# R Textbook Companion for Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber, and Jian Pei[1]

Created by
Shankru Guggari
Ph.D.
Computer Science and Engineering
BMS College of Engineering, Bengaluru
Cross-Checked by
R TBC Team

May 20, 2020

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

# Book Description

**Title:** Data Mining: Concepts and Techniques

**Author:** Jiawei Han, Micheline Kamber, and Jian Pei

**Publisher:** Morgan Kaufmann, USA

**Edition:** 3

**Year:** 2011

**ISBN:** 9780123814791

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

**Exa** Example (Solved example)

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

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

# Contents

# List of R Codes

# Chapter 2

# Getting to know your Data

**R code Exa 2.6.1** Mean

```
1 Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)
2
3 print("Mean")
4
5 print(paste("$",mean(Data)))
```

---

**R code Exa 2.7** Median

```
1 Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)
2 print("Median")
3 print(paste("$",median(Data)))
```

---

**R code Exa 2.8** Mode

```
1 Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)
2
```

```
mode <- function(x) {
   uni_value <- unique(x)
    uni_value[which.max(tabulate(match(x, uni_value)
       ))]
}

print("Mode")
print(paste("$",mode(Data)))
```

**R code Exa 2.9** Midrange

```
Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)


print("Mid range")


Mid_Range<- ((min(Data)+max(Data))/2)

pr_mir<- Mid_Range

print(paste("$",pr_mir))
```

**R code Exa 2.10** Interquartile Range

```
Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)


print("Interquartile Range")


print(IQR(Data))
```

**R code Exa 2.11** Boxplot

```
Data <- data.frame(MG= c(30,36,47,50,52), CY=c
   (25,60,30,21,70))


boxplot(Data,xlab = "Number of Cylinders",ylab = "
   Miles Per Gallon", main = "Summary of Mileage")
```

**R code Exa 2.12** Variance and standard deviation

```
Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)

print("variance")
print(var(Data))


print("Standard Deviation")

######"The answer provided in the textbook is
   19.47"
print(sd(Data))
```

**R code Exa 2.13** Quantile Plot

```
Unit_price = c(40,43,47,74,75,78,115,117,120)
count_of_items_sold =c
   (275,300,250,360,515,540,320,270,350)

qqplot(count_of_items_sold,Unit_price)
```

**R code Exa 2.15** Histogram

```
Unit_price = c(40,43,47,74,75,78,115,117,120)
count_of_items_sold =c
   (275,300,250,360,515,540,320,270,350)

hist(Unit_price,breaks = seq(0, 800, by = 10))
```

**R code Exa 2.18** Dissimilarity between binary attributes

```
Ja <- c(1,1,0,1,0,0,0)

Jim <- c(1,1,1,0,0,0,0)

ma <- c(0,1,1,1,0,1,0)



Ja_Jim = (1+1)/(1+1+1)
print("Distance between Jack and Jim")

print(Ja_Jim)


Ja_ma =(0+1)/(2+0+1)
print("Distance between Jack and mary")

print(Ja_ma)

Jim_ma =(1+2)/(1+1+2)
print("Distance between Jim and mary")
print(Jim_ma)
```

**R code Exa 2.19** Euclidean distance and manhattan distance

```
1
2 x1 <- c(1,2)
3
4 x2 <- c(3,5)
5
6 dif <- x2-x1
7
8 Euclidean<- sqrt(sum(dif^2))
9 print("Euclidean distance")
10 print(Euclidean)
11
12
13 print("Manhattan Distance")
14 Manhattan <- sum(dif)
15
16 print(Manhattan)
```

**R code Exa 2.20** Supremum distance

```
1
2 x1 <- c(1,2)
3
4 x2 <- c(3,5)
5
6 Supremum_Dis<- max(x2)-max(x1)
7
8 print("supremum Distance")
9
10 print(Supremum_Dis)
```

**R code Exa 2.21** Dissimilarity between ordinal attributes

```
1
2  test2 = c(3,1,2,3)
3
4  obj_Id <- c(1,2,3,4)
5
6
7  dif <- obj_Id - test2
8
9  Euclidean<- sqrt(sum(dif^2))
10 print("Euclidean distance")
11 print(Euclidean)
12
13
14
15
16 test2 = c(1,3)
17
18 obj_Id <- c(2,4)
19
20
21 dif <- obj_Id - test2
22
23 Euclidean<- sqrt(sum(dif^2))
24 print("Euclidean distance object 2 and 4")
25 print(Euclidean)
26
27
28
29
30
31
32
```

```
test2 = c(3,3)

obj_Id <- c(1,4)


dif <- obj_Id - test2

Euclidean<- sqrt(sum(dif^2))
print("Euclidean distance of object 1 and 4")
print(Euclidean)
```

**R code Exa 2.23** cosine similarity between two term frequency vectors

```
x <- c(5,0,3,0,2,0,0,2,0,0)
y <- c(3,0,2,0,1,1,0,1,0,1)



x_square<-sqrt
   (5^2+0^2+3^2+0^2+2^2+0^2+0^2+2^2+0^2+0^2)


y_square<-sqrt
   (3^2+0^2+2^2+0^2+1^2+1^2+0^2+1^2+0^2+1^2)



Consine <- ((sum(x*y))/(x_square*y_square))


##### Text Book answer is 0.94
print(Consine)
```

# Chapter 3

# Data Preprocessing

**R code Exa 3.1** Correln analysis of nominal attributes using chi2

```
1 Obs_fre <- c(250,50,200,1000)
2
3 Exp_fre <-c(90,210,360,840)
4
5 chi = sum((Obs_fre - Exp_fre)^2/(Exp_fre))
6
7
8 print(chi)
```

---

**R code Exa 3.2** Covariance analysis of numeric attribute

```
1 AllEle <- c(6,5,4,3,2)
2
3 Hightech <-c(20,10,14,5,5)
4
5
6 E_AllEle <- sum(AllEle)/length(AllEle)
7
```

```
8
9  All<- paste("$",E_AllEle)
10
11 print(All)
12
13
14
15 E_Hightech <- sum(Hightech)/length(Hightech)
16
17
18 hi <-paste("$",E_Hightech)
19
20 print(hi)
21
22
23 print("Covariance")
24
25
26 cov<- (sum(AllEle*Hightech)/length(AllEle))- (4*E_
      Hightech)
27
28 print(cov)
```

**R code Exa 3.3** Histograms

```
1  AllEle <- c
      (1,1,5,5,5,5,5,8,8,10,10,10,10,12,14,14,14,15,15,15,15,15,15,18,1
2
3  hist(AllEle,main="Histogram for price", xlab="Price"
      , ylab= "Count")
```

**R code Exa 3.4** Min Max normalization

```
Min <- 12000
Max <- 98000
Tra <- 73600


Min_max_nor <- (Tra-Min)/(Max-Min)

print("Min-Max Normalization")
print(Min_max_nor)
```

**R code Exa 3.5** Z score normalization

```
Mean <- 54000
std <- 16000
Tra <- 73600


Z_score_nor <- (Tra-Mean)/(std)

print("Z-score Normalization")
print(Z_score_nor)
```

**R code Exa 3.6** Decimal Scalling

```
decscale<- function (x)
{
    vect <- apply(abs(x), 2, max)
    zvect <- ceiling(log10(vect))
    sc_fact <- 10^zvect
    scale(x, center = TRUE, scale = sc_fact)
}


```

```
10 print(decscale(iris[,1:4]))
```

# Chapter 4

# Data warehousing and online analytical processing

**R code Exa 4.3** Fact Constellation

```
1
2
3 # Setup the dimension tables
4
5
6 Citytab <- data.frame(key=c("MY", "Ben", "TU", "HU",
      "GU"),
7                        name=c("MYSORE", "
                             Bengaluru", "Tumkur", "
                             Hubballi", "Gulabarga")
                             ,
8                        country=c("India", "India"
                             , "India", "India", "
                             India"))
9
10
11 weektab <- data.frame(key=1:7,
12                       desc=c("Mon", "Tue", "Wen"
                             , "Thu", "Fri", "Sat",
```

```
                                    "Sun"))
13
14
15 prodtab <- data.frame(key=c("Dal", "Sugar", "Rice"),
       price=c(50, 70, 40))
16
17
18
19 # Function to generate the Sales table
20
21
22 Totalsales <- function(Record_Size) {
23
24
25     location <- sample(Citytab$key, Record_Size,
         replace=T, prob=c(2,2,1,1,1))
26
27     week<- sample(weektab$key, Record_Size, replace=
         T)
28
29     year <- sample(c(2017,2018), Record_Size,
         replace=T)
30
31     product <- sample(prodtab$key, Record_Size,
         replace=T, prob=c(1, 5, 7))
32
33     sales <- data.frame(week=week, year=year,
         location=location, prod=product)
34 }
35
36
37  # create  fact table of sales
38 Table_fact_sales <- Totalsales(100)
39
40 print(Table_fact_sales)
```

**R code Exa 4.4** OLAP operations

```
1
2
3 # Setup the dimension tables
4
5
6 Citytab <- data.frame(key=c("MY", "Ben", "TU", "HU",
      "GU"),
7                        name=c("MYSORE", "
                           Bengaluru", "Tumkur", "
                           Hubballi", "Gulabarga")
                           ,
8                        country=c("India", "India"
                           , "India", "India", "
                           India"))
9
10
11 weektab <- data.frame(key=1:7,
12                        desc=c("Mon", "Tue", "Wen"
                           , "Thu", "Fri", "Sat",
                           "Sun"))
13
14
15 prodtab <- data.frame(key=c("Dal", "Sugar", "Rice"),
       price=c(50, 70, 40))
16
17
18
19 # Function to generate the Sales table
20
21
22 Totalsales <- function(Record_Size) {
23
```

```
24
25      location <- sample(Citytab$key, Record_Size,
            replace=T, prob=c(2,2,1,1,1))
26
27      week<- sample(weektab$key, Record_Size, replace=
            T)
28
29      year <- sample(c(2017,2018), Record_Size,
            replace=T)
30
31      product <- sample(prodtab$key, Record_Size,
            replace=T, prob=c(1, 3, 2))
32
33      sales <- data.frame(week=week, year=year,
            location=location, product=product)
34  }
35
36
37   # create  fact table of sales
38  Table_fact_sales <- Totalsales(20)
39
40  print(Table_fact_sales)
41
42
43
44
45  Income <- tapply(Table_fact_sales$year, Table_fact_
        sales[,c("product", "week", "year")], FUN=
        function(x){return(sum(x))})
46
47
48  print("Showing the cells of income")
49
50  print(Income)
51
52
53  print("Slice")
54
```

```
55 slice<- Income["Dal", "1",]
56
57 print(slice)
58
59
60 print("Roll up")
61
62 print(apply(Income, c("week", "year"), FUN=function(
      x) {return(sum(x, na.rm=TRUE))}))
63
64
65 print("Drill down")
66 print(apply(Income, c("week", "year", "product"),
      FUN=function(x) {return(sum(x, na.rm=TRUE))}))
67
68
69
70 print("piot")
71 print(apply(Income, c("week", "year"), FUN=function(
      x) {return(sum(x, na.rm=TRUE))}))
72
73 print(apply(Income, c("week", "product"), FUN=
      function(x) {return(sum(x, na.rm=TRUE))}))
74
75
76
77 print("Dice")
78 print(Income[,c("1","2"),c("2017","2018")])
```

**R code Exa 4.6** A data cube is a lattice of cuboids

```
1 library(sqldf)
2
3 # dimension tables
4
```

```
5
6  Citytab <- data.frame(key=c("MY", "Ben", "TU", "HU",
       "GU"),
7                        name=c("MYSORE", "
                            Bengaluru", "Tumkur", "
                            Hubballi", "Gulabarga")
                            ,
8                        country=c("India", "India"
                            , "India", "India", "
                            India"))
9
10
11 weektab <- data.frame(key=1:7,
12                       desc=c("Mon", "Tue", "Wen"
                            , "Thu", "Fri", "Sat",
                            "Sun"))
13
14
15 prodtab <- data.frame(key=c("Dal", "Sugar", "Rice"),
       price=c(50, 70, 40))
16
17
18
19 # Function to generate the Total Sales
20
21
22 Totalsales <- function(Record_Size) {
23
24
25     location <- sample(Citytab$key, Record_Size,
          replace=T, prob=c(2,2,1,1,1))
26
27     week<- sample(weektab$key, Record_Size, replace=
          T)
28
29     year <- sample(c(2017,2018), Record_Size,
          replace=T)
30
```

```
31      product <- sample(prodtab$key, Record_Size,
           replace=T, prob=c(1, 3, 2))
32
33      sales <- data.frame(week=week, year=year,
           location=location, product=product)
34  }
35
36
37   # create  fact table of sales
38  Table_fact_sales <- Totalsales(20)
39
40  ###print(Table_fact_sales)
41
42  print("Selecting Mysore location")
43
44  sel<- sqldf("select * from Table_fact_sales where
       location = 'MY'")
45
46  print(sel)
```

**R code Exa 4.8** Join Index

```
1  library(sqldf)
2
3  DataFrame <- data.frame(Seq = rep(10:20, each = 5),
       tra = rep(1:11,5))
4
5  SelQue <- sqldf("select Seq, tra from DataFrame
6  natural join (select Seq, avg(tra) as avg_tra from
       DataFrame group by Seq)
7  where tra> avg_tra")
8
9
10 print(SelQue)
```

**R code Exa 4.9** OLAP query processing

```
1  library(sqldf)
2
3  # dimension tables
4
5
6  Citytab <- data.frame(key=c("MY", "Ben", "TU", "HU",
       "GU"),
7                        name=c("MYSORE", "
                            Bengaluru", "Tumkur", "
                            Hubballi", "Gulabarga")
                            ,
8                        country=c("India", "India"
                            , "India", "India", "
                            India"))
9
10
11 weektab <- data.frame(key=1:7,
12                       desc=c("Mon", "Tue", "Wen"
                            , "Thu", "Fri", "Sat",
                            "Sun"))
13
14
15 prodtab <- data.frame(key=c("Dal", "Sugar", "Rice"),
       price=c(50, 70, 40))
16
17
18
19 # Function to generate the Total Sales
20
21
22 Totalsales <- function(Record_Size) {
23
```

```
24
25      location <- sample(Citytab$key, Record_Size,
            replace=T, prob=c(2,2,1,1,1))
26
27      week<- sample(weektab$key, Record_Size, replace=
            T)
28
29      year <- sample(c(2017,2018), Record_Size,
            replace=T)
30
31      product <- sample(prodtab$key, Record_Size,
            replace=T, prob=c(1, 3, 2))
32
33      sales <- data.frame(week=week, year=year,
            location=location, product=product)
34  }
35
36
37   # create  fact table of sales
38  Table_fact_sales <- Totalsales(20)
39
40  ###print(Table_fact_sales)
41
42  print("Selecting items group by Product")
43
44  sel<- sqldf("select * from Table_fact_sales group by
        product")
45
46  print(sel)
```

# Chapter 5

# Data Cube Technology

**R code Exa 5.9** Construct the inverted index

```
1  A <- c("a1","a1","a1","a2","a2")
2
3  B <- c("b1","b2","b2","b1","b1")
4
5  C <- c("c1","c1","c1","c1","c1")
6
7
8  D <- c("d1","d2","d1","d1","d1")
9
10
11 E <- c("e1","e1","e2","e2","e3")
12
13 #####TID List##########
14
15 print("TID List of a1")
16 print(which("a1" == A))
17
18 print("TID List of a2")
19 print(which("a2" == A))
20
21 print("TID List of b1")
```

```
22 print(which("b1" == B))
23
24
25 print("TID List of b2")
26 print(which("b2" == B))
27
28
29
30
31 print("TID List of c1")
32 print(which("c1" == C))
33
34 print("TID List of d1")
35 print(which("d1" == D))
36
37
38
39 print("TID List of e1")
40 print(which("e1" == E))
41
42
43 print("TID List e2")
44 print(which("e2" == E))
45
46
47 print("TID List of e3")
48 print(which("e3" == E))
49
50 ##### List Size################
51
52 a1 <-length(grep("a1", A))
53
54 print("List size of a1")
55 print(a1)
56
57 a2 <-length(grep("a2", A))
58
59 print("List size of a2")
```

```
60
61 print(a2)
62
63 b1 <-length(grep("b1", B))
64
65
66 print("List size of b1")
67 print(b1)
68
69 b2 <-length(grep("b2", B))
70
71
72 print("List size of b2")
73 print(b2)
74
75
76 c1 <-length(grep("c1", C))
77
78
79 print("List size of c1")
80 print(c1)
81
82
83 d1 <-length(grep("d1", D))
84
85
86 print("List size of d1")
87 print(d1)
88
89
90
91
92 d2 <-length(grep("d2", D))
93
94
95 print("List size of d2")
96 print(d2)
97
```

```
98
99  e1 <-length(grep("e1", E))
100
101
102 print("List size of e1")
103 print(e1)
104
105
106
107 e2 <-length(grep("e2", E))
108
109
110 print("List size of e2")
111 print(e2)
112
113
114
115 e3 <-length(grep("e3", E))
116
117
118 print("List size of e3")
119 print(e3)
```

**R code Exa 5.10** Compute shell fragment

```
1  A <- c("a1","a1","a1","a2","a2")
2
3  B <- c("b1","b2","b2","b1","b1")
4
5  C <- c("c1","c1","c1","c1","c1")
6
7
8  D <- c("d1","d2","d1","d1","d1")
9
10
```

```
11 E <- c("e1","e1","e2","e2","e3")
12
13 #####TID List##########
14
15 print("Cuboid of AB")
16
17 print("TID List of a1 and b1")
18 M1 <- which("a1" == A)
19 M2<- which("b1" == B)
20
21 z1 <- Reduce(intersect, list(M1,M2))
22
23 print(length(z1))
24
25
26 print("List size:")
27 print(length(z1))
28
29
30 print("TID List of a1 and b2")
31
32 M3 <- which("a1" == A)
33 M4<- which("b2" == B)
34
35 z2 <- Reduce(intersect, list(M3,M4))
36 print(z2)
37
38 print("List size:")
39 print(length(z2))
40
41
42 print("TID List of a2 and b1")
43
44 M5 <- which("a2" == A)
45 M6<- which("b1" == B)
46
47 z3 <- Reduce(intersect, list(M5,M6))
48
```

```
49 print(z3)
50
51 print("List size:")
52 print(length(z3))
53
54 print("TID List of a2 and b2")
55
56 M7 <- which("a2" == A)
57 M8<- which("b2" == B)
58
59 z4 <- Reduce(intersect, list(M7,M8))
60
61 print(z4)
62
63 print("List size:")
64 print(length(z4))
65
66
67
68 ############################
69
70
71
72 print("Cuboid of DE")
73
74 print("TID List of d1 and e1")
75 M11 <- which("d1" == D)
76 M12<- which("e1" == E)
77
78 z11 <- Reduce(intersect, list(M11,M12))
79
80 print(length(z11))
81
82
83 print("List size:")
84 print(length(z11))
85
86
```

```
print("TID List of d1 and e2")
M13 <- which("d1" == D)
M14<- which("e2" == E)

z12 <- Reduce(intersect, list(M13,M14))

print(length(z12))


print("List size:")
print(length(z12))



print("TID List of d1 and e3")
M15 <- which("d1" == D)
M16<- which("e3" == E)

z13 <- Reduce(intersect, list(M15,M16))

print(length(z13))


print("List size:")
print(length(z13))


print("TID List of d2 and e1")
M17 <- which("d2" == D)
M18<- which("e1" == E)

z14 <- Reduce(intersect, list(M17,M18))

print(length(z14))


print("List size:")
print(length(z14))
```

**R code Exa 5.11** Computing cubes with average measure

```
1
2 print(cbind(TID=c(1,2,3,4,5),Item_count=c(5,3,8,5,2)
    ,SUM=c(70,10,20,40,30)))
```

# Chapter 6

# Mining frequent patterns associations and correlations basic concepts and methods

**R code Exa 6.9** Correlation analysis using chi2

```
1 library(arules)
2 Video <- c(4000,2000,3500,500)
3 Video_game <-c(4500,1500,3000,1000)
4
5 cor <- sum((Video-Video_game)^2/Video_game)
6
7
8 print(cor)
```

# Chapter 7

# Advanced Pattern mining

**R code Exa 7.1** Mining multilevel association rules

```
## Taken from arules package PDF
library(arules)

data("Groceries")

## Groceries contains a hierarchy stored in itemInfo

Groceries_level2 <- aggregate(Groceries, by = "level2")

inspect(Groceries_level2)
```

**R code Exa 7.2** redundancy among multilevel association rules

```
library(arules)

#### it demonstartes redundant rules and constraints (Example 7.8)
```

```
4
5  data("Income")
6
7
8
9  Ass_rules <- apriori(Income, parameter = list(
       support = 0.5,conf=0.9))
10
11
12 inspect(rules[is.redundant(Ass_rules)])
```

**R code Exa 7.3** Rare patterns and negative patterns

```
1  library(arules)
2
3
4  ### No rare Items in the dataset so I am showing
       other measure like significance of rules
5
6
7  data("Income")
8
9
10
11 Ass_rules <- apriori(Income, parameter = list(
       support = 0.5,conf=0.9))
12
13
14 interestMeasure(Ass_rules,measure = "hyperConfidence
       ",transactions = Income)
```

**R code Exa 7.6** Negatively related patterns

```
library(arules)


### Complement Items from the dataset



data("Adult")
rules <- apriori(Adult)

InMe<- interestMeasure(rules, measure = "kulczynski"
   ,transactions = Adult)


print(InMe)
```

**R code Exa 7.12** closed and maximal itemsets

```
library(arules)

data("Adult")

## find only frequent itemsets which do not contain
   small or large income

items <- apriori(Adult, parameter = list(support=
   0.001, conf=0.001, target="frequent"))

close <-is.closed(items)


print(close)
```

**R code Exa 7.13** pattern distance

```
library(arules)

data("Adult")

sample <- sample(Adult, 10)

## used Jaccard distance

jaccard_dist <- dissimilarity(sample)

hira_clu <- hclust(jaccard_dist , method = "ward.D2")

plot(hira_clu, labels = FALSE, main = "Dendrogram")
```

**R code Exa 7.15** Semantic annotations of a frequent patterns

```
library(arules)

data("Adult")

sample <- sample(Adult, 100)


sample1 <- apriori(Adult[1:50], parameter = list(
   support = 0.6))
sample2 <- apriori(Adult[51:100], parameter = list(
   support = 0.6))


combine_samples <- c(sample1, sample2)


```

```
15 print(duplicated(combine_samples))
```

# Chapter 8

# Classification basic concepts

**R code Exa 8.9** Sensitivity and Specificity

```
1 TP <- 90
2
3 FN <-210
4
5 FP <- 140
6
7 TN <- 9560
```

```
8
9  Sensitivity= TP/(TP+FN)
10
11 print("Sensitivity")
12
13 print(Sensitivity)
14
15 Specificity = TN/(FP+TN)
16
17 print("Specificity")
18
19 print(Specificity)
```

**R code Exa 8.10** Precision and recall

```
1  TP <- 90
2
3  FN <-210
4
5  FP <- 140
6
7  TN <- 9560
8
9
10 Precision = TP/(TP+FP)
11
12 print("Precision")
13
14 print(Precision)
15
16 Recall = TP/(TP+FN)
17
18 print("Recall")
19
20 print(Recall)
```

**R code Exa 8.11** ROC Curve

```
1  library(ROCR)
2
3  dataset<-data.frame(Pre=c
       (0.90,0.80,0.70,0.60,0.55,0.54,0.53,0.51,0.50,0.40)
       ,cls=c(1,1,0,1,1,0,0,0,1,0))
4
5
6  predictions <- prediction(dataset$Pre, dataset$cls)
7
8  model_perf <- performance(predictions,"tpr","fpr")
9
10 plot(model_perf)
```

# Chapter 9

# Classification Advanced methods

**R code Exa 9.1** Backpropagation algorithm

```
### install.packages("neuralnet")


library(neuralnet)
library(MASS)
data <- Boston


## used learning rate 0.9 and one hidden layer




print(net.infert <- neuralnet(medv~nox+rm+age,
   learningrate = 0.9,data,hidden=1,act.fct="tanh"))


prediction(net.infert)
```

**R code Exa 9.3** Error correcting codes

```
1
2
3
4  library(e1071)
5
6
7  C1 <- c(1,1,1,1,1,1,1)
8  C2 <- c(0,0,0,0,1,1,1)
9  C3 <- c(0,0,1,1,0,0,1)
10 C4 <- c(0,1,0,1,0,1,0)
11
12 out <-c(0,0,0,1,0,1,0)
13
14 Out1<-hamming.distance(C1, out)
15
16 print(Out1)
17
18
19 Out2<-hamming.distance(C2, out)
20
21 print(Out2)
22
23 Out3<-hamming.distance(C3, out)
24
25 print(Out3)
26
27
28 Out4<-hamming.distance(C4, out)
29
30 print(Out4)
```

# Chapter 10

# Cluster analysis basic concepts and mehtods

**R code Exa 10.1** Clustering by K means partitioning

```
1
2 data <- iris
3 data$Species <- NULL
4
5 clusters <- kmeans(data, 3)
6
7 plot(data[c("Sepal.Length", "Sepal.Width","Petal.
    Length", "Petal.Width")], col=clusters$cluster)
```

---

**R code Exa 10.2** Drawback of k means

```
1
2 data <- c(1,2,3,8,9,10,25)
3
4 clu1 <- c(1,2,3)
5
```

```
6  clu2 <- c(8,9,10,25)
7
8
9  Mean_clu1 <- mean(clu1)
10
11 Mean_clu2 <- mean(clu2)
12
13
14 su_mean1 <- sum((clu1-Mean_clu1)^2)
15
16 su_mean2 <- sum((clu2-Mean_clu2)^2)
17
18
19 First_Total<-su_mean1+su_mean2
20
21 print("Variation within first partition")
22
23 print(First_Total)
24
25
26
27
28 print("
   ##########################################################
   ")
29
30
31
32 data <- c(1,2,3,8,9,10,25)
33
34 clu3 <- c(1,2,3,8)
35
36 clu4 <- c(9,10,25)
37
38
39 Mean_clu3 <- mean(clu3)
40
41 Mean_clu4 <- mean(clu4)
```

```
42
43
44 su_mean3 <- sum((clu3-Mean_clu3)^2)
45
46 su_mean4 <- sum((clu4-Mean_clu4)^2)
47
48
49 sec_Total<-su_mean3+su_mean4
50
51 print("Variation within second partition")
52
53 print(sec_Total)
```

**R code Exa 10.3** Agglomerative versus divisive hierachical clustering

```
1
2  library(cluster)
3
4 data(iris)
5
6 print("Agglomerative Clustering")
7 agn_hiclu <- agnes(iris, metric = "manhattan", stand
      = TRUE)
8 print(agn_hiclu)
9 plot(agn_hiclu)
10
11
12
13 print("
    ############################################################
    ")
14
15
16
17 data(iris)
```

```
18 print("Devisive Clustering")
19 divisive_clu <- diana(iris, metric = "manhattan",
      stand = TRUE)
20 print(divisive_clu)
21 plot(divisive_clu)
```

**R code Exa 10.4** Single versus complete linkages

```
1 index <- sample(1:dim(iris)[1], 60)
2 newiris <- iris[index,]
3 newiris$Species <- NULL
4
5 ###Apply Hierarchical Clustering
6
7 hier_clu <- hclust(dist(newiris), method="ave")
8 plot(hier_clu , hang = -1, labels=newiris$Species[
     index])
```

**R code Exa 10.7** Density reachability and density connectivity

```
1
2 library(dbscan)
3
4 data(iris)
5
6 iris <- as.matrix(iris[,1:4])
7
8 result_dbscan <- dbscan(iris, eps = .3, minPts = 3)
9
10
11 print(result_dbscan)
12
13
```

```
14 pairs(iris, col = result_dbscan$cluster + 1L)
```

**R code Exa 10.8** core distance and reachability distance

```
1
2 library(dbscan)
3
4 data(iris)
5
6
7
8 result <- optics(iris[,1:4], eps = 10, minPts = 5)
9
10
11 ###Componets of reachability
12 Com_reach <- as.reachability(result)
13
14
15 ###plot(Com_reach, order_labels = TRUE)
16
17
18 dend <- as.dendrogram(Com_reach)
19
20
21 plot(dend)
```

# Chapter 11

# Advanced cluster analysis

**R code Exa 11.5** Probabilistic clusters

```
library(FPDclustering)


Pro_clu <- PDclust(iris[,1:4], k = 2)


print(Pro_clu)
```

**R code Exa 11.7** Fuzzy clustering using the EM algorithm

```
###Install package install.packages("EMCluster")


library(EMCluster)
library(MASS)
library(Matrix)

```

```
8
9
10 Dataset<- data.frame(Fi=c(4,3,9,14,18,21),se=c
      (10,3,6,8,11,7))
11
12
13 d <- as.matrix(Dataset)
14
15 emobj <- simple.init(d, nclass = 2)
16 emobj <- shortemcluster(d, emobj)
17
18 emclu <- emcluster(d, emobj, assign.class = TRUE)
19 print(emclu)
```

**R code Exa 11.14** clustering in a derived space

```
1 ##install.packages("speccalt")
2
3
4 library(speccalt)
5 iris <- local.rbfdot(iris[,1:4])
6 cluster1 <- speccalt(iris) # with automatic
      estimation
7 cluster2 <- speccalt(iris, 4)
8
9
10 View(cluster1)
11
12 View(cluster2)
```

**R code Exa 11.16** Bipartite graph

```
1 ###install.packages("igraph")
```

```
2
3 library(igraph)
4
5 graph <- make_full_bipartite_graph(2, 2, dir=TRUE,
     mode="all")
6
7
8 print(graph, v=TRUE)
9
10
11 plot(graph)
```

**R code Exa 11.19** Measurements based on geodesic distance

```
1
2
3 library(geosphere)
4 #geodesic distance
5 geo_dest<- geodesic(cbind(0,0), 2, 3)
6
7 print(geo_dest)
```

**R code Exa 11.21** cuts and clusters

```
1
2 hc <- hclust(dist(iris[,1:4]))
3
4
5 cutree(hc, k = 1:3) #k = 1 is trivial
6 cutree(hc, h = 100)
7
8 ## Compare the 2 and 10 grouping:
9 gra210 <- cutree(hc, k = c(2,10))
```

```
10
11
12 plot(gra210)
```

**R code Exa 11.23** Hard and soft constraints

```
1
2 library(SoftClustering)
3
4 data(iris)
5
6 Hardclu <- HardKMeans(iris[,1:4],2,2,10)
7
8
9 print(Hardclu)
```

# Chapter 12

# Outlier detection

**R code Exa 12.1** Outliers

```
1 library(outliers)
2
3
4
5 outliers<- outlier(iris[,1:4])
6
7 print(outliers)
```

**R code Exa 12.7** Detecting outliers using clustering

```
1 iris <- iris[,1:4]
2
3
4 kmeansClu <- kmeans(iris, centers=3)
5
6
7
8 centersofclu <- kmeansClu$centers[kmeansClu$cluster,
    ]
```

```
9
10 dist <- sqrt(rowSums((iris - centersofclu)^2))
11
12 outliers <- order(dist, decreasing=T)[1:10]
13
14 print(outliers)
```

**R code Exa 12.8** Univariate outliers detection using maxumum likelihood

```
1 dataset <- c
     (24,28.9,28.9,29,29.1,29.1,29.2,29.2,29.3,29.4)
2
3 mean(dataset)
4 print(sd(dataset))
```

**R code Exa 12.9** Multivariate outlier detection using mahalanobis distance

```
1
2 library(mvoutlier)
3
4 data(iris)
5
6
7 aq.plot(iris[,1:4], alpha=0.1)
```

**R code Exa 12.10** Multivariate outlier detection using the chi2 statistic

```
1
2 library(outliers)
3
```

```
4 dataset <- c
     (24,28.9,28.9,29,29.1,29.1,29.2,29.2,29.3,29.4)
5 chisq.out.test(dataset )
6 print(chisq.out.test(dataset, opposite=TRUE))
```

**R code Exa 12.12** Multivariate outlier detection using multiple clusters

```
1 library(kmodR)
2
3 d<- as.matrix(iris[,1:4])
4
5 print(kmod(d,k=3,l=10, i_max = 100))
```

**R code Exa 12.13** Outlier detection using a histogram

```
1
2
3 d<- iris[,1:4]
4
5 hist(as.matrix(d))
```

**R code Exa 12.14** local proximity based outliers

```
1 library(DMwR)
2
3
4
5 d<- iris[,1:4]
6
7 local_pro<- lofactor(d[,-5],10)
```

```
8
9
10 print(local_pro)
```

**R code Exa 12.15** Detecting outliers as objects that do not belong to any clusters

```
1  library(DMwR)
2  library(lattice)
3  library(grid)
4
5
6
7  d11<- iris[,1:4]
8
9  d <- as.matrix(d11)
10
11 density_scan<- dbscan(d, eps=1, minPts = 5)
12
13
14 print(density_scan)
```

**R code Exa 12.16** clustering based outliers detection using distance to the closest cluster

```
1  library(DMwR)
2  library(lattice)
3  library(grid)
4
5
6
7  dataset<- iris[,1:4]
8
```

```
d <- as.matrix(dataset)

dist<- kNNdist(d, k=4, search="kd")

print(dist)


kNNdistplot(d, k=4)
```

**R code Exa 12.18** detecting outliers in small clusters

```
library(DMwR)
 # Ignore class column "Species", which is a
    categorical column

iris <- iris[,1:4]

outliersIris <- lofactor(iris, k=2)

outliers <- order(outliersIris, decreasing=T)[1:10]

print(outliers)
```

**R code Exa 12.20** Outlier detection by semi supervised learning

```
library(ldbod)

dataset <- as.matrix(iris[,1:4])

local_den<- ldbod(dataset, k=3, nsub=50)


print(local_den)
```

**R code Exa 12.24** Outliers in subspace

```
library(HighDimOut)
library(ggplot2)

result_SOD <- Func.SOD(data = iris[,1:4], k.nn = 10,
    k.sel = 5, alpha = 0.8)

plot(result_SOD)
```

**R code Exa 12.25** angle based outliers

```


###install.packages("abodOutlier")


library(abodOutlier)

data(iris)

Abodf<- abod(iris[,1:4], method = "randomized", n_
   sample_size = 5)


View(Abodf)
```