Final Project: Big Data with R

Lore Zumeta Olaskoaga

June 30, 2017

Questions

- Which are the most repited first time bought products?
- Which is the ranking of the days when people order the numerous baskets?
- Which product is less frequently bought again, the most disappointing product.

The analysis

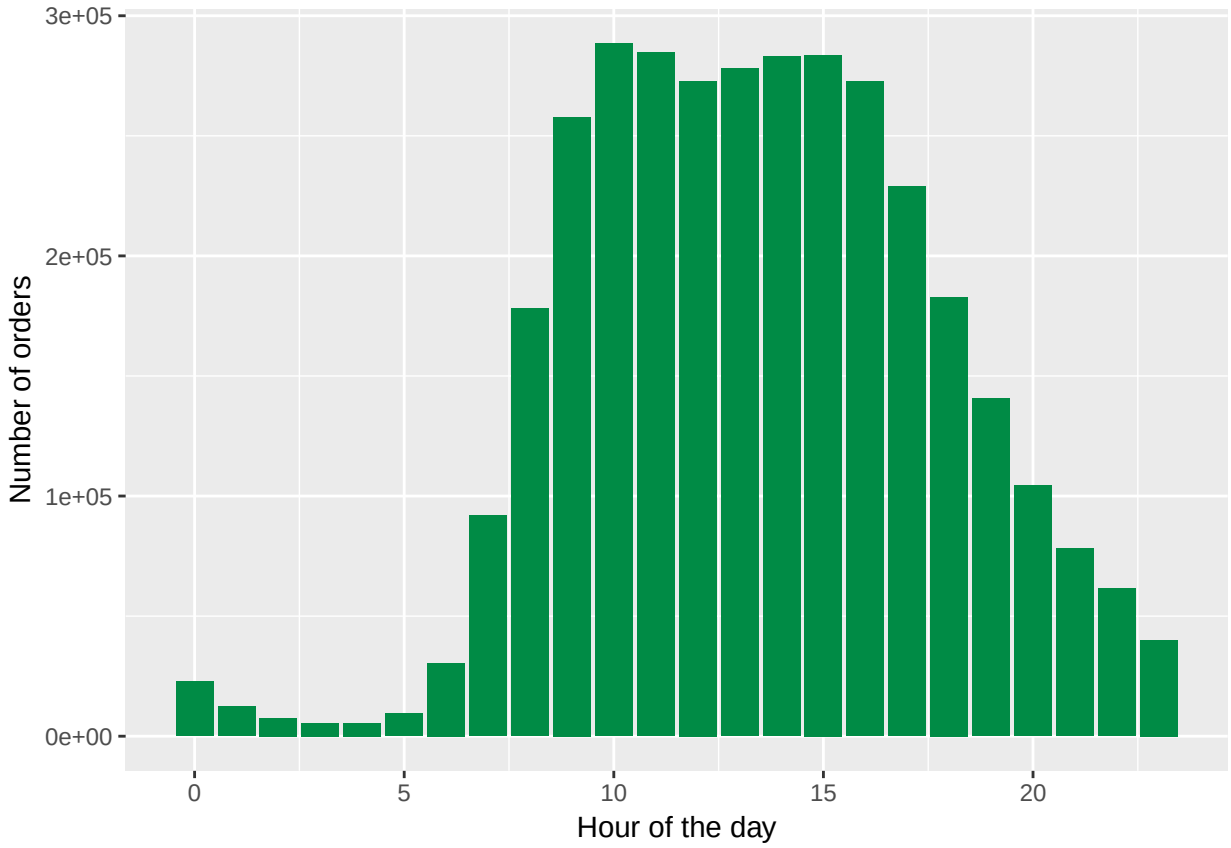
First of all, let us load the needed tables and libraries,

```
source("readDataToMemory.R")
readInstacart()

library(DBI)
library(ggplot2)
library(ggthemes)
library(knitr)

src_tbls(sc)
```

```
## [1] "order_products__prior_tbl" "order_products__train_tbl"
## [3] "orders_tbl"                "products_tbl"
```

Afterwards, let's have a look when people order,

```
orders %>%
  collect %>%
  ggplot(aes(x=order_hour_of_day)) +
  geom_histogram(stat="count",fill="springgreen4") +
  xlab("Hour of the day") +
  ylab("Number of orders")
```



We observe that most orders are between 8:00-18:00.

Now we are going to try to answer the questions made. The first one, which is the most repited first time bought product?

```
first_order <-
    "
    SELECT tp.product_id
    ,   p.product_name
    ,   tp.perc
    ,   tp.n_orders
    FROM (
          SELECT t.product_id
          ,  t.perc
          ,  t.n_orders
          FROM(
              SELECT product_id
              ,   COUNT(1) AS n_orders
              ,   add_to_cart_order
              ,   COUNT(1)/SUM(COUNT(1)) OVER(PARTITION BY product_id) AS perc
              FROM order_products__prior_tbl
              GROUP BY product_id, add_to_cart_order
              ) t
          WHERE (add_to_cart_order=1 AND n_orders>10)
          ORDER BY t.perc DESC
    ) tp
    LEFT JOIN (
        SELECT product_id
```

```
        ,  product_name
        FROM products_tbl
    ) p
    ON tp.product_id = p.product_id
    LIMIT 10"

q1 <- dbGetQuery(sc, first_order)

kable(q1)
```

product_id	product_name	perc	n_orders
35133	Emergency Contraceptive	0.7872340	37
28335	Rehab Energy Iced Tea Orangeade	0.7846154	51
2216	California Champagne	0.7777778	14
14644	Cabernet Sauvignon, H3 Collection, Horse Heaven Hills	0.7368421	14
45328	Flavored Vodka, Peach	0.7042254	50
15511	Draft Sake	0.6944444	25
19675	Organic Raspberry Black Tea	0.6923077	27
14609	Soy Powder Infant Formula	0.6857143	24
14777	Nasal Decongestant Inhaler with Medicated Vapors	0.6818182	30
25524	Infant Formula With Iron	0.6744186	29

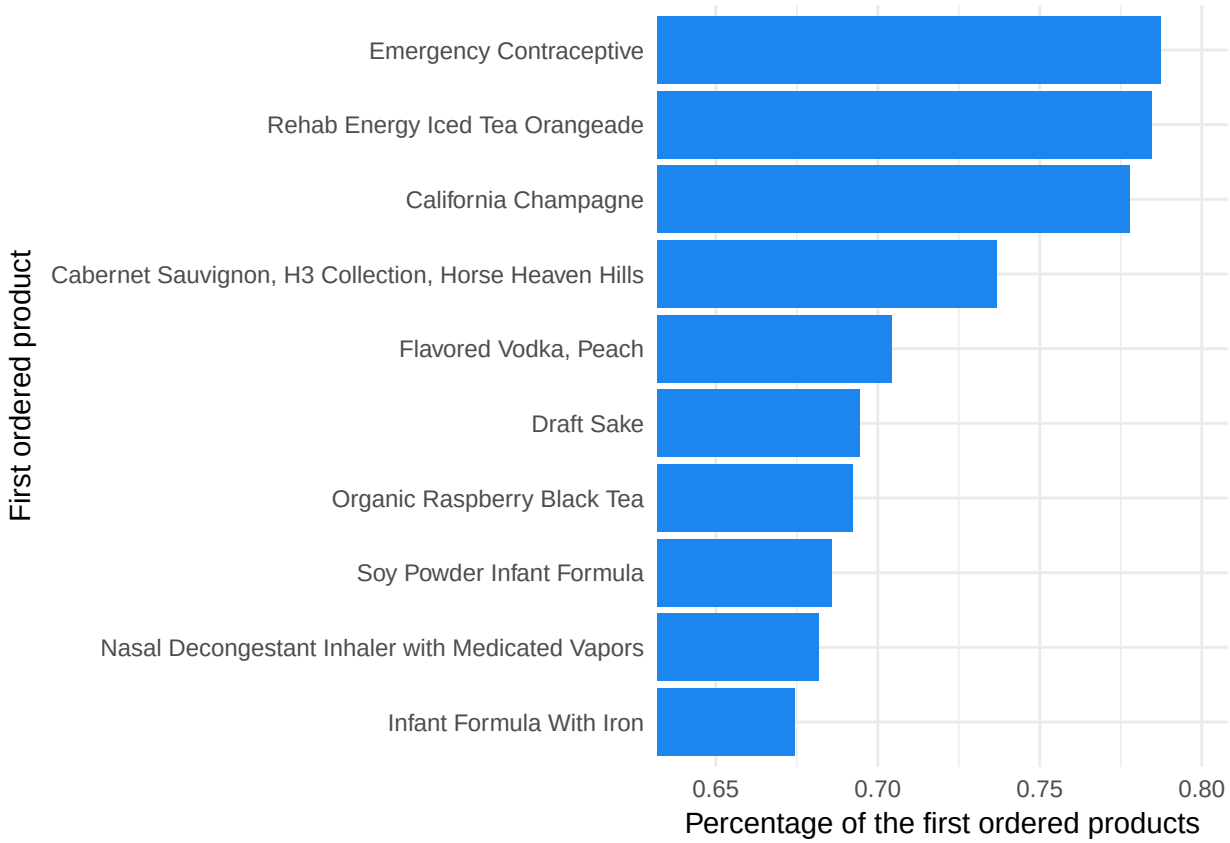
Or, using *Sparklyr*,

```
first_order_p <- order_products__prior %>%
  group_by(product_id, add_to_cart_order) %>%
  summarize(count = n()) %>% mutate(perc=count/sum(count)) %>%
  filter(add_to_cart_order == 1, count>10) %>%
  arrange(desc(perc)) %>%
  left_join(products,by="product_id") %>%
  select(product_name, perc, count) %>%
  ungroup() %>%
  top_n(10, wt=perc)

kable(first_order_p)
```

product_id	product_name	perc	count
35133	Emergency Contraceptive	0.7872340	37
28335	Rehab Energy Iced Tea Orangeade	0.7846154	51
2216	California Champagne	0.7777778	14
14644	Cabernet Sauvignon, H3 Collection, Horse Heaven Hills	0.7368421	14
45328	Flavored Vodka, Peach	0.7042254	50
15511	Draft Sake	0.6944444	25
19675	Organic Raspberry Black Tea	0.6923077	27
14609	Soy Powder Infant Formula	0.6857143	24
14777	Nasal Decongestant Inhaler with Medicated Vapors	0.6818182	30
25524	Infant Formula With Iron	0.6744186	29

Therefore, these are the most frequently first ordered products:

```r
q1 %>%
  ggplot(
    aes(reorder(product_name, perc, function(x) x), perc)) +
  geom_bar(stat="identity", fill='dodgerblue2') +
  coord_flip(ylim=c(0.64,0.8)) +
  scale_y_continuous(label=scales::comma) +
  xlab("First ordered product") +
  ylab("Percentage of the first ordered products") +
  theme_minimal()
```



Second one, which is the day of the week that people order numerous baskets?

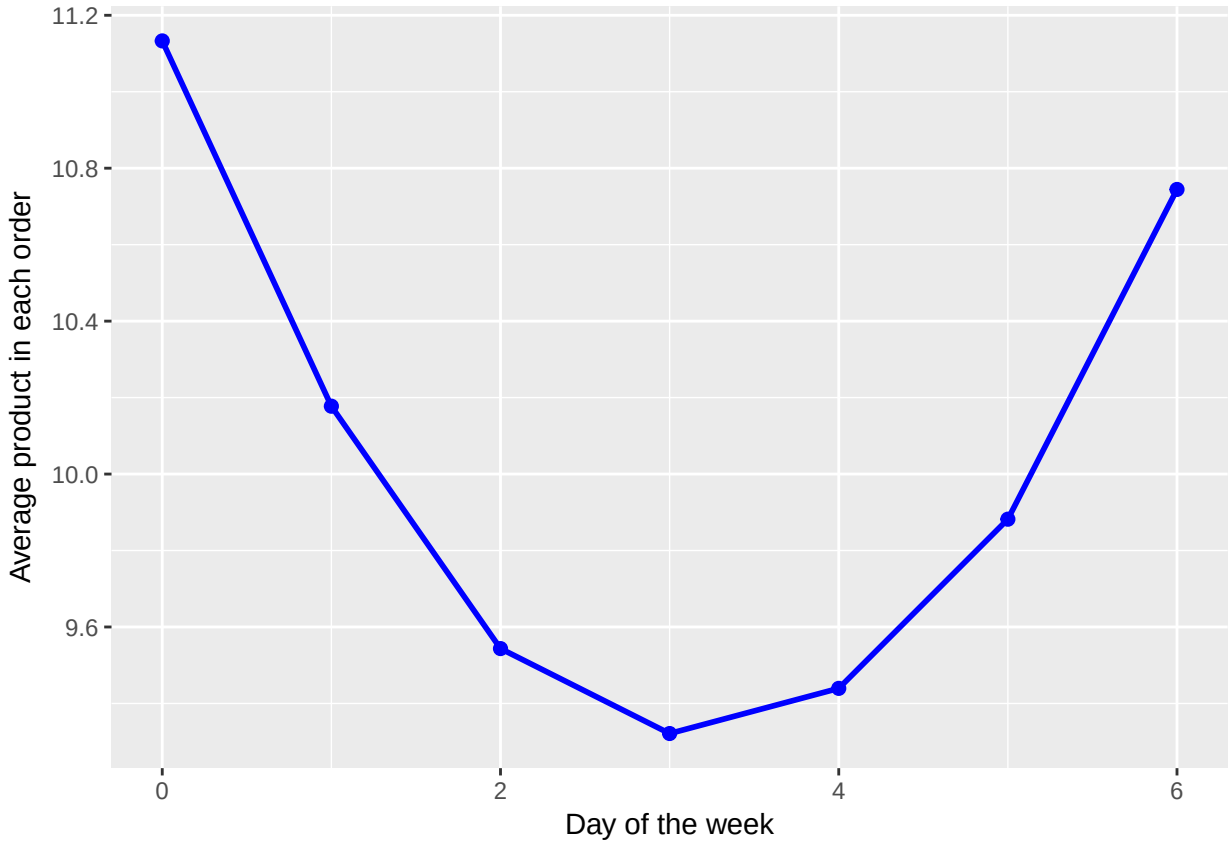
```r
day_num <-
    "
    SELECT order_dow
    ,   AVG(pvg.n_products) AS avg_products
    FROM (
        SELECT order_dow
        ,   p.order_id
        ,   p.n_products
        FROM (
            SELECT order_id
            ,   COUNT(product_id) AS n_products
            FROM order_products__prior_tbl
            GROUP BY order_id
            ORDER BY n_products DESC
        ) p
```

```
        LEFT JOIN (
            SELECT order_id
            ,   order_dow
            FROM orders_tbl
        ) o
        ON p.order_id = o.order_id
        ) pvg
    GROUP BY order_dow
    ORDER BY avg_products DESC
  "

q2 <- dbGetQuery(sc, day_num)

kable(q2)
```

order_dow	avg_products
0	11.132983
6	10.744480
1	10.177484
5	9.881950
2	9.543501
4	9.439436
3	9.321331

Let's plot it,

```
q2 %>%
  arrange(.) %>%
  ggplot(aes(order_dow, avg_products)) +
  geom_line(color="blue", size=1) + geom_point(size=2, color="blue") +
  xlab("Day of the week") +
  ylab("Average product in each order")
```



Thus, it seems that people are more likely to order numerous products on sunday and for instance, on Tuesday they do not feel like doing big orders. Customers may take advantage of the weekend (they have more time) and they may think more about their needs so they order more products.

It will be interesting if the company would offer for packages of products on Tuesday or Wednesday.

Third questiong: the more disappointing reordered products,

```
disappointing <- order_products__prior %>%
  group_by(product_id, reordered == 1) %>%
  summarize(n_prod = n()) %>%
  arrange(n_prod) %>%
  left_join(products,by="product_id") %>%
  select(product_id, product_name, n_prod) %>%
  ungroup() %>%
  head(., 10)

kable(disappointing)
```

product_id	product_name	n_prod
46951	Sweet Traditions Fresh Cut Sweet Potatoes in Light Syrup	1
15870	Steamfresh Whole Green Beans	1
36169	The Itch Eraser	1
42065	Water Enhancer, Stevia, Raspberry Lemonade	1
236	Chicken Meatballs Dog Treats	1
37415	Shine Moroccan Sleek Oil Treatment For Frizzy, Dry Hair	1
47567	Large Cut Caesar Croutons	1
2682	Rosy Lip Therapy	1
34332	Complete Omega 3 6 9 D Lemon	1

product_id	product_name	n_prod
5153	Sleepytime Honey	1

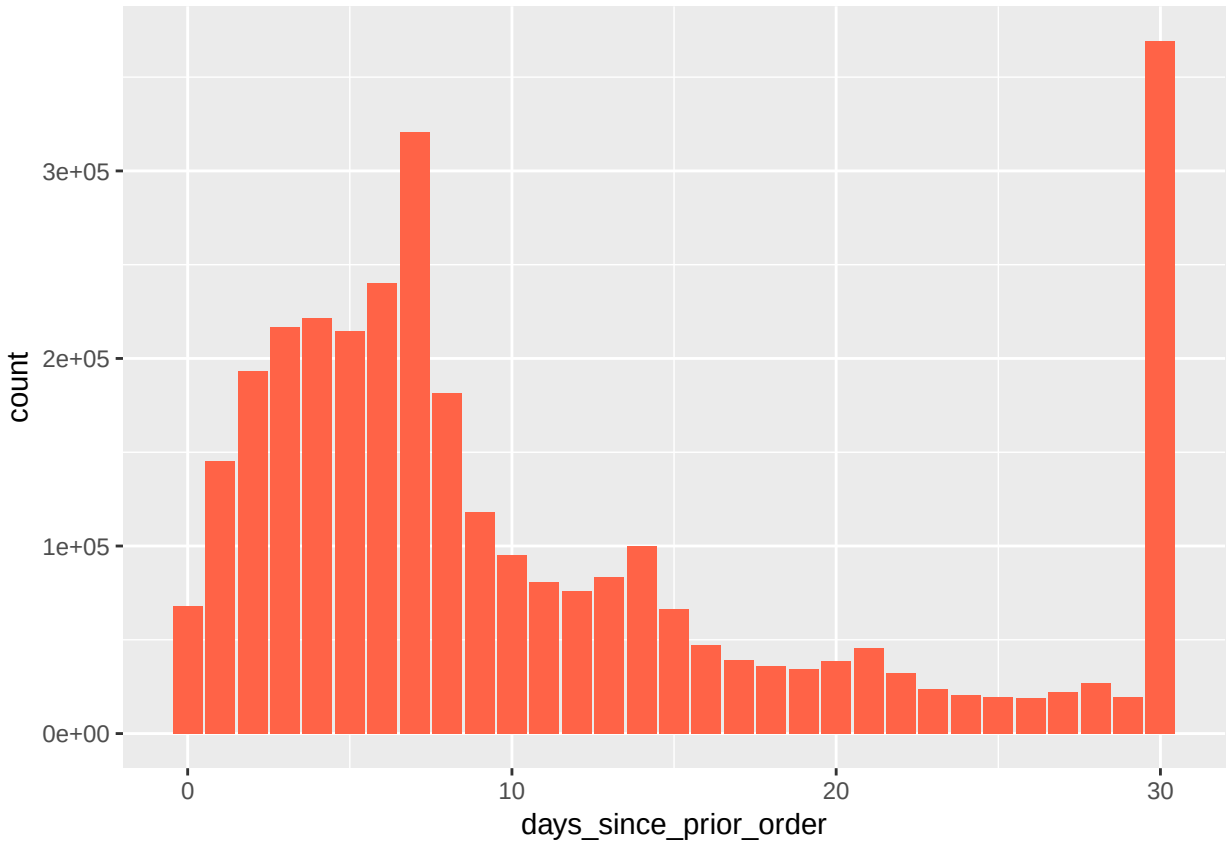
```
disappointing2 <-
    "
        SELECT o.product_id
        ,  p.product_name
        , n_products
        FROM (
            SELECT product_id
            , COUNT(1) AS n_products
            , reordered
            FROM order_products__prior_tbl
            GROUP BY product_id, reordered
            ORDER BY n_products
        ) o
        LEFT JOIN(
            SELECT product_id
            ,   product_name
            FROM products_tbl) p
        ON p.product_id = o.product_id
        WHERE reordered = 1
        LIMIT 10"

q3 <- dbGetQuery(sc, disappointing2)

kable(q3)
```

product_id	product_name	n_products
46951	Sweet Traditions Fresh Cut Sweet Potatoes in Light Syrup	1
36382	Fish Oil With Vitamin D Softgels	1
15870	Steamfresh Whole Green Beans	1
36169	The Itch Eraser	1
42065	Water Enhancer, Stevia, Raspberry Lemonade	1
236	Chicken Meatballs Dog Treats	1
37415	Shine Moroccan Sleek Oil Treatment For Frizzy, Dry Hair	1
47567	Large Cut Caesar Croutons	1
2682	Rosy Lip Therapy	1
34332	Complete Omega 3 6 9 D Lemon	1

Finally, another interesting point would be when the customers order again. Which is the periodicity of each customer, which are the customer habbits, that is, whether they are used to buy at the same hours and day.

For instance if we plot an histogram of number of orders in the days since prior order,

```
orders %>%
  collect %>%
  ggplot(aes(x=days_since_prior_order)) +
  geom_histogram(stat="count",fill="tomato")
```



we see that people seem to order more often after 1 week and one month.

We have posed some other questions in class that I left without answering. These are some of them:

- Which are most often reordered products? Which products have the highest probability of being reordered?
- The customer that comes the highest number of times. (recomend product, replace products)
- Peridoicity of products.
- Product bought together.
- Dependence of orders of buying of product with reordering it.
- Segments of people that buy on the same period of time.

What features we can add to the application

- Recomend new product.
- Suggest product that is very likely to be bought at the moment that person uses app.
- Offers for packages of products.
- When to buy in order to have faster shopping.
- Recomend to not to buy a product.

Finally we disconnect *Spark*,

```
spark_disconnect(sc)
```