Instacart Prediction

Karim Souidi, June 22

Exploratory data analysis

Load libraries

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.5.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.5.2
```

```
library(arules)
```

```
## Warning: package 'arules' was built under R version 3.5.2
```

```
## Warning: package 'Matrix' was built under R version 3.5.2
```

```
library(arulesViz)
```

```
## Warning: package 'arulesViz' was built under R version 3.5.2
```

```
library(grid)
library(magrittr)
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.5.2
```

View each dataset

Products

```
## Observations: 49,688
## Variables: 4
## $ product_id    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...
## $ product_name  <fct> Chocolate Sandwich Cookies, All-Seasons Salt, Ro...
## $ aisle_id      <int> 61, 104, 94, 38, 5, 11, 98, 116, 120, 115, 31, 1...
## $ department_id <int> 19, 13, 7, 1, 13, 11, 7, 1, 16, 7, 7, 1, 11, 17,...
```

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

product_id	product_name	aisle_id	department_id
6	Dry Nose Oil	11	11
7	Pure Coconut Water With Orange	98	7
8	Cut Russet Potatoes Steam N' Mash	116	1
9	Light Strawberry Blueberry Yogurt	120	16
10	Sparkling Orange Juice & Prickly Pear Beverage	115	7
11	Peach Mango Juice	31	7
12	Chocolate Fudge Layer Cake	119	1
13	Saline Nasal Mist	11	11
14	Fresh Scent Dishwasher Cleaner	74	17
15	Overnight Diapers Size 6	56	18

Orders

```
## Observations: 3,421,083
## Variables: 7
## $ order_id               <int> 2539329, 2398795, 473747, 2254736, 4315...
## $ user_id                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, ...
## $ eval_set               <fct> prior, prior, prior, prior, prior, prio...
## $ order_number           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2...
## $ order_dow              <int> 2, 3, 3, 4, 4, 2, 1, 1, 1, 4, 4, 2, 5, ...
## $ order_hour_of_day      <int> 8, 7, 12, 7, 15, 7, 9, 14, 16, 8, 8, 11...
## $ days_since_prior_order <dbl> NA, 15, 21, 29, 28, 19, 20, 14, 0, 30, ...
```

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
3367565	1	prior	6	2	7	19
550135	1	prior	7	1	9	20
3108588	1	prior	8	1	14	14
2295261	1	prior	9	1	16	0
2550362	1	prior	10	4	8	30
1187899	1	train	11	4	8	14
2168274	2	prior	1	2	11	NA
1501582	2	prior	2	5	10	10
1901567	2	prior	3	1	10	3
738281	2	prior	4	2	10	8

Order_products_train

```
## Observations: 1,384,617
## Variables: 4
## $ order_id          <int> 1, 1, 1, 1, 1, 1, 1, 1, 36, 36, 36, 36, 36, ...
## $ product_id        <int> 49302, 11109, 10246, 49683, 43633, 13176, 47...
## $ add_to_cart_order <int> 1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7,...
## $ reordered         <int> 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1,...
```

order_id	product_id	add_to_cart_order	reordered
1	49302	1	1
1	11109	2	1
1	10246	3	0
1	49683	4	0
1	43633	5	1
1	13176	6	0
1	47209	7	0
1	22035	8	1
36	39612	1	0
36	19660	2	1
36	49235	3	0
36	43086	4	1
36	46620	5	1
36	34497	6	1
36	48679	7	1

Order_products_prior

```
## Observations: 32,434,489
## Variables: 4
## $ order_id          <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3,...
## $ product_id        <int> 33120, 28985, 9327, 45918, 30035, 17794, 401...
## $ add_to_cart_order <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6,...
## $ reordered         <int> 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,...
```

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
2	17794	6	1
2	40141	7	1
2	1819	8	1
2	43668	9	0
3	33754	1	1
3	24838	2	1
3	17704	3	1
3	21903	4	1
3	17668	5	1
3	46667	6	1

Departments

```
## Observations: 21
## Variables: 2
## $ department_id <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...
## $ department    <fct> frozen, other, bakery, produce, alcohol, interna...
```

department_id	department
1	frozen
2	other
3	bakery
4	produce
5	alcohol
6	international
7	beverages
8	pets
9	dry goods pasta
10	bulk
11	personal care
12	meat seafood
13	pantry
14	breakfast
15	canned goods

Aisles

```
## Observations: 134
## Variables: 2
## $ aisle_id <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16...
## $ aisle    <fct> prepared soups salads, specialty cheeses, energy gran...
```

aisle_id	aisle
1	prepared soups salads
2	specialty cheeses
3	energy granola bars
4	instant foods
5	marinades meat preparation
6	other
7	packaged meat
8	bakery desserts
9	pasta sauce
10	kitchen supplies
11	cold flu allergy
12	fresh pasta
13	prepared meals
14	tofu meat alternatives
15	packaged seafood

Let's check how many unique oders and better understand the dimensons of these key tables:

Check unque records of key tables

Orders table

```
length(unique(orders$order_id))
```

```
## [1] 3421083
```

Order_prod_prior

```
length(unique(order_prod_prior$order_id))
```

```
## [1] 3214874
```

Order_prod_train

```
length(unique(order_prod_train$order_id))
```

```
## [1] 131209
```

Frequency of Priors, Train and Test

```
table(orders$eval_set)
```

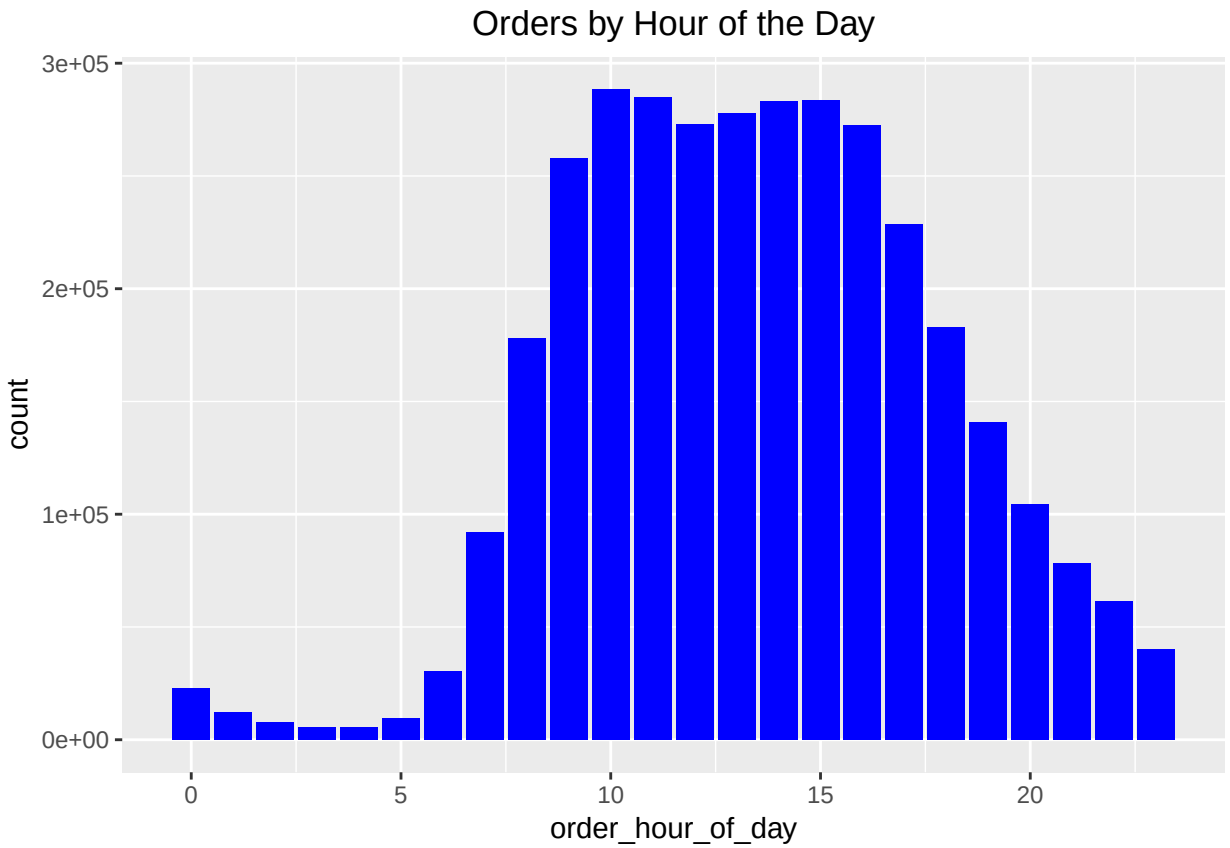
```
##
##   prior    test   train
## 3214874   75000  131209
```

Change types of variables

```
orders$order_hour_of_day <- as.numeric(orders$order_hour_of_day)
orders$eval_set <- as.factor(orders$eval_set)
products$product_name <- as.factor(products$product_name)
departments$department <- as.factor(departments$department_id)
orders$order_dow<- factor(orders$order_dow,
                          labels = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
```

Plot Orders by hour of the day:

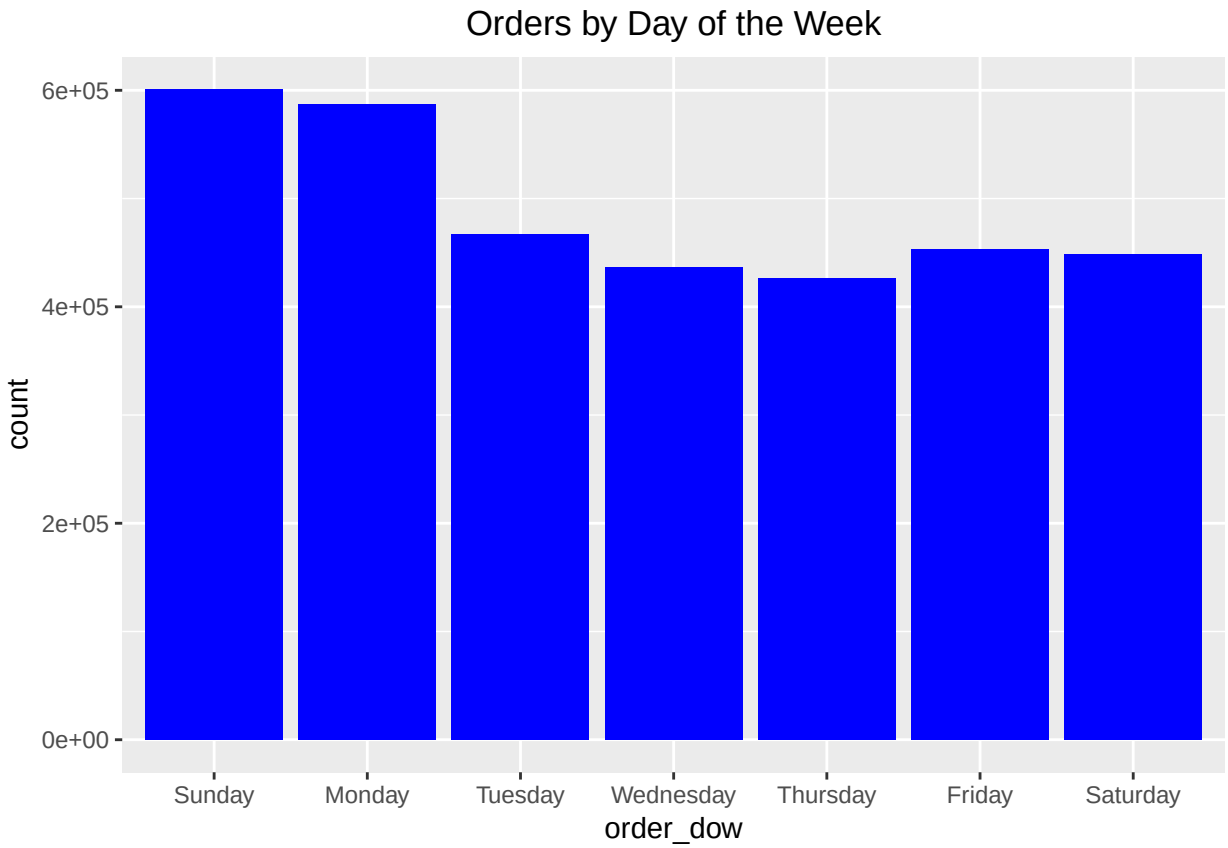
```
ggplot(orders, aes(x=order_hour_of_day)) +
  geom_histogram(stat="count",fill="blue")+
  ggtitle("Orders by Hour of the Day")+
  theme(plot.title = element_text(hjust = 0.5))
```



ggplot(Order_prod_tot_with_user_id, aes(x=order_hour_of_day)) + geom_histogram(stat="count",fill="blue")+ ggtitle("Time to order") + theme(plot.title = element_text(hjust = 0.5))

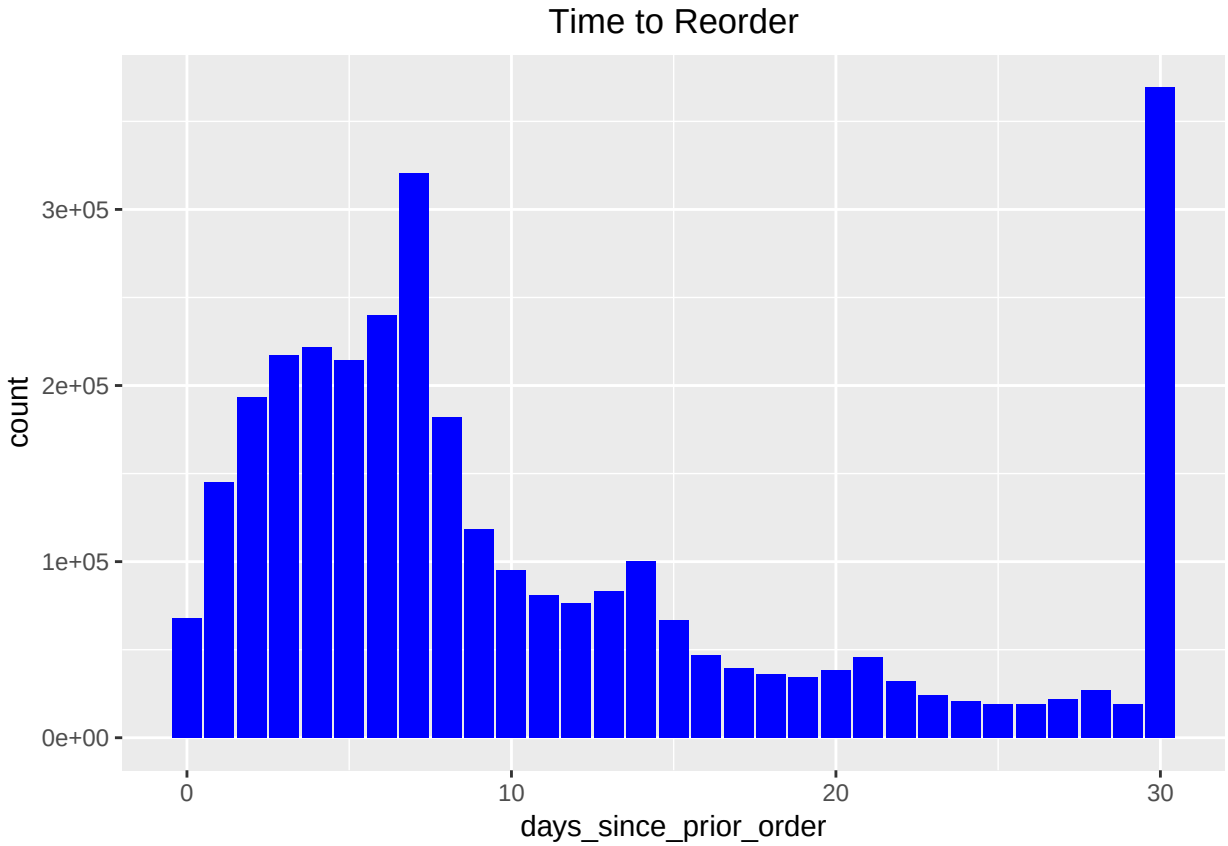
Plot Orders by day of the week

```
ggplot(orders, aes(x=order_dow)) +
  geom_histogram(stat="count",fill="blue")+
  ggtitle("Orders by Day of the Week")+
  theme(plot.title = element_text(hjust = 0.5))
```



Plot time to reorder

```
ggplot(orders, aes(x=days_since_prior_order)) +
geom_histogram(stat="count",fill="blue")+
ggtitle("Time to Reorder") +
theme(plot.title = element_text(hjust = 0.5))
```



Merge products prior and train

```
Orders_prod_tot <- rbind(order_prod_prior, order_prod_train)###
glimpse(Orders_prod_tot)###
```

```
## Observations: 33,819,106
## Variables: 4
## $ order_id          <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3,...
## $ product_id        <int> 33120, 28985, 9327, 45918, 30035, 17794, 401...
## $ add_to_cart_order <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6,...
## $ reordered         <int> 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,...
```

```
kable(head(Orders_prod_tot, 15))
```

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
2	17794	6	1
2	40141	7	1
2	1819	8	1
2	43668	9	0
3	33754	1	1
3	24838	2	1

order_id	product_id	add_to_cart_order	reordered
3	17704	3	1
3	21903	4	1
3	17668	5	1
3	46667	6	1

Join on total orders with products

```
Orders_prod_tot_with_prod <- left_join(Orders_prod_tot, products, by="product_id") ###
```

Remove fields not needed

```
Orders_prod_tot_with_prod$product_id<-NULL
Orders_prod_tot_with_prod$add_to_cart_order<-NULL
Orders_prod_tot_with_prod$reordered<-NULL
Orders_prod_tot_with_prod$aisle_id<-NULL
Orders_prod_tot_with_prod$department_id<-NULL
```

Top 10 products

```
Top_products <- Orders_prod_tot_with_prod %>%
  group_by(product_name) %>%
  summarize(count = n()) %>%
  top_n(10, wt = count) %>%
  arrange(desc(count))
Top_10_products <- Top_products [1:10, ]###
kable(head(Top_10_products, 10))
```

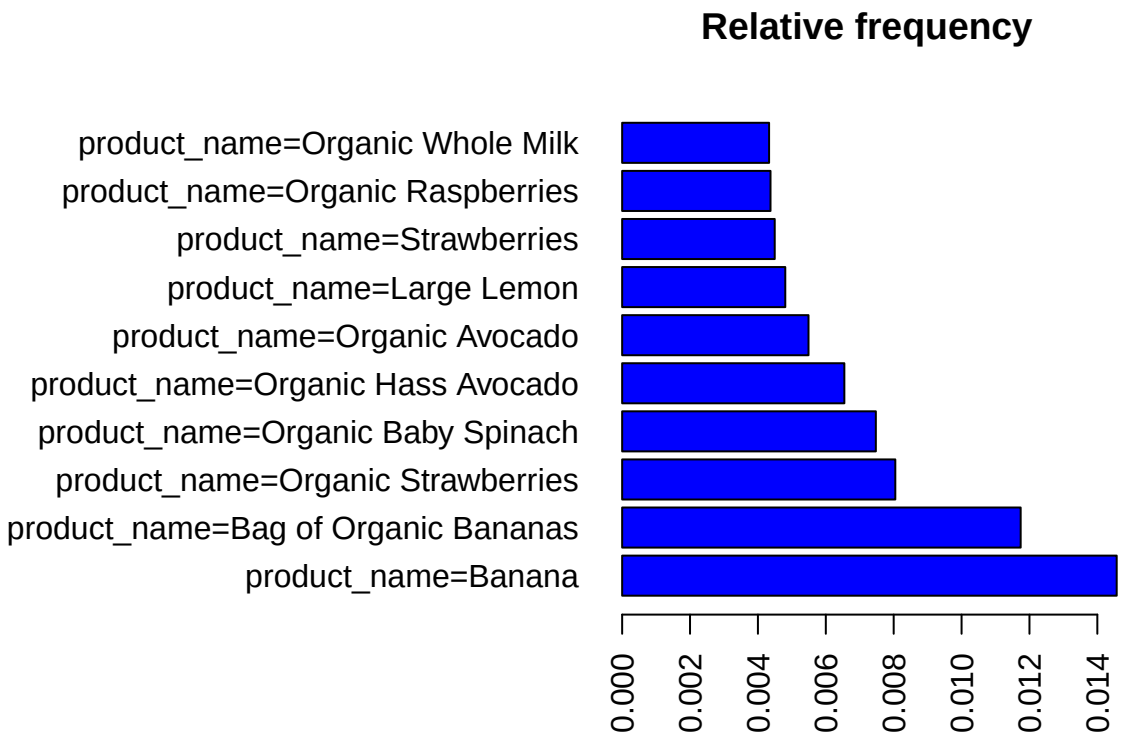
product_name	count
Banana	491291
Bag of Organic Bananas	394930
Organic Strawberries	275577
Organic Baby Spinach	251705
Organic Hass Avocado	220877
Organic Avocado	184224
Large Lemon	160792
Strawberries	149445
Limes	146660
Organic Whole Milk	142813

Association Rules

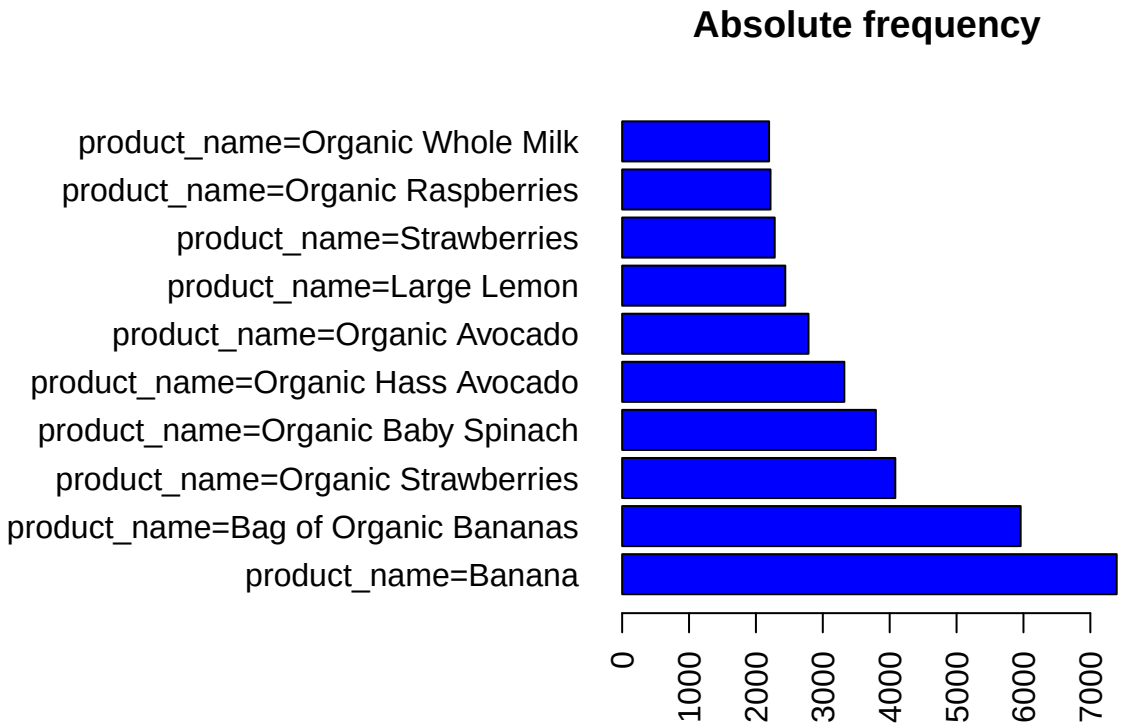
```
Orders_prod_tot_with_prod <- data.table(Orders_prod_tot_with_prod)
sample_order <- sample(unique(Orders_prod_tot_with_prod$order_id), 50000)
sample_order_prod<-subset(Orders_prod_tot_with_prod, order_id %in% sample_order)
sample_order_prod$product_name<-as.factor(sample_order_prod$product_name)
```

```r
sample_order_prod$order_id<-as.factor(sample_order_prod$order_id)
sample_order_prod2 = as(sample_order_prod,"transactions")

itemFrequencyPlot(sample_order_prod2,
                  type="relative",
                  topN=10,
                  horiz=TRUE,
                  col='blue',
                  xlab='',
                  main='Relative frequency')
```




```r
itemFrequencyPlot(sample_order_prod2,
                  type="absolute",
                  topN=10,
                  horiz=TRUE,
                  col='blue',
                  xlab='',
                  main='Absolute frequency')
```




```
rules <- apriori(sample_order_prod2, parameter=list(support=0.001, confidence=0.005))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##       0.005    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 507
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[79273 item(s), 507478 transaction(s)] done [0.54s].
## sorting and recoding items ... [102 item(s)] done [0.02s].
## creating transaction tree ... done [0.06s].
## checking subsets of size 1 done [0.00s].
## writing ... [6 rule(s)] done [0.00s].
## creating S4 object  ... done [0.09s].
```

```
rules
```

```
## set of 6 rules
```

```
summary(rules)
```

```
## set of 6 rules
##
## rule length distribution (lhs + rhs):sizes
```

```
## 1
## 6
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
##
## summary of quality measures:
##     support           confidence            lift       count
##  Min.   :0.005490   Min.   :0.005490   Min.   :1   Min.   :2786
##  1st Qu.:0.006778   1st Qu.:0.006778   1st Qu.:1   1st Qu.:3440
##  Median :0.007761   Median :0.007761   Median :1   Median :3938
##  Mean   :0.008978   Mean   :0.008978   Mean   :1   Mean   :4556
##  3rd Qu.:0.010817   3rd Qu.:0.010817   3rd Qu.:1   3rd Qu.:5490
##  Max.   :0.014570   Max.   :0.014570   Max.   :1   Max.   :7394
##
## mining info:
##                data ntransactions support confidence
##  sample_order_prod2        507478   0.001      0.005
```

```
inspect(rules[1:5])
```

```
##     lhs    rhs                                    support     confidence
## [1] {}  => {product_name=Organic Avocado}        0.005489893 0.005489893
## [2] {}  => {product_name=Organic Hass Avocado}   0.006546097 0.006546097
## [3] {}  => {product_name=Organic Baby Spinach}   0.007474216 0.007474216
## [4] {}  => {product_name=Organic Strawberries}   0.008047640 0.008047640
## [5] {}  => {product_name=Bag of Organic Bananas} 0.011740410 0.011740410
##     lift count
## [1] 1    2786
## [2] 1    3322
## [3] 1    3793
## [4] 1    4084
## [5] 1    5958
```