RFM

May 7, 2024

[36]:
```
import numpy as np
import pandas as pd
```

[37]:
```
df= pd.read_excel("C:/Users/Abdo/Desktop/MyFiles/online_retail_II.xlsx")
```

[38]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype         
---  ------       --------------   -----         
 0   Invoice      525461 non-null  object        
 1   StockCode    525461 non-null  object        
 2   Description  522533 non-null  object        
 3   Quantity     525461 non-null  int64         
 4   InvoiceDate  525461 non-null  datetime64[ns]
 5   Price        525461 non-null  float64       
 6   Customer ID  417534 non-null  float64       
 7   Country      525461 non-null  object        
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 32.1+ MB
```

[39]:
```
df.head()
```

[39]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-12-01 07:45:00	6.95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-12-01 07:45:00	2.10	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-12-01 07:45:00	1.25	13085.0	United Kingdom

[40]:
```
df.isna().sum()
```

[40]:
```
Invoice             0
StockCode           0
Description      2928
Quantity            0
InvoiceDate         0
Price               0
Customer ID    107927
Country             0
dtype: int64
```

[41]:
```
df.dropna(inplace=True)
```

[42]:
```
df.isna().sum()
```

[42]:
```
Invoice        0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
Price          0
Customer ID    0
Country        0
dtype: int64
```

[43]:
```
df.shape
```

[43]:
```
(417534, 8)
```

[45]:
```
df['totalprice'] = df['Quantity'] * df['Price']
df
```

[45]:

	Invoice	StockCode	Description	Quantity	\
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	
1	489434	79323P	PINK CHERRY LIGHTS	12	
2	489434	79323W	WHITE CHERRY LIGHTS	12	
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	
…	…	…	…	…	
525456	538171	22271	FELTCRAFT DOLL ROSIE	2	
525457	538171	22750	FELTCRAFT PRINCESS LOLA DOLL	1	
525458	538171	22751	FELTCRAFT PRINCESS OLIVIA DOLL	1	
525459	538171	20970	PINK FLORAL FELTCRAFT SHOULDER BAG	2	
525460	538171	21931	JUMBO STORAGE BAG SUKI	2	

	InvoiceDate	Price	Customer ID	Country	totalprice
0	2009-12-01 07:45:00	6.95	13085.0	United Kingdom	83.40

```
1       2009-12-01 07:45:00   6.75      13085.0  United Kingdom       81.00
2       2009-12-01 07:45:00   6.75      13085.0  United Kingdom       81.00
3       2009-12-01 07:45:00   2.10      13085.0  United Kingdom      100.80
4       2009-12-01 07:45:00   1.25      13085.0  United Kingdom       30.00
…                       …    …           …              …          …
525456 2010-12-09 20:01:00   2.95      17530.0  United Kingdom        5.90
525457 2010-12-09 20:01:00   3.75      17530.0  United Kingdom        3.75
525458 2010-12-09 20:01:00   3.75      17530.0  United Kingdom        3.75
525459 2010-12-09 20:01:00   3.75      17530.0  United Kingdom        7.50
525460 2010-12-09 20:01:00   1.95      17530.0  United Kingdom        3.90

[417534 rows x 9 columns]
```

[46]:
```
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
```

[47]:
```
df['InvoiceDate'].max()
```

[47]:
```
Timestamp('2010-12-09 20:01:00')
```

[54]:
```
from datetime import datetime

# Define today's date
today_date = datetime.today()

# Then proceed with your aggregation
rfm = df.groupby('Customer ID').agg({
    'InvoiceDate': lambda date: (today_date - date.max()).days,
    'Invoice': lambda num: num.nunique(),
    'totalprice': lambda totalprice: totalprice.sum()
})
```

[55]:
```
rfm
```

[55]:
```
             InvoiceDate  Invoice  totalprice
Customer ID
12346.0             4909       15      -64.68
12347.0             4845        2     1323.32
12348.0             4916        1      222.16
12349.0             4885        4     2646.99
12351.0             4853        1      300.93
…                    …       …          …
18283.0             4860        6      641.77
18284.0             4907        2      436.68
18285.0             5138        1      427.00
18286.0             4954        3     1188.43
18287.0             4860        5     2340.61
```

```
[4383 rows x 3 columns]
```

[56]:
```
rfm.columns=['Recency','Frequency','Monetary']
rfm
```

[56]:
```
             Recency  Frequency  Monetary
Customer ID
12346.0         4909         15    -64.68
12347.0         4845          2   1323.32
12348.0         4916          1    222.16
12349.0         4885          4   2646.99
12351.0         4853          1    300.93
…               …          …        …
18283.0         4860          6    641.77
18284.0         4907          2    436.68
18285.0         5138          1    427.00
18286.0         4954          3   1188.43
18287.0         4860          5   2340.61

[4383 rows x 3 columns]
```

[57]:
```
rfm = rfm[rfm['Monetary']>0]
rfm
```

[57]:
```
             Recency  Frequency  Monetary
Customer ID
12347.0         4845          2   1323.32
12348.0         4916          1    222.16
12349.0         4885          4   2646.99
12351.0         4853          1    300.93
12352.0         4853          2    343.80
…               …          …        …
18283.0         4860          6    641.77
18284.0         4907          2    436.68
18285.0         5138          1    427.00
18286.0         4954          3   1188.43
18287.0         4860          5   2340.61

[4282 rows x 3 columns]
```

[58]:
```
rfm.describe().T
```

[58]:
```
             count         mean          std           min        25%      50%  \
Recency     4282.0  4930.560953    95.734145  4.843000e+03  4859.0000  4892.00
Frequency   4282.0     5.455395    10.148765  1.000000e+00     1.0000     3.00
Monetary    4282.0  1971.709950  8596.562848  1.776357e-15   303.6925   686.65
```

```
                  75%        max
Recency    4972.0000    5216.00
Frequency     6.0000     270.00
Monetary   1683.9625  341776.73
```

[ ]: