Final Project

Begona Dobon Berenguer

July 5, 2017

Contents

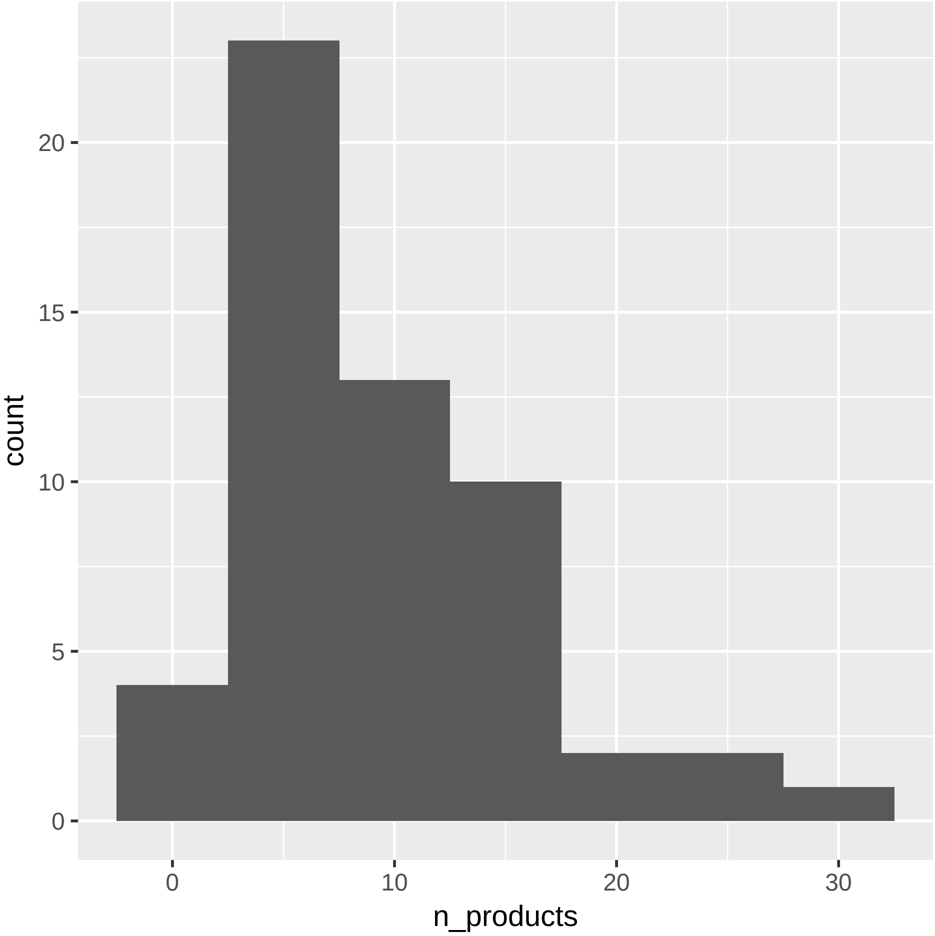
1 NOTE

As my laptop did not have enough memory to work on all the data, I created a subset of the data using only the first 100 lines for each CSV file. I do not know if the last section of the code does not work because it is wrong or because I am not working with all the data and the merging of the first 100 lines does not yeld a result.

2 Is the order of the products in a basket dependent of reordering?

2.1 What is the average number of products in an order?

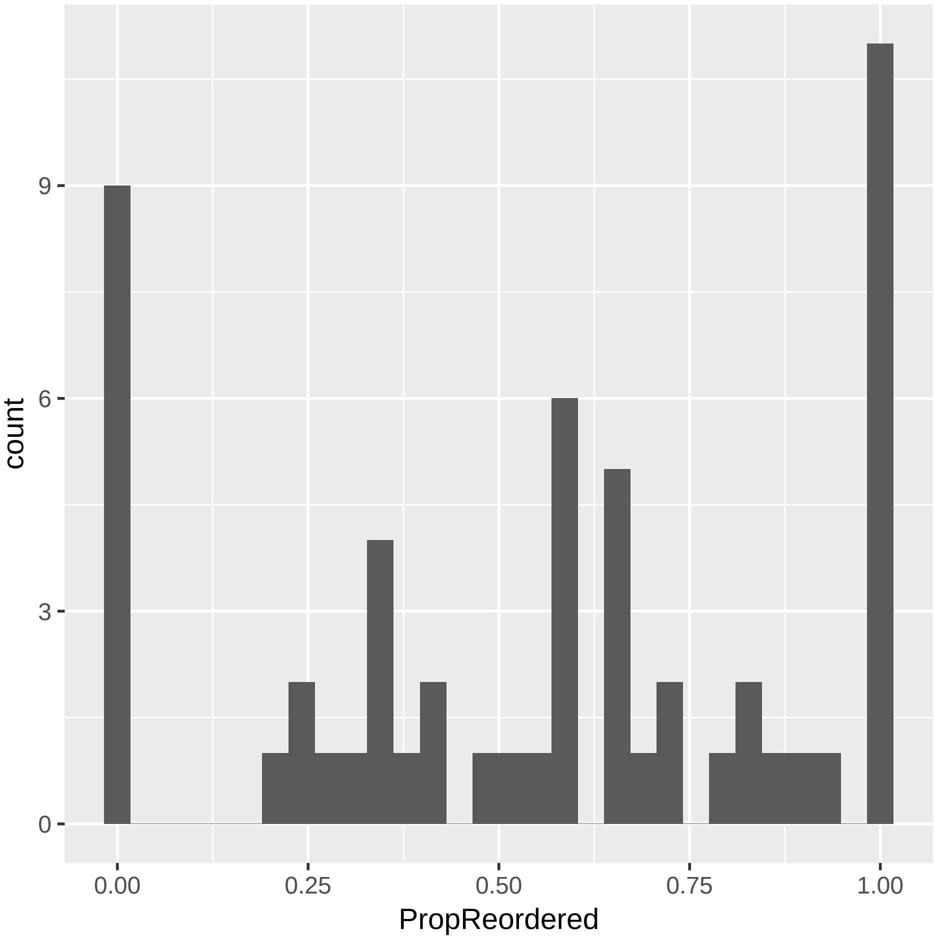
```r
number_products_order <-
  dbGetQuery(sc, "SELECT order_id,
                  COUNT(1) AS n_products
                  FROM order_products__prior_tbl
                  GROUP BY order_id")

number_products_order %>%
  ggplot(aes(n_products)) + geom_histogram(binwidth = 5)
```



An order has on average 9.07 products.

2.2 What proportion of products in a basket are reordered?

```
proportion_reordered_products <-
  dbGetQuery(sc, "SELECT order_id,
                  SUM(reordered) AS totalReordered,
                  MAX(add_to_cart_order) AS sizeBasket,
                  SUM(reordered)/ MAX(add_to_cart_order) AS PropReordered
                  FROM order_products__prior_tbl
                  GROUP BY order_id")

proportion_reordered_products %>%
  ggplot(aes(PropReordered)) + geom_histogram() +
    scale_y_continuous(label=scales::comma)
```

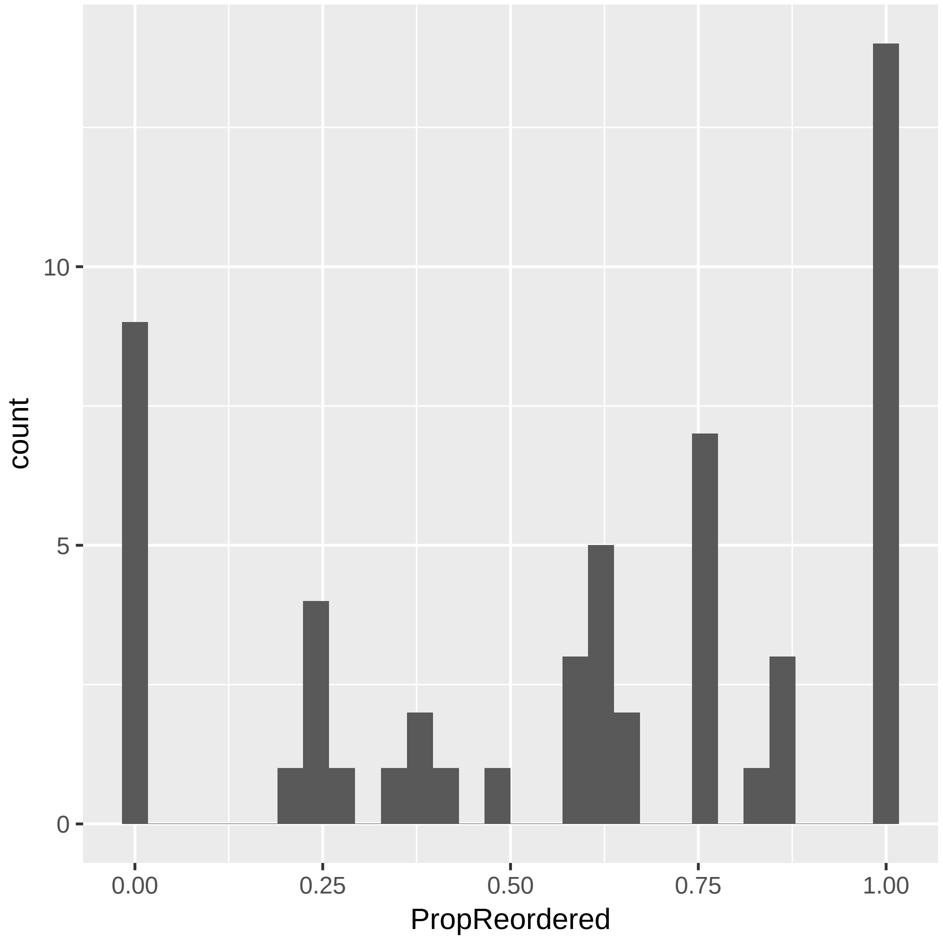


On average, 55.63 % of the products of a basket have been bougth previously.

An order has a median size of 8. I will use that value as a cutoff of the first items added to an order.

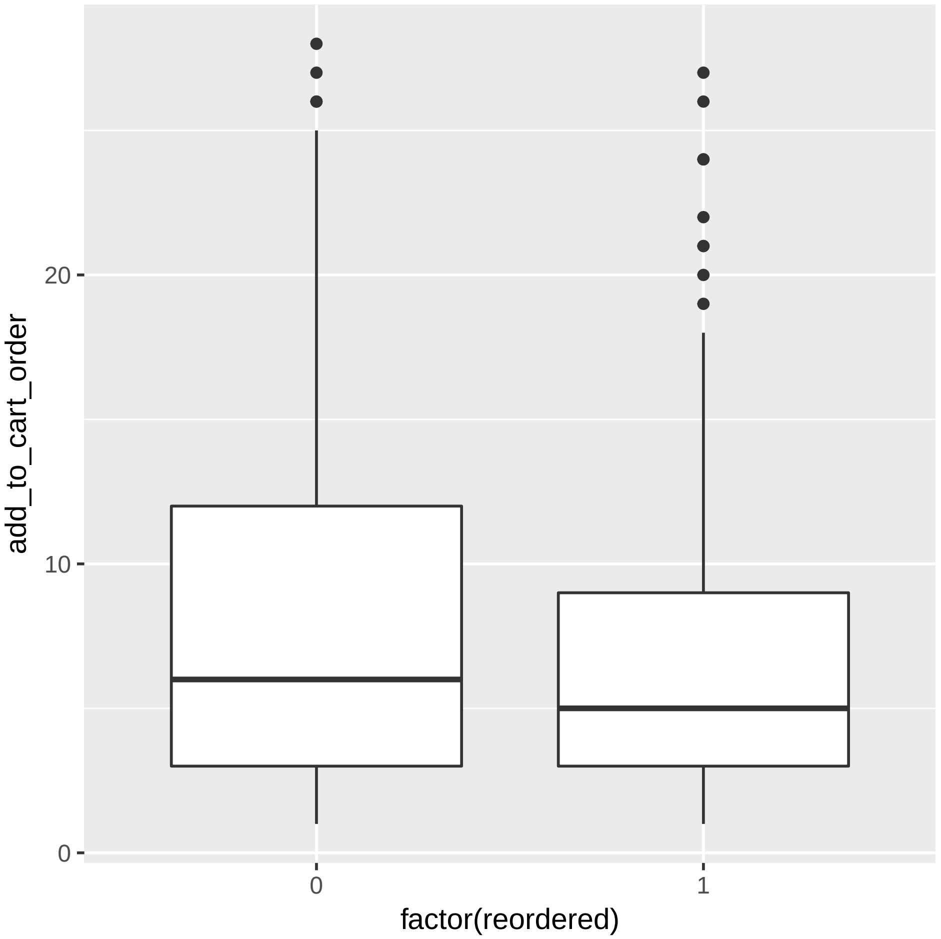
2.3 How many of the first 8 products in a basket are reordered?

```
proportion_FirstReordered_txt <-"SELECT order_id,
                  SUM(reordered) AS totalReordered,
                  MAX(add_to_cart_order) AS sizeBasket,
                  SUM(reordered)/ MAX(add_to_cart_order) AS PropReordered
                  FROM order_products__prior_tbl
                  WHERE add_to_cart_order <= {{cutoff}}
                  GROUP BY order_id"


data <- list(cutoff = median(proportion_reordered_products$sizeBasket))

proportion_FirstReordered_txt %>%
  whisker.render(data) %>%
  dbGetQuery(sc, .) %>%
  ggplot(aes(PropReordered)) + geom_histogram() +
    scale_y_continuous(label=scales::comma)
```



2.4 Do reordered products tend to be added first in the baskets?

```
position_reordered_products <-
  dbGetQuery(sc, "SELECT add_to_cart_order, reordered
                  FROM order_products__prior_tbl")

position_reordered_products %>%
  ggplot(aes(factor(reordered), add_to_cart_order)) + geom_boxplot()
```

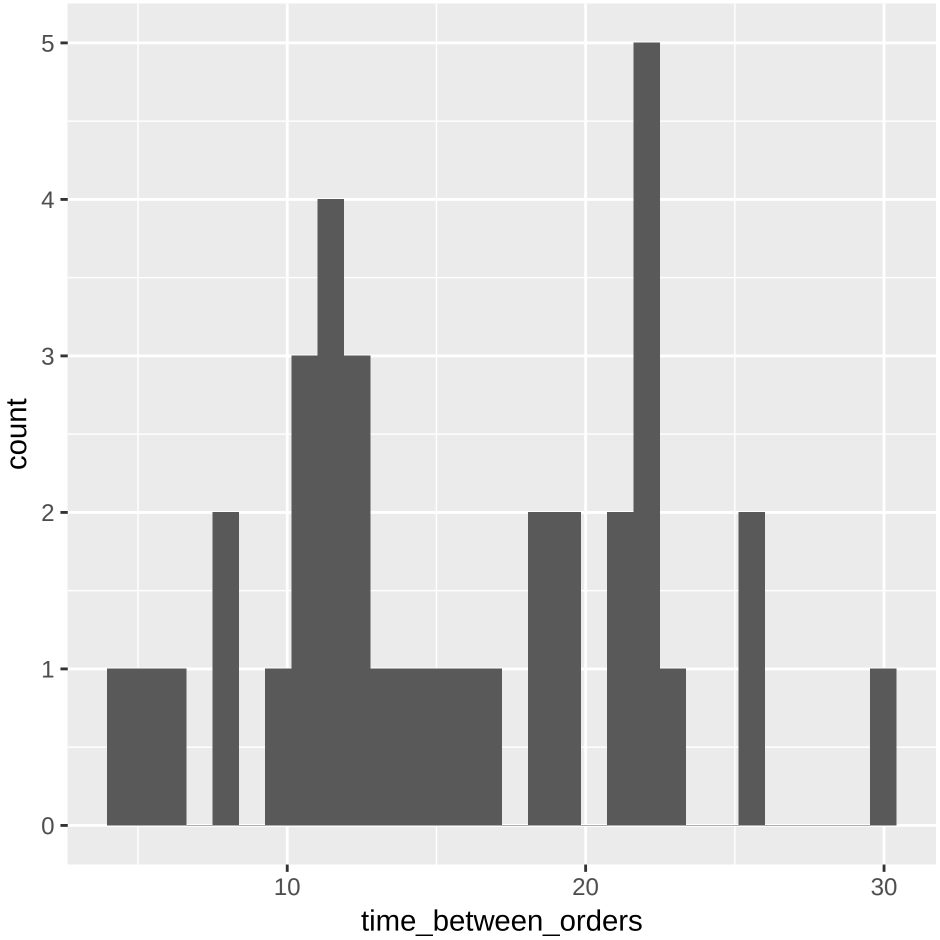


Reordered products seem to be added first in the orders.

3 Time between orders

3.1 How many days happened between orders?

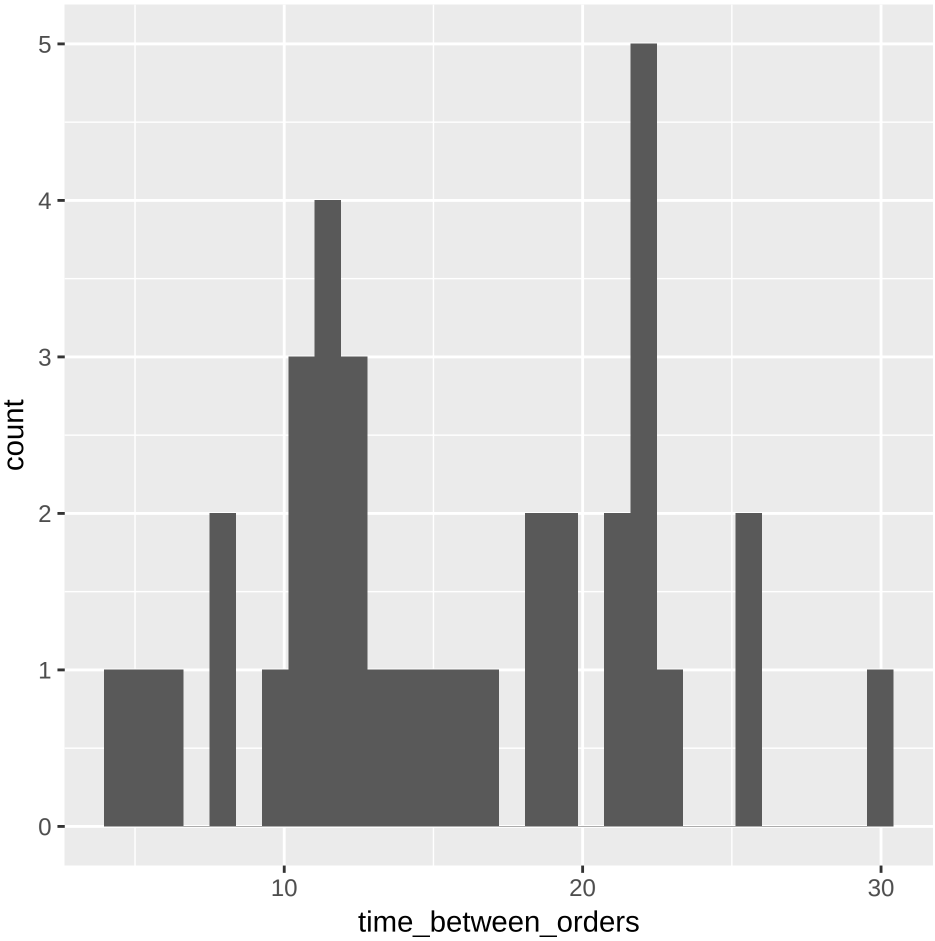
```
days_between_orders <-
  dbGetQuery(sc, "SELECT user_id,
                COUNT(1) AS n_orders,
                AVG(days_since_prior_order) AS time_between_orders
                FROM orders_tbl
                GROUP BY user_id
                ORDER BY user_id DESC")

days_between_orders %>%
  ggplot(aes(time_between_orders)) + geom_histogram() +
    scale_y_continuous(label=scales::comma)
```



On average, a user makes an order every 16 days.

Taking into account only users that have used the app at least twice, to avoid one-time users.

```
days_between_orders %>%
    filter(n_orders >=2) %>%
    ggplot(aes(time_between_orders)) + geom_histogram() +
    scale_y_continuous(label=scales::comma)
```

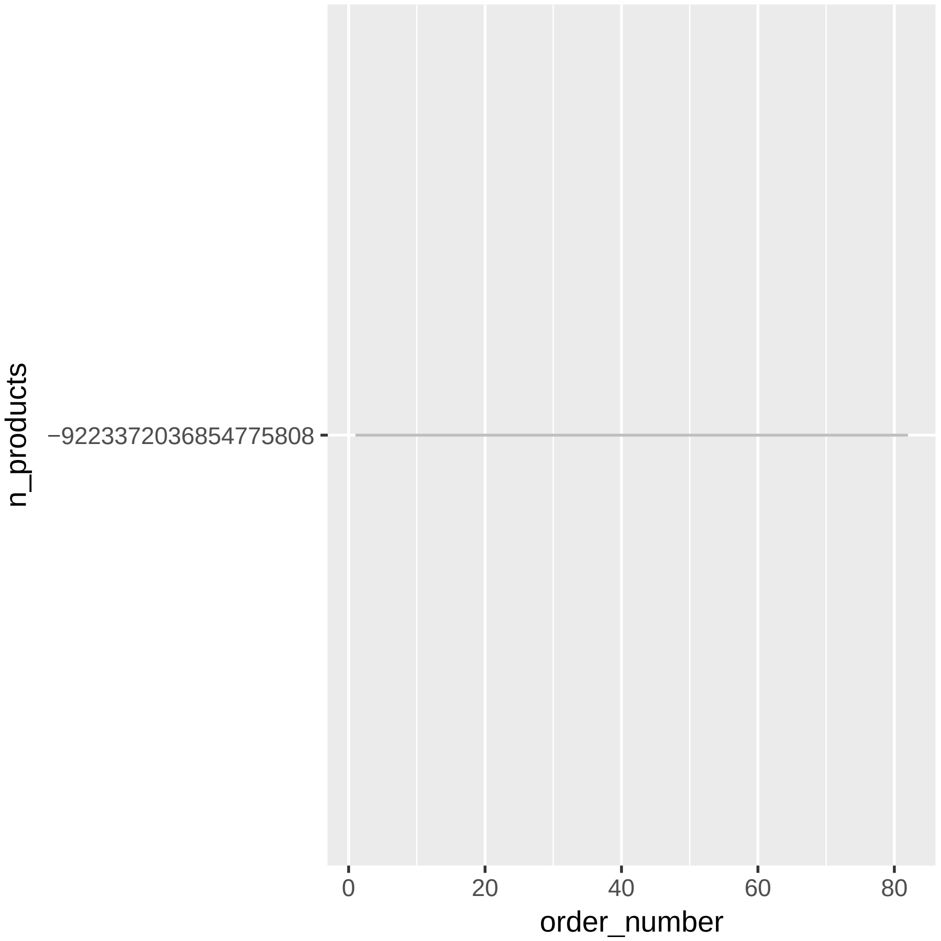


3.2 Do users tend to buy more, less or the same over time?

NOTE: I can not make this work

```
size_basket_over_time_txt <-
"
SELECT user_id, op.order_id, order_number,n_products
FROM (SELECT user_id, order_id, order_number
FROM orders_tbl) op
LEFT JOIN (
SELECT order_id,
COUNT(order_id) AS n_products
FROM order_products__prior_tbl
GROUP BY order_id ) p
ON op.order_id = p.order_id
"
size_basket_over_time <- dbGetQuery(sc,size_basket_over_time_txt)

ggplot(size_basket_over_time, aes(x=order_number, y=n_products),
       group=user_id, col=user_id)+geom_line(color="grey")
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
```


n_products
−9223372036854775808
0
20
40
60
80
order_number