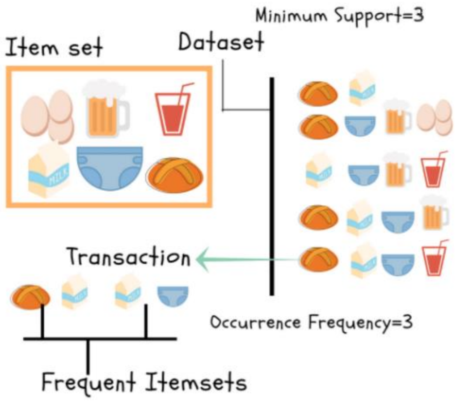
Groceries Market Basket Analysis

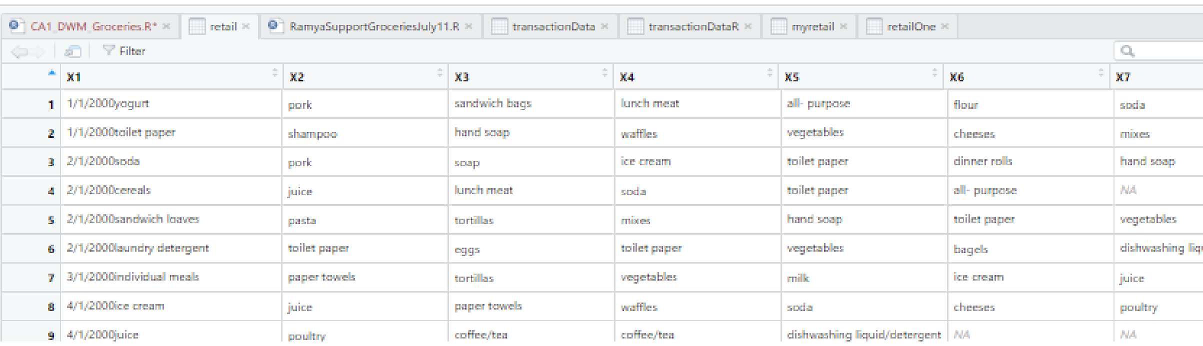


Introduction

This project discusses how a groceries dataset can be tidied using the techniques outlined in "Tidy Data" (Wickham, 2014). It then goes on to discuss how the association rules can be used to analyse the tidied grocery dataset. Data analysis is limited in its efforts to draw significant or meaningful information from data if it has not been tidied. With Hadley Wickham' packages such as Tidyverse, which is a collection of packages, data tidying and cleansing is a speedy and efficient process.

Cleaning data Grocery dataset

The original groceries raw data was as follows:



	X1	X2	X3	X4	X5	X6	X7
1	1/1/2000yogurt	pork	sandwich bags	lunch meat	all- purpose	flour	soda
2	1/1/2000toilet paper	shampoo	hand soap	waffles	vegetables	cheeses	mixes
3	2/1/2000soda	pork	soap	ice cream	toilet paper	dinner rolls	hand soap
4	2/1/2000cereals	juice	lunch meat	soda	toilet paper	all- purpose	NA
5	2/1/2000sandwich loaves	pasta	tortillas	mixes	hand soap	toilet paper	vegetables
6	2/1/2000laundry detergent	toilet paper	eggs	toilet paper	vegetables	bagels	dishwashing liq
7	3/1/2000individual meals	paper towels	tortillas	vegetables	milk	ice cream	juice
8	4/1/2000ice cream	juice	paper towels	waffles	soda	cheeses	poultry
9	4/1/2000juice	poultry	coffee/tea	coffee/tea	dishwashing liquid/detergent	NA	NA

Figure 1 Original Groceries Dataset

In Figure 1 Original Groceries Dataset, each row represents a selection of items in a shopping basket.

The aim of the data cleaning is to combine all shopping items into a single cell because this is the ideal format for the arules package apriori() function. Tidy data makes it easy for an analyst or a computer to extract needed variables because it provides a standard way of structuring data (Wickham, 2014).

The apriori() function is the preferred function for analysing patterns in a shopping basket. In the Figure 1 Original Groceries Dataset, the columns contain values instead of variables so one of the first messy data cleaning tasks is to convert columns to rows.

There are several steps involved in getting the groceries data in a single column separated by commas. Two of more important Hadley Wickham's functions for this purpose are:

- the melt() function which belongs to the reshape package (Wickham, 2007)
- the ddply() function which belongs to the dplyr package (Wickham, 2011)

However, before using the melt and the ddply functions, general messy data needs to be cleaned up. The first column needs to be separated into two columns: date and item. This is achieved using a mixture of the sub() and separate() functions. We also need some way of distinguishing shopping baskets that occur on the same day. This is achieved by adding another column which in this project is called "Shopper" using the rowid_to_column() function.

Having converted column values to rows using the melt() function, the next task is to convert all items in a basket to a single comma separated cell using the Hadley Wickham ddply() function.

Shopper	Date	Item
1	1/1/2000	toilet paper
1	1/1/2000	shampoo
1	1/1/2000	hand soap
1	1/1/2000	waffles
1	1/1/2000	vegetables
1	1/1/2000	cheeses
1	1/1/2000	mixes
1	1/1/2000	milk
1	1/1/2000	sandwich bags
1	1/1/2000	laundry detergent
1	1/1/2000	dishwashing liquid/detergent
1	1/1/2000	waffles
1	1/1/2000	individual meals
1	1/1/2000	hand soap

Figure 2- Item values as rows

The groceries data before applying the ddply() function is shown in Figure 2- Item values as rows and the groceries data after applying the ddply() function is shown in Figure 3 - Basket item values comma separated.

Filter

Shopper	Date	V1
1	1/1/2000	yogurt,pork,sandwich bags,lunch meat,all- purpose,flour,so...
2	1/1/2000	toilet paper,shampoo,hand soap,waffles,vegetables,cheeses,...
3	2/1/2000	soda,pork,soap,ice cream,toilet paper,dinner rolls,hand soap...
4	2/1/2000	cereals,juice,lunch meat,soda,toilet paper,all- purpose
5	2/1/2000	sandwich loaves,pasta,tortillas,mixes,hand soap,toilet paper,...
6	2/1/2000	laundry detergent,toilet paper,eggs,toilet paper,vegetables,...
7	3/1/2000	individual meals,paper towels,tortillas,vegetables,milk,ice cre...
8	4/1/2000	ice cream,juice,paper towels,waffles,soda,cheeses,poultry,toi...
9	4/1/2000	juice,poultry,coffee/tea,coffee/tea,dishwashing liquid/deterg...
10	5/1/2000	ketchup,coffee/tea,toilet paper,pork,flour,milk,soda,dishwas...
11	5/1/2000	sandwich loaves,ice cream,soda,bagels,dishwashing liquid/d...
12	6/1/2000	pork,tortillas,pork,shampoo,lunch meat,pasta,juice,bagels,ve...
13	7/1/2000	sugar,fruits,all- purpose,aluminum foil,laundry detergent,ind...
14	7/1/2000	fruits,dinner rolls,individual meals,shampoo,ketchup,cereals,...
15	7/1/2000	individual meals,ice cream,cereals,paper towels,bagels,mixes...
16	8/1/2000	sugar,sandwich bags,flour,juice,milk,paper towels,cereals,sa...

Figure 3 - Basket item values comma separated

The arules package read.transactions() function can be used to generate transactional data for use with the apriori function.

Grocery dataset association rules

When the data is read in as transaction data, a summary of the transaction data lists key data details such as the most frequent items.

```
 1499 rows (elements/itemsets/transactions) and
 39 columns (items) and a density of 0.373514

most frequent items:
                 vegetables                           poultry                        waffles
                       1088                               613                            587
dishwashing liquid/detergent                        ice cream                        (Other)
                        585                               583                          18380

element (itemset/transaction) length distribution:
sizes
  1   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27
  1  15  57  56  53  71  74  72  79  67  72  89  86  84 104  95  94 114  78  67  36  24   7   3   1
```

Figure 4 - Transaction Data Summary

The are many options when generating rules where you can specify different support, confidence and lengths. For example, specifying .1 support, .8 confidence and max length = 10 results in 443 rules compared 39,3289 rules when the support is set to .01.

```
     lhs                                rhs          support   confidence coverage  lift     count
[1]  {sugar}                         => {vegetables} 0.2933333 0.8000000  0.3666667 1.101928 440
[2]  {laundry detergent}             => {vegetables} 0.3040000 0.8099467  0.3753333 1.115629 456
[3]  {yogurt}                        => {vegetables} 0.3080000 0.8148148  0.3780000 1.122334 462
[4]  {eggs}                          => {vegetables} 0.3106667 0.8146853  0.3813333 1.122156 466
[5]  {aluminum foil}                 => {vegetables} 0.3093333 0.8013817  0.3860000 1.103832 464
[6]  {hand soap,ketchup}             => {vegetables} 0.1106667 0.8058252  0.1373333 1.109952 166
[7]  {hand soap,sandwich loaves}     => {vegetables} 0.1180000 0.8428571  0.1400000 1.160960 177
[8]  {hand soap,sugar}               => {vegetables} 0.1186667 0.8127854  0.1460000 1.119539 178
[9]  {hand soap,paper towels}        => {vegetables} 0.1080000 0.8019802  0.1346667 1.104656 162
[10] {hand soap,individual meals}    => {vegetables} 0.1146667 0.8269231  0.1386667 1.139013 172
```

Figure 5 - Association Rule Summary

Support denotes how often combination appears in data. Confidence denotes % of transactions appearing on lhs (left hand side) items which also contain rhs (right hand side) item. For example, in row 6 of Figure 5 - Association Rule Summary, we can say that 80% people who buy hand soap and ketchup together also buy vegetables.

You are able to filter associations by confidence level and only include items with a confidence above a certain level, for example greater than .40. We can also sort by lift or just look at associations for a particular item, see Figure 6 - Associations Rules Sorted by Lift.

```
> basket.sorted1 <- sort(association.rules1, by = "lift")
> inspect(basket.sorted1[1:10])
     lhs                           rhs          support   confidence coverage  lift     count
[1]  {dinner rolls,eggs}        => {vegetables} 0.1440961 0.8780488  0.1641094 1.209738 216
[2]  {eggs,sandwich bags}       => {vegetables} 0.1227485 0.8761905  0.1400934 1.207178 184
[3]  {cheeses,eggs}             => {vegetables} 0.1454303 0.8755020  0.1661107 1.206229 218
[4]  {aluminum foil,sugar}      => {vegetables} 0.1260841 0.8709677  0.1447632 1.199982 189
[5]  {eggs,sandwich loaves}     => {vegetables} 0.1200801 0.8695652  0.1380921 1.198050 180
[6]  {aluminum foil,eggs}       => {vegetables} 0.1367578 0.8686441  0.1574383 1.196781 205
[7]  {cereals,laundry detergent} => {vegetables} 0.1387592 0.8666667  0.1601067 1.194056 208
[8]  {laundry detergent,yogurt} => {vegetables} 0.1334223 0.8658009  0.1541027 1.192864 200
[9]  {milk,yogurt}              => {vegetables} 0.1400934 0.8641975  0.1621081 1.190655 210
[10] {eggs,poultry}             => {vegetables} 0.1440961 0.8640000  0.1667779 1.190382 216
>
```

Figure 6 - Associations Rules Sorted by Lift

You can do a scatter plot of association rules, see Figure 7 - Associated items with Vegetables on rhs.

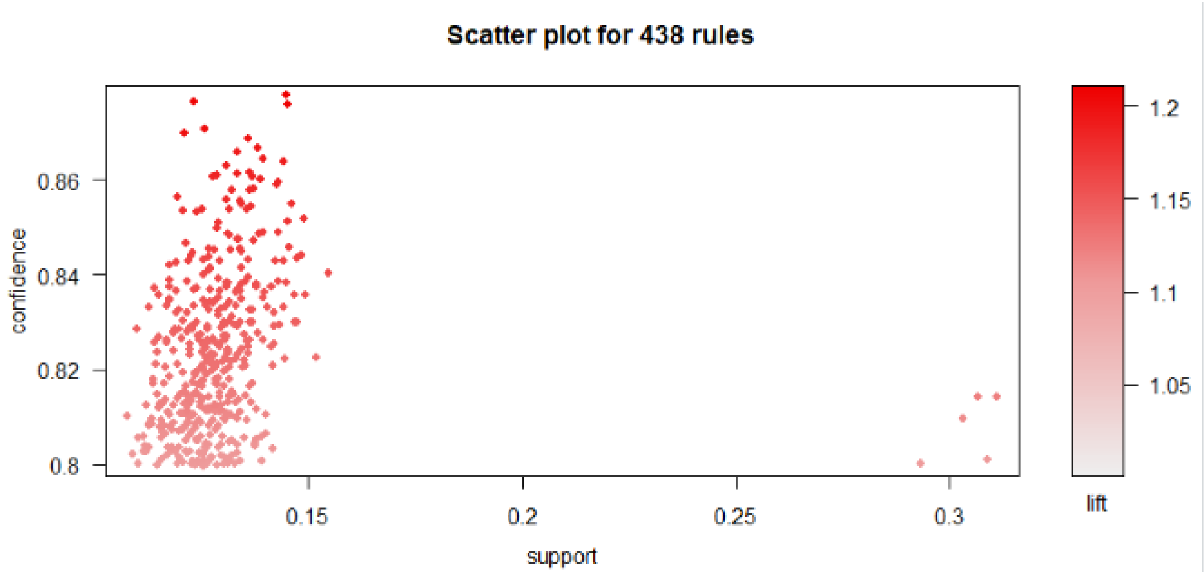


Figure 7 - Associated items with Vegetables on rhs

There is also the facility in RStudio to do interactive maps using the plotly_arules() function as in Figure 8 - Plotly_arules Interactive map

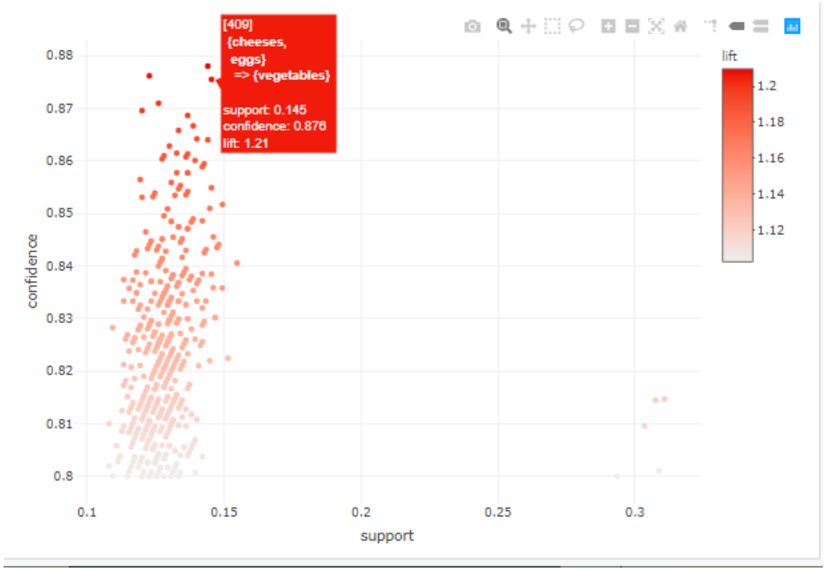


Figure 8 - Plotly_arules Interactive map

The tidy data techniques offered by Hadley Wickham offered a fast-effective way to create tidy data in a format to

References

Anon., 2020. *Package: ROCR.* [Online]
Available at: https://cran.r-project.org/web/packages/ROCR/ROCR.pdf
[Accessed 25 July 2020].

Asansha Sharma: Edureka, 2019. *Creating, Validating and Pruning Decision Tree in R.* [Online]
Available at: https://www.edureka.co/blog/implementation-of-decision-tree/#:~:text=Syntax%20%3A%20printcp%20(%20x%20)%20where,()%20function%20i.e.%20'xerror'.
[Accessed 25 July 2020].

Jason Browsnlee: Machine Learning Mastery, 2016. *Machine Learning Evaluation Metrics in R.* [Online]
Available at: https://machinelearningmastery.com/machine-learning-evaluation-metrics-in-r/
[Accessed 25 July 2020].

Paul Pandey: Medium, 2018. *A Guide to Machine Learning in R for Beginners: Decision Trees.* [Online]
Available at: https://medium.com/analytics-vidhya/a-guide-to-machine-learning-in-r-for-beginners-decision-trees-c24dfd490abb
[Accessed 25 July 2020].

Tony Yiu: Towardsdatascience.com, 2019. *Understanding Random Forest.* [Online]
Available at: https://towardsdatascience.com/understanding-random-forest-58381e0602d2
[Accessed 24 July 2020].

Wickham, H., 2007. *Reshaping Data with the reshape Package.* [Online]
Available at: https://vita.had.co.nz/papers/reshape.pdf
[Accessed 23 July 2020].

Wickham, H., 2011. *The Split-Apply-Combine Strategy for Data Analysis.* [Online]
Available at: https://www.jstatsoft.org/article/view/v040i01/v40i01.pdf
[Accessed 23 July 2020].

Wickham, H., 2014. *Tidy Data: Journal of Statistical Software.* [Online]
Available at: https://vita.had.co.nz/papers/tidy-data.pdf
[Accessed 23 July 2020].