Summer 2022 Data Science Intern Challenge

Bo Chen

Question 1: Given some sample data, write a program to answer the following: click here to access the required data set

On Shopify, we have exactly 100 sneaker shops, and each of these shops sells only one model of shoe. We want to do some analysis of the average order value (AOV). When we look at orders data over a 30 day window, we naively calculate an AOV of $3145.13. Given that we know these shops are selling sneakers, a relatively affordable item, something seems wrong with our analysis.

a) Think about what could be going wrong with our calculation. Think about a better way to evaluate this data.

```r
library(readr)
library(dplyr)
```

```
##
## 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
```

```r
Sample_data<-
  read_csv("~/Desktop/data intern/2019 Winter Data Science Intern Challenge Data Set - Sheet1.csv")
```

```
## Parsed with column specification:
## 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()
## )
```

```r
Sample_data
```

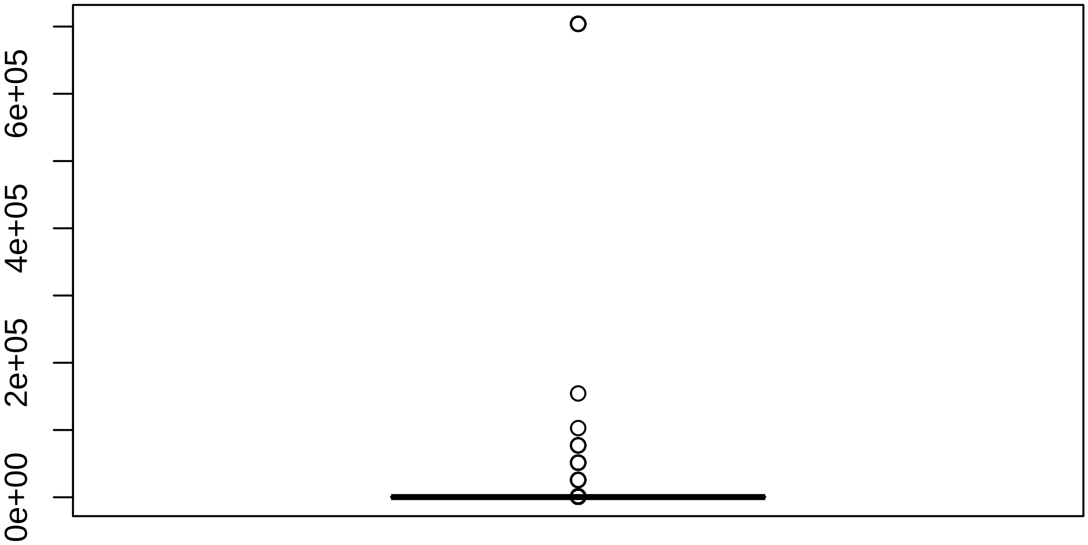
```
## # A tibble: 5,000 x 7
##    order_id shop_id user_id order_amount total_items payment_method created_at
##       <dbl>   <dbl>   <dbl>        <dbl>       <dbl> <chr>          <chr>
##  1        1      53     746          224           2 cash           2017-03-13 ~
##  2        2      92     925           90           1 cash           2017-03-03 ~
##  3        3      44     861          144           1 cash           2017-03-14 ~
##  4        4      18     935          156           1 credit_card    2017-03-26 ~
##  5        5      18     883          156           1 credit_card    2017-03-01 ~
##  6        6      58     882          138           1 credit_card    2017-03-14 ~
##  7        7      87     915          149           1 cash           2017-03-01 ~
##  8        8      22     761          292           2 cash           2017-03-08 ~
##  9        9      64     914          266           2 debit          2017-03-17 ~
## 10       10      52     788           146           1 credit_card    2017-03-30 ~
## # ... with 4,990 more rows
```

```
mean(Sample_data$order_amount)
```

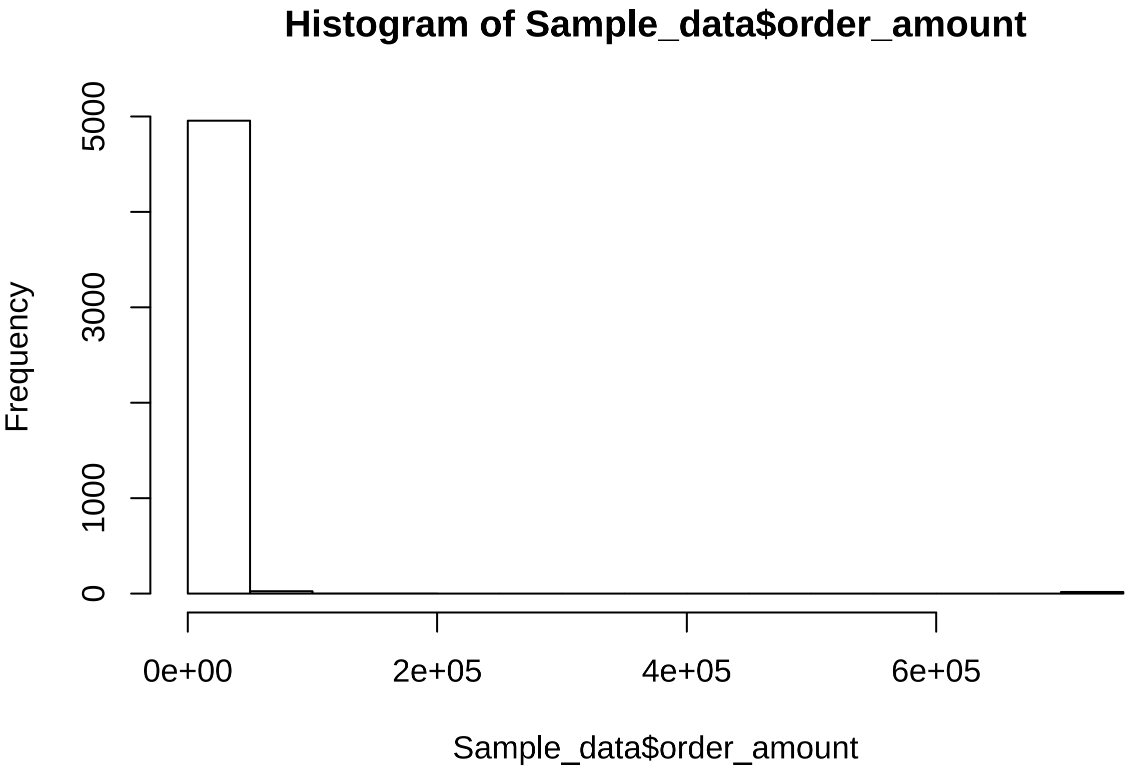
```
## [1] 3145.128
```

The naive calculation of an AOV is the crude overall mean value of the order amount.

```
boxplot(Sample_data$order_amount)
```



```
hist(Sample_data$order_amount)
```



As we can see from the boxplot and the histgram of the order_amount, there are a few extremely large amount order values (outliers), and the naive calcualation of AOV is strongly affected by these outliers. So we need to find a proper solution to minimize the effects of these values when we calculate the AOV.

Note: Based on the data, I believe there are three types of orders: regular retail orders(most of the orders), wholesale orders(outlier) and limited edition retail orders(outlier). Wholesale orders have larger order items and limited edition snearks have extremely high product prices comparing to regular retail orders.

b) What metric would you report for this dataset?

Before I consider the outliers, I would like to calculate the average order value for EACH SHOP.

```
AOV_EACH=Sample_data %>%
  group_by(shop_id) %>%
  summarise_at(vars(order_amount), list(AOV_each_shop = mean))
AOV_EACH$AOV_each_shop
```

```
##   [1]    308.8182    174.3273    305.2500    258.5098    290.3111    383.5085
##   [7]    218.0000    241.0435    234.0000    332.3019    356.7347    352.6981
##  [13]    345.3968    242.0000    308.9423    270.1463    332.0755    342.5882
##  [19]    320.9062    251.5577    308.6957    273.7500    317.6727    320.7273
##  [25]    232.9167    341.2245    334.8704    320.3721    331.6207    295.0714
##  [31]    268.9787    189.9762    376.2750    234.2400    328.0000    254.8000
##  [37]    340.2083    390.8571    268.0000    295.1667    254.0000 235101.4902
##  [43]    333.9138    262.1538    269.3103    347.4419    259.1489    242.7750
##  [49]    279.9057    403.5455    361.8043    316.9268    214.1176    276.6400
##  [55]    327.7500    218.1892    296.7736    254.9492    358.9667    350.2340
##  [61]    344.4400    308.8372    264.9655    272.1860    330.8148    312.8868
##  [67]    272.6216    254.6383    264.1833    343.0678    323.0303    309.5652
##  [73]    335.6897    306.0000    240.7619    321.0714    280.8000  49213.0435
##  [79]    328.4815    299.6667    384.0000    349.7857    248.7857    342.3051
##  [85]    329.2571    277.5000    292.2692    355.5200    379.1475    403.2245
##  [91]    325.9259    162.8571    214.4746    297.7778    318.7692    330.0000
##  [97]    324.0000    245.3621    339.4444    213.6750
```

I found that there are few extremely large AOV for some of shops. There are usually 2 ways to deal with the outliers when cleaning the data: Deleting or replacing. By simply removing the outlier, some information from the data is lost. By replacing, normally we use average or median to replace the outliers. Or we can use imputation method (consider outliers as random missing value) to replace the outliers. For this case, I believe to replace these large AOV values with the MEDIAN is suffcient enough to calculate the final AOV.

I used 1.5 IQR rule to find the oulier

```
median1=median(AOV_EACH$AOV_each_shop)
quantile(AOV_EACH$AOV_each_shop,0.75)+1.5*IQR(AOV_EACH$AOV_each_shop)
```

```
##      75%
## 446.0569
```

```
NEW_AOV_EACH=replace(AOV_EACH$AOV_each_shop , which(AOV_EACH$AOV_each_shop>446.0569) , median1)
mean(NEW_AOV_EACH)
```

```
## [1] 299.8665
```

c) What is its value?

My final AOV is 299.8665.

Question 2: For this question you'll need to use SQL. Follow this link to access the data set required for the challenge. Please use queries to answer the following questions. Paste your queries along with your final numerical answers below.

a) How many orders were shipped by Speedy Express in total?

Answer: 54

Query:

```
SELECT COUNT(DISTINCT a.OrderID) as total FROM [Orders] a
 LEFT JOIN [Shippers] b on a.ShipperID = b.ShipperID
 WHERE b.ShipperName='Speedy Express'
```

b) What is the last name of the employee with the most orders?

Answer: Peacock

Query:

```
Select b.LastName FROM
 (SELECT EmployeeID,COUNT(Distinct OrderID) as total FROM [Orders] GROUP BY EmployeeID) a
 LEFT JOIN [Employees] b on a.EmployeeID =b.EmployeeID ORDER BY a.total DESC LIMIT 1
```

c) What product was ordered the most by customers in Germany?

Answer: Boston Crab Meat

Query:

```
SELECT d.ProductName FROM [OrderDetails] a
 LEFT JOIN [Orders] b on a.OrderId = b.OrderID
 LEFT JOIN [Customers] c on b.CustomerID = c.CustomerID
 LEFT JOIN [Products] d on a.ProductId = d.ProductId
 WHERE c.Country='Germany' GROUP BY a.ProductID ORDER BY SUM(a.Quantity) DESC LIMIT 1
```