Exercises:

1. Crunch the data and tell us whether our return rate is trending up or down. Additional insights are welcome, but not required

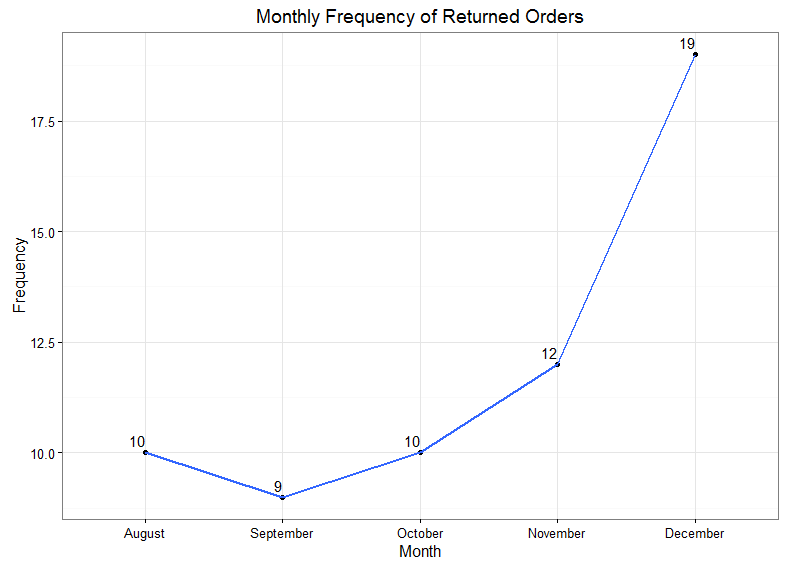
2. Write the SQL code to produce number of completed orders by date (name of source data table is ‘casper\_orders’)

Prerequisites:

* Prior to doing any analysis, I had to transform the data in Microsoft Excel in order for it to show a "monthordered" and "monthreturned" column. It is simply the month name for the dateordered and datereturned column, respectively.
* Used R programming to load data set. I did come across a minor issue with "NA" appearing for blank rows in the datereturned column but this didn't affect the analysis.

First Exercise (First Graph):

The first thing I want to do is separate between the returned orders and complete orders. Therefore I subsetted the entire dataset by table "b" and table "d". Table b shows all the returned orders and table d shows all the complete orders. Once Table b was populated, I graphed the data to see the trend.

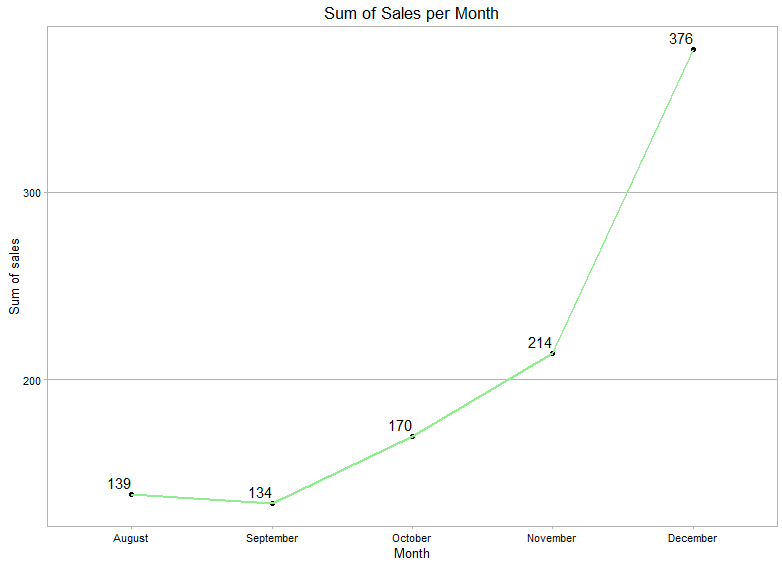


Note: When plotting this graph, I used the month in which the item was ordered for the x axis.

The reason I used the ordered month date was because the return is based on that month's order, not in the return order month - this is to compare apples to apples. The date range in this graph is from August 2016 to January 2017 with the return trending upwards. Although it is trending upwards based on the numbers and the graph, it would be naive to assume this without looking at Table d for the complete orders.

Second graph:

This leads me to plot the sum of all sales per month to see if the sales per month affects the returned orders. In other words, if the number of sales per month is increasing, this might make the return rate trend upward also.



Note: When plotting this graph, I used the complete filter in the orderstatus column and does not include the subtraction of returned items.

Unsurprisingly, this graph shows an upward trend which correlates largely with the trend in return orders. In fact, when I ran a correlation formula in R, I received a 99.3 % rate, which means the rate of returns is highly correlated with the sum of sales per month.

Analysis:

It appears both the sum of sales and rate of return is trending upwards and this would skew the return rate (more sales generated, the higher the likelihood of returns). Lets break this out into percentages to see numerically if the return rate is trending up or down.

|  |  |  |  |
| --- | --- | --- | --- |
| **month** | **numberreturn** | **sum** | **percent** |
| August | 10 | 139 | 7.19% |
| September | 9 | 134 | 6.72% |
| October | 10 | 170 | 5.88% |
| November | 12 | 214 | 5.61% |
| December | 19 | 376 | 5.05% |

As you can see the chart above, the second and third column are based on the graphs mentioned before and the fourth column is the percentage of numberreturn divided by the sum. This chart actually shows the opposite of what the other graph depicts in which the rate of return is trending downward. This is a more accurate analysis of the rate of returns because the number of returns are pitted against the sum of sales, instead of viewing an isolated graphical chart of the return rates.

Conclusion:

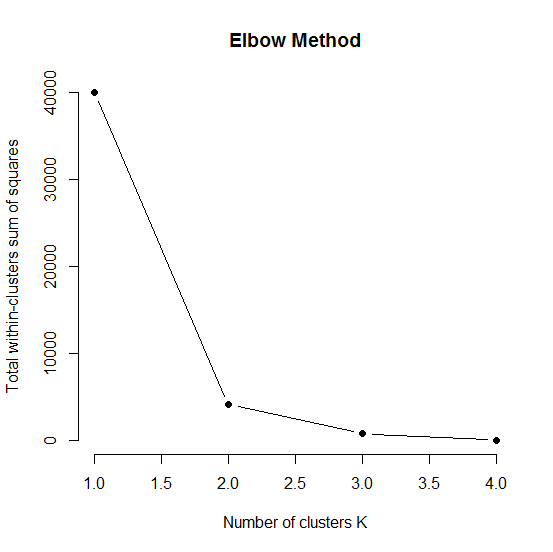
I determined that the rate of return is decreasing by (min = 0.27%, max = 0.83%) and not trending upwards. The chart analysis is a more accurate depiction of the rate of return because the number of returns is being compared with the increasing number of sales per month and a return cannot occur without a sale; the rate of return is dependent on monthly sales. As a conclusion, the results show that the monthly rate of return is trending downward by small incremental percentages.

Additional Insights:

Due to my curiosity, I wanted to see what would happen if I applied a K-Means Clustering algorithm on the set of data. But first, I will have to determine the optimal number of clusters using the elbow method. I will use the below chart as my data input.

|  |  |  |
| --- | --- | --- |
| **month** | **numberreturn** | **sum** |
| August | 10 | 139 |
| September | 9 | 134 |
| October | 10 | 170 |
| November | 12 | 214 |
| December | 19 | 376 |

After plugging the data into the elbow method, the result is:



To determine the optimal number of clusters, you would simply view the elbow bend in the total within-clusters sum of squares (basically the distances between the data points and the center of clusters). After the bend or elbow, the cluster within-clusters sum of squares decreases significantly which means the more clusters you try to apply, the less optimality there is in it. Obviously with more clusters, there are less error but it doesn't make it more optimal or effective. Therefore after seeing the elbow in the graph, we can determine that this is the most optimal number of clusters because adding more clusters would not improve the cluster strength. As a result, the optimal number of clusters is 2. Using the K-Means Clustering algorithm with optimal number of clusters as 2, I get:

|  |  |  |  |
| --- | --- | --- | --- |
| **month** | **numberreturn** | **sum** | **cluster** |
| August | 10 | 139 | 1 |
| September | 9 | 134 | 1 |
| October | 10 | 170 | 1 |
| November | 12 | 214 | 1 |
| December | 19 | 376 | 2 |

This shows that December has its own cluster and the other 4 months are in another cluster. This is insightful because December is considered an outlier in our data set with Christmas as a shopping holiday. You can see that "normal" months without Christmas vary closely with one another. The chart below shows the averages of each cluster per column.

|  |  |  |
| --- | --- | --- |
| **cluster** | **numberreturn** | **sum** |
| 1 | 10.25 | 164.25 |
| 2 | 19 | 376 |

To get a visual idea of the clusters with the months, I constructed a dendrogram using the hierarchical clustering method:



As you can see, there are two separate branches (clusters) with sub branches underneath them. The difference among each branch is based on the differences of their height and their branch. For example, August and September are more closely related than August and October.

Since the data provided is limited, clustering or segmentation processes would also be limited. But if there were additional data, we could split the data 60% training and 40% testing and see the results of the clusters to get an even more accurate reading in regards to understanding seasonality spikes or to provide a predictive model that could potentially forecast the future months. Then compare those predicted results with the actual results that are in the 40% test data set. Finally use that to compare the models and to find the optimal model.

Second Exercise:

I used R programming (sqldf library) to finish the exercise.

SQL Code in R:

library(sqldf)

a <- read.csv("C:\\Users\\Guest\\Documents\\R training\\Casper\\data.csv")

sqla <- sqldf(

'SELECT MONTHORDERED AS MONTH\_ORDER,SUM(ORDERS) AS SUM\_ORDERS

FROM a

WHERE ORDERSTATUS = "complete"

GROUP BY MONTHORDERED

ORDER BY MONTHORDERED ASC'

)