CUSTOMER SEGMENTATION

- RFM ANALYSIS
- K-MEANS MODELLING
- COHORT ANALYSIS

GROUP – 2 CAPSTONE PROJECT - 1

CONTENTS

GROUPS MEMBERS

- Muhsin Ayaz
- İbrahim Şimşek

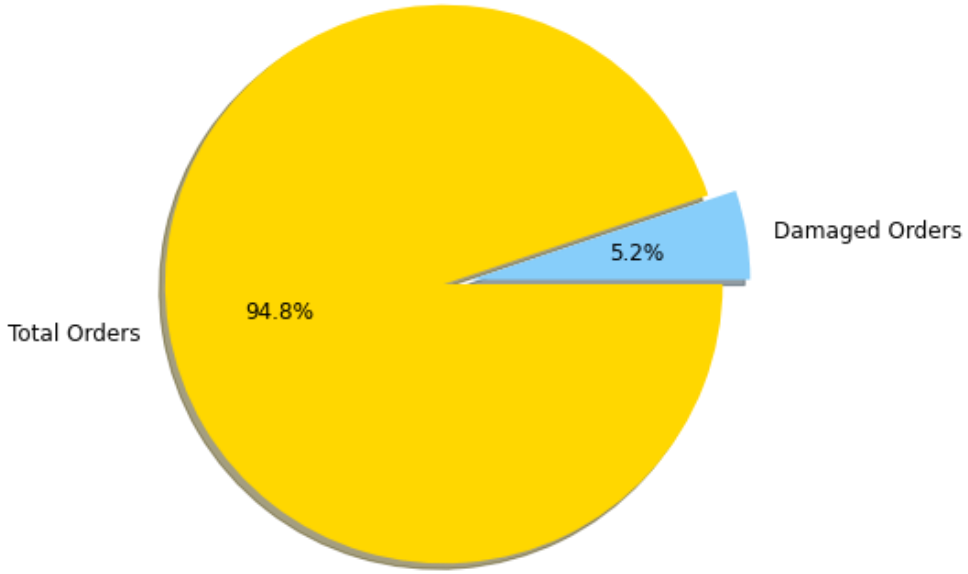
DATA ANALYSIS

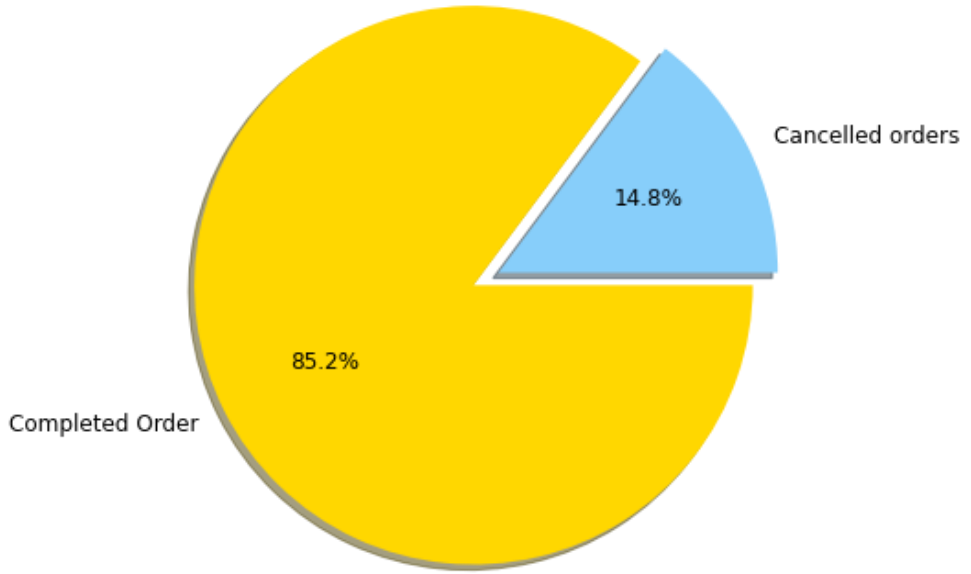
- Fields : 8
- Record : 541909

```
[ ]  df.describe()
```

	Quantity	UnitPrice	CustomerID
count	541909.000000	541909.000000	406829.000000
mean	9.552250	4.611114	15287.690570
std	218.081158	96.759853	1713.600303
min	-80995.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13953.000000
50%	3.000000	2.080000	15152.000000
75%	10.000000	4.130000	16791.000000
max	80995.000000	38970.000000	18287.000000

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

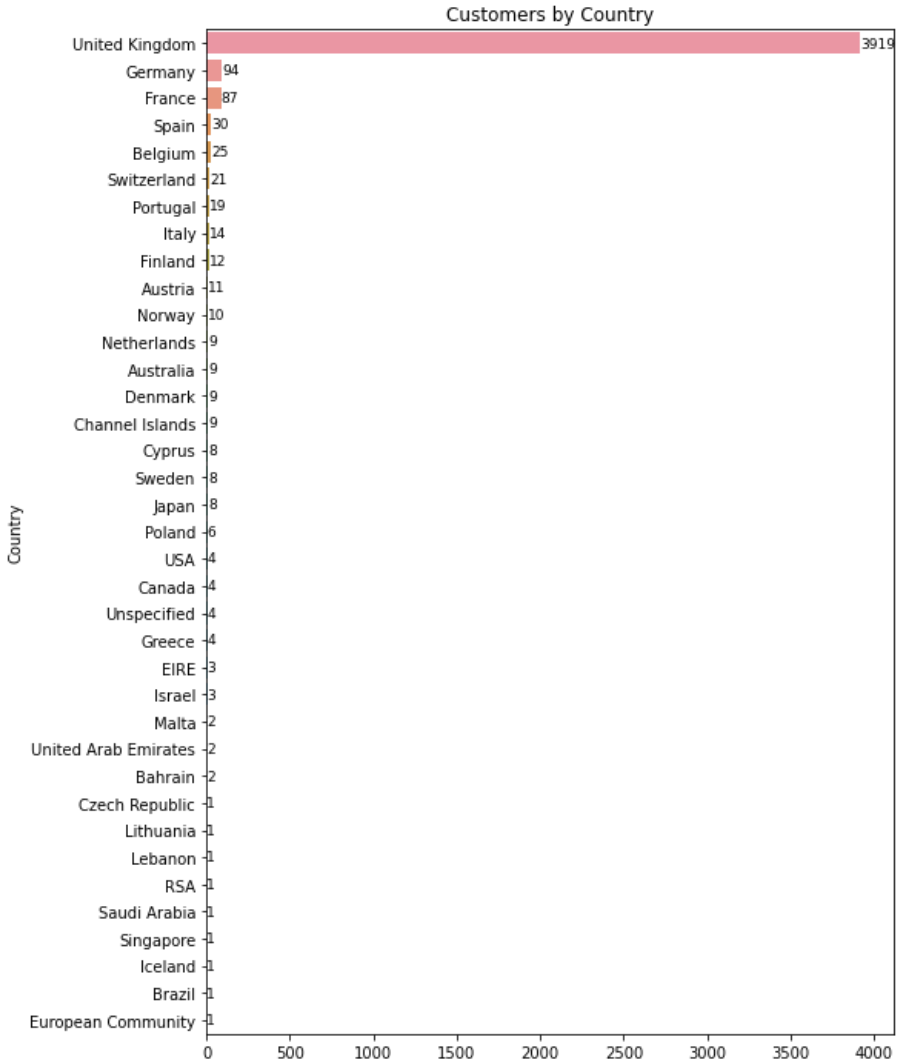




DATA ANALYSIS

- Total Orders : 25900
- Cancelled Orders : 3836
- Damaged Orders : 1336

DATA ANALYSIS



Country	
United Kingdom	6747156.154
Netherlands	284661.540
EIRE	250001.780
Germany	221509.470
France	196626.050
Australia	137009.770
Switzerland	55739.400
Spain	54756.030
Belgium	40910.960
Sweden	36585.410
Japan	35340.620
Norway	35163.460
Portugal	28995.760
Finland	22326.740
Channel Islands	20076.390
Denmark	18768.140
Italy	16890.510
Cyprus	12858.760
Austria	10154.320
Singapore	9120.390
Poland	7213.140
Israel	6988.400
Greece	4710.520
Iceland	4310.000
Canada	3666.380
Unspecified	2660.770
Malta	2505.470
United Arab Emirates	1902.280
USA	1730.920
Lebanon	1693.880
Lithuania	1661.060
European Community	1291.750
Brazil	1143.600
RSA	1002.310
Czech Republic	707.720
Bahrain	548.400
Saudi Arabia	131.170

EDA

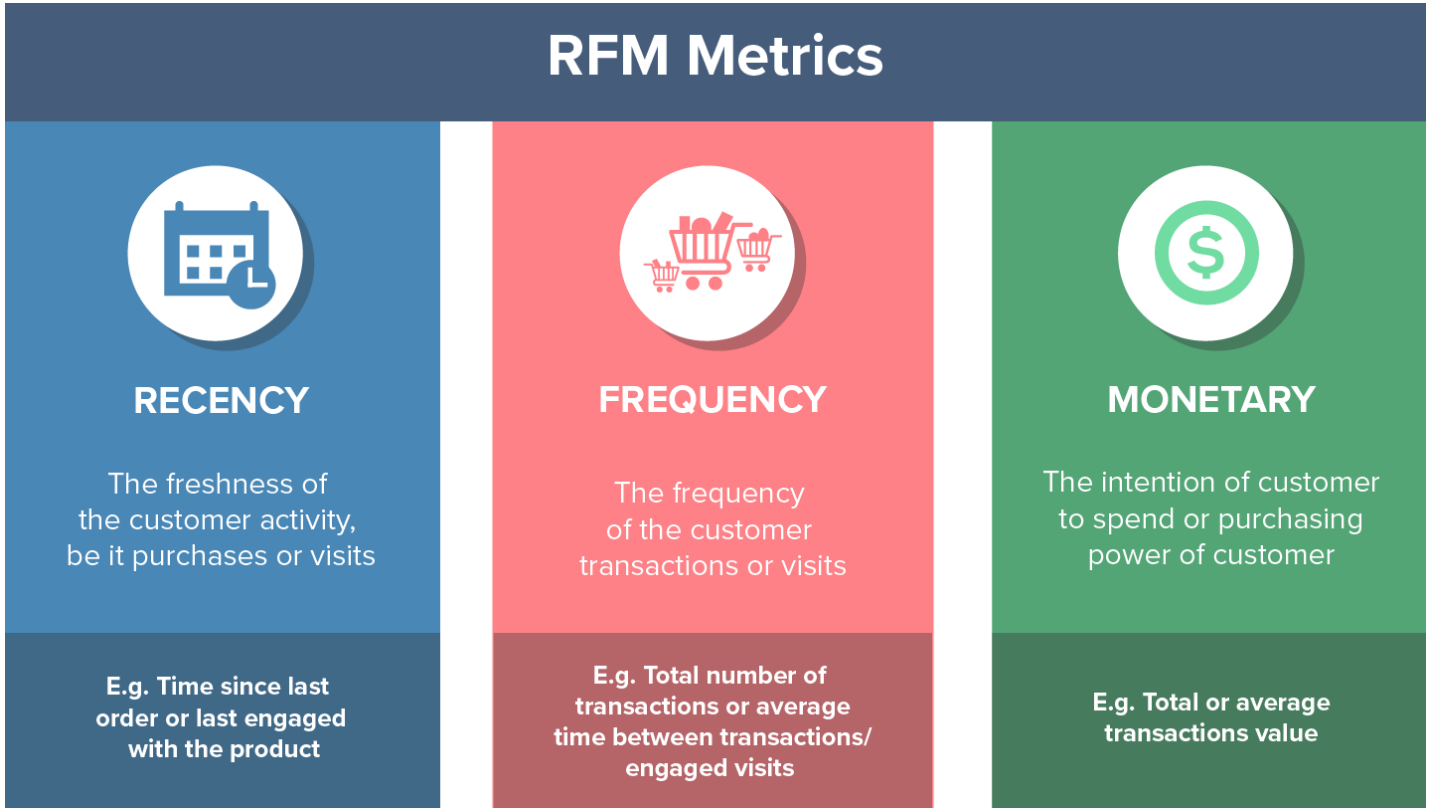
Dropped Records

- Dublicated Records
- Cancelled Invoice
- Negative Quantities
- Negative UnitPrice
- Damaged Records
- Non-CustomersID

```
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    349223 non-null  object
 1   StockCode    349223 non-null  object
 2   Description  349223 non-null  object
 3   Quantity     349223 non-null  int64
 4   InvoiceDate  349223 non-null  datetime64[ns]
 5   UnitPrice    349223 non-null  float64
 6   CustomerID   349223 non-null  float64
 7   Country      349223 non-null  object
 8   Reveneu      349223 non-null  float64
```

Total Records → 541909
After EDA → 349223

RFM ANALYSIS

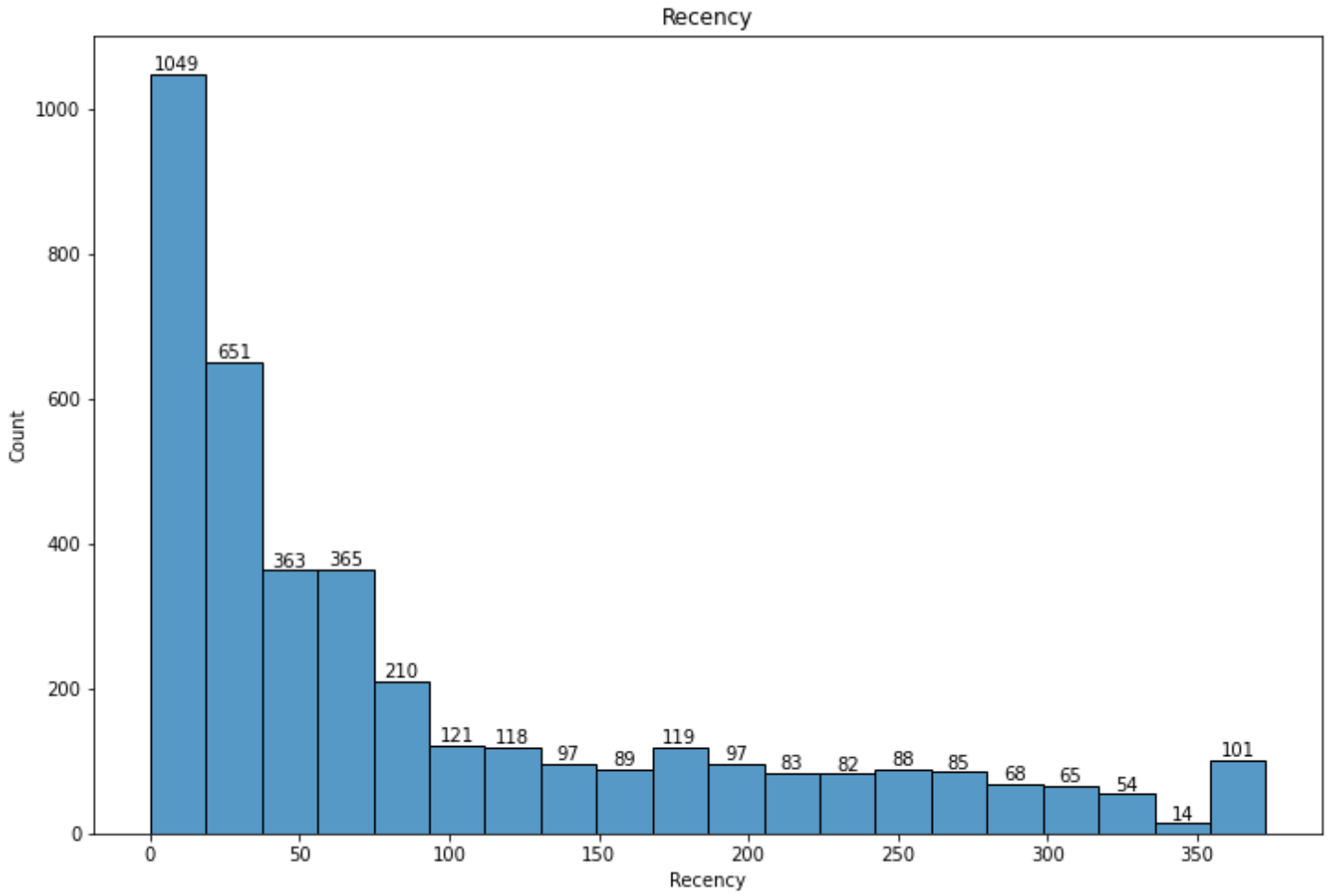


- https://clevertap.com/blog/rfm-analysis/

RFM ANALYSIS

RECENCY

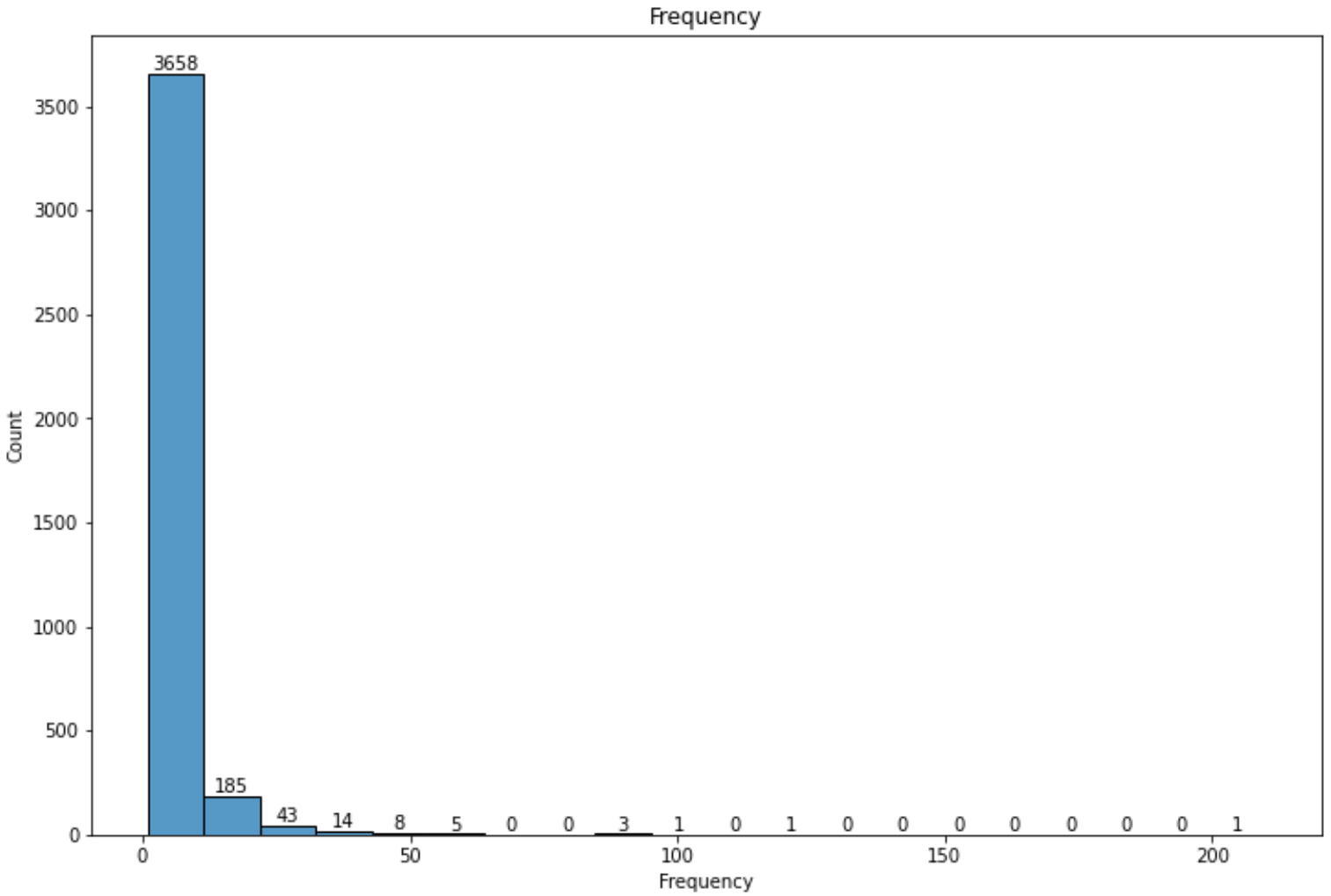
```
df["LastPurchaseDate"] = df.groupby("CustomerID")["Date"].transform(max)

df["LastOrderDays"] = current_date.date() - df["LastPurchaseDate"]

df_rfm["Recency"] = df.groupby("CustomerID")["LastOrderDays"].max().dt.days
```



RFM ANALYSIS

Frequency

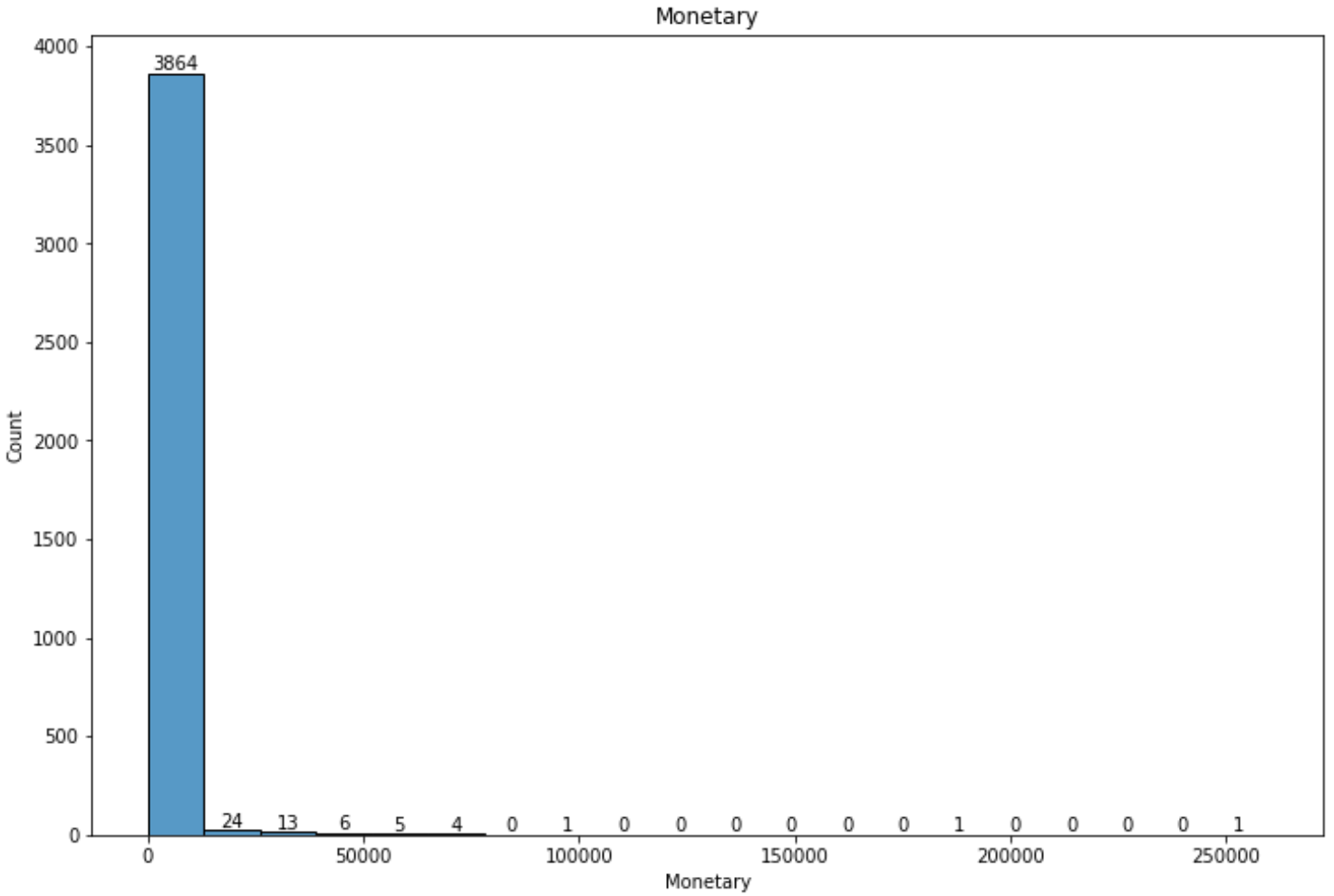
```
df_rfm["Frequency"] = df.groupby(["CustomerID"])["InvoiceNo"].nunique()
```



RFM ANALYSIS

MONETARY

```
[ ]  df_rfm["Monetary"] = df.groupby("CustomerID")["Reveneu"].sum()
```



RFM ANALYSIS

```
[ ]  #Rfm
     def RScore(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
     def RScoreRec(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
```

```
quantiles = df_rfm.quantile(q=[0.25,0.5,0.75])
quantiles = quantiles.to_dict()
# Recency
df_rfm['RecencyTile'] = df_rfm['Recency'].apply(RScoreRec, args=('Recency',quantiles,))
# Frequency
df_rfm['FrequencyTile'] = df_rfm['Frequency'].apply(RScore, args=('Frequency',quantiles,))
# Monetary
df_rfm['MonetaryTile'] = df_rfm['Monetary'].apply(RScore, args=('Monetary',quantiles,))
```

RFM ANALYSIS

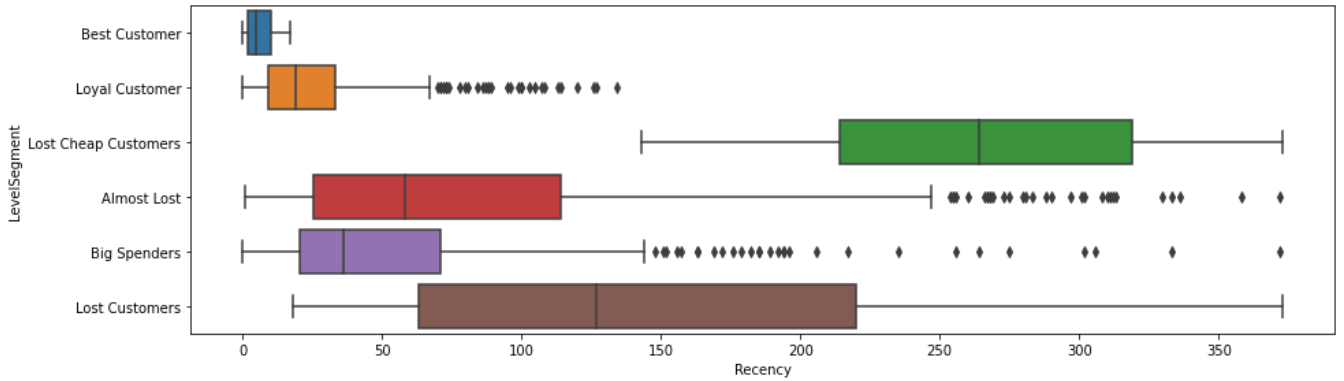
```
df_rfm["RFM_Level"] = df_rfm["FrequencyTile"]+df_rfm["RecencyTile"]+df_rfm["MonetaryTile"]
```

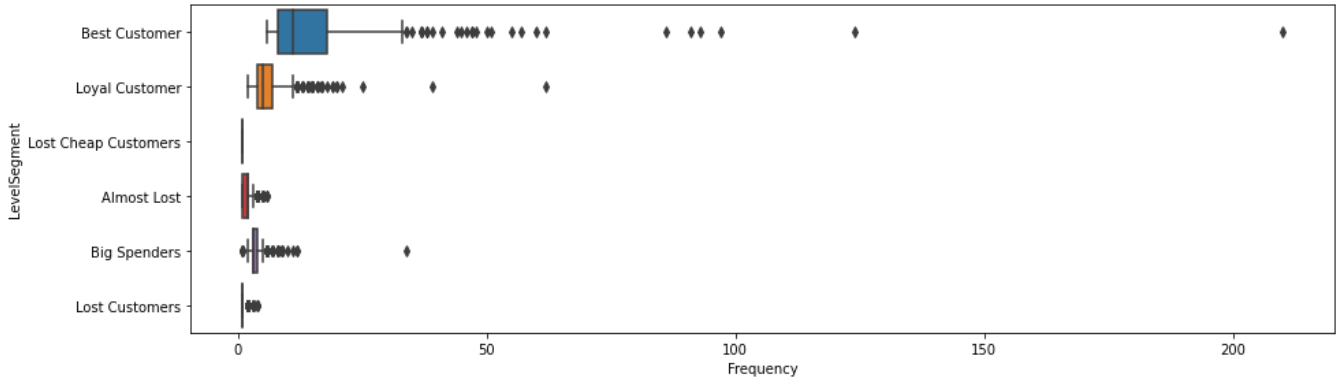
```
df_rfm.head()
```

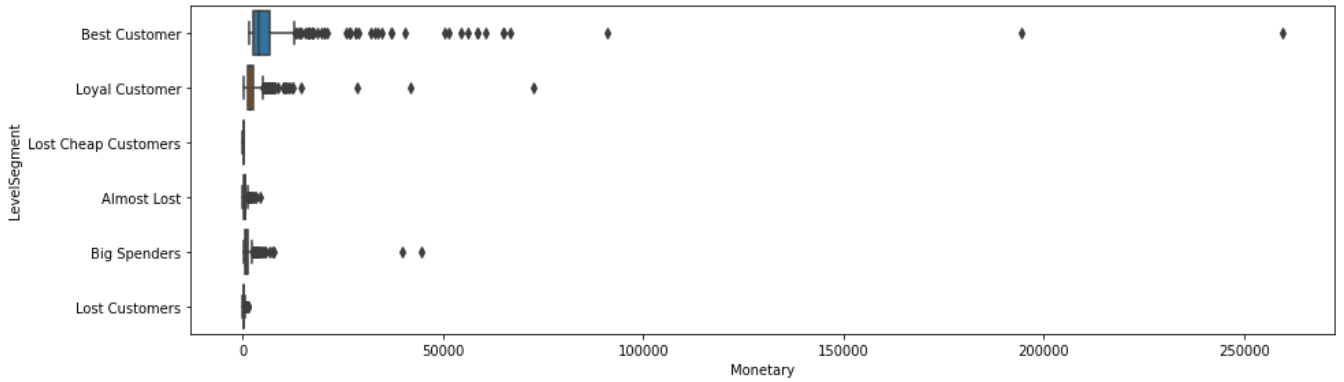
CustomerID	Recency	Frequency	Monetary	RecencyTile	FrequencyTile	MonetaryTile	RFM_Score	RFM_Level
12747.0	2	11	4196.01	4	4	4	444	12
12748.0	0	210	33053.19	4	4	4	444	12
12749.0	3	5	4090.88	4	3	4	434	11
12820.0	3	4	942.34	4	3	3	433	10
12821.0	214	1	92.72	1	1	1	111	3

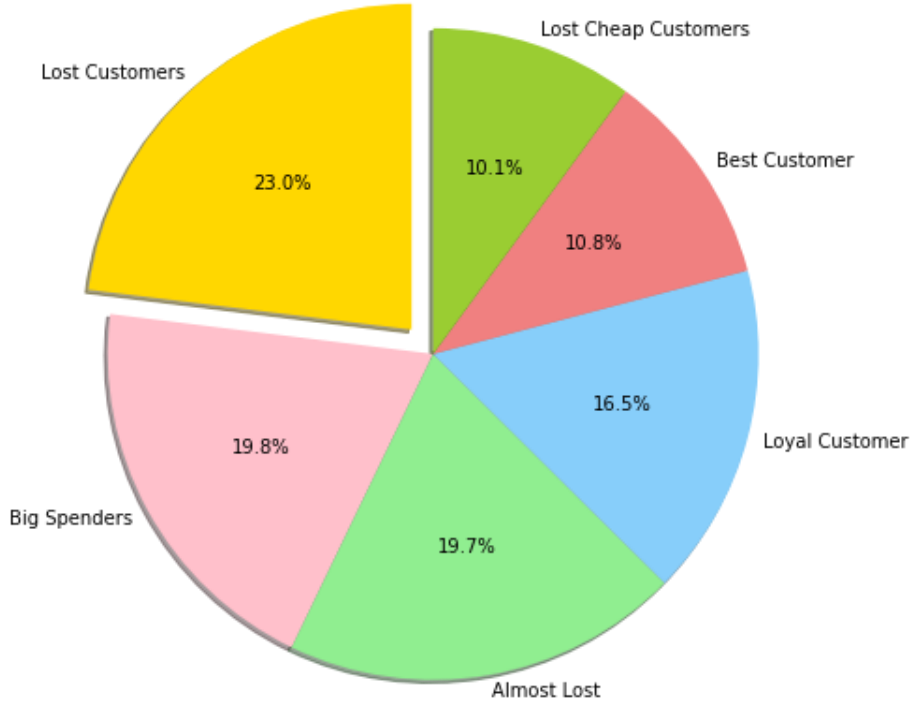
```
def rfm_level_segment(data) :
    if data == 12 :
        return 'Best Customer'
    elif data >= 10 :
        return 'Loyal Customer'
    elif data >= 8 :
        return 'Big Spenders'
    elif data >= 6 :
        return 'Almost Lost'
    elif data >= 4 :
        return 'Lost Customers'
    else :
        return 'Lost Cheap Customers'
```

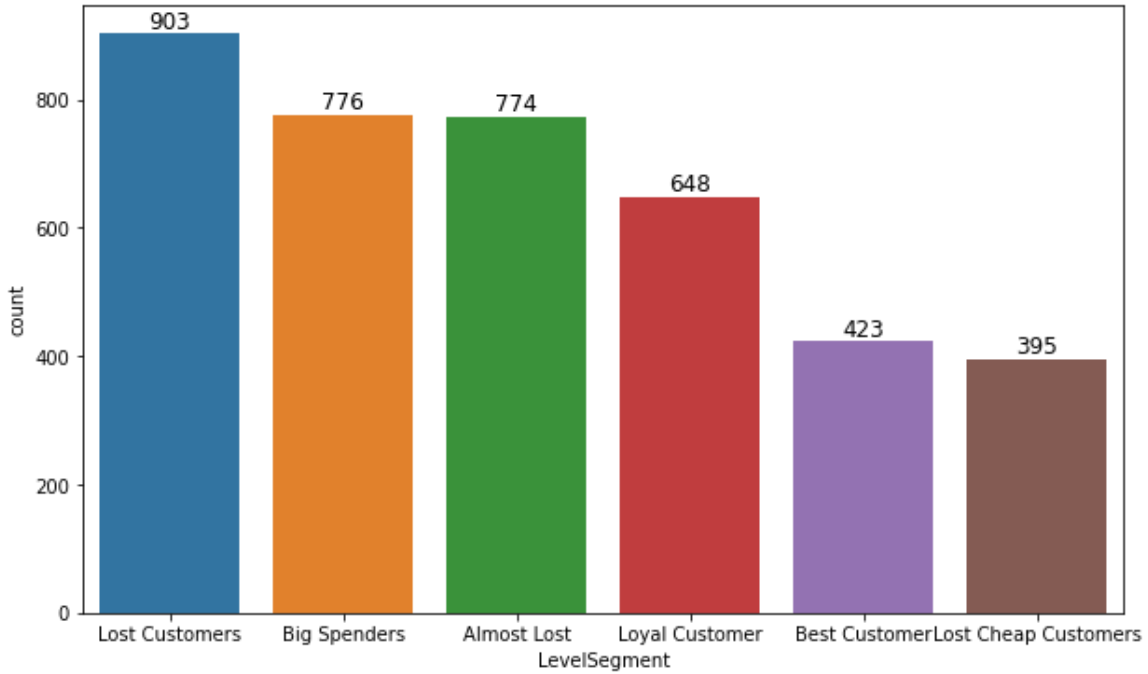
RFM ANALYSIS







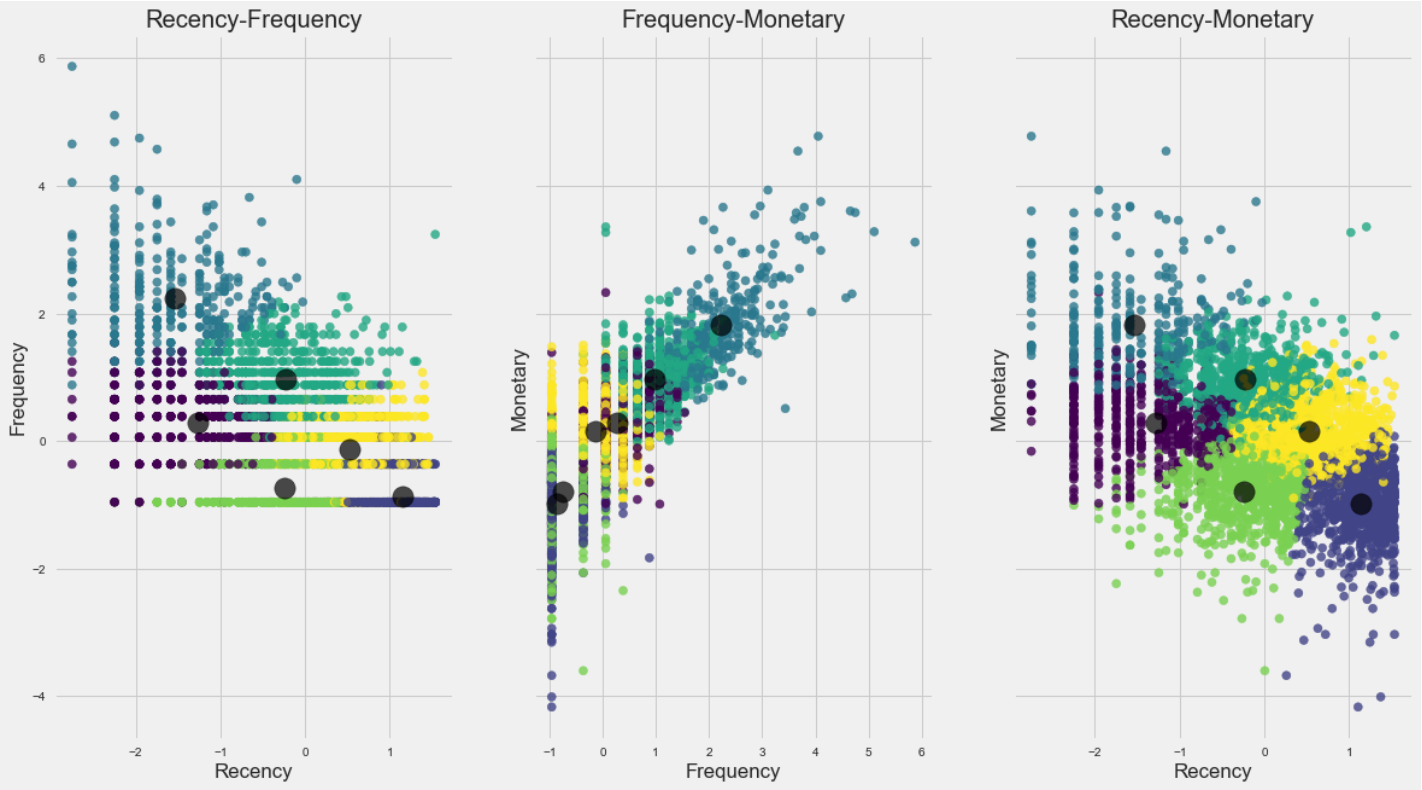




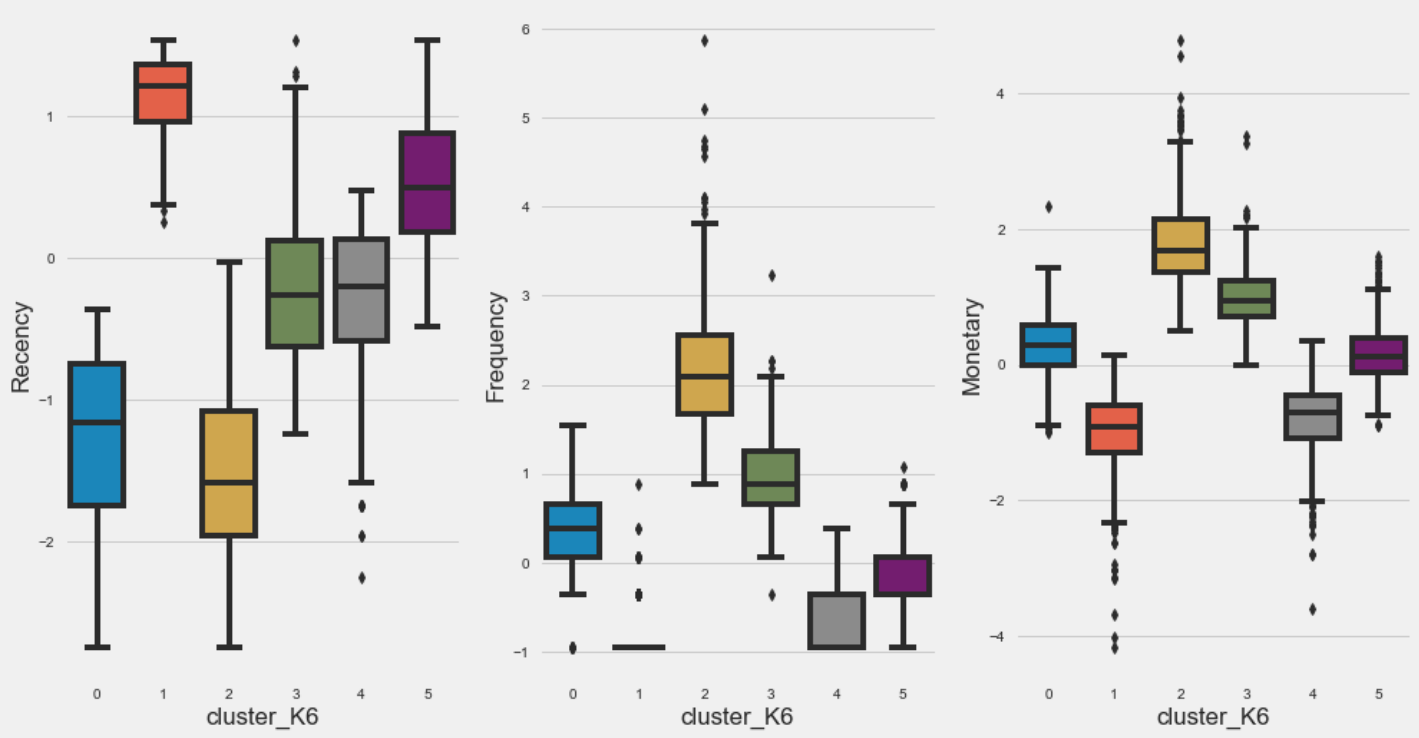
RFM ANALYSIS

K-MEANS

6 CLUSTER

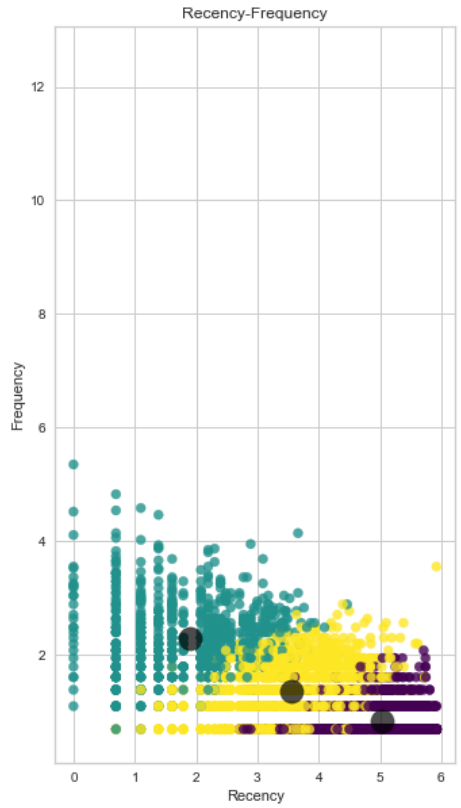


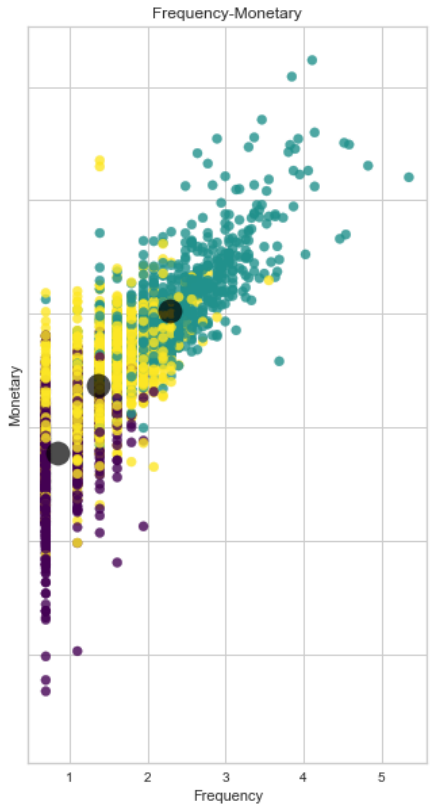
K-MEANS

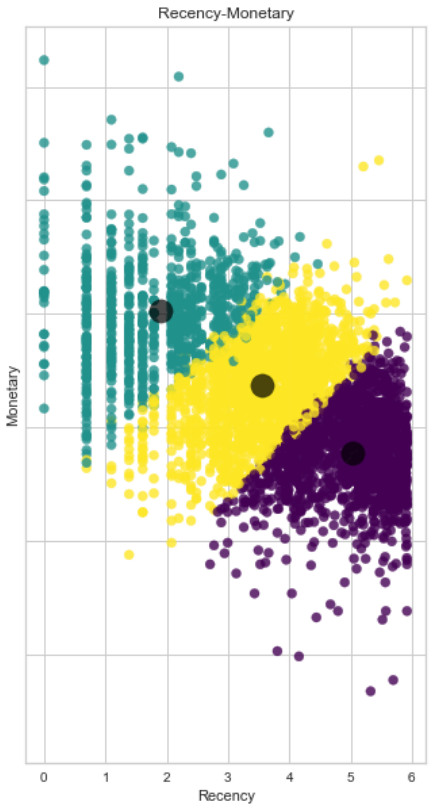

Recency
cluster_K6
Frequency
cluster_K6
Monetary
cluster_K6

K-MEANS

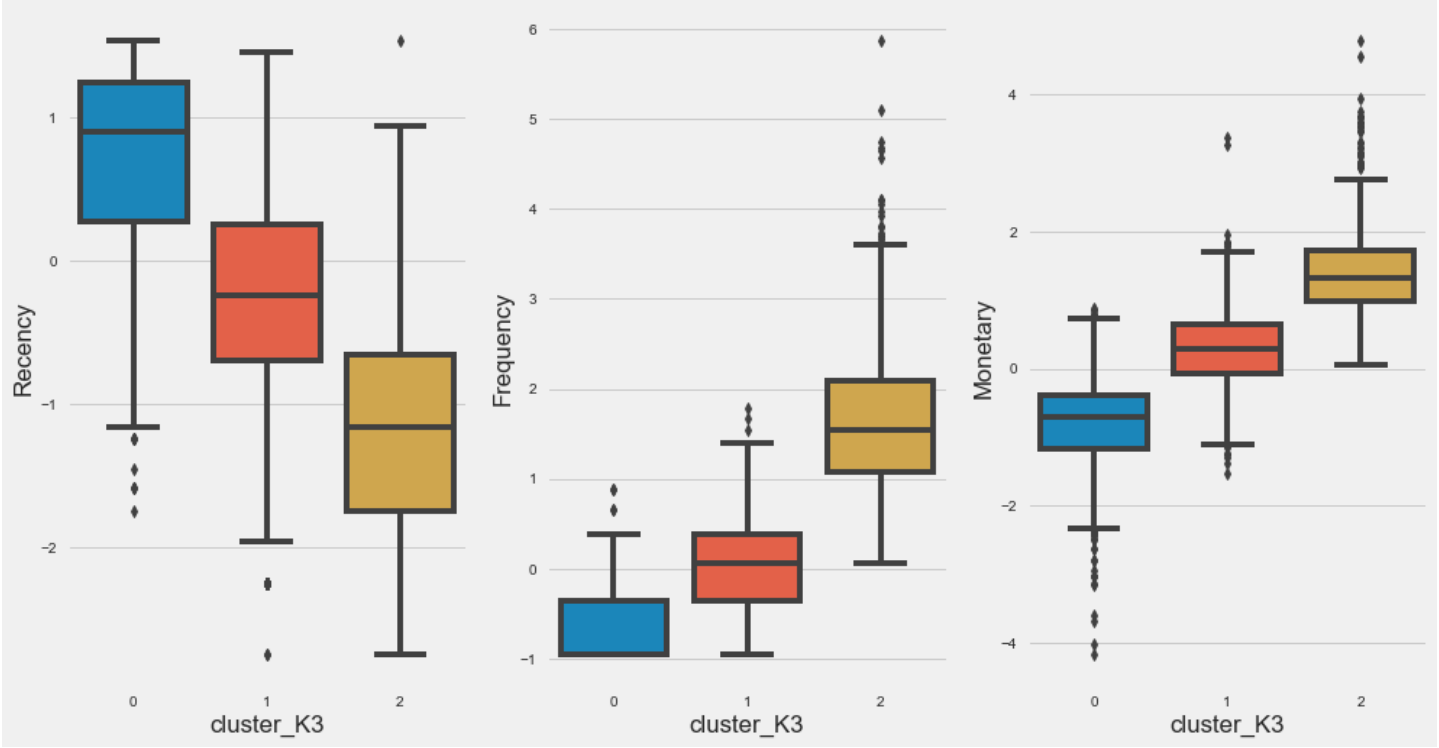
3 CLUSTER





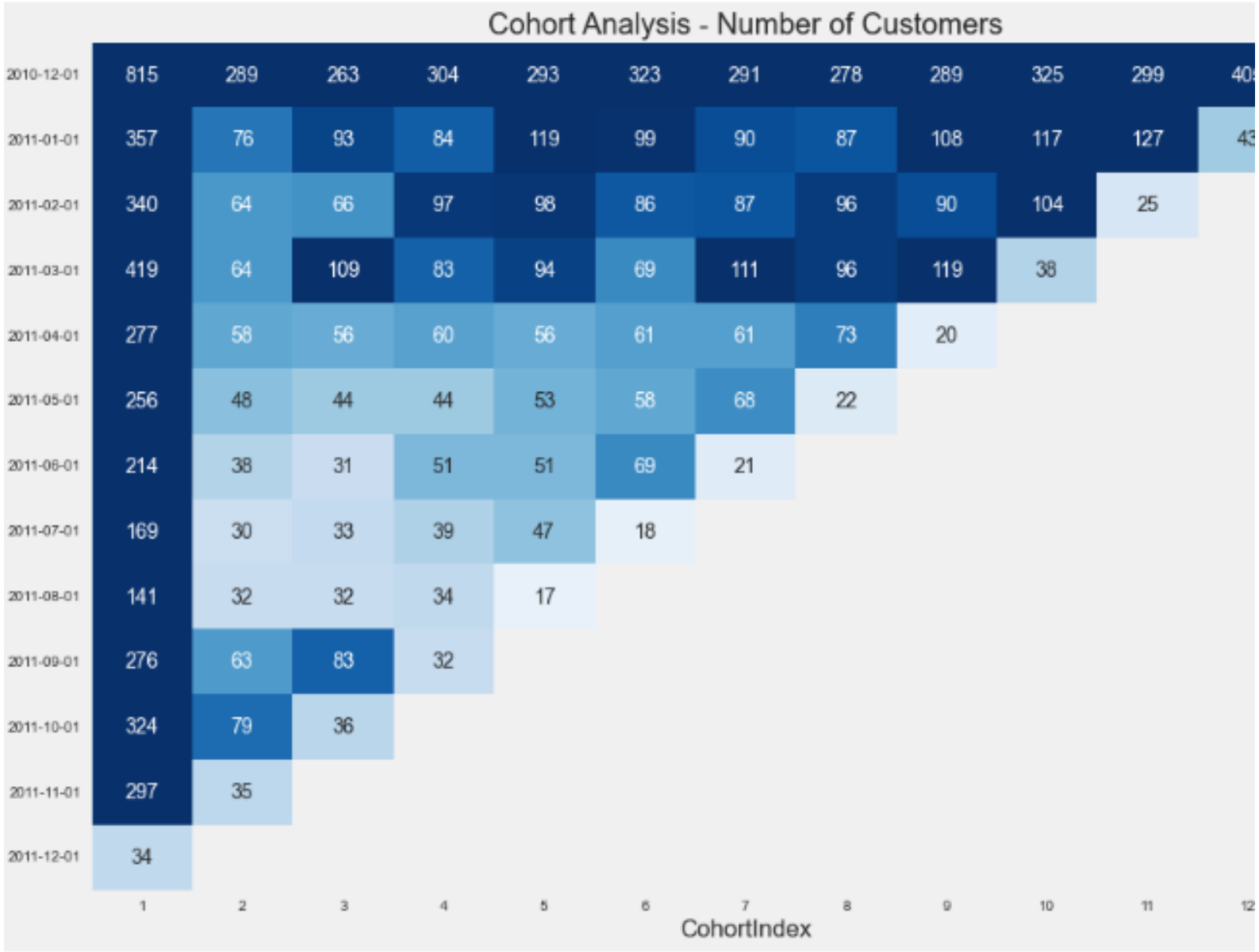


K-MEANS


Recency
1
0
−1
−2
0
1
2
cluster_K3
Frequency
6
5
4
3
2
1
0
−1
0
1
2
cluster_K3
Monetary
4
2
0
−2
−4
0
1
2
cluster_K3

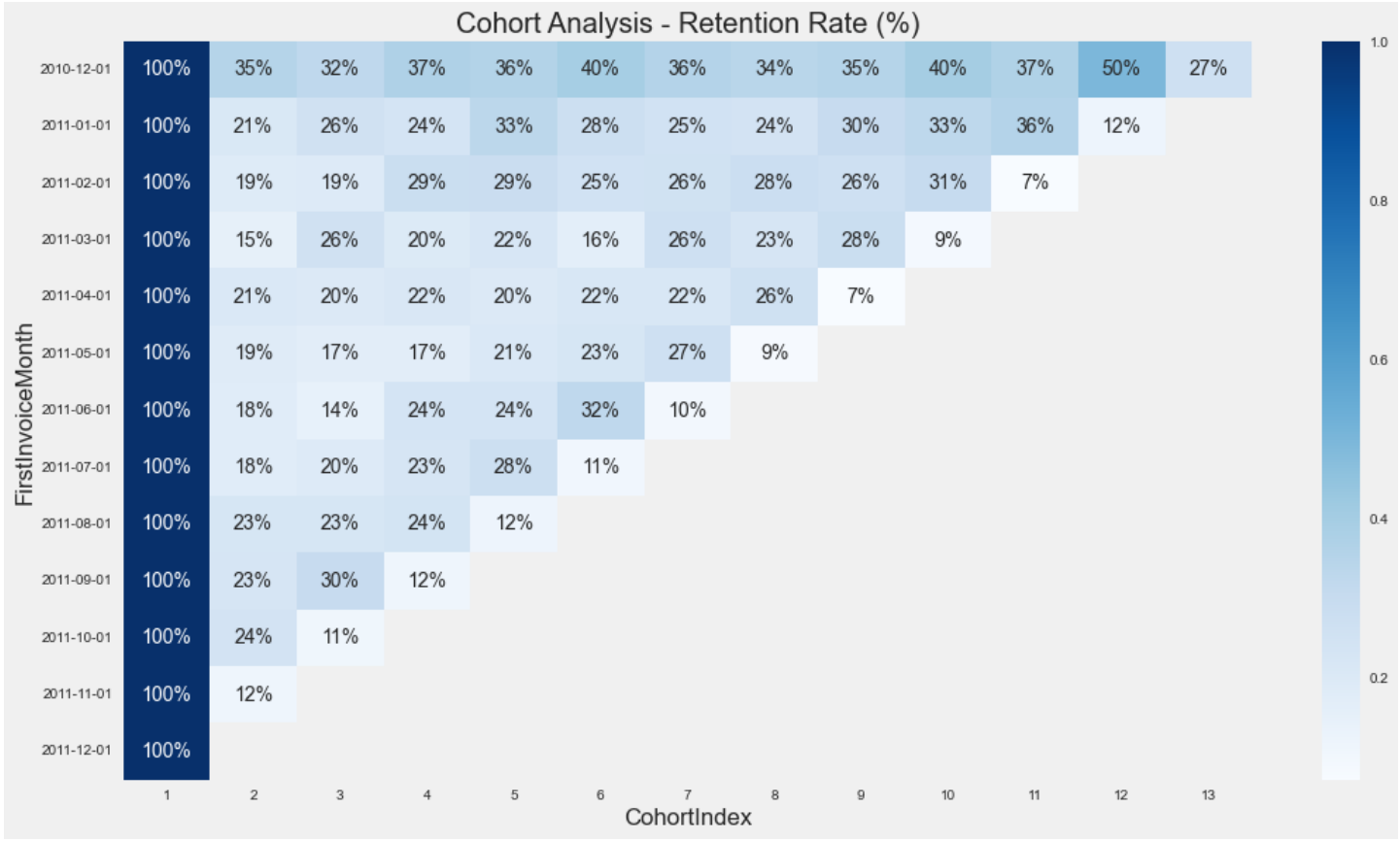
COHORT ANALYSIS

- Number of Customers



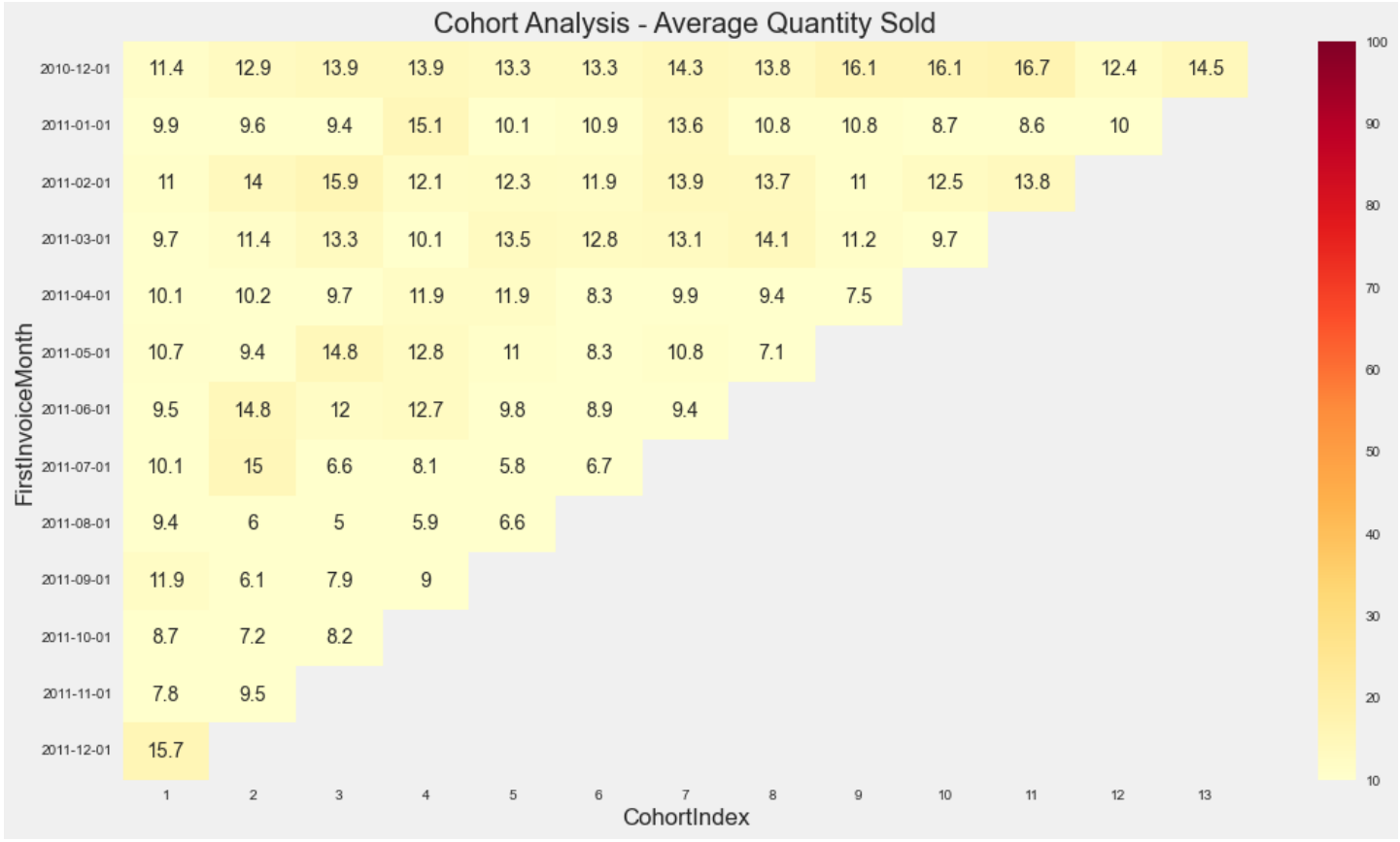
COHORT ANALYSIS

- Retation Rate (%)



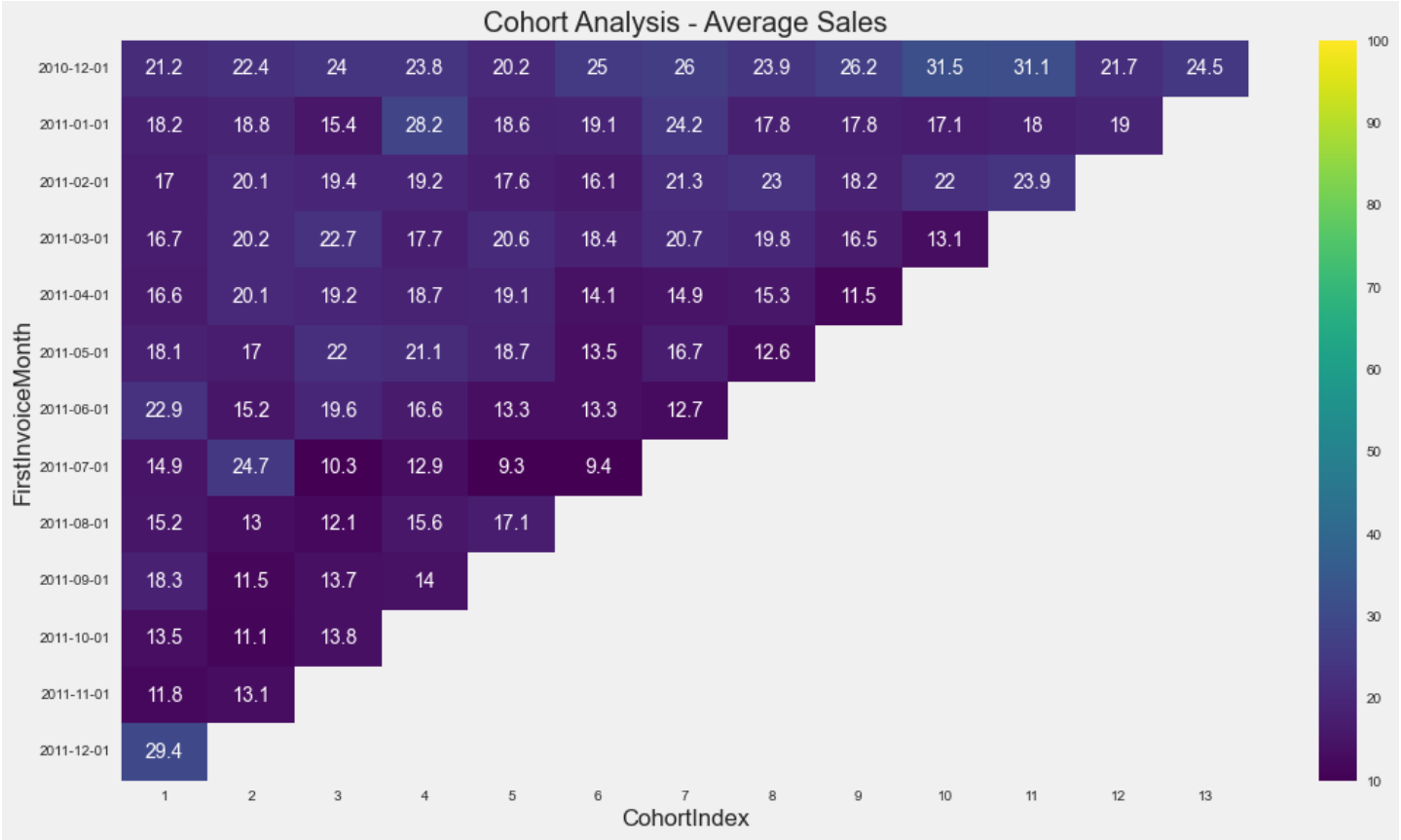
COHORT ANALYSIS

- Avarega Quantity Sold



COHORT ANALYSIS

- Avarege Sales



THANKS