page 629
2
3 library(stats)
4
5 x<-c
    (494.6,510.8,487.5,493.2,502.6,485.0,495.9,498.2,501.6,497.3,492.0)
6 y<-x-500
7
8 #Performing Wilcoxon signed rank test using wilcox.
   test function
9 print(wilcox.test(y))
```

R code Exa 15.2 Paired observations

```
1 #Ex15.2, Page 630
2
3 library(stats)
```

```

4
5 IF_REE<-c
    (1753.7,1604.4,1576.5,1279.7,1754.2,1695.5,1700.1,1717.0)

6 Std_REE<-c
    (1755.0,1691.1,1697.1,1477.7,1785.2,1669.7,1901.3,1735.3)

7 Difference<-IF_REE-Std_REE
8
9 #Performing wilcoxon signed rank test
10 w1<-wilcox.test(Difference)      #or wilcox.test(IF_
    REE,Std_REE,paired=TRUE)
11 print(w1)
12
13 #Along with continuity correction
14 w2<-wilcox.test(Difference,exact=FALSE)  #or wilcox
    .test(IF_REE,Std_REE,paired=TRUE,exact=FALSE)
15 print(w2)

```

R code Exa 15.3 Large sample approximation

```

1 #Ex15.3, Page 631
2 #Here p value is used to make an inference regarding
    the rejection of hypotheses
3 #wilcoxsign_test() from coin package can also be
    used to find z value if needed
4
5 #H0: mu=50000
6 #Ha: mu<50000
7
8 Signed_rank<-c
    (-1,-2,+3,-4,+5,-6,-7,+8,-9,-10,+11,-12,-13,+14,-15,-16,-17,+18,-
9
10 xi_50000<-c
    (-10,-27,36,-55,73,-77,-81,90,-95,-99,113,-127,-129,136,-150,-155

```

```

10 alpha_level<-0.01
11
12 w1<-wilcox.test(xi_50000,Signed_rank,paired=T,exact=
    F,correct=F,alt="two.sided")
13
14 #Using p value:
15 if(w1$p.value<alpha_level) print(paste("H0 is
    rejected since p value,",w1$p.value,"is smaller
    than level,",alpha_level)) else print(paste("H0
    is not rejected"))

```

R code Exa 15.4 The Wilcoxon rank sum test

```

1 #Ex15.4, Page 637
2
3 Polluted<-c(21.3,18.7,23.0,17.1,16.8,20.9,19.7)
4 Unpolluted<-c(14.2,18.3,17.2,18.4,20.0)
5
6 #Performing Wilcoxon rank sum test....
7 print(wilcox.test(Polluted,Unpolluted,conf.level
    =0.99))

```

R code Exa 15.5 A normal approximation for W

```

1 #Ex15.5, Page 638
2 #Answers may vary slightly due to rounding off of
    values
3
4 Allergics<-c
    (67.6,39.6,1651.0,100.0,65.9,1112.0,31.0,102.4,64.7)

```

```

5 Nonallergics<-c
  (34.3,27.3,35.4,48.1,5.2,29.1,4.7,41.7,48.0,6.6,18.9,32.4,45.5)

6
7 m<-length(Allergics)
8 n<-length(Nonallergics)
9
10 #H0: mu1-mu2=0
11
12 new<-c(Allergics,Nonallergics)
13
14 #To find rank of combined vectors
15 r<-rank(new)
16 s<-0
17 for(i in 1:length(Allergics)){
18   s<-s+r[i]
19 }
20 print(paste("Rank sum of allergies:",s))
21
22 #To find mean and variance of W
23 mu_w<-(m+n+1)*m/2
24 print(paste("Mean of W:",mu_w))
25 var_w<-m*n*(m+n+1)/12
26 print(paste("Variance of W:",var_w))
27
28 #To find test statistic value
29 z<-(s-mu_w)/sqrt(var_w)
30 print(paste("Test statistic value:",z))
31
32 alpha<-0.01
33 z1<-qnorm(alpha/2)
34
35 if(z>=z1 || z<=-z1) print(paste("H0 is rejected"))
  else print(paste("H0 is not rejected"))

```

R code Exa 15.6 The Wilcoxon signed rank interval

```
1 #Ex15.6, Page 642
2 #Interval values may vary slightly .....
3
4 #Given data
5 averages<-c
   (4.51,4.55,4.59,4.705,4.72,4.745,4.76,4.795,4.835,4.90,4.915,4.93
6
7 print(summary(averages))
8
9 #To display confidence intervals of the data
10 print(wilcox.test(averages,conf.int=TRUE,conf.level
   =0.95))
```

R code Exa 15.8 The Wilcoxon rank sum interval

```
1 #Ex15.8, Page 643
2 #Answers vary from that of text book(CI in text
   =>(4830,8220))
3
4 Epoxy<-c(10860,11120,11340,12130,14380,13070)
5 Other<-c(4590,4850,6510,5640,6390,0)
6
7 dat<-data.frame(Crush_Strength=c(Epoxy,Other),Bark_
   board=rep(c("Epoxy","Other"),each=6))
8 print(dat)
9
10 #To display differences between crushing strengths
   of both bark boards
11 o<-outer(Epoxy,Other,"-")
12 print(o)
13
14 #To find corresponding CI intervals
```



```
15 print(wilcox.test(Crush_Strength~Bark_board,data=dat
    ,conf.int=TRUE,conf.level=0.95))
```

R code Exa 15.9 The Kruskal Wallis test

```
1 #Ex15.9, Page 646
2 #Answers may vary slightly due to rounding off of
  values
3
4 data<-c
    (309.2,309.7,311.0,316.8,326.5,349.8,409.5,331.0,347.2,348.9,361.0)

5 m1<-matrix(data,nrow=5,dimnames=list(c("4'", "6'", "8'", "10'", "12'"),byrow=TRUE))
6 print(m1)
7
8 N<-length(data)
9
10 #To find ranks of respective data
11 r1<-matrix(rank(m1),nrow=5)
12 cat("Ranks of data\n")
13 print(r1)
14
15 I<-nrow(r1)
16 J<-ncol(r1)
17
18 #To find rank sum and mean with respect to each
  plate length type
19 ri<-c(NULL)
20 ribar<-c(NULL)
21 for(i in 1:nrow(r1)){
22   ri[i]<-sum(r1[i,])
23   ribar[i]<-round(mean(r1[i,]),digits=2)
24 }
25 df1<-data.frame(ri,ribar)
```

```

26 cat("Rank sum and mean\n")
27 print(df1)
28
29 #To find k value
30 sum<-0
31 for(j in 1:nrow(r1)){
32     sum<-sum+(ri[j])^2/J
33 }
34 k<-(12/(N*(N+1)))*sum-3*(N+1)
35 print(paste("k value:",k))
36
37 #To find chi-square value
38 alpha<-0.99
39 chival<-qchisq(alpha,df=I-1)
40 print(paste("Chi-squared value:",chival))
41
42 if(k>=chival) print(paste("H0 is not rejected since",
    ,k,">=",chival)) else print(paste("H0 is
    rejected"))

```

R code Exa 15.10 Friedman test for a randomized block experiment

```

1 #Ex15.10, Page 647
2 #Answers may vary slightly due to rounding off of
  values
3
4 data<-c
    (23.1,57.6,10.5,23.6,11.9,54.6,21.0,20.3,22.7,53.2,9.7,19.6,13.8,
5 m1<-matrix(data,nrow=4,dimnames=list(c("Fear","
    Happiness","Depression","Calmness"),c("1","2","3"
    ,"4","5","6","7","8"))),byrow=TRUE)
6 print(m1)
7
8 #To find ranks of respective data

```

```

 9  r1<-rank(m1[,1])
10  r2<-rank(m1[,2])
11  r3<-rank(m1[,3])
12  r4<-rank(m1[,4])
13  r5<-rank(m1[,5])
14  r6<-rank(m1[,6])
15  r7<-rank(m1[,7])
16  r8<-rank(m1[,8])
17  data1<-data.frame(r1,r2,r3,r4,r5,r6,r7,r8)
18  cat("\nRanks of data:\n")
19  print(data1)
20
21  I<-nrow(data1)
22  J<-ncol(data1)
23
24  #To find rank sum and mean with respect to each
    plate length type
25  ri<-c(NULL)
26  ri2<-c(NULL)
27  for(i in 1:nrow(data1)){
28    ri[i]<-sum(data1[i,])
29    ri2[i]<-(ri[i])^2
30  }
31  df1<-data.frame(ri,ri2)
32  print(df1)
33
34  #To find test statistic
35  Fr<-(12/(I*J*(I+1)))*sum(ri2)-3*J*(I+1)
36  print(paste("Test statistic value:",Fr))
37
38  #To find chi-square value
39  alpha<-0.95
40  chival<-qchisq(alpha,df=I-1)
41  print(paste("Chi-squared value:",chival))
42
43  if(Fr<chival) print(paste("H0 is not rejected since",
    ,Fr,"<",chival)) else print(paste("H0 is
    rejected"))

```


Chapter 16

Quality control methods

R code Exa 16.1 X bar chart based on known parameter values

```
1 #Ex16.1 , Page 655
2 #Answers may vary slightly due to rounding off of
  values
3
4 #Package to be installed: qcc(Quality control charts
  )
5 library(qcc)
6
7 Vis_obs<-c
  (10.37,10.48,10.77,10.47,10.84,10.48,10.41,10.40,10.33,10.73,10.4
8
9 s<-c(rep(1:25,3))
10
11 q<-qcc.groups(Vis_obs,s)
12
13 #To plot X-bar chart
14 q1<-qcc(q,type="xbar",center=10.5,std.dev=0.18)
15 plot(q1)
16 print(summary(q1))
```

R code Exa 16.2 X bar charts based on estimated parameters

```
1 #Ex16.2, Page 657
2 #Answers may vary slightly due to rounding off of
  values
3
4 n<-3
5 k<-25
6 xdoublebar<-261.896/k
7 sbar<-3.834/k
8 a3<-0.886
9
10 #To find control limits
11 LCL<-xdoublebar-n*sbar/(a3*sqrt(n))
12 UCL<-xdoublebar+n*sbar/(a3*sqrt(n))
13 print(paste("LCL:",LCL))
14 print(paste("UCL:",UCL))
15
16 #Old limits:
17 #mu=10.5, sigma=0.18
18
19 #New limits:
20 print(paste("mu_cap:",xdoublebar))
21 print(paste("sigma_cap:",sbar/a3))
```

R code Exa 16.3 X bar charts based on estimated parameters

```
1 #Ex16.3, Page 658
2
3 n<-3
4 k<-25
5 xdoublebar<-261.896/k
```

```

6 rbar<-0.292
7 b3<-1.693
8 sigma_cap<-0.292/b3
9
10 #To find control limits
11 LCL<-xdoublebar-3*rbar/(b3*sqrt(n))
12 UCL<-xdoublebar+3*rbar/(b3*sqrt(n))
13 print(paste("LCL:",round(LCL,digits=3)))
14 print(paste("UCL:",round(UCL,digits=3)))

```

R code Exa 16.4 The S chart

```

1 #Ex16.4, Page 663
2 #Answers may vary slightly due to rounding off of
  values
3
4 #Package to be installed: qcc(Quality control charts
  )
5 library(qcc)
6
7 n<-4
8 k<-22
9 obs1<-c
  (29.7,32.2,35.9,28.8,30.9,30.6,32.3,32.0,24.2,33.7,35.3,28.1,28.7
10
11 obs2<-c
  (29.0,29.3,29.1,27.2,32.6,34.3,27.7,27.9,27.5,24.4,33.2,34.0,28.9
12
13 obs3<-c
  (28.8,32.2,32.1,28.5,28.3,34.8,30.9,31.0,28.5,34.3,31.4,31.0,25.8
14
15 obs4<-c
  (30.2,32.9,31.3,35.7,28.3,26.3,27.8,30.8,31.1,31.0,28.0,30.8,29.7
16
17 SD<-c

```

```

      (0.64,1.60,2.83,3.83,2.11,3.94,2.30,1.76,2.85,4.53,3.09,2.41,1.71,
14  Range<-c
      (1.4,3.6,6.8,8.5,4.3,8.5,4.6,4.1,6.9,9.9,7.3,5.9,3.9,4.0,4.4,5.2,

15  data1<-data.frame(obs1,obs2,obs3,obs4)
16
17  #Sum of SD values
18  si<-sum(SD)
19  print(paste("Sum of SD values:",si))
20  sbar<-mean(SD)
21  print(paste("Mean of SD values:",sbar))
22
23  a4<-0.921
24  LCL<-0
25  UCL<-sbar+3*sbar*sqrt(1-a4^2)/a4
26  print(paste("LCL:",LCL))
27  print(paste("UCL:",UCL))
28
29  #To plot S chart
30  q4<-qcc(data1,type="S")
31  print(summary(q4))

```

R code Exa 16.5 R chart

```

1  #Ex16.5, Page 665
2  #Answers may vary slightly due to rounding off of
   values
3
4  #Package to be installed: qcc(Quality control charts
   )
5  library(qcc)
6
7  des_dim<-c
      (200,250,300,350,400,450,500,550,600,650,700,750,800,850,900,950,

```



```

8  obs1<-c(12,6,5,19,9,9,8,4,11,13,10,8,14,7,14,10,7)
9  obs2<-c(17,9,9,6,14,15,11,14,14,9,14,9,7,9,5,12,11)
10 obs3<-c(6,17,15,11,9,8,12,11,7,9,8,4,9,12,8,10,15)
11 mean<-c
    (11.7,10.7,9.7,12.0,10.7,10.7,10.3,9.7,10.7,10.3,10.7,7.0,10.0,9.7,10.7,10.7,10.7)

12 range<-c(11,11,10,13,5,7,4,10,7,4,6,5,7,5,9,2,8)
13 st_dev<-c
    (5.51,5.69,5.03,6.56,2.84,3.79,2.08,5.13,3.51,2.31,3.06,2.65,3.61,2.65,2.65,2.65,2.65)

14 data1<-data.frame(obs1,obs2,obs3)
15
16 n<-3
17 b3<-1.693
18 c3<-0.888
19
20 #To find sum of range values
21 ri<-sum(range)
22 rbar<-mean(range)
23
24 #To find control limits
25 UCL<-rbar+n*c3*rbar/b3
26 print(paste("LCL:",LCL))
27 print(paste("UCL:",UCL))
28
29 #To display X-bar chart (can only be displayed
    separately by using run method)
30 q1<-qcc(data1,type="xbar")
31 print(summary(q1))
32
33 #To display R chart (can only be displayed
    separately by using run method)
34 q2<-qcc(data1,type="R")
35 print(summary(q2))

```

R code Exa 16.6 The p chart

```
1 #Ex16.6, Page 669
2 #Answers may vary slightly due to rounding off of
  values
3
4 #Package to be installed: qcc(Quality control charts
  )
5 library(qcc)
6
7 Dayi<-c
  (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25)
8 xi<-c
  (7,4,3,6,4,9,6,7,5,3,7,8,4,6,2,9,7,6,7,11,6,7,4,8,6)
9 pi<-c
  (0.07,0.04,0.03,0.06,0.04,0.09,0.06,0.07,0.05,0.03,0.07,0.08,0.04,0.05,0.06,0.04,0.03,0.05,0.07,0.06,0.04,0.05,0.03,0.06)
10 data1<-data.frame(xi,pi)
11
12 #Sum of pi values
13 p<-sum(pi)
14 print(paste("Sum of pi values:",p))
15 pbar<-mean(pi)
16 print(paste("Mean of p values:",pbar))
17
18 #To find control limits
19 LCL<-pbar-3*sqrt(pbar*(1-pbar)/100)
20 UCL<-pbar+3*sqrt(pbar*(1-pbar)/100)
21 print(paste("LCL:",LCL))
22 print(paste("UCL:",UCL))
23
24 #To display p chart
```

```

25 q1<-qcc(data1,sizes=length(Dayi),type="p")
26 print(summary(q1))

```

R code Exa 16.7 c chart for number of defectives

```

1 #Ex16.7, Page 670
2 #Answers may slightly vary due to rounding off of
  values
3
4 #Package to be installed: qcc(Quality control charts
  )
5 library(qcc)
6
7 x<-c
  (7,10,9,12,13,6,13,7,5,11,8,10,13,9,21,10,6,8,3,12,7,11,14,10)
8
9 #To find the sum of x values
10 xi<-sum(x)
11 xbar<-mean(x)
12 print(paste("Sum of x values:",xi))
13 print(paste("Mean of x values:",xbar))
14
15 #To find control limits
16 LCL<-xbar-3*sqrt(xbar)
17 UCL<-xbar+3*sqrt(xbar)
18 print(paste("LCL:",LCL))
19 print(paste("UCL:",UCL))
20
21 #To display c chart
22 q1<-qcc(x,sizes=length(x),type="c")
23 print(summary(q1))
24
25 #From the chart, the 15th value lies above the UCL
26 #Eliminating that observation.....

```

```

27 xbar1<-(sum(x)-x[15])/(length(x)-1)
28 print(paste("Mean value when the 15th value is
    eliminated:",xbar1))
29 LCL1<-xbar1-3*sqrt(xbar1)
30 UCL1<-xbar1+3*sqrt(xbar1)
31 print(paste("Control limits when 15th value is
    eliminated:"))
32 print(paste("New LCL value:",LCL1))
33 print(paste("New UCL value:",UCL1))
34
35 #Remaining 23 observation lie within control lines (
    in-control)

```

R code Exa 16.8 CUSUM procedures

```

1 #Ex16.8, Page 674
2
3 #Package to be installed: qcc(Quality control charts
  )
4 library(qcc)
5
6 n<-4
7 mu=mu0=40
8 sigma<-0.5
9 obs<-c
    (40.77,39.95,40.86,39.21,38.94,39.70,40.37,39.88,40.43,40.27,40.9
10 sa<-c(rep(1,4),rep(2,4),rep(3,4),rep(4,4),rep(5,4),
    rep(6,4),rep(7,4),rep(8,4),rep(9,4),rep(10,4),rep
    (11,4),rep(12,4),rep(13,4),rep(14,4),rep(15,4),
    rep(16,4))
11
12 #To display X-bar chart
13 q<-qcc.groups(data=obs,sample=sa)
14 q4<-qcc(q,center=mu0,std.dev=sigma,type="xbar")

```

```
15 print(summary(q4))
```

R code Exa 16.9 Computational version

```
1 #Ex16.9 , Page 676
2
3 #Package to be installed: qcc(Quality control charts
  )
4 library(qcc)
5
6 mu0<-40
7 delta<-0.3
8 k<-delta/2
9 l<-mu0-k
10 r<-mu0+k
11 obs<-c
    (40.77,39.95,40.86,39.21,38.94,39.70,40.37,39.88,40.43,40.27,40.9
12 sa<-c(rep(1,4),rep(2,4),rep(3,4),rep(4,4),rep(5,4),
    rep(6,4),rep(7,4),rep(8,4),rep(9,4),rep(10,4),rep
    (11,4),rep(12,4),rep(13,4),rep(14,4),rep(15,4),
    rep(16,4))
13 xbar<-c
    (40.20,39.72,40.42,39.98,40.06,39.76,39.65,40.41,40.32,39.84,40.4
14
15 xbar1<-xbar-r
16 xbar2<-xbar-l
17
18 #To find d and e values
19 d0<-0
20 d1<-max(0,d0+(xbar[1]-r))
21 e0<-0
22 e1<-max(0,e0-(xbar[1]-l))
23
```

```

24 d<-c(d1)
25 e<-c(e1)
26 for(i in 2:length(xbar)){
27   d[i]<-max(0,d[i-1]+(xbar[i]-r))
28   e[i]<-max(0,e[i-1]-(xbar[i]-l))
29 }
30
31 df<-data.frame(xbar,xbar1,d,xbar2,e)
32 print(df)
33
34 q<-qcc.groups(data=obs,sample=sa)
35 c<-cusum(q,sizes=4,se.shift=delta,center=mu0)
36 print(summary(c))

```

R code Exa 16.10 Designing a CUSUM procedure

```

1 #Ex16.10 , Page 680
2
3 sigma<-0.004
4 kbar<-0.74
5 delta<-0.003
6
7 n<-round((2*kbar*sigma/delta)^2,digits=0)
8 print(paste("Sample size:",n))
9
10 hbar<-3.2
11 h<-(sigma/sqrt(n))*hbar
12 print(paste("h value:",h))
13
14 print(paste("Out of control signal if d>",h,"or e>",
             h))

```

R code Exa 16.11 Acceptance sampling

```

1 #Ex16.11, Page 681
2
3 P<- function(p) {(1-p)^50+50*p*(1-p)^49+1225*p^2*(1-
  p)^48}
4
5 p<-c
  (0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.10,0.12,0.15)

6
7 #Finding P(A) values
8 PA<-c(NULL)
9 for(i in 1:length(p)){
10   PA[i]<-round(P(p[i]),digits=3)
11 }
12
13 df<-data.frame(p,PA)
14 print(df)
15
16 #To plot data
17 plot(p,PA,ylab="P(A)",main="Curve plotting P(A) and
  p values")

```

R code Exa 16.12 Designing a single sample plan

```

1 #Ex16.12, Page 683
2 #Answers may vary slightly due to rounding off of
  values
3
4 #Package to be installed: AcceptanceSampling
5 library(AcceptanceSampling)
6
7 c<-0:15
8 np1<-c
  (0.051,0.355,0.818,1.366,1.970,2.613,3.285,3.981,4.695,5.425,6.160,6.881,7.601,8.321,9.041,9.761,10.481,11.201,11.921,12.641,13.361,14.081,14.801,15.521,16.241,16.961,17.681,18.401,19.121,19.841,20.561,21.281,22.001,22.721,23.441,24.161,24.881,25.601,26.321,27.041,27.761,28.481,29.201,29.921,30.641,31.361,32.081,32.801,33.521,34.241,34.961,35.681,36.401,37.121,37.841,38.561,39.281,40.001,40.721,41.441,42.161,42.881,43.601,44.321,45.041,45.761,46.481,47.201,47.921,48.641,49.361,50.081,50.801,51.521,52.241,52.961,53.681,54.401,55.121,55.841,56.561,57.281,58.001,58.721,59.441,60.161,60.881,61.601,62.321,63.041,63.761,64.481,65.201,65.921,66.641,67.361,68.081,68.801,69.521,70.241,70.961,71.681,72.401,73.121,73.841,74.561,75.281,76.001,76.721,77.441,78.161,78.881,79.601,80.321,81.041,81.761,82.481,83.201,83.921,84.641,85.361,86.081,86.801,87.521,88.241,88.961,89.681,90.401,91.121,91.841,92.561,93.281,94.001,94.721,95.441,96.161,96.881,97.601,98.321,99.041,99.761,100.481,101.201,101.921,102.641,103.361,104.081,104.801,105.521,106.241,106.961,107.681,108.401,109.121,109.841,110.561,111.281,112.001,112.721,113.441,114.161,114.881,115.601,116.321,117.041,117.761,118.481,119.201,119.921,120.641,121.361,122.081,122.801,123.521,124.241,124.961,125.681,126.401,127.121,127.841,128.561,129.281,130.001,130.721,131.441,132.161,132.881,133.601,134.321,135.041,135.761,136.481,137.201,137.921,138.641,139.361,140.081,140.801,141.521,142.241,142.961,143.681,144.401,145.121,145.841,146.561,147.281,148.001,148.721,149.441,150.161,150.881,151.601,152.321,153.041,153.761,154.481,155.201,155.921,156.641,157.361,158.081,158.801,159.521,160.241,160.961,161.681,162.401,163.121,163.841,164.561,165.281,166.001,166.721,167.441,168.161,168.881,169.601,170.321,171.041,171.761,172.481,173.201,173.921,174.641,175.361,176.081,176.801,177.521,178.241,178.961,179.681,180.401,181.121,181.841,182.561,183.281,184.001,184.721,185.441,186.161,186.881,187.601,188.321,189.041,189.761,190.481,191.201,191.921,192.641,193.361,194.081,194.801,195.521,196.241,196.961,197.681,198.401,199.121,199.841,200.561,201.281,202.001,202.721,203.441,204.161,204.881,205.601,206.321,207.041,207.761,208.481,209.201,209.921,210.641,211.361,212.081,212.801,213.521,214.241,214.961,215.681,216.401,217.121,217.841,218.561,219.281,220.001,220.721,221.441,222.161,222.881,223.601,224.321,225.041,225.761,226.481,227.201,227.921,228.641,229.361,230.081,230.801,231.521,232.241,232.961,233.681,234.401,235.121,235.841,236.561,237.281,238.001,238.721,239.441,240.161,240.881,241.601,242.321,243.041,243.761,244.481,245.201,245.921,246.641,247.361,248.081,248.801,249.521,250.241,250.961,251.681,252.401,253.121,253.841,254.561,255.281,256.001,256.721,257.441,258.161,258.881,259.601,260.321,261.041,261.761,262.481,263.201,263.921,264.641,265.361,266.081,266.801,267.521,268.241,268.961,269.681,270.401,271.121,271.841,272.561,273.281,274.001,274.721,275.441,276.161,276.881,277.601,278.321,279.041,279.761,280.481,281.201,281.921,282.641,283.361,284.081,284.801,285.521,286.241,286.961,287.681,288.401,289.121,289.841,290.561,291.281,292.001,292.721,293.441,294.161,294.881,295.601,296.321,297.041,297.761,298.481,299.201,299.921,300.641,301.361,302.081,302.801,303.521,304.241,304.961,305.681,306.401,307.121,307.841,308.561,309.281,310.001,310.721,311.441,312.161,312.881,313.601,314.321,315.041,315.761,316.481,317.201,317.921,318.641,319.361,320.081,320.801,321.521,322.241,322.961,323.681,324.401,325.121,325.841,326.561,327.281,328.001,328.721,329.441,330.161,330.881,331.601,332.321,333.041,333.761,334.481,335.201,335.921,336.641,337.361,338.081,338.801,339.521,340.241,340.961,341.681,342.401,343.121,343.841,344.561,345.281,346.001,346.721,347.441,348.161,348.881,349.601,350.321,351.041,351.761,352.481,353.201,353.921,354.641,355.361,356.081,356.801,357.521,358.241,358.961,359.681,360.401,361.121,361.841,362.561,363.281,364.001,364.721,365.441,366.161,366.881,367.601,368.321,369.041,369.761,370.481,371.201,371.921,372.641,373.361,374.081,374.801,375.521,376.241,376.961,377.681,378.401,379.121,379.841,380.561,381.281,382.001,382.721,383.441,384.161,384.881,385.601,386.321,387.041,387.761,388.481,389.201,389.921,390.641,391.361,392.081,392.801,393.521,394.241,394.961,395.681,396.401,397.121,397.841,398.561,399.281,400.001,400.721,401.441,402.161,402.881,403.601,404.321,405.041,405.761,406.481,407.201,407.921,408.641,409.361,410.081,410.801,411.521,412.241,412.961,413.681,414.401,415.121,415.841,416.561,417.281,418.001,418.721,419.441,420.161,420.881,421.601,422.321,423.041,423.761,424.481,425.201,425.921,426.641,427.361,428.081,428.801,429.521,430.241,430.961,431.681,432.401,433.121,433.841,434.561,435.281,436.001,436.721,437.441,438.161,438.881,439.601,440.321,441.041,441.761,442.481,443.201,443.921,444.641,445.361,446.081,446.801,447.521,448.241,448.961,449.681,450.401,451.121,451.841,452.561,453.281,454.001,454.721,455.441,456.161,456.881,457.601,458.321,459.041,459.761,460.481,461.201,461.921,462.641,463.361,464.081,464.801,465.521,466.241,466.961,467.681,468.401,469.121,469.841,470.561,471.281,472.001,472.721,473.441,474.161,474.881,475.601,476.321,477.041,477.761,478.481,479.201,479.921,480.641,481.361,482.081,482.801,483.521,484.241,484.961,485.681,486.401,487.121,487.841,488.561,489.281,490.001,490.721,491.441,492.161,492.881,493.601,494.321,495.041,495.761,496.481,497.201,497.921,498.641,499.361,500.081,500.801,501.521,502.241,502.961,503.681,504.401,505.121,505.841,506.561,507.281,508.001,508.721,509.441,510.161,510.881,511.601,512.321,513.041,513.761,514.481,515.201,515.921,516.641,517.361,518.081,518.801,519.521,520.241,520.961,521.681,522.401,523.121,523.841,524.561,525.281,526.001,526.721,527.441,528.161,528.881,529.601,530.321,531.041,531.761,532.481,533.201,533.921,534.641,535.361,536.081,536.801,537.521,538.241,538.961,539.681,540.401,541.121,541.841,542.561,543.281,544.001,544.721,545.441,546.161,546.881,547.601,548.321,549.041,549.761,550.481,551.201,551.921,552.641,553.361,554.081,554.801,555.521,556.241,556.961,557.681,558.401,559.121,559.841,560.561,561.281,562.001,562.721,563.441,564.161,564.881,565.601,566.321,567.041,567.761,568.481,569.201,569.921,570.641,571.361,572.081,572.801,573.521,574.241,574.961,575.681,576.401,577.121,577.841,578.561,579.281,580.001,580.721,581.441,582.161,582.881,583.601,584.321,585.041,585.761,586.481,587.201,587.921,588.641,589.361,590.081,590.801,591.521,592.241,592.961,593.681,594.401,595.121,595.841,596.561,597.281,598.001,598.721,599.441,600.161,600.881,601.601,602.321,603.041,603.761,604.481,605.201,605.921,606.641,607.361,608.081,608.801,609.521,610.241,610.961,611.681,612.401,613.121,613.841,614.561,615.281,616.001,616.721,617.441,618.161,618.881,619.601,620.321,621.041,621.761,622.481,623.201,623.921,624.641,625.361,626.081,626.801,627.521,628.241,628.961,629.681,630.401,631.121,631.841,632.561,633.281,634.001,634.721,635.441,636.161,636.881,637.601,638.321,639.041,639.761,640.481,641.201,641.921,642.641,643.361,644.081,644.801,645.521,646.241,646.961,647.681,648.401,649.121,649.841,650.561,651.281,652.001,652.721,653.441,654.161,654.881,655.601,656.321,657.041,657.761,658.481,659.201,659.921,660.641,661.361,662.081,662.801,663.521,664.241,664.961,665.681,666.401,667.121,667.841,668.561,669.281,670.001,670.721,671.441,672.161,672.881,673.601,674.321,675.041,675.761,676.481,677.201,677.921,678.641,679.361,680.081,680.801,681.521,682.241,682.961,683.681,684.401,685.121,685.841,686.561,687.281,688.001,688.721,689.441,690.161,690.881,691.601,692.321,693.041,693.761,694.481,695.201,695.921,696.641,697.361,698.081,698.801,699.521,700.241,700.961,701.681,702.401,703.121,703.841,704.561,705.281,706.001,706.721,707.441,708.161,708.881,709.601,710.321,711.041,711.761,712.481,713.201,713.921,714.641,715.361,716.081,716.801,717.521,718.241,718.961,719.681,720.401,721.121,721.841,722.561,723.281,724.001,724.721,725.441,726.161,726.881,727.601,728.321,729.041,729.761,730.481,731.201,731.921,732.641,733.361,734.081,734.801,735.521,736.241,736.961,737.681,738.401,739.121,739.841,740.561,741.281,742.001,742.721,743.441,744.161,744.881,745.601,746.321,747.041,747.761,748.481,749.201,749.921,750.641,751.361,752.081,752.801,753.521,754.241,754.961,755.681,756.401,757.121,757.841,758.561,759.281,760.001,760.721,761.441,762.161,762.881,763.601,764.321,765.041,765.761,766.481,767.201,767.921,768.641,769.361,770.081,770.801,771.521,772.241,772.961,773.681,774.401,775.121,775.841,776.561,777.281,778.001,778.721,779.441,780.161,780.881,781.601,782.321,783.041,783.761,784.481,785.201,785.921,786.641,787.361,788.081,788.801,789.521,790.241,790.961,791.681,792.401,793.121,793.841,794.561,795.281,796.001,796.721,797.441,798.161,798.881,799.601,800.321,801.041,801.761,802.481,803.201,803.921,804.641,805.361,806.081,806.801,807.521,808.241,808.961,809.681,810.401,811.121,811.841,812.561,813.281,814.001,814.721,815.441,816.161,816.881,817.601,818.321,819.041,819.761,820.481,821.201,821.921,822.641,823.361,824.081,824.801,825.521,826.241,826.961,827.681,828.401,829.121,829.841,830.561,831.281,832.001,832.721,833.441,834.161,834.881,835.601,836.321,837.041,837.761,838.481,839.201,839.921,840.641,841.361,842.081,842.801,843.521,844.241,844.961,845.681,846.401,847.121,847.841,848.561,849.281,850.001,850.721,851.441,852.161,852.881,853.601,854.321,855.041,855.761,856.481,857.201,857.921,858.641,859.361,860.081,860.801,861.521,862.241,862.961,863.681,864.401,865.121,865.841,866.561,867.281,868.001,868.721,869.441,870.161,870.881,871.601,872.321,873.041,873.761,874.481,875.201,875.921,876.641,877.361,878.081,878.801,879.521,880.241,880.961,881.681,882.401,883.121,883.841,884.561,885.281,886.001,886.721,887.441,888.161,888.881,889.601,890.321,891.041,891.761,892.481,893.201,893.921,894.641,895.361,896.081,896.801,897.521,898.241,898.961,899.681,900.401,901.121,901.841,902.561,903.281,904.001,904.721,905.441,906.161,906.881,907.601,908.321,909.041,909.761,910.481,911.201,911.921,912.641,913.361,914.081,914.801,915.521,916.241,916.961,917.681,918.401,919.121,919.841,920.561,921.281,922.001,922.721,923.441,924.161,924.881,925.601,926.321,927.041,927.761,928.481,929.201,929.921,930.641,931.361,932.081,932.801,933.521,934.241,934.961,935.681,936.401,937.121,937.841,938.561,939.281,940.001,940.721,941.441,942.161,942.881,943.601,944.321,945.041,945.761,946.481,947.201,947.921,948.641,949.361,950.081,950.801,951.521,952.241,952.961,953.681,954.401,955.121,955.841,956.561,957.281,958.001,958.721,959.441,960.161,960.881,961.601,962.321,963.041,963.761,964.481,965.201,965.921,966.641,967.361,968.081,968.801,969.521,970.241,970.961,971.681,972.401,973.121,973.841,974.561,975.281,976.001,976.721,977.441,978.161,978.881,979.601,980.321,981.041,981.761,982.481,983.201,983.921,984.641,985.361,986.081,986.801,987.521,988.241,988.961,989.681,990.401,991.121,991.841,992.561,993.281,994.001,994.721,995.441,996.161,996.881,997.601,998.321,999.041,999.761,1000.481,1001.201,1001.921,1002.641,1003.361,1004.081,1004.801,1005.521,1006.241,1006.961,1007.681,1008.401,1009.121,1009.841,1010.561,1011.281,1012.001,1012.721,1013.441,1014.161,1014.881,1015.601,1016.321,1017.041,1017.761,1018.481,1019.201,1019.921,1020.641,1021.361,1022.081,1022.801,1023.521,1024.241,1024.961,1025.681,1026.401,1027.121,1027.841,1028.561,1029.281,1030.001,1030.721,1031.441,1032.161,1032.881,1033.601,1034.321,1035.041,1035.761,1036.481,1037.201,1037.921,1038.641,1039.361,1040.081,1040.801,1041.521,1042.241,1042.961,1043.681,1044.401,1045.121,1045.841,1046.561,1047.281,1048.001,1048.721,1049.441,1050.161,1050.881,1051.601,1052.321,1053.041,1053.761,1054.481,1055.201,1055.921,1056.641,1057.361,1058.081,1058.801,1059.521,1060.241,1060.961,1061.681,1062.401,1063.121,1063.841,1064.561,1065.281,1066.001,1066.721,1067.441,1068.161,1068.881,1069.601,1070.321,1071.041,1071.761,1072.481,1073.201,1073.921,1074.641,1075.361,1076.081,1076.801,1077.521,107
```

```

9  np2<-c
    (2.30,3.89,5.32,6.68,7.99,9.28,10.53,11.77,12.99,14.21,15.41,16.6
10  p2_p1<-np2/np1
11  df1<-data.frame(c,np1,np2,p2_p1)
12  print(df1)
13  cat("\n")
14
15  AQL<-p1<-0.01
16  LTPD<-p2<-0.045
17
18  #Ratio of p2 to p1
19  r<-LTPD/AQL
20  print(paste("Ratio of p2 to p1:",r))
21
22  for(i in 1:length(c)){
23      if(r>p2_p1[i+1] && r<p2_p1[i]){
24          j<-i
25          np_1<-np1[i]
26          c1<-c[i]
27          c2<-c[i+1]
28          np_2<-np2[i+1]
29
30      }
31  }
32  cat("Ratio lies between",p2_p1[j],"and",p2_p1[j+1],"
    values corresponding to c values",c[j],"and",c[j
    +1],"\n")
33
34  #Single sample plan: 1st c value
35  cat("When c value is",c1,"\n")
36
37  ##To find sample size
38  n<-round(np_1/p1,digits=0)
39  print(paste("Sample size:",n))
40
41  ##To find alpha and beta values
42  alpha1<-1-pbinom(c1,n,p1)

```



```

43 print(paste("Alpha value:",alpha1))
44
45 beta1<-pbinom(c1,n,p2)
46 print(paste("Beta value:",beta1))
47
48 #Single sample plan: 2nd c value
49 cat("When c value is",c2,"\n")
50
51 #To find sample size
52 n<-round(np_2/p2,digits=0)
53 print(paste("Sample size:",n))
54
55 ##To find alpha and beta values
56 alpha2<-1-pbinom(c2,n,p1)
57 print(paste("Alpha value:",alpha2))
58
59 beta2<-pbinom(c2,n,p2)
60 print(paste("Beta value:",beta2))

```
