RFM Segmentation Complete Analysis

I took the dataset of e-commerce company from Kaggle.com, did segmentation based on customers behavior. It will be helpful for the company to making marketing strategies and decisions according to these segments. For instance, company will come to know about each category of customers and what steps need to take in terms of campaigns to retain profitable customers and different type of approach to attracts new customers.

RFM analysis is important to increase revenue by targeting relevant or specific clusters of customers. The benefit of RFM analysis that it generates higher rates of response, increase loyalty and customer lifetime value.

Data set

The dataset Online Retail II downloaded from Kaggle.com, it's a data of online retail store based in the UK of year 2010 – 2011. The company mainly deal in souvenirs products. The majority of the company's customer are corporate customers.

There is total eight variables in the dataset, and it contains 541910 rows. Invoice No is a unique number for each transaction, if it starts with C which means cancelled operations. Stock code is a unique product code, description is a product details, Quantity refers number of products in the invoices have been sold, Invoice Date, unit price is a product price in pounds, customer ID and Country. The fix x below represents the first 5 rows of company's dataset.

```
data.head()
```

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Data Preprocessing

I dropped the duplicates from the country and Customer Id dataset, sort the data in descending order of customer Id. Checked the missing values in the dataset and validate if there are any negative values in the quantity column.

Introduce the RFM class, add the method name **data_cleaning()**. The purpose of that function is to rename the dataset variables, if it found the country equals to United Kingdom drop that records and reset the index. Remove missing values from customer Id and ignore from description column. Filter out the records by applying the condition quantity should be greater than 0. Initially the datatype of date was string, so convert it into datetime. Calculate the total amount by multiplying quantity with unit price.

Add method of calculate RFM, set the latest date 10/12/2011 as the last invoice date was 09/12/2011 this is to calculate the number of days from recent purchase. Calculate the RFM modelling scores for each customer ID. Recency is calculated by calculating number of days from recent purchase, Frequency of customer is calculated by counting total number of invoices of each customer and monetary through doing sum of total amount. Rename the columns invoice date, invoice number and total amount to recency, frequency and monetary respectively.

Add methods for to do recency, frequency, and monetary segmentation. Scores are calculated through quantile method, split into four segments quantile [<=0.25, <=0.5, <=0.75 and >=0.75] to categorized each customer. Add two methods to calculate the scores recency, frequency and monetary. Less recency number means customer visited for shopping like few days ago, so it's better. And more frequency, monetary means frequently customers visited and spending higher on shopping, so higher value of frequency and monetary are better.

I calculate recency score using quantile method, if quantile is less than 0.25 will return 1, less than 0.50 will return 2, less than 0.75 will return 3 and else will return it 4. In calculating frequency and monetary score, if quantile is less than equals to 0.25 return 4, less than 0.50 will return 3, less than 0.75 will return 2 and else will return 1.

```
def recency_score(self,x,p,d):
    if x <= d[p][0.25]:
        return 1
    elif x <= d[p][0.50]:
        return 2
    elif x <= d[p][0.75]:
        return 3
    else:
        return 4
#more freqynecy and monetory are better
def frequency_and_monetory_score(self,x,p,d):
    if x <= d[p][0.25]:
        return 4
    elif x <= d[p][0.50]:
        return 3
    elif x <= d[p][0.75]:
        return 2
    else:
        return 1
```

Define the method to assign the rating to the customers. Calculate the value of recency, frequency and monetary from existing dataset and saved into R, F, M data frame. RFM score is calculated by combining the value of each RFM column, total sum of RFM values. Rating between 1start to 5star is assign to customers on the basis of RFM score, rating labels and length of rating array passes as an arguments in **pd.qcut()** function.

Results and analysis

Apply the RFM class method to calculate the rating of the customers. The fig x below shows the column rfm_rate_customers, calculated through RFM score. Rating is assigned to each customer, based on their RFM score.

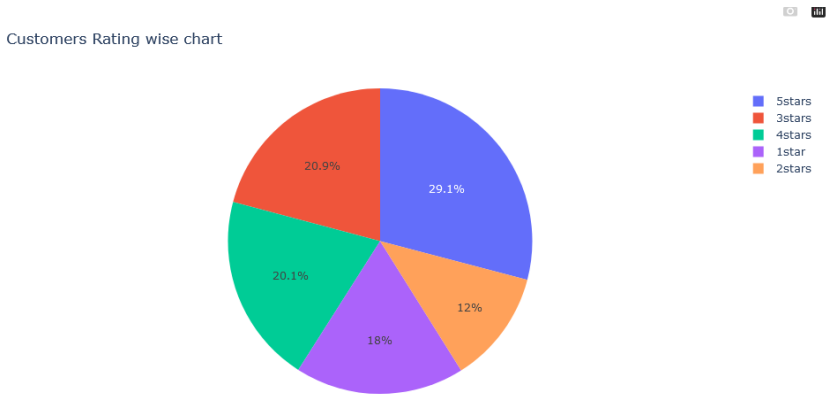
```
data1
```

	index	CustomerID	Recency	Frequency	Monetary	R	F	M	score	rfm_rate_customers
0	0	12346.0	325	1	77183.60	4	4	1	9	3stars
1	1	12747.0	2	103	4196.01	1	1	1	3	5stars
2	2	12748.0	0	4596	33719.73	1	1	1	3	5stars
3	3	12749.0	3	199	4090.88	1	1	1	3	5stars
4	4	12820.0	3	59	942.34	1	2	2	5	5stars
...	...	...	...	...	...	...	...	...	...	...
3916	3916	18280.0	277	10	180.60	4	4	4	12	1star
3917	3917	18281.0	180	7	80.82	4	4	4	12	1star
3918	3918	18282.0	7	12	178.05	1	4	4	9	3stars
3919	3919	18283.0	3	756	2094.88	1	1	1	3	5stars
3920	3920	18287.0	42	70	1837.28	2	2	1	5	5stars

3921 rows × 10 columns

To represent the RFM analysis results, I plot different charts. For instance,

Customers Rating wise chart, pie chart is draw of rating between 1star to 5start. You can differentiate ratings with easily and percentage is clearly representing each segment of rating.

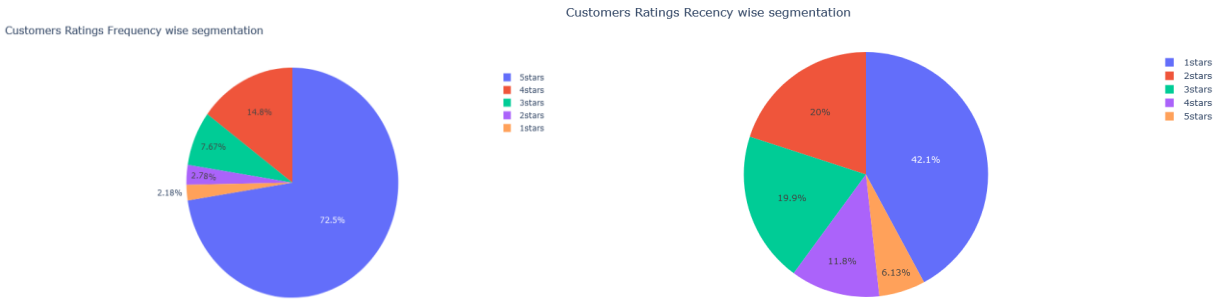


The below code used to plot Frequency, Monetary and Recency charts based on their ratings.

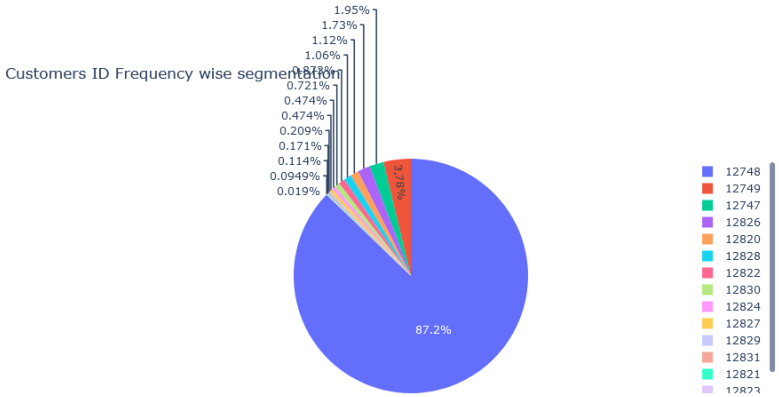
```
[21]:
df_local =  data1.groupby('rfm_rate_customers').agg({"Frequency" : "sum"}).reset_index()
fig = px.pie(df_local , values='Frequency', names="rfm_rate_customers", title='Customers Ratings Frequency wise segmentation
fig.show()

df_local =  data1.groupby('rfm_rate_customers').agg({"Monetary" : "sum"}).reset_index()
fig = px.pie(df_local , values='Monetary', names="rfm_rate_customers", title='Customers  Ratings Monetary wise segmentation'
fig.show()

df_local =  data1.groupby('rfm_rate_customers').agg({"Recency" : "sum"}).reset_index()
fig = px.pie(df_local , values='Recency', names="rfm_rate_customers", title='Customers Ratings Recency wise segmentation')
fig.show()
```



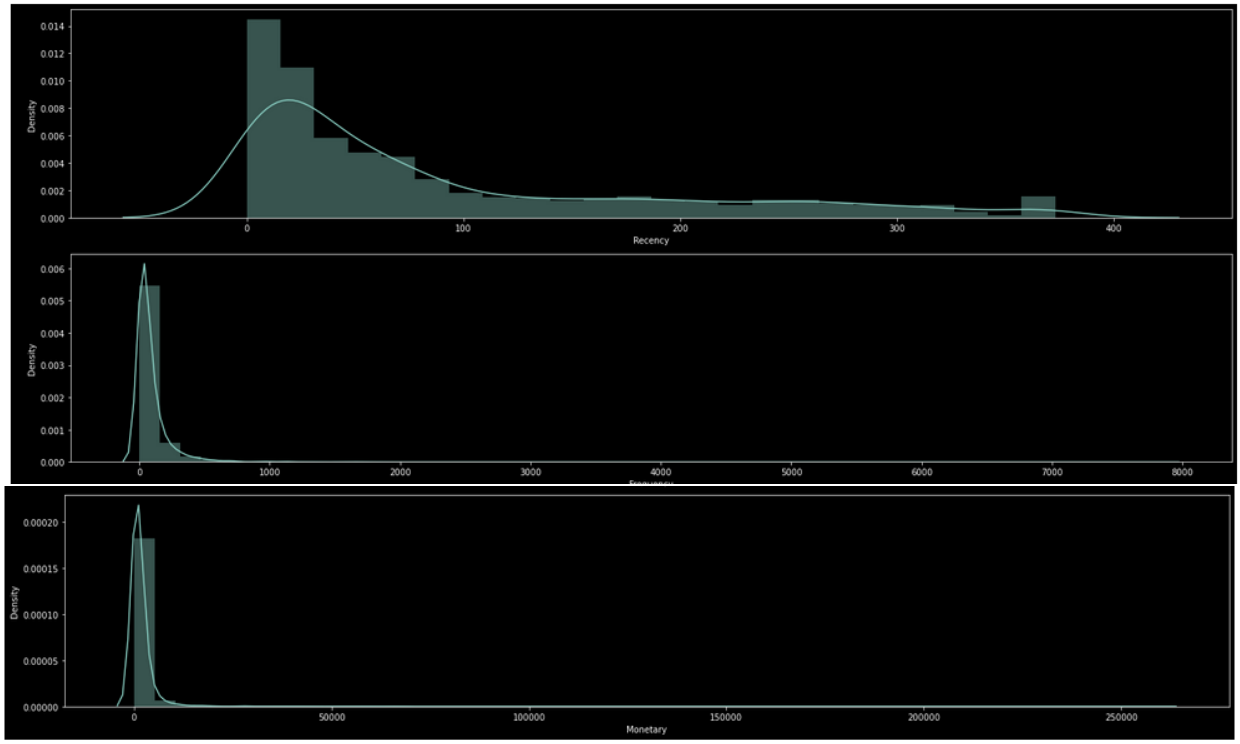
Grouping customer ID's and plot it into frequency wise segmentation. In the below pie chart shows top 15 frequent customers. The most frequently who visited to for shopping his ID is 12748, and he visited 4596 times to the shopping center.



This is the code to plot the density graph, we plot it to show the distribution of recency, frequency, and Monetary dataset.

```
plt.style.use('dark_background')
# plt.style.use('fivethirtyeight')
f,ax = plt.subplots(figsize=(20, 12))
plt.subplot(3, 1, 1); sns.distplot(data1.Recency, label = 'Recency')
plt.subplot(3, 1, 2); sns.distplot(data1.Frequency, label = 'Frequency')
plt.subplot(3, 1, 3); sns.distplot(data1.Monetary, label = 'Monetary')

plt.tight_layout()
plt.show()
```



I also applied k means cluster model on the dataset. Number of clusters used 5, clusters are representing labels. Create recency, frequency, and monetary scatter plots with relationship to each other with clusters.

```
kmeans= Kmeans(data1)
data1 = kmeans.apply_kmeans()
data1.head()
```

	index	CustomerID	Recency	Frequency	Monetary	R	F	M	score	segment	rfm_rate_customers	Clusters
0	0	12346.0	325	1	77183.60	4	4	1	9	9	3stars	4
1	1	12747.0	2	103	4196.01	1	1	1	3	3	5stars	1
2	2	12748.0	0	4596	33719.73	1	1	1	3	3	5stars	1
3	3	12749.0	3	199	4090.88	1	1	1	3	3	5stars	1
4	4	12820.0	3	59	942.34	1	2	2	5	5	5stars	1