ML1000 Assignment 3

Crystal Zhu

07/03/2021

Data Understanding

How do we merge the data files?

There are six data files, excluding the sample_submission.csv file, from the Instacart Market Basket Analysis data - aisles.csv, departments.csv, order_products___train.csv, order_products___prior.csv, orders.csv and products.csv.

(Add data file descriptions later!)

Steps:

1. Merged the aisles data with the products data to obtain Merged dataset 1, so that we know which aisle each product belongs to.
2. Combined the Merged dataset 1 with the department data to obtain Merged dataset 2, so we know which aisle and department each product is from.
3. Add Merged dataset 2, which contains product full information, to order_products___train and order_products___prior files, respectively, to obtain Merged dataset 3 (Train) and Merged dataset 4 (Prior), so that we know the product information (e.g. product names, aisles and departments they belong to) of the products in the training and prior orders.

```
library(arules)


####### Convert the merged dataset into a TRANSACTION FORM FOR R ############
X=read.csv("C:/Users/yunan/Downloads/York U/Machine Learning Cert/Assignment 3/data/orders_TRAIN_produc


#split(x,f) divides the data in the vector x into the groups defined by f
#split(x,f) returns a list, and the components of the list are named by the levels of f
#so basically it returns the frequency table at each level of f(aka, the frequency of each product in o

orders=unique(X$order_id)

set.seed(123)
#select 100 unique orders for demo
order_sample=sample(orders,100,replace = F)

X1=subset(X,order_id %in% order_sample)

X1$order_id=as.factor(X1$order_id)
X1$product_id=as.factor(X1$product_id)
X1$product_name=as.factor(X1$product_name)
```

```r
length(unique(X1$product_name))
```

```
## [1] 786
```

```r
#39,123 unique product names

#create the item list
Order_by_product <- split(X1$product_name, X1$order_id)

Order_by_product[[1]]
```

```
##  [1] Brown Fertile Large Grade AA Eggs   Mini Whole Wheat Pita Bread
##  [3] Organic Spaghetti Squash            Uncured Black Forest Ham
##  [5] Roasted Turkey Breast               Seedless Red Grapes
##  [7] Almond Meal/Flour                   Good Seed Organic Thin Sliced Bread
##  [9] Sea Salt Roasted Seaweed            Banana
## [11] Original Hummus                     Organic Fuji Apple
## [13] Almondmilk Creamer, Vanilla         Organic Reduced Fat Milk
## 786 Levels: 0% Greek Strained Yogurt ... Zucchini Squash
```

```r
#the first element of Order_by_product is all the items from the first order
length(Order_by_product)
```

```
## [1] 100
```

```r
#97
length(unique(X1$order_id))
```

```
## [1] 100
```

```r
#97 - so the length of Order_by_product is the number of orders

#Coerce the Item List to the Transactions class
#convert transaction data in dataframe to transaction object
X1_trans <- as(Order_by_product, "transactions")

X1_trans@data@i[1:5]
```

```
## [1]  24  25  50  80 231
```

```r
#the product index/position from each order sequentially
X1_trans@data@p[1:5]
```

```
## [1]  0 14 25 29 37
```

```r
#the cummulative number of iterms from each order
X1_trans@data@Dim
```

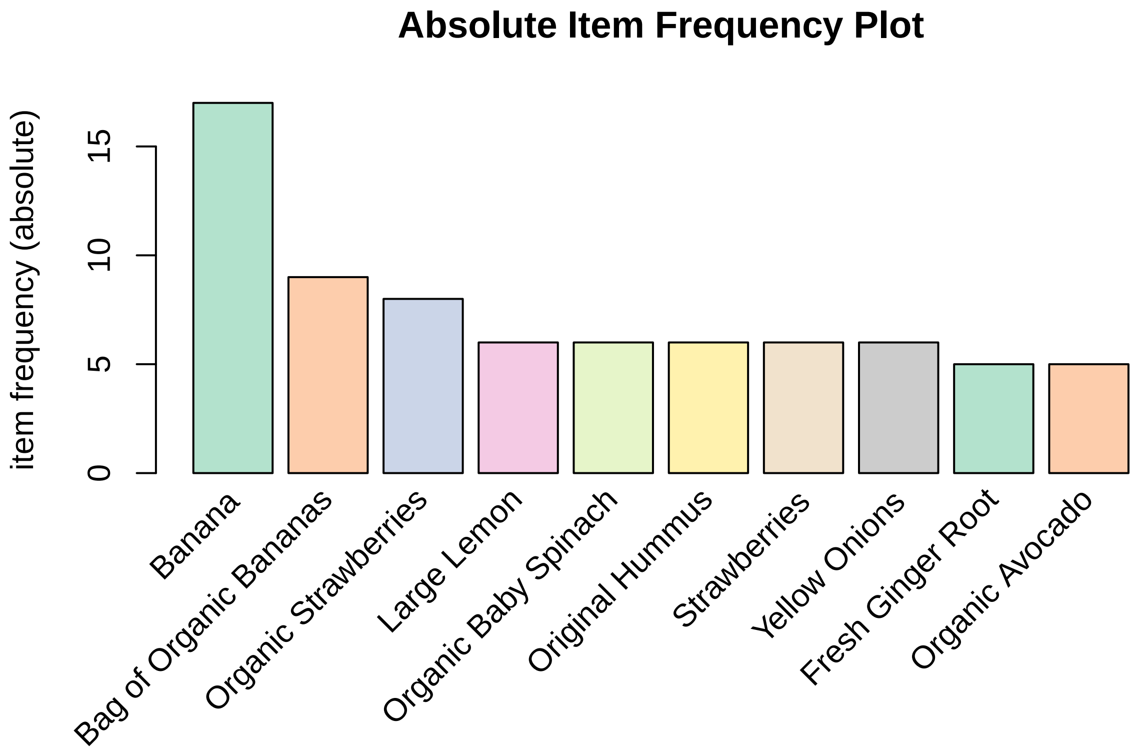
```
## [1] 786 100
```

```r
#number of unique products * number of orders
X1_trans@itemInfo$labels[1:5]
```

```
## [1] "0% Greek Strained Yogurt"  "1% Lowfat Milk"
## [3] "100% Florida Orange Juice" "100% Juice, Variety Pack"
## [5] "100% Liquid Egg Whites"
```

```r
#labels contain the product names in our case
```

```
library(RColorBrewer)
itemFrequencyPlot(X1_trans,topN=10,type="absolute",col=brewer.pal(8,'Pastel2'), main="Absolute Item Fre
```



```
#apply apriori rule
X_apri_rule=apriori(X1_trans,parameter=list(supp=0.02, conf=0.5))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.02      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: 2
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[786 item(s), 100 transaction(s)] done [0.00s].
## sorting and recoding items ... [122 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [174 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(X_apri_rule[1:15])
```

```
##      lhs                                          rhs                          support confidence coverage
## [1]  {Zero Calorie Cola}                       => {Soda}                          0.02  1.0000000     0.02 5
## [2]  {Soda}                                    => {Zero Calorie Cola}             0.02  1.0000000     0.02 5
## [3]  {Clementines}                             => {Bag of Organic Bananas}        0.02  1.0000000     0.02 1
## [4]  {Berry Medley}                            => {Organic Baby Spinach}          0.02  1.0000000     0.02 1
## [5]  {Organic Yellow Onion}                    => {Organic Zucchini}              0.02  1.0000000     0.02 2
## [6]  {Coffee Chocolate Bar}                    => {Banana}                        0.02  1.0000000     0.02
## [7]  {100% Raw Coconut Water}                  => {Sparkling Lemon Water}         0.02  1.0000000     0.02 5
## [8]  {Sparkling Lemon Water}                   => {100% Raw Coconut Water}        0.02  1.0000000     0.02 5
## [9]  {Organic No Salt Added Diced Tomatoes} => {Yellow Onions}                 0.02  1.0000000     0.02 1
## [10] {Small Hass Avocado}                      => {Banana}                        0.02  1.0000000     0.02
## [11] {Crunchy Almond Butter}                   => {Banana}                        0.02  1.0000000     0.02
## [12] {Feta Cheese Crumbles}                    => {Organic Blueberries}           0.02  1.0000000     0.02 3
## [13] {Organic Blueberries}                     => {Feta Cheese Crumbles}          0.02  0.6666667     0.03 3
## [14] {White Corn Tortillas}                    => {Banana}                        0.02  1.0000000     0.02
## [15] {Orange Bell Pepper}                      => {Asparagus}                     0.02  1.0000000     0.02 3
```

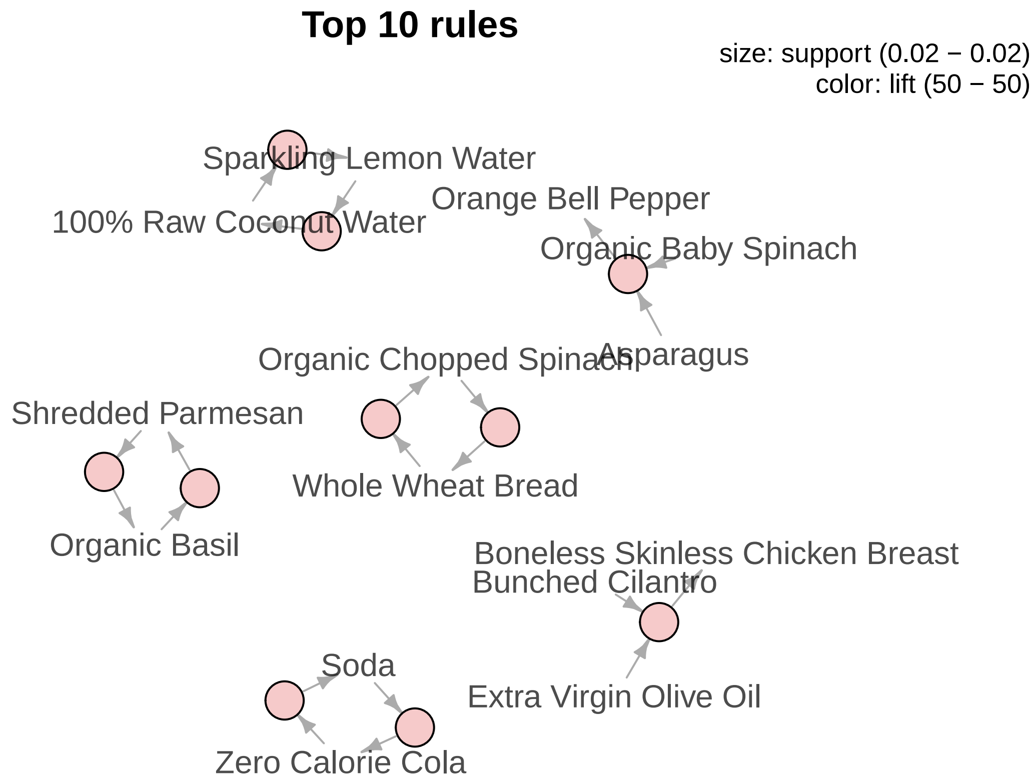
```
#if we select orders from the same person/if the number of transactions is not large, the overlapping o


 summary(X_apri_rule)
```

```
## set of 174 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4
## 100  66   8
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   2.000   2.471   3.000   4.000
##
## summary of quality measures:
##     support          confidence        coverage            lift
##  Min.   :0.02000   Min.   :0.5000   Min.   :0.02000   Min.   : 2.941
##  1st Qu.:0.02000   1st Qu.:0.6667   1st Qu.:0.02000   1st Qu.: 8.333
##  Median :0.02000   Median :1.0000   Median :0.02000   Median :16.667
##  Mean   :0.02075   Mean   :0.8228   Mean   :0.02736   Mean   :20.246
##  3rd Qu.:0.02000   3rd Qu.:1.0000   3rd Qu.:0.03000   3rd Qu.:33.333
##  Max.   :0.04000   Max.   :1.0000   Max.   :0.06000   Max.   :50.000
##      count
##  Min.   :2.000
##  1st Qu.:2.000
##  Median :2.000
##  Mean   :2.075
##  3rd Qu.:2.000
##  Max.   :4.000
##
## mining info:
##      data ntransactions support confidence
##  X1_trans           100    0.02        0.5
```

```
topRules <- head(X_apri_rule, n = 10, by = "lift")

library(arulesViz)
#interactive plot in html
#plot(topRules, method = "graph",  engine = "htmlwidget",main="Top 10 rules")
 plot(topRules, method = "graph",  main="Top 10 rules")
```


Top 10 rules
size: support (0.02 − 0.02)
color: lift (50 − 50)
Sparkling Lemon Water
100% Raw Coconut Water
Orange Bell Pepper
Organic Baby Spinach
Asparagus
Organic Chopped Spinach
Shredded Parmesan
Whole Wheat Bread
Organic Basil
Boneless Skinless Chicken Breast
Bunched Cilantro
Extra Virgin Olive Oil
Soda
Zero Calorie Cola