

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
df = pd.read_excel("C:\\Users\\Sunita\\Desktop\\sales_data.xlsx")
```

In [3]:

```
df.shape
```

Out[3]:

(5000, 40)

In [4]:

```
df.head(15)
```

Out[4]:

	CustomerID	TOTAL_ORDERS	REVENUE	AVERAGE_ORDER_VALUE	CARRIAGE_REVENUE	AVERAGESHIPPING	FIRST_OR
0	22	124	11986.54	96.67	529.59	4.27	
1	29	82	11025.96	134.46	97.92	1.19	
2	83	43	7259.69	168.83	171.69	3.99	
3	95	44	6992.27	158.92	92.82	2.11	
4	124	55	6263.44	113.88	179.04	3.26	
5	153	49	5841.24	119.21	96.84	1.98	
6	187	43	5470.27	127.22	128.77	2.99	
7	219	54	5200.53	96.31	237.53	4.40	
8	258	19	4967.06	261.42	51.91	2.73	
9	308	21	4726.38	225.07	63.88	3.04	
10	491	37	3934.05	106.33	162.50	4.39	
11	492	20	3933.82	196.69	74.87	3.74	
12	572	21	3778.07	179.91	91.07	4.34	
13	595	34	3725.48	109.57	115.78	3.41	
14	613	31	3660.12	118.07	114.77	3.70	

15 rows x 40 columns



In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 40 columns):
#   Column                               Non-Null Count  Dtype
---  -
0   CustomerID                           5000 non-null   int64
1   TOTAL_ORDERS                         5000 non-null   int64
2   REVENUE                              5000 non-null   float64
3   AVERAGE_ORDER_VALUE                 5000 non-null   float64
4   CARRIAGE_REVENUE                     5000 non-null   float64
5   AVERAGESHIPPING                      5000 non-null   float64
```

6	FIRST_ORDER_DATE	5000	non-null	datetime64[ns]
7	LATEST_ORDER_DATE	5000	non-null	datetime64[ns]
8	AVGDAYS BETWEEN ORDERS	5000	non-null	float64
9	DAYSSINCE LAST ORDER	5000	non-null	int64
10	MONDAY_ORDERS	5000	non-null	int64
11	TUESDAY_ORDERS	5000	non-null	int64
12	WEDNESDAY_ORDERS	5000	non-null	int64
13	THURSDAY_ORDERS	5000	non-null	int64
14	FRIDAY_ORDERS	5000	non-null	int64
15	SATURDAY_ORDERS	5000	non-null	int64
16	SUNDAY_ORDERS	5000	non-null	int64
17	MONDAY_REVENUE	5000	non-null	float64
18	TUESDAY_REVENUE	5000	non-null	float64
19	WEDNESDAY_REVENUE	5000	non-null	float64
20	THURSDAY_REVENUE	5000	non-null	float64
21	FRIDAY_REVENUE	5000	non-null	float64
22	SATURDAY_REVENUE	5000	non-null	float64
23	SUNDAY_REVENUE	5000	non-null	float64
24	WEEK1_DAY01_DAY07_ORDERS	5000	non-null	int64
25	WEEK2_DAY08_DAY15_ORDERS	5000	non-null	int64
26	WEEK3_DAY16_DAY23_ORDERS	5000	non-null	int64
27	WEEK4_DAY24_DAY31_ORDERS	5000	non-null	int64
28	WEEK1_DAY01_DAY07_REVENUE	5000	non-null	float64
29	WEEK2_DAY08_DAY15_REVENUE	5000	non-null	float64
30	WEEK3_DAY16_DAY23_REVENUE	5000	non-null	float64
31	WEEK4_DAY24_DAY31_REVENUE	5000	non-null	float64
32	TIME_0000_0600_ORDERS	5000	non-null	int64
33	TIME_0601_1200_ORDERS	5000	non-null	int64
34	TIME_1200_1800_ORDERS	5000	non-null	int64
35	TIME_1801_2359_ORDERS	5000	non-null	int64
36	TIME_0000_0600_REVENUE	5000	non-null	float64
37	TIME_0601_1200_REVENUE	5000	non-null	float64
38	TIME_1200_1800_REVENUE	5000	non-null	float64
39	TIME_1801_2359_REVENUE	5000	non-null	float64

dtypes: datetime64[ns](2), float64(20), int64(18)

memory usage: 1.5 MB

In [6]:

```
df.columns
```

Out[6]:

```
Index(['CustomerID', 'TOTAL_ORDERS', 'REVENUE', 'AVERAGE_ORDER_VALUE',
      'CARRIAGE_REVENUE', 'AVERAGESHIPPING', 'FIRST_ORDER_DATE',
      'LATEST_ORDER_DATE', 'AVGDAYS BETWEEN ORDERS', 'DAYSSINCE LAST ORDER',
      'MONDAY_ORDERS', 'TUESDAY_ORDERS', 'WEDNESDAY_ORDERS',
      'THURSDAY_ORDERS', 'FRIDAY_ORDERS', 'SATURDAY_ORDERS', 'SUNDAY_ORDERS',
      'MONDAY_REVENUE', 'TUESDAY_REVENUE', 'WEDNESDAY_REVENUE',
      'THURSDAY_REVENUE', 'FRIDAY_REVENUE', 'SATURDAY_REVENUE',
      'SUNDAY_REVENUE', 'WEEK1_DAY01_DAY07_ORDERS',
      'WEEK2_DAY08_DAY15_ORDERS', 'WEEK3_DAY16_DAY23_ORDERS',
      'WEEK4_DAY24_DAY31_ORDERS', 'WEEK1_DAY01_DAY07_REVENUE',
      'WEEK2_DAY08_DAY15_REVENUE', 'WEEK3_DAY16_DAY23_REVENUE',
      'WEEK4_DAY24_DAY31_REVENUE', 'TIME_0000_0600_ORDERS',
      'TIME_0601_1200_ORDERS', 'TIME_1200_1800_ORDERS',
      'TIME_1801_2359_ORDERS', 'TIME_0000_0600_REVENUE',
      'TIME_0601_1200_REVENUE', 'TIME_1200_1800_REVENUE',
      'TIME_1801_2359_REVENUE'],
      dtype='object')
```

In [7]:

```
df.isnull().sum()
```

Out[7]:

CustomerID	0
TOTAL_ORDERS	0
REVENUE	0
AVERAGE_ORDER_VALUE	0
CARRIAGE_REVENUE	0
AVERAGESHIPPING	0

```

FIRST_ORDER_DATE      0
LATEST_ORDER_DATE     0
AVGDAYSBETWEENORDERS  0
DAYSSINCELASTORDER    0
MONDAY_ORDERS          0
TUESDAY_ORDERS         0
WEDNESDAY_ORDERS       0
THURSDAY_ORDERS        0
FRIDAY_ORDERS          0
SATURDAY_ORDERS        0
SUNDAY_ORDERS          0
MONDAY_REVENUE         0
TUESDAY_REVENUE        0
WEDNESDAY_REVENUE      0
THURSDAY_REVENUE       0
FRIDAY_REVENUE         0
SATURDAY_REVENUE       0
SUNDAY_REVENUE         0
WEEK1_DAY01_DAY07_ORDERS 0
WEEK2_DAY08_DAY15_ORDERS 0
WEEK3_DAY16_DAY23_ORDERS 0
WEEK4_DAY24_DAY31_ORDERS 0
WEEK1_DAY01_DAY07_REVENUE 0
WEEK2_DAY08_DAY15_REVENUE 0
WEEK3_DAY16_DAY23_REVENUE 0
WEEK4_DAY24_DAY31_REVENUE 0
TIME_0000_0600_ORDERS  0
TIME_0601_1200_ORDERS  0
TIME_1200_1800_ORDERS  0
TIME_1801_2359_ORDERS  0
TIME_0000_0600_REVENUE 0
TIME_0601_1200_REVENUE 0
TIME_1200_1800_REVENUE 0
TIME_1801_2359_REVENUE 0
dtype: int64

```

In [8]:

```

plt.boxplot([df['MONDAY_ORDERS'],df['TUESDAY_ORDERS'],df['WEDNESDAY_ORDERS'],df['THURSDAY_ORDERS'],df['FRIDAY_ORDERS'],df['SATURDAY_ORDERS'],df['SUNDAY_ORDERS']])

```

Out[8]:

```

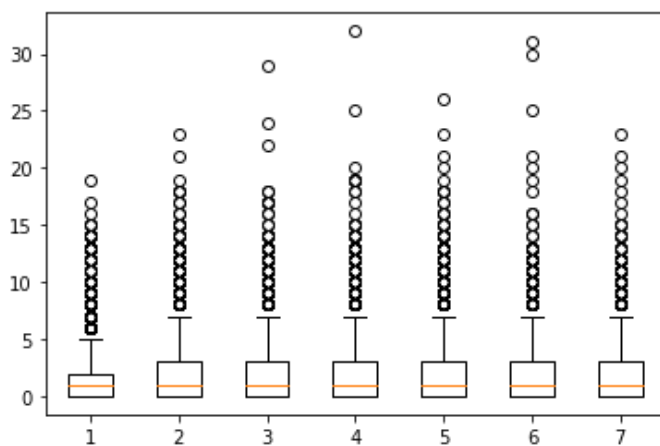
{'whiskers': [<matplotlib.lines.Line2D at 0x1fb221c0fa0>,
 <matplotlib.lines.Line2D at 0x1fb22b91820>,
 <matplotlib.lines.Line2D at 0x1fb22d8cbb0>,
 <matplotlib.lines.Line2D at 0x1fb22204bb0>,
 <matplotlib.lines.Line2D at 0x1fb224bef40>,
 <matplotlib.lines.Line2D at 0x1fb22a9c2e0>,
 <matplotlib.lines.Line2D at 0x1fb229bb760>,
 <matplotlib.lines.Line2D at 0x1fb229bbac0>,
 <matplotlib.lines.Line2D at 0x1fb22606f40>,
 <matplotlib.lines.Line2D at 0x1fb226752e0>,
 <matplotlib.lines.Line2D at 0x1fb224f0760>,
 <matplotlib.lines.Line2D at 0x1fb224f0ac0>,
 <matplotlib.lines.Line2D at 0x1fb2273bf40>,
 <matplotlib.lines.Line2D at 0x1fb228812e0>],
 'caps': [<matplotlib.lines.Line2D at 0x1fb22b91b80>,
 <matplotlib.lines.Line2D at 0x1fb22b91ee0>,
 <matplotlib.lines.Line2D at 0x1fb22204970>,
 <matplotlib.lines.Line2D at 0x1fb224be1c0>,
 <matplotlib.lines.Line2D at 0x1fb22a9c640>,
 <matplotlib.lines.Line2D at 0x1fb22a9c9a0>,
 <matplotlib.lines.Line2D at 0x1fb229bbe20>,
 <matplotlib.lines.Line2D at 0x1fb226061c0>,
 <matplotlib.lines.Line2D at 0x1fb22675640>,
 <matplotlib.lines.Line2D at 0x1fb226759a0>,
 <matplotlib.lines.Line2D at 0x1fb224f0e20>,
 <matplotlib.lines.Line2D at 0x1fb2273b1c0>,
 <matplotlib.lines.Line2D at 0x1fb22881640>,
 <matplotlib.lines.Line2D at 0x1fb228819a0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1fb22e2ad90>,

```

```

<matplotlib.lines.Line2D at 0x1fb22d8c850>,
<matplotlib.lines.Line2D at 0x1fb224bebe0>,
<matplotlib.lines.Line2D at 0x1fb229bb400>,
<matplotlib.lines.Line2D at 0x1fb22606be0>,
<matplotlib.lines.Line2D at 0x1fb224f0400>,
<matplotlib.lines.Line2D at 0x1fb2273bbe0>],
'medians': [<matplotlib.lines.Line2D at 0x1fb22d8c160>,
<matplotlib.lines.Line2D at 0x1fb224be520>,
<matplotlib.lines.Line2D at 0x1fb22a9cd00>,
<matplotlib.lines.Line2D at 0x1fb22606520>,
<matplotlib.lines.Line2D at 0x1fb22675d00>,
<matplotlib.lines.Line2D at 0x1fb2273b520>,
<matplotlib.lines.Line2D at 0x1fb22881d00>],
'fliers': [<matplotlib.lines.Line2D at 0x1fb22d8c4c0>,
<matplotlib.lines.Line2D at 0x1fb224be880>,
<matplotlib.lines.Line2D at 0x1fb229bb0a0>,
<matplotlib.lines.Line2D at 0x1fb22606880>,
<matplotlib.lines.Line2D at 0x1fb224f00a0>,
<matplotlib.lines.Line2D at 0x1fb2273b880>,
<matplotlib.lines.Line2D at 0x1fb227e20a0>],
'means': []}

```



In [9]:

```

print('Monday :',df['MONDAY_ORDERS'].mean(),'Tuesday :',df['TUESDAY_ORDERS'].mean(),'Wedn
esday :',df['WEDNESDAY_ORDERS'].mean(),'Thursday :',df['THURSDAY_ORDERS'].mean(),'Friday
:',df['FRIDAY_ORDERS'].mean(),'Saturday :',df['SATURDAY_ORDERS'].mean(),'Sunday :',df['SU
NDAY_ORDERS'].mean())

```

Monday : 1.629 Tuesday : 1.7544 Wednesday : 1.798 Thursday : 2.134 Friday : 1.9462 Saturd  
ay : 1.6834 Sunday : 1.9254

In [10]:

```

Average_Days_Order = [df['MONDAY_ORDERS'].mean(),df['TUESDAY_ORDERS'].mean(),df['WEDNESDA
Y_ORDERS'].mean(),df['THURSDAY_ORDERS'].mean(),df['FRIDAY_ORDERS'].mean(),df['SATURDAY_OR
DERS'].mean(),df['SUNDAY_ORDERS'].mean()]

```

In [11]:

```
Average_Days_Order
```

Out[11]:

```
[1.629, 1.7544, 1.798, 2.134, 1.9462, 1.6834, 1.9254]
```

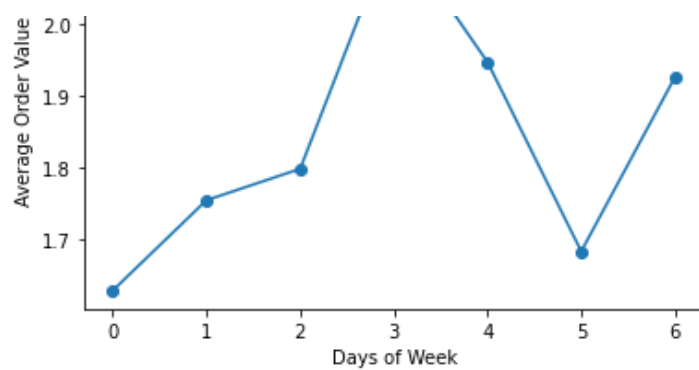
In [12]:

```

plt.plot(Average_Days_Order , marker='o')
plt.xlabel('Days of Week')
plt.ylabel('Average Order Value')
plt.show()

```





## CALCULATING RECENCY

In [13]:

```
import datetime
cdate = datetime.datetime.now()
df['LATEST_ORDER_DATE'] = pd.to_datetime(df['LATEST_ORDER_DATE'])

def get_recency(x):
    #print(cdate-x.max())
    return (cdate-x.max()).days

df_rec = df.groupby(['CustomerID'])['LATEST_ORDER_DATE'].apply(get_recency)
df_rec.name="Recency"
```

## CALCULATING FREQUENCY

In [14]:

```
df_freq = df.groupby("CustomerID")['CustomerID'].apply(len)
df_freq.name = "Frequency"
```

In [15]:

```
df['Monetary'] = df.TOTAL_ORDERS * df.AVERAGE_ORDER_VALUE
df_mon = df.groupby('CustomerID')['Monetary'].apply(sum)
```

In [16]:

```
df2 = pd.merge(left=df_rec.reset_index(),right = df_freq.reset_index(),on="CustomerID")
df2 = pd.merge(left = df2, right=df_mon.reset_index(),on="CustomerID")
df2.head(25)
```

Out[16]:

	CustomerID	Recency	Frequency	Monetary
0	1	297	1	34847.47
1	2	338	1	32487.17
2	3	297	1	24179.13
3	4	249	1	18554.76
4	5	374	1	16884.92
5	6	276	1	16694.04
6	7	360	1	15999.93
7	8	306	1	15840.30
8	9	356	1	14526.75
9	10	257	1	14310.09
10	11	289	1	14153.58

11	12	260	1	13776.02
CustomerID	Recency	Frequency	Monetary	
12	13	356	1	13653.76
13	14	399	1	13355.10
14	15	258	1	13181.00
15	16	358	1	13155.84
16	17	250	1	12970.08
17	18	337	1	12940.00
18	19	279	1	12898.94
19	20	257	1	12237.60
20	21	291	1	12230.88
21	22	245	1	11987.08
22	23	259	1	11829.12
23	24	391	1	11661.90
24	25	362	1	11402.73

In [17]:

```
df2.Recency.unique()
```

Out[17]:

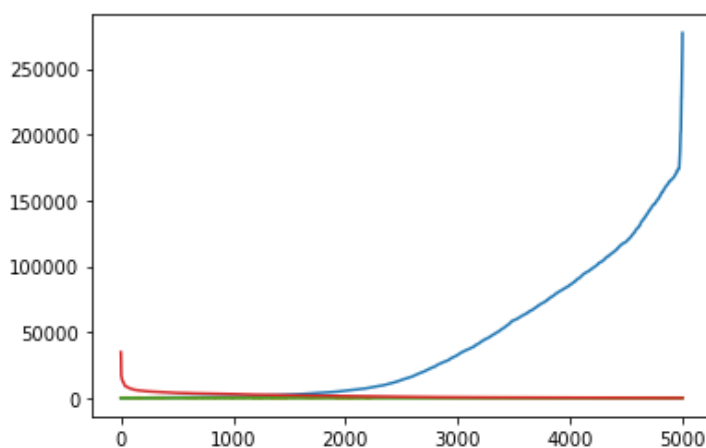
```
array([297, 338, 249, 374, 276, 360, 306, 356, 257, 289, 260, 399, 258,
       358, 250, 337, 279, 291, 245, 259, 391, 362, 302, 430, 342, 318,
       383, 331, 261, 375, 251, 355, 246, 270, 434, 339, 295, 317, 373,
       320, 271, 396, 372, 322, 345, 283, 254, 285, 411, 340, 274, 408,
       380, 308, 323, 248, 387, 247, 353, 351, 304, 448, 269, 301, 344,
       278, 368, 275, 352, 384, 293, 449, 273, 327, 370, 266, 290, 267,
       436, 309, 262, 315, 335, 307, 359, 426, 256, 310, 324, 280, 281,
       294, 429, 319, 347, 336, 365, 305, 364, 377, 330, 366, 367, 284,
       343, 263, 255, 394, 311, 350, 252, 392, 422, 328, 400, 357, 265,
       420, 312, 253, 376, 334, 313, 292, 333, 348, 286, 349, 385, 300,
       321, 314, 379, 298, 287, 316, 303, 288, 413, 404, 268, 272, 329,
       441, 390, 299, 442, 421, 401, 354, 419, 402, 389, 325, 447, 282,
       341, 445, 398, 346, 439, 409, 296, 264, 361, 414, 427, 418, 406,
       424, 277, 440, 326, 435, 437, 425, 415, 393, 369, 407, 381, 410,
       423, 417, 363, 332, 432, 388, 444, 405, 412, 416, 371, 446, 382,
       450, 386, 397, 438, 451, 443, 431, 433, 378, 395, 428, 403],
      dtype=int64)
```

In [18]:

```
plt.plot(df2)
```

Out[18]:

```
[<matplotlib.lines.Line2D at 0x1fb22da7970>,
 <matplotlib.lines.Line2D at 0x1fb22c90eb0>,
 <matplotlib.lines.Line2D at 0x1fb22c90fa0>,
 <matplotlib.lines.Line2D at 0x1fb22c9d0a0>]
```



In [19]:

```
df2.Frequency.unique()
```

Out[19]:

```
array([1], dtype=int64)
```

In [20]:

```
df2['r_quartile'] = pd.qcut(df2.Recency,[0,0.25,0.50,0.75,1.0],labels = [1,2,3,4])
#df2['F'] = pd.qcut(df2.Frequency,[1.0],labels = [4,3,2,1],duplicates='drop')
df2['m_quartile'] = pd.qcut(df2.Monetary,[0,0.25,0.50,0.75,1.0],labels = [4,3,2,1])
df2
```

Out[20]:

CustomerID	Recency	Frequency	Monetary	r_quartile	m_quartile	
0	1	297	1	34847.47	2	1
1	2	338	1	32487.17	3	1
2	3	297	1	24179.13	2	1
3	4	249	1	18554.76	1	1
4	5	374	1	16884.92	3	1
...	...	...	...	...	...	...
4995	248709	298	1	59.96	2	4
4996	248710	363	1	59.94	3	4
4997	259042	372	1	51.94	3	4
4998	266024	376	1	45.94	3	4
4999	277160	301	1	38.49	2	4

5000 rows x 6 columns

In [21]:

```
df2.CustomerID[(df2.r_quartile ==1) & (df2.m_quartile==1)].to_csv("best_customers.csv")
df2.CustomerID[(df2.r_quartile ==3) & (df2.m_quartile==1)].to_csv("almost_lost_customers.csv")
df2.CustomerID[(df2.r_quartile ==4) & (df2.m_quartile==1)].to_csv("lost_customers.csv")
```

In [22]:

```
df2['RFM_Score']=df2.r_quartile.astype(str)+df2.m_quartile.astype(str)
df2
```

Out[22]:

CustomerID	Recency	Frequency	Monetary	r_quartile	m_quartile	RFM_Score	
0	1	297	1	34847.47	2	1	21
1	2	338	1	32487.17	3	1	31
2	3	297	1	24179.13	2	1	21
3	4	249	1	18554.76	1	1	11
4	5	374	1	16884.92	3	1	31
...	...	...	...	...	...	...	...
4995	248709	298	1	59.96	2	4	24
4996	248710	363	1	59.94	3	4	34
4997	259042	372	1	51.94	3	4	34
4998	266024	376	1	45.94	3	4	34

```
4999 CustomerID Recency Frequency Monetary r_quartile m_quartile RFM_Score
```

5000 rows x 7 columns

In [23]:

```
df2.groupby('RFM_Score')['Monetary'].mean()
```

Out[23]:

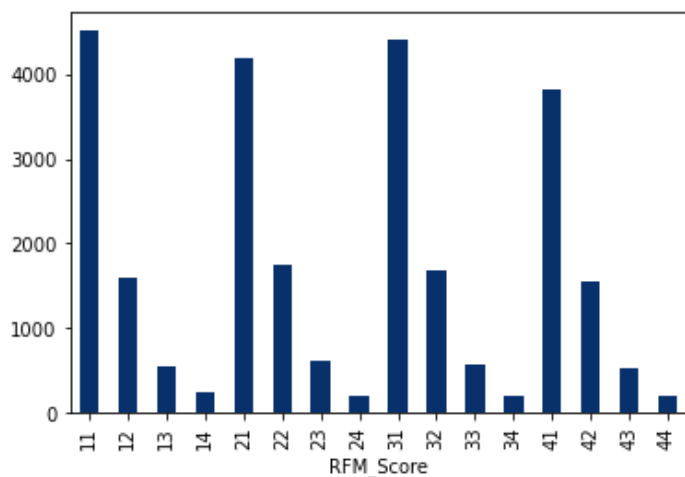
```
RFM_Score
11      4518.969819
12      1588.572226
13       542.984900
14       242.717659
21      4196.159271
22      1755.896911
23       624.468028
24       198.534839
31      4414.418534
32      1675.465973
33       576.519919
34       204.419929
41      3822.378205
42      1553.590270
43       533.282145
44       191.265826
Name: Monetary, dtype: float64
```

In [24]:

```
df2.groupby('RFM_Score')['Monetary'].mean().plot(kind='bar', colormap='Blues_r')
```

Out[24]:

<AxesSubplot:xlabel='RFM\_Score'>



In [25]:

```
df2.groupby('RFM_Score')['Recency'].mean()
```

Out[25]:

```
RFM_Score
11      247.560241
12      246.264151
13      245.191235
14      245.020231
21      276.705085
22      276.073227
23      278.267606
24      280.225806
31      356.466346
32      354.274600
33      363.141700
34      360.635714
```



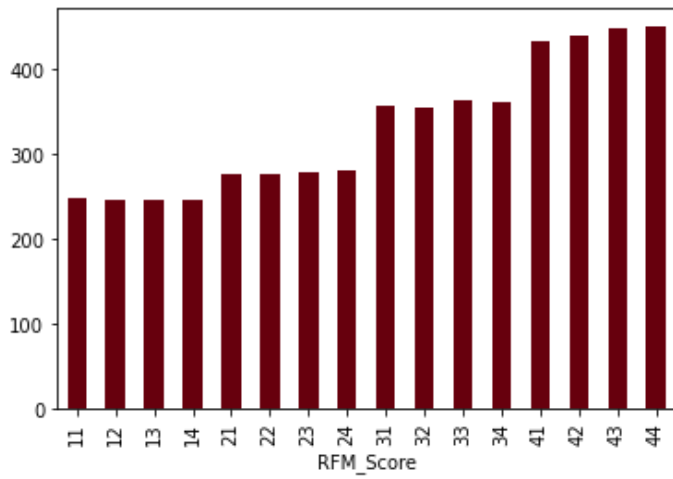
```
41      432.564103
42      438.234234
43      447.019499
44      450.052707
Name: Recency, dtype: float64
```

```
In [26]:
```

```
df2.groupby('RFM_Score')['Recency'].mean().plot(kind='bar', colormap='Reds_r')
```

```
Out[26]:
```

```
<AxesSubplot:xlabel='RFM_Score'>
```



```
In [ ]:
```