DataChallenge

Question 1)

```
library(googlesheets4)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   1.0.2
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
sneakerData <- read.csv('2019WinterDataScienceInternChallengeDataSet-Sheet1.csv')
```

a.

```
summary(sneakerData)
```

```
##     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   2017-03-28 4:00:00 :   3
##  1st Qu.:   1.000   credit_card:1735   2017-03-02 4:00:00 :   2
##  Median :   2.000   debit      :1671   2017-03-07 15:30:37:   2
##  Mean   :   8.787                      2017-03-07 4:00:00 :   2
##  3rd Qu.:   3.000                      2017-03-09 10:46:09:   2
##  Max.   :2000.000                      2017-03-13 2:38:34 :   2
##                                        (Other)            :4987
```

Looking at the summary, we can easily see there are outliners for order_amount as there is at least one maximum value of 704000, much higher than the median value of 284, which can skew the average and give us the answer $3145.13.

```
medSneaker <- median(sneakerData$order_amount)
medSneaker
```

```
## [1] 284
```

A better way to evaluate the average order value is to remove those outliners before we calculate the average.

```
outliers <- boxplot(sneakerData$order_amount, plot = FALSE)$out
```

```
sneakerData2 <- sneakerData[!(sneakerData$order_amount %in% outliers), ]
```

```
aov <- mean(sneakerData2$order_amount)
aov
```

```
## [1] 293.7154
```

As expected, after removing the outliners, we have the new AOV of $293.7154, much closer to the median and more reasonable for sneakers.

b.

For this dataset, I would report the median/average order value, what is the total order/item amount for each shop/payment method.

c.

The median order value is $284.

The average order value after accounting for outliners is $293

The total order/item amount for each shop

```
sneakerData2 %>% group_by(shop_id) %>%
  summarise(totalOrderValue = sum(order_amount),
            totalItemAmount = sum(total_items),
            averageOrderValue = mean(order_amount)
            )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 99 x 4
##    shop_id totalOrderValue totalItemAmount averageOrderValue
##      <int>           <int>           <int>             <dbl>
##  1       1           13588              86              309.
##  2       2            9588             102              174.
##  3       3           14652              99              305.
##  4       4           13184             103              259.
##  5       5           13064              92              290.
##  6       6           18513              99              343.
##  7       7           12208             109              218
##  8       8           11088              84              241.
##  9       9           13806             117              234
## 10      10           16872             114              324.
## # ... with 89 more rows
```

```
sneakerData2 %>% group_by(payment_method) %>%
  summarise(totalOrderValue = sum(order_amount),
            totalItemAmount = sum(total_items),
            averageOrderValue = mean(order_amount)
            )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 4
##   payment_method totalOrderValue totalItemAmount averageOrderValue
##   <fct>                    <int>           <int>             <dbl>
## 1 cash                    450776            3006              290.
## 2 credit_card             491894            3262              293.
## 3 debit                   484493            3208              298.
```

Question 2)

a.

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

There are 54 orders shipped by Speedy Express.

b.

```
SELECT COUNT(EmployeeID), EmployeeID FROM Orders GROUP BY EmployeeID ORDER BY
COUNT(OrderID) DESC;
```

Employee with highest count has 40 and EmployeeID 4

```
SELECT LastName FROM Employees where EmployeeID = 4;
```

Last name is Peacock

c.

```
Select * from (select sum(Quantity) as Q, ProductID from (select ProductID, Quantity from ( select * from
(SELECT country, OrderID FROM Customers INNER JOIN Orders ON Customers.CustomerID =
Orders.CustomerID) t1 INNER join OrderDetails ON t1.OrderID = OrderDetails.OrderID) where country =
‘Germany’) Group By ProductID) Order By Q Desc
```

The product that was ordered the most in Germany has product ID = 40 with quantity 160

```
SELECT * FROM Products where ProductID = 40;
```

The product was Boston Crab Meat