|  |  |
| --- | --- |
| Applying Data SCience To InstaCART’s Public dataset  Project Report | Group D, APDS-01 [2017-18]  Anurag Pandey, Prashant Nair, Rajiv Ranjan, Shaveta Sachdeva, Sonia Grewal, Sunil K Gupta, Vikas Rana  Advanced Programme in Data Sciences, IIM Calcutta |

## Acknowledgment

It is our privilege to express our sincerest regards to Program directors and our mentor, for their valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project. This has been a wonderful project which has helped us to explore the world of analytics and its real world application. We also take this opportunity to thank all our professors who have directly or indirectly helped in our project. Last but not the least we express our thanks to our classmates for their cooperation and support.

## About US

Table of Contents

[Acknowledgment 1](#_Toc508499047)

[What is INSTACART ? 4](#_Toc508499048)

[OBJECTIVE 5](#_Toc508499049)

[The Instacart Online Grocery Shopping Dataset 2017 6](#_Toc508499050)

[Data Dictionary 6](#_Toc508499051)

[Overall Datasets Mapping 9](#_Toc508499052)

[OUR Approach 10](#_Toc508499053)

[Exploratory Data analysis 12](#_Toc508499054)

[Total number of orders placed by users 14](#_Toc508499055)

[Overall maximum order number 14](#_Toc508499056)

[No. of Order by week day 14](#_Toc508499057)

[No. of orders by hour of day 15](#_Toc508499058)

[Frequency of Day of week vs Hour of day 15](#_Toc508499059)

[Frequency Distribution by Days since Prior Order 16](#_Toc508499060)

[Product Sales ns hours 17](#_Toc508499061)

[Most Important Aisles 18](#_Toc508499062)

[Active / Inactive Users 19](#_Toc508499063)

[Frequency of Products ordered 20](#_Toc508499064)

[Reorder Ratio 21](#_Toc508499065)

[When do people order? 26](#_Toc508499066)

[When do they order again? 27](#_Toc508499067)

[How many prior orders are there? 27](#_Toc508499068)

[How many items do people buy? 27](#_Toc508499069)

[Bestsellers 28](#_Toc508499070)

[How often do people order the same items again? 29](#_Toc508499071)

[Most often reordered 29](#_Toc508499072)

[Which item do people put into the cart first? 30](#_Toc508499073)

[Association between time of last order and probability of reorder 30](#_Toc508499074)

[Association between number of orders and probability of reordering 31](#_Toc508499075)

[Organic vs Non-Organic Products 31](#_Toc508499076)

[Organic vs Non-Organic Orders 31](#_Toc508499077)

[Reordering Organic vs Non-Organic 32](#_Toc508499078)

[Visualizing the Product Portfolio 32](#_Toc508499079)

[How often are products from the department/aisle sold? 33](#_Toc508499080)

[Market Basket Analysis 35](#_Toc508499081)

[Introduction 35](#_Toc508499082)

[Methodology 35](#_Toc508499083)

[Step 1 36](#_Toc508499084)

[Step 2 37](#_Toc508499085)

[Step 3 38](#_Toc508499086)

[Step 4 40](#_Toc508499087)

[Step 5 42](#_Toc508499088)

[Data Modelling 45](#_Toc508499089)

[Model Validation (Mean F1 Score) 46](#_Toc508499090)

[RESULTS 47](#_Toc508499091)

[CONCLUSION 48](#_Toc508499092)

**Outline**

Instacart Company Profile

History of Instacart

Problem Statement

Data Exploration

Data Engineering

Data Modelling

Model Validation (Mean F1 Score)

### What is INSTACART ?

Instacart is an American company that operates as a same-day grocery delivery service. Customers select groceries through a web application from various retailers and delivered by a personal shopper. Up until the end of 2017, Instacart only has operations and services in the United States.

How they Service : Instacart's service is mainly provided through a smartphone app, available on iOS and Android platforms, apart from its website. Customers can pay with Android Pay and Apple Pay on their respective platforms. Initially Instacart shoppers simply went to a store and purchased the ordered items at retail and, in addition to the delivery charge, added a mark-up of 10 to 20 percent. As the business has developed, the firm has established relationships with grocery firms which share their (store) existing mark-up, allowing Instacart users to shop at in-store prices

Instacart’s data science team plays a big part in providing this delightful shopping experience. Currently they use transactional data to develop models that predict which products a user will buy again, try for the first time, or add to their cart next during a session.

Recently, Instacart open sourced this data - see their blog post on 3 Million Instacart Orders, Open Sourced. “The Instacart Online Grocery Shopping Dataset 2017”. This anonymized dataset contains a sample of over 3 million grocery orders from more than 200,000 Instacart users.

### OBJECTIVE

* To predict which products will be in a user's next order
* To predict which Users will try it for the first time.
* To predict which users will add particular products to cart next during a session.

The goal of this project is to predict, for each individual, which products that they have purchased before they will purchase again in their final order. This is a challenging multiclass multi label prediction problem where we are asked to predict a basket of about 50,000 potential items for being reordered.

Predictions are submitted as an order ID and a list of product IDs predicted for that order. Predictions can include none, a special product ID indicating that no previously ordered products have been ordered. None can be combined with other product IDs in prediction, but will of course only occur on its own in the ground truth.

Predictions are evaluated using the average F1-score across orders.

We derived relevant features and experimented with different modelling approaches.

Name the modelling approaches

### The Instacart Online Grocery Shopping Dataset 2017

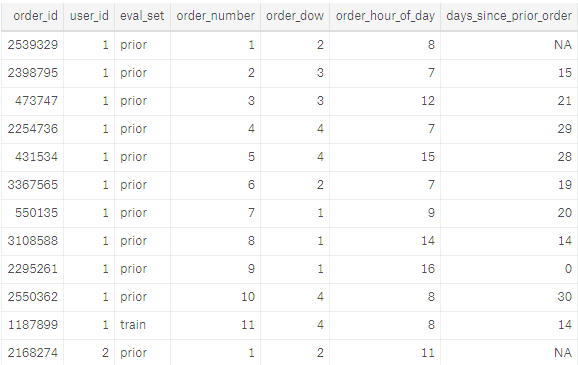
The dataset is anonymized and contains a sample of over 3 million grocery orders from more than 200,000 Insta-cart users. The only information provided about users is their sequence of orders and the products in those orders. All of the IDs in the dataset are entirely randomized, and cannot be linked back to any other ID. Only products that are bought by multiple people at multiple retailers are included, and no retailer ID is provided

### Data Dictionary

#### orders.csv

|  |  |
| --- | --- |
| **orders (3.4m rows, 206k users)** | |
| order\_id | order identifier |
| user\_id | customer identifier |
| eval\_set | which evaluation set this order belongs in (see SET described below) |
| order\_number | the order sequence number for this user (1 = first, n = nth) |
| order\_dow | the day of the week the order was placed on |
| order\_hour\_of\_day | the hour of the day the order was placed on |
| days\_since\_prior | days since the last order, capped at 30 (with NAs for order\_number = 1) |

This file gives a list of all orders we have in the dataset. 1 row per order. For example, we can see that user 1 has 11 orders, 1 of which is in the train set, and 10 of which are prior orders. The orders.csv doesn’t tell us about which products were ordered. This is contained in the order\_products.csv



#### products.csv

|  |  |
| --- | --- |
| **products (50k rows)** | |
| product\_id | product identifier |
| product\_name | name of the product |
| aisle\_id | foreign key |
| department\_id | foreign key |

This file contains the names of the products with their corresponding

product\_id. Furthermore the aisle and department are included.



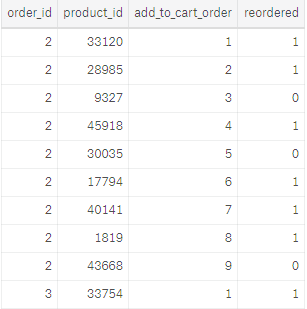
#### order\_products\_SET

|  |  |
| --- | --- |
| **order\_products\_\_SET (30m+ rows)** | |
| order\_id | foreign key |
| product\_id | foreign key |
| add\_to\_cart\_order | order in which each product was added to cart |
| reordered | 1 if this product has been ordered by this user in the past, 0 otherwise |
|  |  |
| where SET is one of the four following evaluation sets (eval\_set in orders) | |
| prior | orders prior to that users most recent order (~3.2m orders) |
| train | training data supplied to participants (~131k orders) |
| test | test data reserved for machine learning competitions (~75k orders) |

This file gives us information about which products (product\_id) were ordered. It also contains information of the order (add\_to\_cart\_order) in which the products were put into the cart and information of whether this product is a re-order(1) or not(0).

For example, we see below that order\_id 1 had 8 products, 4 of which are reorders.

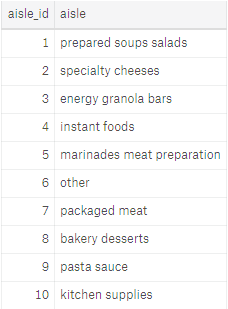
Still we don’t know what these products are. This information is in the products.csv



#### aisles.csv

|  |  |
| --- | --- |
| aisles (134 rows) | |
| aisle\_id | aisle identifier |
| aisle | the name of the aisle |

This file contains the different aisles.



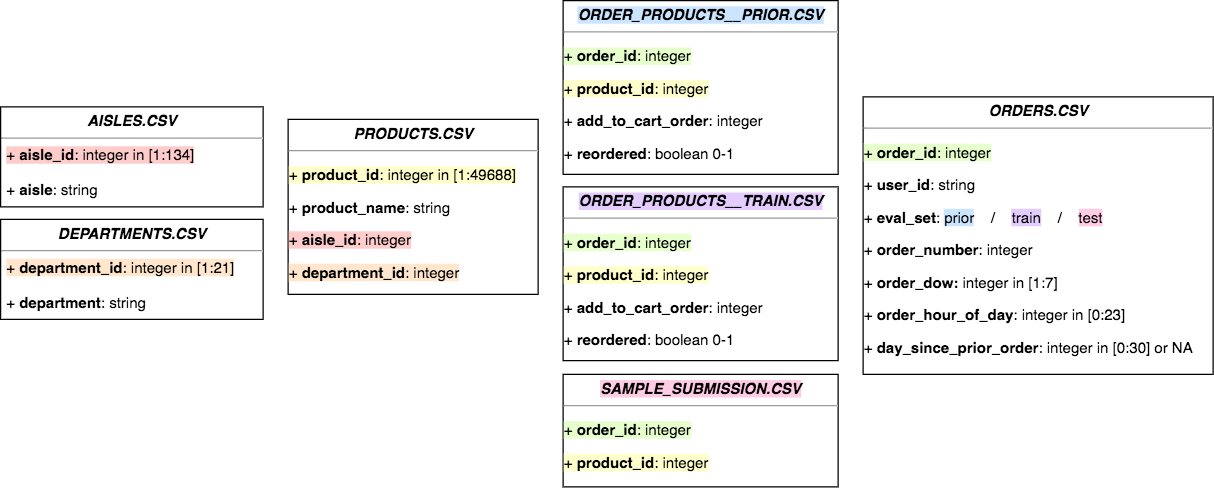
#### Departments.csv

|  |  |
| --- | --- |
| departments (21 rows) | |
| department\_id | department identifier |
| department | the name of the department |

This file contains department id along with department name.



### Overall Datasets Mapping



## OUR Approach

Feature Extraction

Based on the data provided by Instacart, we crafted mainly three categories of features about products, users and orders. Given these complex data files relationships, it involves multiple steps of data munging to efficiently create features and join them.

Table 1: Product Related Features

|  |  |
| --- | --- |
| Features | Description |
| Aisle id | Aisle category for a product |
| Department id | Department category for a product |
| Product Orders | How many times a product was ordered |
| Department wise reorder | How many time product was reordered in each department |

Table 2: User Related Features

|  |  |
| --- | --- |
| Features | Description |
| Total Orders | Total orders from a user |
| Total Items | Total products ordered by a user |
| Total distinct Items | Number of distinct products ordered by user. |
| Re-ordered Products | Products being reordered most frequently before. |
| Average basket | Average number of products in an order for a user |

Table 3: Orders related Features

|  |  |
| --- | --- |
| Features | Description |
| Day of Week | Day of week for an order |
| Order hour of day | Hour of day for an order |
| Days since prior order | Number of days since the last order for same user |
| Day of week vs hour of day | Frequency of day of week divided by hour of day |

Multi-Classification

Neural Network

Pre-processing for Modelling Framework

Before moving to modelling, we analysed the data and did pre-processing on the basis of 3 different categories which are products, orders and users that helped in framing the model more accurately.

Feature Selection

We identify the important variables that helped in creating model to be more accurate. How we did it?

Light GBM

From our experiments with various models, we realized it would be more efficient

to use gradient boosting framework that uses tree based learning algorithm. The biggest advantage of using light GBM is that it handles the large size of data and takes lower memory to run.

# Exploratory Data analysis

We started by analyzing each data set to summarize the features, characteristics and gain useful insights, often with visualizations. As mentioned in Data section above, there are six data tables provided, which are:

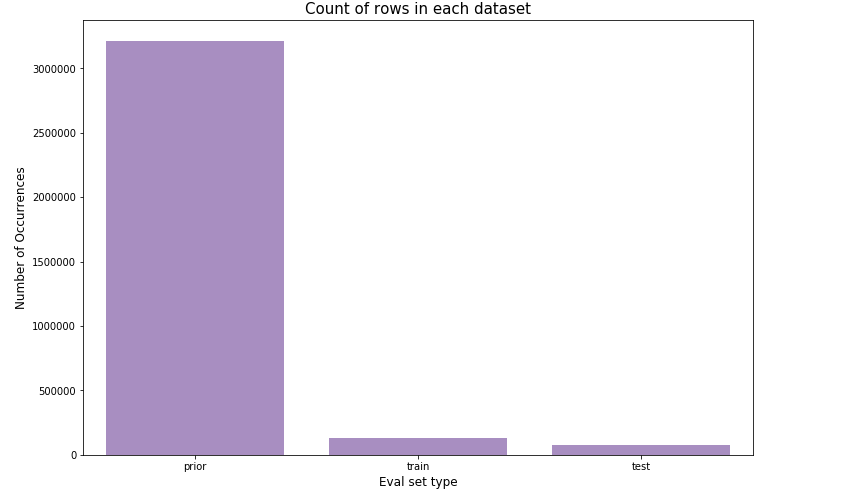
* Aisles
* Departments
* Order\_Products\_Prior
* Order\_Product\_Train
* Orders
* Products

We first analyzed the tables and columns to see which kind of data we have. There are 19 columns in total. Below screen-shot contains header of each csv file along with first 2 records.

|  |
| --- |
|  |

We have unique IDs for each order and product. In Order\_products\_prior and order\_product\_train tables, we have the same columns but the only difference is that one of them is trained data while the other one is the test data.

Below is the representation of total number of rows in each data-set:



##### Grouping of data-sets

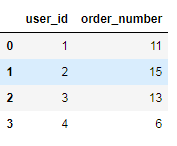
This is an important part of our exploratory analysis, wherein we put together different data-sets/features together to identify few major transactions. Such as:

* Total number of orders placed by users.
* Overall maximum order number
* Frequency of order by week day
* Percentage of products that are reordered.
* Number of products in given order

Let’s now go over these grouping of data-sets through visualization that we performed.

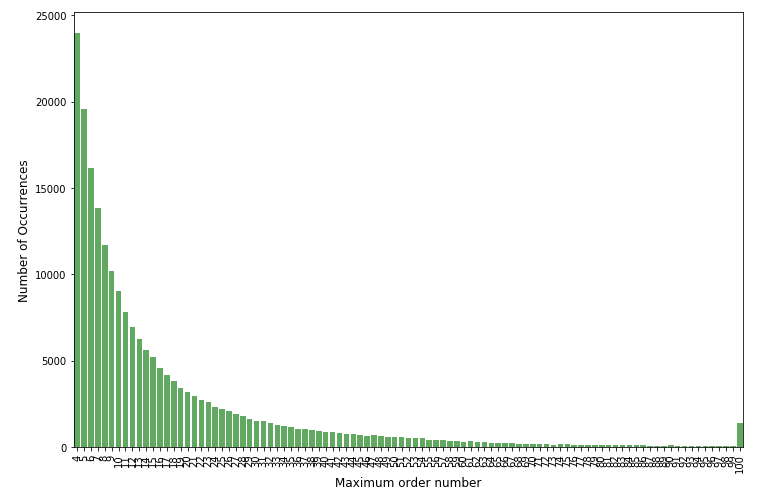
### Total number of orders placed by users

We grouped order number from order data set with user id to identify user id with maximum number of orders and displayed outcome of first four users in below screen:



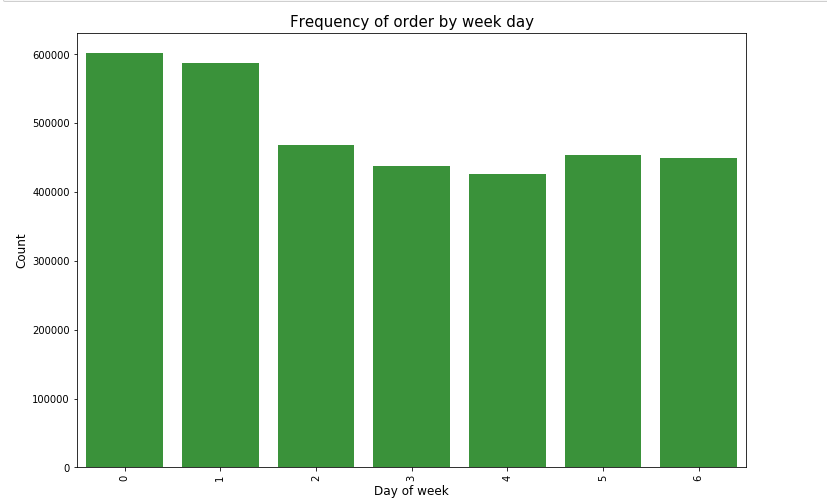
### Overall maximum order number

Here we can see the number of orders placed by all users. We concluded that maximum no of orders placed is capped at 100 and minimum is 4.

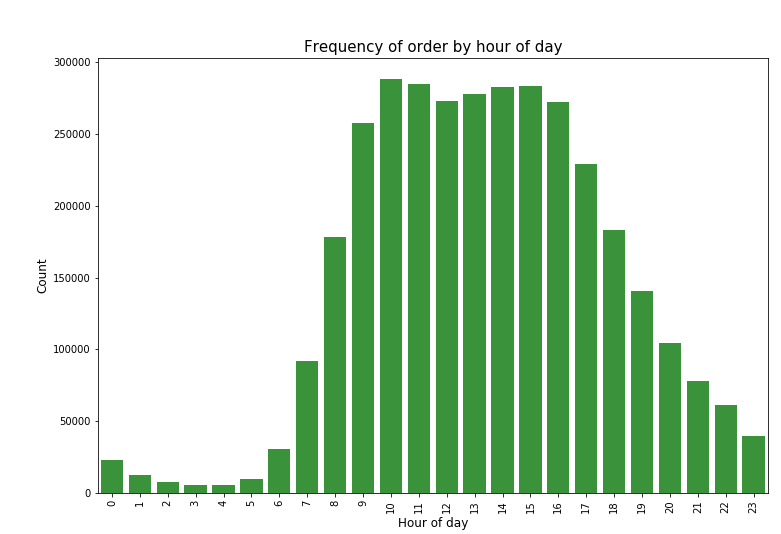


### No. of Order by week day

We are now determining on which day of week; more number of orders was placed and we figured out that maximum numbers of orders were placed during weekend.

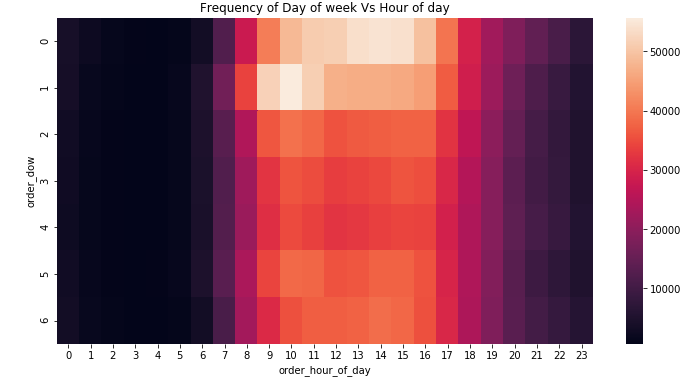


### No. of orders by hour of day

The below graph is plotted to show during which hour of day, more number of orders were placed.

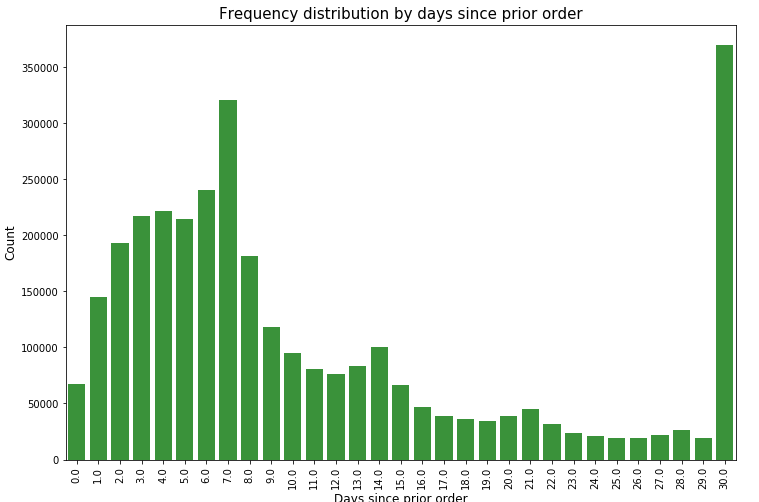
### Frequency of Day of week vs Hour of day

To analyze during which day in week and at which period, we are getting maximum orders, we combine day of week and hour of day and figured out that during weekends and at day time more orders are getting placed.



### Frequency Distribution by Days since Prior Order

We plotted frequency distribution by days since last order by counting days\_since\_prior\_order in order data set. We found that there is abrupt rise in peaks at 7th, 14th and 30th.

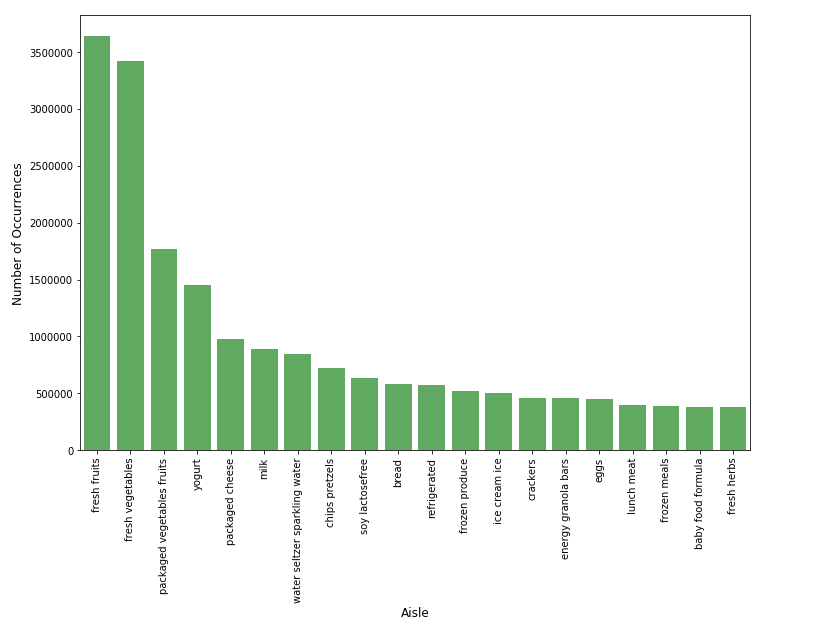


### Product Sales ns hours

|  |
| --- |
|  |

### Most Important Aisles

Below graphs shows all important aisles and number of its occurrences:



### Active / Inactive Users

We want to see how many active and inactive users we have in our dataset. There are 17915 distinct inactive users with a share of 28% and 46017 distinct active users with a share of 72%, so we can set our focus mainly on this 18% share of users because their orders will probably provide us more certain information to make estimations and they are the ones who will likely do shopping regularly and continuously.

|  |
| --- |
|  |

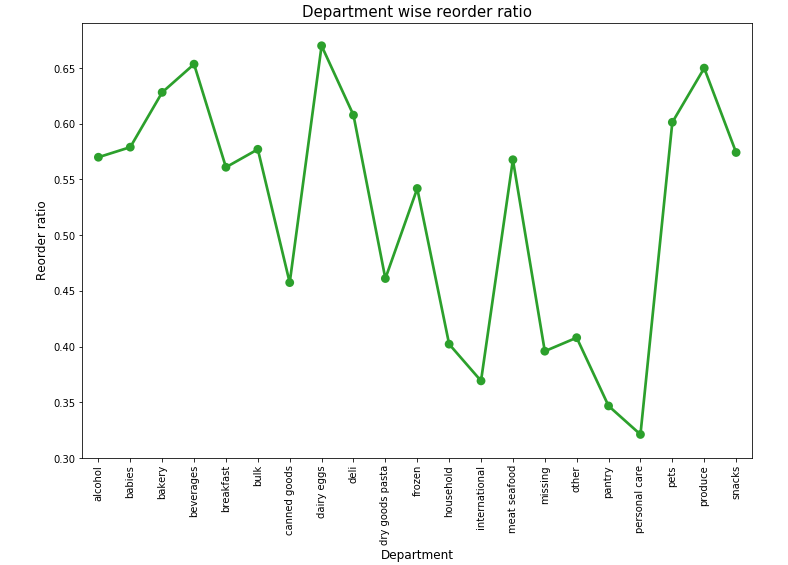
### Frequency of Products ordered



### Reorder Ratio

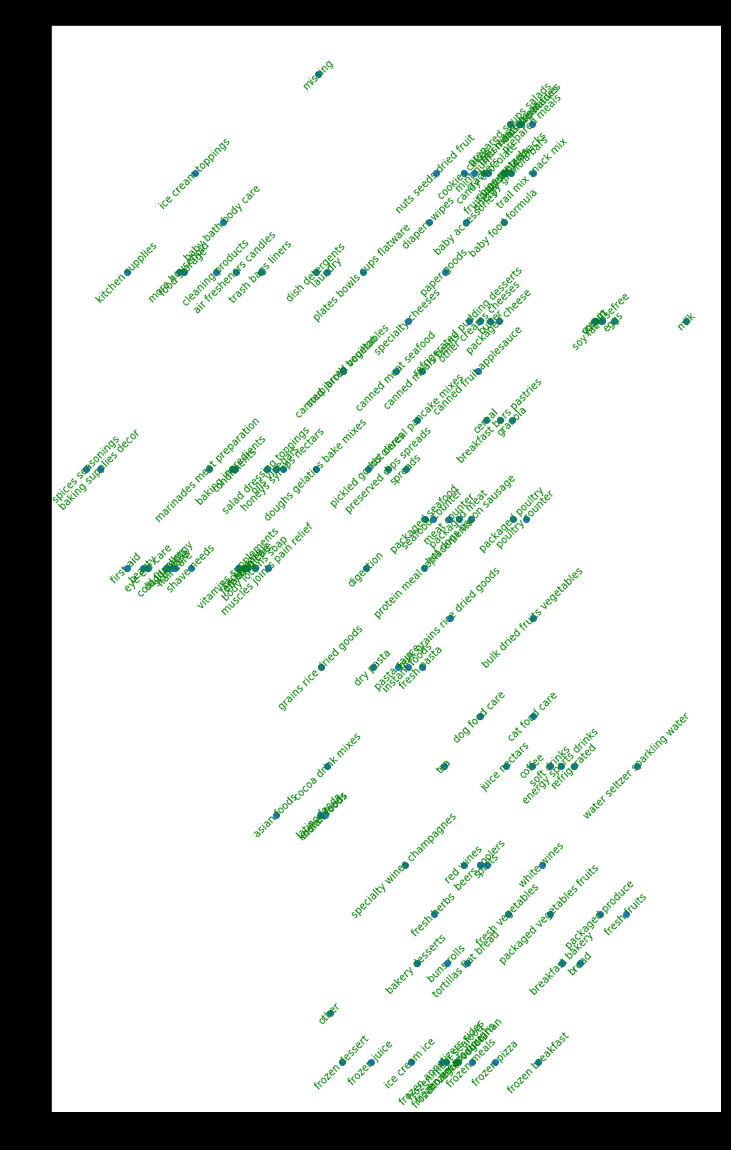
#### Department wise reorder Ratio

We determine which department had maximum reordered item by grouping department and reordered column in order\_product\_prior data set by aggregating its mean and below results were obtained:



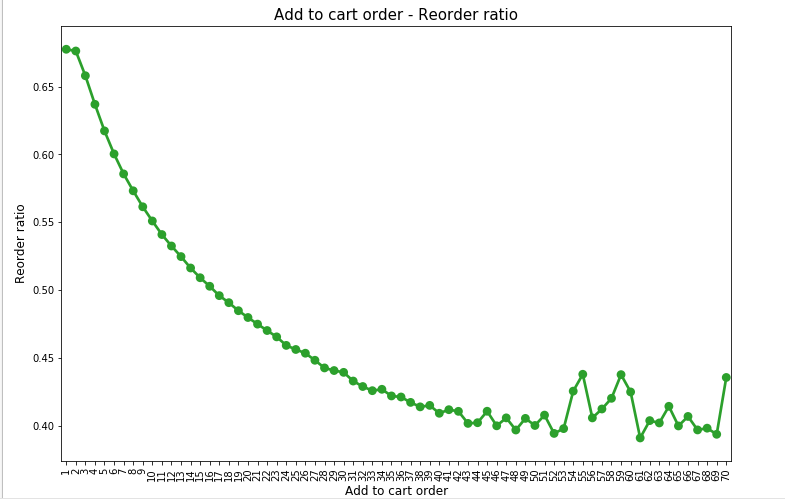
#### Reorder Ratio of Different Aisles + Department

Similarly, we determined aisle with maximum number of reordered.

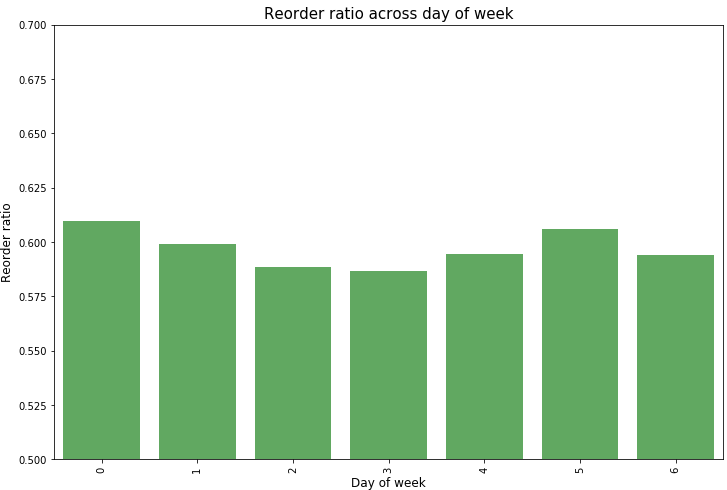


#### Add to Cart Order- Reorder Ratio

We then explored the relationship between how order of adding the product to the cart affects the reorder ratio. Looks like the products that are added to the cart initially are more likely to be reordered again compared to the ones added later.

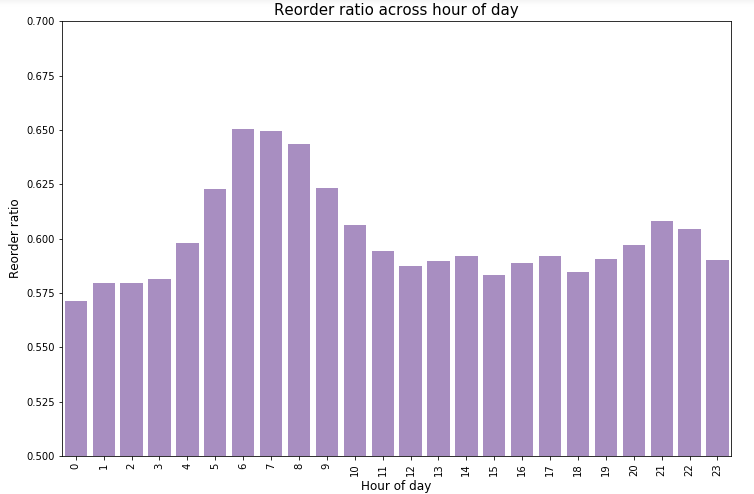


#### Reorder Ratio across day of week

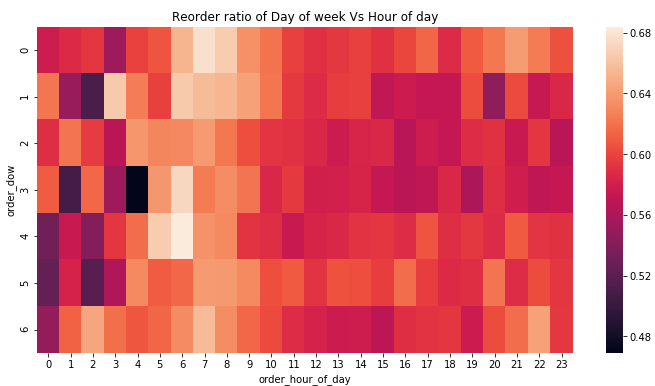
To evaluate reorder ratio across week, we merge train dataset wit order data set and grouped order\_dow and reordered columns to get reorder ratio across day of week.

#### Reorder Ratio across hour of day

Similarly, we grouped order\_hour\_of\_day and reordered columns to find out reorder ratio across hour of day.



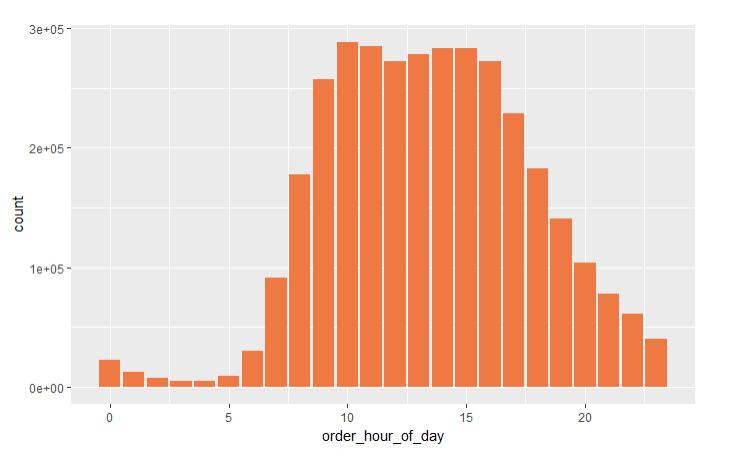
#### Reorder Ratio of Day of week vs Hour of day



### When do people order?

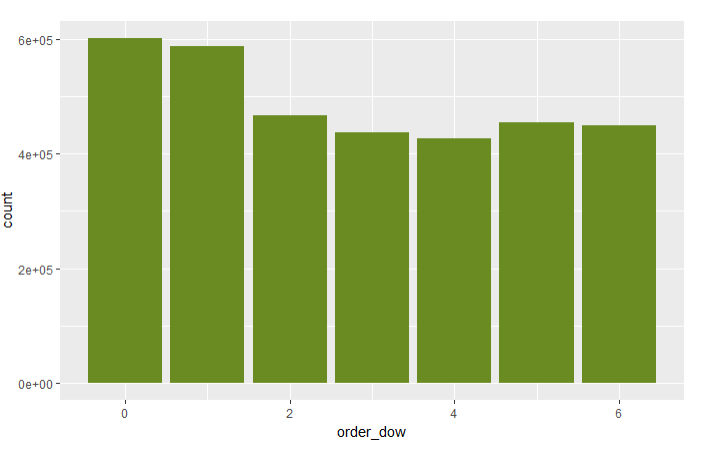
#### Hour of Day

There is a clear effect of hour of day on order volume. Most orders are between 8.00-18.00



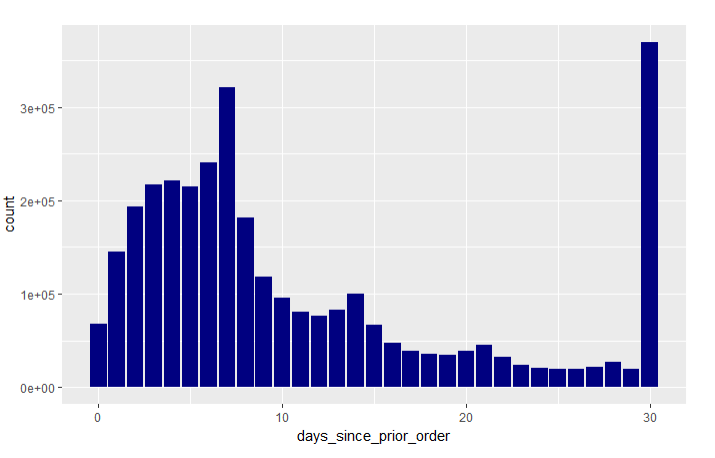
#### Day of Week

There is a clear effect of day of the week. Most orders are on days 0 and 1. Unfortunately there is no info regarding which values represent which day, but one would assume that this is the weekend.



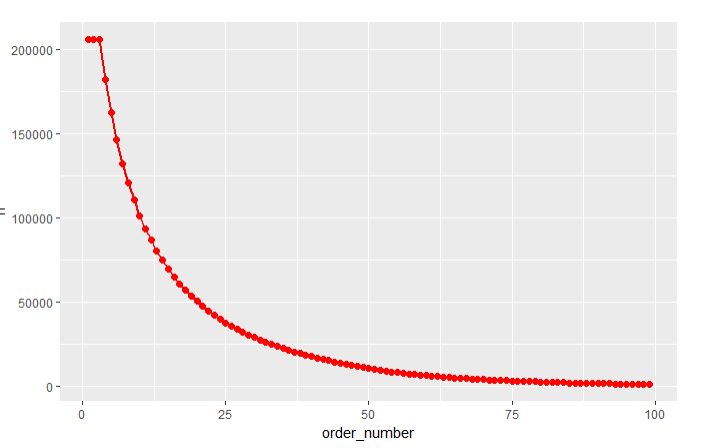
### When do they order again?

People seem to order more often after exactly 7 days, 14 days and 30 days.



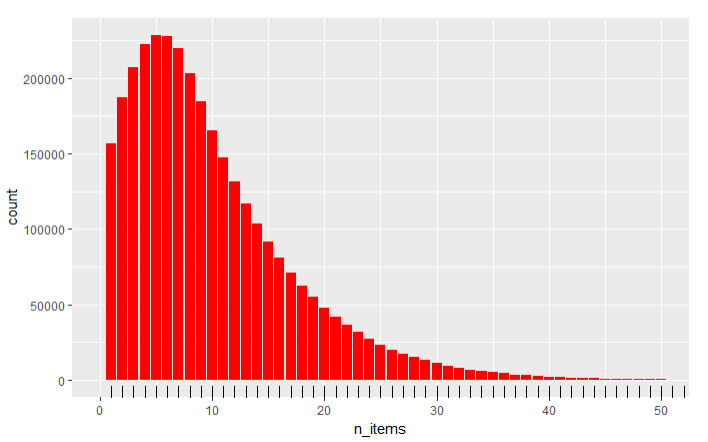
### How many prior orders are there?

We can see that there are always at least 3 prior orders.



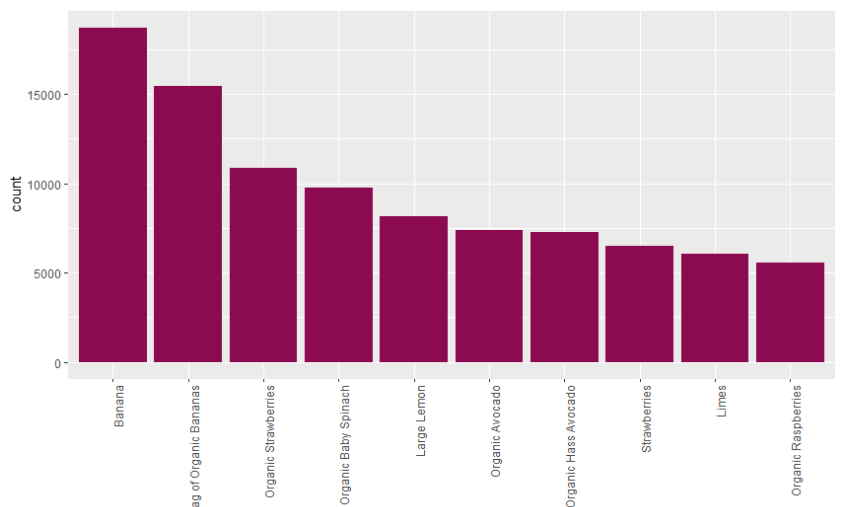
### How many items do people buy?

We can see that people most often order around 5 items. The distributions are comparable between the train and prior order set.



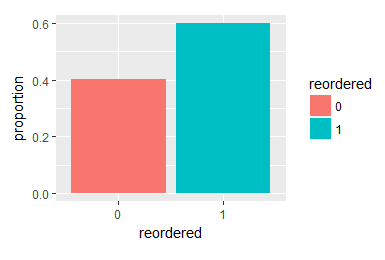
### Bestsellers

Let's have a look which product are sold most often (top10). And the clear winner is **Bananas**



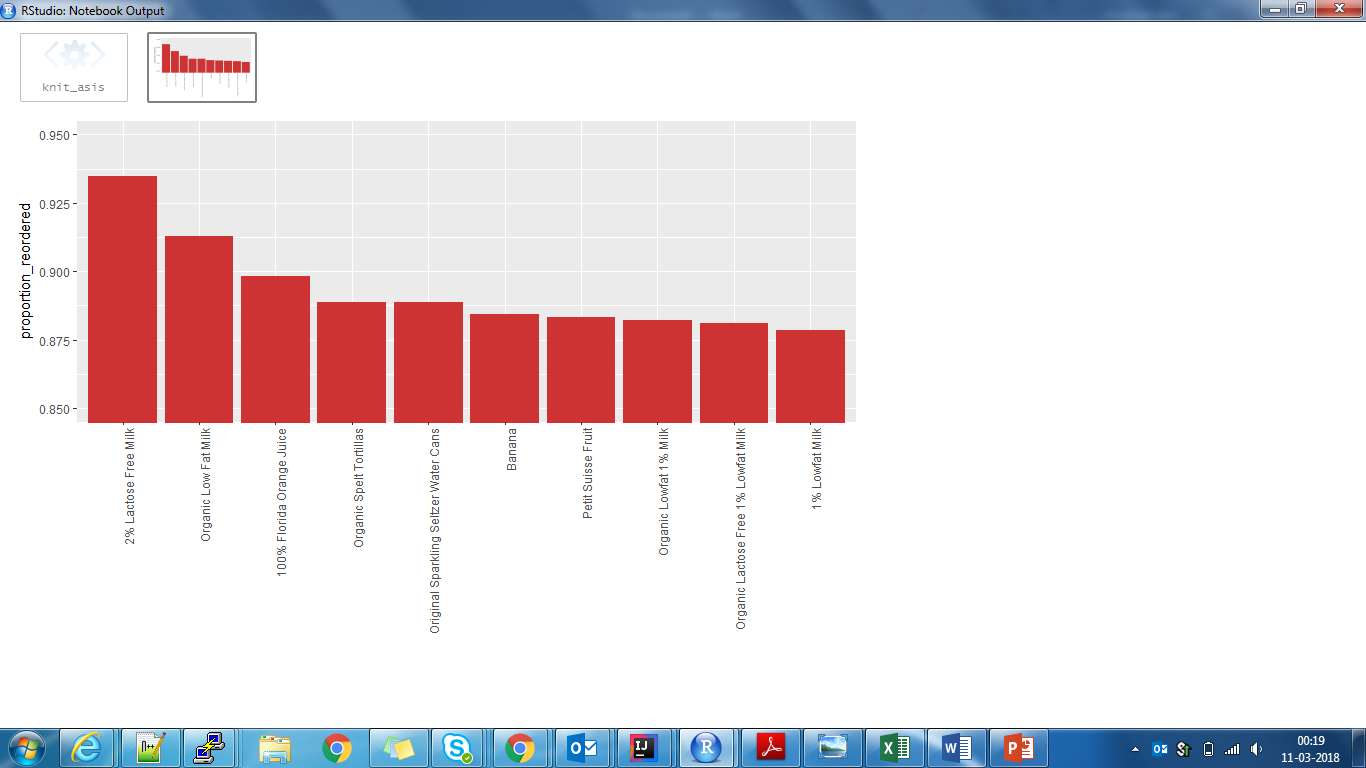
### How often do people order the same items again?

59% of the ordered items are reorders.



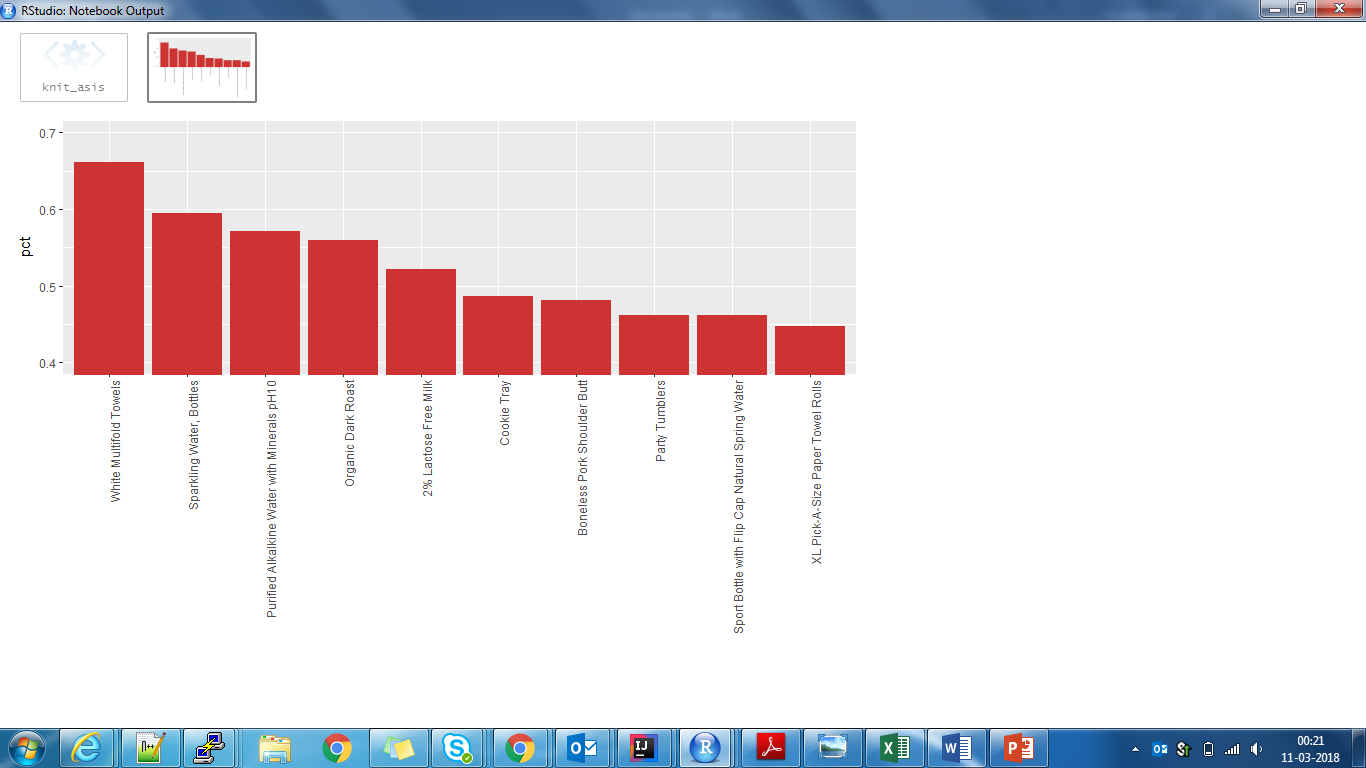
### Most often reordered

These 10 products have the highest probability of being reordered.



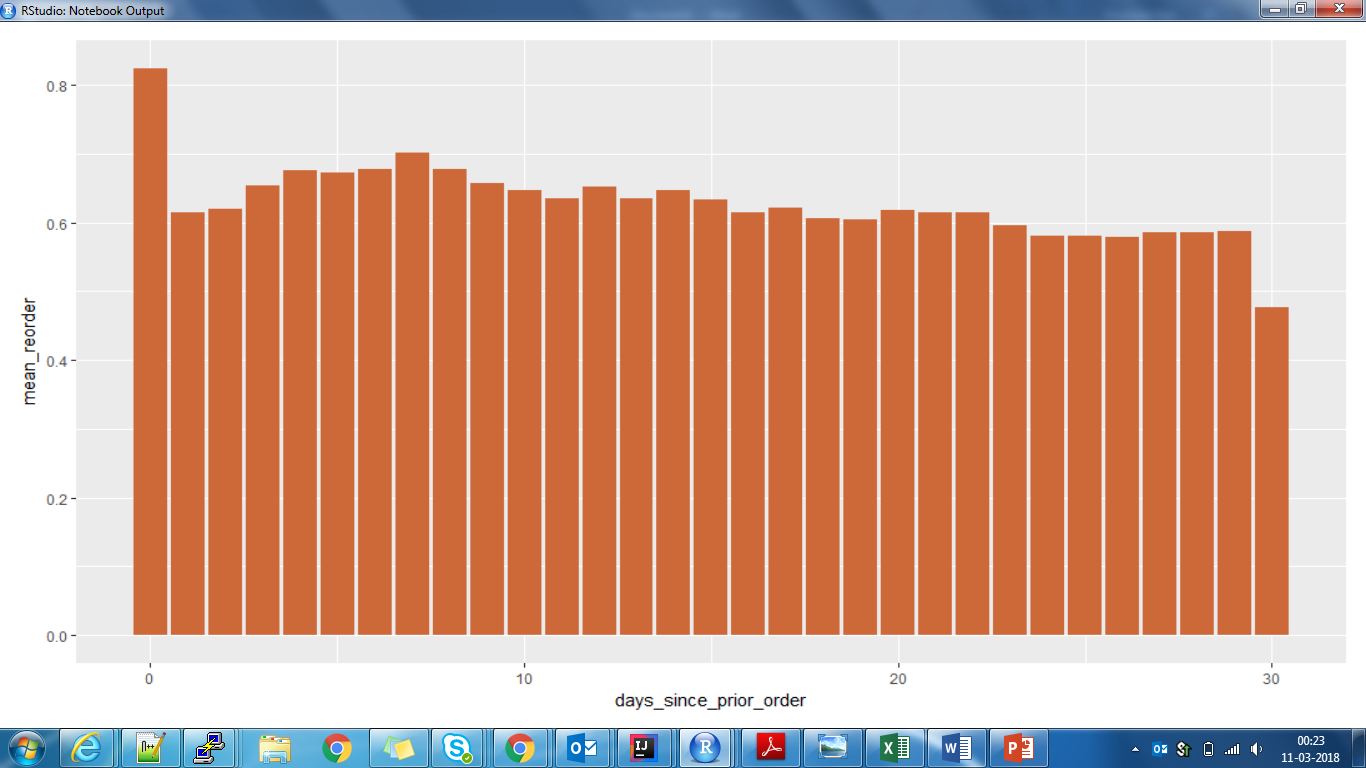
### Which item do people put into the cart first?

People seem to be quite certain about Multifold Towels and if they buy them, put them into their cart first in 66% of the time.



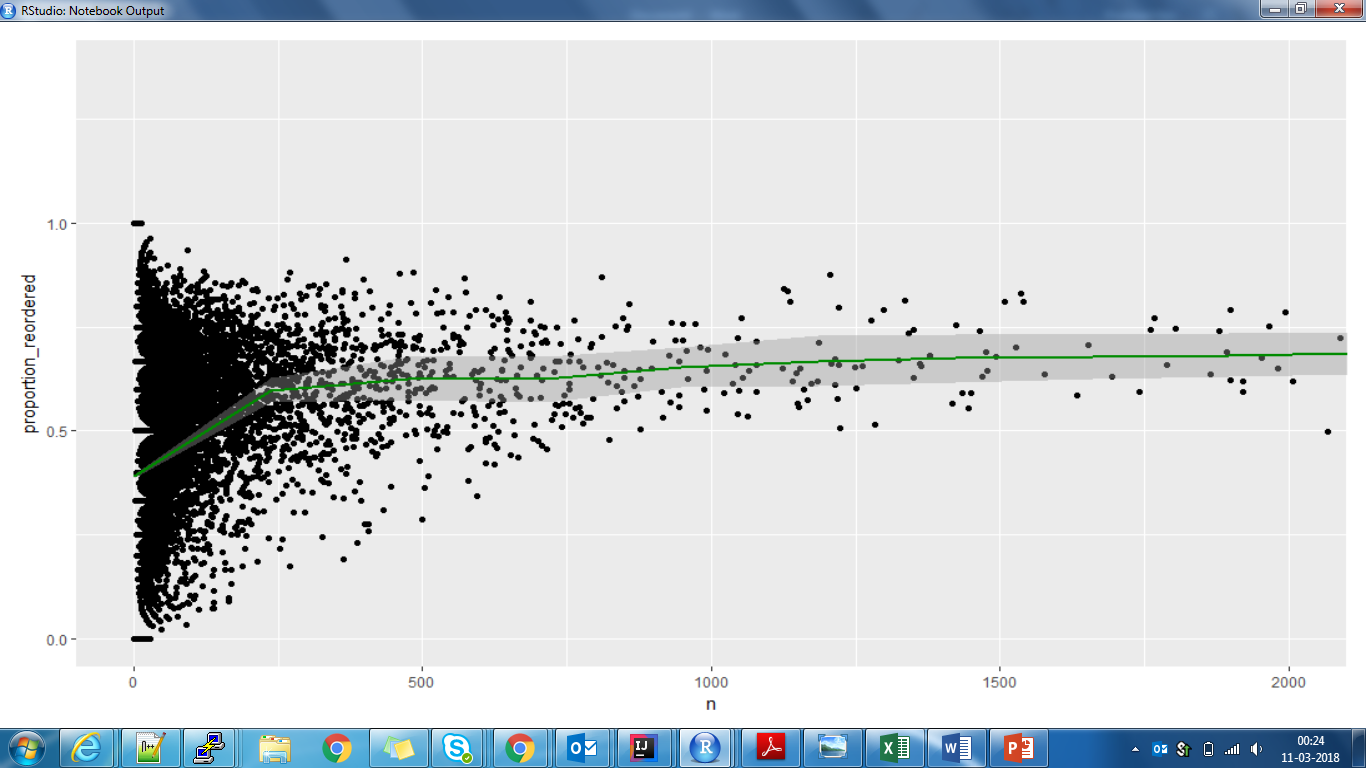
### Association between time of last order and probability of reorder

We can see that if people order again on the same day, they order the same product more often. Whereas when 30 days have passed, they tend to try out new things in their order.



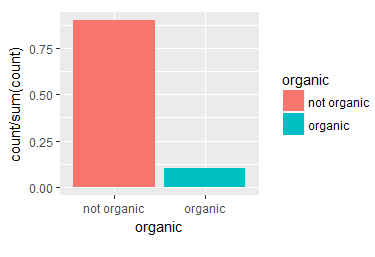
### Association between number of orders and probability of reordering

Products with a high number of orders are naturally more likely to be reordered. However, there seems to be a ceiling effect.



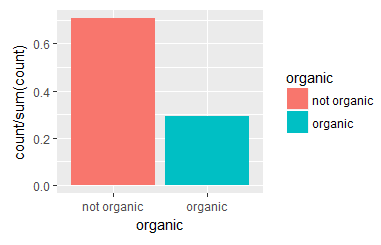
### Organic vs Non-Organic Products

What is the percentage of products that are organic vs. not organic?



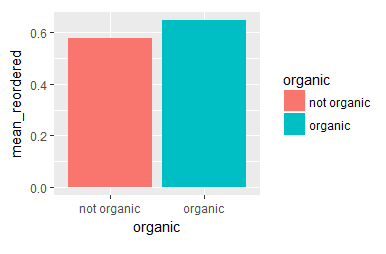
### Organic vs Non-Organic Orders

What is the percentage of orders that are organic vs. not organic?



### Reordering Organic vs Non-Organic

People more often reorder organic products vs non-organic products.



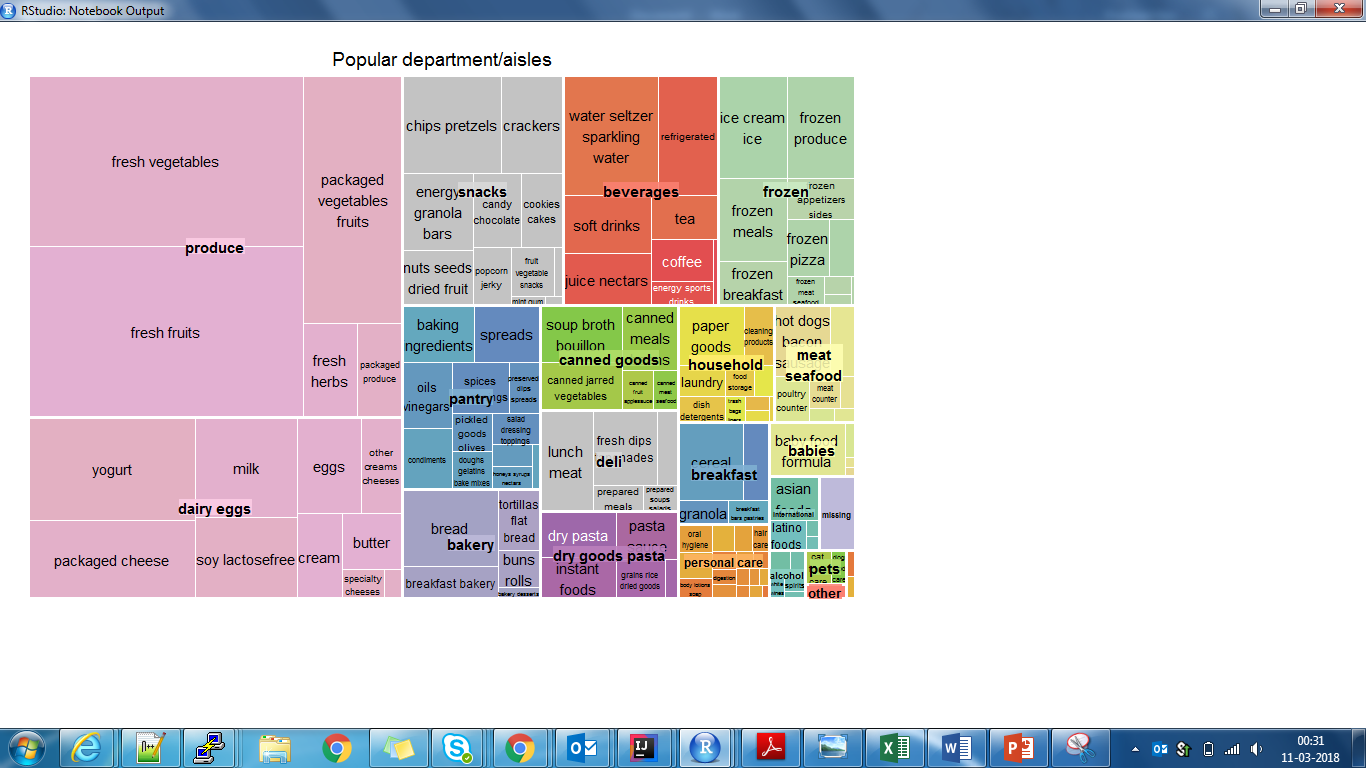
### Visualizing the Product Portfolio

We used treemap package to visualize the structure of product portfolio. In total there are 21 departments containing 134 aisles.



### How often are products from the department/aisle sold?

The size of the boxes shows the number of sales.



# Market Basket Analysis

## Introduction

Mining frequent itemsets and association rules is a popular and well researched approach for discovering interesting relationships between variables in large databases.

Understanding buying patterns can help to increase sales in several ways. If there is a pair of items, X and Y, that are frequently bought together.

* Both X and Y can be placed on the same shelf, so that buyers of one item would be prompted to buy the other.
* Promotional discounts could be applied to just one out of the two items.
* Advertisements on X could be targeted at buyers who purchase Y.
* X and Y could be combined into a new product, such as having Y in flavors of X.

Association rules analysis is a technique to uncover how items are associated to each other. There are three common ways to measure association.

**Support**: This says how popular an itemset is, as measured by the proportion of transactions in which an itemset appears. If it is discovered that sales of items beyond a certain proportion tend to have a significant impact on profits, one might consider using that proportion as the support threshold.

**Confidence**: This says how likely item Y is purchased when item X is purchased, expressed as {X -> Y}. This is measured by the proportion of transactions with item X, in which item Y also appears

**Lift**: This says how likely item Y is purchased when item X is purchased, while controlling for how popular item Y is. A lift value greater than 1 means that item Y is likely to be bought if item X is bought, while a value less than 1 means that item Y is unlikely to be bought if item X is bought.

## Methodology

We performed market basket analysis on Instacart data using R. We used the popular library arules; which is a computational environment for mining association rules and frequent item sets.

Market Basket Analysis allows us to identify items that are frequently bought together. Typically the output of an MBA is in the form of rules. The rules can be simple {A ==> B}, when a customer buys item A then it is (very) likely that the customer buys item B. More complex rules are also possible {A, B ==> D, F}, when a customer buys items A and B then it is likely that he buys items D and F.

Step 1: Find frequent items in the shopping baskets. The support is set to be 0.02

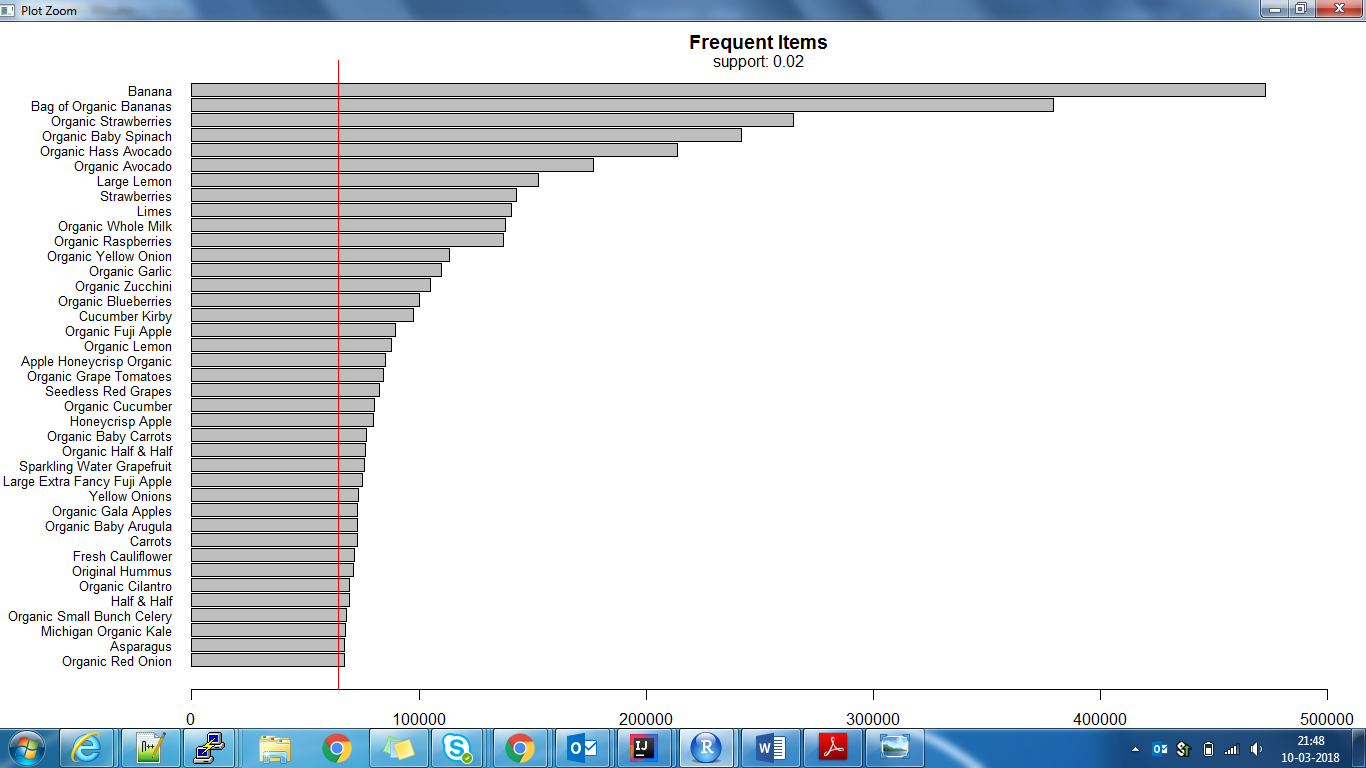


Figure 1: Frequent Items: Support=0.02

Banana is the most favorite item followed by Strawberries! Clearly vegetables and fruits are the most ordered products.

Step 2: Now we apply apriori algorithm to compute the frequent itemsets. We decrease the support threshold to take into account the small probability of observing a frequent itemset of at least size 2.

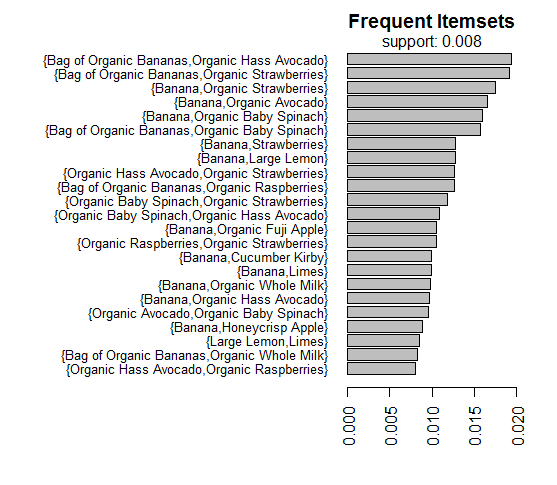
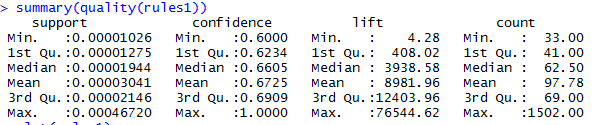


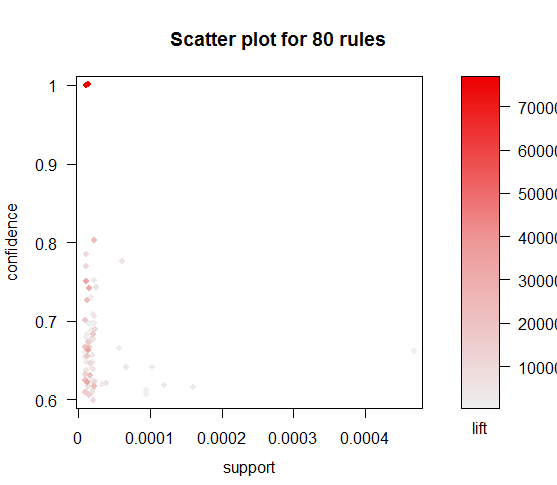
Figure 2: Frequent Itemsets; Support=0.008

Bananas being the most favorite item, rules the show here too! It is part of almost every frequent item set.

Step 3: Lets mine some association rules. First, we use a low support threshold and a high confidence to generate strong rules even for items that are less frequent.

**support = 0.00001, confidence = 0.6**





There are some rules with a large value of lift indicating a strong association between the items. Let's further investigate those critical rules.



Figure 3; Top 10 Rules by Lift



Figure 4: Top 10 rules by Confidence

Step 4: Next, we increase the support and decrease confidence to get rules of some more frequent items but with less confidence.

**support = 0.001, confidence = 0.4**

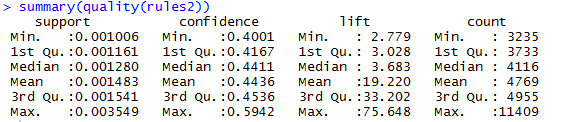




Figure 5: Top 10 rules by Lift

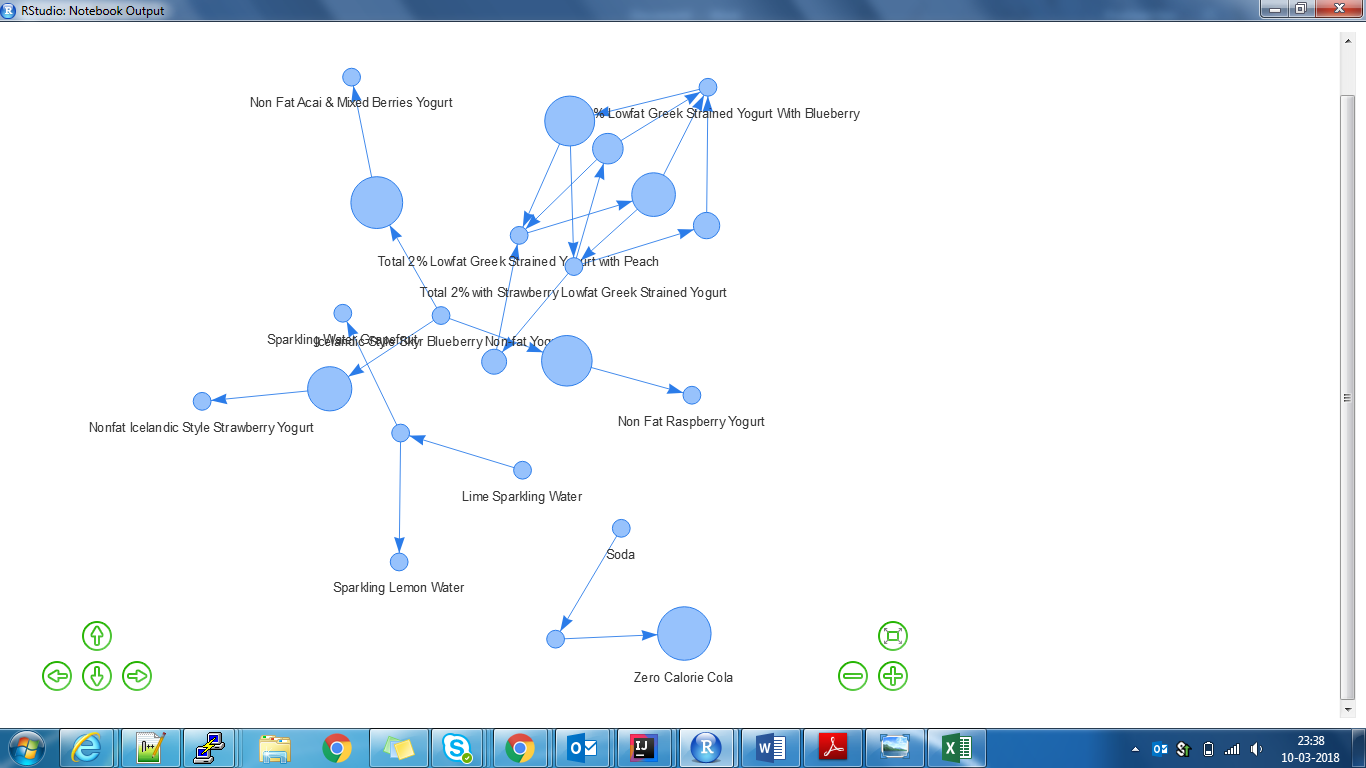
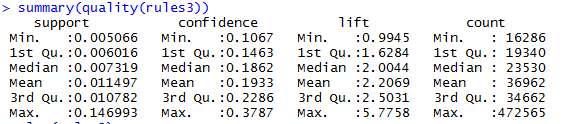


Figure 6: Network visualization of rules

Step 5: Finally, lets further increase support and decrease confidence.

**support = 0.005, confidence = 0.1**



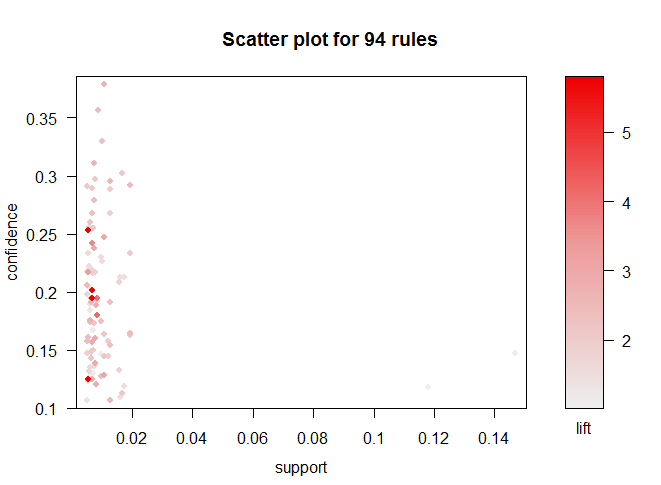




Figure 7: Top 10 rules by Lift

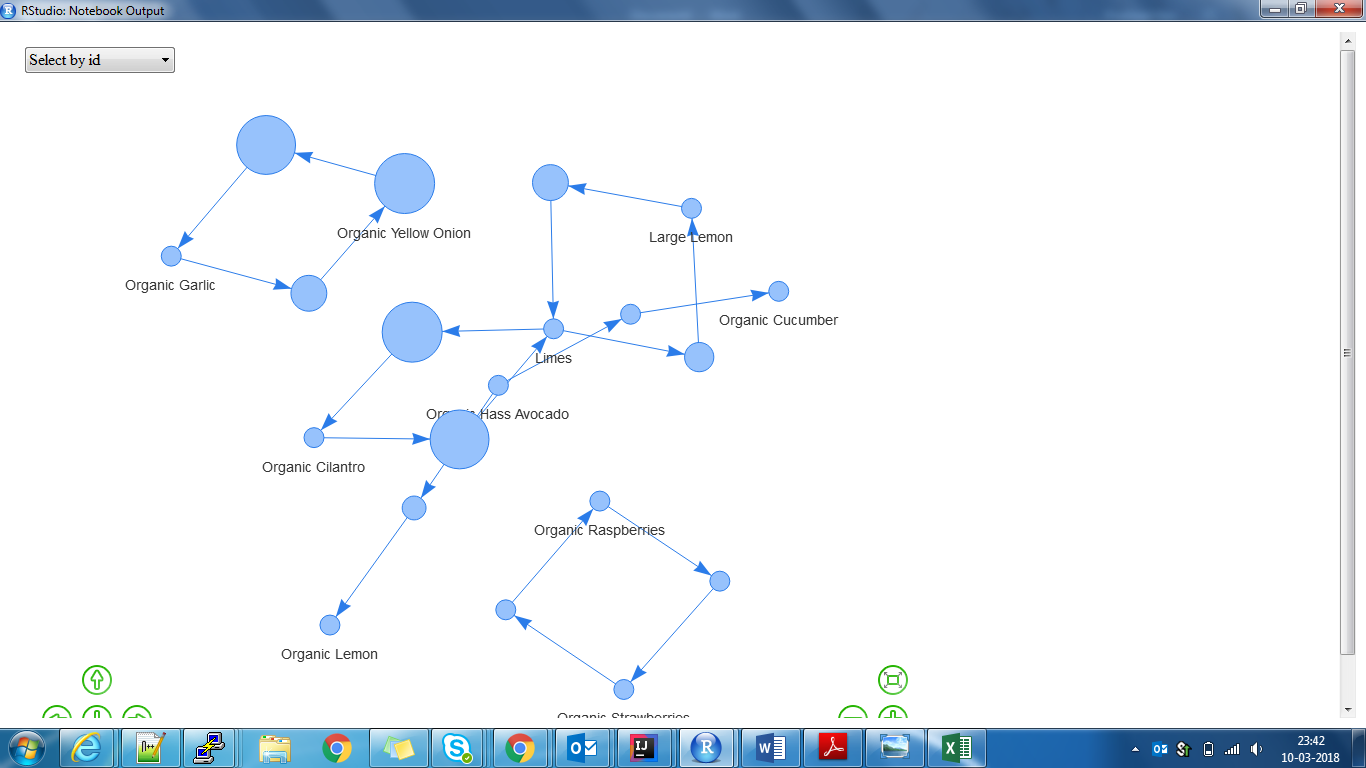




Figure 8: Top 10 rules by Confidence

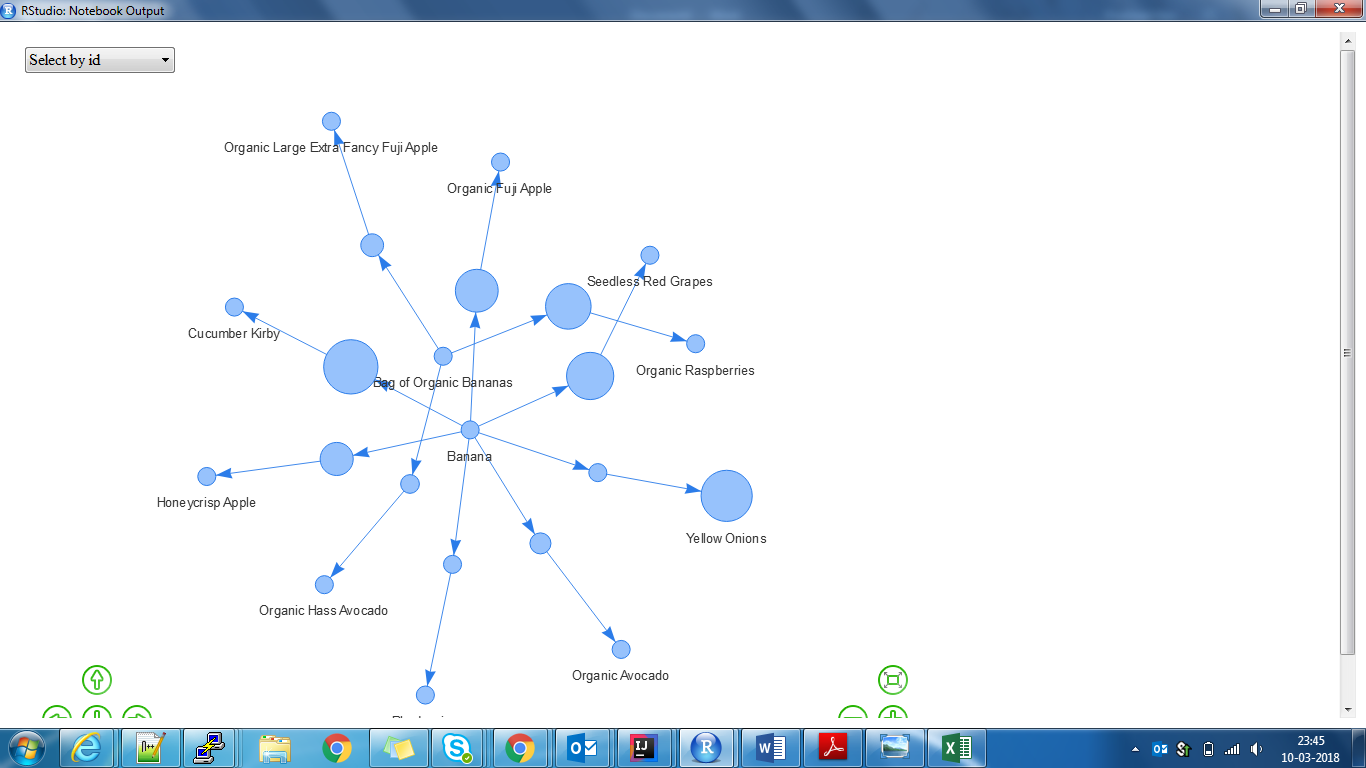


Figure 9: Network Visualization of top 10 rules by confidence

# Data Modelling

# Model Validation (Mean F1 Score)

# RESULTS