Shopify Summer 2022 Data Science Intern Challenge

Minyue Jin

1/19/2022

Short Answers

Question 1

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

The problem might be i) some shops are selling sneakers of unusually high prices; or/and ii) some customers made orders of unusually large quantities of sneakers.

A better way could be either simply deleting those unusual data, or modifying unusual data based on further information and experience. For this specific dataset, I find a shop (shop_id: 78) is selling sneakers at $25725/pair, which could be $257.25/pair in fact according to common life experience. Meanwhile, I find an user (user_id: 607) repeatedly made orders each of 2,000 pairs of sneakers at exactly 4 a.m. on several days. I cannot determine what was going on without more information.

b. What metric would you report for this dataset?

First, since I do not have further knowledge of those shops, I would like to just exclude those unusual data of shop 78 and user 607 when calculating metrics. A simple way to reduce the effect of outliers is to report the median, but since the origin purpose of this analysis was to calculate the AOV, I assume it would be better to stick to the average. Thus I would report the average of "AOVs for each store" for this dataset.

c. What is its value?

The value is 303.24.

Question 2

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

```
SELECT COUNT(*) FROM Orders
INNER JOIN Shippers ON Orders.ShipperID=Shippers.ShipperID
WHERE Shippers.ShipperName='Speedy Express'
GROUP BY Shippers.ShipperName;
```

Answer: 54

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

```
SELECT Top 1 Employees.LastName,COUNT(Orders.EmployeeID) AS Num FROM Employees
LEFT JOIN Orders ON Employees.EmployeeID=Orders.EmployeeID
GROUP BY Employees.LastName
ORDER BY COUNT(Orders.EmployeeID) DESC;
```

Answer: Peacock (40 orders)

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

```
SELECT ProductName FROM Products WHERE ProductID=(
    SELECT TOP 1 OrderDetails.ProductID
    FROM ((OrderDetails
    INNER JOIN Orders ON OrderDetails.OrderID=Orders.OrderID)
    INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID)
    WHERE Customers.Country='Germany'
    GROUP BY OrderDetails.ProductID
    ORDER BY SUM(OrderDetails.Quantity) DESC
);
```

Answer: Boston Crab Meat

Codes & Program for Question 1

Question 1.a

In this part, I took a glimpse into the distribution of order_amount, found those outliers, and made assumptions on what could be wrong.

```
summary(raw$order_amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      90     163     284    3145     390  704000
```

```
library(plyr)
head(arrange(raw,desc(order_amount)),100)
```

order_id <dbl>	shop_id <dbl>	user_id <dbl>	order_amount <dbl>	total_items <dbl>	payment_method <chr>	created_at <S3: POSIXct>
16	42	607	704000	2000	credit_card	2017-03-07 04:00:00
61	42	607	704000	2000	credit_card	2017-03-04 04:00:00
521	42	607	704000	2000	credit_card	2017-03-02 04:00:00
1105	42	607	704000	2000	credit_card	2017-03-24 04:00:00
1363	42	607	704000	2000	credit_card	2017-03-15 04:00:00
1437	42	607	704000	2000	credit_card	2017-03-11 04:00:00
1563	42	607	704000	2000	credit_card	2017-03-19 04:00:00
1603	42	607	704000	2000	credit_card	2017-03-17 04:00:00
2154	42	607	704000	2000	credit_card	2017-03-12 04:00:00
2298	42	607	704000	2000	credit_card	2017-03-07 04:00:00

1-10 of 100 rows Previous 1 2 3 4 5 6 ... 10 Next

Question 1.b&c

In this part, I eliminated the outliers, calculated the Average Order Values by store, and then obtained the average of these AOVs.

```
library(dplyr)
```

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

```
## The following objects are masked from 'package:plyr':
## 
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

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

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

```
filtered<-raw %>%
  filter(shop_id!=78,
         user_id!=607)
```

```
library(dplyr)

aovs_by_shop<-filtered %>%
  group_by(shop_id) %>%
  summarise(total_amount = sum(order_amount),
            num_orders = n()) %>%
  transmute(shop_id = shop_id,
            total_amount=total_amount,
            num_orders=num_orders,
            shop_aov = total_amount/num_orders)

new_aov=mean(aovs_by_shop$shop_aov)
new_aov
```

```
## [1] 303.2435
```

The new AOV value to be reported is 303.24.