DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

NAAC GRADE A+ ACCREDITED UNIVERSITY

CHANDIGARH UNIVERSITY

UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Submitted By: Adarsh Gupta **Submitted To: Er. Navneet Kaur**

Subject Name	Data Mining Lab
Subject Code	20CSP-376
Branch	CSE
Semester	6

CU
CHANDIGARH UNIVERSITY

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF ENGINEERING

Department of Computer Science & Engineering

Subject Name: Data Mining

Subject Code: 20CSP-376

Submitted to:

Faculty name: Er. Navneet Kaur

Submitted by:

Name: Adarsh Gupta

UID: 20BCS4884

Section: 607

Group: B

Ex. No	List of Experiments	Date	Conduct (MM: 12)	Viva (MM: 10)	Record (MM: 8)	Total (MM: 30)	Remarks/Signature
1.1	Demonstration of preprocessing on .arff file using student data .arff.	15-02-2023					
1.2	To perform statistical analysis of data	22-02-2023					
1.3	Demonstration of associate rule mining using apriori algorithm on supermarket data.	22-02-2023					
1.4							
2.1							
2.3							
2.4							
3.1							
3.2							
3.3							

Experiment 1.1

Student Name: Adarsh Gupta **UID:** 20BCS4884

Branch: CSE **Section/Group:** 607-B

Semester: 6TH **Date of Performance:** 15/02/2023

Subject Name: Data Mining Lab **Subject Code:** 20CSP-376

1. Aim/Overview of the practical:

Demonstration of preprocessing on .arff file using student data .arff.

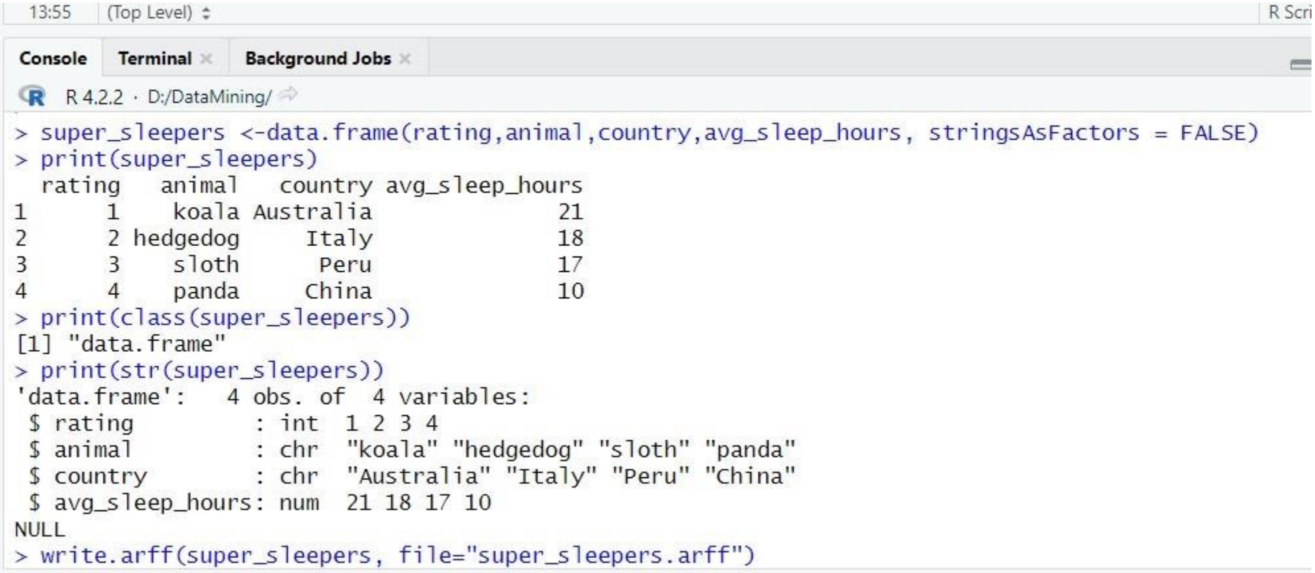
2. Objective:

1. To learn how to create an .arff file.
2. To understand the process of file creation in R.
3. To learn the utilization of RWeka.

3. Code:

```
library(RWeka)
setwd("D:\\DataMining")
getwd()
rating <- 1:4
animal <- c('koala','hedgedog','sloth','panda')
country <-c('Australia','Italy','Peru','China')
avg_sleep_hours <- c(21,18,17,10)

super_sleepers <-data.frame(rating,animal,country,avg_sleep_hours,
stringsAsFactors = FALSE)
print(super_sleepers)
print(class(super_sleepers))
print(str(super_sleepers))
write.arff(super_sleepers, file="super_sleepers.arff")
```

4. Output:



```
> super_sleepers <-data.frame(rating,animal,country,avg_sleep_hours, stringsAsFactors = FALSE)
> print(super_sleepers)
  rating   animal   country avg_sleep_hours
1      1    koala Australia              21
2      2 hedgedog     Italy              18
3      3    sloth      Peru              17
4      4    panda     China              10
> print(class(super_sleepers))
[1] "data.frame"
> print(str(super_sleepers))
'data.frame':	4 obs. of  4 variables:
 $ rating         : int  1 2 3 4
 $ animal         : chr  "koala" "hedgedog" "sloth" "panda"
 $ country        : chr  "Australia" "Italy" "Peru" "China"
 $ avg_sleep_hours: num  21 18 17 10
NULL
> write.arff(super_sleepers, file="super_sleepers.arff")
```

Experiment 1.2

Student Name: Adarsh Gupta	**UID:** 20BCS4884
Branch: CSE	**Section/Group:** 607-B
Semester: 6TH	**Date of Performance:** 22/02/2023
Subject Name: Data Mining Lab	**Subject Code:** 20CSP-376

1. Aim/Overview of the practical:

To perform statistical analysis of data

2. Objective:

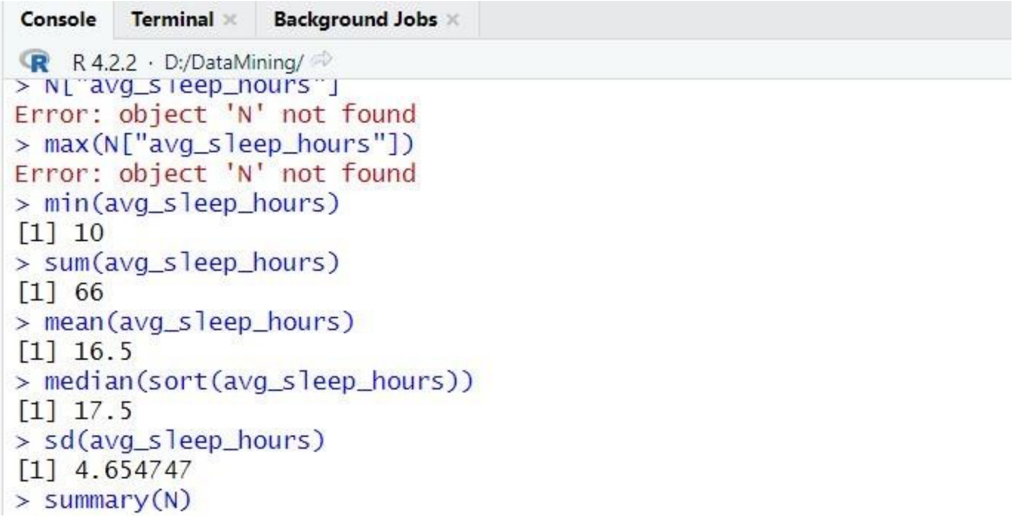
- To perform statistical analysis of data using RWeka.
- To learn how to calculate mean, median and standard deviation in R.
- To learn the utilization of RWeka.

3. Code:

```
library("RWeka")

N = read.arff("D:\\DataMining")

print(N)

cat("\n\n\n")

print(head(N,2))
```

```
dim(N)

names(N)
N["animal"]

N["avg_sleep_hours"]

max(N["avg_sleep_hours"])

min(avg_sleep_hours)

sum(avg_sleep_hours)

mean(avg_sleep_hours)

median(sort(avg_sleep_hours))

sd(avg_sleep_hours)

summary(N)
```

4. **Output:**



```
> N["avg_sleep_hours"]
Error: object 'N' not found
> max(N["avg_sleep_hours"])
Error: object 'N' not found
> min(avg_sleep_hours)
[1] 10
> sum(avg_sleep_hours)
[1] 66
> mean(avg_sleep_hours)
[1] 16.5
> median(sort(avg_sleep_hours))
[1] 17.5
> sd(avg_sleep_hours)
[1] 4.654747
> summary(N)
```

Experiment 1.3

Student Name: Adarsh Gupta **UID:** 20BCS4884

Branch: CSE **Section/Group:** 607-B

Semester: 6TH **Date of Performance:** 22/02/2023

Subject Name: Data Mining Lab **Subject Code:** 20CSP-376

1. Aim/Overview of the practical:

Demonstration of associate rule mining using Apriori algorithm on supermarket data

2. Objective:

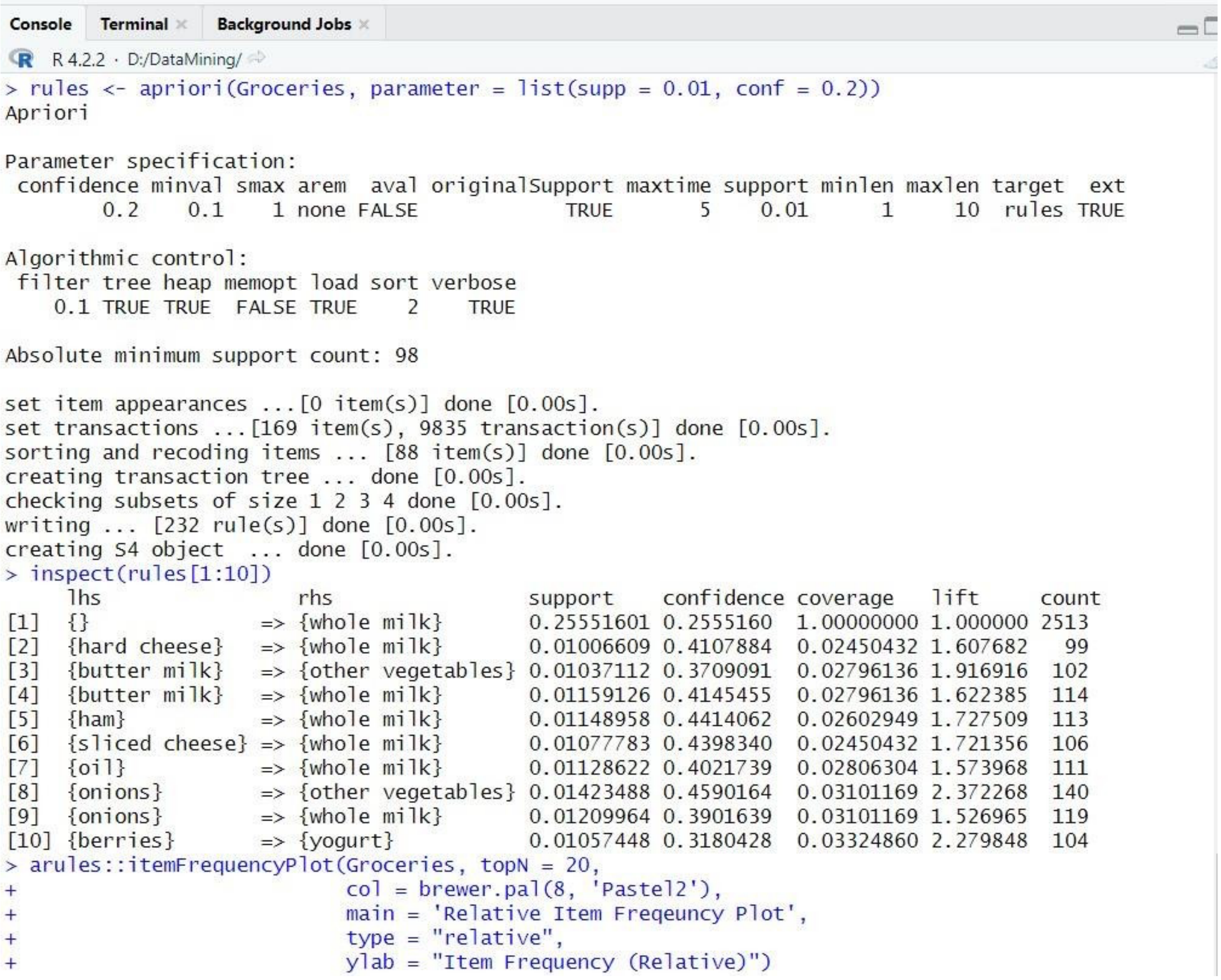
a. To learn how to import.
b. To learn how to perform associate rule mining using Apriori algorithm.
c. To learn the utilization of arules, arulesviz, RcolorBrewer.

3. Code:

```
library(arules)
library(arulesViz)
library(RColorBrewer)

data("Groceries")

rules <- apriori(Groceries, parameter = list(supp = 0.01, conf = 0.2))
```

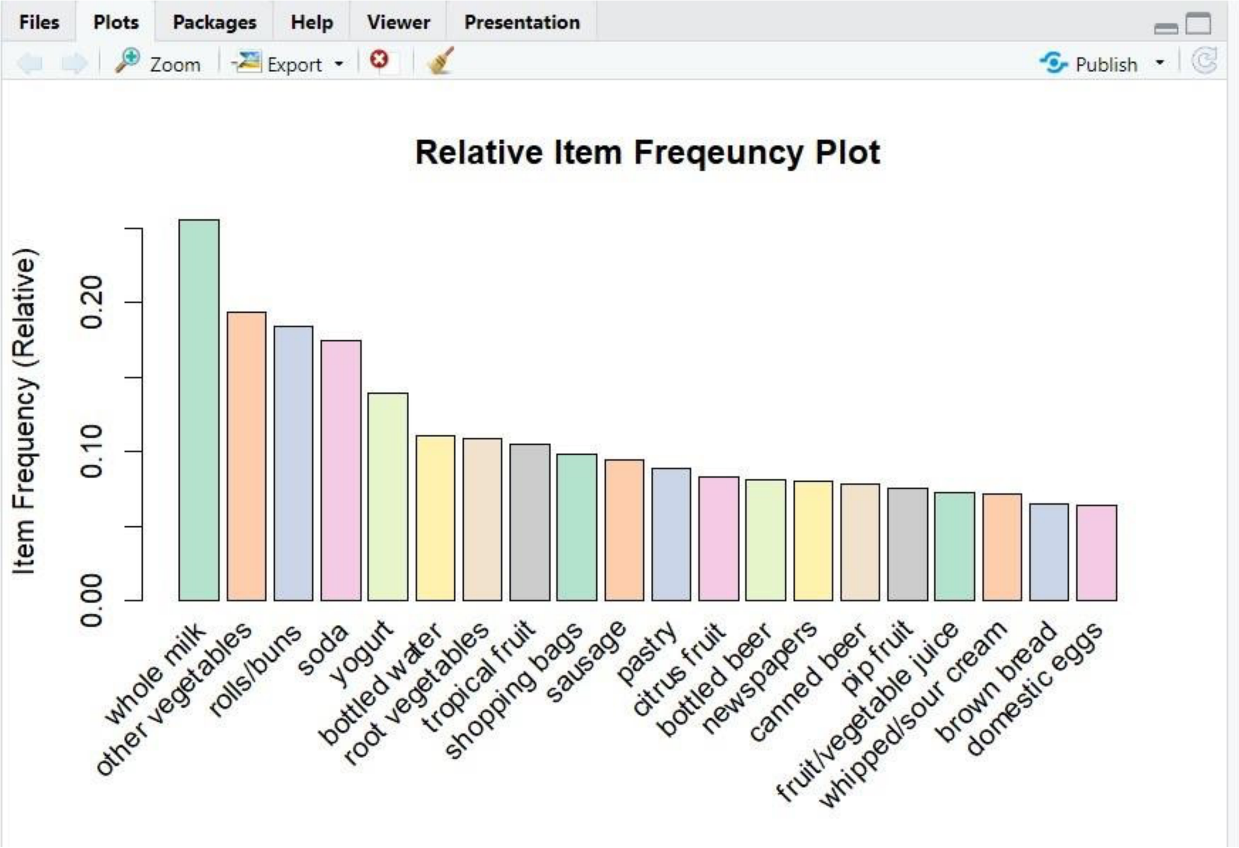
```
inspect(rules[1:10])

arules::itemFrequencyPlot(Groceries, topN = 20,
                          col = brewer.pal(8, 'Pastel2'),
                          main = 'Relative Item Freqeuncy Plot',
                          type = "relative",
                          ylab = "Item Frequency (Relative)")
```

4. Output:



```
> rules <- apriori(Groceries, parameter = list(supp = 0.01, conf = 0.2))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.2    0.1    1 none FALSE            TRUE       5    0.01      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 98

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
sorting and recoding items ... [88 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [232 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> inspect(rules[1:10])
     lhs                  rhs                support    confidence coverage   lift     count
[1]  {}                => {whole milk}       0.25551601 0.2555160  1.00000000 1.000000 2513
[2]  {hard cheese}     => {whole milk}       0.01006609 0.4107884  0.02450432 1.607682   99
[3]  {butter milk}     => {other vegetables} 0.01037112 0.3709091  0.02796136 1.916916  102
[4]  {butter milk}     => {whole milk}       0.01159126 0.4145455  0.02796136 1.622385  114
[5]  {ham}             => {whole milk}       0.01148958 0.4414062  0.02602949 1.727509  113
[6]  {sliced cheese}   => {whole milk}       0.01077783 0.4398340  0.02450432 1.721356  106
[7]  {oil}             => {whole milk}       0.01128622 0.4021739  0.02806304 1.573968  111
[8]  {onions}          => {other vegetables} 0.01423488 0.4590164  0.03101169 2.372268  140
[9]  {onions}          => {whole milk}       0.01209964 0.3901639  0.03101169 1.526965  119
[10] {berries}         => {yogurt}           0.01057448 0.3180428  0.03324860 2.279848  104
> arules::itemFrequencyPlot(Groceries, topN = 20,
+                           col = brewer.pal(8, 'Pastel2'),
+                           main = 'Relative Item Freqeuncy Plot',
+                           type = "relative",
+                           ylab = "Item Frequency (Relative)")
```

Graph output:


Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
Relative Item Freqeuncy Plot
Item Frequency (Relative)
whole milk
other vegetables
rolls/buns
soda
yogurt
bottled water
root vegetables
tropical fruit
shopping bags
sausage
pastry
citrus fruit
bottled beer
newspapers
canned beer
pip fruit
fruit/vegetable juice
whipped/sour cream
brown bread
domestic eggs