RFM (Recency Frequency Monetary) Analysis:- RFM is a method used for analyzing customer value. It is commonly used in database marketing and direct marketing and has received particular attention in retail and professional services industries

RFM stands for the three dimensions:

Recency – How recently did the customer purchase? Frequency – How often do they purchase? Monetary Value – How much do they spend?

```
In [1]: #importing all important package..
import pandas as pd
import numpy as np
import datetime
```

Description Quantity InvoiceDate Price

Customer

Country

In [2]: #load data into pandas dataframe..
 df = pd.read\_csv("Dataset.csv")
 df.head()

Invoice StockCode

VUL	1 4 1	
	$\Gamma - 1$	

Out[2]:

0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	1.12.2009 07:45	6,95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	1.12.2009 07:45	6,75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	1.12.2009 07:45	6,75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	1.12.2009 07:45	2,1	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	1.12.2009 07:45	1,25	13085.0	United Kingdom
	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	<b>Invoice</b> 489434	StockCode 85048	Description  15CM CHRISTMAS GLASS BALL 20 LIGHTS	Quantity 12	1.12.2009 07:45	<b>Price</b> 6,95		Country  United Kingdom
0			15CM CHRISTMAS GLASS BALL 20		1.12.2009		ID	United
	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	1.12.2009 07:45 1.12.2009	6,95	13085.0	United Kingdom United
1	489434 489434	85048 79323P	15CM CHRISTMAS GLASS BALL 20 LIGHTS PINK CHERRY LIGHTS	12	1.12.2009 07:45 1.12.2009 07:45 1.12.2009	6,95 6,75	13085.0 13085.0	United Kingdom United Kingdom United

## In [3]: #information of dataset.. df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 8 columns):
# Column Non-Null Count Dtype

```
Invoice 1048575 non-null object
 0
 1 StockCode 1048575 non-null object
 2 Description 1044203 non-null object
 3 Quantity 1048575 non-null int64
   InvoiceDate 1048575 non-null object
 4
 5 Price 1048575 non-null object
  Customer ID 811893 non-null float64
   Country 1048575 non-null object
 7
dtypes: float64(1), int64(1), object(6)
memory usage: 64.0+ MB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 8 columns):
# Column Non-Null Count Dtype
--- ----
               -----
                               ----
0 Invoice 1048575 non-null object
1 StockCode 1048575 non-null object
 2 Description 1044203 non-null object
 3 Quantity 1048575 non-null int64
 4 InvoiceDate 1048575 non-null object
 5 Price 1048575 non-null object
   Customer ID 811893 non-null float64
7
    Country 1048575 non-null object
dtypes: float64(1), int64(1), object(6)
memory usage: 64.0+ MB
df.head()
```

In [4]: df['Price'] = df['Price'].str.replace(',','.')

011+[4].

Out[4]:		Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
	0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	1.12.2009 07:45	6.95	13085.0	United Kingdom
	1	489434	79323P	PINK CHERRY LIGHTS	12	1.12.2009 07:45	6.75	13085.0	United Kingdom
	2	489434	79323W	WHITE CHERRY LIGHTS	12	1.12.2009 07:45	6.75	13085.0	United Kingdom
	3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	1.12.2009 07:45	2.1	13085.0	United Kingdom
	4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	1.12.2009 07:45	1.25	13085.0	United Kingdom
Out[4]:		Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
Out[4]:	0	<b>Invoice</b> 489434	StockCode 85048	Description  15CM CHRISTMAS GLASS BALL 20 LIGHTS	<b>Quantity</b>	1.12.2009 07:45	<b>Price</b> 6.95		Country United Kingdom
Out[4]:	0			15CM CHRISTMAS GLASS BALL 20		1.12.2009		ID	United
Out[4]:		489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	1.12.2009 07:45 1.12.2009	6.95	13085.0	United Kingdom United
Out[4]:	1	489434 489434	85048 79323P	15CM CHRISTMAS GLASS BALL 20 LIGHTS PINK CHERRY LIGHTS	12	1.12.2009 07:45 1.12.2009 07:45 1.12.2009	6.95 6.75	13085.0 13085.0	United Kingdom United Kingdom United

Customer

```
df = df.astype({"Price":"float64"})
       df.info()
       <class 'pandas.core.frame.DataFrame'>
       RangeIndex: 1048575 entries, 0 to 1048574
       Data columns (total 8 columns):
        # Column Non-Null Count Dtype
                       _____
        0 Invoice 1048575 non-null object
1 StockCode 1048575 non-null object
          Description 1044203 non-null object
        3 Quantity 1048575 non-null int64
        4 InvoiceDate 1048575 non-null object
        5 Price 1048575 non-null float64
          Customer ID 811893 non-null float64
        7 Country 1048575 non-null object
       dtypes: float64(2), int64(1), object(5)
       memory usage: 64.0+ MB
       <class 'pandas.core.frame.DataFrame'>
       RangeIndex: 1048575 entries, 0 to 1048574
       Data columns (total 8 columns):
          Column Non-Null Count Dtype
       ---
                      -----
        0 Invoice 1048575 non-null object
1 StockCode 1048575 non-null object
        2
          Description 1044203 non-null object
        3 Quantity 1048575 non-null int64
        4 InvoiceDate 1048575 non-null object
          Price 1048575 non-null float64
          Customer ID 811893 non-null float64
        7 Country 1048575 non-null object
       dtypes: float64(2), int64(1), object(5)
       memory usage: 64.0+ MB
       # change datatype of InvoiceDate column
In [6]:
       df['InvoiceDate'] = pd.to datetime(df['InvoiceDate'], errors='coerce')
       df.info()
       <class 'pandas.core.frame.DataFrame'>
       RangeIndex: 1048575 entries, 0 to 1048574
       Data columns (total 8 columns):
        # Column Non-Null Count Dtype
                      -----
       ---
        0 Invoice 1048575 non-null object
1 StockCode 1048575 non-null object
          Description 1044203 non-null object
        3 Quantity 1048575 non-null int64
        4 InvoiceDate 1048575 non-null datetime64[ns]
          Price 1048575 non-null float64
          Customer ID 811893 non-null float64
        7 Country 1048575 non-null object
       dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
       memory usage: 64.0+ MB
       <class 'pandas.core.frame.DataFrame'>
       RangeIndex: 1048575 entries, 0 to 1048574
       Data columns (total 8 columns):
          Column Non-Null Count
                                       Dtype
       ---
                      -----
        0 Invoice 1048575 non-null object
1 StockCode 1048575 non-null object
          Description 1044203 non-null object
        2
        3 Quantity 1048575 non-null int64
          InvoiceDate 1048575 non-null datetime64[ns]
        4
          Price 1048575 non-null float64
          Customer ID 811893 non-null float64
          Country 1048575 non-null object
```

dtypes: datetime64[ns](1), float64(2), int64(1), object(4)memory usage: 64.0+ MB In [7]: # change datatype of Customer ID column df['Customer ID'] = df['Customer ID'].astype(str).apply(lambda x: x.replace('.0','')) df.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 1048575 entries, 0 to 1048574 Data columns (total 8 columns): # Column Non-Null Count Dtype --------0 Invoice 1048575 non-null object 1 StockCode 1048575 non-null object 2 Description 1044203 non-null object 3 Quantity 1048575 non-null int64 4 InvoiceDate 1048575 non-null datetime64[ns] 5 Price 1048575 non-null float64 6 Customer ID 1048575 non-null object 7 Country 1048575 non-null object dtypes: datetime64[ns](1), float64(1), int64(1), object(5)memory usage: 64.0+ MB <class 'pandas.core.frame.DataFrame'> RangeIndex: 1048575 entries, 0 to 1048574 Data columns (total 8 columns):

#	Column	Non-Null Count	Dtype
0	Invoice	1048575 non-null	object
1	StockCode	1048575 non-null	object
2	Description	1044203 non-null	object
3	Quantity	1048575 non-null	int64
4	InvoiceDate	1048575 non-null	datetime64[ns]
5	Price	1048575 non-null	float64
6	Customer ID	1048575 non-null	object
7	Country	1048575 non-null	object
dtyp	es: datetime6	4[ns](1), float64(	1), int64(1), object(5)
memo	ry usage: 64.	0+ MB	

In [8]: # rename the column name for Customer ID to Customer ID for future operation

df.rename(columns = {'Customer ID':'Customer ID'}, inplace = True) df.head()

Out[8]:		Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer_ID	Country
	0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-01-12 07:45:00	6.95	13085	United Kingdom
	1	489434	79323P	PINK CHERRY LIGHTS	12	2009-01-12 07:45:00	6.75	13085	United Kingdom
	2	489434	79323W	WHITE CHERRY LIGHTS	12	2009-01-12 07:45:00	6.75	13085	United Kingdom
	3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-01-12 07:45:00	2.10	13085	United Kingdom
	4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-01-12 07:45:00	1.25	13085	United Kingdom
Out[8]:		Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer_ID	Country
	0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-01-12 07:45:00	6.95	13085	United Kingdom
	1	489434	79323P	PINK CHERRY LIGHTS	12	2009-01-12 07:45:00	6.75	13085	United Kingdom

2	489434	79323W	WHITE CHERRY LIGHTS	12	2009-01-12 07:45:00	6.75	13085	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-01-12 07:45:00	2.10	13085	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-01-12 07:45:00	1.25	13085	United Kingdom

In [9]: # creating new column total revenue as Tot\_rev
df["Tot\_Rev"]=(df['Price']\*df['Quantity'])
df.head()

Out[9]:		Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer_ID	Country	Tot_Rev
	0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-01-12 07:45:00	6.95	13085	United Kingdom	83.4
	1	489434	79323P	PINK CHERRY LIGHTS	12	2009-01-12 07:45:00	6.75	13085	United Kingdom	81.0
	2	489434	79323W	WHITE CHERRY LIGHTS	12	2009-01-12 07:45:00	6.75	13085	United Kingdom	81.0
	3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-01-12 07:45:00	2.10	13085	United Kingdom	100.8
	4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-01-12 07:45:00	1.25	13085	United Kingdom	30.0
Out[9]:		Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer_ID	Country	Tot_Rev
Out[9]:	0	<b>Invoice</b> 489434	StockCode 85048	Description  15CM CHRISTMAS GLASS BALL 20 LIGHTS	Quantity 12	2009-01-12 07:45:00	<b>Price</b> 6.95	Customer_ID  13085	Country  United Kingdom	<b>Tot_Rev</b> 83.4
Out[9]:	0			15CM CHRISTMAS		2009-01-12			United	
Out[9]:		489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-01-12 07:45:00 2009-01-12	6.95	13085	United Kingdom United	83.4
Out[9]:	1	489434 489434	85048 79323P	15CM CHRISTMAS GLASS BALL 20 LIGHTS PINK CHERRY LIGHTS	12	2009-01-12 07:45:00 2009-01-12 07:45:00 2009-01-12	6.95 6.75	13085 13085	United Kingdom United Kingdom	83.4

In [10]: #sorting the Customer\_ID
 df = df.sort\_values(by="Customer\_ID", ascending=0)
 df.head()

Out[10]:		Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer_ID	Country	Tot_Rev
	524287	538074	21743	NaN	-80	2010-09-12 14:11:00	0.00	nan	United Kingdom	-0.00
	676234	549524	22359	GLASS JAR KINGS CHOICE	1	2011-08-04 15:42:00	5.79	nan	United Kingdom	5.79
	676236	549524	22362	GLASS JAR PEACOCK BATH SALTS	1	2011-08-04 15:42:00	5.79	nan	United Kingdom	5.79
	676237	549524	22411	JUMBO SHOPPER VINTAGE RED PAISLEY	6	2011-08-04 15:42:00	4.13	nan	United Kingdom	24.78

				BIN CREAM		15:42:00			Kingdom	
ut[10]:		Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer_ID	Country	Tot_Rev
	524287	538074	21743	NaN	-80	2010-09-12 14:11:00	0.00	nan	United Kingdom	-0.00
	676234	549524	22359	GLASS JAR KINGS CHOICE	1	2011-08-04 15:42:00	5.79	nan	United Kingdom	5.79
	676236	549524	22362	GLASS JAR PEACOCK BATH SALTS	1	2011-08-04 15:42:00	5.79	nan	United Kingdom	5.79
	676237	549524	22411	JUMBO SHOPPER VINTAGE RED PAISLEY	6	2011-08-04 15:42:00	4.13	nan	United Kingdom	24.78
	676238	549524	22424	ENAMEL BREAD BIN CREAM	1	2011-08-04 15:42:00	24.96	nan	United Kingdom	24.96

1 2011-08-04 24.96 nan

Description Quantity InvoiceDate Price Customer\_ID Country Tot\_Rev

United

24.96

ENAMEL BREAD

In [11]: #removing Not a Number rows in dataframe
 df=df.dropna()
 df.head()

Invoice StockCode

Out[11]:

**676238** 549524

22424

L 3				•	-			_	•	_
	676234	549524	22359	GLASS JAR KINGS CHOICE	1	2011-08-04 15:42:00	5.79	nan	United Kingdom	5.79
	676236	549524	22362	GLASS JAR PEACOCK BATH SALTS	1	2011-08-04 15:42:00	5.79	nan	United Kingdom	5.79
	676237	549524	22411	JUMBO SHOPPER VINTAGE RED PAISLEY	6	2011-08-04 15:42:00	4.13	nan	United Kingdom	24.78
	676238	549524	22424	ENAMEL BREAD BIN CREAM	1	2011-08-04 15:42:00	24.96	nan	United Kingdom	24.96
	676239	549524	22425	ENAMEL COLANDER CREAM	1	2011-08-04 15:42:00	9.96	nan	United Kingdom	9.96
Out[11]:		Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer_ID	Country	Tot_Rev
Out[11]:	676234	<b>Invoice</b> 549524	StockCode 22359	<b>Description</b> GLASS JAR KINGS CHOICE	<b>Quantity</b>	2011-08-04 15:42:00	<b>Price</b> 5.79	Customer_ID nan	Country United Kingdom	<b>Tot_Rev</b> 5.79
Out[11]:	676234 676236	549524		GLASS JAR KINGS		2011-08-04			United	
Out[11]:		549524 549524	22359	GLASS JAR KINGS CHOICE GLASS JAR PEACOCK BATH	1	2011-08-04 15:42:00 2011-08-04	5.79	nan	United Kingdom United	5.79
Out[11]:	676236	549524 549524 549524	22359	GLASS JAR KINGS CHOICE GLASS JAR PEACOCK BATH SALTS JUMBO SHOPPER VINTAGE RED	1	2011-08-04 15:42:00 2011-08-04 15:42:00 2011-08-04	5.79	nan	United Kingdom United Kingdom	5.79

In [12]: # In customer column their are un-related values need to removed df.drop(df[df['Customer ID'] == 'nan'].index, inplace = True) df.head() Description Quantity InvoiceDate Price Customer\_ID Country Tot\_Rev Out[12]: Invoice StockCode WATERMELON 2010-05-17 United **199827** 508581 20960 20 1.25 18287 25.0 11:55:00 Kingdom **BATH SPONGE** HEN HOUSE

DECORATION

12

2010-05-17

11:55:00

1.65

United

Kingdom

19.8

18287

	724195	554065	22064	PINK DOUGHNUT TRINKET POT	12	2011-05-22 10:39:00	1.65	18287	United Kingdom	19.8
	199826	508581	22285	DECORATION HEN ON NEST, HANGING	12	2010-05-17 11:55:00	1.65	18287	United Kingdom	19.8
	199832	508581	85041	SET/4 PINK ORCHID CANDLES IN BOWL	12	2010-05-17 11:55:00	1.65	18287	United Kingdom	19.8
Out[12]:		Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer_ID	Country	Tot_Rev
	199827	508581	20960	WATERMELON BATH SPONGE	20	2010-05-17 11:55:00	1.25	18287	United Kingdom	25.0
	199825	508581	22284	HEN HOUSE DECORATION	12	2010-05-17 11:55:00	1.65	18287	United Kingdom	19.8
	724195	554065	22064	PINK DOUGHNUT TRINKET POT	12	2011-05-22 10:39:00	1.65	18287	United Kingdom	19.8
	199826	508581	22285	DECORATION HEN ON NEST, HANGING	12	2010-05-17 11:55:00	1.65	18287	United Kingdom	19.8
	199832	508581	85041	SET/4 PINK ORCHID CANDLES IN BOWL	12	2010-05-17 11:55:00	1.65	18287	United Kingdom	19.8

```
In [13]: #create empty data frame
         RFM = pd.DataFrame()
         print(RFM)
```

**199825** 508581

22284

Empty DataFrame Columns: [] Index: [] Empty DataFrame

Columns: [] Index: []

In [14]: # calculating no of time purchase made by each customer RFM['Freq'] = df.groupby("Customer ID").StockCode.count() RFM.head()

## Out[14]: Freq

Customer_ID	
12346	48
12347	242
12348	51
12349	180

```
Out[14]:
                      Freq
          Customer_ID
               12346
                        48
               12347
                       242
               12348
                        51
               12349
                       180
               12350
                        17
In [15]:
          # calculating Total revenue purchased by each customer
          RFM['Tot Rev'] = df.groupby("Customer ID").Tot Rev.sum()
          RFM.head()
Out[15]:
                      Freq Tot_Rev
          Customer_ID
               12346
                        48
                             -64.68
               12347
                       242 5408.50
               12348
                        51
                            2019.40
               12349
                       180
                           4404.54
               12350
                        17
                             334.40
Out[15]:
                      Freq Tot_Rev
          Customer ID
               12346
                        48
                             -64.68
               12347
                       242 5408.50
               12348
                           2019.40
                        51
               12349
                       180 4404.54
               12350
                        17
                             334.40
In [16]:
          # calculating Average revenue purchased by each customer
          RFM['Avg Rev'] = RFM['Tot Rev']/RFM['Freq']
          RFM.head()
Out[16]:
                      Freq Tot_Rev
                                   Avg_Rev
          Customer_ID
               12346
                        48
                             -64.68
                                   -1.347500
               12347
                       242 5408.50 22.349174
               12348
                        51
                            2019.40 39.596078
               12349
                       180
                           4404.54 24.469667
               12350
                        17
                             334.40 19.670588
```

12350

Out[16]:

Freq Tot\_Rev

Avg\_Rev

17

```
Customer_ID
               12346
                        48
                           -64.68 -1.347500
               12347
                       242 5408.50 22.349174
               12348
                        51 2019.40 39.596078
               12349
                       180 4404.54 24.469667
               12350
                        17
                             334.40 19.670588
In [17]: #sorting by Customer ID
          RFM = RFM.sort values(by="Customer ID", ascending=0)
          RFM.head()
Out[17]:
                      Freq Tot_Rev
                                     Avg_Rev
          Customer ID
               18287
                       156 4177.89 26.781346
               18286
                        70 1188.43 16.977571
               18285
                        12
                            427.00 35.583333
               18284
                        29
                             436.68 15.057931
               18283
                       936 2528.65
                                    2.701549
Out[17]:
                      Freq Tot_Rev Avg_Rev
          Customer_ID
               18287
                       156 4177.89 26.781346
               18286
                        70 1188.43 16.977571
               18285
                        12 427.00 35.583333
               18284
                        29
                            436.68 15.057931
                       936 2528.65 2.701549
               18283
          RFM = RFM.dropna()
In [18]:
          RFM.head()
Out[18]:
                      Freq Tot_Rev
                                     Avg_Rev
          Customer_ID
               18287
                       156 4177.89 26.781346
               18286
                        70 1188.43 16.977571
               18285
                        12
                           427.00 35.583333
               18284
                        29
                            436.68 15.057931
               18283
                       936 2528.65
                                    2.701549
Out[18]:
                      Freq Tot_Rev Avg_Rev
          Customer_ID
               18287
                       156 4177.89 26.781346
```

18286

70 1188.43 16.977571

```
18284
                         436.68 15.057931
              18283
                     936 2528.65 2.701549
         #Creating the column called order date(which will taken has todays date)
In [19]:
         RFM['order date'] = '1.1.2012 00:00'
         # change datatype of order date column
In [20]:
         RFM['order date'] = pd.to datetime(RFM['order date'] , errors='coerce')
         RFM.info()
         <class 'pandas.core.frame.DataFrame'>
         Index: 5924 entries, 18287 to 12346
         Data columns (total 4 columns):
          # Column Non-Null Count Dtype
         --- ----
                          _____
            Freq
                          5924 non-null int64
          0
          1 Tot_Rev 5924 non-null float64
2 Avg_Rev 5924 non-null float64
          3 order date 5924 non-null datetime64[ns]
         dtypes: datetime64[ns](1), float64(2), int64(1)
         memory usage: 231.4+ KB
         <class 'pandas.core.frame.DataFrame'>
         Index: 5924 entries, 18287 to 12346
         Data columns (total 4 columns):
          # Column Non-Null Count Dtype
                         -----
            Freq
                         5924 non-null int64
          0
          1 Tot_Rev 5924 non-null float64
2 Avg Rev 5924 non-null float64
          3 order date 5924 non-null datetime64[ns]
         dtypes: datetime64[ns](1), float64(2), int64(1)
         memory usage: 231.4+ KB
         # calulating recent date purchansed by each cust
In [21]:
         RFM['MAX InvoiceDate'] = df.groupby("Customer ID").InvoiceDate.max()
         RFM.head()
Out[21]:
                    Freq Tot_Rev Avg_Rev order_date
                                                     MAX InvoiceDate
         Customer ID
              18287
                     156 4177.89 26.781346 2012-01-01 2011-12-10 10:23:00
              18286
                      70 1188.43 16.977571 2012-01-01 2010-08-20 11:57:00
              18285
                          427.00 35.583333 2012-01-01 2010-02-17 10:24:00
              18284
                      29
                          436.68 15.057931 2012-01-01 2010-06-10 12:31:00
              18283
                     936 2528.65
                                 2.701549 2012-01-01 2011-11-30 12:59:00
Out[21]:
                    Freq Tot_Rev
                                 Avg_Rev order_date
                                                     MAX_InvoiceDate
         Customer_ID
              18287
                     156 4177.89 26.781346 2012-01-01 2011-12-10 10:23:00
              18286
                      70 1188.43 16.977571 2012-01-01 2010-08-20 11:57:00
              18285
                      12
                          427.00 35.583333 2012-01-01 2010-02-17 10:24:00
              18284
                      29
                         436.68 15.057931 2012-01-01 2010-06-10 12:31:00
```

18285

427.00 35.583333

```
# calulating duration from last purchased
In [22]:
           RFM['duration'] = RFM['order date'] - RFM['MAX InvoiceDate']
           RFM.head()
Out[22]:
                        Freq Tot_Rev
                                       Avg_Rev order_date
                                                               MAX_InvoiceDate
                                                                                         duration
           Customer_ID
                 18287
                         156
                              4177.89 26.781346 2012-01-01 2011-12-10 10:23:00
                                                                                  21 days 13:37:00
                 18286
                          70
                              1188.43
                                       16.977571 2012-01-01
                                                              2010-08-20 11:57:00
                                                                                 498 days 12:03:00
                 18285
                          12
                               427.00
                                       35.583333 2012-01-01
                                                              2010-02-17 10:24:00
                                                                                 682 days 13:36:00
                 18284
                          29
                               436.68
                                       15.057931 2012-01-01
                                                              2010-06-10 12:31:00
                                                                                 569 days 11:29:00
                 18283
                              2528.65
                                                              2011-11-30 12:59:00
                         936
                                        2.701549
                                                  2012-01-01
                                                                                   31 days 11:01:00
                        Freq Tot Rev
Out[22]:
                                        Avg_Rev order_date
                                                                MAX_InvoiceDate
                                                                                         duration
           Customer ID
                 18287
                         156
                              4177.89 26.781346 2012-01-01 2011-12-10 10:23:00
                                                                                   21 days 13:37:00
                 18286
                              1188.43 16.977571
                                                  2012-01-01 2010-08-20 11:57:00
                                                                                 498 days 12:03:00
                          70
                 18285
                          12
                                427.00 35.583333 2012-01-01
                                                              2010-02-17 10:24:00
                                                                                 682 days 13:36:00
                 18284
                          29
                                436.68 15.057931 2012-01-01
                                                              2010-06-10 12:31:00
                                                                                  569 days 11:29:00
                 18283
                         936
                              2528.65
                                        2.701549 2012-01-01 2011-11-30 12:59:00
                                                                                   31 days 11:01:00
          RFM = RFM.drop(['order date', 'MAX InvoiceDate'], axis=1)
In [23]:
           RFM.head()
                        Freq Tot_Rev
                                                         duration
Out[23]:
                                        Avg_Rev
           Customer_ID
                 18287
                         156
                              4177.89 26.781346
                                                   21 days 13:37:00
                 18286
                          70
                              1188.43
                                       16.977571
                                                  498 days 12:03:00
                 18285
                          12
                                427.00
                                      35.583333
                                                  682 days 13:36:00
                 18284
                          29
                                436.68
                                      15.057931
                                                  569 days 11:29:00
                 18283
                         936
                               2528.65
                                        2.701549
                                                   31 days 11:01:00
Out[23]:
                        Freq Tot Rev
                                                         duration
                                        Avg_Rev
           Customer ID
                 18287
                         156
                              4177.89 26.781346
                                                   21 days 13:37:00
                 18286
                          70
                              1188.43 16.977571
                                                  498 days 12:03:00
                 18285
                          12
                               427.00 35.583333
                                                  682 days 13:36:00
                 18284
                          29
                                436.68
                                       15.057931
                                                  569 days 11:29:00
                 18283
                         936
                              2528.65
                                        2.701549
                                                   31 days 11:01:00
```

In [24]: # this is RFM (Recency Frequency Monetary) table for all customers
 RFM = RFM.reset index()

```
RFM.head()
Out[24]:
               Customer ID
                             Freq
                                   Tot Rev
                                              Avg_Rev
                                                                duration
           0
                     18287
                              156
                                    4177.89
                                             26.781346
                                                          21 days 13:37:00
           1
                     18286
                               70
                                    1188.43
                                             16.977571
                                                         498 days 12:03:00
           2
                     18285
                               12
                                     427.00
                                             35.583333
                                                        682 days 13:36:00
            3
                     18284
                               29
                                     436.68
                                             15.057931
                                                         569 days 11:29:00
                              936
                                    2528.65
            4
                      18283
                                              2.701549
                                                          31 days 11:01:00
               Customer_ID
Out[24]:
                                   Tot Rev
                                              Avg_Rev
                                                                duration
                             Freq
           0
                     18287
                                                          21 days 13:37:00
                              156
                                    4177.89
                                             26.781346
            1
                     18286
                               70
                                    1188.43
                                            16.977571
                                                        498 days 12:03:00
           2
                     18285
                               12
                                             35.583333
                                                        682 days 13:36:00
                                     427.00
                                            15.057931
           3
                     18284
                               29
                                     436.68
                                                         569 days 11:29:00
           4
                     18283
                              936
                                    2528.65
                                              2.701549
                                                          31 days 11:01:00
            #creating 10 different buckets for duration (Deciles)
In [25]:
            RFM['Recency Deciles'] = pd.qcut(RFM['duration'],10)
            RFM.head()
               Customer ID
                             Freq Tot_Rev
                                              Avg_Rev
                                                                duration
                                                                                                      Recency_Deciles
Out[25]:
                                                          21 days 13:37:00 (21 days 06:40:59.999999999, 32 days 12:04:36]
           0
                     18287
                              156
                                    4177.89
                                             26.781346
           1
                     18286
                               70
                                    1188.43
                                            16.977571
                                                        498 days 12:03:00
                                                                                   (436 days 12:19:24, 557 days 12:04:00]
                                     427.00 35.583333
           2
                     18285
                               12
                                                        682 days 13:36:00
                                                                                  (557 days 12:04:00, 1083 days 14:05:00]
           3
                     18284
                               29
                                     436.68 15.057931
                                                         569 days 11:29:00
                                                                                  (557 days 12:04:00, 1083 days 14:05:00]
           4
                     18283
                              936
                                    2528.65
                                              2.701549
                                                          31 days 11:01:00
                                                                           (21 days 06:40:59.99999999, 32 days 12:04:36]
              Customer_ID
                                              Avg_Rev
                                                                duration
Out[25]:
                             Freq
                                   Tot_Rev
                                                                                                      Recency_Deciles
           0
                     18287
                              156
                                    4177.89
                                             26.781346
                                                          21 days 13:37:00
                                                                           (21 days 06:40:59.999999999, 32 days 12:04:36]
                     18286
                               70
                                    1188.43
                                             16.977571
                                                         498 days 12:03:00
                                                                                   (436 days 12:19:24, 557 days 12:04:00]
                                                        682 days 13:36:00
           2
                     18285
                               12
                                     427.00
                                             35.583333
                                                                                  (557 days 12:04:00, 1083 days 14:05:00]
            3
                     18284
                               29
                                     436.68
                                            15.057931
                                                         569 days 11:29:00
                                                                                  (557 days 12:04:00, 1083 days 14:05:00]
                              936
                                    2528.65
                                                                           (21 days 06:40:59.99999999, 32 days 12:04:36]
            4
                     18283
                                              2.701549
                                                          31 days 11:01:00
```

```
In [26]: #create empty dataframe called vintage
  vin = pd.DataFrame()
  print(vin)
```

Empty DataFrame
Columns: []
Index: []
Empty DataFrame
Columns: []
Index: []

In [27]: # calulating total revenue for each bucket
 vin["Tot\_rev"] = RFM.groupby('Recency\_Deciles').Tot\_Rev.sum()

vin.head(10)

Out[27]:		Tot_rev	
	Recency_Deciles		
	(21 days 06:40:59.99999999, 32 days 12:04:36]	5711857.734	
	(32 days 12:04:36, 43 days 13:59:12]	2968252.921	
	(43 days 13:59:12, 61 days 10:34:24.000000001]	2275605.871	
	(61 days 10:34:24.000000001, 87 days 18:32:36.000000017]	1533952.931	
	(87 days 18:32:36.00000017, 135 days 08:26:00]	1262303.344	
	(135 days 08:26:00, 216 days 09:30:36.000000004]	965468.294	
	(216 days 09:30:36.00000004, 345 days 09:39:30]	586407.400	
	(345 days 09:39:30, 436 days 12:19:24]	536943.362	
	(436 days 12:19:24, 557 days 12:04:00]	398391.840	
	(557 days 12:04:00, 1083 days 14:05:00]	164913.601	
t[27]:		Tot_rev	
	Recency_Deciles		
	(21 days 06:40:59.99999999, 32 days 12:04:36]	5711857.734	
	(32 days 12:04:36, 43 days 13:59:12]	2968252.921	
	(43 days 13:59:12, 61 days 10:34:24.000000001]	2275605.871	
	(61 days 10:34:24.000000001, 87 days 18:32:36.000000017]	1533952.931	
	(87 days 18:32:36.00000017, 135 days 08:26:00]	1262303.344	
	(135 days 08:26:00, 216 days 09:30:36.000000004]	965468.294	
	(216 days 09:30:36.000000004, 345 days 09:39:30]	586407.400	
	(345 days 09:39:30, 436 days 12:19:24]	536943.362	
	(436 days 12:19:24, 557 days 12:04:00]	398391.840	
	(557 days 12:04:00, 1083 days 14:05:00]	164913.601	
In [28]:	<pre># calculating cumulative revenue vin["CumSum_Rev"] = vin["Tot_rev"].cumsum() vin.head(10)</pre>		
Out[28]:		Tot_rev	CumSum_Rev
	Recency_Deciles		
	(21 days 06:40:59.99999999, 32 days 12:04:36]	5711857.734	5.711858e+06
	(32 days 12:04:36, 43 days 13:59:12]	2968252.921	8.680111e+06
	(43 days 13:59:12, 61 days 10:34:24.00000001]	2275605.871	1.095572e+07
	(61 days 10:34:24.00000001, 87 days 18:32:36.00000017]	1533952.931	1.248967e+07
	(87 days 18:32:36.00000017, 135 days 08:26:00]	1262303.344	1.375197e+07
	(135 days 08:26:00, 216 days 09:30:36.000000004]	965468.294	1.471744e+07

(216 days 09:30:36.000000004, 345 days 09:39:30] 586407.400 1.530385e+07

	(345 days 09:39:30, 436 days 12:19:24]	536943.362	1.584079e+07	
	(436 days 12:19:24, 557 days 12:04:00]	398391.840	1.623918e+07	
	(557 days 12:04:00, 1083 days 14:05:00]	164913.601	1.640410e+07	
Out[28]:		Tot_rev	CumSum_Rev	
	Recency_Deciles			
	(21 days 06:40:59.99999999, 32 days 12:04:36]	5711857.734	5.711858e+06	
	(32 days 12:04:36, 43 days 13:59:12]	2968252.921	8.680111e+06	
	(43 days 13:59:12, 61 days 10:34:24.000000001]	2275605.871	1.095572e+07	
	(61 days 10:34:24.000000001, 87 days 18:32:36.000000017]	1533952.931	1.248967e+07	
	(87 days 18:32:36.00000017, 135 days 08:26:00]	1262303.344	1.375197e+07	
	(135 days 08:26:00, 216 days 09:30:36.000000004]	965468.294	1.471744e+07	
	(216 days 09:30:36.00000004, 345 days 09:39:30]	586407.400	1.530385e+07	
	(345 days 09:39:30, 436 days 12:19:24]	536943.362	1.584079e+07	
	(436 days 12:19:24, 557 days 12:04:00]	398391.840	1.623918e+07	
	(557 days 12:04:00, 1083 days 14:05:00]	164913.601	1.640410e+07	
In [29]:	<pre># calculating total revenue vin["Tot_rev_Across_Deciles"] = RFM.Tot_Rev vin.head(10)</pre>	.sum()		
Out[29]:		Tot_rev	CumSum_Rev	Tot_rev_Across_Deciles
Out[29]:	Recency_Deciles	Tot_rev	CumSum_Rev	Tot_rev_Across_Deciles
Out[29]:	Recency_Deciles (21 days 06:40:59.99999999, 32 days 12:04:36]			Tot_rev_Across_Deciles  1.640410e+07
Out[29]:				
Out[29]:	(21 days 06:40:59.99999999, 32 days 12:04:36]	5711857.734	5.711858e+06	1.640410e+07
Out[29]:	(21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12]	5711857.734 2968252.921	5.711858e+06 8.680111e+06	1.640410e+07 1.640410e+07
Out[29]:	(21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12] (43 days 13:59:12, 61 days 10:34:24.000000001]	5711857.734 2968252.921 2275605.871	5.711858e+06 8.680111e+06 1.095572e+07	1.640410e+07 1.640410e+07 1.640410e+07
Out[29]:	(21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12] (43 days 13:59:12, 61 days 10:34:24.000000001] (61 days 10:34:24.000000001, 87 days 18:32:36.0000000017]	5711857.734 2968252.921 2275605.871 1533952.931	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07
Out[29]:	(21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12] (43 days 13:59:12, 61 days 10:34:24.000000001] (61 days 10:34:24.000000001, 87 days 18:32:36.000000017] (87 days 18:32:36.000000017, 135 days 08:26:00]	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07
Out[29]:	(21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12] (43 days 13:59:12, 61 days 10:34:24.000000001] (61 days 10:34:24.000000001, 87 days 18:32:36.0000000017] (87 days 18:32:36.000000017, 135 days 08:26:00] (135 days 08:26:00, 216 days 09:30:36.000000004]	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344 965468.294	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07 1.471744e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07
Out[29]:	(21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12] (43 days 13:59:12, 61 days 10:34:24.000000001] (61 days 10:34:24.000000001, 87 days 18:32:36.0000000017] (87 days 18:32:36.000000017, 135 days 08:26:00] (135 days 08:26:00, 216 days 09:30:36.000000004] (216 days 09:30:36.000000004, 345 days 09:39:30]	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344 965468.294 586407.400	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07 1.471744e+07 1.530385e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07
Out[29]:	(21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12] (43 days 13:59:12, 61 days 10:34:24.000000001] (61 days 10:34:24.000000001, 87 days 18:32:36.0000000017] (87 days 18:32:36.000000017, 135 days 08:26:00] (135 days 08:26:00, 216 days 09:30:36.000000004] (216 days 09:30:36.000000004, 345 days 09:39:30] (345 days 09:39:30, 436 days 12:19:24]	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344 965468.294 586407.400 536943.362	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07 1.471744e+07 1.530385e+07 1.584079e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07
	(21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12] (43 days 13:59:12, 61 days 10:34:24.0000000001] (61 days 10:34:24.000000001, 87 days 18:32:36.0000000017] (87 days 18:32:36.000000017, 135 days 08:26:00] (135 days 08:26:00, 216 days 09:30:36.000000004] (216 days 09:30:36.000000004, 345 days 09:39:30] (345 days 09:39:30, 436 days 12:19:24] (436 days 12:19:24, 557 days 12:04:00]	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344 965468.294 586407.400 536943.362 398391.840	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07 1.471744e+07 1.530385e+07 1.584079e+07 1.623918e+07 1.640410e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07
Out[29]:	(21 days 06:40:59.999999999, 32 days 12:04:36]	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344 965468.294 586407.400 536943.362 398391.840 164913.601	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07 1.471744e+07 1.530385e+07 1.584079e+07 1.623918e+07 1.640410e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07
	(21 days 06:40:59.9999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12] (43 days 13:59:12, 61 days 10:34:24.0000000001] (61 days 10:34:24.000000001, 87 days 18:32:36.0000000017] (87 days 18:32:36.000000017, 135 days 08:26:00] (135 days 08:26:00, 216 days 09:30:36.000000004] (216 days 09:30:36.000000004, 345 days 09:39:30] (345 days 09:39:30, 436 days 12:19:24] (436 days 12:19:24, 557 days 12:04:00] (557 days 12:04:00, 1083 days 14:05:00]	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344 965468.294 586407.400 536943.362 398391.840 164913.601 Tot_rev	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07 1.471744e+07 1.530385e+07 1.623918e+07 1.640410e+07  CumSum_Rev	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 Tot_rev_Across_Deciles
	(21 days 06:40:59.999999999, 32 days 12:04:36]	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344 965468.294 586407.400 536943.362 398391.840 164913.601	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07 1.471744e+07 1.530385e+07 1.584079e+07 1.623918e+07 1.640410e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07
	(21 days 06:40:59.999999999, 32 days 12:04:36]	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344 965468.294 586407.400 536943.362 398391.840 164913.601 Tot_rev  5711857.734	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07 1.471744e+07 1.530385e+07 1.623918e+07 1.640410e+07  CumSum_Rev  5.711858e+06	1.640410e+07

(61 days 10:34:24.000000001, 87 days 18:32:36.000000017] 1533952.931 1.248967e+07

1.640410e+07

(87 days 18:32:36.00000017, 135 days 08:26:00]	1262303.344	1.375197e+07	1.640410e+07
(135 days 08:26:00, 216 days 09:30:36.000000004]	965468.294	1.471744e+07	1.640410e+07
(216 days 09:30:36.00000004, 345 days 09:39:30]	586407.400	1.530385e+07	1.640410e+07
(345 days 09:39:30, 436 days 12:19:24]	536943.362	1.584079e+07	1.640410e+07
(436 days 12:19:24, 557 days 12:04:00]	398391.840	1.623918e+07	1.640410e+07
(557 days 12:04:00, 1083 days 14:05:00]	164913.601	1.640410e+07	1.640410e+07

In [30]: # calculating percentage cumilative revenue with respect to total revenue for all the cu
vin["perc\_tot\_rev"] = vin["CumSum\_Rev"]/vin["Tot\_rev\_Across\_Deciles"]
vin.head(10)

Out[30]:		Tot_rev	CumSum_Rev	Tot_rev_Across_Deciles	perc_tot_rev
	Recency_Deciles				
	(21 days 06:40:59.999999999, 32 days 12:04:36]	5711857.734	5.711858e+06	1.640410e+07	0.348197
	(32 days 12:04:36, 43 days 13:59:12]	2968252.921	8.680111e+06	1.640410e+07	0.529143
	(43 days 13:59:12, 61 days 10:34:24.000000001]	2275605.871	1.095572e+07	1.640410e+07	0.667865
	(61 days 10:34:24.00000001, 87 days 18:32:36.000000017]	1533952.931	1.248967e+07	1.640410e+07	0.761375
	(87 days 18:32:36.00000017, 135 days 08:26:00]	1262303.344	1.375197e+07	1.640410e+07	0.838325
	(135 days 08:26:00, 216 days 09:30:36.000000004]	965468.294	1.471744e+07	1.640410e+07	0.897181
	(216 days 09:30:36.000000004, 345 days 09:39:30]	586407.400	1.530385e+07	1.640410e+07	0.932928
	(345 days 09:39:30, 436 days 12:19:24]	536943.362	1.584079e+07	1.640410e+07	0.965661
	(436 days 12:19:24, 557 days 12:04:00]	398391.840	1.623918e+07	1.640410e+07	0.989947
	(557 days 12:04:00, 1083 days 14:05:00]	164913.601	1.640410e+07	1.640410e+07	1.000000
	` , , . ,				
Out[30]:		Tot_rev	CumSum_Rev	Tot_rev_Across_Deciles	perc_tot_rev
Out[30]:	Recency_Deciles	Tot_rev	CumSum_Rev	Tot_rev_Across_Deciles	
Out[30]:		<b>Tot_rev</b> 5711857.734	<b>CumSum_Rev</b> 5.711858e+06	Tot_rev_Across_Deciles  1.640410e+07	
Out[30]:	Recency_Deciles				perc_tot_rev
Out[30]:	Recency_Deciles (21 days 06:40:59.999999999, 32 days 12:04:36]	5711857.734	5.711858e+06	1.640410e+07	<b>perc_tot_rev</b> 0.348197
Out[30]:	Recency_Deciles (21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12]	5711857.734 2968252.921	5.711858e+06 8.680111e+06	1.640410e+07 1.640410e+07	0.348197 0.529143
Out[30]:	Recency_Deciles  (21 days 06:40:59.9999999999, 32 days 12:04:36]  (32 days 12:04:36, 43 days 13:59:12]  (43 days 13:59:12, 61 days 10:34:24.000000001]  (61 days 10:34:24.000000001, 87 days	5711857.734 2968252.921 2275605.871	5.711858e+06 8.680111e+06 1.095572e+07	1.640410e+07 1.640410e+07 1.640410e+07	0.348197 0.529143 0.667865
Out[30]:	Recency_Deciles  (21 days 06:40:59.99999999999999999999999999999999999	5711857.734 2968252.921 2275605.871 1533952.931	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07	0.348197 0.529143 0.667865 0.761375
Out[30]:	Recency_Deciles  (21 days 06:40:59.99999999999999999999999999999999999	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07	0.348197 0.529143 0.667865 0.761375 0.838325

398391.840 1.623918e+07

1.640410e+07

0.989947

(436 days 12:19:24, 557 days 12:04:00]

```
In [31]:
           #creating 10 different buckets wr.t frequency column
           RFM['Feq Deciles'] = pd.qcut(RFM['Freq'],10)
           RFM.head()
                                                             duration
                                                                                                                Feq_Deciles
Out[31]:
              Customer_ID
                            Freq
                                  Tot_Rev
                                             Avg_Rev
                                                                                             Recency_Deciles
                                                              21 days
                                                                           (21 days 06:40:59.99999999, 32 days
           0
                     18287
                             156
                                   4177.89
                                            26.781346
                                                                                                               (114.0, 180.0]
                                                              13:37:00
                                                                                                    12:04:36]
                                                             498 days
           1
                     18286
                              70
                                   1188.43
                                           16.977571
                                                                          (436 days 12:19:24, 557 days 12:04:00]
                                                                                                                 (53.0, 76.0]
                                                              12:03:00
                                                             682 days
           2
                     18285
                              12
                                    427.00
                                            35.583333
                                                                         (557 days 12:04:00, 1083 days 14:05:00]
                                                                                                                  (8.0, 16.0]
                                                              13:36:00
                                                             569 days
           3
                     18284
                              29
                                    436.68
                                           15.057931
                                                                         (557 days 12:04:00, 1083 days 14:05:00]
                                                                                                                 (25.0, 37.0]
                                                              11:29:00
                                                              31 days
                                                                           (21 days 06:40:59.99999999, 32 days
                                                                                                                     (319.7,
           4
                     18283
                             936
                                   2528.65
                                             2.701549
                                                              11:01:00
                                                                                                    12:04:36]
                                                                                                                   12780.0]
Out[31]:
              Customer_ID
                            Freq
                                   Tot_Rev
                                             Avg_Rev
                                                             duration
                                                                                             Recency_Deciles
                                                                                                                Feq_Deciles
                                                              21 days
                                                                           (21 days 06:40:59.99999999, 32 days
           0
                     18287
                             156
                                   4177.89
                                            26.781346
                                                                                                               (114.0, 180.0]
                                                              13:37:00
                                                                                                    12:04:36]
                                                             498 days
           1
                     18286
                              70
                                   1188.43
                                           16.977571
                                                                          (436 days 12:19:24, 557 days 12:04:00]
                                                                                                                 (53.0, 76.0]
                                                              12:03:00
                                                             682 days
           2
                     18285
                                    427.00
                                           35.583333
                              12
                                                                         (557 days 12:04:00, 1083 days 14:05:00]
                                                                                                                  (8.0, 16.0]
                                                              13:36:00
                                                             569 days
           3
                     18284
                              29
                                    436.68 15.057931
                                                                         (557 days 12:04:00, 1083 days 14:05:00]
                                                                                                                 (25.0, 37.0]
                                                              11:29:00
                                                              31 days
                                                                           (21 days 06:40:59.99999999, 32 days
                                                                                                                     (319.7,
           4
                     18283
                             936
                                   2528.65
                                             2.701549
                                                              11:01:00
                                                                                                    12:04:36]
                                                                                                                   12780.0]
           # create empty dataframe
In [32]:
           rec = pd.DataFrame()
           print(rec)
           Empty DataFrame
           Columns: []
           Index: []
           Empty DataFrame
           Columns: []
           Index: []
In [33]:
           # calculating total revenue for each bucket
           rec["Tot rev"] = RFM.groupby('Feq Deciles').Tot Rev.sum()
           rec.head(10)
Out[33]:
                                 Tot_rev
               Feq_Deciles
                (0.999, 8.0]
                               72193.160
```

164913.601

1.640410e+07

1.640410e+07

1.000000

(557 days 12:04:00, 1083 days 14:05:00]

(8.0, 16.0]

(16.0, 25.0]

(25.0, 37.0]

242098.130

274574.731

383583.741

```
(37.0, 53.0]
                            603851.613
               (53.0, 76.0]
                            773353.640
              (76.0, 114.0) 1070299.192
             (114.0, 180.0) 1630626.023
             (180.0, 319.7] 2761936.172
           (319.7, 12780.0) 8591580.896
Out[33]:
                                Tot_rev
               Feq_Deciles
               (0.999, 8.0]
                              72193.160
               (8.0, 16.0]
                             242098.130
               (16.0, 25.0]
                             274574.731
               (25.0, 37.0]
                             383583.741
               (37.0, 53.0]
                             603851.613
               (53.0, 76.0]
                            773353.640
              (76.0, 114.0] 1070299.192
             (114.0, 180.0) 1630626.023
             (180.0, 319.7] 2761936.172
           (319.7, 12780.0) 8591580.896
In [34]:
           # calculating cumulative revenue
           rec["CumSum_Rev"] = rec["Tot_rev"].cumsum()
           rec.head(10)
Out[34]:
                                Tot_rev CumSum_Rev
               Feq_Deciles
               (0.999, 8.0]
                             72193.160 7.219316e+04
                             242098.130 3.142913e+05
                (8.0, 16.0]
                             274574.731 5.888660e+05
               (16.0, 25.0]
               (25.0, 37.0]
                             383583.741 9.724498e+05
               (37.0, 53.0]
                             603851.613 1.576301e+06
               (53.0, 76.0]
                            773353.640 2.349655e+06
              (76.0, 114.0] 1070299.192 3.419954e+06
             (114.0, 180.0] 1630626.023 5.050580e+06
             (180.0, 319.7] 2761936.172 7.812516e+06
           (319.7, 12780.0] 8591580.896 1.640410e+07
                                Tot_rev CumSum_Rev
Out[34]:
               Feq_Deciles
```

(0.999, 8.0]

(8.0, 16.0]

72193.160 7.219316e+04

242098.130 3.142913e+05

(16.0, 25.0]	274574.731	5.888660e+05
(25.0, 37.0]	383583.741	9.724498e+05
(37.0, 53.0]	603851.613	1.576301e+06
(53.0, 76.0]	773353.640	2.349655e+06
(76.0, 114.0]	1070299.192	3.419954e+06
(114.0, 180.0]	1630626.023	5.050580e+06
(180.0, 319.7]	2761936.172	7.812516e+06
(319.7, 12780.0]	8591580.896	1.640410e+07

In [35]: # calculating total revenue for each bucket
 rec["Tot\_rev\_Across\_Deciles"] = RFM.Tot\_Rev.sum()
 rec.head(10)

Out[35]: Tot\_rev CumSum\_Rev Tot\_rev\_Across\_Deciles

Feq_Deciles			
(0.999, 8.0]	72193.160	7.219316e+04	1.640410e+07
(8.0, 16.0]	242098.130	3.142913e+05	1.640410e+07
(16.0, 25.0]	274574.731	5.888660e+05	1.640410e+07
(25.0, 37.0]	383583.741	9.724498e+05	1.640410e+07
(37.0, 53.0]	603851.613	1.576301e+06	1.640410e+07
(53.0, 76.0]	773353.640	2.349655e+06	1.640410e+07
(76.0, 114.0]	1070299.192	3.419954e+06	1.640410e+07
(114.0, 180.0]	1630626.023	5.050580e+06	1.640410e+07
(180.0, 319.7]	2761936.172	7.812516e+06	1.640410e+07
(319.7, 12780.0]	8591580.896	1.640410e+07	1.640410e+07

## Out[35]: Tot\_rev CumSum\_Rev Tot\_rev\_Across\_Deciles

Feq_Deciles			
(0.999, 8.0]	72193.160	7.219316e+04	1.640410e+07
(8.0, 16.0]	242098.130	3.142913e+05	1.640410e+07
(16.0, 25.0]	274574.731	5.888660e+05	1.640410e+07
(25.0, 37.0]	383583.741	9.724498e+05	1.640410e+07
(37.0, 53.0]	603851.613	1.576301e+06	1.640410e+07
(53.0, 76.0]	773353.640	2.349655e+06	1.640410e+07
(76.0, 114.0]	1070299.192	3.419954e+06	1.640410e+07
(114.0, 180.0]	1630626.023	5.050580e+06	1.640410e+07
(180.0, 319.7]	2761936.172	7.812516e+06	1.640410e+07
(319.7, 12780.0]	8591580.896	1.640410e+07	1.640410e+07

rec["perc\_tot\_rev"] = rec["CumSum\_Rev"]/rec["Tot\_rev\_Across\_Deciles"]
vin.head(10)

Out[36]:		Tot_rev	CumSum_Rev	Tot_rev_Across_Deciles	perc_tot_rev
	Recency_Deciles				
	(21 days 06:40:59.999999999, 32 days 12:04:36]	5711857.734	5.711858e+06	1.640410e+07	0.348197
	(32 days 12:04:36, 43 days 13:59:12]	2968252.921	8.680111e+06	1.640410e+07	0.529143
	(43 days 13:59:12, 61 days 10:34:24.000000001]	2275605.871	1.095572e+07	1.640410e+07	0.667865
	(61 days 10:34:24.000000001, 87 days 18:32:36.000000017]	1533952.931	1.248967e+07	1.640410e+07	0.761375
	(87 days 18:32:36.000000017, 135 days 08:26:00]	1262303.344	1.375197e+07	1.640410e+07	0.838325
	(135 days 08:26:00, 216 days 09:30:36.000000004]	965468.294	1.471744e+07	1.640410e+07	0.897181
	(216 days 09:30:36.000000004, 345 days 09:39:30]	586407.400	1.530385e+07	1.640410e+07	0.932928
	(345 days 09:39:30, 436 days 12:19:24]	536943.362	1.584079e+07	1.640410e+07	0.965661
	(436 days 12:19:24, 557 days 12:04:00]	398391.840	1.623918e+07	1.640410e+07	0.989947
	(557 days 12:04:00, 1083 days 14:05:00]	164913.601	1.640410e+07	1.640410e+07	1.000000
Out[36]:		Tot_rev	CumSum_Rev	Tot_rev_Across_Deciles	perc_tot_rev
Out[36]:	Recency_Deciles	Tot_rev	CumSum_Rev	Tot_rev_Across_Deciles	perc_tot_rev
Out[36]:	Recency_Deciles (21 days 06:40:59.999999999, 32 days 12:04:36]	<b>Tot_rev</b> 5711857.734	<b>CumSum_Rev</b> 5.711858e+06	Tot_rev_Across_Deciles  1.640410e+07	<b>perc_tot_rev</b> 0.348197
Out[36]:					
Out[36]:	(21 days 06:40:59.999999999, 32 days 12:04:36]	5711857.734	5.711858e+06	1.640410e+07	0.348197
Out[36]:	(21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12]	5711857.734 2968252.921	5.711858e+06 8.680111e+06	1.640410e+07 1.640410e+07	0.348197
Out[36]:	(21 days 06:40:59.999999999, 32 days 12:04:36] (32 days 12:04:36, 43 days 13:59:12] (43 days 13:59:12, 61 days 10:34:24.000000001] (61 days 10:34:24.000000001, 87 days	5711857.734 2968252.921 2275605.871	5.711858e+06 8.680111e+06 1.095572e+07	1.640410e+07 1.640410e+07 1.640410e+07	0.348197 0.529143 0.667865
Out[36]:	(21 days 06:40:59.99999999999999999999999999999999999	5711857.734 2968252.921 2275605.871 1533952.931	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07	0.348197 0.529143 0.667865 0.761375
Out[36]:	(21 days 06:40:59.99999999999999999999999999999999999	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07	0.348197 0.529143 0.667865 0.761375 0.838325
Out[36]:	(21 days 06:40:59.99999999999999999999999999999999999	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344 965468.294	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07 1.471744e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07	0.348197 0.529143 0.667865 0.761375 0.838325 0.897181
Out[36]:	(21 days 06:40:59.99999999999999999999999999999999999	5711857.734 2968252.921 2275605.871 1533952.931 1262303.344 965468.294 586407.400	5.711858e+06 8.680111e+06 1.095572e+07 1.248967e+07 1.375197e+07 1.471744e+07 1.530385e+07	1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07 1.640410e+07	0.348197 0.529143 0.667865 0.761375 0.838325 0.897181 0.932928

Always open for feedback and suggestions. If it helps Thumbs Up !!!