Shopify Data Science Challenge

Yu Jin Lee

9/3/2020

Question 1

In this part of the challenge, we are interested in understanding how much each customer spends per order, or average order value. I will perform an exploratory data analysis to understand the data first, consider the given AOV of $3145.13, and brainstorm a potentially more appropriate metric of sales per order.

Understanding the data

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 5000 obs. of  7 variables:
##  $ order_id      : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ shop_id       : num  53 92 44 18 18 58 87 22 64 52 ...
##  $ user_id       : num  746 925 861 935 883 882 915 761 914 788 ...
##  $ order_amount  : num  224 90 144 156 156 138 149 292 266 146 ...
##  $ total_items   : num  2 1 1 1 1 1 1 2 2 1 ...
##  $ payment_method: chr  "cash" "cash" "cash" "credit_card" ...
##  $ created_at    : chr  "2017-03-13 12:36:56" "2017-03-03 17:38:52" "2017-03-14 4:23:56" "2017-03-26
##  - attr(*, "spec")=
##   .. cols(
##   ..   order_id = col_double(),
##   ..   shop_id = col_double(),
##   ..   user_id = col_double(),
##   ..   order_amount = col_double(),
##   ..   total_items = col_double(),
##   ..   payment_method = col_character(),
##   ..   created_at = col_character()
##   .. )
```

I see that there are 5000 cases and 7 variables: order id, shop id, user id, order amount, total items, payment method, and order timestamp.

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

##     order_id       shop_id          user_id       order_amount
##  Min.   :   1   Min.   :  1.00   Min.   :607.0   Min.   :    90
##  1st Qu.:1251   1st Qu.: 24.00   1st Qu.:775.0   1st Qu.:   163
##  Median :2500   Median : 50.00   Median :849.0   Median :   284
##  Mean   :2500   Mean   : 50.08   Mean   :849.1   Mean   :  3145
##  3rd Qu.:3750   3rd Qu.: 75.00   3rd Qu.:925.0   3rd Qu.:   390
##  Max.   :5000   Max.   :100.00   Max.   :999.0   Max.   :704000
##   total_items          payment_method   created_at
```

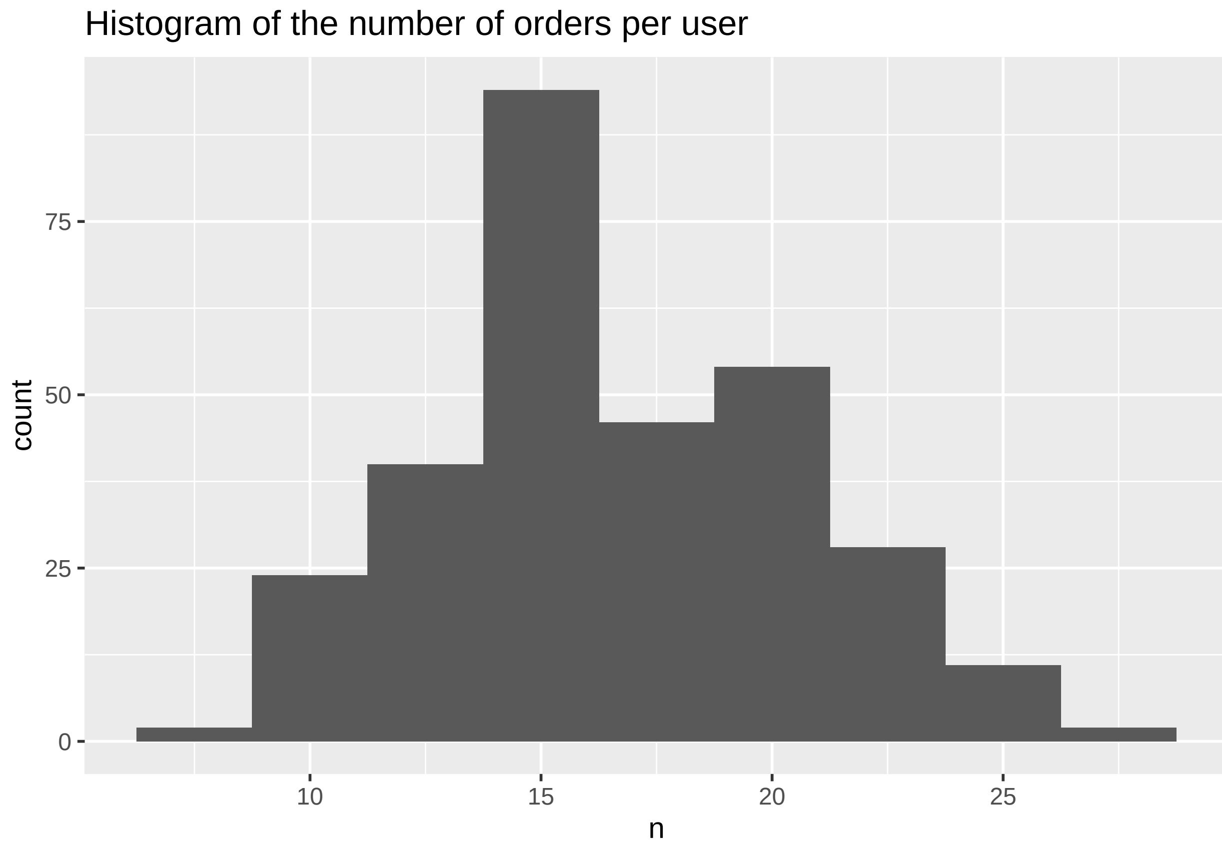
```
##  Min.   :   1.000   cash        :1594   Min.   :2017-03-01 00:08:09
##  1st Qu.:   1.000   credit_card:1735   1st Qu.:2017-03-08 07:08:03
##  Median :   2.000   debit       :1671   Median :2017-03-16 00:21:20
##  Mean   :   8.787                       Mean   :2017-03-15 22:20:37
##  3rd Qu.:   3.000                       3rd Qu.:2017-03-23 10:39:58
##  Max.   :2000.000                       Max.   :2017-03-30 23:55:35
```

As given, there are 100 sneaker shops, and we are given orders that were made in the month of March in 2017. There are $999 - 607 + 1 = 398$ unique customers, and the amount of order ranges from \$90 to \$704000. (I assume that units dollars.) The ranges for order amount and total items are quite large. However, the interquartile range for each variable is relatively narrow, and the mean and median are pretty different, which suggest that there may be outliers on the excessive side because mean is more sensitive to outliers than median. Furthermore, I am worried about potential error in data entry because of maximum values that are much larger than the rest of the data for order amount and total items.

Understanding the dimension of the data

In this section, if I had more time, I would dive deep into the data to uncover patterns in orders, such as prevalence of repeat customer, individual v. corporate customer, retail v. wholesale shops, popular payment methods, and average cost of sneakers. However, for the sake of focusing on the question, I will keep this semi-digression short and sweet. In particular, I want to explore the behavior of super loyal customers, whose orders drive up the mean values for order amount and total items.

```r
sneaker_shop %>%
  count(user_id) %>%
  ggplot() +
  geom_histogram(aes(x=n), binwidth = 2.5)+
  ggtitle("Histogram of the number of orders per user")
```



We see that most customers have made at least 10 orders, which would be somewhat unusual if the customers were individuals. I am now inclined to believe that most customers are retail shops placing bulk orders of popular products to stock their inventory.

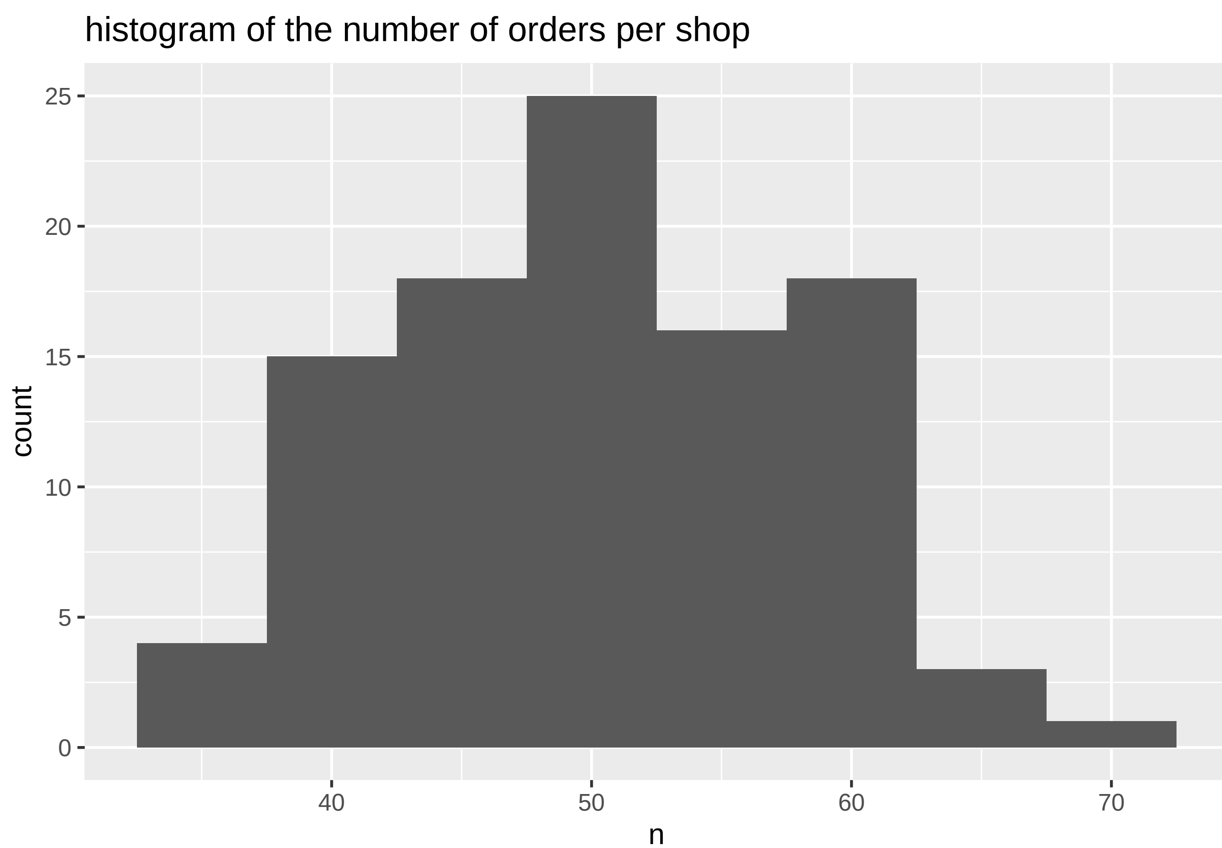
```
potential_retails <- sneaker_shop %>%
  count(user_id) %>%
  filter(n > 10)

sneaker_shop %>%
  filter(user_id %in% potential_retails$user_id) %>%
  count(user_id, shop_id) %>%
  arrange(desc(n))
```

```
## # A tibble: 4,484 x 3
##    user_id shop_id     n
##      <dbl>   <dbl> <int>
##  1     607      42    17
##  2     990      80     4
##  3     705      19     3
##  4     749      65     3
##  5     762      68     3
##  6     764      60     3
##  7     768      84     3
##  8     774      26     3
##  9     789      84     3
## 10     790       9     3
## # ... with 4,474 more rows
```

I see that most, if not all, of such customers whom I suspect to be retail shops have made at most 3 orders from a given shop. This observation adds weight to my hypothesis that these users are retail shops placing orders for their inventory of many kinds of sneakers. However, the user 607 noticeably has placed 17 orders from shop 42, which means that the user made an order from shop 42 every other day on average. I'm a bit suspicious about this behavior, but I will keep in mind and proceed.

Now I am interested in learning about the shops and their product, First I check the distribution of order amount and the price of sneakers sold at each shop.

```
sneaker_shop %>%
  mutate(avg_price = order_amount / total_items) %>%
  group_by(shop_id) %>%
  count(avg_price) %>%
  ggplot() +
  geom_histogram(aes(x=n), binwidth = 5) +
  ggtitle("histogram of the number of orders per shop")
```



The distribution of the number of orders made at each shop in one month is pretty standard - it's normal with a mean around 50. This suggests that there aren't any unusually popular or unpopular shops, and the sales across shops are comparable.

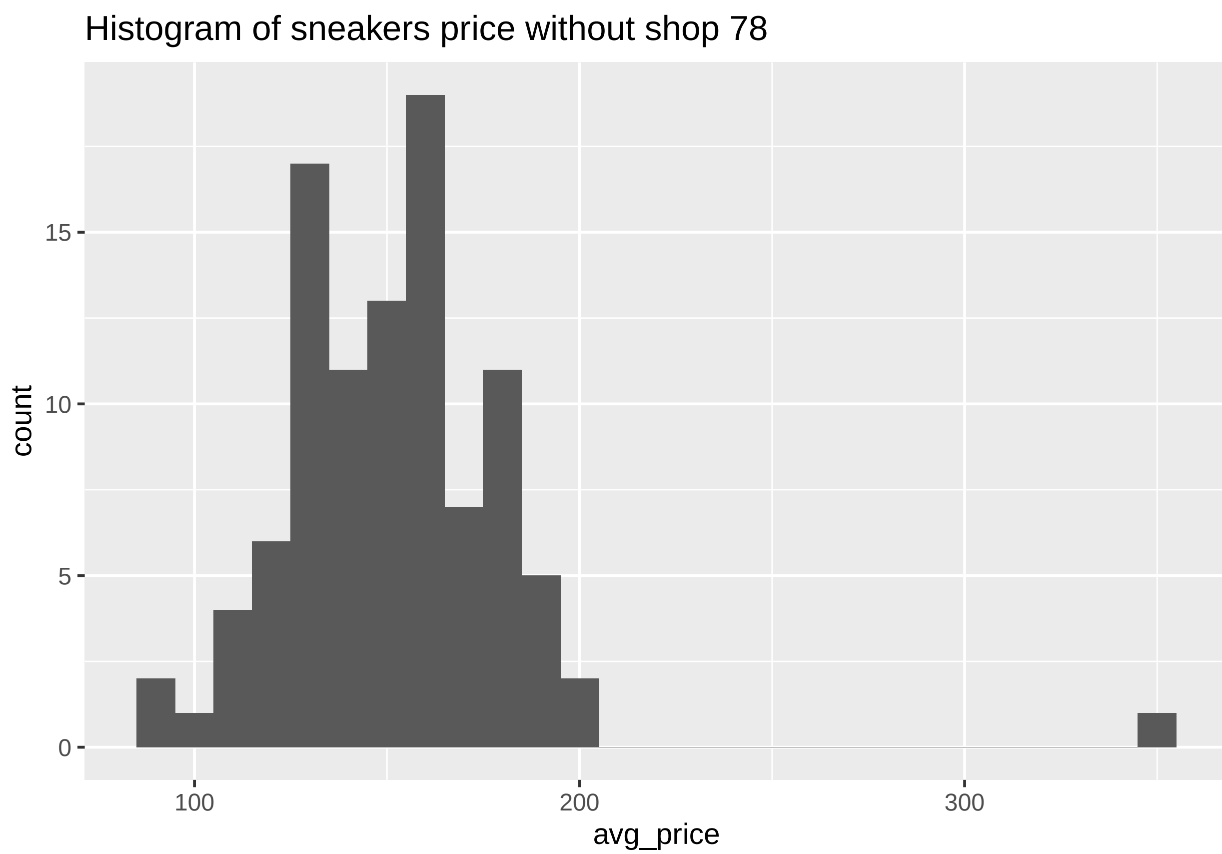
```
sneaker_shop %>%
  mutate(avg_price = order_amount / total_items) %>%
  group_by(shop_id) %>%
  count(avg_price) %>%
  arrange(desc(avg_price))
```

```
## # A tibble: 100 x 3
## # Groups:   shop_id [100]
```

```
##    shop_id avg_price     n
##      <dbl>     <dbl> <int>
##  1      78     25725    46
##  2      42       352    51
##  3      12       201    53
##  4      89       196    61
##  5      99       195    54
##  6      50       193    44
##  7      38       190    35
##  8       6       187    59
##  9      51       187    46
## 10      11       184    49
## # ... with 90 more rows
```

However, tbe comparison of the sneakers price shows that the shops 78 and 42 are pretty unusual with respect to their price: a pair of sneakers at the shop 78 costs almost 100 times the next expensive sneakers on the market! This number is too extreme and unreasonable that I flag the shop 78 as having had a data entry problem. Since less than 1 percent of the orders were made by shop 78, (46 out of 5000 orders), I choose to disregard this shop and proceed with analysis.

```
sneaker_shop %>%
  mutate(avg_price = order_amount / total_items) %>%
  group_by(shop_id) %>%
  count(avg_price) %>%
  filter(shop_id != 78) %>%
  ggplot() +
  geom_histogram(aes(x=avg_price), binwidth = 10) +
  ggtitle("Histogram of sneakers price without shop 78")
```



I see that even after excluding shop 78, the price of the second most expensive snekers is quite unusual compared to the price of rest of the shops. I also recall that this shopt selling the second most expensive pair of sneakers on the market had 17 orders from one customer in a month, which raised a red flag then. Nevertheless, given that 17 orders is within one magnitude of order volumes of other shops and that the price is on the same magnitude as others, I am not convinced that shop 42 is an outlier that should be excluded for potential data entry problem. Noting that mean is not robust against outliers, however, I am more inclined to evaluate this data using other metrics of average like mode or median.

Metric problem:

```
sneaker_shop %>%
summarize(aov = sum(order_amount)/5000)
```

```
## # A tibble: 1 x 1
##     aov
##   <dbl>
## 1 3145.
```

The the naive aov of \$3145.13 is calculated by diving the toal order amount across all shops over the total order volume (= 5000). As we have seen, however, this suffers from a few problems:

1. Shop 78 seems to have a data entry problem given its price that is more than 100 times the second most expensive pair of sneakers on the market.
2. Although we do not have strong evidence that shop 42 had a data entry problem, its price is pretty conspicuous compared to the rest of the shops. Such an outlier value will pull the mean higher.
3. The naive AOC glosses over the important details of the sales data such as the variation in the sneakers price and quantity per order.

The AOV is the most common metric to monitor the success of a business. Since our dataset consists of 100 shops, we are interested in understanding the sneakers market on shopify in general. To capture the idea of an "average order" on this market, I will report the median sneakers price across shops X the most common (mode) quantity of items per order, excluding the data from shop 78. Median is a good measure of average for the sneakers price because mean is not robust against outliers, and a given price is unlikely to be repeated. (It is unlikely that two different shops have the same price for their sneakers). Mode is an appropriate measure of average for the quantity per order because the quantity per order takes an integer value (and thus taking a mean should be avoided), and it meausres what's the most popular number of sneakers sold in each order. [On a side note, I believe it will be useful to check in with shops 78 and 42 to see whether there was a glitch in the system for certain orders. I only explore such errors only in so far as they help me think about the metric in this analysis]

```
sneaker_shop %>%
  filter(shop_id != 78) %>%
  count(total_items)
```

```
## # A tibble: 8 x 2
##   total_items     n
##         <dbl> <int>
## 1           1  1811
## 2           2  1816
## 3           3   932
## 4           4   292
## 5           5    77
## 6           6     8
## 7           8     1
## 8        2000    17
```

We note that the most common quantity per order is 2, closely folloewd by 1. While calculating the mode value for the number of orders, I notice that there are 17 orders with 2000 orders, which seems suspicious.

```
sneaker_shop %>%
  mutate(avg_price = order_amount / total_items) %>%
  filter(total_items == 2000) %>%
  arrange(created_at)
```

```
## # A tibble: 17 x 8
##    order_id shop_id user_id order_amount total_items payment_method
##       <dbl>   <dbl>   <dbl>        <dbl>       <dbl> <fct>
##  1      521      42     607       704000        2000 credit_card
##  2     4647      42     607       704000        2000 credit_card
##  3       61      42     607       704000        2000 credit_card
##  4       16      42     607       704000        2000 credit_card
##  5     2298      42     607       704000        2000 credit_card
##  6     1437      42     607       704000        2000 credit_card
##  7     2154      42     607       704000        2000 credit_card
##  8     1363      42     607       704000        2000 credit_card
##  9     1603      42     607       704000        2000 credit_card
## 10     1563      42     607       704000        2000 credit_card
## 11     4869      42     607       704000        2000 credit_card
## 12     1105      42     607       704000        2000 credit_card
## 13     3333      42     607       704000        2000 credit_card
## 14     4883      42     607       704000        2000 credit_card
## 15     2836      42     607       704000        2000 credit_card
## 16     2970      42     607       704000        2000 credit_card
## 17     4057      42     607       704000        2000 credit_card
```

```
## # ... with 2 more variables: created_at <dttm>, avg_price <dbl>
```

Indeed, we see that the all 17 orders of 2000 items were made at 4AM almost every other day by user 607 only at shop 42.

```
sneaker_shop %>%
  filter(user_id == 607)
```

```
## # A tibble: 17 x 7
##    order_id shop_id user_id order_amount total_items payment_method
##       <dbl>   <dbl>   <dbl>        <dbl>       <dbl> <fct>
##  1       16      42     607       704000        2000 credit_card
##  2       61      42     607       704000        2000 credit_card
##  3      521      42     607       704000        2000 credit_card
##  4     1105      42     607       704000        2000 credit_card
##  5     1363      42     607       704000        2000 credit_card
##  6     1437      42     607       704000        2000 credit_card
##  7     1563      42     607       704000        2000 credit_card
##  8     1603      42     607       704000        2000 credit_card
##  9     2154      42     607       704000        2000 credit_card
## 10     2298      42     607       704000        2000 credit_card
## 11     2836      42     607       704000        2000 credit_card
## 12     2970      42     607       704000        2000 credit_card
## 13     3333      42     607       704000        2000 credit_card
## 14     4057      42     607       704000        2000 credit_card
## 15     4647      42     607       704000        2000 credit_card
## 16     4869      42     607       704000        2000 credit_card
## 17     4883      42     607       704000        2000 credit_card
## # ... with 1 more variable: created_at <dttm>
```

We also see that user 607 exclusively only shopped at shop 42 in this unusual pattern. This does not reflect a normal purchasing behavior. Luckily, our metric of choice - mode- is robust against outliers as such.

```
sneakers_price <- sneaker_shop %>%
  filter(shop_id != 78) %>%
  mutate(avg_price = order_amount / total_items)
median(sneakers_price$avg_price)
```

```
## [1] 153
```

The median sneakers price is \$153, which seems reasonable.

Since the most common quantity per order is 2 and the median sneakers price is \$153, the average order value is $2 \times \$153 = \$206$.