# R Textbook Companion for Statistical Techniques In Business And Economics by Douglas A. Lind, William G. Marchal, Samuel A. Wathen[1]

Created by
Swaraj Vishwas Sawant
B.Sc.
Data Science
Vidyalankar School Of Information Technology
Cross-Checked by
R TBC Team

May 22, 2025

[1]Funded by a grant from the National Mission on Education through ICT - http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and R codes written in it can be downloaded from the "Textbook Companion Project" section at the website - https://r.fossee.in.

# Book Description

**Title:** Statistical Techniques In Business And Economics

**Author:** Douglas A. Lind, William G. Marchal, Samuel A. Wathen

**Publisher:** Mcgraw-hill Education, New York

**Edition:** 17

**Year:** 2018

**ISBN:** 978-1-259-66636-0

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

**Exa** Example (Solved example)

**Eqn** Equation (Particular equation of the above book)

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

# Contents

List of R Codes 4

2 Describing Data 5

3 Describing Data 11

4 Describing Data 22

5 A Survey of Probability Concepts 32

6 Discrete Probability Distributions 39

7 Continuous Probability Distributions 44

8 Sampling Methods and the Central Limit Theorem 51

9 Estimation and Confidence Intervals 57

10 One Sample Tests of Hypothesis 65

11 Two Sample Tests of Hypothesis 70

12 Analysis of Variance 74

13 Correlation and Linear Regression 77

14 Multiple Regression Analysis 86

15 Nonparametric Methods 91

**16 Nonparametric Methods** **99**

**17 Index Numbers** **107**

**18 Time Series and Forecasting** **115**

**19 Statistical Process Control and Quality Management** **122**

# List of R Codes


| | | |
|---|---|---|
| Exa 2.2 | GRAPHIC PRESENTATION OF QUALITATIVE DATA | 5 |
| Exa 2.3 | FREQUENCY DISTRIBUTIONS | 6 |
| Exa 2.4.1 | Histogram | 7 |
| Exa 2.4.2 | Cumulative Distributions | 8 |
| Exa 3.1.1 | Population Mean | 11 |
| Exa 3.1.2 | Sample Mean | 11 |
| Exa 3.1.3 | Median | 12 |
| Exa 3.1.4 | Mode | 12 |
| Exa 3.1.5 | Software Solution | 13 |
| Exa 3.2 | WEIGHTED MEAN | 14 |
| Exa 3.3.1 | GEOMETRIC MEAN | 15 |
| Exa 3.3.2 | GEOMETRIC MEAN | 15 |
| Exa 3.4.1 | Range | 16 |
| Exa 3.4.2 | Variance | 16 |
| Exa 3.4.3 | Population Variance | 17 |
| Exa 3.4.4 | Sample Variance | 17 |
| Exa 3.4.5 | Sample Standard Deviation | 17 |
| Exa 3.5.1 | Chebyshev Theorem | 18 |
| Exa 3.5.2 | Empirical Rule | 18 |
| Exa 3.6.1 | Arithmetic Mean of Grouped Data | 19 |
| Exa 3.6.2 | Standard Deviation of Grouped Data | 20 |
| Exa 4.1 | DOT PLOTS | 22 |
| Exa 4.2 | STEM AND LEAF DISPLAYS | 23 |
| Exa 4.3 | Quartiles Deciles and Percentiles | 23 |
| Exa 4.4.1 | BOX PLOTS | 24 |
| Exa 4.4.2 | BOX PLOTS | 25 |
| Exa 4.5 | SKEWNESS | 26 |

| | | |
|---|---|---|
| Exa 4.6 | DESCRIBING THE RELATIONSHIP BETWEEN TWO VARIABLES | 27 |
| Exa 4.7 | CONTINGENCY TABLES | 28 |
| Exa 5.2.1 | Classical Probability | 32 |
| Exa 5.2.2 | Empirical Probability | 32 |
| Exa 5.3.1 | Special Rule of Addition | 33 |
| Exa 5.3.2 | Complement Rule | 33 |
| Exa 5.3.3 | General Rule of Addition | 34 |
| Exa 5.4.1 | Special Rule of Multiplication | 35 |
| Exa 5.4.2 | General Rule of Multiplication | 35 |
| Exa 5.5 | CONTINGENCY TABLES | 35 |
| Exa 5.6 | BAYES THEOREM | 36 |
| Exa 5.7.1 | Multiplication Formula | 37 |
| Exa 5.7.2 | Permutation | 37 |
| Exa 5.7.3 | Permutation | 38 |
| Exa 5.7.4 | Combination | 38 |
| Exa 6.1 | PROBABILITY DISTRIBUTION | 39 |
| Exa 6.3 | MEAN AND VARIANCE OF A PROBABILITY DISTRIBUTION | 39 |
| Exa 6.4.1 | BINOMIAL PROBABILITY | 40 |
| Exa 6.4.2 | Binomial Probability Tables | 41 |
| Exa 6.4.3 | Cumulative Binomial Probability Distributions | 41 |
| Exa 6.5 | HYPERGEOMETRIC DISTRIBUTION | 42 |
| Exa 6.6.1 | POISSON DISTRIBUTION | 42 |
| Exa 6.6.2 | POISSON DISTRIBUTION | 42 |
| Exa 7.1 | UNIFORM DISTRIBUTION | 44 |
| Exa 7.3.1 | STANDARD NORMAL PROBABILITY DISTRIBUTION | 46 |
| Exa 7.3.2 | Empirical Rule | 46 |
| Exa 7.3.3 | Areas under the Normal Curve | 47 |
| Exa 7.3.4 | Areas under the Normal Curve | 47 |
| Exa 7.3.5 | Areas under the Normal Curve | 48 |
| Exa 7.3.6 | Areas under the Normal Curve | 48 |
| Exa 7.3.7 | Empirical Rule | 49 |
| Exa 7.5.1 | EXPONENTIAL DISTRIBUTION | 49 |
| Exa 7.5.2 | EXPONENTIAL DISTRIBUTION | 50 |
| Exa 8.1 | Simple Random Sampling | 51 |
| Exa 8.2 | SAMPLING ERROR | 51 |

Exa 8.3 SAMPLING DISTRIBUTION OF THE SAMPLE MEAN 52
Exa 8.4 THE CENTRAL LIMIT THEOREM . . . . . . . . . . 53
Exa 8.5 SAMPLING DISTRIBUTION OF THE SAMPLE MEAN 55
Exa 9.2.1 Population Standard Deviation . . . . . . . . . . . . . 57
Exa 9.2.2 Computer Simulation . . . . . . . . . . . . . . . . . . . 58
Exa 9.2.3 Population Standard Deviation . . . . . . . . . . . . . 59
Exa 9.2.4 Population Mean . . . . . . . . . . . . . . . . . . . . . 60
Exa 9.3 CONFIDENCE INTERVAL FOR A POPULATION PROPORTION . . . . . . . . . . . . . . . . . . . . . . . . . . 61
Exa 9.4.1 Sample Size to Estimate a Population Mean . . . . . . 62
Exa 9.4.2 Sample Size to Estimate a Population Proportion . . . 63
Exa 9.5 FINITE POPULATION CORRECTION FACTOR . . 63
Exa 10.4 Two Tailed Test . . . . . . . . . . . . . . . . . . . . . . 65
Exa 10.6.1 TESTING A MEAN . . . . . . . . . . . . . . . . . . . . 66
Exa 10.6.2 TESTING A MEAN . . . . . . . . . . . . . . . . . . . . 67
Exa 10.7 TYPE II ERROR . . . . . . . . . . . . . . . . . . . . 68
Exa 11.1 TWO SAMPLE TEST . . . . . . . . . . . . . . . . . . . 70
Exa 11.2.1 Two Sample Pooled Test . . . . . . . . . . . . . . . . . 70
Exa 11.2.2 Unequal Population Standard Deviations . . . . . . . 72
Exa 11.3 PAIRED t TEST . . . . . . . . . . . . . . . . . . . . . 73
Exa 12.1 Testing a Hypothesis of Equal Population Variances . 74
Exa 12.2.1 ANOVA . . . . . . . . . . . . . . . . . . . . . . . . . . 74
Exa 12.2.2 ANOVA Test . . . . . . . . . . . . . . . . . . . . . . . 75
Exa 12.4 TWO WAY ANALYSIS OF VARIANCE . . . . . . . 75
Exa 13.1 CORRELATION ANALYSIS . . . . . . . . . . . . . . . 77
Exa 13.2.1 CORRELATION COEFFICIENT . . . . . . . . . . . 78
Exa 13.2.2 t TEST FOR THE CORRELATION COEFFICIENT 80
Exa 13.3 REGRESSION ANALYSIS . . . . . . . . . . . . . . . 81
Exa 13.6 Constructing Confidence and Prediction Intervals . . . 82
Exa 13.7 TRANSFORMING DATA . . . . . . . . . . . . . . . . 83
Exa 14.1 MULTIPLE REGRESSION ANALYSIS . . . . . . . . 86
Exa 14.4 Multicollinearity . . . . . . . . . . . . . . . . . . . . . . 87
Exa 14.6 REGRESSION MODELS WITH INTERACTION . . 88
Exa 14.8 REVIEW OF MULTIPLE REGRESSION . . . . . . . 89
Exa 15.1 TEST A HYPOTHESIS OF A POPULATION PROPORTION . . . . . . . . . . . . . . . . . . . . . . . . . . 91
Exa 15.2 TWO SAMPLE TESTS ABOUT PROPORTIONS . . 92
Exa 15.3.1 Hypothesis Test of Equal Expected Frequencies . . . . 94

Exa 15.3.2 Hypothesis Test of Unequal Expected Frequencies . . . 95
Exa 15.5 TESTING THE HYPOTHESIS THAT A DISTRIBUTION IS NORMAL . . . . . . . . . . . . . . . . . . . 96
Exa 15.6 CONTINGENCY TABLE ANALYSIS . . . . . . . . . 97
Exa 16.1.1 THE SIGN TEST . . . . . . . . . . . . . . . . . . . . 99
Exa 16.1.2 Normal Approximation to the Binomial . . . . . . . . 100
Exa 16.2 TESTING A HYPOTHESIS ABOUT A MEDIAN . . 101
Exa 16.3 WILCOXON SIGNED RANK TEST FOR DEPENDENT POPULATIONS . . . . . . . . . . . . . . . . . . . . . 101
Exa 16.4 WILCOXON RANK SUM TEST FOR INDEPENDENT POPULATIONS . . . . . . . . . . . . . . . . . . . . . 102
Exa 16.5 KRUSKAL WALLIS TEST . . . . . . . . . . . . . . 104
Exa 16.6 RANK ORDER CORRELATION . . . . . . . . . . . 105
Exa 17.1.1 SIMPLE INDEX NUMBERS . . . . . . . . . . . . . . 107
Exa 17.1.2 SIMPLE INDEX NUMBERS . . . . . . . . . . . . . . 107
Exa 17.1.3 SIMPLE INDEX NUMBERS . . . . . . . . . . . . . . 108
Exa 17.3.1 WEIGHTED INDEXES . . . . . . . . . . . . . . . . . 109
Exa 17.3.2 Paasche Price Index . . . . . . . . . . . . . . . . . . . 110
Exa 17.3.3 Fishers Ideal Index . . . . . . . . . . . . . . . . . . . 110
Exa 17.3.4 Value Index . . . . . . . . . . . . . . . . . . . . . . . . 111
Exa 17.4 SPECIAL PURPOSE INDEXES . . . . . . . . . . . . 111
Exa 17.5.1 USING AN INDEX AS A DEFLATOR . . . . . . . . 112
Exa 17.5.2 USING AN INDEX TO FIND PURCHASING POWER 112
Exa 17.5.3 Shifting the Base . . . . . . . . . . . . . . . . . . . . . 113
Exa 18.3 WEIGHTED MOVING AVERAGE . . . . . . . . . . 115
Exa 18.4 Least Squares Method . . . . . . . . . . . . . . . . . . 117
Exa 18.6 SEASONAL VARIATION . . . . . . . . . . . . . . . . 118
Exa 18.7 Deseasonalized Data to Forecast . . . . . . . . . . . . 119
Exa 18.8 DURBIN WATSON STATISTIC . . . . . . . . . . . . 120
Exa 19.3 SOURCES OF VARIATION . . . . . . . . . . . . . . 122
Exa 19.4.1 PURPOSE AND TYPES OF QUALITY CONTROL CHARTS . . . . . . . . . . . . . . . . . . . . . . . . . 123
Exa 19.4.2 Control Charts for Variables . . . . . . . . . . . . . . 125
Exa 19.6.1 p Charts . . . . . . . . . . . . . . . . . . . . . . . . . . 126
Exa 19.6.2 cBar Charts . . . . . . . . . . . . . . . . . . . . . . . . 128
Exa 19.7 ACCEPTANCE SAMPLING . . . . . . . . . . . . . . 129

# Chapter 2

# Describing Data

**R code Exa 2.2** GRAPHIC PRESENTATION OF QUALITATIVE DATA

```
#Page No.23
ratings<-c("Awesome", "Excellent", "Good", "Poor")
frequency<-c(102, 58, 30, 10)

percentages<-(frequency/sum(frequency))*100

cat("The ease of navigation is measured on an ordinal scale, ranked from 'Poor' to 'Awesome'.\n")

barplot(frequency,names.arg=ratings,col=c("yellow", "skyblue", "orange", "lightgreen"),main="Ease of Navigation - Bar Chart",ylab="Frequency",xlab="Ease of Navigation Ratings")

labels_with_percentages <- paste0(ratings, " (", round(percentages, 1), "%)")
pie(frequency,labels=labels_with_percentages,col=c("yellow", "skyblue", "orange", "lightgreen"),main="Ease of Navigation - Pie Chart with Percentages")
```

**R code Exa 2.3** FREQUENCY DISTRIBUTIONS

```
#Page No.26
profit_data <- c(1387, 2148, 2201, 963, 820, 2230, 3043, 2584, 2370,
                 1754, 2207, 996, 1298, 1266, 2341, 1059, 2666, 2637,
                 1817, 2252, 2813, 1410, 1741, 3292, 1674, 2991, 1426,
                 1040, 1428, 323, 1553, 1772, 1108, 1807, 934, 2944,
                 1273, 1889, 352, 1648, 1932, 1295, 2056, 2063, 2147,
                 1529, 1166, 482, 2071, 2350, 1344, 2236, 2083, 1973,
                 3082, 1320, 1144, 2116, 2422, 1906, 2928, 2856, 2502,
                 1951, 2265, 1485, 1500, 2446, 1952, 1269, 2989, 783,
                 2692, 1323, 1509, 1549, 369, 2070, 1717, 910, 1538,
                 1206, 1760, 1638, 2348, 978, 2454, 1797, 1536, 2339,
                 1342, 1919, 1961, 2498, 1238, 1606, 1955, 1957, 2700,
                 443, 2357, 2127, 294, 1818, 1680, 2199, 2240, 2222,
                 754, 2866, 2430, 1115, 1824, 1827, 2482, 2695, 2597,
                 1621, 732, 1704, 1124, 1907, 1915, 2701, 1325, 2742,
                 870, 1464, 1876, 1532, 1938, 2084, 3210, 2250, 1837,
                 1174, 1626, 2010, 1688, 1940, 2639,
```

```
377, 2279, 2842,
                    1412, 1762, 2165, 1822, 2197, 842, 1220, 2626, 2434,
                    1809, 1915, 2231, 1897, 2646, 1963, 1401, 1501, 1640,
                    2415, 2119, 2389, 2445, 1461, 2059, 2175, 1752, 1821,
                    1546, 1766, 335, 2886, 1731, 2338, 1118, 2058, 2487)

breaks_fixed <- seq(200, 3400, by = 400)

freq_table <- table(cut(profit_data, breaks_fixed, right = FALSE))

print(freq_table)

par(mfrow = c(1, 2))

hist(profit_data, breaks = breaks_fixed, main = "Histogram of Vehicle Profits",
     xlab = "Profit", ylab = "Frequency", col = "lightblue", border = "black")

barplot(freq_table,
        main = "Bar Chart of Vehicle Profits",
        xlab = "Profit Range", ylab = "Frequency", col = "lightgreen")
```

---

**R code Exa 2.4.1** Histogram

```
#Page No.33
profit_intervals <- c("200-600", "600-1000", "1000-1400",
                      "1400-1800", "1800-2200", "
```

```
                          2200-2600",
                       "2600-3000", "3000-3400")
frequencies <- c(8, 11, 23, 38, 45, 32, 19, 4)

data <- data.frame(
  Interval = profit_intervals,
  Frequency = frequencies
)

print(data, row.names = FALSE)

barplot(frequencies, names.arg = profit_intervals,
        main = "Bar Chart of Vehicle Profits",
        xlab = "Profit Range", ylab = "Frequency",
        col = "lightgreen", border = "black")
```

**R code Exa 2.4.2** Cumulative Distributions

```
#Page No.38
profit_intervals <- c("200-600", "600-1000", "
   1000-1400", "1400-1800",
                       "1800-2200", "2200-2600", "
                          2600-3000", "3000-3400")
frequencies <- c(8, 11, 23, 38, 45, 32, 19, 4)
upper_limits <- c(600, 1000, 1400, 1800, 2200, 2600,
    3000, 3400)

cum_freq <- cumsum(frequencies)
total <- sum(frequencies)
cum_rel_freq <- cum_freq / total

table1 <- data.frame(
  Profit = paste0("< $", upper_limits),
  Cumulative_Frequency = cum_freq,
  Found_by = sapply(1:length(frequencies), function(
```

```
    i) paste(frequencies[1:i], collapse = " + "))
)

cat("Table 1: Cumulative Frequency Table\n")
print(table1)

table2 <- data.frame(
  Profit = paste0("< $", upper_limits),
  Cumulative_Frequency = cum_freq,
  Cumulative_Relative_Frequency = round(cum_rel_freq, 4),
  Percentage = paste0(round(cum_rel_freq * 100, 1), "%")
)

cat("\nTable 2: Cumulative Relative Frequency Table\n")
print(table2)

par(mfrow = c(1, 2))
plot(upper_limits, cum_freq, type = "o", col = "blue", xlab = "Profit ($)",
     ylab = "Cumulative Frequency", main = "Cumulative Frequency Polygon", pch = 16, lwd = 2)

plot(upper_limits, cum_rel_freq, type = "o", col = "green", xlab = "Profit ($)",
     ylab = "Cumulative Relative Frequency", main = "Cumulative Relative Frequency Polygon", pch = 16, lwd = 2)

profit_60th <- 1600
profit_75_percentile <- 2300

cat("\nProfit earned on 60 vehicles is less than $", profit_60th, "\n")
cat("75% of vehicles earned a profit of less than $"
```

```
, profit_75_percentile, "\n")
```

# Chapter 3

# Describing Data

**R code Exa 3.1.1** Population Mean

```
#Page No.53

data <- c(11, 4, 10, 4, 9, 3, 8, 10, 3, 14, 1, 10,
   3, 5, 2, 2, 5, 6, 1, 2, 2, 3, 7, 1, 3, 7, 8, 10,
   1, 4, 7, 5, 2, 2, 5, 1, 1, 3, 3, 1, 2, 1)

mean_value <- mean(data)

rounded_mean_value <- round(mean_value,2)

print(rounded_mean_value)
```

**R code Exa 3.1.2** Sample Mean

```
#Page No.55

data <- c(90,77,94,89,119,112,91,110,92,100,113,83)

```

```
mean_value <- mean(data)

print(mean_value)
```

**R code Exa 3.1.3** Median

```
#Page No.58

facebook_hours <- c(3, 5, 7, 5, 9, 1, 3, 9, 17, 10)

median_value <- median(facebook_hours)

print(median_value)
```

**R code Exa 3.1.4** Mode

```
#Page No.60

distance_data <- c(11, 4, 10, 4, 9, 3, 8, 10, 3, 14,
    1, 10, 3, 5, 2, 2, 5, 6, 1, 2, 2, 3, 7, 1, 3, 7,
    8, 10, 1, 4, 7, 5, 2, 2, 5, 1, 1, 3, 3, 1, 2, 1)

find_mode <- function(x) {
  freq_table <- table(x)
  max_freq <- max(freq_table)
  mode_values <- as.numeric(names(freq_table[freq_
     table == max_freq]))
  return(mode_values)
}

mode_value <- find_mode(distance_data)

print(mode_value)
```

**R code Exa 3.1.5** Software Solution

```
#Page No.64
profits <- c(1387, 2148, 2201, 963, 820, 2230, 3043, 2584, 2370,
             1754, 2207, 996, 1298, 1266, 2341, 1059, 2666, 2637,
             1817, 2252, 2813, 1410, 1741, 3292, 1674, 2991, 1426,
             1040, 1428, 323, 1553, 1772, 1108, 1807, 934, 2944,
             1273, 1889, 352, 1648, 1932, 1295, 2056, 2063, 2147,
             1529, 1166, 482, 2071, 2350, 1344, 2236, 2083, 1973,
             3082, 1320, 1144, 2116, 2422, 1906, 2928, 2856, 2502,
             1951, 2265, 1485, 1500, 2446, 1952, 1269, 2989, 783,
             2692, 1323, 1509, 1549, 369, 2070, 1717, 910, 1538,
             1206, 1760, 1638, 2348, 978, 2454, 1797, 1536, 2339,
             1342, 1919, 1961, 2498, 1238, 1606, 1955, 1957, 2700,
             443, 2357, 2127, 294, 1818, 1680, 2199, 2240, 2222,
             754, 2866, 2430, 1115, 1824, 1827, 2482, 2695, 2597,
             1621, 732, 1704, 1124, 1907, 1915, 2701, 1325, 2742,
             870, 1464, 1876, 1532, 1938, 2084, 3210, 2250, 1837,
             1174, 1626, 2010, 1688, 1940, 2639,
```

```
                    377, 2279, 2842,
                 1412, 1762, 2165, 1822, 2197, 842, 1220, 2626, 2434,
                 1809, 1915, 2231, 1897, 2646, 1963, 1401, 1501, 1640,
                 2415, 2119, 2389, 2445, 1461, 2059, 2175, 1752, 1821,
                 1546, 1766, 335, 2886, 1731, 2338, 1118, 2058, 2487)

mean_profit <- mean(profits)
cat("Mean Profit: $", round(mean_profit, 2), "\n")

median_profit <- median(profits)
cat("Median Profit: $", round(median_profit, 2), "\n")
```

**R code Exa 3.2** WEIGHTED MEAN

```
#Page No.65

num_employee_1 <- 14
num_employee_2 <- 10
num_employee_3 <- 2

rate_1 <- 16.50
rate_2 <- 19.00
rate_3 <- 25.00

total_1 <- num_employee_1 * rate_1
total_2 <- num_employee_2 * rate_2
total_3 <- num_employee_3 * rate_3

total_payment <- total_1 + total_2 + total_3

```

```
total_employees <- num_employee_1 + num_employee_2 + num_employee_3

mean_hourly_rate <- total_payment / total_employees

print(mean_hourly_rate)

rounded_mean_hourly_rate <- round(mean_hourly_rate,2)

print(rounded_mean_hourly_rate)
```

**R code Exa 3.3.1** GEOMETRIC MEAN

```
#Page No.67
returns <- c(0.30, 0.20, -0.40, 2.00)
growth_factors <- 1 + returns

geo_mean <- prod(growth_factors)^(1/length(returns)) - 1
geo_mean_percent <- round(geo_mean * 100,1)
cat("Geometric Mean Rate of Return is", geo_mean_percent, "%")
```

**R code Exa 3.3.2** GEOMETRIC MEAN

```
# Page No.68
value_start <- 258295
value_end <- 613599

n_years <- 2014 - 1990

```

```r
geo_mean <- (value_end / value_start)^(1 / n_years) - 1
geo_mean_percent <- round(geo_mean * 100, 2)
cat("Average annual percent increase is", geo_mean_percent, "%")
```

**R code Exa 3.4.1** Range

```r
#Page no.70
baton_rouge <- c(48, 52)
tucson <- c(40, 60)

range_baton_rouge <- diff(baton_rouge)
cat("Range for Baton Rouge plant:", range_baton_rouge, "monitors\n")

range_tucson <- diff(tucson)
cat("Range for Tucson plant:", range_tucson, " monitors\n")
```

**R code Exa 3.4.2** Variance

```r
#Page No.71
orange <- c(20, 40, 50, 60, 80)
ontario <- c(20, 45, 50, 55, 80)

stats <- function(x) c(mean = mean(x), median = median(x), range = diff(range(x)), variance = sum((x - mean(x))^2) / length(x))
list(Orange = stats(orange), Ontario = stats(ontario))
```

**R code Exa 3.4.3** Population Variance

```
#Page No.74
citations <- c(19, 17, 22, 18, 28, 34, 45, 39, 38, 44, 34, 10)

mean_citations <- mean(citations)
population_variance <- sum((citations - mean_citations)^2) / length(citations)

list(mean = mean_citations, variance = population_variance)
```

**R code Exa 3.4.4** Sample Variance

```
#Page No.77
wages <- c(12, 20, 16, 18, 19)

mean_wages <- mean(wages)
sample_variance <- sum((wages - mean_wages)^2) / (length(wages) - 1)

list(mean = mean_wages, variance = sample_variance)
```

**R code Exa 3.4.5** Sample Standard Deviation

```
#Page No.78
sample_variance <- 10

```

```
sample_sd <- round(sqrt(sample_variance),2)

sample_sd
```

**R code Exa 3.5.1** Chebyshev Theorem

```
#Page No.80
k <- 3.5
percentage <- 1 - 1 / (k^2)
round (percentage * 100, 0)
```

**R code Exa 3.5.2** Empirical Rule

```
#Page No.81
mean <- 500
std_dev <- 20

range_68 <- c(mean - 1 * std_dev, mean + 1 * std_dev
    )
range_95 <- c(mean - 2 * std_dev, mean + 2 * std_dev
    )
range_997 <- c(mean - 3 * std_dev, mean + 3 * std_
    dev)

cat("68% of the rentals are between:", range_68[1],
    "and", range_68[2], "\n")
cat("95% of the rentals are between:", range_95[1],
    "and", range_95[2], "\n")
cat("99.7% of the rentals are between:", range_
    997[1], "and", range_997[2], "\n")

#The answer may vary due to difference in
    representation.
```

**R code Exa 3.6.1** Arithmetic Mean of Grouped Data

```r
#Page no.82
profit_intervals <- data.frame(
  Lower = c(200, 600, 1000, 1400, 1800, 2200, 2600, 3000),
  Upper = c(600, 1000, 1400, 1800, 2200, 2600, 3000, 3400),
  Frequency = c(8, 11, 23, 38, 45, 32, 19, 4)
)

profit_intervals$Midpoint <- (profit_intervals$Lower + profit_intervals$Upper) / 2

profit_intervals$fx <- profit_intervals$Frequency * profit_intervals$Midpoint

total_fx <- sum(profit_intervals$fx)
total_freq <- sum(profit_intervals$Frequency)

mean_profit <- total_fx / total_freq

cat("Profit Distribution with Midpoints & f*M:\n")
print(profit_intervals[, c("Lower", "Upper", "Frequency", "Midpoint", "fx")], row.names = FALSE)

cat("\nTotal (fM): $", total_fx, "\nTotal Vehicles: ", total_freq, "\n")
cat("Arithmetic Mean Profit per Vehicle: $", round(mean_profit, 2))
```

**R code Exa 3.6.2** Standard Deviation of Grouped Data

```r
#Page No.83
profit_intervals <- data.frame(
  Lower = c(200, 600, 1000, 1400, 1800, 2200, 2600, 3000),
  Upper = c(600, 1000, 1400, 1800, 2200, 2600, 3000, 3400),
  Frequency = c(8, 11, 23, 38, 45, 32, 19, 4)
)

profit_intervals$Midpoint <- (profit_intervals$Lower + profit_intervals$Upper) / 2

profit_intervals$fx <- profit_intervals$Frequency * profit_intervals$Midpoint

total_fx <- sum(profit_intervals$fx)
total_freq <- sum(profit_intervals$Frequency)
mean_profit <- total_fx / total_freq

profit_intervals$Deviation_Squared <- (profit_intervals$Midpoint - mean_profit)^2

profit_intervals$fx2 <- profit_intervals$Frequency * profit_intervals$Deviation_Squared

total_fx2 <- sum(profit_intervals$fx2)
std_dev <- sqrt(total_fx2 / (total_freq - 1))

cat("      Profit Distribution with Computed Deviations:\n")
print(profit_intervals[, c("Lower", "Upper", "Frequency", "Midpoint", "fx", "Deviation_Squared", "fx2")], row.names = FALSE)

cat("\nTotal  (fM): $", total_fx, "\nTotal  f (M-x)^2: ", total_fx2, "\nTotal Vehicles: ", total_
```

```
freq, "\n")

cat("      Arithmetic Mean Profit per Vehicle: $", round(mean_profit, 2), "\n")
cat("      Standard Deviation of Profit: $", round(std_dev, 2), "\n\n")
```

# Chapter 4

# Describing Data

**R code Exa 4.1** DOT PLOTS

```
#Page No.95
tionesta <- c(23, 33, 27, 28, 39, 26, 30, 32, 28,
   33, 35, 32, 29, 25, 36, 31, 32, 27, 35, 32, 35,
   37, 36, 30)
sheffield <- c(31, 35, 44, 36, 34, 37, 30, 37, 43,
   31, 40, 31, 32, 44, 36, 34, 43, 36, 26, 38, 37,
   30, 42, 33)

vehicles <- c(tionesta, sheffield)
dealership <- rep(c(1, 2), each = 24)

dotchart(vehicles,
         groups = dealership,
         main = "Number of Vehicles Serviced at
            Tionesta and Sheffield Dealerships",
         xlab = "Number of Vehicles Serviced",
         col = c("blue", "red")[dealership],
         pch = 16)

cat("Summary Statistics for Tionesta:\n")
print(summary(tionesta))
```

```
cat("\nSummary Statistics for Sheffield:\n")
print(summary(sheffield))
```

**R code Exa 4.2** STEM AND LEAF DISPLAYS

```
#Page No.98
attendance <- c(96, 93, 88, 117, 127, 95, 113, 96,
    108, 94, 148, 156,
                139, 142, 94, 107, 125, 155, 155,
                    103, 112, 127, 117, 120,
                112, 135, 132, 111, 125, 104, 106,
                    139, 134, 119, 97, 89,
                118, 136, 125, 143, 120, 103, 113,
                    124, 138)

cat("Stem-and-Leaf Plot:\n")
stem(attendance)

cat("\nSmallest Attendance:", min(attendance))
cat("\nLargest Attendance:", max(attendance))
```

**R code Exa 4.3** Quartiles Deciles and Percentiles

```
#Page No.103
commissions <- c(2038, 1758, 1721, 1637, 2097, 2047,
     2205, 1787, 2287,
                 1940, 2311, 2054, 2406, 1471, 1460)

sorted_commissions <- sort(commissions)

n <- length(sorted_commissions)

```

```
median_index <- (n + 1) * 50 / 100
median_value <- sorted_commissions[median_index]

q1_index <- (n + 1) * 25 / 100
q1_value <- sorted_commissions[q1_index]

q3_index <- (n + 1) * 75 / 100
q3_value <- sorted_commissions[q3_index]

cat("Sorted Commissions:\n", sorted_commissions, "\n")

cat("\nMedian (50th percentile):", median_value)
cat("\nFirst Quartile (Q1 - 25th percentile):", q1_value)
cat("\nThird Quartile (Q3 - 75th percentile):", q3_value)
```

**R code Exa 4.4.1** BOX PLOTS

```
#Page No.107
delivery_times <- c(13, 15, 18, 22, 30)

boxplot(delivery_times, horizontal = TRUE, col = "lightgreen", main = "Delivery Time Box Plot",
        xlab = "Minutes", ylim = c(12, 32))

axis(1, at = seq(12, 32, by = 2))

text(13, 1.2, "Minimum\nvalue", pos = 3, cex = 0.8)
text(15, 1.2, expression(Q[1]), pos = 3, cex = 0.8)
text(18, 1.2, "Median", pos = 3, cex = 0.8)
text(22, 1.2, expression(Q[3]), pos = 3, cex = 0.8)
text(30, 1.2, "Maximum\nvalue", pos = 3, cex = 0.8)
```

**R code Exa 4.4.2** BOX PLOTS

```
#Page No.108
ages <- c(
  21, 23, 24, 25, 26, 27, 27, 28, 28, 29, 29, 30, 30, 30, 31, 31, 31, 31, 31,
  32, 32, 32, 32, 33, 33, 34, 34, 34, 35, 35, 35, 36, 36, 37, 37, 37, 37, 37, 37,
  38, 38, 39, 39, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 41, 41, 41, 41, 41, 41,
  42, 42, 42, 42, 42, 42, 42, 42, 43, 43, 43, 43, 44, 44, 44, 44, 44, 44, 44, 44,
  44, 45, 45, 45, 45, 45, 45, 45, 45, 46, 46, 46, 46, 46, 46, 46, 46, 46, 46, 47,
  47, 47, 47, 47, 48, 48, 48, 48, 48, 48, 48, 49, 49, 49, 49, 49, 49, 50, 50, 50,
  50, 50, 51, 51, 51, 51, 51, 51, 52, 52, 52, 52, 52, 52, 52, 52, 53, 53, 53, 53,
  54, 54, 54, 54, 55, 55, 55, 55, 55, 56, 56, 56, 56, 56, 57, 57, 57, 57, 58, 58,
  58, 58, 58, 59, 59, 60, 61, 61, 62, 62, 63, 64, 65, 65, 65, 68, 69, 70, 72, 72, 73
)

boxplot(ages, horizontal = TRUE, col = "lightblue", main = "Box Plot of Age of Buyers",
        xlab = "Age", ylim = c(20, 75))
axis(1, at = seq(20, 75, by = 5))

Q1 <- quantile(ages, 0.25)
Q2 <- median(ages)
Q3 <- quantile(ages, 0.75)

IQR_val <- IQR(ages)
```

```
Lower_bound <- Q1 - 1.5 * IQR_val
Upper_bound <- Q3 + 1.5 * IQR_val

Min <- min(ages[ages >= Lower_bound])
Max <- max(ages[ages <= Upper_bound])

outliers <- ages[ages < Lower_bound | ages > Upper_
   bound]

text(Min, 1.2, "Minimum\nvalue", pos = 3, cex = 0.8)
text(Q1, 1.2, expression(Q[1]), pos = 3, cex = 0.8)
text(Q2, 1.2, "Median", pos = 3, cex = 0.8)
text(Q3, 1.2, expression(Q[3]), pos = 3, cex = 0.8)
text(Max, 1.2, "Maximum\nvalue", pos = 3, cex = 0.8)

cat("Outliers Detected (Age):", outliers, "\n")
```

**R code Exa 4.5** SKEWNESS

```
#Page No.111
library(e1071)

data <- c(0.09, 0.13, 0.41, 0.51, 1.12, 1.20, 1.49,
   3.18, 3.50,
          6.36, 7.83, 8.92, 10.13, 12.99, 16.40)

mean_value <- mean(data)
median_value <- median(data)
std_dev <- sd(data)

pearson_skewness <- 3 * (mean_value - median_value)
   / std_dev

software_skewness <- skewness(data, type = 1)

```

```
cat("Mean:", round(mean_value, 2), "\n")
cat("Median:", round(median_value, 2), "\n")
cat("Standard Deviation:", round(std_dev, 2), "\n")
cat("Pearson s Coefficient of Skewness:", round(pearson_skewness, 3), "\n")
cat("Software Method Skewness:", round(software_skewness, 3), "\n")

if (pearson_skewness > 0) {
  cat("The distribution is Positively Skewed (Right Skewed).\n")
} else if (pearson_skewness < 0) {
  cat("The distribution is Negatively Skewed (Left Skewed).\n")
} else {
  cat("The distribution is Symmetric.\n")
}

#The answer may vary due to difference in representation.
```

**R code Exa 4.6** DESCRIBING THE RELATIONSHIP BETWEEN TWO VARIABLES

```
#Page No.115
age <- c(21,23,24,25,26,27,27,28,28,29,29,30,30,30,31,31,31,31,31,32,32,3
profit <- c(1387,1754,1817,1040,1273,1529,3082,1951,2692,1342,1206,443,1621,

min_length <- min(length(age), length(profit))

age <- age[1:min_length]
```

```
profit <- profit[1:min_length]

plot(age, profit,
     main = "Profit and Age of Buyer at Applewood
        Auto Group",
     xlab = "Age (Years)",
     ylab = "Profit per Vehicle ($)",
     pch = 16,
     col = "green",
     xlim = c(0, 80),
     ylim = c(0, 3500)
     )
```

**R code Exa 4.7** CONTINGENCY TABLES

```
#Page No.117
profit <- c(1387, 1754, 1817, 1040, 1273, 1529,
   3082, 1951, 2692, 1342, 1206, 443, 1621, 754,
   1174,
            2415, 1412, 870, 1809, 2207, 1546, 2252,
                2148, 1889, 1428, 1320, 1166, 2265,
               1919, 1323,
            1761, 2357, 2866, 1464, 1761, 1626,
               1915, 2119, 732, 1766, 2201, 2813,
               996, 1961, 1509,
            2430, 1144, 323, 1638, 1485, 352, 482,
               2127, 2389, 2165, 1876, 2231, 2010,
               1704, 1553,
            963, 1298, 2071, 335, 2116, 1410, 1648,
               1500, 2348, 2498, 1549, 1532, 1897,
               294, 1115,
            2445, 1822, 1124, 1688, 2886, 1266,
               1932, 2422, 820, 1772, 2350, 2446,
               1741, 369, 1238,
            1818, 1907, 1940, 1938, 978, 2646, 1824,
```

```
        2197, 1461, 1731, 3292, 2230, 2341, 1952, 2070,
        1344, 1295, 1108, 1906, 2454, 1827, 1680, 1915, 2084, 1606, 2639, 3043, 2059, 1963, 842,
        2338, 1674, 1059, 1807, 2928, 2056, 2236, 1269, 1797, 1955, 1717, 2701, 2482, 2199, 3210,
        1220, 1401, 377, 2175, 2991, 1118, 2666, 2584, 2063, 2083, 2856, 934, 2989, 2695, 1957,
        1536, 2240, 910, 1325, 2250, 2626, 2279, 2058, 1752, 2637, 1501, 2370, 1426, 2944, 2147,
        1973, 2502, 783, 1538, 2339, 2700, 2597, 2222, 2742, 1837, 2842, 2434, 1640, 1821, 2487)

location <- c("Tionesta", "Sheffield", "Sheffield", "Sheffield", "Kane", "Sheffield", "Kane", "Kane",
        "Tionesta", "Kane", "Sheffield", "Kane", "Sheffield", "Olean", "Kane", "Kane",
        "Sheffield", "Tionesta", "Tionesta", "Sheffield", "Sheffield", "Tionesta", "Tionesta",
        "Olean", "Kane", "Tionesta", "Olean", "Olean", "Tionesta", "Olean", "Kane", "Kane",
        "Kane", "Olean", "Olean", "Tionesta", "Tionesta", "Kane", "Olean", "Sheffield",
        "Sheffield", "Tionesta", "Kane", "Sheffield", "Kane", "Tionesta", "Tionesta", "Kane",
        "Sheffield", "Sheffield", "Sheffield", "Olean", "Olean", "Kane", "Tionesta", "Kane",
```

```
"Tionesta", "Tionesta", "Sheffield", "Tionesta", "Kane", "Tionesta", "Kane", "Olean",
"Kane", "Kane", "Olean", "Tionesta", "Tionesta", "Tionesta", "Kane", "Tionesta",
"Sheffield", "Kane", "Kane", "Kane", "Kane", "Tionesta", "Kane", "Olean", "Olean",
"Tionesta", "Kane", "Kane", "Olean", "Sheffield", "Olean", "Olean", "Olean",
"Sheffield", "Kane", "Olean", "Kane", "Kane", "Kane", "Tionesta", "Olean", "Sheffield",
"Kane", "Tionesta", "Olean", "Tionesta", "Sheffield", "Tionesta", "Kane", "Sheffield",
"Sheffield", "Sheffield", "Kane", "Kane", "Tionesta", "Kane", "Tionesta", "Tionesta",
"Olean", "Sheffield", "Kane", "Sheffield", "Sheffield", "Kane", "Tionesta",
"Sheffield", "Kane", "Tionesta", "Kane", "Sheffield", "Tionesta", "Tionesta", "Kane",
"Olean", "Sheffield", "Sheffield", "Olean", "Tionesta", "Olean", "Olean", "Tionesta",
"Olean", "Olean", "Tionesta", "Sheffield", "Tionesta", "Olean", "Kane", "Sheffield",
"Olean", "Sheffield", "Tionesta", "Kane", "Sheffield", "Kane", "Olean", "Sheffield",
"Olean", "Sheffield", "Sheffield", "Sheffield", "Kane", "Kane", "
```

```
                    Sheffield", "Sheffield",
                    "Tionesta", "Sheffield", "Olean", "Olean", "Kane", "Olean", "Sheffield", "Olean",
                    "Olean", "Kane", "Sheffield", "Kane", "Tionesta", "Sheffield", "Kane", "Olean", "Olean",
                    "Tionesta", "Olean")

median_profit <- median(profit)
profit_category <- ifelse(profit > median_profit, "Above Median", "Below Median")
contingency_table <- addmargins(table(profit_category, location))

print(contingency_table)
```

# Chapter 5

# A Survey of Probability Concepts

**R code Exa 5.2.1** Classical Probability

```
#Page No.136
outcomes <- 1:6

even_numbers <- outcomes[outcomes %% 2 == 0]
prob_even <- length(even_numbers) / length(outcomes)

cat("Probability of rolling an even number:", prob_
    even, "\n")
```

---

**R code Exa 5.2.2** Empirical Probability

```
#Page No.138
total_flights <- 113
successful_flights <- 111

prob_success <- round(successful_flights / total_
    flights, 2)
```

```
cat("Probability of a successful space mission:", prob_success)
```

**R code Exa 5.3.1** Special Rule of Addition

```
#Page No.142
P_A <- 0.025
P_C <- 0.075

P_AorC <- P_A + P_C

cat("Probability that a package is either underweight or overweight:", P_AorC)
```

**R code Exa 5.3.2** Complement Rule

```
#Page No.143
library(VennDiagram)

P_A <- 0.025
P_C <- 0.075
P_B <- 1 - (P_A + P_C)

cat("Probability of a satisfactory bag:", P_B, "\n")

draw.pairwise.venn(area1 = P_A, area2 = P_C, cross.area = 0,
                   category = c("A", "C"),
                   fill = c("red", "blue"))

grid.text("not (A or C)\n0.90", x = 0.6, y = 0.2, gp = gpar(fontsize = 12, col = "black"))
```

**R code Exa 5.3.3** General Rule of Addition

```r
#Page No.145
library(VennDiagram)

P_A <- 4/52
P_B <- 13/52
P_AandB <- 1/52

P_AorB <- P_A + P_B - P_AandB

cat("P(A or B) =", P_AorB, "\n")

venn.plot <- draw.pairwise.venn(
  area1 = P_A*52,
  area2 = P_B*52,
  cross.area = P_AandB*52,
  category = c("Kings", "Hearts"),
  fill = c("brown", "seagreen3"),
  alpha = 0.5,
  cat.pos = c(-30, 30),
  cat.dist = c(0.03, 0.03),
  label.col = "black",
  cat.col = c("black", "black"),
  fontface = "bold",
  cex = 1.5,
  cat.cex = 1.2
)

grid.draw(venn.plot)

# The answer may vary due to difference in
   representation
```

**R code Exa 5.4.1** Special Rule of Multiplication

```
#Page No.148
P_R1 <- 0.60
P_R2 <- 0.60

P_both <- P_R1 * P_R2
P_R1_not_R2 <- P_R1 * (1 - P_R2)
P_not_R1_R2 <- (1 - P_R1) * P_R2
P_neither <- (1 - P_R1) * (1 - P_R2)

cat("P(R1 and R2) =", P_both, "\n")
cat("P(R1 and not R2) =", P_R1_not_R2, "\n")
cat("P(not R1 and R2) =", P_not_R1_R2, "\n")
cat("P(not R1 and not R2) =", P_neither, "\n")
cat("Total Probability =", P_both + P_R1_not_R2 + P_
   not_R1_R2 + P_neither, "\n")
```

**R code Exa 5.4.2** General Rule of Multiplication

```
#Page No.149
W1 <- 9/12
W2 <- 8/11

white <- round(W1 * W2, 2)

cat("P(W1 and W2) =", white)
```

**R code Exa 5.5** CONTINGENCY TABLES

```r
#Page No.151
N <- 500
M6 <- 50
M2 <- 75 + 200
A60 <- 175
M6_A60 <- 30

P_M6 <- M6 / N
cat("P(M6) =", P_M6, "\n")

P_M2 <- M2 / N
cat("P(M2) =", P_M2, "\n")

P_M6_A60 <- (M6 / N) + (A60 / N) - (M6_A60 / N)
cat("P(M6 OR A60) =", P_M6_A60, "\n")

P_M6_given_A60 <- M6_A60 / A60
cat("P(M6 | A60) =", P_M6_given_A60, "\n")

P_M6_AND_A60 <- (M6 / N) * (M6_A60 / M6)
cat("P(M6 AND A60) =", P_M6_AND_A60, "\n")

P_M6_given_A30 <- 5 / 100  # From table: 5 out of
    100 in this category
cat("P(M6 | A30) =", P_M6_given_A30, "\n")

if (P_M6_given_A60 != P_M6_given_A30) {
  cat("Conclusion: Age and movie attendance are NOT
     independent.\n")
} else {
  cat("Conclusion: Age and movie attendance are
     independent.\n")
}
```

**R code Exa 5.6** BAYES THEOREM

```
#Page No.158
P_A1 <- 0.45
P_A2 <- 0.30
P_A3 <- 0.25

P_B1_A1 <- 0.03
P_B1_A2 <- 0.06
P_B1_A3 <- 0.04

P_B1 <- (P_B1_A1 * P_A1) + (P_B1_A2 * P_A2) + (P_B1_
    A3 * P_A3)
P_A2_B1 <- (P_B1_A2 * P_A2) / P_B1

cat("P(A2 | B1) =", round(P_A2_B1, 4))
```

**R code Exa 5.7.1** Multiplication Formula

```
#Page No.162
m <- 3
n <- 2

total_combinations <- m * n

cat("Total number of different vehicles:", total_
    combinations)
```

**R code Exa 5.7.2** Permutation

```
#Page No.164
n <- 3
r <- 3

perm <- factorial(n) / factorial(n - r)
```

```r

cat("Total number of ways to assemble the parts:", perm)
```

**R code Exa 5.7.3** Permutation

```r
#Page No.164
n <- 8
r <- 3

perm <- factorial(n) / factorial(n - r)

cat("Total number of ways to arrange the video segments:", perm, "\n")
```

**R code Exa 5.7.4** Combination

```r
#Page No.165
n <- 7
r <- 3

comb <- factorial(n) / (factorial(r) * factorial(n - r))

cat("Total number of different teams:", comb, "\n")
```

# Chapter 6

# Discrete Probability Distributions

**R code Exa 6.1** PROBABILITY DISTRIBUTION

```
#Page No.177
outcomes <- c(0, 1, 2, 3)
probabilities <- c(1/8, 3/8, 3/8, 1/8)

prob_dist <- data.frame(Number_of_Heads = outcomes, Probability = probabilities)
print(prob_dist)

barplot(probabilities, names.arg = outcomes, col = "lightgreen",
        main = "Probability Distribution of Number of Heads",
        xlab = "Number of Heads", ylab = "Probability",
        ylim = c(0, 0.5))  # Set y-axis limit to 0.5
```

**R code Exa 6.3** MEAN AND VARIANCE OF A PROBABILITY DISTRIBUTION

```
#Page No.181
x <- c(0, 1, 2, 3, 4)
P_x <- c(0.1, 0.2, 0.3, 0.3, 0.1)

mean_value <- sum(x * P_x)

variance_value <- sum(((x - mean_value)^2) * P_x)

std_dev <- sqrt(variance_value)

cat("Mean (Expected Value) =", mean_value)
cat("Variance =", variance_value)
cat("Standard Deviation =", round(std_dev,3))
```

**R code Exa 6.4.1** BINOMIAL PROBABILITY

```
#Page No.186
n <- 5
p <- 0.20

probabilities <- dbinom(0:5, size = n, prob = p)

data.frame(X = 0:5, Probability = probabilities)

barplot(probabilities, names.arg = 0:5, col = "green
   ",
        main = "Probability Distribution for the
           Number of Late Flights",
        xlab = "Number of Late Flights", ylab = "
           Probability")
```

**R code Exa 6.4.2** Binomial Probability Tables

```r
#Page No.187
n <- 6
p <- 0.05

probabilities <- dbinom(0:6, size = n, prob = p)

data.frame(X = 0:6, Probability = round(
   probabilities, 6))

mean_value <- n * p
variance_value <- n * p * (1 - p)

cat("Mean (Expected Value):", mean_value)
cat("Variance:", variance_value)
```

**R code Exa 6.4.3** Cumulative Binomial Probability Distributions

```r
#Page No.191
n <- 12
p <- 0.762

prob_x7 <- dbinom(7, n, p)
prob_x_geq_7 <- sum(dbinom(7:12, n, p))

cat("P(X = 7):", round(prob_x7, 4))
cat("P(X    7):", round(prob_x_geq_7, 4))
```

**R code Exa 6.5** HYPERGEOMETRIC DISTRIBUTION

```r
#Page No.195
N <- 50
S <- 40
n <- 5
x <- 4

prob_x4 <- dhyper(x, S, N - S, n)

cat("P(X = 4):", round(prob_x4, 3))
```

**R code Exa 6.6.1** POISSON DISTRIBUTION

```r
#Page No.198
mu <- 20 / 500

p_0 <- dpois(0, mu)
p_at_least_1 <- 1 - p_0

cat("Mean number of lost bags per flight:", round(mu, 4), "\n")
cat("P(X = 0):", round(p_0, 4), "\n")
cat("P(X    1):", round(p_at_least_1, 4), "\n")
```

**R code Exa 6.6.2** POISSON DISTRIBUTION

```r
#Page No.200
mu <- 30 * 0.05

p_0 <- dpois(0, mu)

p_at_least_1 <- 1 - p_0
```

```
cat("Expected number of hurricanes in 30 years:", mu, "\n")
cat("P(X = 0):", round(p_0, 4), "\n")
cat("P(X    1):", round(p_at_least_1, 4), "\n")
```

# Chapter 7

# Continuous Probability Distributions

**R code Exa 7.1** UNIFORM DISTRIBUTION

```
#Page No.211
library(ggplot2)

min_time <- 0
max_time <- 30
height <- 1 / (max_time - min_time)
height <- 1 / (max_time - min_time)
area <- height * (max_time - min_time)
mean_wait <- (min_time + max_time) / 2
std_dev <- sqrt((max_time - min_time)^2 / 12)
p_more_than_25 <- height * (max_time - 25)
p_between_10_20 <- height * (20 - 10)

cat("Height of uniform distribution:", height, "\n")
cat("Area of uniform distribution:", area, "\n")
cat("Mean wait time:", mean_wait, "minutes\n")
cat("Standard deviation of wait times:", round(std_
   dev, 2), "minutes\n")
cat("P(wait > 25):", round(p_more_than_25, 4), "\n")
```

```r
cat("P(10 < wait < 20):", round(p_between_10_20, 4), "\n")


x_vals <- seq(min_time, max_time, by = 0.1)
y_vals <- rep(height, length(x_vals))
df <- data.frame(x = x_vals, y = y_vals)

ggplot(df, aes(x, y)) +
  geom_line(size = 1.2, color = "blue") +
  geom_area(fill = "lightblue", alpha = 0.5) +
  ggtitle("Uniform Probability Distribution of Waiting Time") +
  xlab("Waiting Time (minutes)") + ylab("Probability Density") +
  theme_minimal()

ggplot(df, aes(x, y)) +
  geom_line(size = 1.2, color = "blue") +
  geom_area(data = subset(df, x > 25), aes(x, y), fill = "orange", alpha = 0.5) +
  ggtitle("Probability of Waiting More Than 25 Minutes") +
  xlab("Waiting Time (minutes)") + ylab("Probability Density") +
  theme_minimal()

ggplot(df, aes(x, y)) +
  geom_line(size = 1.2, color = "blue") +
  geom_area(data = subset(df, x > 10 & x < 20), aes(x, y), fill = "green", alpha = 0.5) +
  ggtitle("Probability of Waiting Between 10 and 20 Minutes") +
  xlab("Waiting Time (minutes)") + ylab("Probability Density") +
  theme_minimal()
```

**R code Exa 7.3.1** STANDARD NORMAL PROBABILITY DISTRIBUTION

```
#Page No.218
mu <- 1000
sigma <- 100
x1 <- 1100
x2 <- 900

z1 <- (x1 - mu) / sigma
z2 <- (x2 - mu) / sigma

cat("Z-score for $1100 income:", z1)
cat("Z-score for $900 income:", z2)
```

---

**R code Exa 7.3.2** Empirical Rule

```
#Page No.219
mu <- 19.0
sigma <- 1.2

range_68 <- c(mu - 1*sigma, mu + 1*sigma)
range_95 <- c(mu - 2*sigma, mu + 2*sigma)
range_99_7 <- c(mu - 3*sigma, mu + 3*sigma)

cat("68% of batteries fail between:", range_68, "
    hours")
cat("95% of batteries fail between:", range_95, "
    hours")
cat("Practically all batteries fail between:", range
    _99_7, "hours")
```

---

**R code Exa 7.3.3** Areas under the Normal Curve

```
#Page No.221
mu <- 1000
sigma <- 100

z_1100 <- (1100 - mu) / sigma
p_1000_to_1100 <- pnorm(z_1100) - pnorm(0)

p_less_1100 <- pnorm(z_1100)

cat("Probability of earning between $1000 and $1100:
  ", round(p_1000_to_1100, 4))
cat("Probability of earning less than $1100:", round
  (p_less_1100, 4))
```

---

**R code Exa 7.3.4** Areas under the Normal Curve

```
#Page No.222
mu <- 1000
sigma <- 100

z_790 <- (790 - mu) / sigma
p_790_to_1000 <- pnorm(0) - pnorm(z_790)

p_less_790 <- pnorm(z_790)

cat("Probability of earning between $790 and $1000:"
  , round(p_790_to_1000, 4))
cat("Probability of earning less than $790:", round(
  p_less_790, 4))
```

---

**R code Exa 7.3.5** Areas under the Normal Curve

```
#Page No.224
mu <- 1000
sigma <- 100

z_840 <- (840 - mu) / sigma
z_1200 <- (1200 - mu) / sigma

p_840_to_mean <- pnorm(0) - pnorm(z_840)
p_mean_to_1200 <- pnorm(z_1200) - pnorm(0)

p_840_to_1200 <- p_840_to_mean + p_mean_to_1200

cat("Probability of earning between $840 and $1,200:
    ", round(p_840_to_1200, 4))
#The answer may slightly vary due to rounding off
    values
```

---

**R code Exa 7.3.6** Areas under the Normal Curve

```
#Page No.225
mu <- 1000
sigma <- 100

z_1150 <- (1150 - mu) / sigma
z_1250 <- (1250 - mu) / sigma

p_1000_to_1250 <- pnorm(z_1250) - pnorm(0)
p_1000_to_1150 <- pnorm(z_1150) - pnorm(0)

p_1150_to_1250 <- p_1000_to_1250 - p_1000_to_1150
```

```
cat("Probability of earning between $1,150 and $1,250:", round(p_1150_to_1250, 4))
```

**R code Exa 7.3.7** Empirical Rule

```
#Page No.227
mu <- 67900
sigma <- 2050
p <- 0.04

z <- qnorm(p)

x_min <- mu + (z * sigma)

cat("Minimum guaranteed mileage:", round(x_min, 0), "miles")
#The answer may slightly vary due to rounding off values
```

**R code Exa 7.5.1** EXPONENTIAL DISTRIBUTION

```
#Page No.235
lambda <- 1/20

p_less_5 <- 1 - exp(-lambda * 5)

p_less_40 <- 1 - exp(-lambda * 40)
p_more_40 <- 1 - p_less_40

cat("P(Arrival < 5 seconds):", round(p_less_5, 4))
cat("P(Arrival > 40 seconds):", round(p_more_40, 4))
```

**R code Exa 7.5.2** EXPONENTIAL DISTRIBUTION

```
#Page No.237
mean_time_to_failure <- 4000
lambda_rate <- 1 / mean_time_to_failure
probability_of_failure <- 0.05

x <- log(1 - probability_of_failure) / -lambda_rate

cat("The warranty period should be set at
    approximately", round(x, 2), "hours.\n")
```

# Chapter 8

# Sampling Methods and the Central Limit Theorem

**R code Exa 8.1** Simple Random Sampling

```
#Page No.253
june_rentals <- c(0, 2, 3, 2, 3, 4, 2, 3, 4, 7,
                  3, 4, 4, 4, 7, 0, 5, 3, 6, 2,
                  3, 2, 3, 6, 0, 4, 1, 1, 3, 3)

sampled_days <- sample(june_rentals, size = 5,
   replace = TRUE)

cat("Randomly selected sample of 5 nights:", sampled
   _days)
#The answer may vary due to difference in
   representation.
```

**R code Exa 8.2** SAMPLING ERROR

```
#Page No.259
```

```
rentals <- c(0, 2, 3, 2, 3, 4, 2, 3, 4, 7,
             3, 4, 4, 4, 7, 0, 5, 3, 6, 2,
             3, 2, 3, 6, 0, 4, 1, 1, 3, 3)

population_mean <- mean(rentals)
cat("Population Mean (  ):", population_mean)

sample1 <- sample(rentals, 5, replace = FALSE)
sample2 <- sample(rentals, 5, replace = FALSE)
sample3 <- sample(rentals, 5, replace = FALSE)

sample_mean1 <- mean(sample1)
sample_mean2 <- mean(sample2)
sample_mean3 <- mean(sample3)

error1 <- sample_mean1 - population_mean
error2 <- sample_mean2 - population_mean
error3 <- sample_mean3 - population_mean

cat("Sample 1:", sample1, "\nSample Mean 1:", sample_mean1, "Sampling Error 1:", error1)
cat("Sample 2:", sample2, "\nSample Mean 2:", sample_mean2, "Sampling Error 2:", error2)
cat("Sample 3:", sample3, "\nSample Mean 3:", sample_mean3, "Sampling Error 3:", error3)

#The answer may vary due to difference in representation.
```

**R code Exa 8.3** SAMPLING DISTRIBUTION OF THE SAMPLE MEAN

```
#Page No.261
population <- c(14, 14, 16, 16, 14, 16, 18)

population_mean <- round(mean(population),2)
```

```
cat("1. Population Mean ( ):", population_mean)

samples <- combn(population, 2)

sample_means <- colMeans(samples)

sampling_distribution <- round((table(sample_means) / length(sample_means)),4)

cat("2. Sampling Distribution of the Sample Mean for Samples of Size 2:")
print(sampling_distribution)

sampling_mean <- round(mean(sample_means),2)
cat("3. Mean of the Sampling Distribution ( x ):", sampling_mean)

cat("4. Observations:")
cat("- The mean of the sampling distribution is equal to the population mean.")
cat("- The spread of the sample means is less than the population spread.")
cat("- The sampling distribution tends to be more symmetric compared to the population distribution.")

hist(sample_means, breaks=5, col="blue", main="Sampling Distribution of Sample Means",
     xlab="Sample Mean", ylab="Frequency", border="black")
```

**R code Exa 8.4** THE CENTRAL LIMIT THEOREM

```
#Page No.266
population <- c(11, 4, 18, 2, 1, 2, 0, 2, 2, 4,
```

```r
                  3, 4, 1, 2, 2, 3, 3, 19, 8, 3,
                  7, 1, 0, 2, 7, 0, 4, 5, 1, 14,
                  16, 8, 9, 1, 1, 2, 5, 10, 2, 3)

population_mean <- mean(population)
cat("Population Mean (  ):", round(population_mean, 2))

hist(population, breaks = 10, col = "lightblue", border = "black",
     main = "Population Distribution of Years of Service",
     xlab = "Years of Service", ylab = "Frequency")

sample_means <- function(population, sample_size, num_samples = 25) {
  means <- numeric(num_samples)
  for (i in 1:num_samples) {
    sample <- sample(population, size = sample_size, replace = FALSE)
    means[i] <- mean(sample)
  }
  return(means)
}

sample_means_5 <- sample_means(population, sample_size = 5)

hist(sample_means_5, breaks = 10, col = "lightgreen", border = "black",
     main = "Sampling Distribution of Sample Mean (n=5)",
     xlab = "Sample Mean", ylab = "Frequency")

sample_means_20 <- sample_means(population, sample_size = 20)

hist(sample_means_20, breaks = 10, col = "lightcoral
```

```
", border = "black",
    main = "Sampling Distribution of Sample Mean (n=20)",
    xlab = "Sample Mean", ylab = "Frequency")

mean_sample_means_5 <- mean(sample_means_5)
mean_sample_means_20 <- mean(sample_means_20)

cat("Mean of Sample Means (n=5):", round(mean_sample_means_5, 2))
cat("Mean of Sample Means (n=20):", round(mean_sample_means_20, 2))
```

**R code Exa 8.5** SAMPLING DISTRIBUTION OF THE SAMPLE MEAN

```
#Page No.274
mu <- 31.2
sigma <- 0.4
n <- 16
x_bar <- 31.38

SE <- sigma / sqrt(n)

z_score <- (x_bar - mu) / SE

p_value <- 1 - pnorm(z_score)

cat("Z-Score:", round(z_score, 2))
cat("P-value:", round(p_value, 4))

if (p_value < 0.05) {
  cat("Conclusion: Since the probability is less than 5%, it is unlikely that the sample mean would be this high by random chance. The process is likely overfilling the bottles.")
```

```
} else {
  cat("Conclusion: The sample mean is not significantly different from the population mean. The filling process is working as expected.")
}

x <- seq(-3, 3, length=100)
y <- dnorm(x)
plot(x, y, type="l", lwd=2, col="blue", main="Standard Normal Distribution",
     xlab="Z-score", ylab="Density")
abline(v = z_score, col="red", lwd=2, lty=2)

#The answer may vary due to difference in representation.
```

# Chapter 9

# Estimation and Confidence Intervals

**R code Exa 9.2.1** Population Standard Deviation

```
#Page No.287
sample_mean <- 45420
sigma <- 2050
n <- 49
confidence_level <- 0.95

cat("The population mean is unknown. The best
   estimate is the sample mean:", sample_mean)

std_error <- sigma / sqrt(n)
z_score <- qnorm((1 + confidence_level) / 2)
margin_of_error <- z_score * std_error
lower_limit <- sample_mean - margin_of_error
upper_limit <- sample_mean + margin_of_error

cat("The 95% confidence interval is ($", round(lower
   _limit, 2), ", $", round(upper_limit, 2), ")")

cat("We are 95% confident that the true population
```

```
mean lies between $",
  round(lower_limit, 2), " and $", round(upper_limit, 2))
cat("If we repeated this process many times, about 95% of the confidence intervals would contain the true mean.")

#The answer may slightly vary due to rounding off values.
```

**R code Exa 9.2.2** Computer Simulation

```
#Page No.289
set.seed(123)

population_mean <- 50
population_sd <- 5
sample_size <- 30
num_samples <- 60
z_value <- 1.96

standard_error <- population_sd / sqrt(sample_size)

sample_means <- numeric(num_samples)
lower_bounds <- numeric(num_samples)
upper_bounds <- numeric(num_samples)
contains_mean <- logical(num_samples)

for (i in 1:num_samples) {
  sample <- rnorm(sample_size, mean = population_mean, sd = population_sd)
  sample_means[i] <- mean(sample)
  margin_of_error <- z_value * standard_error
  lower_bounds[i] <- sample_means[i] - margin_of_error
```

```
  upper_bounds[i] <- sample_means[i] + margin_of_error
  contains_mean[i] <- (lower_bounds[i] <= population_mean) & (upper_bounds[i] >= population_mean)
}

num_containing_mean <- sum(contains_mean)
num_not_containing_mean <- num_samples - num_containing_mean

excluded_intervals <- which(!contains_mean)

cat("Total samples:", num_samples)
cat("Confidence intervals containing 50:", num_containing_mean, "(", round((num_containing_mean / num_samples) * 100, 2), "% )")
cat("Confidence intervals NOT containing 50:", num_not_containing_mean, "(", round((num_not_containing_mean / num_samples) * 100, 2), "% )")
cat("Indices of excluded intervals:", excluded_intervals)

results_df <- data.frame(
  Sample = 1:num_samples,
  Sample_Mean = sample_means,
  Lower_Bound = lower_bounds,
  Upper_Bound = upper_bounds,
  Contains_50 = contains_mean
)

print(results_df)

#The answer may vary due to difference in representation.
```

**R code Exa 9.2.3** Population Standard Deviation

```
#Page No.294
sample_mean <- 0.32
sample_sd <- 0.09
sample_size <- 10
df <- sample_size - 1

t_value <- 2.262

margin_of_error <- t_value * (sample_sd / sqrt(sample_size))

lower_bound <- sample_mean - margin_of_error
upper_bound <- sample_mean + margin_of_error

cat("95% Confidence Interval for Population Mean:")
cat("(", round(lower_bound, 3), ",", round(upper_bound, 3), ")")

if (lower_bound <= 0.30 & upper_bound >= 0.30) {
  cat("Since 0.30 is within the interval, it is reasonable to conclude that the population mean could be 0.30.")
} else {
  cat("Since 0.30 is NOT within the interval, it is unlikely that the population mean is 0.30.")
}
```

---

**R code Exa 9.2.4** Population Mean

```
#Page No.296
sample_mean <- 49.348
sample_sd <- 9.012
sample_size <- 20
```

```r
df <- sample_size - 1

t_value <- 2.093

margin_of_error <- t_value * (sample_sd / sqrt(sample_size))

lower_bound <- sample_mean - margin_of_error
upper_bound <- sample_mean + margin_of_error

cat("95% Confidence Interval for Population Mean:")
cat("(", round(lower_bound, 3), ",", round(upper_bound, 3), ")")

if (lower_bound <= 50 & upper_bound >= 50) {
  cat("Since $50 is within the interval, it is reasonable to conclude that the population mean could be $50.")
} else {
  cat("Since $50 is NOT within the interval, it is unlikely that the population mean is $50.")
}

if (lower_bound <= 60 & upper_bound >= 60) {
  cat("Since $60 is within the interval, it is reasonable to conclude that the population mean could be $60.")
} else {
  cat("Since $60 is NOT within the interval, it is unlikely that the population mean is $60.")
}
```

**R code Exa 9.3** CONFIDENCE INTERVAL FOR A POPULATION PROPORTION

```r
#Page No.301
sample_size <- 2000
success_count <- 1600
sample_proportion <- success_count / sample_size

z_value <- 1.96

SE <- sqrt(sample_proportion * (1 - sample_proportion) / sample_size)

lower_bound <- sample_proportion - z_value * SE
upper_bound <- sample_proportion + z_value * SE

cat("95% Confidence Interval for Population Proportion:")
cat("(", round(lower_bound, 3), ",", round(upper_bound, 3), ")")

if (lower_bound > 0.75) {
  cat("Since the lower bound is greater than 0.75, the merger proposal will likely pass.")
} else {
  cat("Since the lower bound is less than or equal to 0.75, the merger proposal may not pass.")
}
```

**R code Exa 9.4.1** Sample Size to Estimate a Population Mean

```r
#Page No.305
sigma <- 1000
E <- 100

z_95 <- 1.96
z_99 <- 2.576

```

```
n_95 <- ( (z_95 * sigma) / E )^2

n_99 <- ( (z_99 * sigma) / E )^2

cat("Sample size required for 95% confidence:", n_95)
cat("Sample size required for 99% confidence:", round(n_99,2))

increase <- ((n_99 / n_95) * 100) - 100
cat("Percentage increase in sample size when confidence level increases from 95% to 99%:", round(increase, 2), "%")
```

**R code Exa 9.4.2** Sample Size to Estimate a Population Proportion

```
#Page No.306
E <- 0.10
z <- 1.645
p <- 0.5

n <- (p * (1 - p)) * (z / E)^2

cat("Sample size required for 90% confidence:", round(n,2), "\n")
```

**R code Exa 9.5** FINITE POPULATION CORRECTION FACTOR

```
#Page No.308
x_bar <- 450
s <- 75
N <- 250
n <- 40
```

```
df <- n - 1
t_value <- 1.685

FPC <- sqrt((N - n) / (N - 1))

SE <- (s / sqrt(n)) * FPC

margin_of_error <- t_value * SE
lower_bound <- x_bar - margin_of_error
upper_bound <- x_bar + margin_of_error

cat("The best estimate we have of the population mean is the sample mean, which is $", x_bar)
cat("The endpoints of the confidence interval are $", round(lower_bound, 2), " and $", round(upper_bound, 2))
cat("It is likely that the population mean is more than $", round(lower_bound, 2), " but less than $", round(upper_bound, 2))
```

# Chapter 10

# One Sample Tests of Hypothesis

**R code Exa 10.4** Two Tailed Test

```
#Page No.327
mu <- 200
sigma <- 16
n <- 50
x_bar <- 203.5
alpha <- 0.01

z_critical <- qnorm(1 - alpha/2)
z <- (x_bar - mu) / (sigma / sqrt(n))

cat("Decision rule: If z is not between -", round(z_critical, 3), " and ", round(z_critical, 3), ", reject H0.")
cat("Computed z-value:", round(z, 3))
if (abs(z) > z_critical) {
  cat("Since the computed z-value (", round(z, 3), ") is outside the range of -", round(z_critical, 3), " to ", round(z_critical, 3), ", we reject H0.")
```

```
} else {
  cat("Since the computed z-value (", round(z, 3), "
    ) is within the range of -", round(z_critical,
    3), " to ", round(z_critical, 3), ", we do not
    reject H0.")
}
```

**R code Exa 10.6.1** TESTING A MEAN

```
#Page No.334
claims <- c(45, 49, 62, 40, 43, 61, 48, 53, 67, 63,
   78, 64,
             48, 54, 51, 56, 63, 69, 58, 51, 58, 59,
                56, 57, 38, 76)
mu_0 <- 60

t_test_result <- t.test(claims, mu = mu_0,
   alternative = "less", conf.level = 0.99)
print(t_test_result)

x_bar <- mean(claims)
s <- sd(claims)
n <- length(claims)
df <- n - 1

t_stat <- (x_bar - mu_0) / (s / sqrt(n))
cat("Computed t-Statistic:", t_stat)

t_critical <- qt(0.01, df)
cat("Critical t-Value:", t_critical)

if (t_stat < t_critical) {
  decision <- "Reject H0: There is enough evidence
     to say the mean cost is less than $60."
} else {
```

```
  decision <- "Fail to reject H0: There is not enough evidence to say the mean cost is less than $60."
}

cat("Decision:", decision)
cat("Interpretation: Since the computed t-statistic (", t_stat, ") is greater than the critical value (", t_critical,
    "), we fail to reject H0. This means the difference of", round(abs(x_bar - mu_0), 3),
    "between the sample mean and the population mean could be due to sampling error.")
```

**R code Exa 10.6.2** TESTING A MEAN

```
#Page No.337
parking_times <- c(30, 24, 28, 22, 14, 2, 39, 23, 23, 28, 12, 31)
mu_0 <- 15

t_test_result <- t.test(parking_times, mu = mu_0, alternative = "greater", conf.level = 0.95)
print(t_test_result)

x_bar <- mean(parking_times)
s <- sd(parking_times)
n <- length(parking_times)
df <- n - 1

t_stat <- (x_bar - mu_0) / (s / sqrt(n))
cat("Computed t-Statistic:", t_stat)

t_critical <- qt(0.05, df, lower.tail = FALSE)
cat("Critical t-Value:", t_critical)
```

```r

if (t_stat > t_critical) {
  decision <- "Reject H0: There is enough evidence to say the mean time in the lot is more than 15 minutes."
} else {
  decision <- "Fail to reject H0: There is not enough evidence to say the mean time in the lot is more than 15 minutes."
}

cat("Decision:", decision)
cat("Interpretation: Since the computed t-statistic (", round(t_stat,3), ") is greater than the critical value (", round(t_critical,3),
    "), we reject H0. This means the mean time spent in the lot is significantly greater than 15 minutes, and the airport may need to add more parking places.")
```

**R code Exa 10.7** TYPE II ERROR

```r
#Page No.343
mu0 <- 10000
sigma <- 400
n <- 100
xc_lower <- 9922
xc_upper <- 10078

calculate_beta <- function(mu1, xc, sigma, n) {
  z <- (xc - mu1) / (sigma / sqrt(n))
  beta <- pnorm(z)
  return(beta)
}

```

```
mu1_9880 <- 9880
beta_9880 <- calculate_beta(mu1_9880, xc_lower, sigma, n)
cat("Probability of Type II Error ( ) for 1 = 9,880: ", 1-round(beta_9880, 4), "\n")

mu1_10100 <- 10100
beta_10100 <- calculate_beta(mu1_10100, xc_upper, sigma, n)
cat("Probability of Type II Error ( ) for 1 = 10,100: ", round(beta_10100, 4), "\n")
```

# Chapter 11

# Two Sample Tests of Hypothesis

**R code Exa 11.1** TWO SAMPLE TEST

```
#Page No.356
xS <- 5.50
xF <- 5.30
sigmaS <- 0.40
sigmaF <- 0.30
nS <- 50
nF <- 100

z_value <- (xS - xF) / sqrt((sigmaS^2 / nS) + (
    sigmaF^2 / nF))
p_value <- 1 - pnorm(z_value)

cat("Test Statistic (z-value):", round(z_value, 4))
cat("P-value:", format(p_value, scientific = FALSE,
    digits = 6))
```

**R code Exa 11.2.1** Two Sample Pooled Test

```r
#Page No.361
welles <- c(2, 4, 9, 3, 2)
atkins <- c(3, 7, 5, 8, 4, 3)

mean_welles <- mean(welles)
mean_atkins <- mean(atkins)

sd_welles <- sd(welles)
sd_atkins <- sd(atkins)

n_welles <- length(welles)
n_atkins <- length(atkins)

sp_squared <- (((n_welles - 1) * sd_welles^2) + ((n_atkins - 1) * sd_atkins^2)) / (n_welles + n_atkins - 2)
sp <- sqrt(sp_squared)

t_value <- (mean_welles - mean_atkins) / (sp * sqrt((1/n_welles) + (1/n_atkins)))
df <- n_welles + n_atkins - 2
p_value <- 2 * pt(abs(t_value), df = df, lower.tail = FALSE)

cat("Mean (Welles):", round(mean_welles, 4))
cat("Mean (Atkins):", round(mean_atkins, 4))
cat("Standard Deviation (Welles):", round(sd_welles, 4))
cat("Standard Deviation (Atkins):", round(sd_atkins, 4))
cat("Pooled Standard Deviation:", round(sp, 4))
cat("T-Statistic:", round(t_value, 4))
cat("Degrees of Freedom:", df)
cat("P-Value:", format(p_value, scientific = FALSE, digits = 6))

```

```r
t.test(welles, atkins, var.equal = TRUE)
```

**R code Exa 11.2.2** Unequal Population Standard Deviations

```r
#Page No.367
store_brand <- c(8, 8, 3, 1, 9, 7, 5, 5, 12)
name_brand <- c(12, 11, 10, 6, 8, 9, 9, 10, 11, 9,
   8, 10)

mean_store <- mean(store_brand)
mean_name <- mean(name_brand)

sd_store <- sd(store_brand)
sd_name <- sd(name_brand)

n_store <- length(store_brand)
n_name <- length(name_brand)

df <- ((sd_store^2 / n_store) + (sd_name^2 / n_name)
   )^2 /
  (((sd_store^2 / n_store)^2 / (n_store - 1)) + ((sd
     _name^2 / n_name)^2 / (n_name - 1)))
t_value <- (mean_store - mean_name) / sqrt((sd_store
   ^2 / n_store) + (sd_name^2 / n_name))
p_value <- 2 * pt(abs(t_value), df = df, lower.tail
   = FALSE)

cat("Mean (Store Brand):", round(mean_store, 4))
cat("Mean (Name Brand):", round(mean_name, 4))
cat("Standard Deviation (Store Brand):", round(sd_
   store, 4))
cat("Standard Deviation (Name Brand):", round(sd_
   name, 4))
cat("Degrees of Freedom:", round(df, 2))
cat("T-Statistic:", round(t_value, 4))
```

```r
cat("P-Value:", format(p_value, scientific = FALSE,
    digits = 6))

t.test(store_brand, name_brand, var.equal = FALSE)
```

**R code Exa 11.3** PAIRED t TEST

```r
#Page No.371
schadek <- c(235, 210, 231, 242, 205, 230, 231, 210,
     225, 249)
bowyer <- c(228, 205, 219, 240, 198, 223, 227, 215,
    222, 245)

differences <- schadek - bowyer
mean_d <- mean(differences)
sd_d <- sd(differences)
n <- length(differences)
df <- n - 1
t_value <- mean_d / (sd_d / sqrt(n))
p_value <- 2 * pt(abs(t_value), df = df, lower.tail
    = FALSE)

cat("Mean of Differences:", round(mean_d, 4))
cat("Standard Deviation of Differences:", round(sd_d
    , 4))
cat("Degrees of Freedom:", df)
cat("T-Statistic:", round(t_value, 4))
cat("P-Value:", format(p_value, scientific = FALSE,
    digits = 6))

t.test(schadek, bowyer, paired = TRUE)
```

# Chapter 12

# Analysis of Variance

**R code Exa 12.1** Testing a Hypothesis of Equal Population Variances

```
#Page No.389
us_route_25 <- c(52, 67, 56, 45, 70, 54, 64)
interstate_75 <- c(59, 60, 61, 51, 56, 63, 57, 65)

result <- var.test(us_route_25, interstate_75)

print(result)
#The answer may slightly vary due to rounding off
   values.
```

---

**R code Exa 12.2.1** ANOVA

```
#Page No.393
wolfe <- c(55, 54, 59, 56)
white <- c(66, 76, 67, 71)
korosa <- c(47, 51, 46, 48)

data <- data.frame(
```

```
  Customers = c(wolfe, white, korosa),
  Employee = rep(c("Wolfe", "White", "Korosa"), each
      = 4)
)

anova_result <- aov(Customers ~ Employee, data =
   data)
summary(anova_result)

#The answer may vary due to difference in
   representation.
```

**R code Exa 12.2.2** ANOVA Test

```
#Page No.396
Northern <- c(94, 90, 85, 80)
WTA <- c(75, 68, 77, 83, 88)
Pocono <- c(70, 73, 76, 78, 80, 68, 65)
Branson <- c(68, 70, 72, 65, 74, 65)

satisfaction_data <- data.frame(
  Airline = rep(c("Northern", "WTA", "Pocono", "
     Branson"), times = c(4, 5, 7, 6)),
  Satisfaction_Score = c(Northern, WTA, Pocono,
     Branson)
)

anova_result <- aov(Satisfaction_Score ~ Airline,
   data = satisfaction_data)
summary(anova_result)
```

**R code Exa 12.4** TWO WAY ANALYSIS OF VARIANCE

```r
#Page No.407
travel_time <- data.frame(
  Driver = rep(c("Deans", "Snaverly", "Ormson", "Zollaco", "Filbeck"), each = 4),
  Route = rep(c("US_6", "West_End", "Hickory_St", "Rte_59"), times = 5),
  Time = c(18, 17, 21, 22,
           16, 23, 23, 22,
           21, 21, 26, 22,
           23, 22, 29, 25,
           25, 24, 28, 28)
)

anova_result <- aov(Time ~ Route, data = travel_time)
summary(anova_result)
```

# Chapter 13

# Correlation and Linear Regression

**R code Exa 13.1** CORRELATION ANALYSIS

```
#Page No.438
library(ggplot2)

sales_data <- data.frame(
  Sales_Rep = c("Brian Virost", "Carlos Ramirez", "Carol Saia", "Greg Fish", "Jeff Hall",
                "Mark Reynolds", "Meryl Rumsey", "Mike Kiel", "Ray Snarsky", "Rich Niles",
                "Ron Broderick", "Sal Spina", "Soni Jones", "Susan Welch", "Tom Keller"),
  Sales_Calls = c(96, 40, 104, 128, 164, 76, 72, 80, 36, 84, 180, 132, 120, 44, 84),
  Copiers_Sold = c(41, 41, 51, 60, 61, 29, 39, 50, 28, 43, 70, 56, 45, 31, 30)
)

ggplot(sales_data, aes(x = Sales_Calls, y = Copiers_
```

```
    Sold)) +
  geom_point(color = "blue", size = 3) +
  labs(title = "Scatter Plot: Sales Calls vs Copiers
       Sold",
       x = "Sales Calls",
       y = "Copiers Sold")

cor_coeff <- cor(sales_data$Sales_Calls, sales_data$
    Copiers_Sold)
cat("Correlation Coefficient:", cor_coeff)

if (cor_coeff > 0) {
  cat("There is a positive correlation, meaning that
       as sales calls increase, copiers sold tend to
      increase.")
} else if (cor_coeff < 0) {
  cat("There is a negative correlation, meaning that
       as sales calls increase, copiers sold tend to
      decrease.")
} else {
  cat("There is no correlation between sales calls
      and copiers sold.")
}
```

**R code Exa 13.2.1** CORRELATION COEFFICIENT

```
#Page No.444
library(ggplot2)

age <- c(21, 23, 24, 25, 26, 27, 27, 28, 28, 29, 29,
     30, 30, 30, 31, 31, 31, 31, 31, 32, 32, 32, 32,
    33, 33, 34, 34, 34, 35, 35, 35, 36, 36, 37, 37,
    37, 37, 37, 37, 38, 38, 39, 39, 40, 40, 40, 40,
    40, 40, 40, 40, 40, 40, 41, 41, 41, 41, 41, 41,
    42, 42, 42, 42, 42, 42, 42, 42, 43, 43, 43, 43,
```

```
    44, 44, 44, 44, 44, 44, 44, 44, 44, 45, 45, 45,
    45, 45, 45, 45, 45, 46, 46, 46, 46, 46, 46, 46,
    46, 46, 46, 47, 47, 47, 47, 47, 48, 48, 48, 48,
    48, 48, 48, 49, 49, 49, 49, 49, 49, 50, 50, 50,
    50, 50, 51, 51, 51, 51, 51, 51, 52, 52, 52, 52,
    52, 52, 52, 52, 53, 53, 53, 53, 54, 54, 54, 54,
    55, 55, 55, 55, 55, 56, 56, 56, 56, 56, 57, 57,
    57, 57, 58, 58, 58, 58, 58, 59, 59, 60, 61, 61,
    62, 62, 63, 64, 65, 65, 65, 68, 69, 70, 72, 72,
    73)
profit <- c(1387, 1754, 1817, 1040, 1273, 1529,
    3082, 1951, 2692, 1342, 1206, 443, 1621, 754,
    1174, 2415, 1412, 870, 1809, 2207, 1546, 2252,
    2148, 1889, 1428, 1320, 1166, 2265, 1919, 1323,
    1761, 2357, 2866, 1464, 1761, 1626, 1915, 2119,
    732, 1766, 2201, 2813, 996, 1961, 1509, 2430,
    1144, 323, 1638, 1485, 352, 482, 2127, 2389,
    2165, 1876, 2231, 2010, 1704, 1553, 963, 1298,
    2071, 335, 2116, 1410, 1648, 1500, 2348, 2498,
    1549, 1532, 1897, 294, 1115, 2445, 1822, 1124,
    1688, 2886, 1266, 1932, 2422, 820, 1772, 2350,
    2446, 1741, 369, 1238, 1818, 1907, 1940, 1938,
    978, 2646, 1824, 2197, 1461, 1731, 3292, 2230,
    2341, 1952, 2070, 1344, 1295, 1108, 1906, 2454,
    1827, 1680, 1915, 2084, 1606, 2639, 3043, 2059,
    1963, 842, 2338, 1674, 1059, 1807, 2928, 2056,
    2236, 1269, 1797, 1955, 1717, 2701, 2482, 2199,
    3210, 1220, 1401, 377, 2175, 2991, 1118, 2666,
    2584, 2063, 2083, 2856, 934, 2989, 2695, 1957,
    1536, 2240, 910, 1325, 2250, 2626, 2279, 2058,
    1752, 2637, 1501, 2370, 1426, 2944, 2147, 1973,
    2502, 783, 1538, 2339, 2700, 2597, 2222, 2742,
    1837, 2842, 2434, 1640, 1821, 2487)

data <- data.frame(Age = age, Profit = profit)

ggplot(data, aes(x = Age, y = Profit)) +
  geom_point(color = "blue", size = 2) +
```

```
  labs(title = "Scatter Plot: Age vs Vehicle Profit",
       x = "Age of Buyer",
       y = "Profit on Vehicle Sale")

cor_coeff <- cor(data$Age, data$Profit)
cat("Correlation Coefficient:", round(cor_coeff,3))

if (cor_coeff > 0) {
  cat("There is a positive correlation, meaning that as the buyer's age increases, the profit on vehicle sales tends to increase.")
} else if (cor_coeff < 0) {
  cat("There is a negative correlation, meaning that as the buyer's age increases, the profit on vehicle sales tends to decrease.")
} else {
  cat("There is no correlation between buyer's age and vehicle profit.")
}
```

**R code Exa 13.2.2** t TEST FOR THE CORRELATION COEFFICIENT

```
#Page No.449
library(stats)

age <- c(21,23,24,25,26,27,27,28,28,29,29,30,30,30,31,31,31,31,31,32,32,3
profit <- c(1387,1754,1817,1040,1273,1529,3082,1951,2692,1342,1206,443,1621,

r <- round(cor(age, profit),3)
n <- length(age)
```

```
t_statistic <- round((r * sqrt(n - 2)) / sqrt(1 - r
   ^2),3)

alpha <- 0.05
df <- n - 2
t_critical <- round(qt(1 - alpha, df),3)

cat("Correlation coefficient (r):", r)
cat("Test statistic (t):", t_statistic)
cat("Critical t-value:", t_critical)
```

**R code Exa 13.3** REGRESSION ANALYSIS

```
#Page No.454
sales_calls <- c(96, 40, 104, 128, 164, 76, 72, 80,
   36, 84, 180, 132, 120, 44, 84)
copiers_sold <- c(41, 41, 51, 60, 61, 29, 39, 50,
   28, 43, 70, 56, 45, 31, 30)

x_mean <- mean(sales_calls)
y_mean <- mean(copiers_sold)

sx <- sd(sales_calls)
sy <- sd(copiers_sold)

r <- cor(sales_calls, copiers_sold)

b <- r * (sy / sx)

a <- y_mean - (b * x_mean)

cat("Regression Equation:    =", round(a, 4), "+",
   round(b, 4), "* x")

predicted_copiers <- a + b * 100
```

```
cat("Predicted Copiers Sold for 100 Calls:", round(
   predicted_copiers, 4))
```

**R code Exa 13.6** Constructing Confidence and Prediction Intervals

```
#Page No.469
library(ggplot2)

sales_calls <- c(96, 40, 104, 128, 164, 76, 72, 80,
   36, 84, 180, 132, 120, 44, 84)
copiers_sold <- c(41, 41, 51, 60, 61, 29, 39, 50,
   28, 43, 70, 56, 45, 31, 30)
data <- data.frame(sales_calls, copiers_sold)

model <- lm(copiers_sold ~ sales_calls, data = data)
summary(model)

a <- coef(model)[1]
b <- coef(model)[2]

x_new <- 50
y_hat <- a + b * x_new
cat("Expected sales for 50 calls:", y_hat)

n <- nrow(data)
x_mean <- mean(data$sales_calls)
SSE <- sum(residuals(model)^2)
s_yx <- sqrt(SSE / (n - 2))
SSX <- sum((data$sales_calls - x_mean)^2)
t_val <- qt(0.975, df = n - 2)

conf_width <- t_val * s_yx * sqrt(1/n + (x_new - x_
   mean)^2 / SSX)
pred_width <- t_val * s_yx * sqrt(1 + 1/n + (x_new -
    x_mean)^2 / SSX)
```

```
cat("95% Confidence Interval: [", y_hat - conf_width, ",", y_hat + conf_width, "]")
cat("95% Prediction Interval: [", y_hat - pred_width, ",", y_hat + pred_width, "]")

ggplot(data, aes(x = sales_calls, y = copiers_sold)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red", se = TRUE) +
  labs(title = "Sales Calls vs Copiers Sold", x = "Sales Calls", y = "Copiers Sold") +
  theme_minimal()

#The answer may slightly vary due to rounding off values.
```

---

**R code Exa 13.7** TRANSFORMING DATA

```
#Page No.472
library(ggplot2)

price <- c(0.50, 1.35, 0.79, 1.71, 1.38, 1.22, 1.03, 1.84, 1.73, 1.62,
           0.76, 1.79, 1.57, 1.27, 0.96, 0.52, 0.64, 1.05, 0.72, 0.75)
sales <- c(181, 33, 91, 13, 34, 47, 73, 11, 15, 20,
           91, 13, 22, 34, 74, 164, 129, 55, 107, 119)

data <- data.frame(price, sales)

model <- lm(sales ~ price, data = data)
summary(model)
```

```
ggplot(data, aes(x = price, y = sales)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Price vs. Sales", x = "Price ($)", y = "Sales") +
  theme_minimal()

r_squared <- summary(model)$r.squared
correlation <- sqrt(r_squared) * sign(coef(model)[2])
cat("Correlation Coefficient:", correlation)

log_sales <- log10(sales)
data$log_sales <- log_sales

log_model <- lm(log_sales ~ price, data = data)
summary(log_model)

ggplot(data, aes(x = price, y = log_sales)) +
  geom_point(color = "green") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Price vs. Log(Sales)", x = "Price ($)", y = "Log(Sales)") +
  theme_minimal()

x_new <- 1.25
log_pred <- predict(log_model, newdata = data.frame(price = x_new))
predicted_sales <- 10^log_pred  # Antilog to get actual sales
cat("Predicted Sales at $1.25:", round(predicted_sales), "bottles")

#The answer may slightly vary due to rounding off values.
```

# Chapter 14

# Multiple Regression Analysis

**R code Exa 14.1** MULTIPLE REGRESSION ANALYSIS

```
#Page No.490
data <- data.frame(
  Heating_Cost = c(250, 360, 165, 43, 92, 200, 355, 290, 230, 120,
                   73, 205, 400, 320, 72, 272, 94, 190, 235, 139),
  Temperature = c(35, 29, 36, 60, 65, 30, 10, 7, 21, 55,
                  54, 48, 20, 39, 60, 20, 58, 40, 27, 30),
  Insulation = c(3, 4, 7, 6, 5, 5, 6, 10, 9, 2,
                 12, 5, 5, 4, 8, 5, 7, 8, 9, 7),
  Furnace_Age = c(6, 10, 3, 9, 6, 5, 7, 10, 11, 5,
                  4, 1, 15, 7, 6, 8, 3, 11, 8, 5)
)

model <- lm(Heating_Cost ~ Temperature + Insulation + Furnace_Age, data = data)

summary(model)

```

```
new_data <- data.frame(Temperature = 30, Insulation = 5, Furnace_Age = 10)
predicted_cost <- predict(model, new_data)

cat("The predicted heating cost is:", round(predicted_cost, 2), "dollars")

#The answer may slightly vary due to rounding off values.
```

**R code Exa 14.4** Multicollinearity

```
#Page No.510
library(car)

data <- data.frame(
  Heating_Cost = c(250, 360, 165, 43, 92, 200, 355, 290, 230, 120,
                   73, 205, 400, 320, 72, 272, 94, 190, 235, 139),
  Temperature = c(35, 29, 36, 60, 65, 30, 10, 7, 21, 55,
                  54, 48, 20, 39, 60, 20, 58, 40, 27, 30),
  Insulation = c(3, 4, 7, 6, 5, 5, 6, 10, 9, 2,
                 12, 5, 5, 4, 8, 5, 7, 8, 9, 7),
  Furnace_Age = c(6, 10, 3, 9, 6, 5, 7, 10, 11, 5,
                  4, 1, 15, 7, 6, 8, 3, 11, 8, 5)
)

cor_matrix <- cor(data[, -1])
print("Correlation Matrix:")
print(cor_matrix)

model <- lm(Heating_Cost ~ Temperature + Insulation
```

```
    + Furnace_Age, data = data)

vif_values <- vif(model)
print("Variance Inflation Factors (VIF):")
print(vif_values)

#The answer may slightly vary due to rounding off
    values.
```

**R code Exa 14.6** REGRESSION MODELS WITH INTERACTION

```
#Page No.515
library(car)

data <- data.frame(
  Heating_Cost = c(250, 360, 165, 43, 92, 200, 355,
     290, 230, 120,
                   73, 205, 400, 320, 72, 272, 94,
                      190, 235, 139),
  Temperature = c(35, 29, 36, 60, 65, 30, 10, 7, 21,
      55,
                  54, 48, 20, 39, 60, 20, 58, 40,
                     27, 30),
  Insulation = c(3, 4, 7, 6, 5, 5, 6, 10, 9, 2,
                 12, 5, 5, 4, 8, 5, 7, 8, 9, 7),
  Furnace_Age = c(6, 10, 3, 9, 6, 5, 7, 10, 11, 5,
                  4, 1, 15, 7, 6, 8, 3, 11, 8, 5)
)

data$Interaction <- data$Temperature * data$
    Insulation
model <- lm(Heating_Cost ~ Temperature + Insulation
    + Interaction, data = data)
summary(model)

```

```
#The answer may vary due to difference in
   representation.
```

**R code Exa 14.8** REVIEW OF MULTIPLE REGRESSION

```
#Page No. 521
library(ggplot2)
library(car)

data <- data.frame(
  Income = c(100.7, 99.0, 102.0, 100.7, 100.0, 95.2,
      101.0, 101.8, 102.0, 92.7,
             99.8, 101.0, 95.0, 97.5, 98.8, 101.5,
                100.7, 100.2, 104.3, 100.2,
             101.5, 101.0, 102.3, 100.2, 96.3),
  Value = c(190, 121, 161, 161, 179, 99, 114, 202,
     184, 90,
            181, 143, 132, 127, 153, 145, 174, 177,
               188, 153,
            150, 173, 163, 150, 139),
  Education = c(14, 15, 14, 14, 14, 14, 15, 14, 13,
     14,
                14, 15, 14, 14, 14, 14, 15, 15, 15,
                   15,
                16, 13, 14, 15, 14),
  Age = c(53, 49, 44, 39, 53, 46, 42, 49, 37, 43,
          48, 54, 44, 37, 50, 50, 52, 47, 49, 53,
          58, 42, 46, 50, 45),
  Mortgage = c(230, 370, 397, 181, 378, 304, 285,
     551, 370, 135,
               332, 217, 490, 220, 270, 279, 329,
                  274, 433, 333,
               148, 390, 142, 343, 373),
  Gender = c(1, 1, 1, 1, 0, 0, 1, 0, 0, 0,
             1, 1, 0, 0, 1, 1, 1, 0, 1, 1,
```

```r
                0, 1, 1, 0, 0)
)

cor_matrix <- cor(data)
print("Correlation Matrix:")
print(cor_matrix)

model <- lm(Income ~ Value + Education + Age + Mortgage + Gender, data = data)
summary(model)

vif_values <- vif(model)
print("Variance Inflation Factor (VIF):")
print(vif_values)

model_refined <- stepAIC(model, direction = "backward")
summary(model_refined)

par(mfrow=c(2,2))
plot(model_refined)

hist(residuals(model_refined), main="Histogram of Residuals", xlab="Residuals", col="lightblue", border="black")

plot(fitted(model_refined), residuals(model_refined), main="Residual Plot", xlab="Fitted Values", ylab="Residuals")
abline(h=0, col="red")

cat("Final Regression Equation:")
final_coefficients <- coef(model_refined)
print(final_coefficients)

#The answer may vary due to difference in representation.
```

# Chapter 15

# Nonparametric Methods

**R code Exa 15.1** TEST A HYPOTHESIS OF A POPULATION PROPORTION

```
#Page No.547
library(ggplot2)

n <- 2000
p_hat <- 1550 / n
pi_0 <- 0.80
alpha <- 0.05

z_score <- (p_hat - pi_0) / sqrt(pi_0 * (1 - pi_0) /
    n)
z_critical <- qnorm(alpha)
p_value <- pnorm(z_score)

cat("Sample Proportion (p-hat):", p_hat)
cat("Test Statistic (z-score):", z_score)
cat("Critical Value (z-critical):", z_critical)
cat("P-value:", p_value)

if (z_score < z_critical) {
  cat("Conclusion: Reject H0. The governor does NOT
```

```
      have enough support for re-election.")
} else {
  cat("Conclusion: Do NOT reject H0. The governor
      still has a chance of re-election.")
}

x <- seq(-4, 4, length = 1000)
y <- dnorm(x)

plot(x, y, type = "l", lwd = 2, col = "blue",
     main = "Hypothesis Test for Proportion",
     xlab = "Z-Score", ylab = "Density")

polygon(c(seq(-4, z_critical, length = 100), z_
   critical),
        c(dnorm(seq(-4, z_critical, length = 100)),
           0),
        col = "red", border = NA)

abline(v = z_critical, col = "red", lwd = 2, lty =
   2)
abline(v = z_score, col = "black", lwd = 2, lty = 2)

text(z_critical, 0.02, paste("Critical Value (z =",
   round(z_critical, 2), ")"), pos = 4, col = "red")
text(z_score, 0.05, paste("Observed z =", round(z_
   score, 2)), pos = 4, col = "black")

#The answer may slightly vary due to rounding off
   values.
```

**R code Exa 15.2** TWO SAMPLE TESTS ABOUT PROPORTIONS

```
#Page No.551
x1 <- 19
```

```
n1 <- 100
x2 <- 62
n2 <- 200

p1 <- x1 / n1
p2 <- x2 / n2

pc <- (x1 + x2) / (n1 + n2)
se <- sqrt(pc * (1 - pc) * (1/n1 + 1/n2))
z_value <- (p1 - p2) / se
p_value <- 2 * (1 - pnorm(abs(z_value)))

cat("Proportion of working women liking fragrance:", p1)
cat("Proportion of stay-at-home women liking fragrance:", p2)
cat("Pooled proportion:", pc)
cat("Z-score:", z_value)
cat("P-value:", p_value)

=alpha <- 0.05
if (p_value < alpha) {
  cat("Reject H0: There is a significant difference in proportions.")
} else {
  cat("Fail to reject H0: No significant difference in proportions.")
}

test_result <- prop.test(c(x1, x2), c(n1, n2), correct = FALSE)
print(test_result)

x <- seq(-4, 4, length = 100)
y <- dnorm(x)

plot(x, y, type = "l", lwd = 2, col = "blue",
     main = "Two-Proportion Z-Test: Rejection
```

```
     Regions",
     xlab = "Z-score", ylab = "Density")

abline(v = -1.96, col = "red", lwd = 2, lty = 2)
abline(v = 1.96, col = "red", lwd = 2, lty = 2)

abline(v = z_value, col = "green", lwd = 2)

polygon(c(x[x <= -1.96], -1.96), c(y[x <= -1.96], 0), col = rgb(1, 0, 0, 0.5))
polygon(c(x[x >= 1.96], 1.96), c(y[x >= 1.96], 0), col = rgb(1, 0, 0, 0.5))

legend("topright", legend = c("Critical Values ( 1.96)", "Computed Z-score", "Rejection Regions"),
       col = c("red", "green", "red"), lwd = 2, lty = c(2, 1, 1))
```

**R code Exa 15.3.1** Hypothesis Test of Equal Expected Frequencies

```
#Page No.555
observed <- c(32, 24, 35, 29)

expected <- rep(sum(observed) / length(observed), length(observed))
chi_square_test <- chisq.test(observed, p = rep(1/length(observed), length(observed)))
print(chi_square_test)

chi_square_value <- chi_square_test$statistic
p_value <- chi_square_test$p.value
df <- length(observed) - 1
critical_value <- qchisq(0.95, df)

x <- seq(0, 10, by=0.1)
```

```
y <- dchisq(x, df)

plot(x, y, type="l", lwd=2, col="blue",
     main="Chi-Square Distribution (df=3)",
     xlab=expression(chi^2), ylab="Density")

abline(v = critical_value, col="red", lwd=2, lty=2)
text(critical_value + 0.5, max(y)/2, "Critical Value
  ", col="red")

abline(v = chi_square_value, col="green", lwd=2, lty
  =2)
text(chi_square_value - 0.5, max(y)/3, "Chi-Square
  Value", col="green")

if (chi_square_value > critical_value) {
  cat("Reject the null hypothesis: Preferences are
     not equal.")
} else {
  cat("Fail to reject the null hypothesis:
     Preferences are equal.")
}
```

**R code Exa 15.3.2** Hypothesis Test of Unequal Expected Frequencies

```
#Page No.562
observed <- c(55, 50, 32, 13)
expected <- c(60, 45, 30, 15)

chi_square_test <- chisq.test(observed, p = expected
    / sum(expected), rescale.p = TRUE)

chi_square_value <- chi_square_test$statistic
p_value <- chi_square_test$p.value

```

```r
df <- length(observed) - 1

critical_value <- qchisq(0.95, df)
print(chi_square_test)

if (chi_square_value > critical_value) {
  cat("Reject the null hypothesis: Local and national admission rates are different.")
} else {
  cat("Fail to reject the null hypothesis: No significant difference in hospital admissions.")
}

x <- seq(0, 10, by=0.1)
y <- dchisq(x, df)

plot(x, y, type="l", lwd=2, col="blue",
     main="Chi-Square Distribution (df=3)",
     xlab=expression(chi^2), ylab="Density")

abline(v = critical_value, col="red", lwd=2, lty=2)
text(critical_value + 0.5, max(y)/2, "Critical Value", col="red")

abline(v = chi_square_value, col="green", lwd=2, lty=2)
text(chi_square_value - 0.5, max(y)/3, "Chi-Square Value", col="green")
```

**R code Exa 15.5** TESTING THE HYPOTHESIS THAT A DISTRIBUTION IS NORMAL

```r
#Page No.566
fo <- c(8, 11, 23, 38, 45, 32, 19, 4)
```

```
fe <- c(4.82, 12.29, 27.00, 40.86, 42.61, 31.00,
   14.96, 6.46)

chi_sq_result <- sum((fo - fe)^2 / fe)
df <- length(fo) - 2 - 1
critical_value <- qchisq(0.95, df)

cat("Computed Chi-Square Value:", chi_sq_result)
cat("Critical Value ( = 0.05):", critical_value)

if (chi_sq_result > critical_value) {
  cat("Reject H0: The population does NOT follow a
     normal distribution.")
} else {
  cat("Fail to Reject H0: The population follows a
     normal distribution.")
}
```

**R code Exa 15.6** CONTINGENCY TABLE ANALYSIS

```
#Page No.570
observed <- matrix(c(30, 17, 8,
                     140, 127, 58),
                   nrow = 2, byrow = TRUE)

rownames(observed) <- c("Salary", "Hourly")
colnames(observed) <- c("Satisfied", "Neutral", "
   Dissatisfied")

chi_sq_test <- chisq.test(observed)
print(chi_sq_test)

df <- (nrow(observed) - 1) * (ncol(observed) - 1)
critical_value <- qchisq(0.95, df)

```

```
cat("\nComputed Chi-Square Value:", chi_sq_test$statistic)
cat("Critical Chi-Square Value (  = 0.05):", critical_value)
cat("P-value:", chi_sq_test$p.value)

if (chi_sq_test$statistic > critical_value) {
  cat("Reject H0: Pay type and satisfaction level are related.")
} else {
  cat("Fail to Reject H0: No evidence that pay type and satisfaction level are related.")
}
```

# Chapter 16

# Nonparametric Methods

**R code Exa 16.1.1** THE SIGN TEST

```
#Page No.584
library(BSDA)

before <- c("Good", "Fair", "Excellent", "Poor", "Excellent", "Good", "Poor",
            "Excellent", "Good", "Poor", "Good", "Fair", "Good", "Good", "Poor")
after <- c("Outstanding", "Excellent", "Good", "Good", "Excellent", "Outstanding", "Fair",
           "Outstanding", "Poor", "Good", "Outstanding", "Excellent", "Fair", "Outstanding", "Good")
signs <- sign(match(after, c("Poor", "Fair", "Good", "Excellent", "Outstanding")) -
                match(before, c("Poor", "Fair", "Good", "Excellent", "Outstanding")))

signs <- signs[signs != 0]

n_success <- sum(signs > 0)
```

```r
n_total <- length(signs)

test_result <- binom.test(n_success, n_total, p = 0.5, alternative = "greater")
print(test_result)

#The answer may vary due to difference in representation.
```

**R code Exa 16.1.2** Normal Approximation to the Binomial

```r
#Page No.589
n <- 64
x <- 42
p0 <- 0.5

z <- ((x - 0.5) - (p0 * n)) / (0.5 * sqrt(n))
p_value <- 2 * (1 - pnorm(abs(z)))

cat("Z-score:", z)
cat("P-value:", p_value)

alpha <- 0.05
z_critical <- qnorm(1 - alpha/2)

if (abs(z) > z_critical) {
  cat("Reject H0: There is a significant difference in preference.")
} else {
  cat("Fail to reject H0: No significant preference detected.")
}
```

**R code Exa 16.2** TESTING A HYPOTHESIS ABOUT A MEDIAN

```r
#Page No.591
n <- 100
x <- 60
p0 <- 0.5

z <- ((x - 0.5) - (p0 * n)) / (0.5 * sqrt(n))
p_value <- 2 * (1 - pnorm(abs(z)))

cat("Z-score:", z)
cat("P-value:", p_value)

alpha <- 0.10
z_critical <- qnorm(1 - alpha/2)

if (abs(z) > z_critical) {
  cat("Reject H0: There is a significant difference
     in median spending.")
} else {
  cat("Fail to reject H0: No significant difference
     in median spending.")
}
```

**R code Exa 16.3** WILCOXON SIGNED RANK TEST FOR DEPENDENT POPULATIONS

```r
#Page No.593
spicy_ratings <- c(14, 8, 6, 18, 20, 16, 14, 6, 19,
   18, 16, 18, 4, 7, 16)
current_ratings <- c(12, 16, 2, 4, 12, 16, 5, 16,
   10, 10, 13, 2, 13, 14, 4)

differences <- spicy_ratings - current_ratings
differences <- differences[differences != 0]
```

```
abs_differences <- abs(differences)
ranks <- rank(abs_differences)

R_plus <- sum(ranks[differences > 0])
R_minus <- sum(ranks[differences < 0])

T_statistic <- min(R_plus, R_minus)

wilcoxon_test <- wilcox.test(spicy_ratings, current_ratings, paired=TRUE, alternative="greater")

print("Wilcoxon Signed-Rank Test")
print(paste("Test Statistic (T):", T_statistic))
print(paste("P-value:", round(wilcoxon_test$p.value, 4)))

alpha <- 0.05
if (T_statistic <= 25) {  # Critical value for n = 14,  = 0.05
  print("Reject H0: Customers prefer the spicy flavor.")
} else {
  print("Fail to reject H0: No strong evidence that customers prefer the spicy flavor.")
}
```

**R code Exa 16.4** WILCOXON RANK SUM TEST FOR INDEPENDENT POPULATIONS

```
#Page No.598
atlanta_bags <- c(11, 15, 10, 18, 11, 20, 24, 22, 25)
chicago_bags <- c(13, 14, 10, 8, 16, 9, 17, 21)

```

```r
all_bags <- c(atlanta_bags, chicago_bags)
ranks <- rank(all_bags)

ranks_atlanta <- ranks[1:length(atlanta_bags)]
ranks_chicago <- ranks[(length(atlanta_bags) + 1):length(all_bags)]

W_atlanta <- sum(ranks_atlanta)
W_chicago <- sum(ranks_chicago)

n1 <- length(atlanta_bags)
n2 <- length(chicago_bags)

z_value_atlanta <- (W_atlanta - (n1 * (n1 + n2 + 1)) / 2) / sqrt((n1 * n2 * (n1 + n2 + 1)) / 12)
z_value_chicago <- (W_chicago - (n2 * (n1 + n2 + 1)) / 2) / sqrt((n1 * n2 * (n1 + n2 + 1)) / 12)

p_value_atlanta <- 1 - pnorm(z_value_atlanta)
p_value_chicago <- pnorm(z_value_chicago)

wilcoxon_test <- wilcox.test(atlanta_bags, chicago_bags, alternative = "greater")

print("Wilcoxon Rank-Sum Test")
print(paste("Rank Sum for Atlanta (W):", W_atlanta))
print(paste("Rank Sum for Chicago (W):", W_chicago))
print(paste("Computed z-value (Atlanta as population 1):", round(z_value_atlanta, 2)))
print(paste("Computed z-value (Chicago as population 1):", round(z_value_chicago, 2)))
print(paste("Manual p-value (Atlanta > Chicago):", round(p_value_atlanta, 4)))
print(paste("Manual p-value (Chicago < Atlanta):", round(p_value_chicago, 4)))
print(paste("Wilcoxon test p-value:", round(wilcoxon_test$p.value, 4)))

```

```
alpha <- 0.05
if (z_value_atlanta > 1.645) {
  print("Reject H0: More gate-checked bags for Atlanta flights.")
} else {
  print("Fail to reject H0: No strong evidence that Atlanta has more gate-checked bags.")
}

if (z_value_chicago < -1.645) {
  print("Reject H0: Chicago has significantly fewer gate-checked bags.")
} else {
  print("Fail to reject H0: No strong evidence that Chicago has fewer gate-checked bags.")
}
```

**R code Exa 16.5** KRUSKAL WALLIS TEST

```
#Page No.602
chicago <- c(8, 9, 10, 7, 11, 6, 8, 12)
atlanta <- c(15, 14, 13, 16, 15, 14, 13, 17, 16)
wilcox.test(chicago, atlanta, alternative = "less")

st_lukes <- c(56, 39, 48, 38, 73, 60, 62)
swedish_medical <- c(103, 87, 51, 95, 68, 42, 107, 89)
piedmont <- c(42, 38, 89, 75, 35, 61)

waiting_times <- data.frame(
  time = c(st_lukes, swedish_medical, piedmont),
  hospital = rep(c("St. Luke's", "Swedish Medical", "Piedmont"),
                 times = c(length(st_lukes), length(swedish_medical), length(
```

```
piedmont)))
)

kruskal.test(time ~ hospital, data = waiting_times)
summary(aov(time ~ hospital, data = waiting_times))
```

**R code Exa 16.6** RANK ORDER CORRELATION

```
#Page No.608
library(ggplot2)

shopper_age <- c(28, 50, 44, 32, 55, 60, 38, 22, 21, 45, 52, 33, 19, 17, 21)
browsing_time <- c(342, 125, 121, 257, 56, 225, 185, 141, 342, 169, 218, 241, 583, 394, 249)

ggplot(data = data.frame(shopper_age, browsing_time), aes(x = shopper_age, y = browsing_time)) +
  geom_point(color = "blue", size = 3) +
  labs(title = "Scatter Plot of Age vs. Browsing Time", x = "Age", y = "Browsing Time (minutes)") +
  theme_minimal()

correlation_value <- cor(shopper_age, browsing_time, method = "spearman")
if (correlation_value < 0) {
  association_type <- "Strong Inverse (Negative) Relationship"
} else {
  association_type <- "No Clear Negative Relationship"
}
print(paste("The data suggests a", association_type))
```

```r

outliers <- boxplot.stats(browsing_time)$out
if (length(outliers) > 0) {
  print("There are potential outliers in browsing time:")
  print(outliers)
} else {
  print("No major outliers detected.")
}

correlation_test <- cor.test(shopper_age, browsing_time, method = "spearman")
print(correlation_test)

alpha <- 0.05
p_value <- correlation_test$p.value

if (p_value < alpha) {
  print("Reject the null hypothesis: There is significant evidence of a negative association.")
} else {
  print("Fail to reject the null hypothesis: No significant evidence of a negative association.")
}
```

# Chapter 17

# Index Numbers

**R code Exa 17.1.1** SIMPLE INDEX NUMBERS

```
#Page No.623
earnings_2000 <- 14.02
earnings_2016 <- 21.37

index_2016 <- (earnings_2016 / earnings_2000) * 100
percentage_increase <- index_2016 - 100

cat("Index of hourly earnings for 2016 (Base year:
   2000):", round(index_2016, 2))
cat("Percentage increase in hourly earnings:", round
   (percentage_increase, 2), "%")
```

**R code Exa 17.1.2** SIMPLE INDEX NUMBERS

```
#Page No.624
population_BC <- 4657947
population_Ontario <- 13730187

```

```r
population_index <- (population_BC / population_Ontario) * 100
percentage_difference <- 100 - population_index

cat("Population index of British Columbia compared to Ontario:", round(population_index, 1))
cat("Percentage difference:", round(percentage_difference, 1), "%")
```

**R code Exa 17.1.3** SIMPLE INDEX NUMBERS

```r
#Page No.624
airports <- c("Hartsfield-Jackson Atlanta", "Los Angeles", "Chicago O Hare",
              "Dallas/Fort Worth", "Denver", "John F. Kennedy",
              "San Francisco", "Miami", "Charlotte Douglas", "McCarran")
passengers <- c(96.2, 70.7, 70.0, 63.6, 53.5, 53.3, 47.1, 44.4, 44.3, 42.9)

base_passengers <- passengers[10]

index_values <- (passengers / base_passengers) * 100
percentage_difference <- index_values - 100

airport_data <- data.frame(Airport = airports,
                           Passengers = passengers,
                           Index = round(index_values, 1),
                           Difference_from_McCarran = round(percentage_difference, 1))
print(airport_data)

```

```
#The answer may vary due to difference in representation.
```

**R code Exa 17.3.1** WEIGHTED INDEXES

```
#Page No.630
items <- c("Bread", "Eggs", "Milk", "Apples", "Orange Juice", "Coffee")
price_2003 <- c(1.042, 1.175, 2.686, 0.911, 1.848, 2.999)
quantity_2003 <- c(50, 26, 102, 30, 40, 12)

price_2015 <- c(1.440, 2.133, 3.463, 1.265, 2.678, 4.827)
quantity_2015 <- c(55, 20, 130, 40, 41, 12)

expenditure_2003 <- price_2003 * quantity_2003
total_expenditure_2003 <- sum(expenditure_2003)

expenditure_2015_using_2003_quantity <- price_2015 * quantity_2003
total_expenditure_2015 <- sum(expenditure_2015_using_2003_quantity)

laspeyres_index <- (total_expenditure_2015 / total_expenditure_2003) * 100

cat("Total expenditure in 2003 (Base Year):", round(total_expenditure_2003, 2))
cat("Total expenditure in 2015 using 2003 quantities:", round(total_expenditure_2015, 2))
cat("Laspeyres Price Index for 2015:", round(laspeyres_index, 2))
cat("Price increase over the period:", round(laspeyres_index - 100, 2), "%")
```

**R code Exa 17.3.2** Paasche Price Index

```r
#Page No.631
items <- c("Bread", "Eggs", "Milk", "Apples", "Orange Juice", "Coffee")
price_2003 <- c(1.04, 1.18, 2.69, 0.91, 1.85, 3.00)
quantity_2003 <- c(50, 26, 102, 30, 40, 12)

price_2015 <- c(1.44, 2.13, 3.46, 1.27, 2.68, 4.83)
quantity_2015 <- c(55, 20, 130, 40, 41, 12)

expenditure_2015 <- price_2015 * quantity_2015
total_expenditure_2015 <- sum(expenditure_2015)

expenditure_2003_using_2015_quantity <- price_2003 * quantity_2015
total_expenditure_2003 <- sum(expenditure_2003_using_2015_quantity)
paasche_index <- (total_expenditure_2015 / total_expenditure_2003) * 100

cat("Total expenditure in 2003 using 2015 quantities:", round(total_expenditure_2003, 2))
cat("Total expenditure in 2015:", round(total_expenditure_2015, 2))
cat("Paasche Price Index for 2015:", round(paasche_index, 2))
cat("Price increase over the period:", round(paasche_index - 100, 2), "%")
```

**R code Exa 17.3.3** Fishers Ideal Index

```
#Page No.632
laspeyres_index <- 138.44
paasche_index <- 136.70

fishers_index <- sqrt(laspeyres_index * paasche_index)
cat("Fisher s Ideal Index:", round(fishers_index, 2))
```

**R code Exa 17.3.4** Value Index

```
#Page No.634
price_2000 <- c(1, 30, 10)
quantity_2000 <- c(1000, 100, 500)
price_2017 <- c(2, 40, 8)
quantity_2017 <- c(900, 120, 500)

total_sales_2000 <- sum(price_2000 * quantity_2000)
total_sales_2017 <- sum(price_2017 * quantity_2017)

value_index <- (total_sales_2017 / total_sales_2000) * 100
cat("Value Index for 2017 (Base Year 2000 = 100):", round(value_index, 1))
```

**R code Exa 17.4** SPECIAL PURPOSE INDEXES

```
#Page No.636
year_2005 <- c(20, 100, 50, 500)
year_2016 <- c(44, 125, 18, 700)

weights <- c(0.40, 0.30, 0.10, 0.20)
index_values <- (year_2016 / year_2005) * 100
```

```

general_business_activity_index <- sum(index_values
    * weights)
cat("General Business Activity Index for 2016 (Base
    Year 2005 = 100):", round(general_business_
    activity_index, 1))
```

**R code Exa 17.5.1** USING AN INDEX AS A DEFLATOR

```
#Page No.642
years <- c(1982, 1990, 1995, 2000, 2005, 2010, 2015)
sales <- c(875000, 1482000, 1491000, 1502000,
    1515000, 1596000, 1697000)
ppi <- c(100.0, 119.2, 127.9, 138.0, 155.7, 179.8,
    193.9)

constant_dollars <- (sales / ppi) * 100

result <- data.frame(Year = years, Sales = sales,
    PPI = ppi, Deflated_Sales = round(constant_
    dollars,2))
print(result)
```

**R code Exa 17.5.2** USING AN INDEX TO FIND PURCHASING POWER

```
#Page No.643
cpi <- 200.0

purchasing_power <- 1 / (cpi / 100)

cat("Purchasing Power of the Dollar:", round(
    purchasing_power, 2), "dollars")
```

**R code Exa 17.5.3** Shifting the Base

```r
#Page No.644
library(ggplot2)

years <- c(2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016)
djia <- c(10452.74, 10783.75, 10718.30, 12459.54, 13261.82, 8772.25, 10430.69, 11577.43, 12221.19, 13104.30, 16572.17, 17823.07, 17405.48)
nasdaq <- c(2011.08, 2184.75, 2216.53, 2429.72, 2653.91, 1578.87, 2294.41, 2676.65, 2657.39, 3091.33, 4160.03, 4760.24, 4897.65)

djia_index <- (djia / djia[1]) * 100
nasdaq_index <- (nasdaq / nasdaq[1]) * 100

data <- data.frame(Year = years, DJIA = djia, DJIA_Index = djia_index, NASDAQ = nasdaq, NASDAQ_Index = nasdaq_index)

djia_2016_value <- djia[length(djia)]
djia_base_value <- djia[1]
djia_2016_index <- djia_index[length(djia_index)]

cat(sprintf("Calculation of DJIA Index for 2016:"))
cat(sprintf("Index = (%.2f / %.2f) * 100 = %.2f", djia_2016_value, djia_base_value, djia_2016_index))
print(data)

ggplot(data, aes(x = Year)) +
  geom_line(aes(y = DJIA_Index, color = "DJIA"), size = 1.2) +
```

```
  geom_line(aes(y = NASDAQ_Index, color = "NASDAQ"), size = 1.2) +
  labs(title = "DJIA vs NASDAQ Index (Base Year: 2004 = 100)",
       x = "Year", y = "Index Value") +
  scale_color_manual(name = "Index", values = c("DJIA" = "blue", "NASDAQ" = "brown")) +
  theme_minimal()

#The answer may vary due to difference in representation.
```

# Chapter 18

# Time Series and Forecasting

**R code Exa 18.3** WEIGHTED MOVING AVERAGE

```r
#Page No.661
library(ggplot2)

year <- 1996:2015
attendance <- c(7445, 7405, 11450, 11224, 11703, 11890, 12380, 12181, 12557, 12700,
                19300, 22100, 22720, 21136, 22785, 23377, 23300, 23500, 23300, 24400)

moving_avg <- rep(NA, length(attendance))
for (i in 2:(length(attendance)-1)) {
  moving_avg[i] <- mean(attendance[(i-1):(i+1)])
}

weights <- c(0.2, 0.3, 0.5)
weighted_moving_avg <- rep(NA, length(attendance))
for (i in 2:(length(attendance)-1)) {
  weighted_moving_avg[i] <- sum(attendance[(i-1):(i+1)] * weights)
}
```

```r

df <- data.frame(Year = year, Attendance = attendance,
                 Moving_Avg = moving_avg, Weighted_Moving_Avg = weighted_moving_avg)

cat("Year  Attendance(000)  3-Year Moving Avg  3-Year Weighted Moving Avg")
for (i in 1:nrow(df)) {
  cat(df$Year[i], df$Attendance[i],
      ifelse(is.na(df$Moving_Avg[i]), "", sprintf("%10.2f", df$Moving_Avg[i])),
      ifelse(is.na(df$Weighted_Moving_Avg[i]), "", sprintf("%10.2f", df$Weighted_Moving_Avg[i])))
}

ggplot(df, aes(x = Year)) +
  geom_line(aes(y = Attendance, color = "Attendance (000's)"), size = 1) +
  geom_line(aes(y = Moving_Avg, color = "3-Year Moving Avg"), size = 1) +
  geom_line(aes(y = Weighted_Moving_Avg, color = "3-Year Weighted Moving Avg"), size = 1) +
  labs(title = "Attendance, 3-Year Moving Average, and Weighted Moving Average",
       x = "Year", y = "Attendance (000's)") +
  scale_color_manual(values = c("Attendance (000's)" = "green",
                                "3-Year Moving Avg" = "orange",
                                "3-Year Weighted Moving Avg" = "purple")) +
  theme_minimal()
```

**R code Exa 18.4** Least Squares Method

```
#Page No.665
library(ggplot2)

year <- 2012:2016
time_t <- 1:5
sales <- c(7.0, 10.0, 9.0, 11.0, 13.0)

model <- lm(sales ~ time_t)

a <- coef(model)[1]
b <- coef(model)[2]

cat(sprintf("Trend Equation:    = %.1f + %.1ft", a, b))
cat(sprintf("Interpretation: Sales are increasing at a rate of %.1f million dollars per year.", b))

t_2018 <- 7
sales_2018 <- a + b * t_2018
cat(sprintf("Forecasted Sales for 2018:    = %.1f + %.1f(%d) = %.1f million dollars", a, b, t_2018, sales_2018))

df <- data.frame(Year = year, Time_t = time_t, Sales = sales, Fitted_Sales = fitted(model))

cat("Year  Sales ($ million)  Coded Time (t)  Estimated Sales (   )")
for (i in 1:nrow(df)) {
  cat(df$Year[i], sprintf("%10.1f", df$Sales[i]), df$Time_t[i], sprintf("%10.1f", df$Fitted_Sales[i]), "\n")
```

```r
}

# Generate Plot
ggplot(df, aes(x = Year)) +
  geom_point(aes(y = Sales), color = "blue", size = 3) +
  geom_line(aes(y = Sales), color = "blue", linetype = "dashed", size = 1) +
  geom_line(aes(y = Fitted_Sales), color = "red", size = 1.2) +
  labs(title = "Sales and Trend Line for Jensen Foods (2012-2016)",
       x = "Year", y = "Sales ($ million)") +
  theme_minimal()
```

**R code Exa 18.6** SEASONAL VARIATION

```r
#Page No.671
library(zoo)

sales_data <- data.frame(
  Year = rep(2012:2017, each = 4),
  Quarter = rep(c("Winter", "Spring", "Summer", "Fall"), times = 6),
  Sales = c(6.7, 4.6, 10.0, 12.7, 6.5, 4.6, 9.8, 13.6,
            6.9, 5.0, 10.4, 14.1, 7.0, 5.5, 10.8, 15.0,
            7.1, 5.7, 11.1, 14.5, 8.0, 6.2, 11.4, 14.9)
)

sales_data$Moving_Total <- rollapply(sales_data$Sales, width = 4, FUN = sum, align = "center", fill = NA)
```

```
sales_data$Moving_Avg <- sales_data$Moving_Total / 4
sales_data$Centered_Moving_Avg <- rollapply(sales_data$Moving_Avg, width = 2, FUN = mean, align = "center", fill = NA)
sales_data$Specific_Seasonal_Index <- sales_data$Sales / sales_data$Centered_Moving_Avg
seasonal_indices <- aggregate(Specific_Seasonal_Index ~ Quarter, data = sales_data, FUN = mean, na.rm = TRUE)
correction_factor <- 4.00 / sum(seasonal_indices$Specific_Seasonal_Index)
seasonal_indices$Adjusted_Index <- seasonal_indices$Specific_Seasonal_Index * correction_factor
seasonal_indices$Final_Index <- round(seasonal_indices$Adjusted_Index * 100, 1)

print(seasonal_indices)
```

**R code Exa 18.7** Deseasonalized Data to Forecast

```
#Page No.678
sales_data <- data.frame(
  Year = rep(2012:2017, each = 4),
  Quarter = rep(c("Winter", "Spring", "Summer", "Fall"), times = 6),
  Sales = c(6.7, 4.6, 10.0, 12.7, 6.5, 4.6, 9.8, 13.6,
            6.9, 5.0, 10.4, 14.1, 7.0, 5.5, 10.8, 15.0,
            7.1, 5.7, 11.1, 14.5, 8.0, 6.2, 11.4, 14.9),
  Seasonal_Index = c(0.765, 0.575, 1.141, 1.519, 0.765, 0.575, 1.141, 1.519,
                     0.765, 0.575, 1.141, 1.519, 0.765, 0.575, 1.141, 1.519,
```

```
                     0.765, 0.575, 1.141, 1.519, 0.765, 0.575, 1.141, 1.519)
)

sales_data$t <- 1:nrow(sales_data)
sales_data$Deseasonalized_Sales <- sales_data$Sales / sales_data$Seasonal_Index
model <- lm(Deseasonalized_Sales ~ t, data = sales_data)
summary(model)

a <- coef(model)[1]
b <- coef(model)[2]

future_t <- 25:28
quarters_2018 <- c("Winter", "Spring", "Summer", "Fall")
seasonal_indices <- c(0.765, 0.575, 1.141, 1.519)

deseasonalized_forecast <- a + b * future_t
final_forecast <- deseasonalized_forecast * seasonal_indices

forecast_2018 <- data.frame(
  Quarter = quarters_2018,
  Time_Period = future_t,
  Deseasonalized_Forecast = round(deseasonalized_forecast, 5),
  Seasonal_Index = seasonal_indices,
  Final_Quarterly_Forecast = round(final_forecast, 5)
)
print(forecast_2018)
```

**R code Exa 18.8** DURBIN WATSON STATISTIC

```r
#Page No.682
library(lmtest)

data <- data.frame(
  Advertising = c(5.5, 5.5, 5.3, 5.5, 5.4, 5.3, 5.5, 5.7, 5.9, 6.2,
                  6.3, 5.9, 6.1, 6.2, 6.2, 6.5, 6.7, 6.9, 6.5, 6.4),
  Sales = c(153, 156, 153, 147, 159, 160, 147, 147, 152, 160,
            169, 176, 176, 179, 184, 181, 192, 205, 215, 209)
)

model <- lm(Sales ~ Advertising, data = data)
summary(model)

intercept <- coef(model)[1]
slope <- coef(model)[2]

increase_in_advertising <- 1
increase_in_sales <- slope * increase_in_advertising
increase_in_sales

dw_test <- dwtest(model)
print(dw_test)

#The answer may vary due to difference in representation.
```

# Chapter 19

# Statistical Process Control and Quality Management

**R code Exa 19.3** SOURCES OF VARIATION

```
#Page No.702
library(ggplot2)
library(dplyr)

water_usage <- data.frame(
  Activity = c("Laundering", "Watering lawn", "Personal bathing", "Cooking",
               "Swimming pool", "Dishwashing", "Car washing", "Drinking"),
  Gallons = c(24.9, 143.7, 106.7, 5.1, 28.3, 12.3, 10.4, 7.9)
)

total_usage <- sum(water_usage$Gallons)

water_usage <- water_usage %>%
  arrange(desc(Gallons)) %>%
  mutate(Percent = (Gallons / total_usage) * 100,
         Cumulative = cumsum(Percent))
```

```r

print(water_usage)

ggplot(water_usage, aes(x = reorder(Activity, -
    Gallons), y = Gallons)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_line(aes(y = Cumulative * max(water_usage$
     Gallons) / 100, group = 1),
              color = "red", linewidth = 1.2) +
  geom_point(aes(y = Cumulative * max(water_usage$
     Gallons) / 100),
               color = "red", size = 3) +
  scale_y_continuous(
    name = "Gallons Used Per Day",
    sec.axis = sec_axis(~ . * 100 / max(water_usage$
       Gallons), name = "Cumulative Percentage")
  ) +
  labs(title = "Pareto Chart for Water Usage",
       x = "Activity",
       y = "Gallons Used Per Day") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust
      = 1))
```

**R code Exa 19.4.1** PURPOSE AND TYPES OF QUALITY CONTROL CHARTS

```r
#Page No.706
library(ggplot2)
library(dplyr)

call_data <- data.frame(
  Sample = 1:16,
  Time1 = c(8, 7, 11, 12, 11, 7, 10, 8, 8, 12, 7, 9,
      10, 8, 10, 9),
```

```
  Time2 = c(9, 10, 12, 8, 10, 7, 7, 11, 11, 9, 7, 9, 12, 11, 13, 11),
  Time3 = c(15, 7, 10, 6, 6, 10, 4, 11, 8, 12, 9, 4, 12, 9, 9, 8),
  Time4 = c(4, 6, 9, 9, 14, 4, 10, 7, 14, 17, 17, 4, 12, 6, 4, 5),
  Time5 = c(11, 8, 10, 12, 11, 11, 10, 7, 12, 11, 13, 11, 12, 8, 9, 11)
)

call_data <- call_data %>%
  rowwise() %>%
  mutate(
    Mean = mean(c(Time1, Time2, Time3, Time4, Time5)),
    Range = max(c(Time1, Time2, Time3, Time4, Time5)) - min(c(Time1, Time2, Time3, Time4, Time5))
  )

x_bar <- mean(call_data$Mean)
R_bar <- mean(call_data$Range)

A2 <- 0.577

UCL <- x_bar + A2 * R_bar
LCL <- x_bar - A2 * R_bar

cat("Overall Mean:", x_bar)
cat("Average Range:", R_bar)
cat("Upper Control Limit (UCL):", UCL)
cat("Lower Control Limit (LCL):", LCL)

ggplot(call_data, aes(x = Sample, y = Mean)) +
  geom_line(color = "blue") +
  geom_point(color = "blue", size = 3) +
  geom_hline(yintercept = x_bar, linetype = "dashed", color = "black", size = 1) +
  geom_hline(yintercept = UCL, linetype = "dashed",
```

```
      color = "red", size = 1) +
  geom_hline(yintercept = LCL, linetype = "dashed",
      color = "red", size = 1) +
  labs(
    title = "Control Chart for Mean Call Duration",
    x = "Sample Number (Hour)",
    y = "Mean Call Duration (Minutes)"
  ) +
  theme_minimal()
```

**R code Exa 19.4.2** Control Charts for Variables

```
#Page No.706
library(ggplot2)
library(dplyr)

call_data <- data.frame(
  Sample = 1:16,
  Time1 = c(8, 7, 11, 12, 11, 7, 10, 8, 8, 12, 7, 9,
       10, 8, 10, 9),
  Time2 = c(9, 10, 12, 8, 10, 7, 7, 11, 11, 9, 7, 9,
       12, 11, 13, 11),
  Time3 = c(15, 7, 10, 6, 6, 10, 4, 11, 8, 12, 9, 4,
       12, 9, 9, 8),
  Time4 = c(4, 6, 9, 9, 14, 4, 10, 7, 14, 17, 17, 4,
       12, 6, 4, 5),
  Time5 = c(11, 8, 10, 12, 11, 11, 10, 7, 12, 11,
      13, 11, 12, 8, 9, 11)
)

call_data <- call_data %>%
  rowwise() %>%
  mutate(
    Range = max(c(Time1, Time2, Time3, Time4, Time5)
        ) - min(c(Time1, Time2, Time3, Time4, Time5))
```

```
  )

R_bar <- mean(call_data$Range)

D3 <- 0
D4 <- 2.115

UCL_R <- D4 * R_bar
LCL_R <- D3 * R_bar

cat("Average Range (R ):", R_bar)
cat("Upper Control Limit (UCL_R):", UCL_R)
cat("Lower Control Limit (LCL_R):", LCL_R)

ggplot(call_data, aes(x = Sample, y = Range)) +
  geom_line(color = "blue") +
  geom_point(color = "blue", size = 3) +
  geom_hline(yintercept = R_bar, linetype = "dashed", color = "black", size = 1) +
  geom_hline(yintercept = UCL_R, linetype = "dashed", color = "red", size = 1) +
  geom_hline(yintercept = LCL_R, linetype = "dashed", color = "red", size = 1) +
  labs(
    title = "Control Chart for Range (Call Duration)",
    x = "Sample Number (Hour)",
    y = "Range of Call Duration (Minutes)"
  ) +
  theme_minimal()
```

**R code Exa 19.6.1** p Charts

```
#Page No.714
library(ggplot2)
```

```r
library(dplyr)

quality_data <- data.frame(
  Date = c(rep("10-Oct", 4), rep("11-Oct", 4), rep("12-Oct", 4), rep("13-Oct", 4),
           rep("14-Oct", 4), rep("17-Oct", 4), rep("18-Oct", 4), rep("19-Oct", 4),
           rep("20-Oct", 4), rep("21-Oct", 4)),
  Sample_Size = rep(50, 40),
  Defects = c(1, 0, 9, 9, 4, 4, 5, 3, 9, 3, 10, 2, 2, 4, 9, 4,
              6, 9, 2, 4, 7, 9, 0, 8, 6, 9, 6, 1, 4, 5, 2, 5,
              0, 0, 4, 7, 5, 1, 9, 9)
)

quality_data <- quality_data %>%
  mutate(Proportion_Defective = Defects / Sample_Size)

total_defects <- sum(quality_data$Defects)
total_samples <- sum(quality_data$Sample_Size)
p_bar <- total_defects / total_samples

n <- 50
sigma_p <- sqrt(p_bar * (1 - p_bar) / n)
UCL_p <- p_bar + 3 * sigma_p
LCL_p <- max(0, p_bar - 3 * sigma_p)

cat("Overall Proportion Defective (p ):", p_bar)
cat("Upper Control Limit (UCL_p):", UCL_p)
cat("Lower Control Limit (LCL_p):", LCL_p)

ggplot(quality_data, aes(x = as.factor(Date), y = Proportion_Defective, group = 1)) +
  geom_point(color = "blue", size = 3) +
  geom_line(color = "blue") +
  geom_hline(yintercept = p_bar, linetype = "dashed"
```

```
  , color = "black", size = 1) +
  geom_hline(yintercept = UCL_p, linetype = "dashed", color = "red", size = 1) +
  geom_hline(yintercept = LCL_p, linetype = "dashed", color = "red", size = 1) +
  labs(
    title = "p-Chart for Defective Mirrors",
    x = "Date",
    y = "Proportion Defective"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

**R code Exa 19.6.2** cBar Charts

```
#Page No.717
library(ggplot2)

misspelled_words <- c(5, 6, 3, 0, 4, 5, 1, 2, 7, 4)
c_bar <- mean(misspelled_words)

sigma_c <- sqrt(c_bar)
UCL_c <- c_bar + 3 * sigma_c
LCL_c <- max(0, c_bar - 3 * sigma_c)

cat("Mean number of misspelled words ( c  ):", c_bar)
cat("Upper Control Limit (UCL_c):", UCL_c)
cat("Lower Control Limit (LCL_c):", LCL_c)

data <- data.frame(
  Day = 1:10,
  Misspelled_Words = misspelled_words
)

```

```r
ggplot(data, aes(x = Day, y = Misspelled_Words)) +
  geom_point(color = "blue", size = 3) +
  geom_line(color = "blue") +
  geom_hline(yintercept = c_bar, linetype = "dashed", color = "black", size = 1) +
  geom_hline(yintercept = UCL_c, linetype = "dashed", color = "red", size = 1) +
  geom_hline(yintercept = LCL_c, linetype = "dashed", color = "red", size = 1) +
  labs(
    title = "c-Chart for Misspelled Words per Edition",
    x = "Day",
    y = "Number of Misspelled Words"
  ) +
  theme_minimal() +
  scale_x_continuous(breaks = 1:10) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

**R code Exa 19.7** ACCEPTANCE SAMPLING

```r
#Page No.720
library(ggplot2)

n <- 20
c <- 2

pi_values <- seq(0, 0.3, by = 0.01)

P_acceptance <- sapply(pi_values, function(pi) {
  sum(dbinom(0:c, size = n, prob = pi))
})

data <- data.frame(Defect_Probability = pi_values,
```

```
    Acceptance_Probability = P_acceptance)

ggplot(data, aes(x = Defect_Probability, y =
    Acceptance_Probability)) +
  geom_line(color = "blue", size = 1.2) +
  geom_point(color = "red") +
  labs(
    title = "OC Curve for Acceptance Sampling Plan",
    x = "Proportion of Defective Items (  )",
    y = "Probability of Accepting the Lot"
  ) +
  theme_minimal()
```