# Shopify Data Science Intern Challenge

Siyi Wei

27/04/2021

## Contents

# Fall 2021 Data Science Intern Challenge

This is an intern challenge project for Shopify Data Science Intern. I hope my approach could help those who need it.

## Exploratory Data Analysis

We first need to check the dataset that matches the meta data description. There are 100 unique sneaker shops and the time window is from 2017-03-01 to 2017-03-30. The naive AOV is calculated by the mean of order amount, which is 3145.138. This number is nodoubtly too large for sneaker stores. We could assume there are outliers or influential points exist in the dataset.

```
## [1] "There are 100 unique shops in dataset"
```

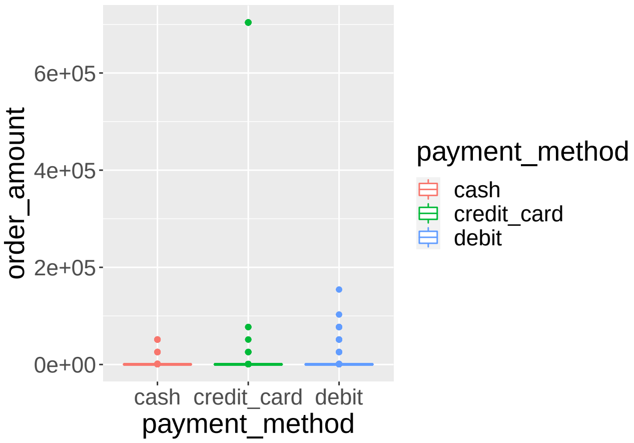
```
## [1] "The time window is from 2017-03-01 to 2017-03-30"
```

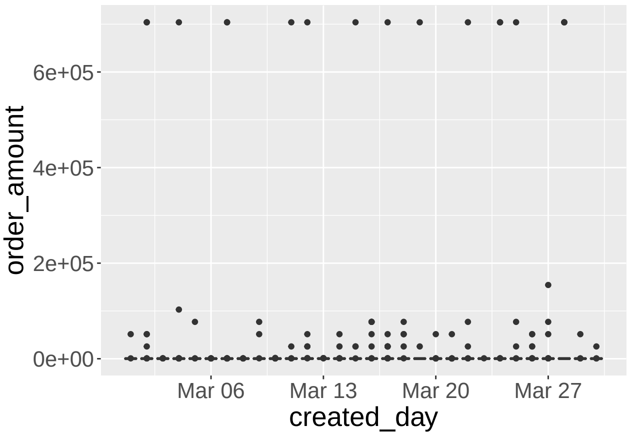
```
## [1] "The naive AOV which calculated by the average of order amount is 3145.128"
```

Next, our parameter of interest would be the order_amount. We want to explore the correlation between the parameter of interest and other covariates to find anything that is unusual.

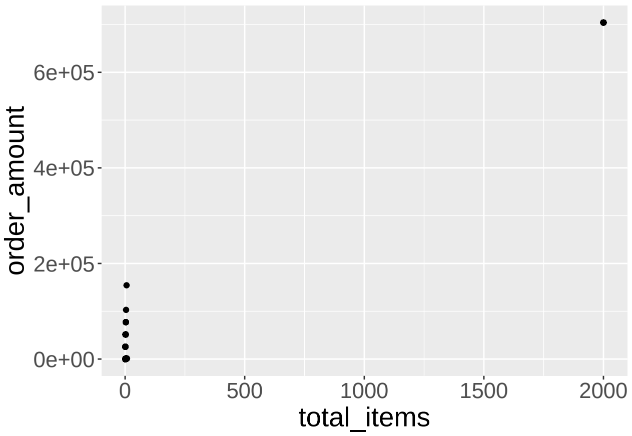
For the relations between order_amount and payment method. We could see there are outliers for all payment methods. But there is a significantly shifted outlier in credit_card.

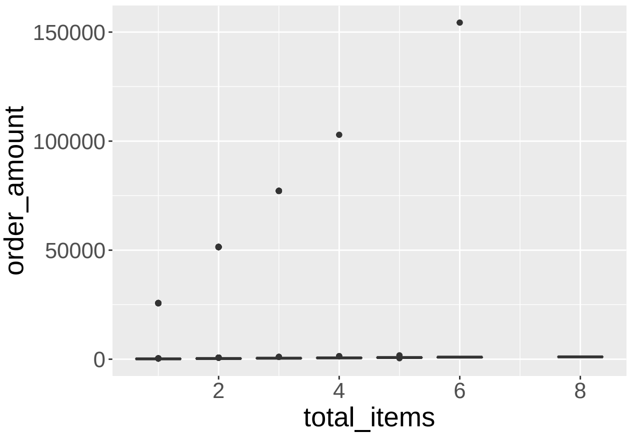
For the relations between order_amount and the time window. We could see most of the outliers are randomly distributed. However for one outlier, which is also a significantly shifted ourlier, there is a trivial pattern. The order with the same amount happened every 2-4 days. We need to keep this pattern in mind, it could solidify our assumption that this is a wholesale store later.





From the previous two relations we could determine there is one huge outlier in our dataset. From the left plot below we could conclude this outlier is due to the huge amount of items per order. However, we also observe few outliers even with little amount of items. By removing the most shifted outliers on the right plot. We could see there are some stores selling expensive sneakers, with much higher per-item prices. We assume they may be the luxury stores.





## Report metrics for different subgroups

From the previous analysis we could assume there are three different types of orders in our datasets. The normal orders (with decent price and amount of items), the orders with a huge amount of items (descent price) and the orders with much higher price from luxury stores. Since the last two groups have much more order value than the normal orders. There would be some potential problems if we calculate AOV across those three groups. However, those comments are still assumptions since those abnormal orders could also be fraudulent orders.

To solve this issue we could use two approaches. If we still want to report AOV. We could report the AOV for three different subgroups, the normal sneaker store, the luxury sneaker store and the wholesale store with a huge amount of items per order. We could also report the per-item value for those three different subgroups. Which gave us a better vision of their price strategy.

| | Normal Store | Wholesale Store | Luxury Store |
|---|---|---|---|
| AOV | 302.58 | 704000 | 49213.04 |

| | Normal Store | Wholesale Store | Luxury Store |
|---|---|---|---|
| item price | 151.78 | 352 | 25725 |

```
## [1] "The Normal Store has the AOV to be 302.581"
```

```
## [1] "The Wholesale Store has the AOV to be 704000"
```

```
## [1] "The Luxury Store has the AOV price to be 49213.043"
```

```
## [1] "The Normal Store has the per-item price to be 151.789"
```

```
## [1] "The Wholesale Store has the per-item price to be 352"
```

```
## [1] "The Luxury Store has the per-item price to be 25725"
```

# SQL Questions

## Answers

```
How many orders were shipped by Speedy Express in total?
Results: 54
Query:
SELECT count(*) as count
FROM Shippers as s JOIN Orders as o
ON s.ShipperID = o.ShipperID
WHERE s.ShipperName = "Speedy Express"

What is the last name of the employee with the most orders?
Results: Peacock
Query:
SELECT count(e.EmployeeID) as count, e.EmployeeID as ID, e.LastName
FROM Employees as e JOIN Orders as o
ON e.EmployeeID = o.EmployeeID
GROUP BY e.EmployeeID
```

```
ORDER BY count(e.EmployeeID) DESC
LIMIT 1

What product was ordered the most by customers in Germany?
Results: Boston Crab Meat
Query:
SELECT p.ProductName as name, SUM(od.Quantity) as quantity
FROM Customers as c JOIN Orders as o JOIN OrderDetails as od JOIN Products as p
ON c.CustomerID = o.CustomerID and o.OrderID = od.OrderID
and od.ProductID = p.ProductID
WHERE c.Country = "Germany"
GROUP BY p.ProductName
ORDER BY SUM(od.Quantity) DESC
LIMIT 1
```

# Code Appendix in R

```r
shop_data <- read.csv("./Intern_dataset.csv")
shop_data$created_day <- as.Date(shop_data$created_at,
    formax = "%Y-%m-%d")
shop_data$payment_method <- as.factor(shop_data$payment_method)


paste0("There are ", length(unique(shop_data$shop_id)),
    " unique shops in dataset")
paste0("The time window is from ", min(shop_data$created_day),
    " to ", max(shop_data$created_day))
paste0("The naive AOV which calculated by the average of order amount is ",
    mean(shop_data$order_amount))
```

```r
ggplot(shop_data, aes(x = payment_method, y = order_amount,
    colour = payment_method)) + geom_boxplot() + theme(text = element_text(size = 20))
ggplot(shop_data, aes(x = created_day, y = order_amount,
    group = created_day)) + geom_boxplot() + theme(text = element_text(size = 20))
```

```r
ro_data <- shop_data[shop_data$total_items != 2000,
    ]
ggplot(data = shop_data, aes(x = total_items, y = order_amount)) +
    geom_point() + theme(text = element_text(size = 20))

ggplot(data = ro_data, aes(x = total_items, y = order_amount,
    group = total_items)) + geom_boxplot() + theme(text = element_text(size = 20))
```

```
ns_price <- shop_data %>%
    mutate(item_price = order_amount/total_items) %>%
    filter(item_price < 1000)

ls_price <- shop_data %>%
    mutate(item_price = order_amount/total_items) %>%
    filter(item_price >= 1000)

ns_price_less_amount <- ns_price %>%
    filter(total_items < 2000)

ns_price_more_amount <- ns_price %>%
    filter(total_items >= 2000)


paste0("The Normal Store has the AOV to be ", round(mean(ns_price_less_amount$order_amou
    3))
paste0("The Wholesale Store has the AOV to be ", mean(ns_price_more_amount$order_amount)
paste0("The Luxury Store has the AOV price to be ",
    round(mean(ls_price$order_amount), 3))


paste0("The Normal Store has the per-item price to be ",
    round(mean(ns_price_less_amount$item_price), 3))
paste0("The Wholesale Store has the per-item price to be ",
    mean(ns_price_more_amount$item_price))
paste0("The Luxury Store has the per-item price to be ",
    mean(ls_price$item_price))
```