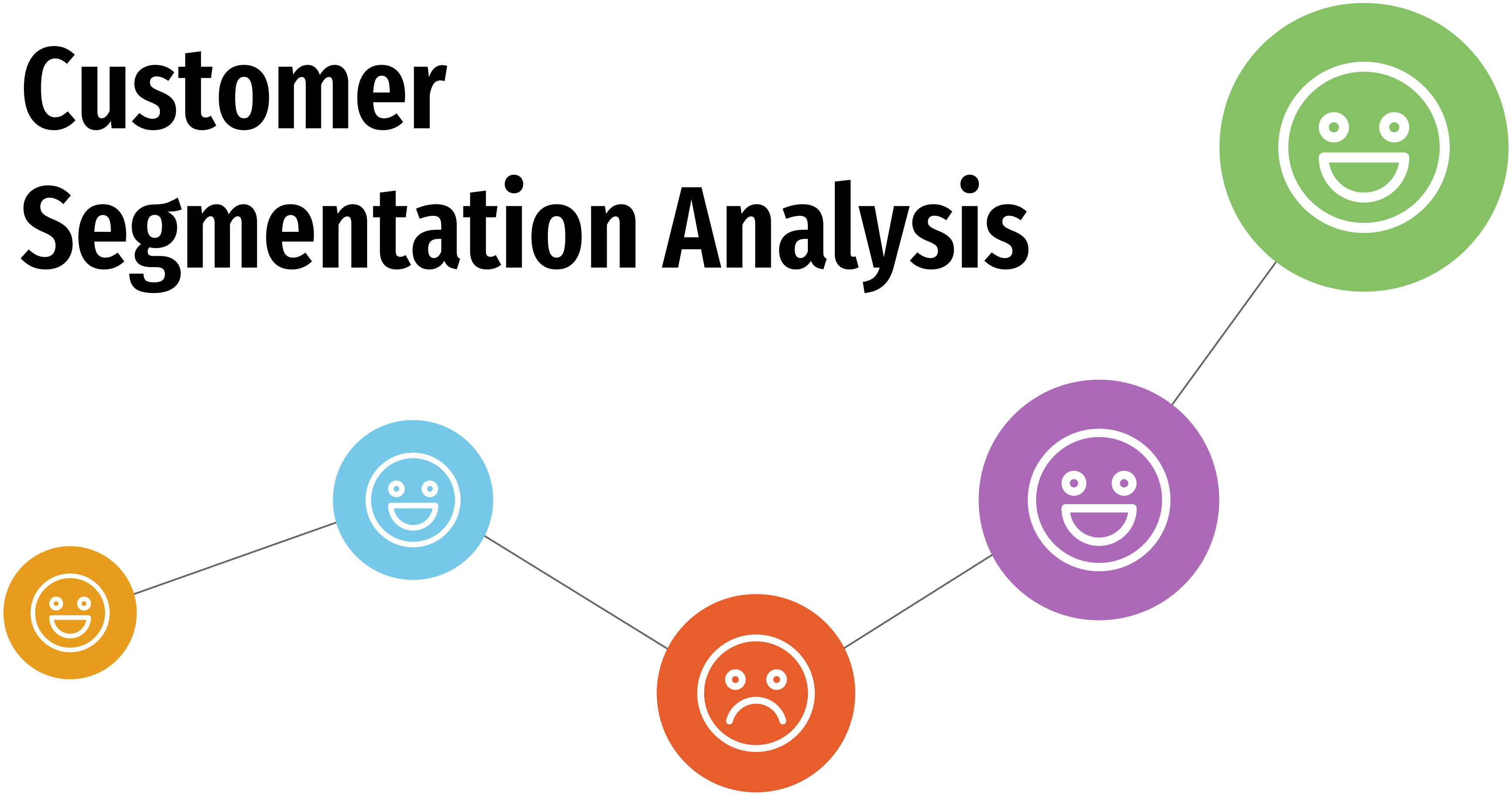
Customer Segmentation Analysis



Executive Summary

In this case study, a company has planning for customer loyalty program. In order for the program to run effectively, the company asks you as Sales Manager to create the right target customer from the existing transaction data. Due to this background, we need to analyze and create a segmentation that is in accordance with the transaction habits of our customers

Objectives :

1. Define the right number of customer groups / cluster based on the amount of spent value, the frequency over a certain period, and the last time they made a transaction.
2. Describe each cluster then evaluate the results of the analysis for learning and provide recommendations for loyalty programs

Result Summary :

Compared to the RFM method, the K-Means method is preferred chosen, because it is easier to interpret the characteristics of transaction behavior. From the results of K-Means, we also find a sufficient number of customers for loyalty program opportunities.

Data Introduction

The data used for analysis is data with a period of four years. It has transactions from January 1 2011 until April 31 2014. There were 51,290 transaction lines data during the period.

We have 24 columns consisting of customer identity (including area), place (outlet) and transaction date. The number of transactions and also the details of the items purchased (Brand-Category- Subcategory- ID product)

DataSource: **Global_Superstore Data**

Tools :

Google Colaboratory for Jupyter environment

Yellowbrick Library Package for visualization

Explanatory Data Analysis 1

Check data type and missing value

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 24 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   row_id          51290 non-null  int64
 1   order_id        51290 non-null  object
 2   order_date      51290 non-null  object
 3   ship_date       51290 non-null  object
 4   ship_mode       51290 non-null  object
 5   customer_id     51290 non-null  object
 6   customer_name   51290 non-null  object
 7   segment         51290 non-null  object
 8   city            51290 non-null  object
 9   state           51290 non-null  object
 10  country         51290 non-null  object
 11  postal_code     9994 non-null   float64
 12  market          51290 non-null  object
 13  region          51290 non-null  object
 14  product_id      51290 non-null  object
 15  category        51290 non-null  object
 16  sub_category    51290 non-null  object
 17  product_name    51290 non-null  object
 18  sales           51290 non-null  float64
 19  quantity        51290 non-null  int64
 20  discount        51290 non-null  float64
 21  profit          51290 non-null  float64
 22  shipping_cost   51290 non-null  float64
 23  order_priority  51290 non-null  object
dtypes: float64(5), int64(2), object(17)
memory usage: 9.4+ MB
```

No issue in datatype and no null on each column. Except postal code, we can change to the type object. but its ok for don't anything to this null rows

```
[26] # check total unique customer for each group category
     df=data.groupby("category")["customer_id"].nunique()
     df
```

```
category
Furniture          1427
Office Supplies    1585
Technology         1485
Name: customer_id, dtype: int64
```

There are no significant difference of total unique customer from each category

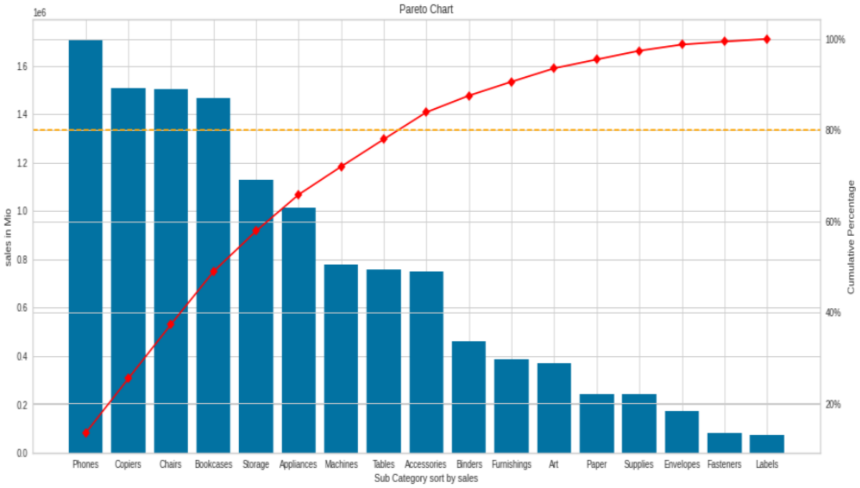
```
data.describe()
```

	row_id	postal_code	sales	quantity	discount	profit	shipping_cost
count	51290.00000	9994.000000	51290.000000	51290.000000	51290.000000	51290.000000	51290.000000
mean	25645.50000	55190.379428	246.490581	3.476545	0.142908	28.610982	26.375818
std	14806.29199	32063.693350	487.565361	2.278766	0.212280	174.340972	57.296810
min	1.00000	1040.000000	0.444000	1.000000	0.000000	-6599.978000	0.002000
25%	12823.25000	23223.000000	30.758625	2.000000	0.000000	0.000000	2.610000
50%	25645.50000	56430.500000	85.053000	3.000000	0.000000	9.240000	7.790000
75%	38467.75000	90008.000000	251.053200	5.000000	0.200000	36.810000	24.450000
max	51290.00000	99301.000000	22638.480000	14.000000	0.850000	8399.976000	933.570000

Check for max and min for quantity and discount to make sure that was make sense

Explanatory Data Analysis 2

Create pareto to check top 80% sales contribution from category and subcategory



Top Subcategory: Phones, Copiers, Chairs, Bookcases, Storage, Appliances, Machines, Tables, dan Accessories.

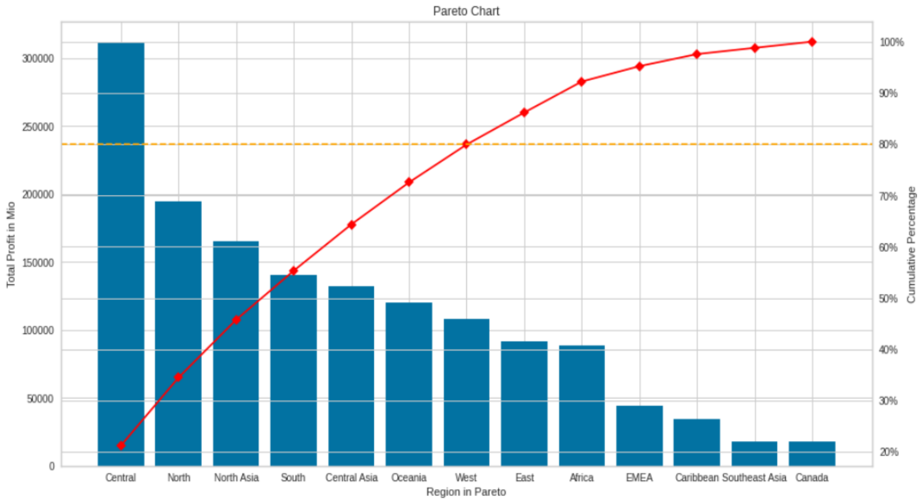
Check also profit for each top category from previous pareto.

	sub_category	sales	cum_percentage_sales	profit	cum_percentage_profit
0	Phones	1.706824e+06	13.50	216717.00580	32.39
1	Copiers	1.509436e+06	25.44	258567.54818	17.62
2	Chairs	1.501682e+06	37.32	140396.26750	62.64
3	Bookcases	1.466572e+06	48.92	161924.41950	43.42
4	Storage	1.127086e+06	57.83	108461.48980	78.87
5	Appliances	1.011064e+06	65.83	141680.58940	53.08
6	Machines	7.790601e+05	71.99	58867.87300	91.85
7	Tables	7.570419e+05	77.98	-64083.38870	100.00
8	Accessories	7.492370e+05	83.91	129626.30620	71.48
9	Binders	4.619115e+05	87.56	72449.84600	83.81
10	Furnishings	3.855783e+05	90.61	46967.42550	99.00

We can drop subcategory tables and Machines because they give minus profit contribution or bad on profit

Explanatory Data Analysis 3

Create pareto to check top 80% profit contributors from Region and area



From the chart there are 8 region as a top contributors :

- Central
- North
- North Asia
- South
- Central Asia
- Oceania
- West

How about contribution from each city ?

```python
city_cont= city_cont[city_cont["cum_percentage"] <= 80.00]
city_cont
```

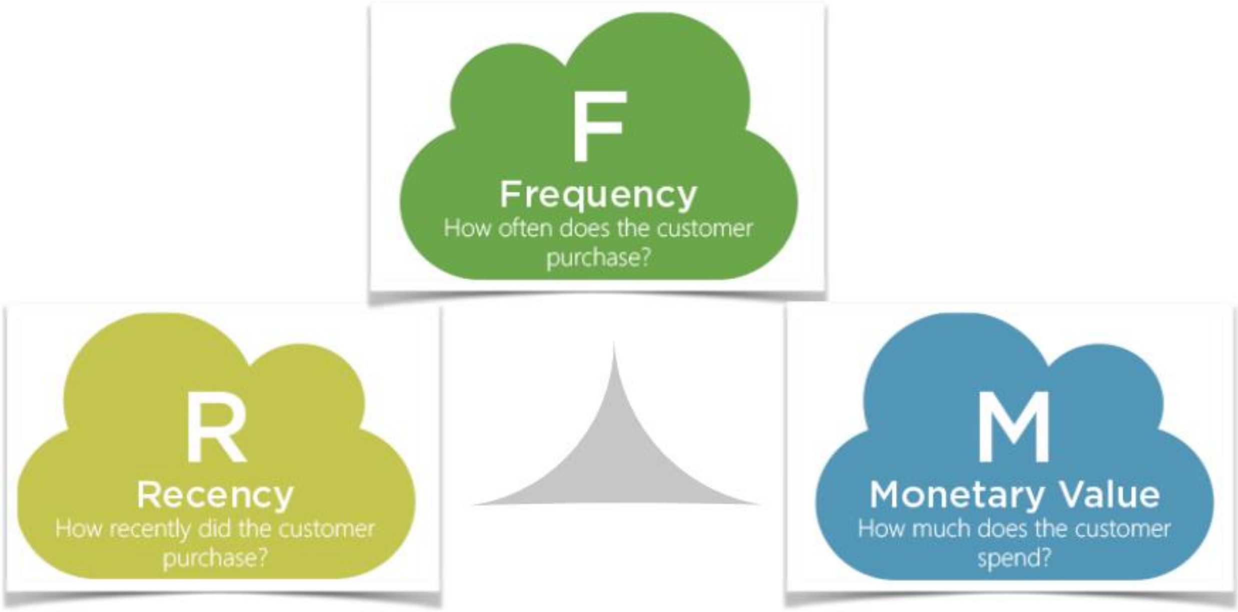
	city	profit	cum_percentage
2290	New York City	62036.98370	4.23
1910	Los Angeles	30440.75790	6.30
2936	Seattle	29156.09670	8.29
1989	Managua	17853.71804	9.51
2843	San Francisco	17507.38540	10.70
...	...	...	...
723	Chongqing	1727.64000	79.43
423	Blagoveshchensk	1709.37000	79.55
2653	Rajshahi	1708.92000	79.66
2071	Mecca	1707.21000	79.78
3146	Tamworth	1702.82100	79.89

274 rows × 3 columns

From 3.636 city we found 274 top city. It will give us more insight for next step when cluster / segment customer has defined.

RFM ANALYSIS

RFM


F
Frequency
How often does the customer purchase?
R
Recency
How recently did the customer purchase?
M
Monetary Value
How much does the customer spend?

Create Data for Model

Transaction data that we have is still in the form of raw data. Where each customer can make multiple purchases for a variety of different products and even do transactions in different markets with different payment methods.

	customer_name	category	product_name	market	sales	order_id
0	Aaron Bergman	Furniture	Bush Library with Doors, Mobile	APAC	660.3120	1
1	Aaron Bergman	Furniture	Deflect-O Door Stop, Erganomic	APAC	372.9132	1
2	Aaron Bergman	Furniture	Eldon Clock, Black	LATAM	75.3600	1
3	Aaron Bergman	Furniture	Eldon Photo Frame, Erganomic	APAC	141.8250	1
4	Aaron Bergman	Furniture	Global Push Button Manager's Chair, Indigo	US	48.7120	1

for above sample data: Aaaron Bergman bought five types of product in different markets and different total values

Meanwhile, to make this RFM model we need to know the behavior for each customer. so that we can assess the behavior of the transaction that makes the customer include to what cluster.

From 4 years period from data transaction, there are 1.590 unique customer id that we will check their recent transaction, much value they spent, and how often they doing transaction (frequency)

```
data_for_model
```

	customer_id	Recency	Frequency	MonetaryValue
0	AA-10315	8	42	13747.41300
1	AA-10375	6	42	5884.19500
2	AA-10480	125	38	17695.58978
3	AA-10645	28	73	15343.89070
4	AA-315	2	8	2243.25600
...	...	...	...	...
1585	YS-21880	9	54	18703.60600
1586	ZC-11910	200	1	7.17300
1587	ZC-21910	3	84	28472.81926
1588	ZD-11925	3	18	2951.22600
1589	ZD-21925	1	36	9479.34440

1590 rows × 4 columns

RFM Segmentation Score

As explained on the data introduction, we will analyze the whole of customer data, where the date of the period used is the last date of the transaction minus the start date (from all of transaction history) after calculating, the period is four years. Start on 1 Jan 2011 until 31 Dec 2014.
Each matrix that is measured has a different unit. **Matrix Recency** is the number of days from 31 Dec 2014 minus the last transaction date for each customer, **therefore the unit is days**.
Frequency is calculated as the number of transactions carried out in **units of times**.
and **monetary** spent is definitely the **currency that was paid during the period** and the result of the accumulation of spending many times.

So that each matrix (red box) must be converted into an ordinal scale (green box). 1 – 4 means worst to best, this scale is devided by quartiles. Then the RFM score for each customer can be calculated.

	customer_id	Recency	Frequency	MonetaryValue	R	F	M	rfm_score
0	AA-10315	8	42	13747.41300	4	3	4	434
1	AA-10375	6	42	5884.19500	4	3	2	432
2	AA-10480	125	38	17695.58978	1	3	4	134
3	AA-10645	28	73	15343.89070	3	4	4	344
4	AA-315	2	8	2243.25600	4	1	2	412

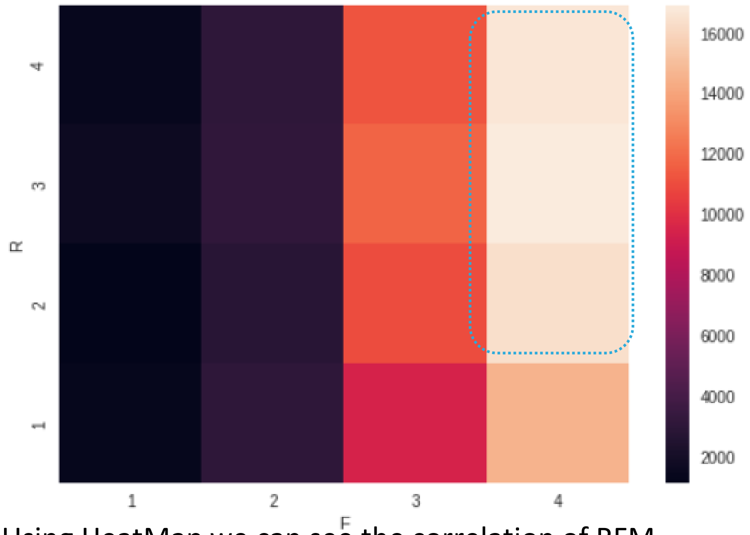
Now, we get the RFM Score for each customer ID, so let check the distribution of RFM Score!!

Post Segmentation Analysis

Heat Map Visualization

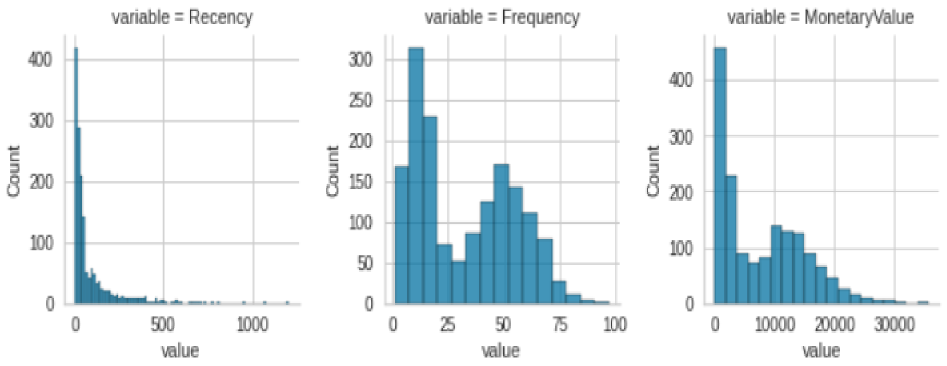
```
sns.heatmap(
    pd.pivot_table(data_for_model[["R", "F","MonetaryValue"]], values = "MonetaryValue", index = ["R"], columns = ["F"])
)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f666f95cc50>
```



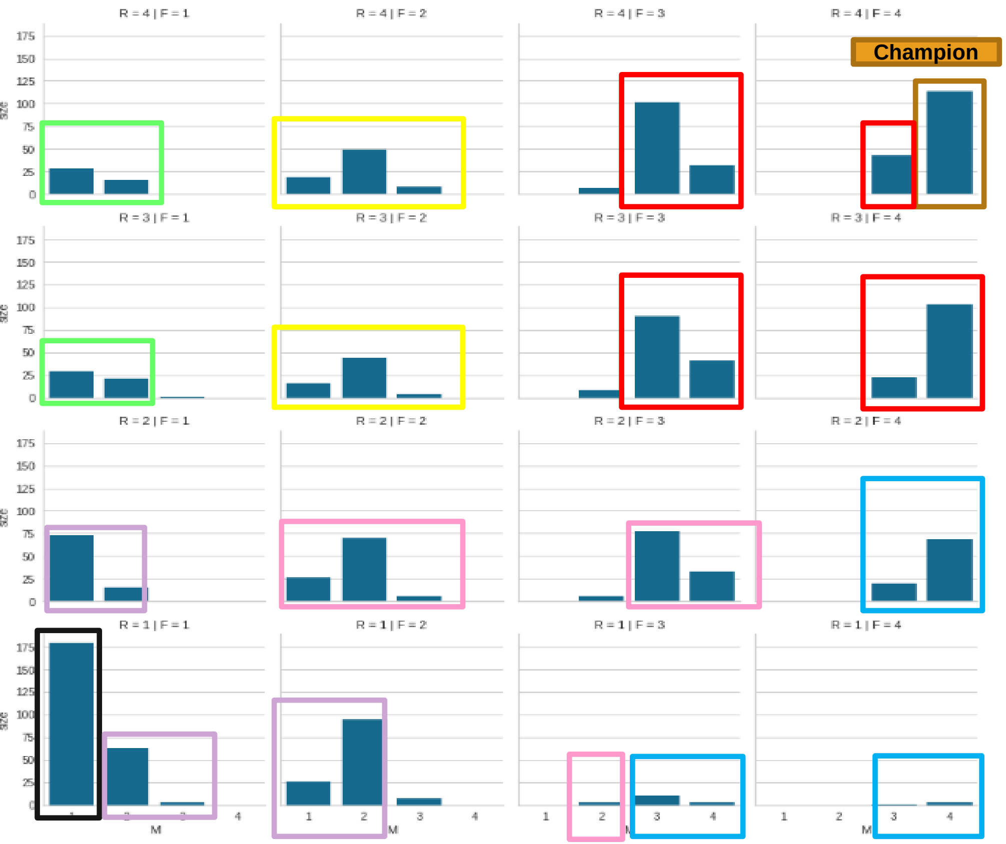
Using HeatMap we can see the correlation of RFM element. **The brighter columns then RFM Score get better value.**

RFM Distribution



- From recency distribution we found that most of transaction done at Quartile 1 value < 250, mostly have value= 0. that seems good for recency value.
- Most of the last transactions have done at q1 value < 250, mostly frequency distribution show just little customers who have frequency more than 75 times. But this has good distribution on middle (Q2-Q3). we have to check spending of this middle customer. Whether is possible to upgrade to the loyal customer or not.

Post Segmentation Analysis



Basically, from the number of scales for each matrix, 64 types of RFM scores can be created.

However, as can be seen on left image, not all RFM scores are filled in. So we can group multiple RFMs into more interpretable group definitions.

·fm_score
434
432
134
344
412

Totally 64 maximal type of RFM Score

- Champion (gold box) score 444: Best value on each element. has highest RFM Values.
- Loyal Customer (red box) : one level under Champion. They have 3-4 values for RFM.
- Promising (Yellow box): customers who have transactions often but still have small value. or rarely transact but once they spend a large amount.
- Recent (Green box) : Customer whose transactions recent in very small amount.
- Needing Attention (Pink Box) : customers who haven’t transacted in a long time
- Can’t lose them (Blue Box) : customers who used to transact often but haven’t transact for a long time.
- At risk (Purple Box) : customers who already at lowest level and are risk to churn
- Lost (Black Box): Confirmed customer churn

Post Segmentation Analysis

Check size for each segment

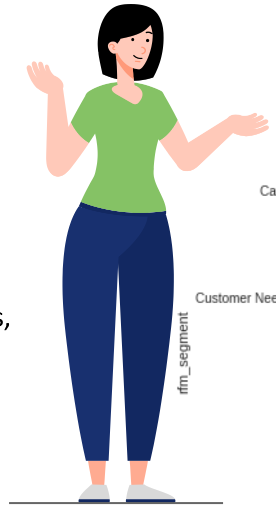
```
rfm_segment
At Risk                       285
Can't Lose Them               218
Champion                      114
Customer Needing Attention    160
Lost                          180
Loyal Customer                434
Promising                     170
Recent Customers               29
```

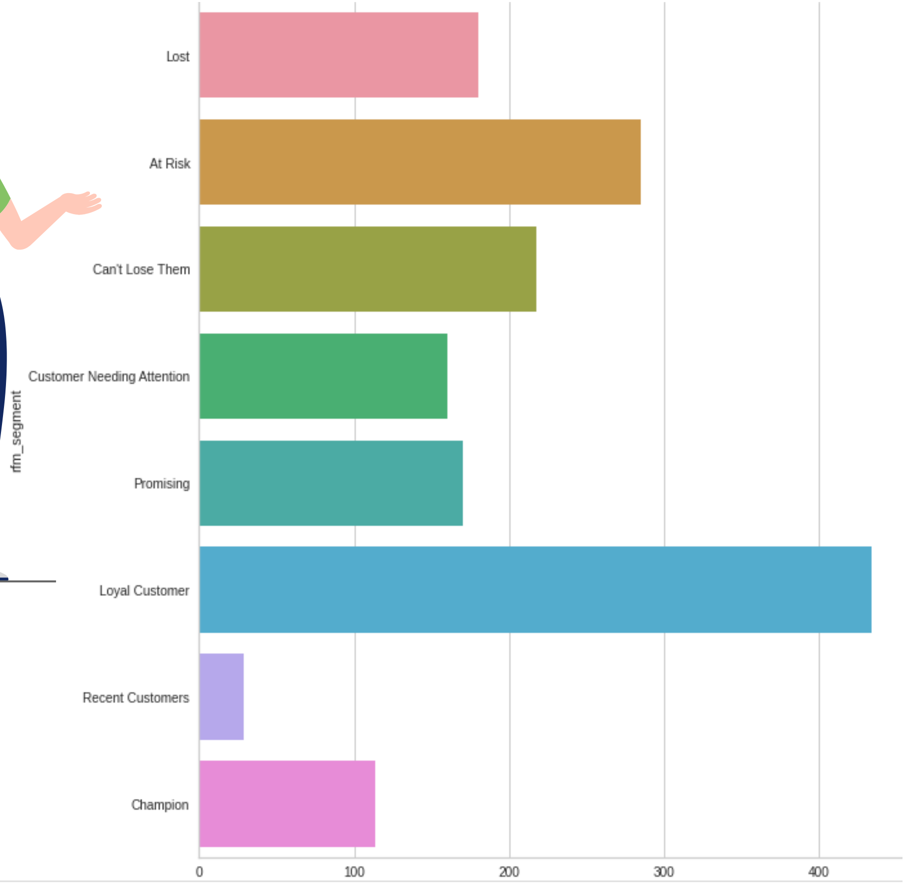
RFM Segmentation give us 8 clusters. Where the number of customers for the category of loyal customers is 434 customers, 27 % from total Customers

all this loyal customers can be assessed further to increase, so that it will become a Champion.

The second largest number after loyal customers are customers in the "At Risk" group. This also needs certain treatment so that it doesn't fall into the "Lost" group.

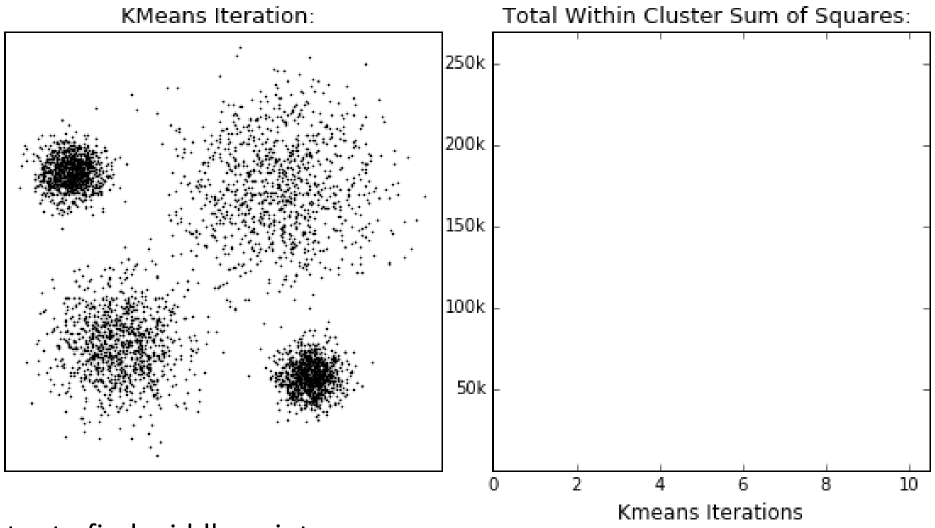
Different treatments will be adjusted to every customer group. If the company wants to give each of these treatment, the consideration is that the costs incurred are also getting bigger.





K-Means Clustering

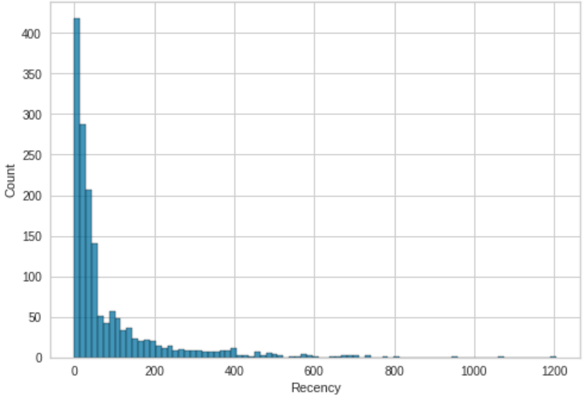
K-Means Clustering



Use mean value from each cluster to find middle point
Initiate random point every time we run the algorithm (need to seed the RNG)
Manually set the number of segments we want

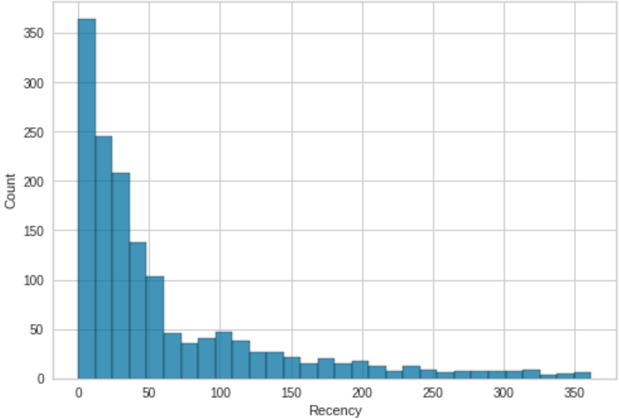
K-Means Method is **sensitive to the outliers** and only applicable if **mean is defined.**

Prepared data for K-Means Clustering

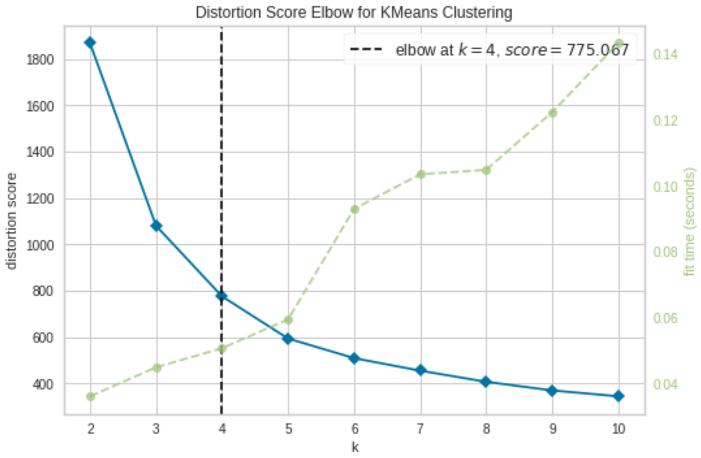


The distribution picture above shows a very large difference between the number of customers with a recency of 0-200 compared to the number of customers with a recency of over 360 days (one years), only 75 customers with a recency > 300 days, or only 5% of the total.

This 5% will be take out from the analysis, so that data recency which is too low does not interfere to the mean values of between clusters. The Recency period is also will be more valid, because 95% of the last transactions from all customers are in the last year.



Define Cluster Optimal with Elbow Method



With the elbow method we will quickly find out the optimal cluster point because this method has calculated the average point distance between clusters.

The principle of a good optimal cluster is when the distance between clusters is getting bigger but the distance of the points in one cluster is getting smaller.

- The K-means method was sensitive to the variance, so it had to be reduced by the Standard Scaler.

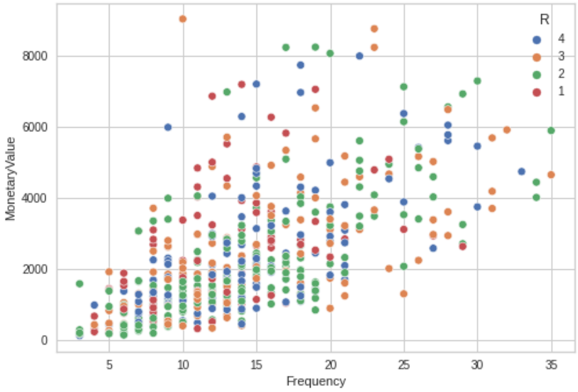
customer_id	Recency	Frequency	MonetaryValue
AA-10315	8	42	13747.41300
AA-10375	6	42	5884.19500
AA-10480	125	38	17695.58978
AA-10645	28	73	15343.89070
AA-315	2	8	2243.25600
...	...	...	...
YS-21880	9	54	18703.60600
ZC-11910	200	1	7.17300
ZC-21910	3	84	28472.81926
ZD-11925	3	18	2951.22600
ZD-21925	1	36	9479.34440

	Recency	Frequency	MonetaryValue
0	-0.723857	0.390390	0.786057
1	-0.750743	0.390390	-0.347442
2	0.849010	0.206139	1.355196
3	-0.454991	1.818336	1.016193
4	-0.804517	-1.175743	-0.872292
...	...	...	...
1506	-0.710413	0.943143	1.500503
1507	1.857258	-1.498183	-1.194628
1508	-0.791073	2.325026	2.908757
1509	-0.791073	-0.715116	-0.770236
1510	-0.817960	0.114014	0.170806

1511 rows × 3 columns

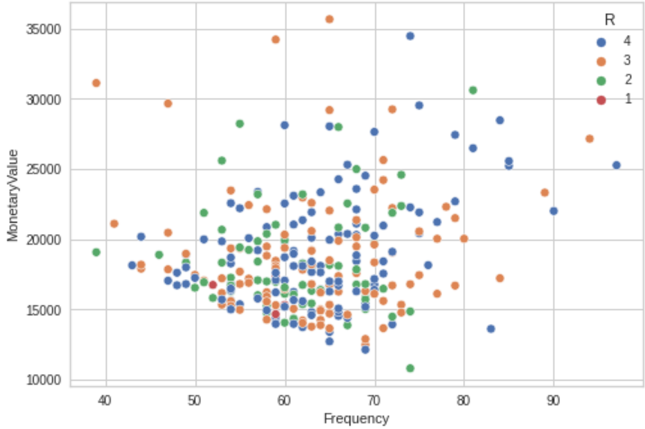
Result K-Means Cluster

Cluster 0



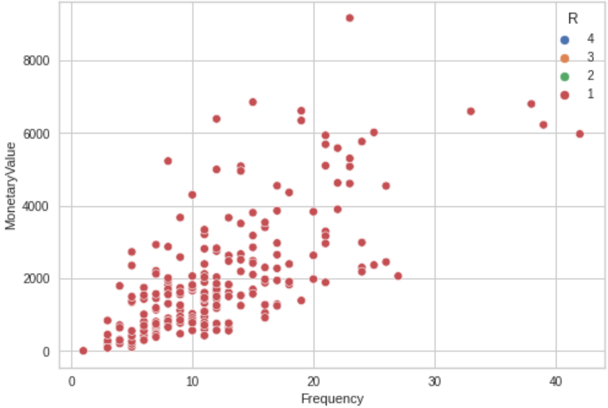
The most frequency range = 5-25 times , with the last transaction in the second quarter quite a lot. The most spent under the number 400.

Cluster 1



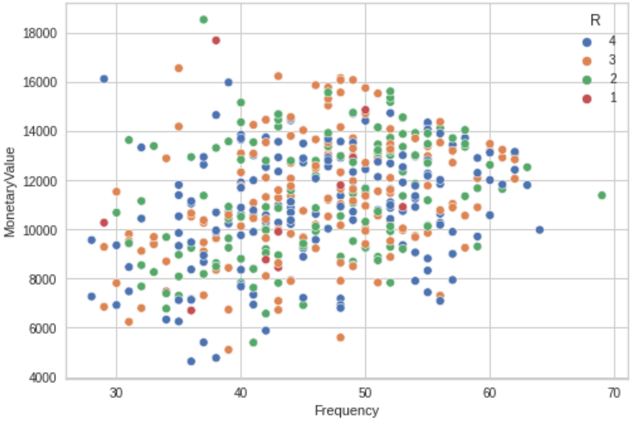
This cluster can be considered as a Gold Customer where the transaction frequency is quite high 50-80 times even up to the 90s. and also enormous monetary value. Especially in the range of 800-1600 customers which is quite high.

Cluster 2



This cluster can be considered a cluster that is very risky for churn. Apart from not transacting for a long time (last in Q1) and the group who spent in Q1 with nominal under 200 - 300 is quite a lot.

Cluster 3



This cluster has a pretty good recency, seen from the blue dots, where the spent value is also quite high. Especially in the range of 8000 - 15000. Very potential to be upgraded to Gold Customer.

Cluster Analysis and Recommendation

 # Customers of Each K-Means Cluster

Due to models tested (RFM and K-Means) I would recommend K-Means as a clustering model. where the transaction data period used is only the last year's transaction data. described earlier in the K-Means analysis.

The K-Means cluster is easier to interpret to create future business opportunities.

Then from this K-Means segmentation, you can also check what product clusters or subcategories are most frequently purchased. And it can be seen which region has a greater probability of buying it according to Pareto data.

Cluster 3 can be used as a new opportunity for retention so that they can become more loyal to provide more intensive promotions because the monetary value of customers in cluster 3 is quite large even though the frequency is as high as customers in Cluster 1

Meanwhile, Cluster 0 where the potential loyal customers are still below Cluster 3. needs to be considered because there are quite a lot of customers above 500 customers. which is a shame if you have to stop trading in Q2 and Q3.

and customers in Clusters 1 and 3 if seen are also quite a lot. With nearly 500 customers in cluster 3. Cluster 1 has almost 300 customers. This is a great opportunity for the Customer loyalty program

Thank You