Fall 2021 Data Science Intern Challenge

Question 1

First, I will put the data into a dataframe and determine the calculation that caused AOV to be $3145.13.

```
m <-  read.csv("C:/Users/DAVID/Documents/shopify_data_science/data.csv")
summary(m)
```

```
##     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   Length:5000        Length:5000       
##  1st Qu.:   1.000   Class :character   Class :character  
##  Median :   2.000   Mode  :character   Mode  :character  
##  Mean   :   8.787                                        
##  3rd Qu.:   3.000                                        
##  Max.   :2000.000                                        
```

```
n = length(m$order_id)
```

The means of the order_amount and total_items seem to be heavily skewed due to large outliers. The calculation of Average Order Value is calculated: $AOV = \frac{Revenue}{Number\ of\ Orders}$

```
revenue = sum(m$order_amount)
revenue
```

```
## [1] 15725640
```

```
aov = revenue / n
aov
```

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

So, the AOV was calculated summing up all the order_amounts and divided it by the total number of orders. But looking carefully at the data, we can see if there are any outliers. I wil check for any orders with a large quantity of total_items.

```r
length(m[m$total_items > 10, ]$order_id)
```

```
## [1] 17
```

```r
m[m$total_items > 10, ]
```

```
##      order_id shop_id user_id order_amount total_items payment_method
## 16         16      42     607       704000        2000    credit_card
## 61         61      42     607       704000        2000    credit_card
## 521       521      42     607       704000        2000    credit_card
## 1105     1105      42     607       704000        2000    credit_card
## 1363     1363      42     607       704000        2000    credit_card
## 1437     1437      42     607       704000        2000    credit_card
## 1563     1563      42     607       704000        2000    credit_card
## 1603     1603      42     607       704000        2000    credit_card
## 2154     2154      42     607       704000        2000    credit_card
## 2298     2298      42     607       704000        2000    credit_card
## 2836     2836      42     607       704000        2000    credit_card
## 2970     2970      42     607       704000        2000    credit_card
## 3333     3333      42     607       704000        2000    credit_card
## 4057     4057      42     607       704000        2000    credit_card
## 4647     4647      42     607       704000        2000    credit_card
## 4869     4869      42     607       704000        2000    credit_card
## 4883     4883      42     607       704000        2000    credit_card
##              created_at
## 16   2017-03-07 4:00:00
## 61   2017-03-04 4:00:00
## 521  2017-03-02 4:00:00
## 1105 2017-03-24 4:00:00
## 1363 2017-03-15 4:00:00
## 1437 2017-03-11 4:00:00
## 1563 2017-03-19 4:00:00
## 1603 2017-03-17 4:00:00
## 2154 2017-03-12 4:00:00
## 2298 2017-03-07 4:00:00
## 2836 2017-03-28 4:00:00
## 2970 2017-03-28 4:00:00
## 3333 2017-03-24 4:00:00
## 4057 2017-03-28 4:00:00
## 4647 2017-03-02 4:00:00
## 4869 2017-03-22 4:00:00
## 4883 2017-03-25 4:00:00
```

Looking at orders where the total_items are greater than 10, we see that there is a single user made mass orders of 2000 items from the same shop 17 times throughout the 30 days. This single user's orders are what is caused our naive calculation of AOV to be so high. We can filter this user's orders out and see if we get a more expected AOV.

```r
m2 = m[m$total_items < 10, ]
n2 = length(m2$order_id)
revenue2= sum(m2$order_amount)
revenue2
```

```
## [1] 3757640
```

```
aov2 = revenue2 / n2
aov2
```

```
## [1] 754.0919
```

After removing the outliers from the dataset, we calculate a much more expected AOV of $754.09. We can report this value with a note of the large orders placed by the individual.

Question 2

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

```
SELECT COUNT(*) FROM Shippers JOIN Orders ON Shippers.ShipperID = Orders.ShipperID
WHERE ShipperName = ’Speedy Express’;
```

Answer: 54

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

```
WITH f as
(WITH e as
(SELECT EmployeeID, COUNT(*) AS NumOfOrders FROM Orders GROUP BY EmployeeID)
SELECT EmployeeID, MAX(NumOfOrders) FROM e)
SELECT LastName FROM Employees JOIN f ON Employees.EmployeeID = f.EmployeeID;
```

Answer: Peacock

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

```
WITH f AS
(WITH e AS
(SELECT * FROM OrderDetails WHERE OrderID IN
(SELECT OrderID FROM Customers JOIN Orders ON Customers.CustomerID = Orders.CustomerID
WHERE Country=’Germany’))
SELECT ProductID, SUM(Quantity) AS NumOfOrders FROM e
GROUP BY ProductID ORDER BY NumOfOrders DESC LIMIT 1)
SELECT ProductName FROM Products JOIN f ON f.ProductID = Products.ProductID
```

Answer: Boston Crab Meat