

```
[1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[2]: df=pd.read_csv('OnlineRetail.csv', encoding = 'unicode_escape')
df
```

| | InvoiceNo | StockCode | Description | Quantity | InvoiceDate | UnitPrice | CustomerID | Country |
|--------|-----------|-----------|-------------------------------------|----------|-----------------|-----------|------------|----------------|
| 0 | 536365 | 85123A | WHITE HANGING HEART T-LIGHT HOLDER | 6 | 12/1/2010 8:26 | 2.55 | 17850.0 | United Kingdom |
| 1 | 536365 | 71053 | WHITE METAL LANTERN | 6 | 12/1/2010 8:26 | 3.39 | 17850.0 | United Kingdom |
| 2 | 536365 | 84406B | CREAM CUPID HEARTS COAT HANGER | 8 | 12/1/2010 8:26 | 2.75 | 17850.0 | United Kingdom |
| 3 | 536365 | 84029G | KNITTED UNION FLAG HOT WATER BOTTLE | 6 | 12/1/2010 8:26 | 3.39 | 17850.0 | United Kingdom |
| 4 | 536365 | 84029E | RED WOOLLY HOTTIE WHITE HEART. | 6 | 12/1/2010 8:26 | 3.39 | 17850.0 | United Kingdom |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 541904 | 581587 | 22613 | PACK OF 20 SPACEBOY NAPKINS | 12 | 12/9/2011 12:50 | 0.85 | 12680.0 | France |
| 541905 | 581587 | 22899 | CHILDREN'S APRON DOLLY GIRL | 6 | 12/9/2011 12:50 | 2.10 | 12680.0 | France |
| 541906 | 581587 | 23254 | CHILDRENS CUTLERY DOLLY GIRL | 4 | 12/9/2011 12:50 | 4.15 | 12680.0 | France |
| 541907 | 581587 | 23255 | CHILDRENS CUTLERY CIRCUS PARADE | 4 | 12/9/2011 12:50 | 4.15 | 12680.0 | France |
| 541908 | 581587 | 22138 | BAKING SET 9 PIECE RETROSPOT | 3 | 12/9/2011 12:50 | 4.95 | 12680.0 | France |

541909 rows × 8 columns

```
[3]: df.dtypes
```

```
[3]: InvoiceNo      object
StockCode       object
Description     object
Quantity        int64
InvoiceDate    object
UnitPrice      float64
CustomerID     float64
Country        object
dtype: object
```

```
[4]: df.tail(5)
```

```
[4]: currentDate = pd.to_datetime("2012-01-01")
df['InvoiceDate']=pd.to_datetime(df["InvoiceDate"])
```

```
[5]: df
```

| | 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 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 541904 | 581587 | 22613 | PACK OF 20 SPACEBOY NAPKINS | 12 | 2011-12-09 12:50:00 | 0.85 | 12680.0 | France |
| 541905 | 581587 | 22899 | CHILDREN'S APRON DOLLY GIRL | 6 | 2011-12-09 12:50:00 | 2.10 | 12680.0 | France |
| 541906 | 581587 | 23254 | CHILDRENS CUTLERY DOLLY GIRL | 4 | 2011-12-09 12:50:00 | 4.15 | 12680.0 | France |
| 541907 | 581587 | 23255 | CHILDRENS CUTLERY CIRCUS PARADE | 4 | 2011-12-09 12:50:00 | 4.15 | 12680.0 | France |
| 541908 | 581587 | 22138 | BAKING SET 9 PIECE RETROSPOT | 3 | 2011-12-09 12:50:00 | 4.95 | 12680.0 | France |

541909 rows × 8 columns

```
[7]: currentDate
```

```
[7]: Timestamp('2012-01-01 00:00:00')
```

```
[6]: df.dtypes
```

```
[6]: InvoiceNo      object
StockCode       object
Description     object
Quantity        int64
InvoiceDate    datetime64[ns]
UnitPrice      float64
```

```
[1]:      type
CustomerID      float64
Country         object
dtype: object

[9]: maxdate = df.groupby(["CustomerID"]).max()[['InvoiceDate']]
```

```
[9]:      InvoiceDate
```

| CustomerID | InvoiceDate |
|------------|---------------------|
| 12346.0 | 2011-01-18 10:17:00 |
| 12347.0 | 2011-12-07 15:52:00 |
| 12348.0 | 2011-09-25 13:13:00 |
| 12349.0 | 2011-11-21 09:51:00 |
| 12350.0 | 2011-02-02 16:01:00 |
| ... | ... |
| 18280.0 | 2011-03-07 09:52:00 |
| 18281.0 | 2011-06-12 10:53:00 |
| 18282.0 | 2011-12-02 11:43:00 |
| 18283.0 | 2011-12-06 12:02:00 |
| 18287.0 | 2011-10-28 09:29:00 |

4372 rows × 1 columns

```
[10]: (currentDate - maxdate['InvoiceDate']).dt.days
```

```
[10]: CustomerID
12346.0    347
12347.0     24
12348.0     97
12349.0     40
12350.0    332
...
18280.0    299
18281.0    202
18282.0     29
18283.0     25
18287.0     64
Name: InvoiceDate, Length: 4372, dtype: int64
```

```
[11]: maxdate['InvoiceAge'] = (currentDate - maxdate['InvoiceDate']).dt.days
```

```
[11]:      InvoiceDate  InvoiceAge
```

| CustomerID | InvoiceDate | InvoiceAge |
|------------|---------------------|------------|
| 12346.0 | 2011-01-18 10:17:00 | 347 |
| 12347.0 | 2011-12-07 15:52:00 | 24 |
| 12348.0 | 2011-09-25 13:13:00 | 97 |
| 12349.0 | 2011-11-21 09:51:00 | 40 |
| 12350.0 | 2011-02-02 16:01:00 | 332 |
| ... | ... | ... |
| 18280.0 | 2011-03-07 09:52:00 | 299 |
| 18281.0 | 2011-06-12 10:53:00 | 202 |
| 18282.0 | 2011-12-02 11:43:00 | 29 |
| 18283.0 | 2011-12-06 12:02:00 | 25 |
| 18287.0 | 2011-10-28 09:29:00 | 64 |

4372 rows × 2 columns

```
[12]: #recency
recency = maxdate.drop('InvoiceDate', axis=1)
recency
```

```
[12]:      InvoiceAge
```

| CustomerID | InvoiceAge |
|------------|------------|
| 12346.0 | 347 |
| 12347.0 | 24 |
| 12348.0 | 97 |
| 12349.0 | 40 |
| 12350.0 | 332 |
| ... | ... |

```

18280.0    299
18281.0    202
18282.0    29
18283.0    25
18287.0    64

```

4372 rows × 1 columns

```
[13]: freq=df.drop_duplicates(subset="InvoiceNo")
freq
```

| | 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 |
| 7 | 536366 | 22633 | HAND WARMER UNION JACK | 6 | 2010-12-01 08:28:00 | 1.85 | 17850.0 | United Kingdom |
| 9 | 536367 | 84879 | ASSORTED COLOUR BIRD ORNAMENT | 32 | 2010-12-01 08:34:00 | 1.69 | 13047.0 | United Kingdom |
| 21 | 536368 | 22960 | JAM MAKING SET WITH JARS | 6 | 2010-12-01 08:34:00 | 4.25 | 13047.0 | United Kingdom |
| 25 | 536369 | 21756 | BATH BUILDING BLOCK WORD | 3 | 2010-12-01 08:35:00 | 5.95 | 13047.0 | United Kingdom |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 541865 | 581583 | 20725 | LUNCH BAG RED RETROSPOT | 40 | 2011-12-09 12:23:00 | 1.45 | 13777.0 | United Kingdom |
| 541867 | 581584 | 20832 | RED FLOCK LOVE HEART PHOTO FRAME | 72 | 2011-12-09 12:25:00 | 0.72 | 13777.0 | United Kingdom |
| 541869 | 581585 | 22481 | BLACK TEA TOWEL CLASSIC DESIGN | 12 | 2011-12-09 12:31:00 | 0.39 | 15804.0 | United Kingdom |
| 541890 | 581586 | 22061 | LARGE CAKE STAND HANGING STRAWBERRY | 8 | 2011-12-09 12:49:00 | 2.95 | 13113.0 | United Kingdom |
| 541894 | 581587 | 22631 | CIRCUS PARADE LUNCH BOX | 12 | 2011-12-09 12:50:00 | 1.95 | 12680.0 | France |

25900 rows × 8 columns

```
[14]: freq = freq.groupby(["CustomerID"]).count()[["InvoiceNo"]]
freq
```

| | InvoiceNo |
|----------------|-----------|
| CustomerID | |
| 12346.0 | 2 |
| 12347.0 | 7 |
| 12348.0 | 4 |
| 12349.0 | 1 |
| 12350.0 | 1 |
| ... | ... |
| 18280.0 | 1 |
| 18281.0 | 1 |
| 18282.0 | 3 |
| 18283.0 | 16 |
| 18287.0 | 3 |

4372 rows × 1 columns

```
[15]: #monetary
df["total"] = df["Quantity"]*df["UnitPrice"]
```

```
[16]: df
```

| | InvoiceNo | StockCode | Description | Quantity | InvoiceDate | UnitPrice | CustomerID | Country | total |
|---------------|-----------|-----------|-------------------------------------|----------|---------------------|-----------|------------|----------------|-------|
| 0 | 536365 | 85123A | WHITE HANGING HEART T-LIGHT HOLDER | 6 | 2010-12-01 08:26:00 | 2.55 | 17850.0 | United Kingdom | 15.30 |
| 1 | 536365 | 71053 | WHITE METAL LANTERN | 6 | 2010-12-01 08:26:00 | 3.39 | 17850.0 | United Kingdom | 20.34 |
| 2 | 536365 | 84406B | CREAM CUPID HEARTS COAT HANGER | 8 | 2010-12-01 08:26:00 | 2.75 | 17850.0 | United Kingdom | 22.00 |
| 3 | 536365 | 84029G | KNITTED UNION FLAG HOT WATER BOTTLE | 6 | 2010-12-01 08:26:00 | 3.39 | 17850.0 | United Kingdom | 20.34 |
| 4 | 536365 | 84029E | RED WOOLLY HOTTIE WHITE HEART. | 6 | 2010-12-01 08:26:00 | 3.39 | 17850.0 | United Kingdom | 20.34 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 541904 | 581587 | 22613 | PACK OF 20 SPACEBOY NAPKINS | 12 | 2011-12-09 12:50:00 | 0.85 | 12680.0 | France | 10.20 |
| 541905 | 581587 | 22899 | CHILDREN'S APRON DOLLY GIRL | 6 | 2011-12-09 12:50:00 | 2.10 | 12680.0 | France | 12.60 |
| 541906 | 581587 | 23254 | CHILDRENS CUTLERY DOLLY GIRL | 4 | 2011-12-09 12:50:00 | 4.15 | 12680.0 | France | 16.60 |
| 541907 | 581587 | 23255 | CHILDRENS CUTLERY CIRCUS PARADE | 4 | 2011-12-09 12:50:00 | 4.15 | 12680.0 | France | 16.60 |
| 541908 | 581587 | 22138 | BAKING SET 9 PIECE RETROSPOT | 3 | 2011-12-09 12:50:00 | 4.95 | 12680.0 | France | 14.85 |

541909 rows × 9 columns

```
[17]: money=df.groupby(["CustomerID"])[["total"]].sum()  
money
```

```
[17]:      total
```

CustomerID

| | |
|---------|---------|
| 12346.0 | 0.00 |
| 12347.0 | 4310.00 |
| 12348.0 | 1797.24 |
| 12349.0 | 1757.55 |
| 12350.0 | 334.40 |
| ... | ... |
| 18280.0 | 180.60 |
| 18281.0 | 80.82 |
| 18282.0 | 176.60 |
| 18283.0 | 2094.88 |
| 18287.0 | 1837.28 |

4372 rows × 1 columns

```
[18]: df[df['CustomerID']==12346.0]
```

```
[18]:   InvoiceNo StockCode          Description  Quantity  InvoiceDate  UnitPrice  CustomerID  Country  total  
61619      541431    23166 MEDIUM CERAMIC TOP STORAGE JAR    74215 2011-01-18 10:01:00     1.04  12346.0 United Kingdom  77183.6  
61624      C541433    23166 MEDIUM CERAMIC TOP STORAGE JAR   -74215 2011-01-18 10:17:00     1.04  12346.0 United Kingdom -77183.6
```

```
[19]: RFM=pd.concat([recency,freq,money], axis=1)  
RFM
```

```
[19]:      InvoiceAge  InvoiceNo  total
```

CustomerID

| | | | |
|---------|-----|-----|---------|
| 12346.0 | 347 | 2 | 0.00 |
| 12347.0 | 24 | 7 | 4310.00 |
| 12348.0 | 97 | 4 | 1797.24 |
| 12349.0 | 40 | 1 | 1757.55 |
| 12350.0 | 332 | 1 | 334.40 |
| ... | ... | ... | ... |
| 18280.0 | 299 | 1 | 180.60 |
| 18281.0 | 202 | 1 | 80.82 |
| 18282.0 | 29 | 3 | 176.60 |
| 18283.0 | 25 | 16 | 2094.88 |
| 18287.0 | 64 | 3 | 1837.28 |

4372 rows × 3 columns

```
[20]: RFM.columns = ['Recency', 'Frequency', 'Monetary']  
RFM
```

```
[20]:      Recency  Frequency  Monetary
```

CustomerID

| | | | |
|---------|-----|-----|---------|
| 12346.0 | 347 | 2 | 0.00 |
| 12347.0 | 24 | 7 | 4310.00 |
| 12348.0 | 97 | 4 | 1797.24 |
| 12349.0 | 40 | 1 | 1757.55 |
| 12350.0 | 332 | 1 | 334.40 |
| ... | ... | ... | ... |
| 18280.0 | 299 | 1 | 180.60 |
| 18281.0 | 202 | 1 | 80.82 |
| 18282.0 | 29 | 3 | 176.60 |
| 18283.0 | 25 | 16 | 2094.88 |
| 18287.0 | 64 | 3 | 1837.28 |

4372 rows × 3 columns

```
[21]: from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
scaled=scaler.fit_transform(RFM)

[22]: scaled
```

```
[22]: array([[ 2.31656788, -0.32936215, -0.23100099],
       [-0.88904975,  0.20610242,  0.29343167],
       [-0.16456032, -0.11517632, -0.01231622],
       ...,
       [-0.83942719, -0.22226923, -0.20951263],
       [-0.87912524,  1.16993863,  0.02390005],
       [-0.49206924, -0.22226923, -0.00744423]])
```

```
[23]: scaled.shape
```

```
[23]: (4372, 3)
```

```
[24]: from sklearn.cluster import KMeans
```

To find out the value of K

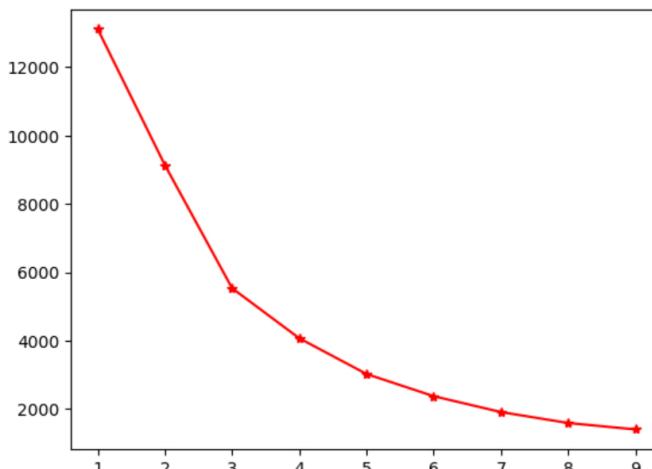
```
[25]: k_range = range(1,10)
sse = []
for k in k_range:
    km = KMeans(n_clusters=k)
    km.fit(scaled)
    sse.append(km.inertia_)
```

```
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)
```

```
[26]: sse
```

```
[26]: [13116.000000000002,
      9134.642381209183,
      5530.931653683228,
      4065.047158594225,
      3020.2733516539765,
      2370.1546197959087,
      1906.3743552963688,
      1585.5649930710322,
      1397.970817541668]
```

```
[27]: plt.plot(k_range, sse, 'r-*')
plt.xlabel ='k'
plt.ylabel = 'Sum of Squared Error'
```



Object Creation and data fitting

```
[28]: kmeans=KMeans(n_clusters=3)
kmeans.fit(scaled)

C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4.
Set the value of `n_init` explicitly to suppress the warning
super().__check_params_vs_input(X, default_n_init=10)
```

```
[28]: KMeans(n_clusters=3)
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

```
[30]: kmeans.labels_
```

```
[30]: array([2, 0, 0, ..., 0, 0, 0])
```

```
[31]: kmeans.labels_.shape
```

```
[31]: (4372,)
```

```
[32]: RFM["Clusters"]=(kmeans.labels_+1)
```

```
[33]: RFM
```

```
[33]:      Recency  Frequency  Monetary  Clusters
```

CustomerID

| | | | | |
|---------|-----|-----|---------|-----|
| 12346.0 | 347 | 2 | 0.00 | 3 |
| 12347.0 | 24 | 7 | 4310.00 | 1 |
| 12348.0 | 97 | 4 | 1797.24 | 1 |
| 12349.0 | 40 | 1 | 1757.55 | 1 |
| 12350.0 | 332 | 1 | 334.40 | 3 |
| ... | ... | ... | ... | ... |
| 18280.0 | 299 | 1 | 180.60 | 3 |
| 18281.0 | 202 | 1 | 80.82 | 3 |
| 18282.0 | 29 | 3 | 176.60 | 1 |
| 18283.0 | 25 | 16 | 2094.88 | 1 |
| 18287.0 | 64 | 3 | 1837.28 | 1 |

4372 rows × 4 columns

```
[34]: RFM.Clusters.unique()
```

```
[34]: array([3, 1, 2])
```

```
[35]: RFM[RFM.Clusters==1]
```

```
[35]:      Recency  Frequency  Monetary  Clusters
```

CustomerID

| | | | | |
|---------|-----|-----|---------|-----|
| 12347.0 | 24 | 7 | 4310.00 | 1 |
| 12348.0 | 97 | 4 | 1797.24 | 1 |
| 12349.0 | 40 | 1 | 1757.55 | 1 |
| 12352.0 | 58 | 11 | 1545.41 | 1 |
| 12356.0 | 44 | 3 | 2811.43 | 1 |
| ... | ... | ... | ... | ... |
| 18277.0 | 80 | 2 | 97.63 | 1 |
| 18278.0 | 95 | 1 | 173.90 | 1 |
| 18282.0 | 29 | 3 | 176.60 | 1 |
| 18283.0 | 25 | 16 | 2094.88 | 1 |
| 18287.0 | 64 | 3 | 1837.28 | 1 |

3242 rows × 4 columns

```
[36]: RFM[RFM.Clusters==2]
```

```
[36]:      Recency  Frequency  Monetary  Clusters
```

CustomerID

| | | | | |
|---------|----|-----|-----------|---|
| 12415.0 | 46 | 26 | 123725.45 | 2 |
| 12748.0 | 22 | 224 | 29072.10 | 2 |
| 12971.0 | 25 | 89 | 10930.26 | 2 |
| 13080.0 | 24 | 110 | 57305.00 | 2 |

| | | | | |
|---------|----|-----|-----------|---|
| 13009.0 | 24 | 110 | 31000.00 | 2 |
| 13408.0 | 23 | 81 | 27487.41 | 2 |
| 13694.0 | 25 | 60 | 62653.10 | 2 |
| 13798.0 | 23 | 63 | 36351.42 | 2 |
| 14096.0 | 26 | 34 | 57120.91 | 2 |
| 14156.0 | 31 | 66 | 113384.14 | 2 |
| 14298.0 | 25 | 45 | 50862.44 | 2 |
| 14606.0 | 23 | 128 | 11713.85 | 2 |
| 14646.0 | 23 | 77 | 279489.02 | 2 |
| 14911.0 | 23 | 248 | 132572.62 | 2 |
| 15061.0 | 25 | 55 | 54228.74 | 2 |
| 15311.0 | 22 | 118 | 59419.34 | 2 |
| 16029.0 | 60 | 76 | 50992.61 | 2 |
| 16422.0 | 39 | 75 | 33805.69 | 2 |
| 16684.0 | 26 | 31 | 65892.08 | 2 |
| 17450.0 | 30 | 55 | 187482.17 | 2 |
| 17511.0 | 24 | 46 | 88125.38 | 2 |
| 17841.0 | 23 | 169 | 40340.78 | 2 |
| 17949.0 | 23 | 52 | 52750.84 | 2 |
| 18102.0 | 22 | 62 | 256438.49 | 2 |

```
[37]: RFM[RFM.Clusters==3]
```

| CustomerID | Recency | Frequency | Monetary | Clusters |
|------------|---------|-----------|----------|----------|
| 12346.0 | 347 | 2 | 0.00 | 3 |
| 12350.0 | 332 | 1 | 334.40 | 3 |
| 12353.0 | 226 | 1 | 89.00 | 3 |
| 12354.0 | 254 | 1 | 1079.40 | 3 |
| 12355.0 | 236 | 1 | 459.40 | 3 |
| ... | ... | ... | ... | ... |
| 18260.0 | 194 | 8 | 2595.00 | 3 |
| 18262.0 | 162 | 1 | 149.48 | 3 |
| 18269.0 | 380 | 2 | 138.90 | 3 |
| 18280.0 | 299 | 1 | 180.60 | 3 |
| 18281.0 | 202 | 1 | 80.82 | 3 |

1107 rows × 4 columns

```
[38]: final=RFM.groupby("Clusters").mean()[["Recency", "Frequency", "Monetary"]]
final
```

| Clusters | Recency | Frequency | Monetary |
|----------|------------|-----------|--------------|
| 1 | 61.560148 | 5.595620 | 1822.303551 |
| 2 | 27.521739 | 86.869565 | 81835.857391 |
| 3 | 267.719964 | 1.852755 | 460.644066 |

```
[39]: def func(row):
    if row["Clusters"]==2:
        return 'Diamond'
    elif row["Clusters"]==1:
        return 'Gold'
    else:
        return 'Silver'
```

```
[40]: RFM['group']=RFM.apply(func, axis=1)
```

```
[41]: RFM
```

| CustomerID | Recency | Frequency | Monetary | Clusters | group |
|------------|---------|-----------|----------|----------|--------|
| 12346.0 | 347 | 2 | 0.00 | 3 | Silver |
| 12347.0 | 24 | 7 | 4310.00 | 1 | Gold |
| 12348.0 | 87 | 1 | 1707.04 | 1 | Gold |

| | | | | | |
|---------|-----|-----|---------|-----|--------|
| 12348.0 | 97 | 4 | 1797.24 | 1 | Gold |
| 12349.0 | 40 | 1 | 1757.55 | 1 | Gold |
| 12350.0 | 332 | 1 | 334.40 | 3 | Silver |
| ... | ... | ... | ... | ... | ... |
| 18280.0 | 299 | 1 | 180.60 | 3 | Silver |
| 18281.0 | 202 | 1 | 80.82 | 3 | Silver |
| 18282.0 | 29 | 3 | 176.60 | 1 | Gold |
| 18283.0 | 25 | 16 | 2094.88 | 1 | Gold |
| 18287.0 | 64 | 3 | 1837.28 | 1 | Gold |

4372 rows × 5 columns

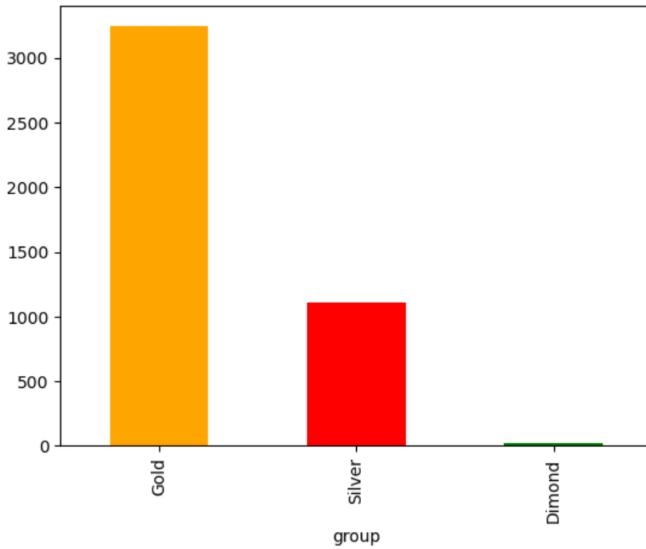
```
[42]: result=RFM.groupby('group').value_counts()
```

```
[43]: result
```

```
[43]: group
Gold      3242
Silver    1107
Dimond    23
Name: count, dtype: int64
```

```
[44]: result.plot(kind="bar", color=["Orange","Red","Green"])
```

```
[44]: <Axes: xlabel='group'>
```



```
[ ]: ABC = RFM[RFM.Clusters==3]
```

```
[ ]: ABC[ABC['Monetary']==ABC['Monetary'].min()]
```

```
[ ]:
```