# Assignment 4

## CMSC462

## 2024-11-24

```r
library(arules)
```

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

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

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

```r
library(arulesViz)
library(readxl)
library(plyr)
library(ggplot2)

Data <- read_excel("BreadBasket_DMS.xlsx")
df <- data.frame(Data)
df$Item <- as.factor(df$Item)
df$Transaction <- as.factor(df$Transaction)
head(df)
```

```
##          Date                Time Transaction          Item
## 1 2016-10-30 1899-12-31 09:58:11           1         Bread
## 2 2016-10-30 1899-12-31 10:05:34           2   Scandinavian
## 3 2016-10-30 1899-12-31 10:05:34           2   Scandinavian
## 4 2016-10-30 1899-12-31 10:07:57           3 Hot chocolate
## 5 2016-10-30 1899-12-31 10:07:57           3           Jam
## 6 2016-10-30 1899-12-31 10:07:57           3       Cookies
```

```r
# Convert the data into transaction format, split items by transaction
Cafe <- split(df$Item, df$Transaction)
head(Cafe)
```

```
## $`1`
## [1] Bread
## 95 Levels: Adjustment Afternoon with the baker Alfajores ... Victorian Sponge
##
## $`2`
```

```
## [1] Scandinavian Scandinavian
## 95 Levels: Adjustment Afternoon with the baker Alfajores ... Victorian Sponge
##
## $`3`
## [1] Hot chocolate Jam          Cookies
## 95 Levels: Adjustment Afternoon with the baker Alfajores ... Victorian Sponge
##
## $`4`
## [1] Muffin
## 95 Levels: Adjustment Afternoon with the baker Alfajores ... Victorian Sponge
##
## $`5`
## [1] Coffee Pastry Bread
## 95 Levels: Adjustment Afternoon with the baker Alfajores ... Victorian Sponge
##
## $`6`
## [1] Medialuna Pastry    Muffin
## 95 Levels: Adjustment Afternoon with the baker Alfajores ... Victorian Sponge
```

```r
# Convert to transaction object for arules package
Cafe_transactions <- as(Cafe, "transactions")
```

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

```r
head(Cafe_transactions)
```

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

```r
# 1 – Descriptive statistics – Summary, Absolute Item Frequency Histogram
summary(df)
```

```
##       Date                              Time
##  Min.   :2016-10-30 00:00:00.00   Min.   :1899-12-31 01:21:05.0
##  1st Qu.:2016-12-03 00:00:00.00   1st Qu.:1899-12-31 10:56:40.0
##  Median :2017-01-21 00:00:00.00   Median :1899-12-31 12:37:30.0
##  Mean   :2017-01-17 02:09:18.26   Mean   :1899-12-31 12:45:01.4
##  3rd Qu.:2017-02-28 00:00:00.00   3rd Qu.:1899-12-31 14:29:37.0
##  Max.   :2017-04-09 00:00:00.00   Max.   :1899-12-31 23:38:41.0
##
##   Transaction         Item
##  6474   :   12   Coffee :5471
##  6279   :   11   Bread  :3325
##  6412   :   11   Tea    :1435
##  6716   :   11   Cake   :1025
##  3673   :   10   Pastry : 856
##  6045   :   10   NONE   : 786
##  (Other):21228   (Other):8395
```

```r
# Calculate absolute item frequency
absolute_frequencies <- itemFrequency(Cafe_transactions, type = "absolute")
absolute_freq_df <- data.frame(
  Item = names(absolute_frequencies),
  Frequency = absolute_frequencies)

# Sort and filter top 10 items
top_10_items <- absolute_freq_df[order(-absolute_freq_df$Frequency), ][1:10, ]

# Plot top 10 items with ggplot2
ggplot(top_10_items, aes(x = reorder(Item, Frequency), y = Frequency)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Top 10 Items by Frequency", x = "Items", y = "Frequency")
```

## Top 10 Items by Frequency
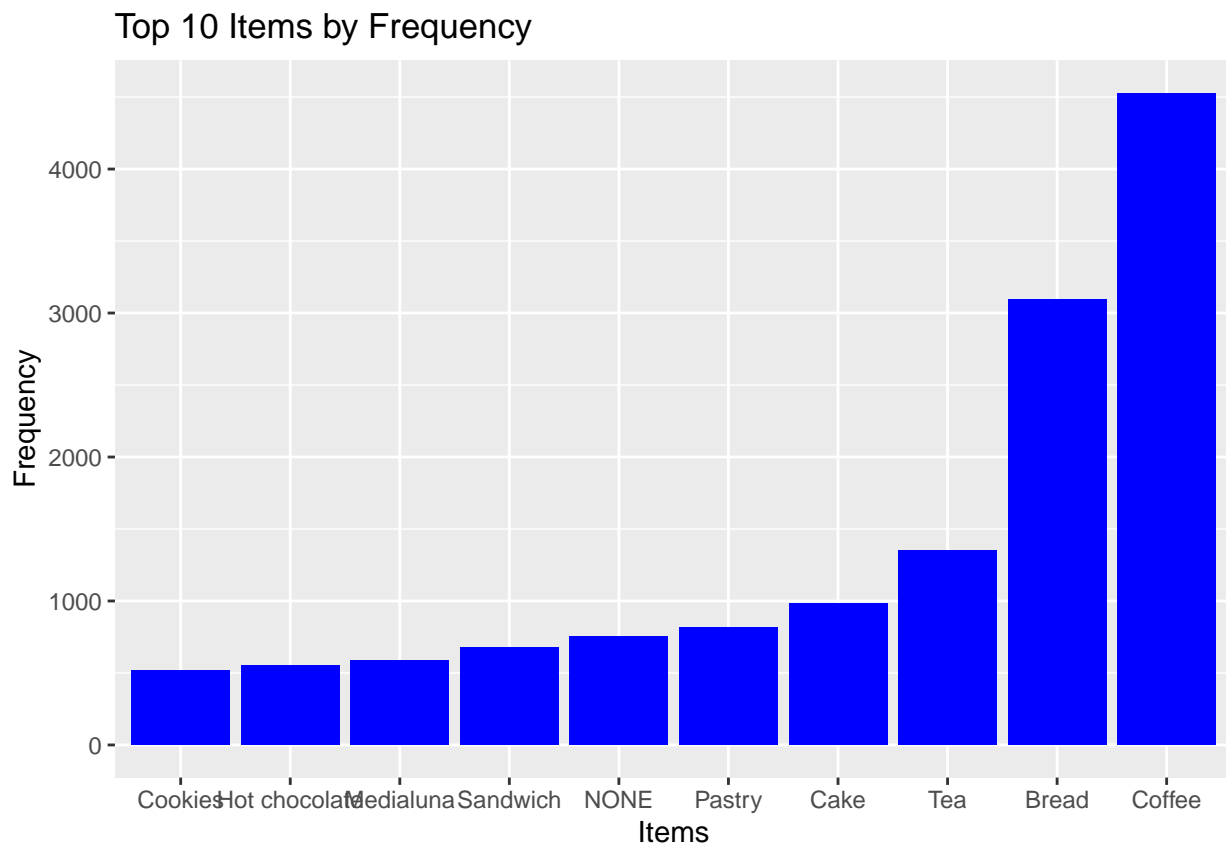


```r
# 2 - Do a market basket analysis and uncover the association rules. Among the association
# rules, filter out any rule with "Coffee" on the right-hand side. Also, set the maxlen as 3
# or 4 and minlen as 2. Next, choose a metrics and using it select top five rules and
# describe those. Do those rules make sense?

# Generate association rules
rules <- apriori(
  Cafe_transactions,
  parameter = list(support = 0.001, confidence = 0.1, maxlen = 4, minlen = 2)
)
```

## Apriori

```
## 
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5   0.001      2
##  maxlen target  ext
##       4  rules TRUE
## 
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
## 
## Absolute minimum support count: 9
## 
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[95 item(s), 9531 transaction(s)] done [0.00s].
## sorting and recoding items ... [58 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4


## Warning in apriori(Cafe_transactions, parameter = list(support = 0.001, :
## Mining stopped (maxlen reached). Only patterns up to a length of 4 returned!


##  done [0.00s].
## writing ... [578 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
# Filter rules with Coffee on rhs
filtered_rules <- subset(rules, rhs %in% "Coffee")

# Exclude rules where Coffee is on rhs
rules_no_coffee_rhs <- subset(rules, !(rhs %in% "Coffee"))
inspect(rules_no_coffee_rhs[1:20])
```

```
##      lhs                       rhs                     support    confidence
## [1]  {Eggs}                 => {Bread}                 0.001468891 0.5000000
## [2]  {Focaccia}             => {Bread}                 0.001993495 0.3518519
## [3]  {Tartine}              => {NONE}                  0.001363970 0.2826087
## [4]  {Bakewell}             => {Tea}                   0.001259049 0.2500000
## [5]  {Bakewell}             => {Bread}                 0.001154129 0.2291667
## [6]  {Vegan mincepie}       => {Cake}                  0.001259049 0.2307692
## [7]  {Vegan mincepie}       => {Bread}                 0.001363970 0.2500000
## [8]  {Art Tray}             => {NONE}                  0.001154129 0.2894737
## [9]  {Art Tray}             => {Tea}                   0.001154129 0.2894737
## [10] {Extra Salami or Feta} => {Salad}                 0.001678733 0.4210526
## [11] {Salad}                => {Extra Salami or Feta}  0.001678733 0.1616162
## [12] {Extra Salami or Feta} => {NONE}                  0.001783653 0.4473684
## [13] {Extra Salami or Feta} => {Bread}                 0.001049208 0.2631579
## [14] {Keeping It Local}     => {Tea}                   0.001783653 0.2698413
## [15] {Keeping It Local}     => {Bread}                 0.002203336 0.3333333
## [16] {The Nomad}            => {NONE}                  0.001259049 0.2068966
## [17] {The Nomad}            => {Tea}                   0.001783653 0.2931034
## [18] {The Nomad}            => {Bread}                 0.001678733 0.2758621
## [19] {Frittata}             => {Tea}                   0.001888574 0.2222222
```

```
## [20] {Frittata}            => {Bread}               0.002623020 0.3086420
##       coverage    lift      count
## [1]   0.002937782  1.5387472 14
## [2]   0.005665722  1.0828221 19
## [3]   0.004826356  3.5770830 13
## [4]   0.005036198  1.7650000 12
## [5]   0.005036198  0.7052591 11
## [6]   0.005455881  2.2374990 12
## [7]   0.005455881  0.7693736 13
## [8]   0.003986990  3.6639757 11
## [9]   0.003986990  2.0436842 11
## [10] 0.003986990 40.5358852 16
## [11] 0.010387158 40.5358852 16
## [12] 0.003986990  5.6625079 17
## [13] 0.003986990  0.8098669 10
## [14] 0.006610009  1.9050794 17
## [15] 0.006610009  1.0258314 21
## [16] 0.006085406  2.6187663 12
## [17] 0.006085406  2.0693103 17
## [18] 0.006085406  0.8489640 16
## [19] 0.008498584  1.5688889 18
## [20] 0.008498584  0.9498439 25
```

```r
# Sort top 5 rules by lift
top_rules <- head(sort(rules_no_coffee_rhs, by = "lift"), 5)

# Display the top 5 rules
inspect(top_rules)
```

```
##      lhs                          rhs                     support
## [1] {Coffee, Salad}               => {Extra Salami or Feta} 0.001468891
## [2] {Coffee, Extra Salami or Feta} => {Salad}               0.001468891
## [3] {Salad}                       => {Extra Salami or Feta} 0.001678733
## [4] {Extra Salami or Feta}        => {Salad}               0.001678733
## [5] {Alfajores, Cookies}          => {Juice}               0.001049208
##      confidence coverage    lift     count
## [1] 0.2258065  0.006505089 56.63582 14
## [2] 0.4516129  0.003252544 43.47801 14
## [3] 0.1616162  0.010387158 40.53589 16
## [4] 0.4210526  0.003986990 40.53589 16
## [5] 0.4347826  0.002413178 11.35319 10
```

```r
cat("High Lift indicates strong association; rules with high lift show strong correlation
    between lhs and rhs.
    1. Customers who purchase both coffee and salad are 56.64 times more likely to
    also add 'Extra Salami or Feta' than would be expected at random.
    2. Customers who purchase both coffee and 'Extra Salami or Feta' are 43.48 times
    more likely to include a salad in their purchase than would be expected at random.
    3. Customers who purchase a salad are 40.54 times more likely to also purchase
    'Extra Salami or Feta' than would be expected at random.
    4. Customers ordering 'Extra Salami or Feta' are 40.54 times more likely to
    include a salad than would be expected at random.
    5. Customers who purchase Alfajores and cookies are 11.35 times more likely to
```

```
    include juice in their purchase than would be expected at random.
    ")
```

## High Lift indicates strong association; rules with high lift show strong correlation
##     between lhs and rhs.
##     1. Customers who purchase both coffee and salad are 56.64 times more likely to
##     also add 'Extra Salami or Feta' than would be expected at random.
##     2. Customers who purchase both coffee and 'Extra Salami or Feta' are 43.48 times
##     more likely to include a salad in their purchase than would be expected at random.
##     3. Customers who purchase a salad are 40.54 times more likely to also purchase
##     'Extra Salami or Feta' than would be expected at random.
##     4. Customers ordering 'Extra Salami or Feta' are 40.54 times more likely to
##     include a salad than would be expected at random.
##     5. Customers who purchase Alfajores and cookies are 11.35 times more likely to
##     include juice in their purchase than would be expected at random.
##