Assignment 3

#Libraries used:

```
library(arules)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(arulesViz)
```

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'seriation':
##   method         from
##   reorder.hclust gclus
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:arules':
##
##     intersect, recode, setdiff, setequal, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

#Reading data:

```
data <- read.csv("groceries_v2.csv")
head(data)
```

```
##       citrus.fruit semi.finished.bread      margarine
ready.soups
## 1   tropical fruit              yogurt         coffee
## 2       whole milk
## 3        pip fruit              yogurt  cream cheese              meat
spreads
```

```
## 4 other vegetables          whole milk condensed milk long life bakery
product
## 5       whole milk              butter         yogurt
rice
## 6 abrasive cleaner
```

#Using read.transactions() function converting into transaction format:

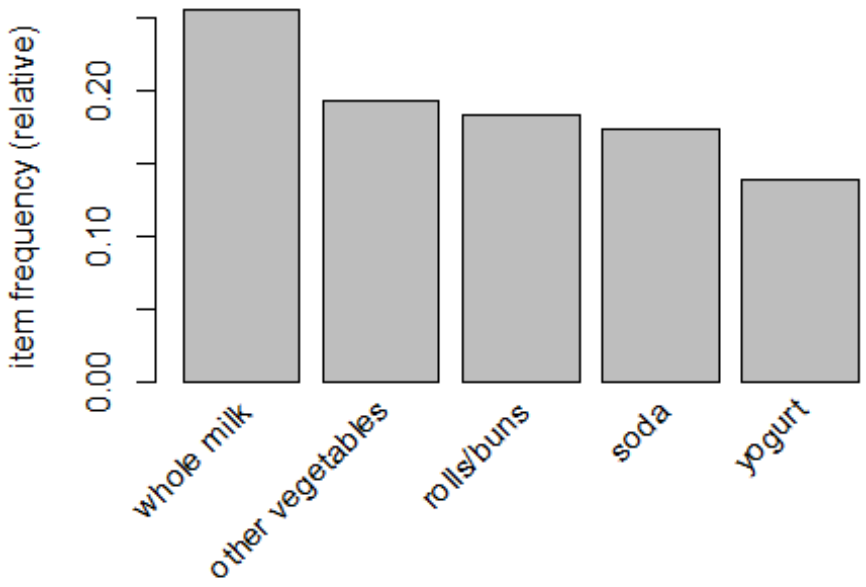
```
txn = read.transactions(file="groceries_v2.csv", rm.duplicates= TRUE,
format="basket",sep=",")

## Warning in readLines(file, encoding = encoding): incomplete final line
found on
## 'groceries_v2.csv'

head(txn)

## transactions in sparse format with
##  6 transactions (rows) and
##  169 items (columns)

itemFrequencyPlot(txn, topN = 5)
```



#Setting support and confidence levels

```
rules <- apriori(txn, parameter = list(sup = 0.01, conf = 0.5,
target="rules"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target   ext
##      10  rules FALSE
##
## 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), 9834 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 ... [15 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

#Sorting rules according to confidence:

```
rules_sorted <- sort(rules, by='confidence', decreasing = TRUE)
inspect(rules_sorted)
```

```
##      lhs                                    rhs                support
## [1]  {citrus fruit,root vegetables}      => {other vegetables}
0.01037218
## [2]  {root vegetables,tropical fruit}    => {other vegetables}
0.01230425
## [3]  {curd,yogurt}                       => {whole milk}
0.01006711
## [4]  {butter,other vegetables}           => {whole milk}
0.01149075
## [5]  {root vegetables,tropical fruit}    => {whole milk}
0.01199919
## [6]  {root vegetables,yogurt}            => {whole milk}
0.01454139
## [7]  {domestic eggs,other vegetables}    => {whole milk}
0.01230425
## [8]  {whipped/sour cream,yogurt}         => {whole milk}
0.01088062
## [9]  {rolls/buns,root vegetables}        => {whole milk}
0.01271100
## [10] {other vegetables,pip fruit}        => {whole milk}
0.01352451
## [11] {tropical fruit,yogurt}             => {whole milk}
0.01515152
```

```
## [12] {other vegetables,yogurt}             => {whole milk}
0.02226968
## [13] {other vegetables,whipped/sour cream} => {whole milk}
0.01464308
## [14] {rolls/buns,root vegetables}          => {other vegetables}
0.01220256
## [15] {root vegetables,yogurt}              => {other vegetables}
0.01291438
##      confidence lift     count
## [1]  0.5862069  3.030893 102
## [2]  0.5845411  3.022280 121
## [3]  0.5823529  2.278893  99
## [4]  0.5736041  2.244657 113
## [5]  0.5700483  2.230742 118
## [6]  0.5629921  2.203130 143
## [7]  0.5525114  2.162116 121
## [8]  0.5245098  2.052539 107
## [9]  0.5230126  2.046679 125
## [10] 0.5175097  2.025146 133
## [11] 0.5173611  2.024564 149
## [12] 0.5128806  2.007030 219
## [13] 0.5070423  1.984184 144
## [14] 0.5020921  2.595990 120
## [15] 0.5000000  2.585174 127
```