

## Custom Profitability Using Recency Frequency Monetary Analysis

```
import pandas as pd
import datetime as dt
import matplotlib.pyplot as plt # for plotting graphs
import seaborn as sns # for plotting graphs
```

```
print("Start")
```

```
# Load in data
```

```
df = pd.read_excel('online_retail_II.xlsx')
```

Invoice	StockCode	Description	Quantity	InvoiceDate	Price	CustomerID	Country
489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12.00000	2009-12-01 07:45:00	6.95000	13085.00000	United Kingdom
489434	79323P	PINK CHERRY LIGHTS	12.00000	2009-12-01 07:45:00	6.75000	13085.00000	United Kingdom
489434	79323W	WHITE CHERRY LIGHTS	12.00000	2009-12-01 07:45:00	6.75000	13085.00000	United Kingdom
489434	22041	RECORD FRAME 7" SINGLE SIZE	48.00000	2009-12-01 07:45:00	2.10000	13085.00000	United Kingdom
489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24.00000	2009-12-01 07:45:00	1.25000	13085.00000	United Kingdom
489434	22064	PINK DOUGHNUT TRINKET POT	24.00000	2009-12-01 07:45:00	1.65000	13085.00000	United Kingdom
489434	21871	SAVE THE PLANET MUG	24.00000	2009-12-01 07:45:00	1.25000	13085.00000	United Kingdom
489434	21523	FANCY FONT HOME SWEET HOME DOORMAT	10.00000	2009-12-01 07:45:00	5.95000	13085.00000	United Kingdom
489435	22350	CAT BOWL	12.00000	2009-12-01 07:46:00	2.55000	13085.00000	United Kingdom
489435	22349	DOG BOWL, CHASING BALL DESIGN	12.00000	2009-12-01 07:46:00	3.75000	13085.00000	United Kingdom
489435	22195	HEART MEASURING SPOONS LARGE	24.00000	2009-12-01 07:46:00	1.65000	13085.00000	United Kingdom
489435	22353	LUNCHBOX WITH CUTLERY FAIRY CAKES	12.00000	2009-12-01 07:46:00	2.55000	13085.00000	United Kingdom
489436	48173C	DOOR MAT BLACK FLOCK	10.00000	2009-12-01 09:06:00	5.95000	13078.00000	United Kingdom
489436	21755	LOVE BUILDING BLOCK WORD	18.00000	2009-12-01 09:06:00	5.45000	13078.00000	United Kingdom
489436	21754	HOME BUILDING BLOCK WORD	3.00000	2009-12-01 09:06:00	5.95000	13078.00000	United Kingdom
489436	84879	ASSORTED COLOUR BIRD ORNAMENT	16.00000	2009-12-01 09:06:00	1.69000	13078.00000	United Kingdom
489436	22119	PEACE WOODEN BLOCK LETTERS	3.00000	2009-12-01 09:06:00	6.95000	13078.00000	United Kingdom
489436	22142	CHRISTMAS CRAFT WHITE FAIRY	12.00000	2009-12-01 09:06:00	1.45000	13078.00000	United Kingdom

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	CustomerID	Country
525443	538171	21329	DINOSAURS WRITING SET	2.00000	2010-12-09 20:01:00	1.65000	17530.00000	United Kingdom
525444	538171	21327	SKULLS WRITING SET	2.00000	2010-12-09 20:01:00	1.65000	17530.00000	United Kingdom
525445	538171	22666	RECIPE BOX PANTRY YELLOW DESIGN	1.00000	2010-12-09 20:01:00	2.95000	17530.00000	United Kingdom
525446	538171	22154	ANGEL DECORATION 3 BUTTONS	48.00000	2010-12-09 20:01:00	0.42000	17530.00000	United Kingdom
525447	538171	21875	KINGS CHOICE MUG	2.00000	2010-12-09 20:01:00	1.25000	17530.00000	United Kingdom
525448	538171	21871	SAVE THE PLANET MUG	2.00000	2010-12-09 20:01:00	1.25000	17530.00000	United Kingdom
525449	538171	21877	HOME SWEET HOME MUG	2.00000	2010-12-09 20:01:00	1.25000	17530.00000	United Kingdom
525450	538171	22747	POPPY'S PLAYHOUSE BATHROOM	2.00000	2010-12-09 20:01:00	2.10000	17530.00000	United Kingdom
525451	538171	22748	POPPY'S PLAYHOUSE KITCHEN	2.00000	2010-12-09 20:01:00	2.10000	17530.00000	United Kingdom
525452	538171	22745	POPPY'S PLAYHOUSE BEDROOM	2.00000	2010-12-09 20:01:00	2.10000	17530.00000	United Kingdom
525453	538171	22558	CLOTHES PEGS RETROSPOT PACK 24	4.00000	2010-12-09 20:01:00	1.49000	17530.00000	United Kingdom
525454	538171	21671	RED SPOT CERAMIC DRAWER KNOB	6.00000	2010-12-09 20:01:00	1.25000	17530.00000	United Kingdom
525455	538171	20971	PINK BLUE FELT CRAFT TRINKET BOX	2.00000	2010-12-09 20:01:00	1.25000	17530.00000	United Kingdom
525456	538171	22271	FELTCRAFT DOLL ROSIE	2.00000	2010-12-09 20:01:00	2.95000	17530.00000	United Kingdom
525457	538171	22750	FELTCRAFT PRINCESS LOLA DOLL	1.00000	2010-12-09 20:01:00	3.75000	17530.00000	United Kingdom
525458	538171	22751	FELTCRAFT PRINCESS OLIVIA DOLL	1.00000	2010-12-09 20:01:00	3.75000	17530.00000	United Kingdom
525459	538171	20970	PINK FLORAL FELTCRAFT SHOULDER BAG	2.00000	2010-12-09 20:01:00	3.75000	17530.00000	United Kingdom
525460	538171	21931	JUMBO STORAGE BAG SUKI	2.00000	2010-12-09 20:01:00	1.95000	17530.00000	United Kingdom

```
print(df.head())
```

	Invoice	StockCode	...	Customer ID	Country
0	489434	85048	...	13085.0	United Kingdom
1	489434	79323P	...	13085.0	United Kingdom
2	489434	79323W	...	13085.0	United Kingdom
3	489434	22041	...	13085.0	United Kingdom
4	489434	21232	...	13085.0	United Kingdom

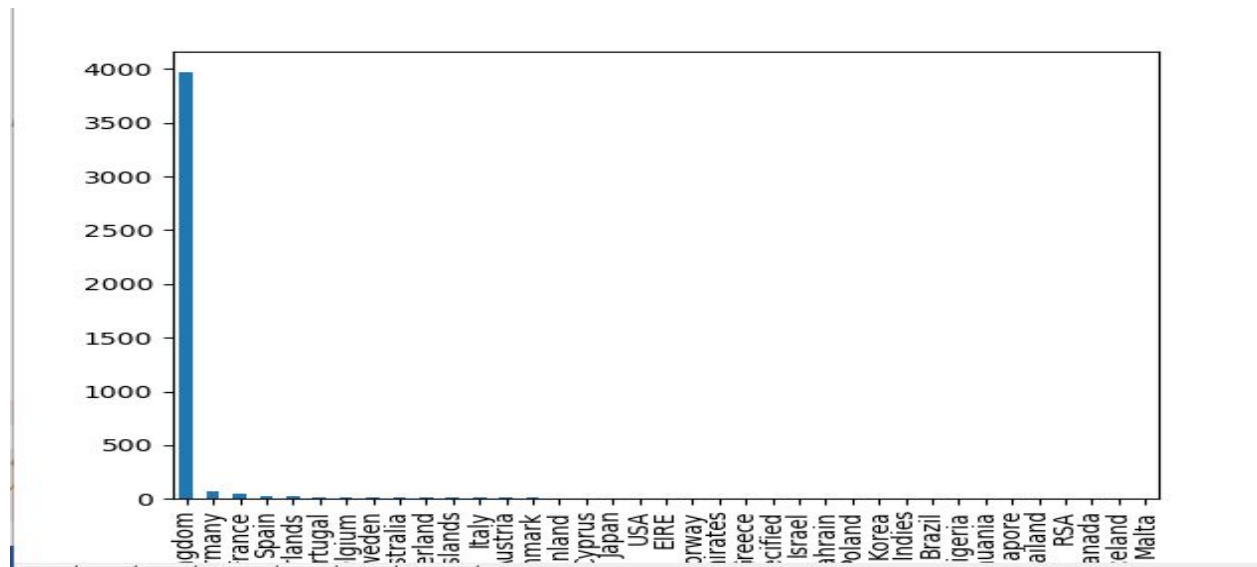
```
filtered=df[['Country','CustomerID']].drop_duplicates()
```

	Country	CustomerID
0	United Kingdom	13085.00000
12	United Kingdom	13078.00000
31	United Kingdom	15362.00000
54	United Kingdom	18102.00000
71	France	12682.00000
90	United Kingdom	18087.00000
96	United Kingdom	13635.00000
119	United Kingdom	14110.00000
126	USA	12636.00000
127	United Kingdom	17519.00000
146	United Kingdom	13758.00000
173	Belgium	12362.00000
174	United Kingdom	15413.00000
187	Australia	16321.00000
209	United Kingdom	16167.00000
224	United Kingdom	17865.00000
244	United Kingdom	17592.00000
264	United Kingdom	13767.00000

```
unique_country = df['Country'].unique()
print(unique_country)
```

```
['United Kingdom' 'France' 'USA' 'Belgium' 'Australia' 'EIRE' 'Germany'
 'Portugal' 'Denmark' 'Netherlands' 'Poland' 'Channel Islands' 'Spain'
 'Cyprus' 'Greece' 'Norway' 'Austria' 'Sweden' 'United Arab Emirates'
 'Finland' 'Italy' 'Switzerland' 'Japan' 'Unspecified' 'Nigeria' 'Malta'
 'RSA' 'Singapore' 'Bahrain' 'Thailand' 'Israel' 'Lithuania' 'West Indies'
 'Korea' 'Brazil' 'Canada' 'Iceland']
```

```
filtered.Country.value_counts().plot(kind='bar')
```



Get United Kingdom Data

```
data = df[df.Country == 'United Kingdom']
```



	CustomerID	InvoiceDate	Invoice	Quantity	Price
0	13085.00000	2009-12-01 07:45:00	489434	12.00000	6.95000
1	13085.00000	2009-12-01 07:45:00	489434	12.00000	6.75000
2	13085.00000	2009-12-01 07:45:00	489434	12.00000	6.75000
3	13085.00000	2009-12-01 07:45:00	489434	48.00000	2.10000
4	13085.00000	2009-12-01 07:45:00	489434	24.00000	1.25000
5	13085.00000	2009-12-01 07:45:00	489434	24.00000	1.65000
6	13085.00000	2009-12-01 07:45:00	489434	24.00000	1.25000
7	13085.00000	2009-12-01 07:45:00	489434	10.00000	5.95000
8	13085.00000	2009-12-01 07:46:00	489435	12.00000	2.55000
9	13085.00000	2009-12-01 07:46:00	489435	12.00000	3.75000
10	13085.00000	2009-12-01 07:46:00	489435	24.00000	1.65000
11	13085.00000	2009-12-01 07:46:00	489435	12.00000	2.55000
12	13078.00000	2009-12-01 09:06:00	489436	10.00000	5.95000
13	13078.00000	2009-12-01 09:06:00	489436	18.00000	5.45000
14	13078.00000	2009-12-01 09:06:00	489436	3.00000	5.95000
15	13078.00000	2009-12-01 09:06:00	489436	16.00000	1.69000
16	13078.00000	2009-12-01 09:06:00	489436	3.00000	6.95000
17	13078.00000	2009-12-01 09:06:00	489436	12.00000	1.45000

525443	17530.00000	2010-12-09 20:01:00	538171	2.00000	1.65000
525444	17530.00000	2010-12-09 20:01:00	538171	2.00000	1.65000
525445	17530.00000	2010-12-09 20:01:00	538171	1.00000	2.95000
525446	17530.00000	2010-12-09 20:01:00	538171	48.00000	0.42000
525447	17530.00000	2010-12-09 20:01:00	538171	2.00000	1.25000
525448	17530.00000	2010-12-09 20:01:00	538171	2.00000	1.25000
525449	17530.00000	2010-12-09 20:01:00	538171	2.00000	1.25000
525450	17530.00000	2010-12-09 20:01:00	538171	2.00000	2.10000
525451	17530.00000	2010-12-09 20:01:00	538171	2.00000	2.10000
525452	17530.00000	2010-12-09 20:01:00	538171	2.00000	2.10000
525453	17530.00000	2010-12-09 20:01:00	538171	4.00000	1.49000
525454	17530.00000	2010-12-09 20:01:00	538171	6.00000	1.25000
525455	17530.00000	2010-12-09 20:01:00	538171	2.00000	1.25000
525456	17530.00000	2010-12-09 20:01:00	538171	2.00000	2.95000
525457	17530.00000	2010-12-09 20:01:00	538171	1.00000	3.75000
525458	17530.00000	2010-12-09 20:01:00	538171	1.00000	3.75000
525459	17530.00000	2010-12-09 20:01:00	538171	2.00000	3.75000
525460	17530.00000	2010-12-09 20:01:00	538171	2.00000	1.95000

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
```

describe = df.describe()

	Quantity	Price	Customer ID
count	525461.000000	525461.000000	417534.000000
mean	10.337667	4.688834	15360.645478
std	107.424110	146.126914	1680.811316
min	-9600.000000	-53594.360000	12346.000000
25%	1.000000	1.250000	13983.000000
50%	3.000000	2.100000	15311.000000
75%	10.000000	4.210000	16799.000000
max	19152.000000	25111.090000	18287.000000

Removing Quantity below zero, Price below zero

```
df = df[(df["Quantity"]>0) ]
df = df[(df["Price x"]>0) ]
print(df.describe())
```

	Quantity	Price	Customer ID
count	511566.000000	511566.000000	407664.000000
mean	11.400150	4.252563	15368.592598
std	86.761177	63.664629	1679.762138
min	1.000000	0.001000	12346.000000
25%	1.000000	1.250000	13997.000000
50%	3.000000	2.100000	15321.000000
75%	10.000000	4.210000	16812.000000
max	19152.000000	25111.090000	18287.000000

**Customer Profitability =  
Highest Monetary + Highest Frequency + Lowest Recency**

```
data=df[['Customer ID','InvoiceDate','Invoice','Quantity','Price']]
data['TotalProfit'] = data['Quantity'] * data['Price']
```

	CustomerID	InvoiceDate	Invoice	Quantity	Price	TotalProfit
0	13085.00000	2009-12-01 07:45:00	489434	12.00000	6.95000	83.40000
1	13085.00000	2009-12-01 07:45:00	489434	12.00000	6.75000	81.00000
2	13085.00000	2009-12-01 07:45:00	489434	12.00000	6.75000	81.00000
3	13085.00000	2009-12-01 07:45:00	489434	48.00000	2.10000	100.80000
4	13085.00000	2009-12-01 07:45:00	489434	24.00000	1.25000	30.00000
5	13085.00000	2009-12-01 07:45:00	489434	24.00000	1.65000	39.60000
6	13085.00000	2009-12-01 07:45:00	489434	24.00000	1.25000	30.00000
7	13085.00000	2009-12-01 07:45:00	489434	10.00000	5.95000	59.50000
8	13085.00000	2009-12-01 07:46:00	489435	12.00000	2.55000	30.60000
9	13085.00000	2009-12-01 07:46:00	489435	12.00000	3.75000	45.00000
10	13085.00000	2009-12-01 07:46:00	489435	24.00000	1.65000	39.60000
11	13085.00000	2009-12-01 07:46:00	489435	12.00000	2.55000	30.60000
12	13078.00000	2009-12-01 09:06:00	489436	10.00000	5.95000	59.50000
13	13078.00000	2009-12-01 09:06:00	489436	18.00000	5.45000	98.10000
14	13078.00000	2009-12-01 09:06:00	489436	3.00000	5.95000	17.85000
15	13078.00000	2009-12-01 09:06:00	489436	16.00000	1.69000	27.04000
16	13078.00000	2009-12-01 09:06:00	489436	3.00000	6.95000	20.85000
17	13078.00000	2009-12-01 09:06:00	489436	12.00000	1.45000	17.40000

```
current_time = dt.datetime(2020,7,1)
```

```
> current_time = {datetime} 2020-07-01 00:00:00
```



```
rfm= data.groupby('Customer ID').agg({'InvoiceDate': lambda date: (current_time -
date.max()).days,'Invoice': lambda num: len(num),'Price': lambda price: price.sum()})
```

```
rfm_columns = rfm.columns;
rfm.columns=['CustomerID','monetary','frequency','recency']
rfm_columns = rfm.columns;
rfm['recency'] = rfm['recency'].astype(int)
```

	monetary	frequency	recency
12346.0	3655.00000	33.00000	206.00000
12347.0	3493.00000	71.00000	162.00000
12348.0	3564.00000	20.00000	14.00000
12349.0	3533.00000	102.00000	875.00000
12351.0	3501.00000	21.00000	49.00000
12352.0	3501.00000	18.00000	54.00000
12353.0	3534.00000	20.00000	38.00000
12355.0	3693.00000	22.00000	52.00000
12356.0	3506.00000	84.00000	261.00000
12357.0	3514.00000	165.00000	1395.00000
12358.0	3501.00000	58.00000	256.00000
12359.0	3551.00000	117.00000	553.00000
12360.0	3505.00000	88.00000	238.00000
12361.0	3588.00000	19.00000	78.00000
12362.0	3864.00000	1.00000	130.00000
12366.0	3759.00000	3.00000	5.00000
12368.0	3754.00000	18.00000	79.00000
12369.0	3539.00000	93.00000	292.00000

	monetary	frequency	recency
CustomerID			
12346.0	3655	33	206
12347.0	3493	71	162
12348.0	3564	20	14
12349.0	3533	102	875
12351.0	3501	21	49

```
rfm['recency_q'] = pd.qcut(rfm['recency'], 4, ['1','2','3','4'])
rfm['frequency_q'] = pd.qcut(rfm['frequency'], 4, ['4','3','2','1'])
rfm['monetary_q'] = pd.qcut(rfm['monetary'], 4, ['4','3','2','1'])
print(rfm.head())
```



```
rfm['RFM_Score'] = rfm.recency_q.astype(str)+ rfm.frequency_q.astype(str) + rfm.monetary_q.astype(str)
```

	monetary	frequency	recency	recency_q	frequency_q	monetary_q	RFM_Score
12346.0	3655.00000	33.00000	206.00000	3	3	1	331
12347.0	3493.00000	71.00000	162.00000	3	2	4	324
12348.0	3564.00000	20.00000	14.00000	1	3	2	132
12349.0	3533.00000	102.00000	875.00000	4	2	3	423
12351.0	3501.00000	21.00000	49.00000	1	3	4	134
12352.0	3501.00000	18.00000	54.00000	1	4	4	144
12353.0	3534.00000	20.00000	38.00000	1	3	3	133
12355.0	3693.00000	22.00000	52.00000	1	3	1	131
12356.0	3506.00000	84.00000	261.00000	3	2	4	324
12357.0	3514.00000	165.00000	1395.00000	4	1	3	413
12358.0	3501.00000	58.00000	256.00000	3	2	4	324
12359.0	3551.00000	117.00000	553.00000	4	1	2	412
12360.0	3505.00000	88.00000	238.00000	3	2	4	324
12361.0	3588.00000	19.00000	78.00000	2	3	2	232
12362.0	3864.00000	1.00000	130.00000	2	4	1	241
12366.0	3759.00000	3.00000	5.00000	1	4	1	141
12368.0	3754.00000	18.00000	79.00000	2	4	1	241
12369.0	3539.00000	93.00000	292.00000	3	2	3	323

CustomerID	monetary	frequency	recency	recency_q	frequency_q	monetary_q
12346.0	3655	33	206	3	3	1
12347.0	3493	71	162	3	2	4
12348.0	3564	20	14	1	3	2
12349.0	3533	102	875	4	2	3
12351.0	3501	21	49	1	3	4

Customs profitability sorted by **RFM score = min for best values** and then by **max Monetary Values**

```
rfm['RFM_Score'] = rfm['RFM_Score'].astype(int)
customer_profitability
= rfm.sort_values(by=['RFM_Score','monetary'], ascending=[True,False])
```

	monetary	frequency	recency	recency_q	frequency_q	monetary_q	RFM_Score
13902.0	3758.00000	63.00000	52.00000	1	2	1	121.00000
14771.0	3747.00000	47.00000	53.00000	1	2	1	121.00000
13989.0	3590.00000	45.00000	50.00000	1	2	2	122.00000
13687.0	3564.00000	45.00000	6.00000	1	2	2	122.00000
13027.0	3511.00000	60.00000	24.00000	1	2	3	123.00000
13526.0	3864.00000	44.00000	45.00000	1	3	1	131.00000
14654.0	3864.00000	27.00000	35.00000	1	3	1	131.00000
17441.0	3858.00000	21.00000	43.00000	1	3	1	131.00000
12941.0	3850.00000	27.00000	44.00000	1	3	1	131.00000
17103.0	3848.00000	32.00000	53.00000	1	3	1	131.00000
15152.0	3827.00000	29.00000	56.00000	1	3	1	131.00000
16641.0	3827.00000	24.00000	47.00000	1	3	1	131.00000
13315.0	3815.00000	22.00000	43.00000	1	3	1	131.00000
12956.0	3810.00000	23.00000	42.00000	1	3	1	131.00000
16673.0	3805.00000	20.00000	56.00000	1	3	1	131.00000
15022.0	3803.00000	19.00000	52.00000	1	3	1	131.00000
15448.0	3800.00000	21.00000	50.00000	1	3	1	131.00000
16834.0	3800.00000	24.00000	34.00000	1	3	1	131.00000

	monetary	frequency	recency	recency_q	frequency_q	monetary_q	RFM_Score
14538.0	3656.00000	33.00000	336.00000	4	3	1	431.00000
14427.0	3620.00000	27.00000	386.00000	4	3	2	432.00000
15809.0	3585.00000	23.00000	360.00000	4	3	2	432.00000
14063.0	3564.00000	40.00000	13443.00000	4	3	2	432.00000
12980.0	3512.00000	32.00000	466.00000	4	3	3	433.00000
17050.0	3501.00000	41.00000	372.00000	4	3	4	434.00000
15760.0	3756.00000	2.00000	13916.00000	4	4	1	441.00000
12918.0	3752.00000	1.00000	10953.00000	4	4	1	441.00000
14308.0	3675.00000	2.00000	1147.00000	4	4	1	441.00000
14255.0	3672.00000	1.00000	1000.00000	4	4	1	441.00000
16943.0	3665.00000	2.00000	482.00000	4	4	1	441.00000
12737.0	3624.00000	2.00000	3710.00000	4	4	2	442.00000
14328.0	3564.00000	1.00000	445.00000	4	4	2	442.00000
14802.0	3564.00000	2.00000	1502.00000	4	4	2	442.00000
15734.0	3562.00000	17.00000	368.00000	4	4	2	442.00000
13776.0	3554.00000	2.00000	623.00000	4	4	2	442.00000
15202.0	3546.00000	3.00000	6001.00000	4	4	2	442.00000
16915.0	3512.00000	17.00000	512.00000	4	4	3	443.00000