# Shopify Intern

Junwei Hu

5/13/2022

## Question 1

### When I first looked at the AOV, it's too large so I thought it might be miscalculated, or there might be some missing values or wrong values in the data. Then I looked at the dataset and did some saummary and calculation

```
data <- read.csv("2019 Winter Data Science Intern Challenge Data Set - Sheet1.csv")
head(data)
```

```
##   order_id shop_id user_id order_amount total_items payment_method
## 1        1      53     746          224           2           cash
## 2        2      92     925           90           1           cash
## 3        3      44     861          144           1           cash
## 4        4      18     935          156           1    credit_card
## 5        5      18     883          156           1    credit_card
## 6        6      58     882          138           1    credit_card
##            created_at
## 1 2017-03-13 12:36:56
## 2 2017-03-03 17:38:52
## 3  2017-03-14 4:23:56
## 4 2017-03-26 12:43:37
## 5  2017-03-01 4:35:11
## 6 2017-03-14 15:25:01
```

```
attach(data)
```

### Recalculate the AOV by diving the total amount by the total items

```
total_amount = sum(data$order_amount)
total_item = sum(data$total_items)

new_AOV = total_amount / total_item
new_AOV
```

```
## [1] 357.9215
```

## $357.92 sounds much better than $3145.13 for AOV, but it is still kind high for "relatively affordable sneakers"

## Next I will dig into each shop to see if there is any wrong values

```
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

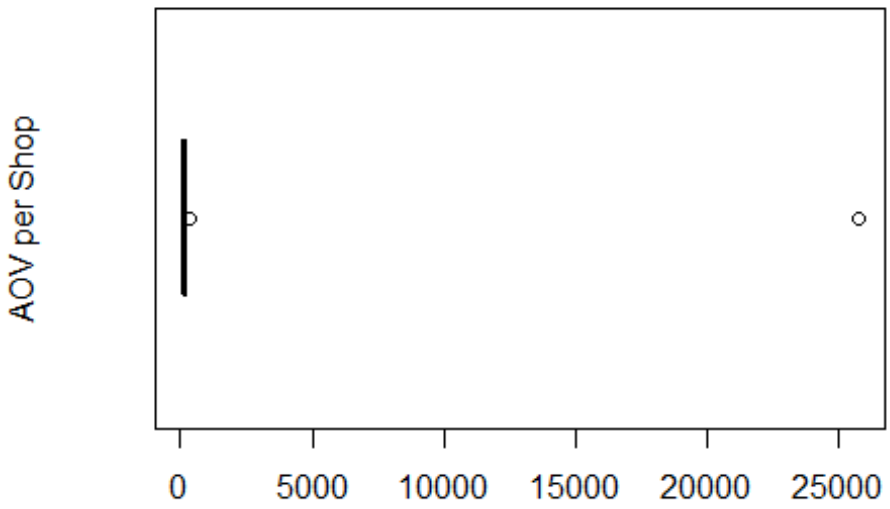
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
shop_average <- data %>%
  group_by(shop_id) %>%
  summarise(total_amount = sum(order_amount),
            total_item = sum(total_items)) %>%
  transmute(shop_id = shop_id,
            shop_average = total_amount/total_item) %>%
  arrange(desc(shop_average))

shop_average
```

```
## # A tibble: 100 x 2
##    shop_id shop_average
##      <int>        <dbl>
##  1      78        25725
##  2      42          352
##  3      12          201
##  4      89          196
##  5      99          195
##  6      50          193
##  7      38          190
##  8       6          187
##  9      51          187
## 10      11          184
## # ... with 90 more rows
```

```
boxplot(shop_average$shop_average, ylab = "AOV per Shop",
        horizontal = TRUE)
```




**So from the table and the plot above, there are two outliers, shop 78 with AOV 25725 and shop 42 with AOV 352**

**But I feel like 352 could be reasonable so there must be something wrong with shop 78**

**Then exclude shop 78 and calculate the AOV again**

```
mean(shop_average[shop_average$shop_id != 78,]$shop_average)

## [1] 152.2626

median(shop_average$shop_average)

## [1] 153

tab <- table(shop_average$shop_average)
names(tab[tab == max(tab)])

## [1] "153"
```

**I calculated the new AOV after removing shop 78, which is $152.26**

**But before we figure out what happeed to shop 78, it is not appropriate to just remove shop 78**

**So I calculate the median value of the AOV, which is $153**

**And also the mode value, the most common AOV, which is also $153**

**I would report mode value of AOV since it is the most common value of all the shops**

**And the value is $153**

## Question 2

(1)

```
SELECT
COUNT(ShipperID)
FROM Orders
WHERE ShipperID == 1
```

**54** orders were shipped by Speedy Express

(2)

```
SELECT e.LastName
From Employees AS e
WHERE (
SELECT o.EmployeeID
FROM Orders AS o
GROUP BY o.EmployeeID
ORDER BY COUNT(o.EmployeeID) DESC
LIMIT 1) == e.EmployeeID
```

**Peacock** is the last name of the employee with the most orders

(3)

```
FROM Customers AS c, OrderDetails AS od, Orders AS O, Products AS p
WHERE c.Country == "Germany" AND c.CustomerID == o.CUstomerID AND
                o.OrderID == od.OrderID AND od.ProductID == p.ProductID
GROUP BY p.ProductID
Order By SUM(Quantity) desc
```

**Boston Crab Meat** was ordered the most by customers in Germany