RFM Analysis of the Retail Data

Problem statement:

To perform customer segmentation using RFM analysis. The segments can be ordered from most valuable (highest recency, frequency, and value) to least helpful (lowest recency, frequency, and value).

RFM analysis is applied to present data at the aggregate level and is used to segment customers into homogenous groups. Identifying the most valuable RFM segments can capitalize on chance relationships in the data used for this analysis.

As the title suggests, there are three main variables:

R-recency, F-frequency, and M-monetary;

Recency- How recently has the customer made a transaction with us?

Frequency- How frequent is the customer in ordering/buying some product from us?

Monetary- How much does the customer spend on purchasing products from us?

The dataset contains the columns **InvoiceNo, StockCode, Description, Quantity, UnitPrice, CustomerID, and Country.** The description and Customer ID columns contain missing values, so we treated the description with a median and dropped the customer ID with null values. We check for duplicate values and drop the duplicate ones. Our dataset look like this;

```
[9]: data.head()
```

[9]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	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

We are calculating R,F, M for customers who have made a purchase:

```
52]: data['InvoiceDate'] = pd.to_datetime(data['InvoiceDate'])
```

```
53]: # Convert the datetime values to "Month Year" format
data['month_year']=data['InvoiceDate']
data['month_year'] = data['InvoiceDate'].dt.to_period('M')
```

```
54]: data['diff'] = max(data['InvoiceDate']) - data['InvoiceDate']
recency = data.groupby('CustomerID')['diff'].min()
recency = recency.reset_index()
recency.head()
```

```
frequency = data.groupby('CustomerID')['InvoiceDate'].count()
frequency = frequency.reset_index()
frequency.head()
```

```
data['Amount'] = data['Quantity']*data['UnitPrice']
monetary = data.groupby('CustomerID')['Amount'].sum()
monetary = monetary.reset_index()
monetary.tail()
```

```
rfm['Recency_labels']=pd.cut(rfm['Recency'], bins=5,labels=['newest','newer','medium','older','oldest'])
```

```
rfm['Frequency_labels']=pd.cut(rfm['Frequency'], bins=5,labels=['less','lesser','medium','high','more'])
```

```
rfm['Monetary_labels']=pd.cut(rfm['Monetary'], bins=4,labels=['lowest','lower','average','high'])
```

```
rfm['RFM_segment']=rfm['Recency_labels'].astype(str)+rfm['Frequency_labels'].astype(str)+rfm['Monetary_labels'].astype(str)
```

```
recency_dict = {'newest':5,'newer':4,'medium':3,'older':2,'oldest':1}
```

```
frequency_dict = {'more':5,'high':4,'medium':3,'lesser':2,'less':1}
```

```
monetary_dict = {'high':4,'average':3,'lower':2,'lowest':1}
```

```
 = rfm['Recency_labels'].map(recency_dict).astype(int)+rfm['Frequency_labels'].map(frequency_dict).astype(int)+rfm['Frequency_labels'].map(frequency_dict).astype(int)
```

```
rfm.head()
```

	CustomerID	Recency	Frequency	Monetary	Recency_labels	Frequency_labels	Monetary_labels	RFM_segment	RFM_score
0	12346.0	325	2	0.00	oldest	less	lowest	oldestlesslowest	3
1	12347.0	1	182	4310.00	newest	less	lowest	newestlesslowest	7
2	12348.0	74	31	1797.24	newest	less	lowest	newestlesslowest	7
3	12349.0	18	73	1757.55	newest	less	lowest	newestlesslowest	7
4	12350.0	309	17	334.40	oldest	less	lowest	oldestlesslowest	3

```
import numpy as np

# Assuming you have an existing DataFrame named 'rfm' with an 'RFM_score' column
rfm["customer_segment"] = np.select(
    [rfm['RFM_score'] > 10, (10 >= rfm['RFM_score']) & (rfm['RFM_score'] >= 5), rfm['RFM_score'] < 5],
    ["Top Customers", "Medium Value Customer", "Low Value Customer"],
    default="Unknown"
)
```

```
rfm
```

	CustomerID	Recency	Frequency	Monetary	Recency_labels	Frequency_labels	Monetary_labels	RFM_segment	RFM_score	customer_segment
0	12346.0	325	2	0.00	oldest	less	lowest	oldestlesslowest	3	Low Value Customer
1	12347.0	1	182	4310.00	newest	less	lowest	newestlesslowest	7	Medium Value Customer
2	12348.0	74	31	1797.24	newest	less	lowest	newestlesslowest	7	Medium Value Customer
3	12349.0	18	73	1757.55	newest	less	lowest	newestlesslowest	7	Medium Value Customer
4	12350.0	309	17	334.40	oldest	less	lowest	oldestlesslowest	3	Low Value Customer
...	...	...	...	...	...	...	...	...	...	...
4367	18280.0	277	10	180.60	older	less	lowest	olderlesslowest	4	Low Value Customer
4368	18281.0	180	7	80.82	medium	less	lowest	mediumlesslowest	5	Medium Value Customer
4369	18282.0	7	13	176.60	newest	less	lowest	newestlesslowest	7	Medium Value Customer
4370	18283.0	3	721	2045.53	newest	less	lowest	newestlesslowest	7	Medium Value Customer
4371	18287.0	42	70	1837.28	newest	less	lowest	newestlesslowest	7	Medium Value Customer

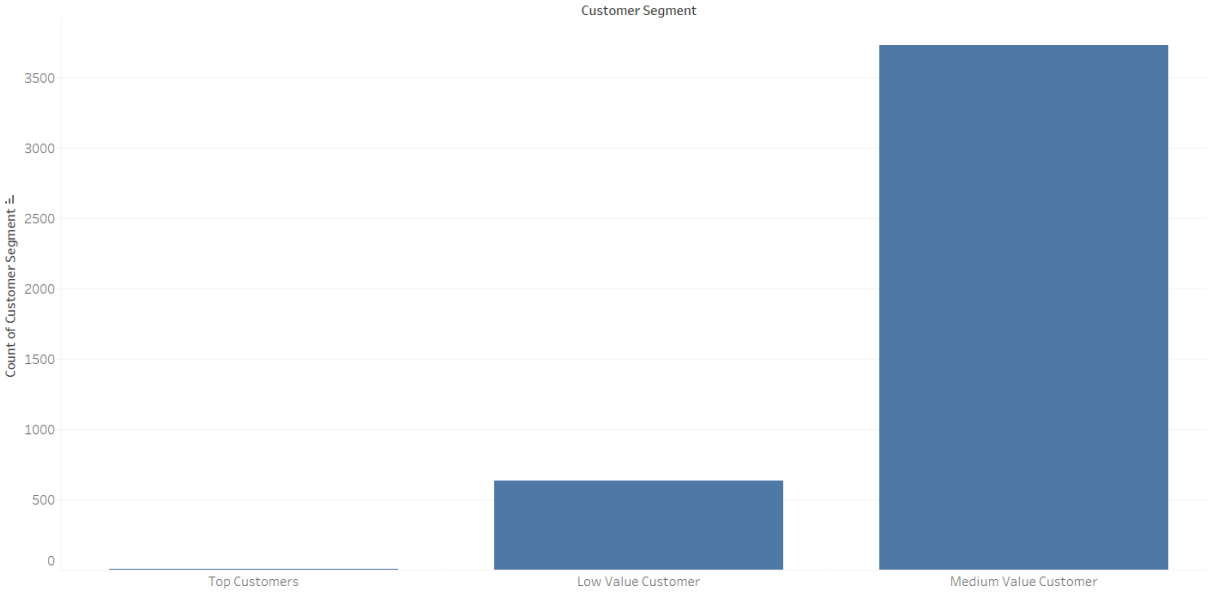
4372 rows × 10 columns

For month_year 2010-12, WHITE HANGING HEART T-LIGHT HOLDER and 72 SWEETHEART FAIRY CAKE CASES had high sales, and for month_year 2011-12, 6 RIBBONS ELEGANT CHRISTMAS had high sales.

The retention rate measures customers who return to purchase more products or continue using the services. The maximum difference in recency is for CustomerID 18287.0 of 373.0 days, and the minimum difference is for CustomerID 12346.0 of 0 days.

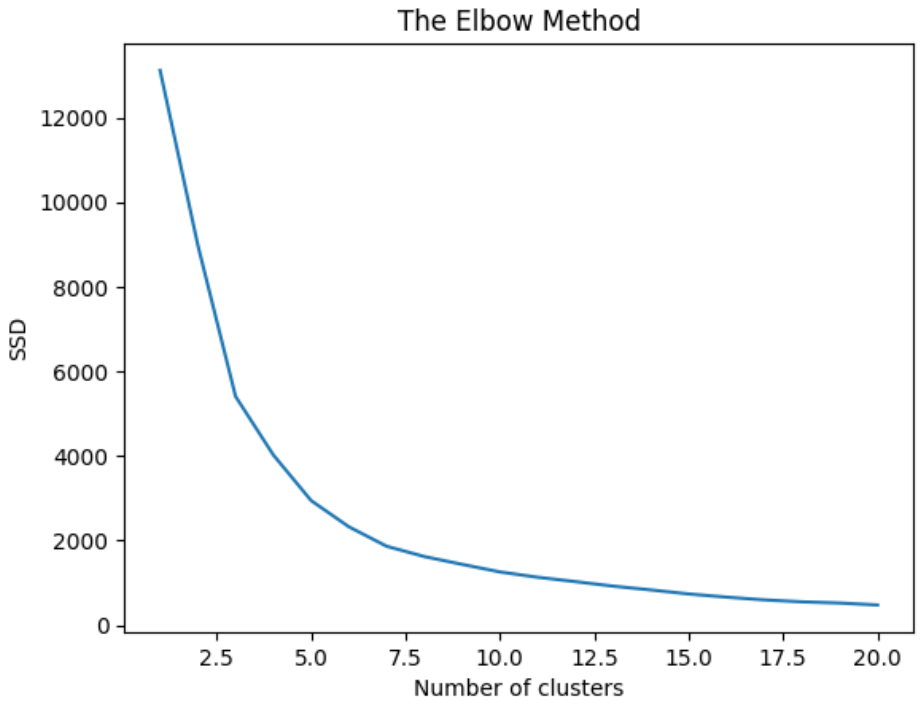
The highest Monetary (total amount of money a customer spent) amount was for customer ID 14646.0, with an amount of 279489.02, and the lowest monetary (total amount of money a customer spent) amount was for customer ID 17448.0, with an amount of -4287.63.

The lowest frequency (number of purchases) recorded is 1 for customer ID 12503.0 and the highest frequency (number of purchases) recorded is 7812 for customer ID 17841.0



We created clusters using the k-means algorithm(an unsupervised learning algorithm),

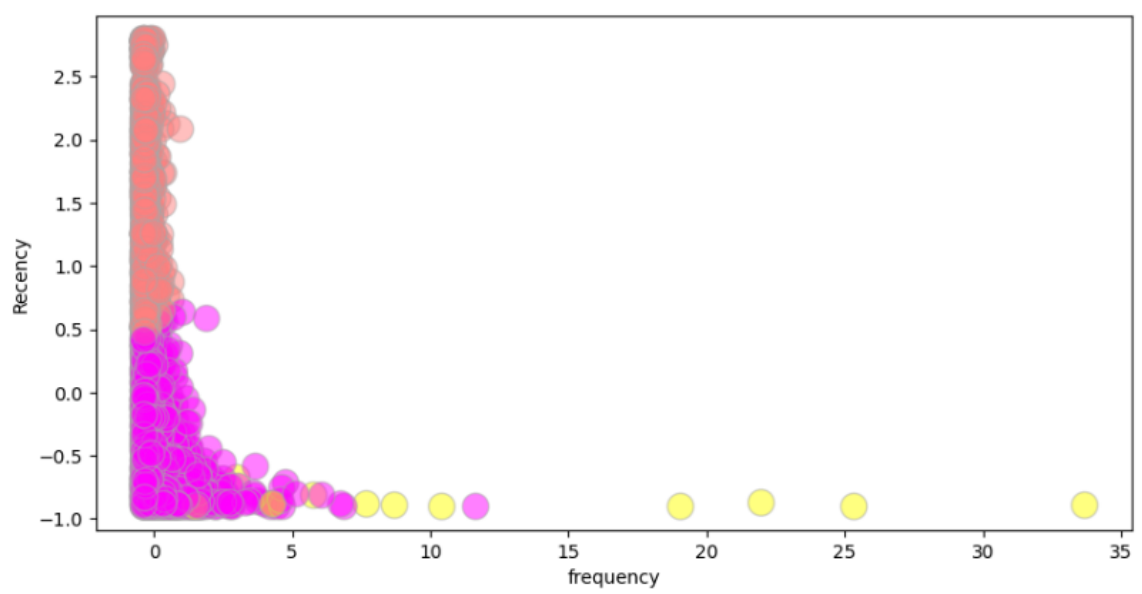
```
58]: ssd =[]

for num_clusters in range(1,21):
    kmeans = KMeans(n_clusters = num_clusters, max_iter=100)
    kmeans.fit(rfm_normalized)

    ssd.append(kmeans.inertia_)
plt.plot(range(1,21), ssd)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('SSD')
plt.show()
```



The optimum number of clusters to formed is 3.

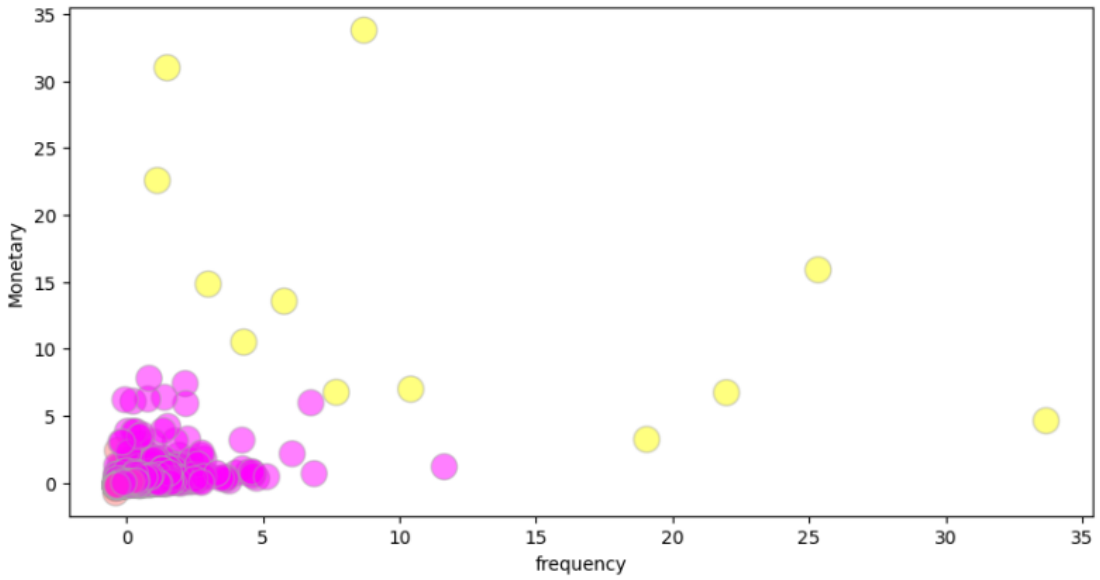
Here, the scatter plots are:

```
plt.rcParams["figure.figsize"] = (10,5)
plt.scatter(rfm_normalized['Frequency'],rfm_normalized['Recency'],
            c=rfm_normalized['cluster'],
            s=200,
            cmap='spring',
            alpha=0.5,
            edgecolor='darkgrey')
plt.xlabel("frequency")
plt.ylabel("Recency")
plt.show()
```

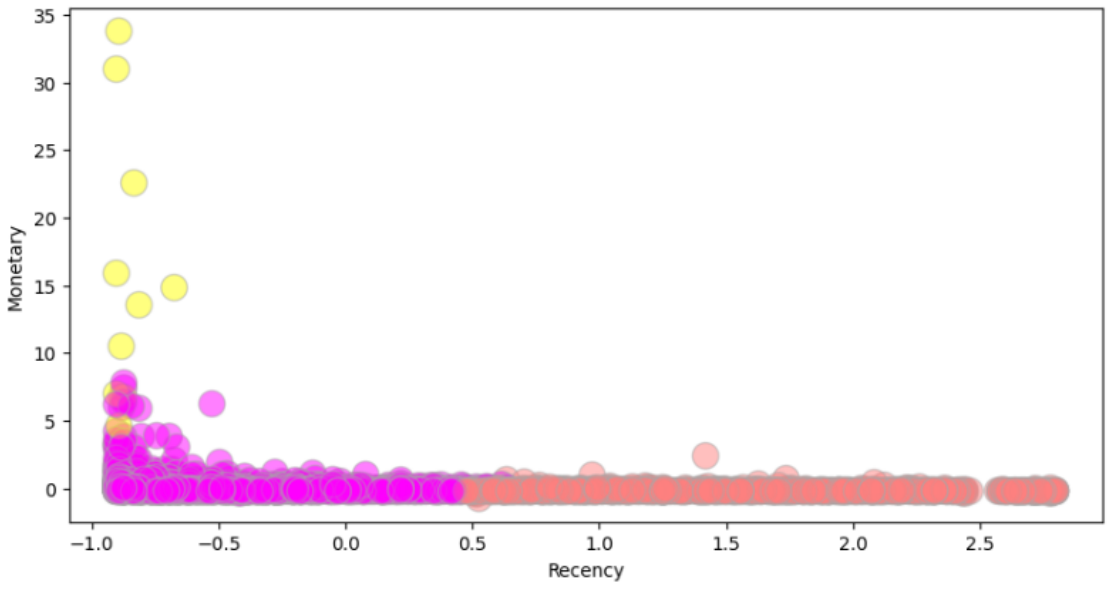


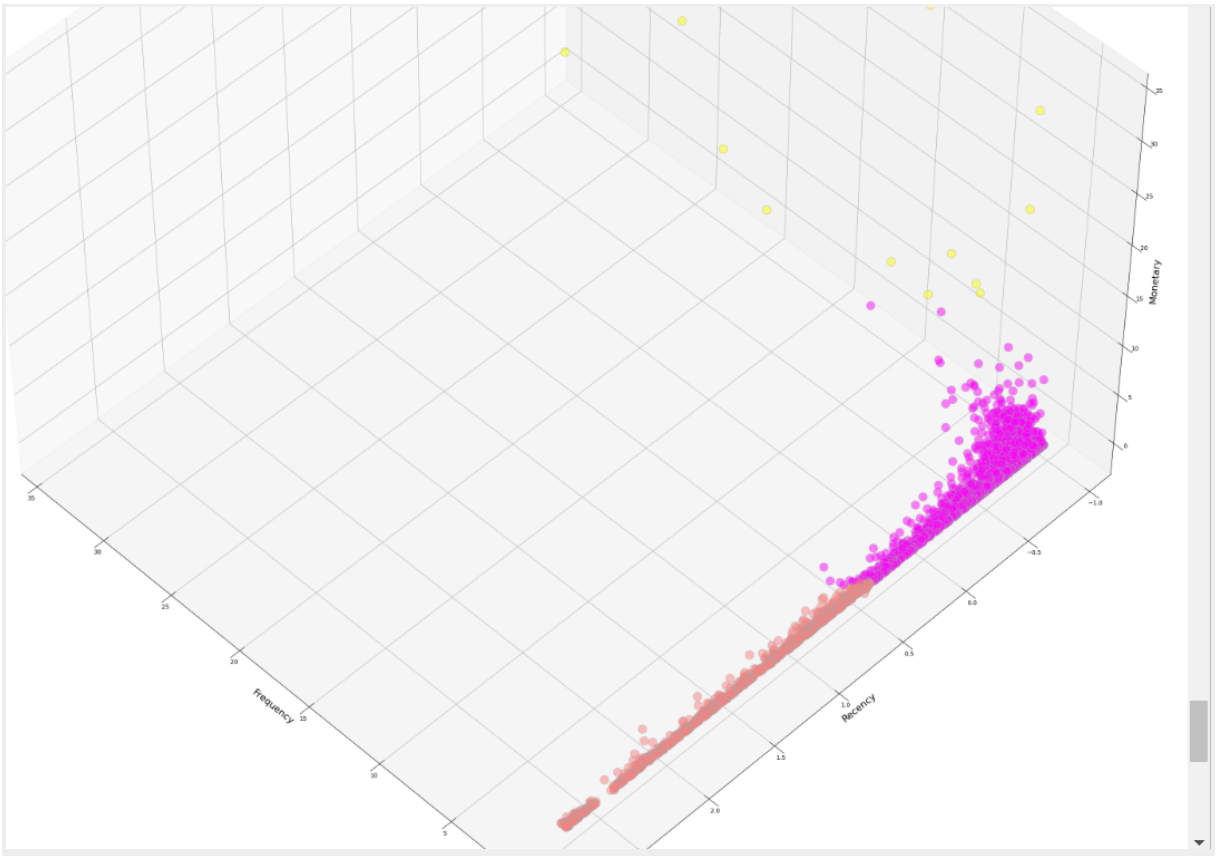
frequency

8]:
```
plt.scatter(rfm_normalized['Frequency'],rfm_normalized['Monetary'],
            c=rfm_normalized['cluster'],
            s=200,
            cmap='spring',
            alpha=0.5,
            edgecolor='darkgrey')
plt.xlabel("frequency")
plt.ylabel("Monetary")
plt.show()
```



89]:
```
plt.scatter(rfm_normalized['Recency'],rfm_normalized['Monetary'],
            c=rfm_normalized['cluster'],
            s=200,
            cmap='spring',
            alpha=0.5,
            edgecolor='darkgrey')
plt.xlabel("Recency")
plt.ylabel("Monetary")
plt.show()
```

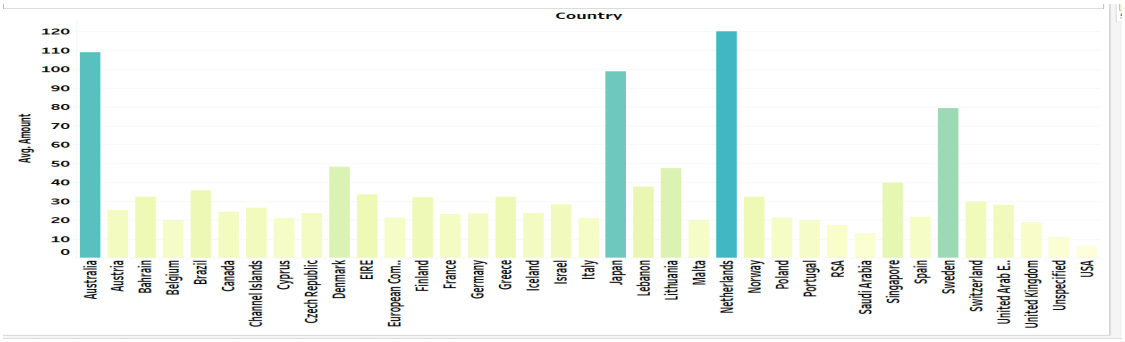




Here, in the scatter plot for Recency, frequency, and monetary value, the pink cluster represents the customers who are not that recent and have less monetary value, even though they tend to buy frequently. Most recent customers have medium monetary value and frequency, and a light cluster with very few recent customers has high monetary value as well as buying more frequently.

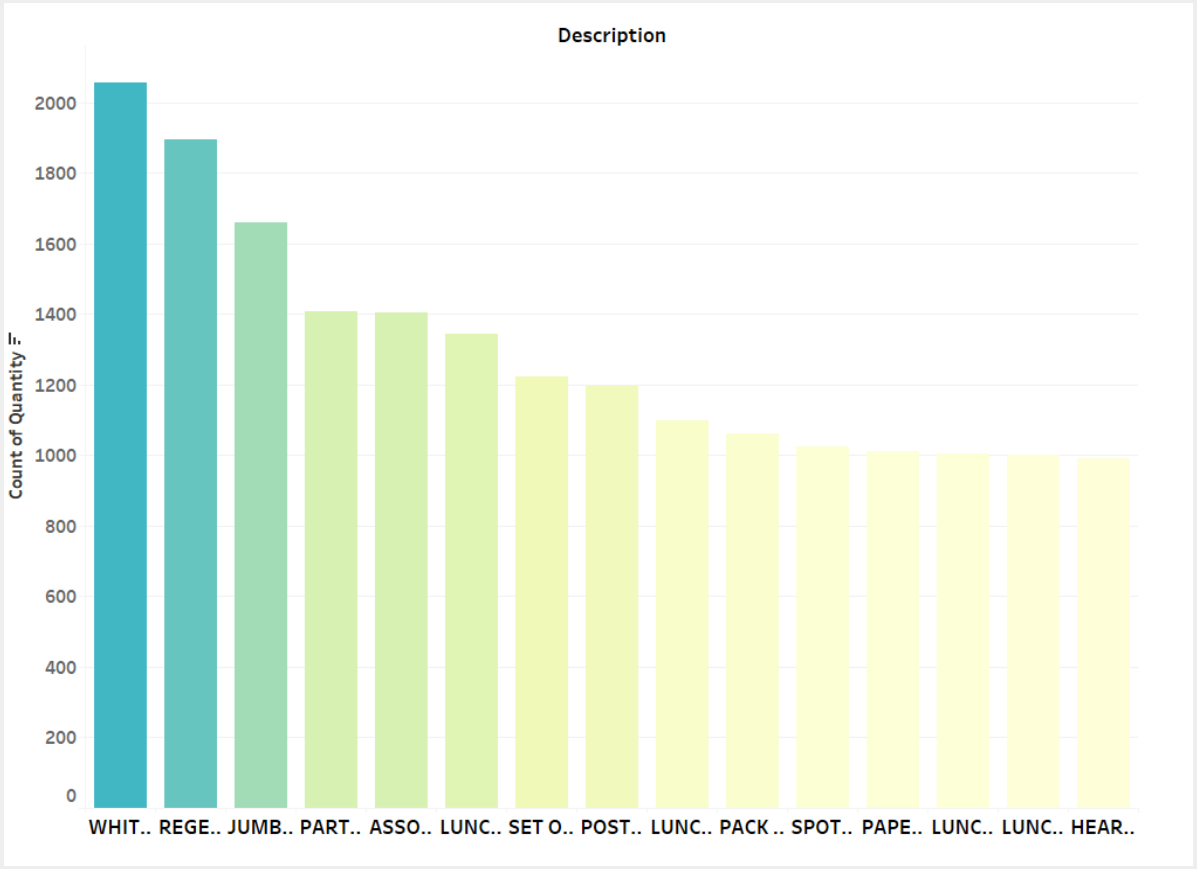
Here, are some visualizations from Tableau:

a) Country-wise analysis to demonstrate Average spending. Use a bar chart to show monthly figures.

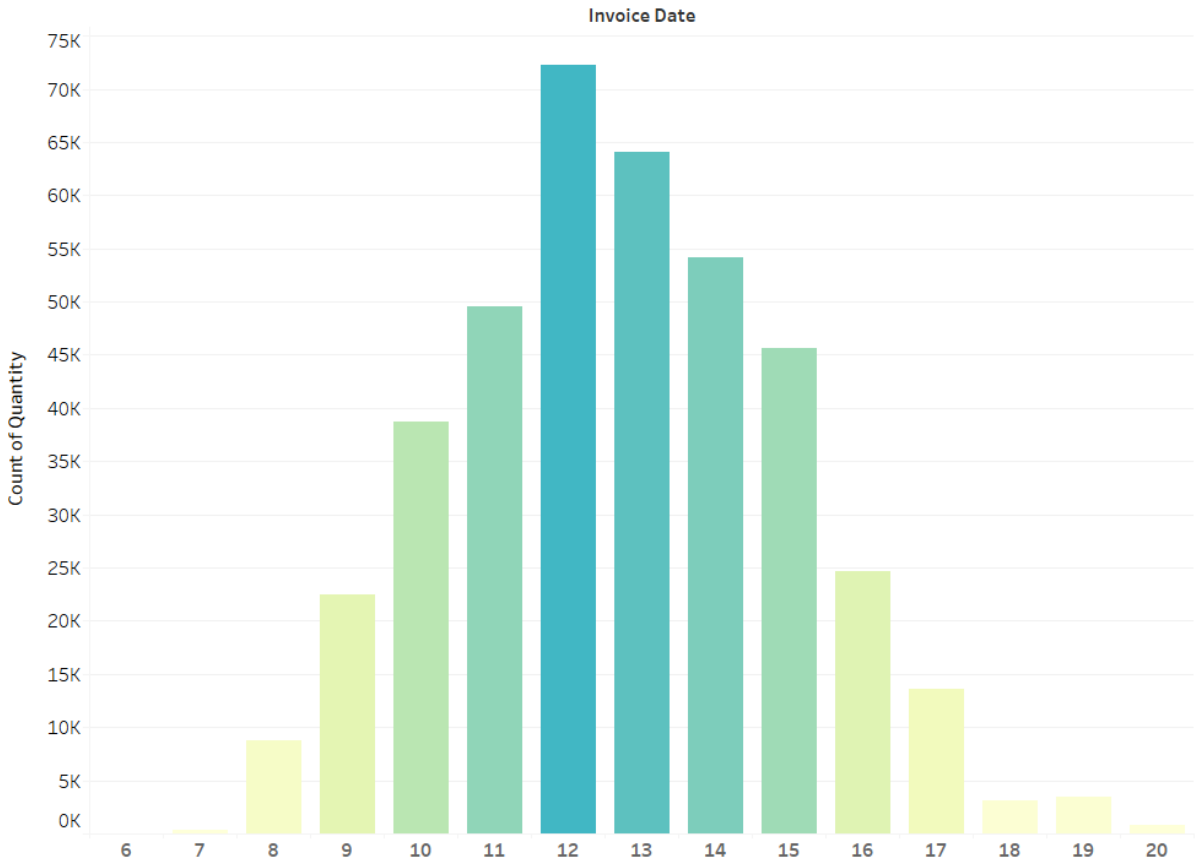


Average spending is higher in the Netherlands, Australia, Japan, and Sweden.

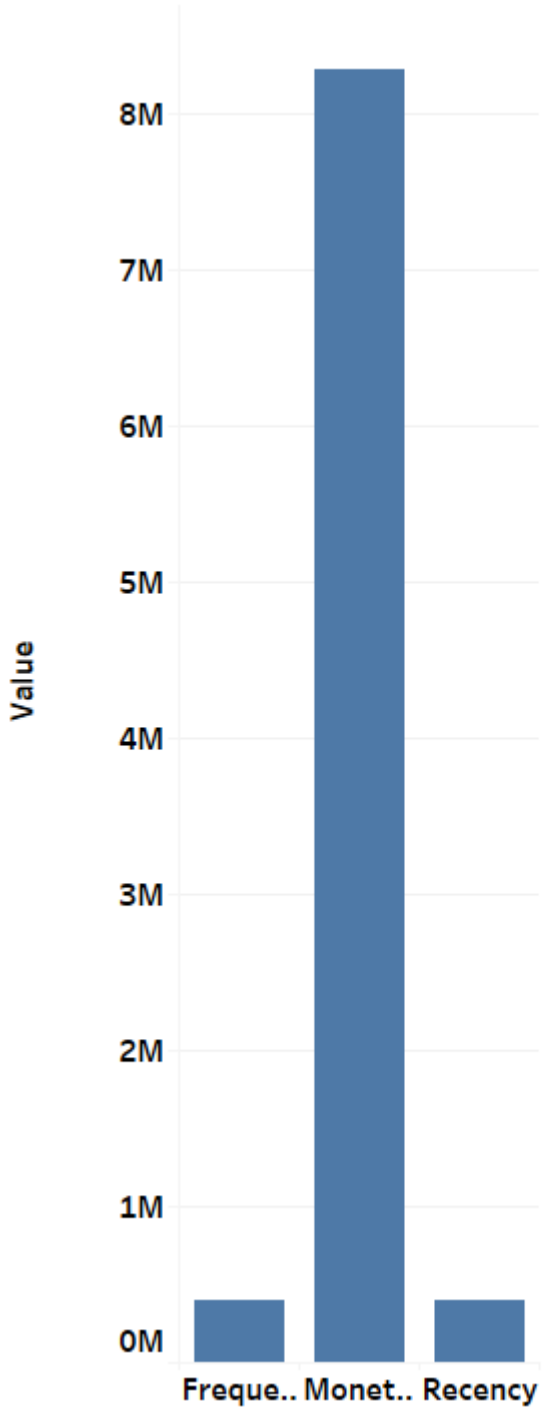
b) A bar graph of top 15 products, which are mostly ordered by the users, to show the number of products sold.



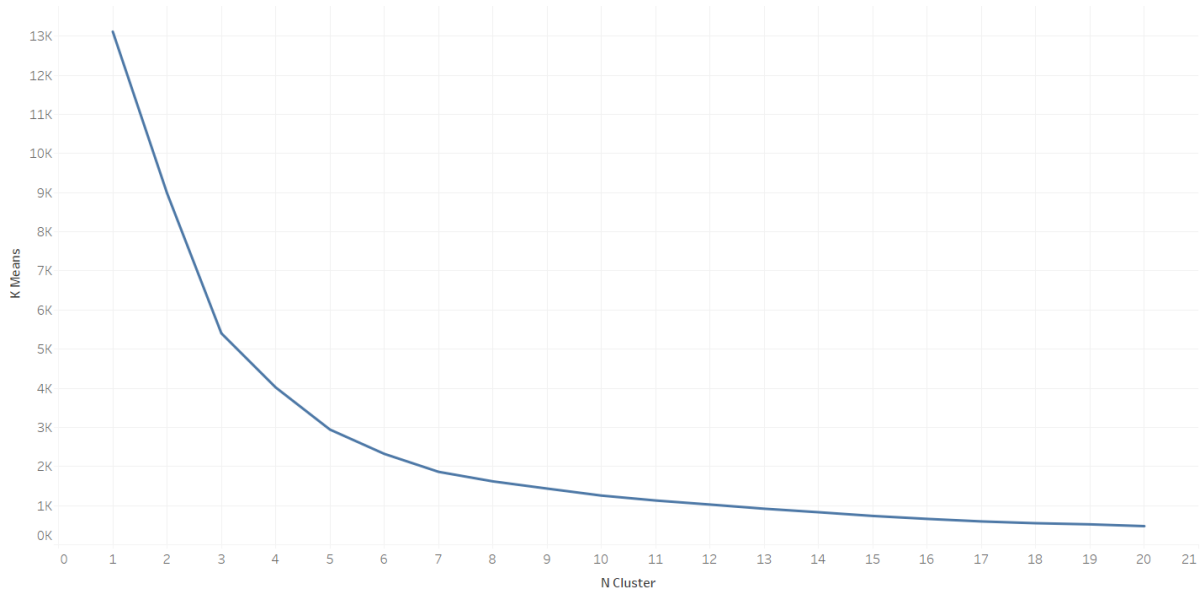
c) Bar graph to show the count of orders Vs. hours throughout the day. What are the peak hours per your chart?



d) Plot the distribution of RFM values using histograms and frequency-charts



e) Plot error(cost) vs no of clusters selected

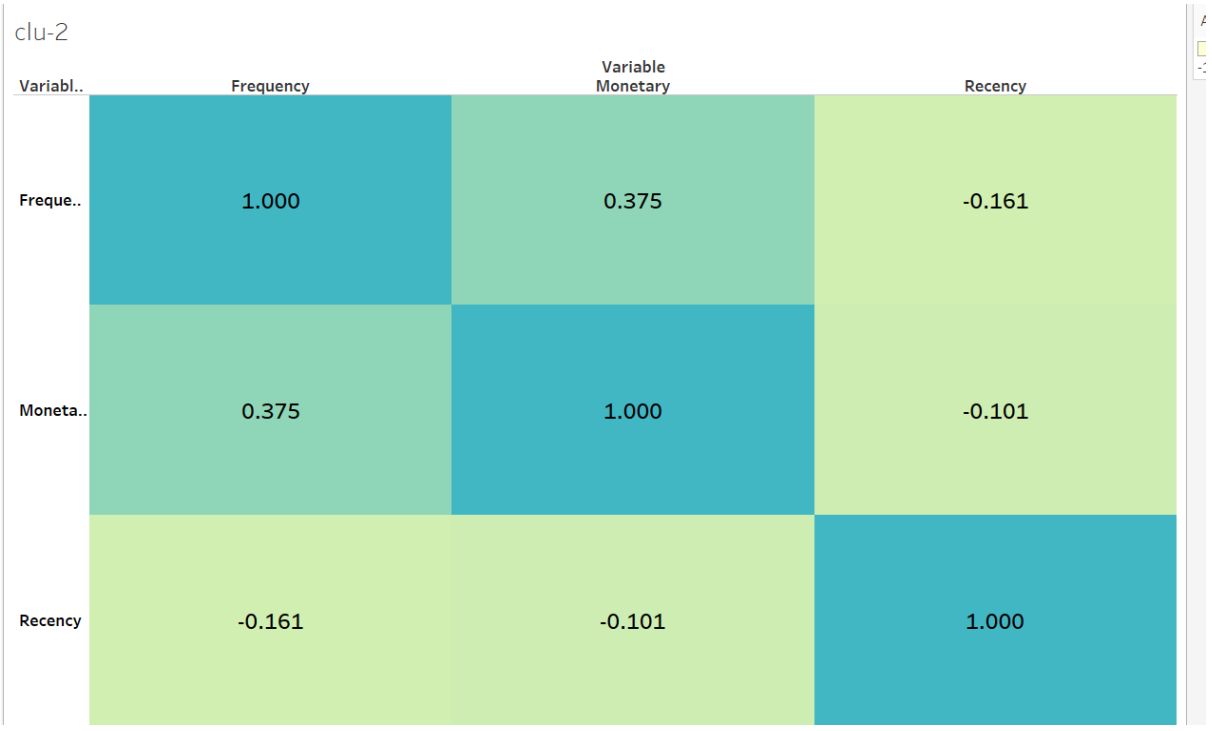


f) Visualize to compare the RFM values of the clusters using heatmap

Variable ..	Frequency	Monetary	Recency
Frequency	1.000	0.472	-0.276
Monetary	0.472	1.000	-0.205
Recency	-0.276	-0.205	1.000

As frequency increases, the monetary value also increases, and recency and frequency are inversely related. Recency and monetary value are inversely related in cluster 0.

As frequency increases, the monetary value decreases, and recency and frequency are inversely related. Recency and monetary value are directly related in cluster 1.



As frequency increases, the monetary value also increases, and recency and frequency are inversely related. Recency and monetary value are inversely related in cluster 2.