

Importing the libraries and previewing the data

```
#install.packages('arules')
library(arules)
```

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

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

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

READING DATA

```
df =read.transactions("http://bit.ly/SupermarketDatasetII", sep = ",")
```

```
## Warning in asMethod(object): removing duplicated items in transactions
```

Checking the data

```
head(df)
```

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

```
tail(df)
```

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

```
summary(df)
```

```
## transactions as itemMatrix in sparse format with
## 7501 rows (elements/itemsets/transactions) and
## 119 columns (items) and a density of 0.03288973
##
## most frequent items:
## mineral water      eggs      spaghetti french fries      chocolate
##           1788           1348           1306           1282           1229
##           (Other)
##           22405
```

```
##
## element (itemset/transaction) length distribution:
## sizes
##      1      2      3      4      5      6      7      8      9     10     11     12     13     14     15     16
## 1754 1358 1044  816  667  493  391  324  259  139  102   67   40   22   17    4
##      18     19     20
##       1      2      1
##
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.000   2.000   3.000   3.914   5.000  20.000
##
## includes extended item information - examples:
##              labels
## 1             almonds
## 2 antioxydant juice
## 3             asparagus
```

Cleaning the data

```
# Previewing our first 5 rows
#
inspect(df[1:5])
```

```
##      items
## [1] {almonds,
##      antioxydant juice,
##      avocado,
##      cottage cheese,
##      energy drink,
##      frozen smoothie,
##      green grapes,
##      green tea,
##      honey,
##      low fat yogurt,
##      mineral water,
##      olive oil,
##      salad,
##      salmon,
##      shrimp,
##      spinach,
##      tomato juice,
##      vegetables mix,
##      whole wheat flour,
##      yams}
## [2] {burgers,
##      eggs,
##      meatballs}
## [3] {chutney}
## [4] {avocado,
##      turkey}
## [5] {energy bar,
##      green tea,
```

```
##      milk,
##      mineral water,
##      whole wheat rice}
```

Association Rules

```
items = as.data.frame(itemLabels(df))
colnames(items) <- "Item"
head(items, 5)
```

```
##           Item
## 1      almonds
## 2 antioxydant juice
## 3      asparagus
## 4      avocado
## 5    babies food
```

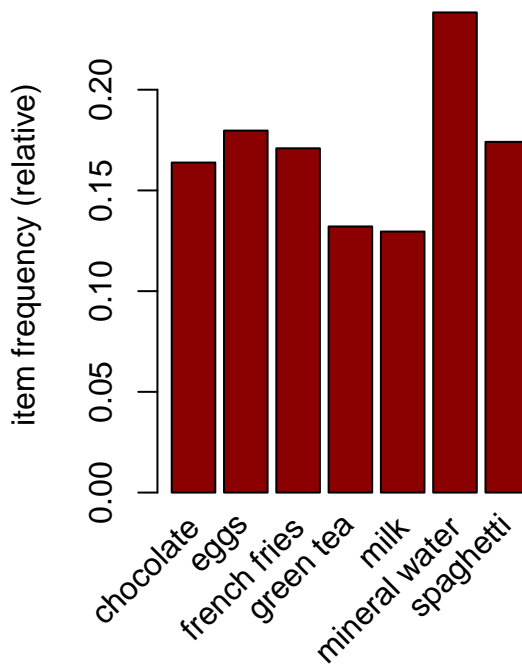
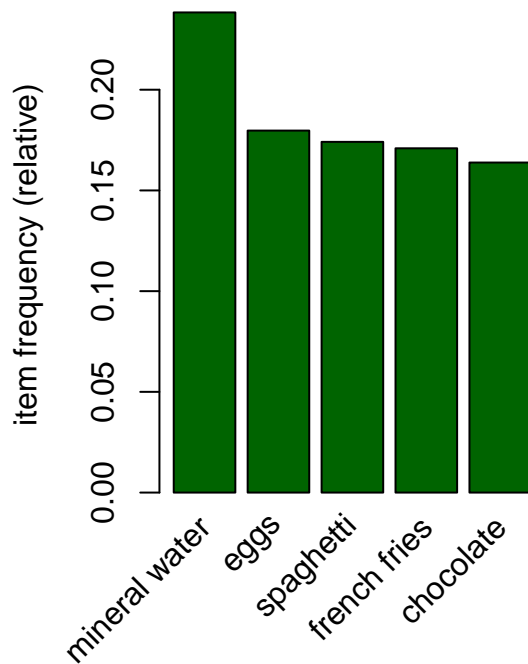
```
# Exploring the frequency of some articles
itemFrequency(df[, 8:10], type = "absolute")
```

```
##   black tea blueberries  body spray
##         107          69          86
```

```
round(itemFrequency(df[, 8:10], type = "relative")*100, 2)
```

```
##   black tea blueberries  body spray
##         1.43          0.92          1.15
```

```
# Displaying top 5 most common items in the dataset
#
par(mfrow = c(1, 2))
# plot the frequency of items
itemFrequencyPlot(df, topN = 5, col="darkgreen")
itemFrequencyPlot(df, support = 0.1, col="darkred")
```



```
# Building a model based on association
ass = apriori (df, parameter = list(supp = 0.001, conf = 0.8))
```

```
## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport maxtime support minlen
##          0.8   0.1   1 none FALSE             TRUE     5  0.001     1
## maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
## filter tree heap memopt load sort verbose
##    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 7
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[119 item(s), 7501 transaction(s)] done [0.00s].
## sorting and recoding items ... [116 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [74 rule(s)] done [0.00s].
## creating S4 object ... done [0.00s].
```

```
ass
```

```
## set of 74 rules
```

```
# Ordering the rules by level of confidence
```

```
ass = sort(ass, by="lift", decreasing=TRUE)  
inspect(ass[1:10])
```

##	lhs	rhs	support	confidence	coverage	lift	count
## [1]	{eggs, mineral water, pasta}	=> {shrimp}	0.001333156	0.9090909	0.001466471	12.722185	1
## [2]	{french fries, mushroom cream sauce, pasta}	=> {escalope}	0.001066524	1.0000000	0.001066524	12.606723	1
## [3]	{milk, pasta}	=> {shrimp}	0.001599787	0.8571429	0.001866418	11.995203	1
## [4]	{mushroom cream sauce, pasta}	=> {escalope}	0.002532996	0.9500000	0.002666311	11.976387	1
## [5]	{chocolate, ground beef, milk, mineral water, spaghetti}	=> {frozen vegetables}	0.001066524	0.8888889	0.001199840	9.325253	1
## [6]	{herb & pepper, mineral water, rice}	=> {ground beef}	0.001333156	0.9090909	0.001466471	9.252498	1
## [7]	{grated cheese, mineral water, rice}	=> {ground beef}	0.001066524	0.8888889	0.001199840	9.046887	1
## [8]	{cake, meatballs, mineral water}	=> {milk}	0.001066524	1.0000000	0.001066524	7.717078	1
## [9]	{escalope, hot dogs, mineral water}	=> {milk}	0.001066524	0.8888889	0.001199840	6.859625	1
## [10]	{meatballs, whole wheat pasta}	=> {milk}	0.001333156	0.8333333	0.001599787	6.430898	1

```
# Ordering these rules by level of confidence
```

```
ass = sort(ass, by="confidence", decreasing=TRUE)  
inspect(ass[1:10])
```

##	lhs	rhs	support	confidence	coverage	lift	count
## [1]	{french fries, mushroom cream sauce, pasta}	=> {escalope}	0.001066524	1.0000000	0.001066524	12.606723	8
## [2]	{cake, meatballs, mineral water}	=> {milk}	0.001066524	1.0000000	0.001066524	7.717078	8
## [3]	{ground beef, light cream,						

##	olive oil}	=> {mineral water}	0.001199840	1.0000000	0.001199840	4.195190	9
## [4]	{cake,						
##	olive oil,						
##	shrimp}	=> {mineral water}	0.001199840	1.0000000	0.001199840	4.195190	9
## [5]	{mushroom cream sauce,						
##	pasta}	=> {escalope}	0.002532996	0.9500000	0.002666311	11.976387	19
## [6]	{red wine,						
##	soup}	=> {mineral water}	0.001866418	0.9333333	0.001999733	3.915511	14
## [7]	{eggs,						
##	mineral water,						
##	pasta}	=> {shrimp}	0.001333156	0.9090909	0.001466471	12.722185	10
## [8]	{herb & pepper,						
##	mineral water,						
##	rice}	=> {ground beef}	0.001333156	0.9090909	0.001466471	9.252498	10
## [9]	{ground beef,						
##	pancakes,						
##	whole wheat rice}	=> {mineral water}	0.001333156	0.9090909	0.001466471	3.813809	10
## [10]	{frozen vegetables,						
##	milk,						
##	spaghetti,						
##	turkey}	=> {mineral water}	0.001199840	0.9000000	0.001333156	3.775671	9

RECOMMENDATIONS

1. The top 3 purchased items were: mineral water, eggs, pasta,
2. Judging by the confidence interval, cake, eggs, pasta, meatballs and mineral water should be placed in aisles that are closer to each other since there
3. Foods high in protein should also be placed together since customers tend to purchase them together