We care about your needs!

How People Order Online? Analysis of 3 million Instacart orders.

BDA-503 - Term Project - Berk Orbay

As group **cleveR**, we will work on the online order behaviors of Instacart customers witH group members:

- Ahmet Özmen
- Devrim Nesipoglu
- Numan Çagatay Atmaca
- Recep Durdu

#About the Company Instacart is a San Francisco based online shopping website that provide service for online grocery shopping orders and same-day delivery service from various local stores. Instacart has founded in San Francisco on June 2012 and operating in 39 states across United States of America today.

Our Purpose

In this project, our work will be composed of 3 main areas.

1.Exploratory Analysis on the data
2.Segmentation
3.Recommendation

Our study will begin with exploratory data analysis. We will try to explain customer behaviours and pruchasing patterns.We will try to reach facts like "what time of the day customers order most", "What is avarege basket size?", "Which products ordered together" etc.

After exploratory analysis, we will try to group customers according to their purchasing behaviours. We will exemine their orders according to product category, order time, order amount in order to find similarities and differences.

At the end, we will try to find what a customer most likely to buy in his/her next order and try to suggest them the products.

About the Data

Instacart Market Basket web site has a recommendation feature, suggestion the users some items that he / she may buy again. Its dataset is provided as for non-commercial use. Our task is how consumers discover and purchase groceries by predicting which item will be reordered on the next order.

The data provided should give a better understanding of both the characteristics of its products and the relationship with the orders. 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 randomized and cannot be linked back to any other ID. It includes orders from many different retailers and is a heavily biased subset of Instacart's production data. No retailer ID is provided.

The dataset consists of information about 3.4 million grocery orders which is distributed to six csv files; "orders", "products", "order_products", "order_products_prior", "aisles", "departments". A definition of each variable and an explanation are given in the following sections.

You can reach dataset from instacard. Lets look at the row numbers and variables in each file:

Products

This file contains the names of the products with their corresponding product_id. Furthermore the aisle and department are included.

- product_id: product identifier
- product_name: name of the product
- aisle_id: foreign key
- department_id: foreign key

```r
products<-read.csv("data/products.csv")
kable(head(products,5),align="l")
```

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich Cookies	61	19
2	All-Seasons Salt	104	13
3	Robust Golden Unsweetened Oolong Tea	94	7
4	Smart Ones Classic Favorites Mini Rigatoni With Vodka Cream Sauce	38	1
5	Green Chile Anytime Sauce	5	13

Aisles

This file contains the names of the aisles with their aisle_id and name of the aisle.

- aisle_id: aisle identifier
- aisle: the name of the aisle

```r
aisles<-read.csv("data/aisles.csv")
kable(head(aisles,5),align="l")
```

aisle_id	aisle
1	prepared soups salads
2	specialty cheeses
3	energy granola bars
4	instant foods
5	marinades meat preparation

Departments

This file contains the names of the departments with their department_id and name of the department.

- department_id: department identifier
- department: the name of the department

```r
departments<-read.csv("data/departments.csv")
kable(head(departments,5),align="l")
```

department_id	department
1	frozen
2	other
3	bakery

department_id	department
4	produce
5	alcohol

Orders

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
- 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)

```r
#orders<-read.csv("data/orders.csv" ,nrows = 1000000)
orders<-read.csv("data/orders.csv")
kable(head(orders,5),align="l")
```

order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
2539329	1	prior	1	2	8	NA
2398795	1	prior	2	3	7	15
473747	1	prior	3	3	12	21
2254736	1	prior	4	4	7	29
431534	1	prior	5	4	15	28

Order_Products_Train

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).

- 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

```r
#op_train<-read.csv("data/order_products__train.csv",nrows = 1000000)
op_train<-read.csv("data/order_products__train.csv")
kable(head(op_train,5),align="l")
```

order_id	product_id	add_to_cart_order	reordered
1	49302	1	1
1	11109	2	1
1	10246	3	0

order_id	product_id	add_to_cart_order	reordered
1	49683	4	0
1	43633	5	1

Order_Products_Prior

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).

- 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

```
#op_prior<-read.csv("data/order_products__prior.csv",nrows = 1000000)
op_prior<-read.csv("data/order_products__prior.csv")
kable(head(op_prior,5),align="l")
```

order_id	product_id	add_to_cart_order	reordered
2	33120	1	1
2	28985	2	1
2	9327	3	0
2	45918	4	1
2	30035	5	0

Data Join Process

In this step, tables that read from csv files are joined. In other words, we create a model from the tables in order to analysis data easily.

```
orders_product<-rbind(op_prior,op_train)
orders_full<-left_join(orders,orders_product,by="order_id")
orders_full<-left_join(orders_full,products,by="product_id")
orders_full<-left_join(orders_full,aisles,by="aisle_id")
orders_full<-left_join(orders_full,departments,by="department_id")
rm(orders_product, aisles,op_prior,op_train,orders,products,departments)
#orders_full <-orders_full %>% filter(row_number()<=1000000)
```

```
order_number_by_user_id<-orders_full %>%  group_by(user_id) %>% summarise(number_of_order=n_distinct(or
```

Exploratory Data Analysis

General Information

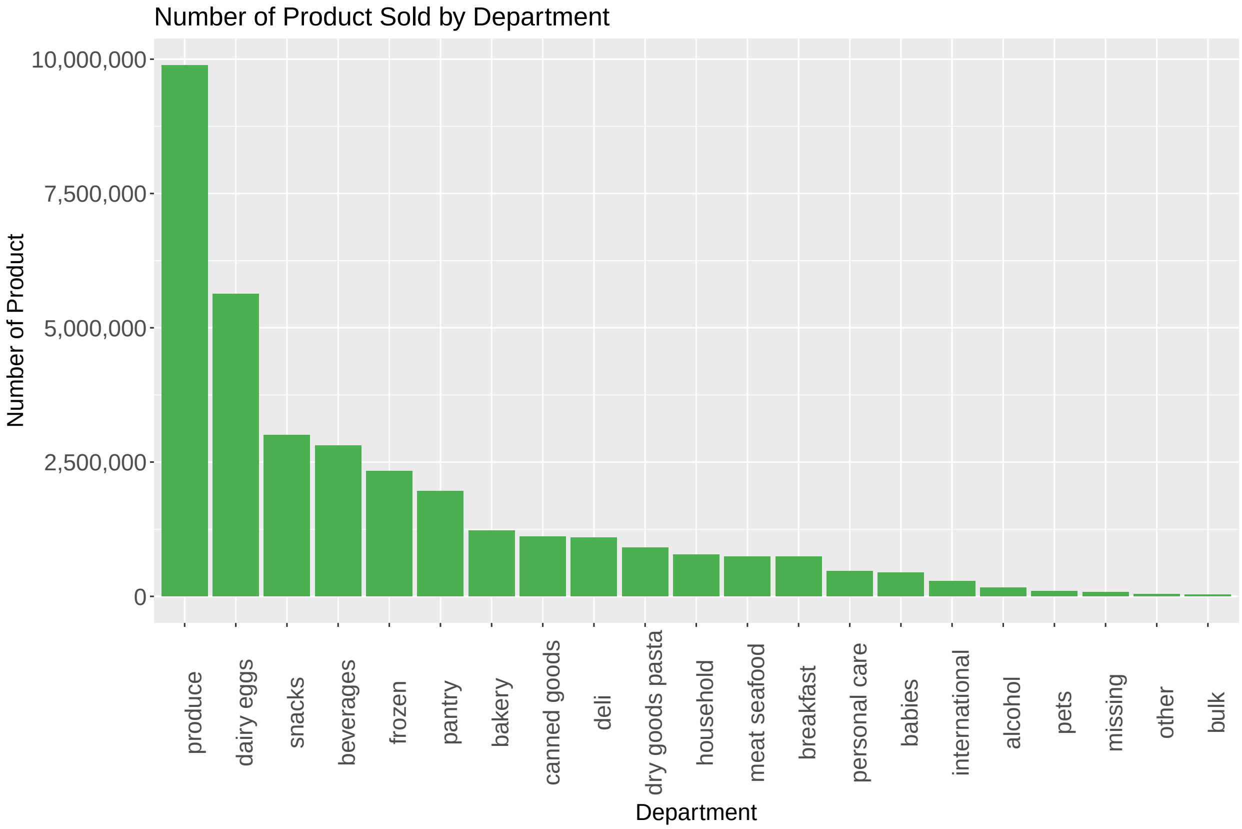
Data includes *3421083* order. There are *206209* users who placed on order. Minumum of order placed by the users is *4* . Maxiumum of order placed by the users is *100*

```
#rm(order_number_by_user_id)
temp<- orders_full %>% group_by(department) %>% summarise(count=n()) %>% mutate(per=count/sum(count) ) %
per<- temp %>% filter(row_number()<=4) %>% summarise(sum=round(sum(per),2)*100)
```

Bestseller Departmants

When we look at the number of products sold in each department, department produce and dairy eggs are obvious leaders in sales quantity. Produce department contain products such as vegetable and fruit in the form of fresh or package whereas dairy eggs contain yogurt, packaged cheese, milk and eggs, etc. Also we can see that *63 %* of the sales are come from 4 departments.

```
Product_number<-orders_full %>% filter(!is.na(department)) %>%  group_by(department) %>% summarise(Numbe
ggplot(Product_number,aes(x=reorder(department,-Number_of_Product),y=Number_of_Product)) +geom_bar(fill=
```

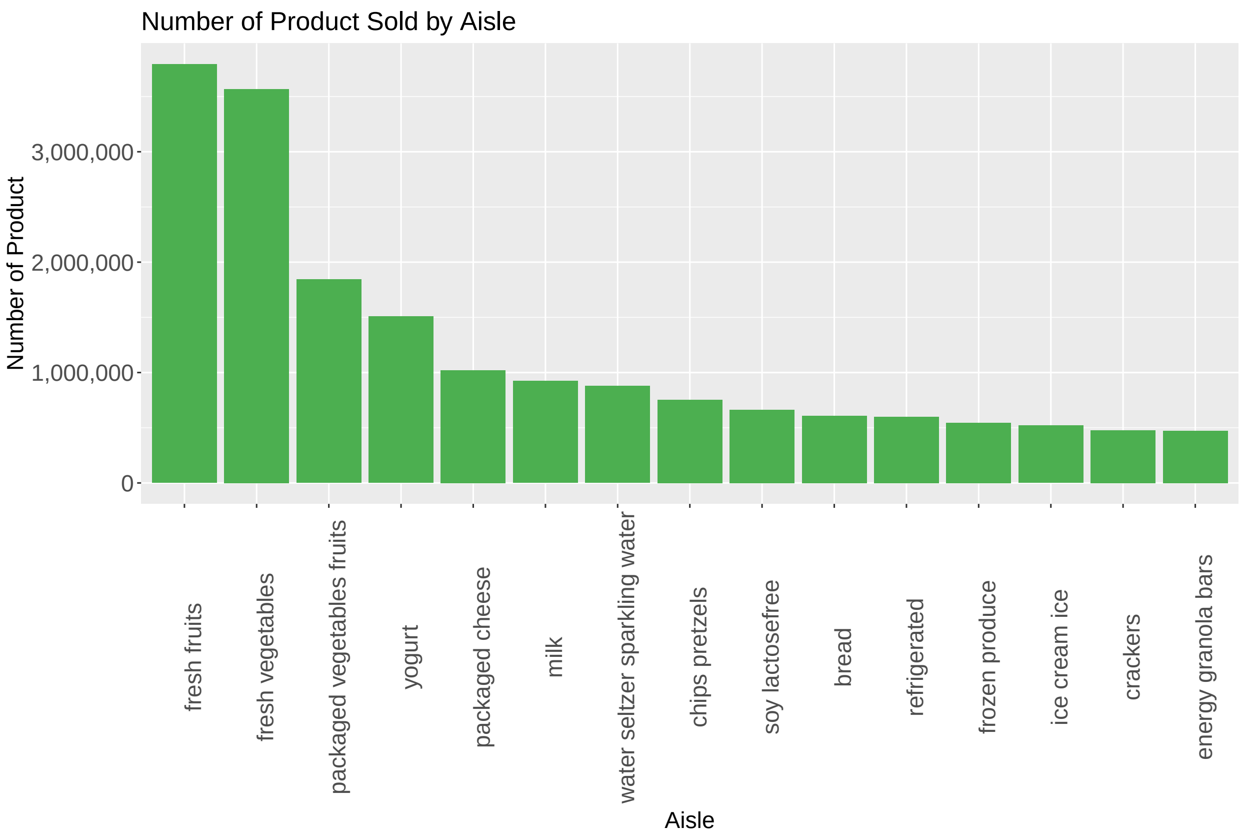


```
rm(Product_number,per,temp)
```

Bestseller Aisles

Most ordered product's aisles are looking in tandem with product's department. Fresh vegetables and fruits aisles take the lead in sales quantity by far.

```
Product_number<-orders_full %>% filter(!is.na(aisle)) %>%  group_by(aisle) %>% summarise(Number_of_Produ
ggplot(Product_number %>%  filter(row_number()<=15),aes(x=reorder(aisle,-Number_of_Product),y=Number_of_
```



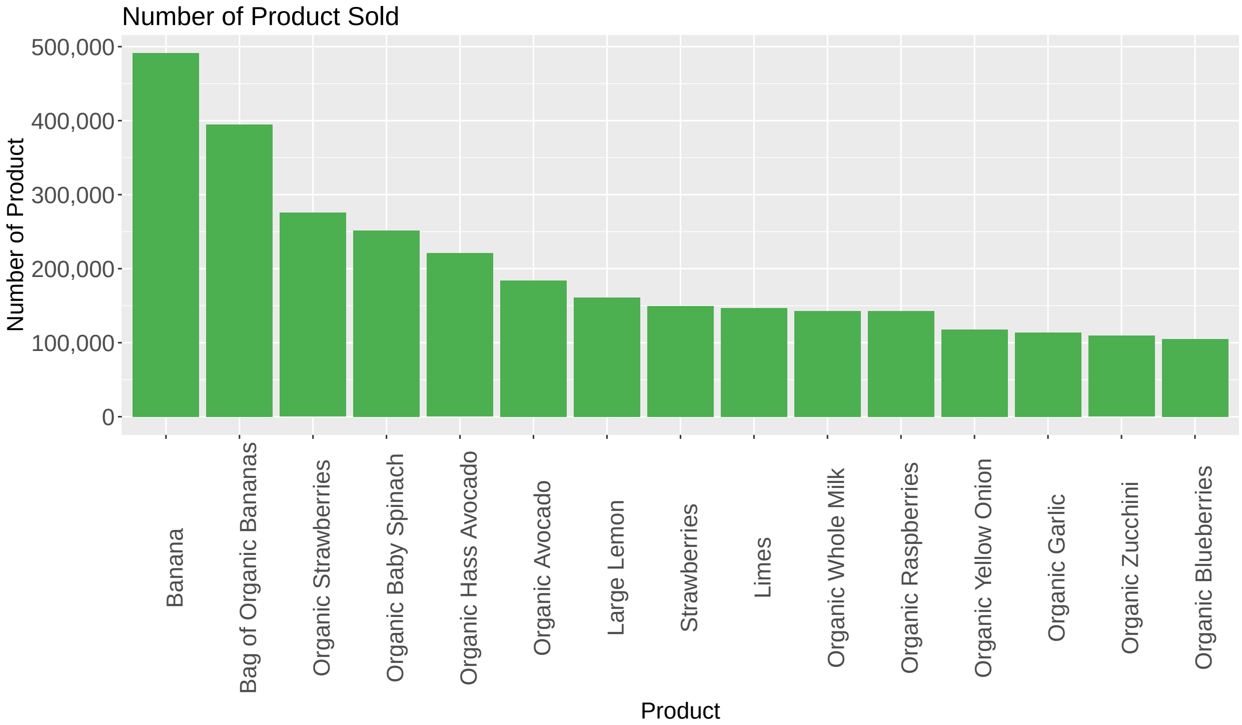
```
rm(Product_number)
```

Bestseller Products

If we put top 15 products on the cart, we can see that Bananas are the most ordered products followed by Bag of Organic Bananas and Organic Strawberries.

```
Product_number<-orders_full %>% filter(!is.na(product_name)) %>%  group_by(product_name) %>% summarise(N

ggplot(Product_number %>%  filter(row_number()<=15),aes(x=reorder(product_name,-Number_of_Product),y=Num
```



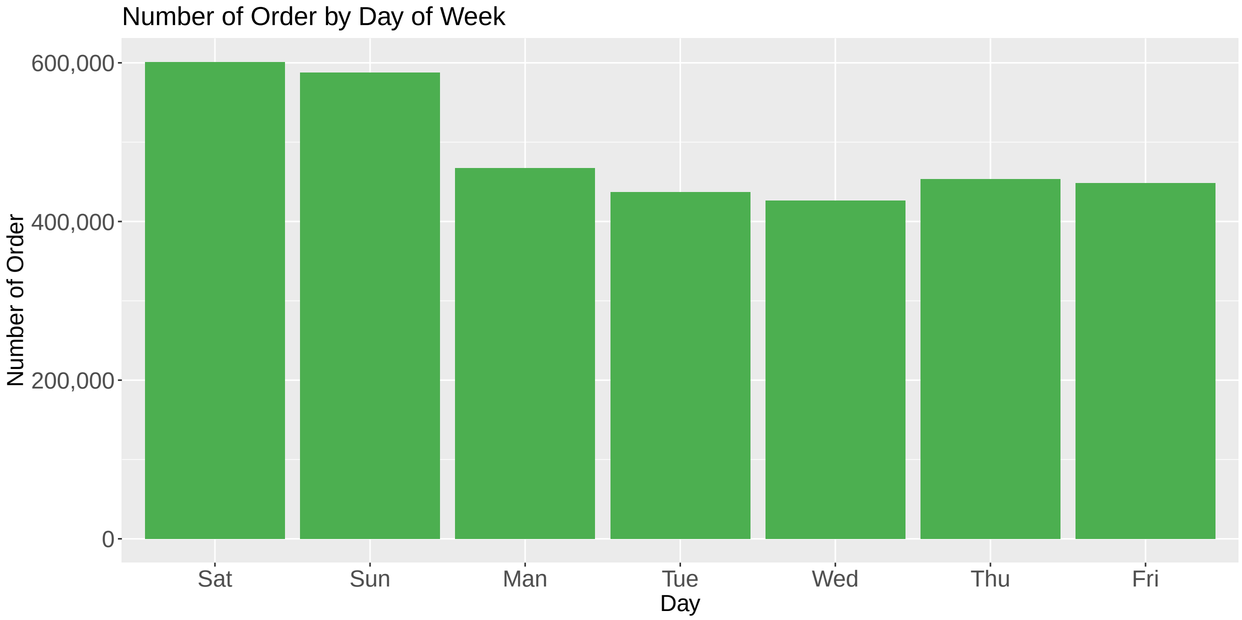
```
rm(Product_number)
```

Order Number according to Day Of Week

It can be said that customers order at weekends more. (In the row data days described by number from 0 to 6. We assume that 0 equals Saturday)

```
#ls(orders_full)
orders_full<-orders_full %>% mutate(order_day=ifelse(order_dow==0,"Sat",
ifelse(order_dow==1,"Sun",
ifelse(order_dow==2,"Man",
ifelse(order_dow==3,"Tue",
ifelse(order_dow==4,"Wed",
ifelse(order_dow==5,"Thu",
ifelse(order_dow==6,"Fri",NA))))))))


order_number_by_day<-orders_full %>%  group_by(order_day,order_dow) %>% summarise(number_of_order=n_dis
ggplot(order_number_by_day,aes(x=reorder(order_day,order_dow),y=number_of_order)) +geom_bar(fill="#4caf
```

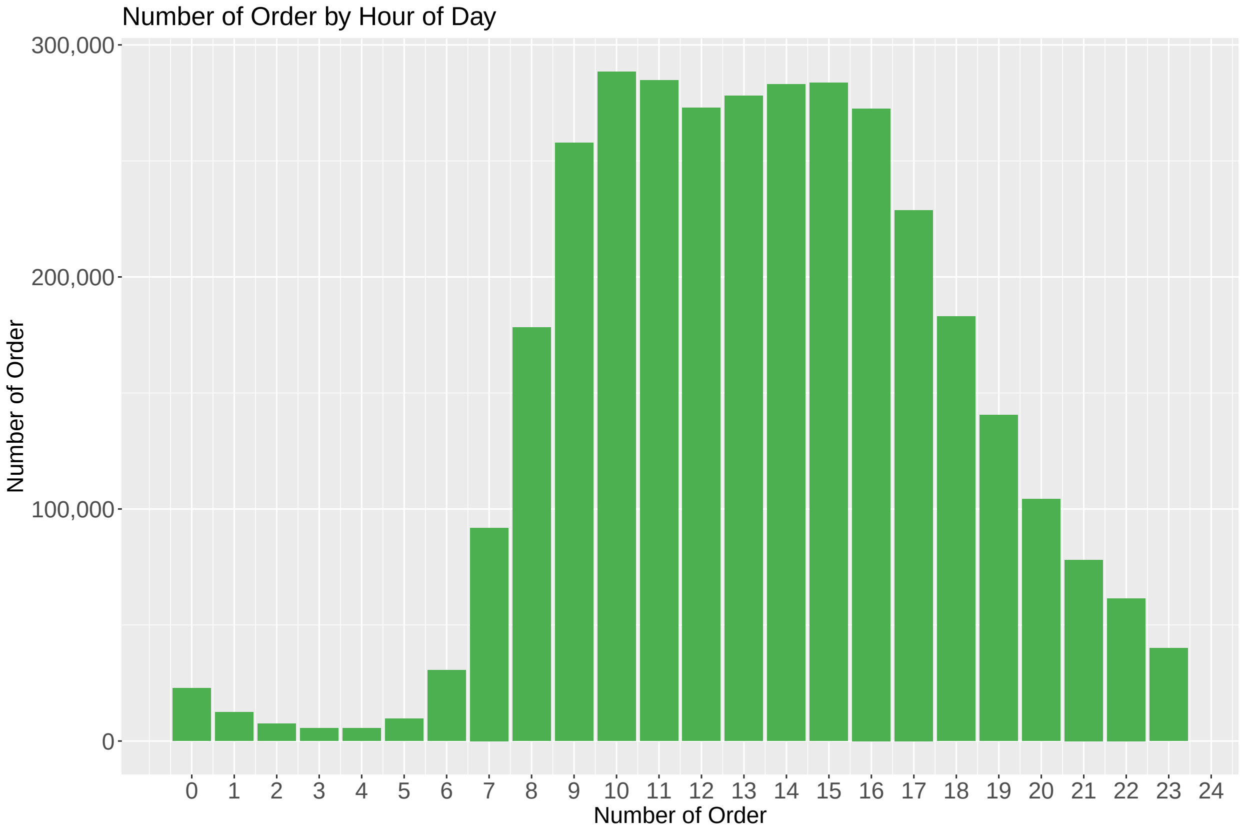


```
rm(order_number_by_day)
```

Order Number according to Hour of Day

Most orders come during daytime, from 7 am to 7 pm, which is not very surprising.

```
#ls(orders_full)
#orders_full <- mutate(orders_full , order_day=derivedFactor('Sat'=order_dow==0,'Sun'=order_dow==1,'Man
order_number_by_hour<-orders_full %>%  group_by(order_hour_of_day) %>% summarise(number_of_order=n_dist
ggplot(order_number_by_hour,aes(x=order_hour_of_day,y=number_of_order)) +geom_bar(fill="#4caf50", stat =
```



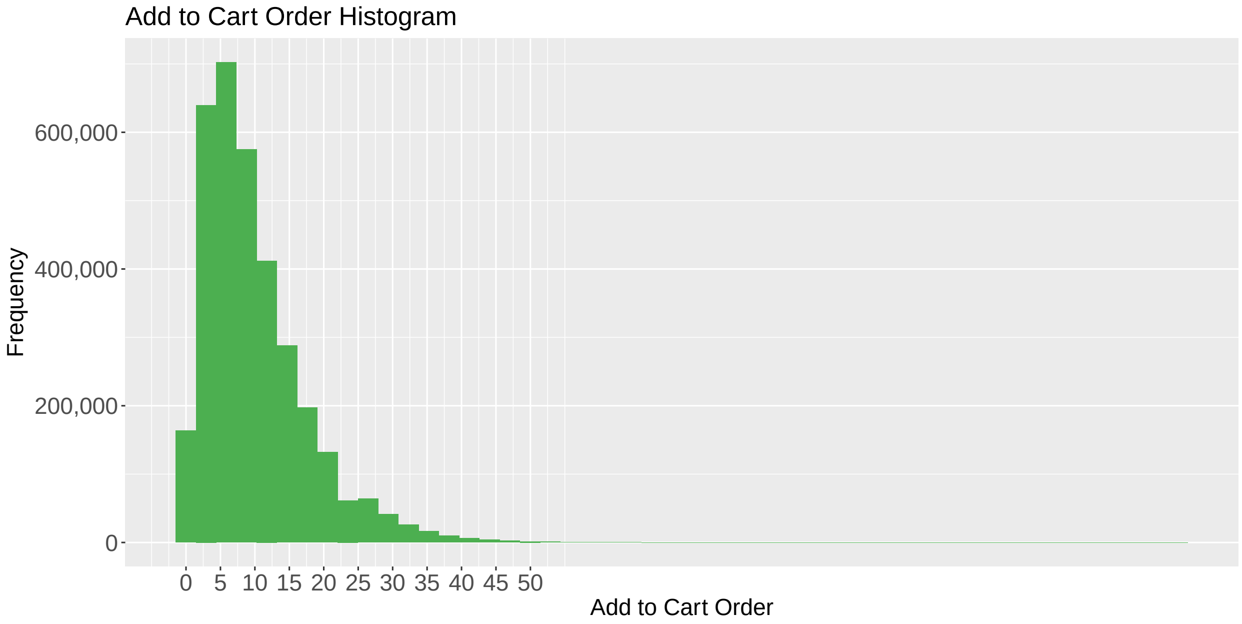
```
rm(order_number_by_hour)
```

Item Number in the Orders

People mostly add to cart less than 10 items when they order. As it clear in the graph, after 6 item per order there is negative correlation between number of items in the order and order number.

```
Max_Products_by_Order_Id<-orders_full %>%  group_by(order_id) %>% summarise(max_product=max(add_to_cart_
ggplot(Max_Products_by_Order_Id,aes(x=max_product)) + geom_histogram( fill="#4caf50" ,bins = 50)  +
  ggtitle('Add to Cart Order Histogram')+theme(axis.text = element_text(size=16) ,axis.title = element_
```

```
## Warning: Removed 75000 rows containing non-finite values (stat_bin).
```



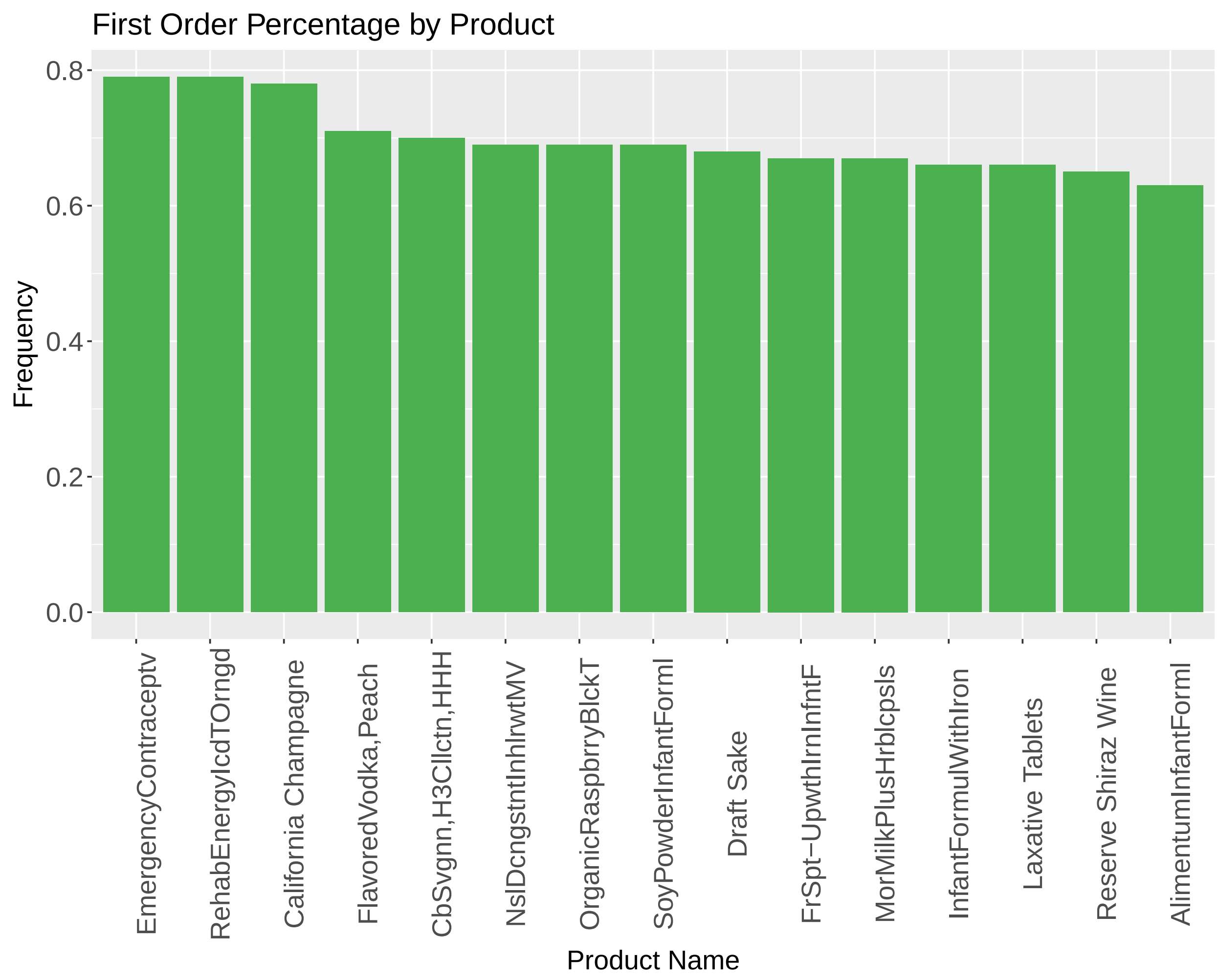
```
rm(Max_Products_by_Order_Id)
```

First Order Ratio by Product

People put those products first to cart if they buy them. Emergency contraceptive, Rethab Energy Iced Tea and California Champagne around 80% of time added the cart first when they were bought.

```
first_add_product<-orders_full %>%  group_by(product_name) %>% summarise(first_percentage=round(n_disti

ggplot(first_add_product%>%filter(row_number()<=15) ,aes(x=reorder(product_name,-first_percentage),y=fi
```



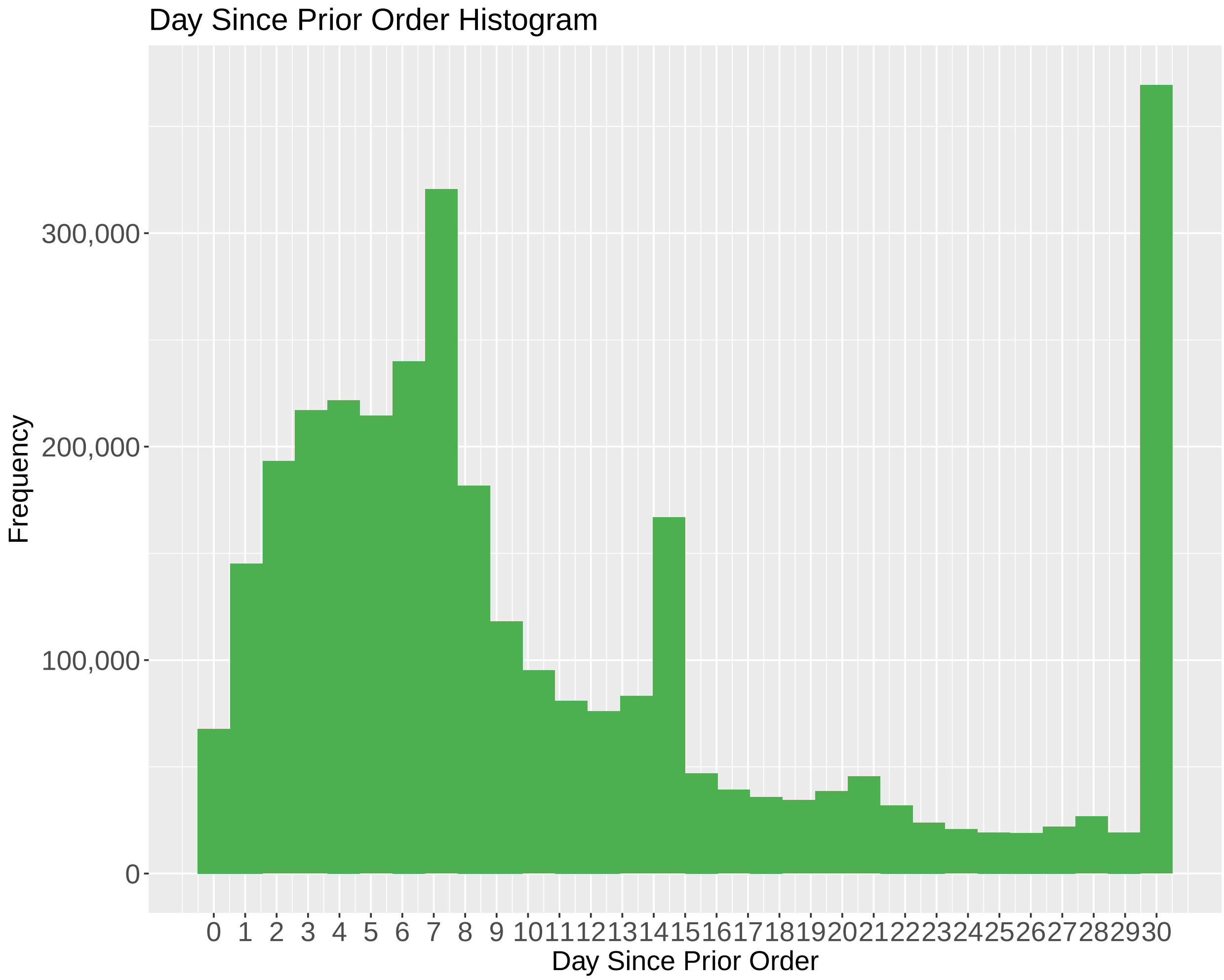
```
rm(first_add_product)
```

Day Since Prior Order

People tend to order more in 7 or less days after their prior order

```
days_since_prior_order_data<-orders_full %>%  group_by(order_id) %>% summarise(days_since_prior_order=ma
ggplot(days_since_prior_order_data,aes(x=days_since_prior_order)) + geom_histogram(
fill="#4caf50" ,bins = 30) +theme(axis.text = element_text(size = 16),axis.title = element_text(size = 
```

```
## Warning: Removed 206209 rows containing non-finite values (stat_bin).
```

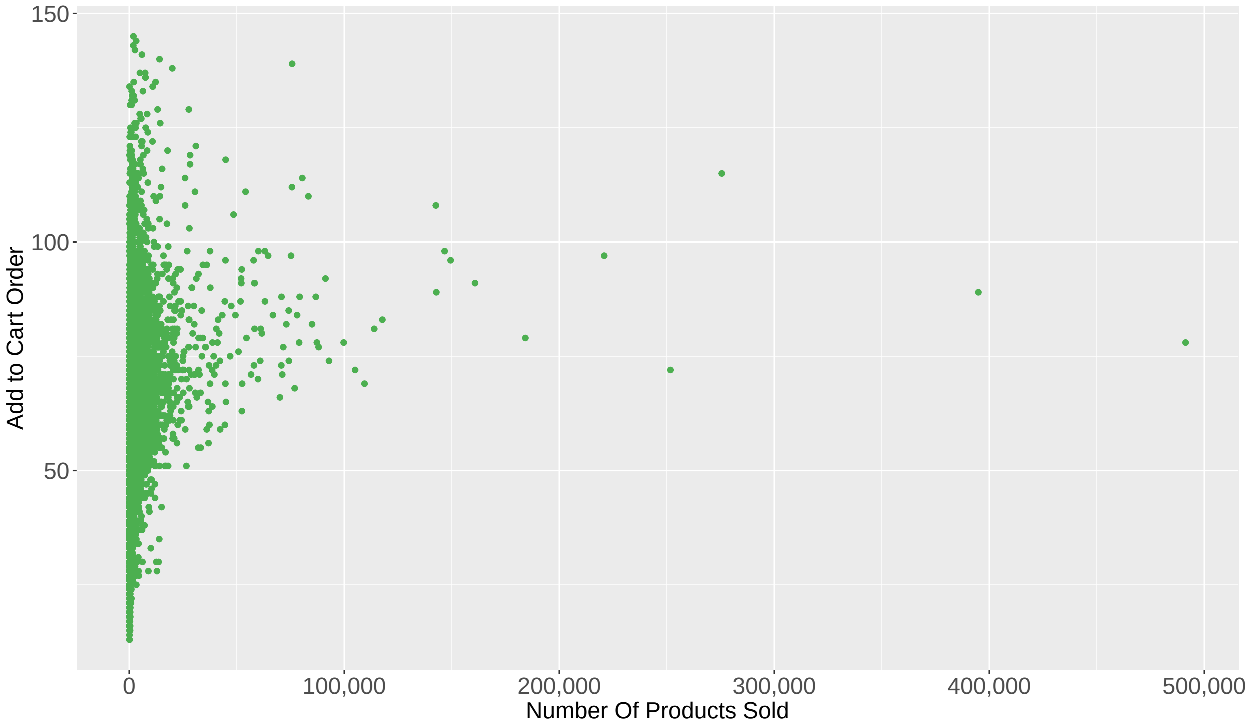


```
rm(days_since_prior_order_data)
```

Relationship Add to Cart Order and Product Sales

The chart show that products between 50 and 100 add to cart order is the sold more than others. So there is no relationship add to cart order and number of products sold.

```
Product_number<-orders_full %>% filter(!is.na(product_name)) %>%  group_by(product_name) %>% summarise(N
ggplot(Product_number,aes(x=Number_of_Product,y=add_to_cart_order)) + geom_point(color="#4caf50") +theme
```



```
rm(Product_number)
```

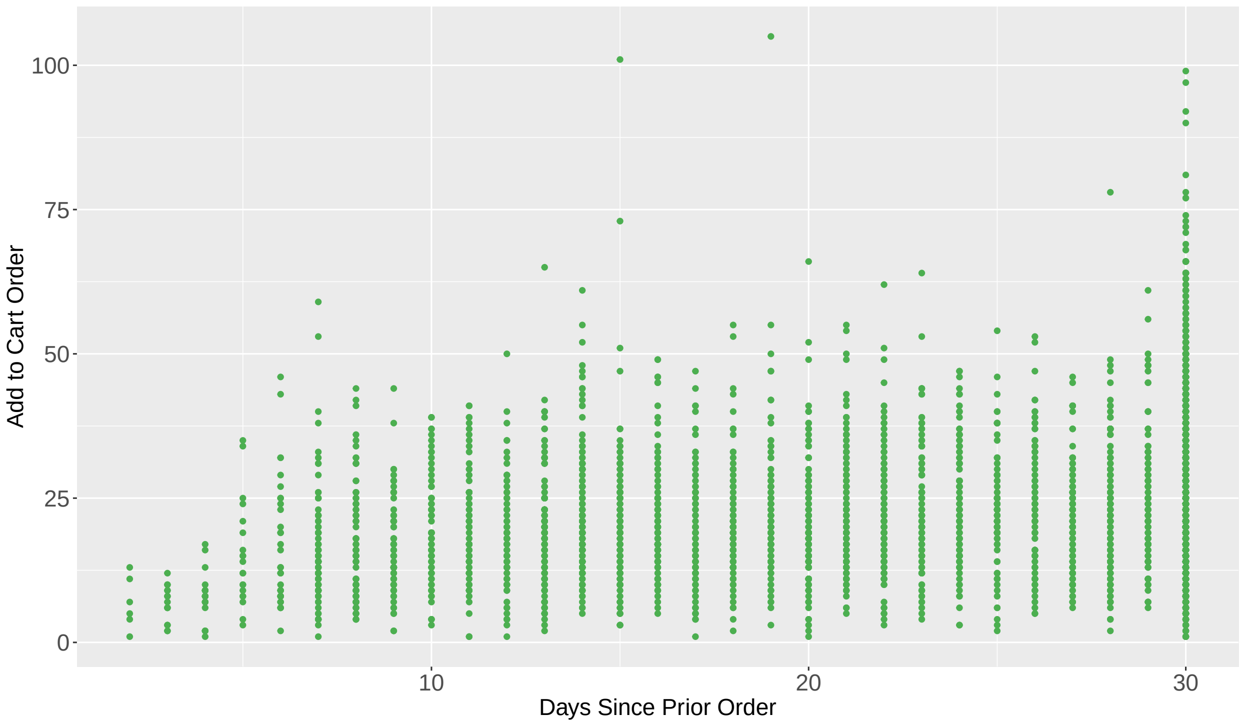
Relationship Add to Cart Order and Day Since Prior Order

The cart show that as Day since Prior Order increases, the number of products in order will increase. As it is seen below, in days between 1 and 5, there are few products in order. In conclusion people tend add more products to basket as days since prior order increase.

```
temp<-orders_full  %>%  group_by(product_name) %>% summarise(maximum_product=max(add_to_cart_order) , da

ggplot(temp,aes(x=days_since_prior_order,y=maximum_product)) + geom_point(color="#4caf50") +theme(axis.t
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



```
rm(temp)
```

Basket Analysis

We conduct market basket analysis so that looking for combination of products' aisles that occur together frequently in transaction. We assign support as 0.1 because we want to show of product which is sold together more than 300.000 times. In addition we assign confidence as 0.50 since we want to show products sold together more than %50 chance.

```
train_set= orders_full%>% filter(eval_set=='prior')
order_product_pivot <- train_set %>% select(order_id,aisle_id)
rm(train_set)

order_product_pivot <- as(split(order_product_pivot$aisle_id, order_product_pivot$order_id), "transacti
```

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

```
rules <- apriori(order_product_pivot,parameter=list(supp=0.1, conf=0.50, target="rules", maxlen=2,minle
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5     0.1      2
##  maxlen target   ext
##       2  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
```

```
## Absolute minimum support count: 321487
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[134 item(s), 3214874 transaction(s)] done [1.60s].
## sorting and recoding items ... [16 item(s)] done [0.26s].
## creating transaction tree ... done [3.58s].
## checking subsets of size 1 2
```

```
## Warning in apriori(order_product_pivot, parameter = list(supp = 0.1, conf
## = 0.5, : Mining stopped (maxlen reached). Only patterns up to a length of 2
## returned!
```

```
##  done [0.00s].
## writing ... [15 rule(s)] done [0.00s].
## creating S4 object  ... done [0.47s].
```

```
options(digits=5)
kable(data.frame(inspect(rules)) %>% arrange(desc(confidence)))
```

```
##      lhs      rhs   support confidence lift   count
## [1]  {91}  => {24}  0.11763 0.69296    1.2440  378158
## [2]  {115} => {24}  0.10993 0.57552    1.0332  353415
## [3]  {112} => {24}  0.11237 0.68531    1.2303  361249
## [4]  {107} => {24}  0.10531 0.62920    1.1296  338545
## [5]  {84}  => {83}  0.12509 0.51164    1.1522  402145
## [6]  {84}  => {24}  0.16442 0.67252    1.2073  528593
## [7]  {21}  => {83}  0.13487 0.58761    1.3232  433598
## [8]  {21}  => {24}  0.15498 0.67520    1.2122  498231
## [9]  {120} => {83}  0.14404 0.54667    1.2311  463077
## [10] {120} => {24}  0.18807 0.71377    1.2814  604624
## [11] {123} => {83}  0.23445 0.63917    1.4393  753740
## [12] {83}  => {123} 0.23445 0.52797    1.4393  753740
## [13] {123} => {24}  0.27027 0.73681    1.3228  868883
## [14] {83}  => {24}  0.31776 0.71557    1.2846 1021564
## [15] {24}  => {83}  0.31776 0.57046    1.2846 1021564
```

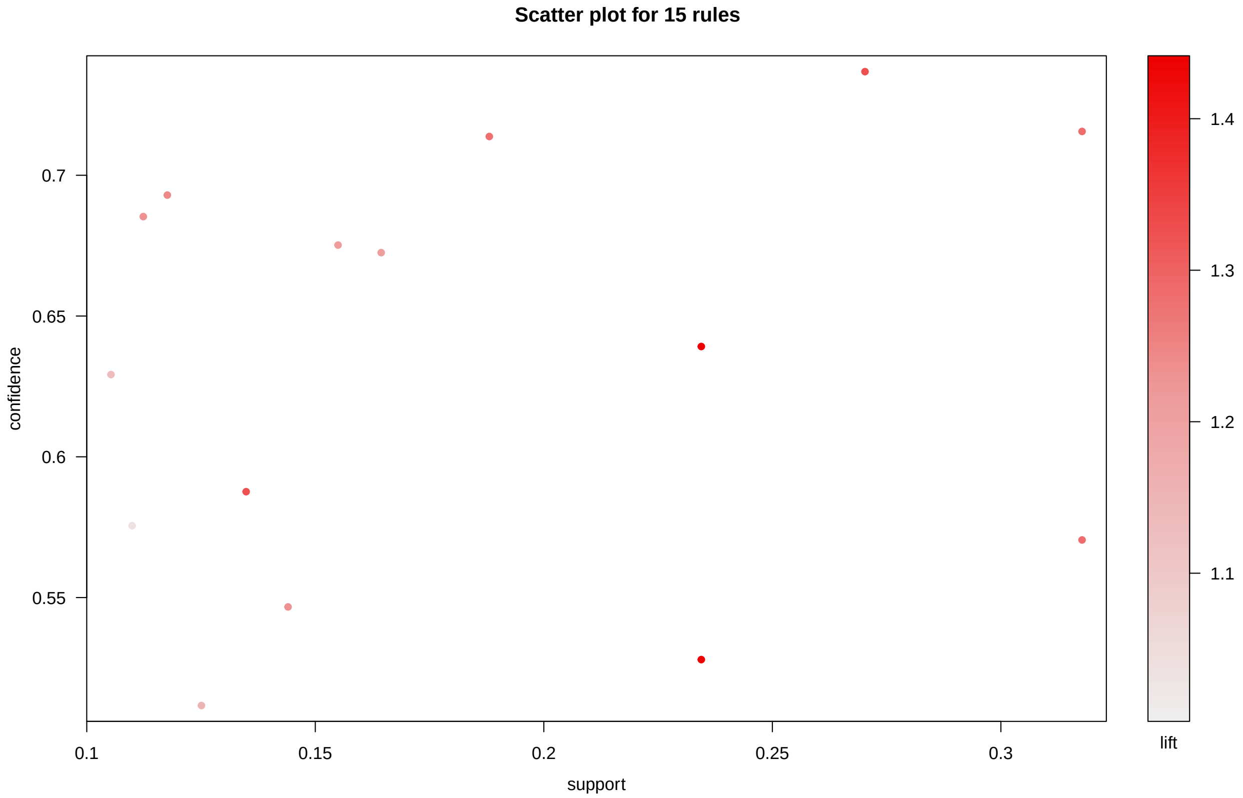
lhs	Var.2	rhs	support	confidence	lift	count
{123}	=>	{24}	0.27027	0.73681	1.3228	868883
{83}	=>	{24}	0.31776	0.71557	1.2846	1021564
{120}	=>	{24}	0.18807	0.71377	1.2814	604624
{91}	=>	{24}	0.11763	0.69296	1.2440	378158
{112}	=>	{24}	0.11237	0.68531	1.2303	361249
{21}	=>	{24}	0.15498	0.67520	1.2122	498231
{84}	=>	{24}	0.16442	0.67252	1.2073	528593
{123}	=>	{83}	0.23445	0.63917	1.4393	753740
{107}	=>	{24}	0.10531	0.62920	1.1296	338545
{21}	=>	{83}	0.13487	0.58761	1.3232	433598
{115}	=>	{24}	0.10993	0.57552	1.0332	353415
{24}	=>	{83}	0.31776	0.57046	1.2846	1021564
{120}	=>	{83}	0.14404	0.54667	1.2310	463077
{83}	=>	{123}	0.23445	0.52797	1.4393	753740
{84}	=>	{83}	0.12509	0.51164	1.1522	402145

```
#rules
```

As it is shown in the chart that greatest value for support 0.317 means that fresh fruit and fresh vegatables added cart together most frequently(more than 1mn times). with 0.715 confidence value. But greatest value for confidence belongs to packaged vegatables fruits and fresh fruits with 0.736 which means 74% of packaged vegatables fruits transaction fresh fruit also are bought. On the other hand with 0.105 support value chips pretzels and fresh fruits added to cart least freaquently.Finally lowest confidence value is between milk fresh vegatables, 51%.

As it seen on the graph there are not significant correlation for support, confidence and lift of product of aisles.

```
plot(rules )
```

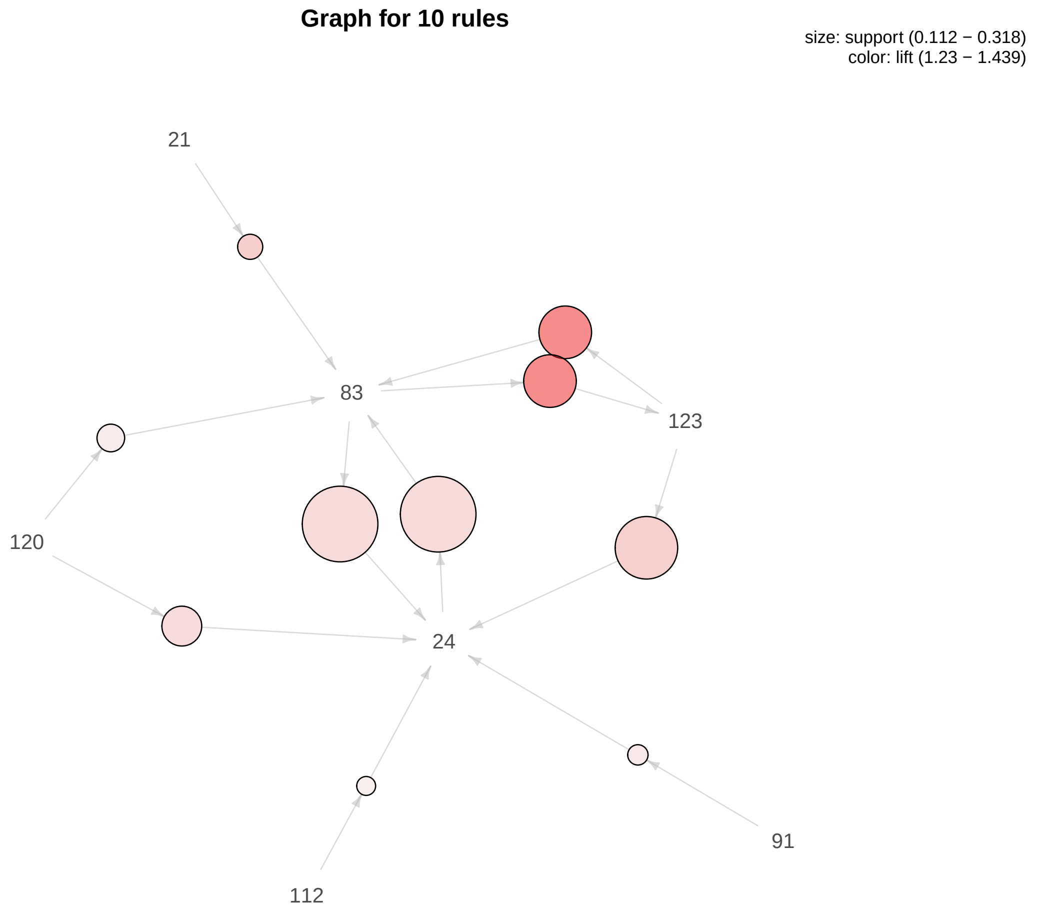


On the last graph, it is clear that some products from fresh fruits and fresh vegatables aisles added most of the transactions.

```
plot(head(sort(rules , by="lift"),10), method="graph", control=list(type="items"))
```

```
## Warning: Unknown control parameters: type

## Available control parameters (with default values):
## main  =  Graph for 10 rules
## nodeColors    =  c("#66CC6680", "#9999CC80")
## nodeCol   =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE121
## edgeCol   =  c("#474747FF", "#494949FF", "#4B4B4BFF", "#4D4D4DFF", "#4F4F4FFF", "#515151FF", "#53535
## alpha     =  0.5
## cex   =  1
## itemLabels    =  TRUE
## labelCol  =  #000000B3
## measureLabels     =  FALSE
## precision     =  3
## layout    =  NULL
## layoutParams  =  list()
## arrowSize     =  0.5
## engine    =  igraph
## plot  =  TRUE
## plot_options  =  list()
## max   =  100
## verbose   =  FALSE
```



References

"The Instacart Online Grocery Shopping Dataset 2017", Accessed from https://www.instacart.com/datasets/grocery-shopping-2017 on October 24,2017

"Isntacart Market Basket Analysis", Accessed from https://www.kaggle.com/c/instacart-market-basket-analysis on October 25,2017