Shopify Challenge

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Reading the data I prefer file.choose(), where I can manually select the data into the console

shopify\_data<-read.csv(file.choose(), header=T)

Let’s see what the data looks like

head(shopify\_data,10)

## order\_id shop\_id user\_id order\_amount total\_items payment\_method  
## 1 1 53 746 224 2 cash  
## 2 2 92 925 90 1 cash  
## 3 3 44 861 144 1 cash  
## 4 4 18 935 156 1 credit\_card  
## 5 5 18 883 156 1 credit\_card  
## 6 6 58 882 138 1 credit\_card  
## 7 7 87 915 149 1 cash  
## 8 8 22 761 292 2 cash  
## 9 9 64 914 266 2 debit  
## 10 10 52 788 146 1 credit\_card  
## created\_at  
## 1 2017-03-13 12:36:56  
## 2 2017-03-03 17:38:52  
## 3 2017-03-14 4:23:56  
## 4 2017-03-26 12:43:37  
## 5 2017-03-01 4:35:11  
## 6 2017-03-14 15:25:01  
## 7 2017-03-01 21:37:57  
## 8 2017-03-08 2:05:38  
## 9 2017-03-17 20:56:50  
## 10 2017-03-30 21:08:26

tail(shopify\_data,10)

## order\_id shop\_id user\_id order\_amount total\_items payment\_method  
## 4991 4991 24 860 140 1 cash  
## 4992 4992 61 707 158 1 cash  
## 4993 4993 49 739 258 2 debit  
## 4994 4994 4 834 384 3 debit  
## 4995 4995 12 954 201 1 cash  
## 4996 4996 73 993 330 2 debit  
## 4997 4997 48 789 234 2 cash  
## 4998 4998 56 867 351 3 cash  
## 4999 4999 60 825 354 2 credit\_card  
## 5000 5000 44 734 288 2 debit  
## created\_at  
## 4991 2017-03-26 0:32:39  
## 4992 2017-03-13 18:56:04  
## 4993 2017-03-24 13:48:08  
## 4994 2017-03-18 4:12:20  
## 4995 2017-03-22 0:38:11  
## 4996 2017-03-30 13:47:17  
## 4997 2017-03-16 20:36:16  
## 4998 2017-03-19 5:42:42  
## 4999 2017-03-16 14:51:18  
## 5000 2017-03-18 15:48:18

Loading my data wrangling and visualization libraries

library("tidyverse")

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.0.2 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library("ggplot2")

Quick look up of shopify data set summary

summary(shopify\_data)

## 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

Something quicky stands out to me that the mean order\_amount was used as the average order value. Doesn’t seem right to me but I’d have to investigate what average order value means or represents.

Visited this link <https://www.shopify.com/blog/average-order-value#averageorder> and they defined average order value as Revenue/orders.

Problem solved.

Time to calculate the actual AOV.

Q1. A better way to evaluate this data is by adding the order amount

I almost used base R to quickly solve the AOV shopify\_AOV<- (sum(shopify\_datatotal\_items)) but I looked back at the head() and discovered that some shop\_id are repeated on the sheet.!

Quick view of how many shops we have.

shops <- group\_by(shopify\_data, shop\_id)  
  
summarise(shops,n= n()) %>%   
 arrange (desc(n))

## # A tibble: 100 x 2  
## shop\_id n  
## <int> <int>  
## 1 53 68  
## 2 71 66  
## 3 19 64  
## 4 13 63  
## 5 89 61  
## 6 59 60  
## 7 69 60  
## 8 6 59  
## 9 9 59  
## 10 41 59  
## # ... with 90 more rows

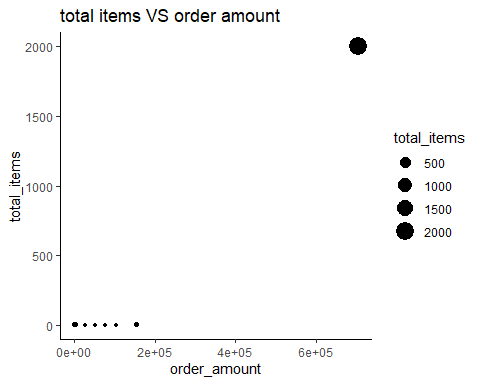
Time to use dplyr, lot of shift ctrl M :)

shopify\_data %>%   
 group\_by(shop\_id) %>%   
 summarise (shopify\_AOV= sum(order\_amount)/sum(total\_items)) %>%   
 summarise (mean(shopify\_AOV))

## # A tibble: 1 x 1  
## `mean(shopify\_AOV)`  
## <dbl>  
## 1 408.

Time for some data VIZ, pictures tell a thousand words time to see what’s going on per shop

ggplot(shopify\_data)+  
 geom\_jitter(aes(order\_amount,total\_items, size=total\_items))+  
 scale\_color\_gradient(low="orange", high="purple")+  
 theme\_classic()+  
 labs(title = "total items VS order amount",x="order\_amount",y= "total\_items")



I’d take a minute to pause and recondsider using the mean as a measure of central tendency!!! That’s a huge outlier!!!!

I honestly should have visualized and done some data inspection at the start. I’d use the median.

shopify\_data %>%   
 group\_by(shop\_id) %>%   
 summarise (shopify\_AOV= sum(order\_amount)/sum(total\_items)) %>%   
 summarise (median(shopify\_AOV))

## # A tibble: 1 x 1  
## `median(shopify\_AOV)`  
## <dbl>  
## 1 153

The value i’ll report is 153, also I notify the team about this outlier. Which can be found by using arrange(desc())

```