

```
In [2]:  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
%matplotlib inline  
import seaborn as sns  
from datetime import datetime, timedelta
```

```
In [3]:  
spend = pd.read_csv(r'C:\Users\Admin\Downloads\spend.csv')  
repayment = pd.read_csv(r'C:\Users\Admin\Downloads\Repayment.csv')  
customer = pd.read_csv(r'C:\Users\Admin\Downloads\Customer Acquisition.csv')  
print(spend)  
print(repayment)  
print(customer)
```

	Sl No:	Customer	Month	Type	Amount
0	1	A1	12-Jan-04	JEWELLERY	485470.80
1	2	A1	3-Jan-04	PETRO	410556.13
2	3	A1	15-Jan-04	CLOTHES	23740.46
3	4	A1	25-Jan-04	FOOD	484342.47
4	5	A1	17-Jan-05	CAMERA	369694.07
...	...	...	...	...	...
1495	1496	A67	4-Feb-06	BUS TICKET	356872.73
1496	1497	A68	25-Mar-06	BUS TICKET	204971.10
1497	1498	A69	31-Mar-06	BUS TICKET	50449.44
1498	1499	A70	23-Mar-06	BUS TICKET	80593.94
1499	1500	A71	24-Mar-06	BUS TICKET	194447.62

[1500 rows x 5 columns]

	SL No:	Customer	Month	Amount	Unnamed: 4
0	NaN	A1	12-Jan-04	495414.75	NaN
1	2.0	A1	3-Jan-04	245899.02	NaN
2	3.0	A1	15-Jan-04	259490.06	NaN
3	4.0	A1	25-Jan-04	437555.12	NaN
4	5.0	A1	17-Jan-05	165972.88	NaN
...	...	...	...	...	...
1518	NaN	Nan	Nan	NaN	NaN
1519	NaN	Nan	Nan	NaN	NaN
1520	NaN	Nan	Nan	NaN	NaN
1521	NaN	Nan	Nan	NaN	NaN
1522	NaN	Nan	Nan	NaN	NaN

[1523 rows x 5 columns]

	No	Customer	Age	City	Product	Limit	Company	Segment
0	1	A1	76	BANGALORE	Gold	500000.0	C1	Self Employed
1	2	A2	71	CALCUTTA	Silver	100000.0	C2	Salaried_MNC
2	3	A3	34	COCHIN	Platinum	10000.0	C3	Salaried_Pvt
3	4	A4	47	BOMBAY	Platinum	10001.0	C4	Govt
4	5	A5	56	BANGALORE	Platinum	10002.0	C5	Normal Salary
...	...	...	...	...	...	...	...	...
95	96	A96	54	CHENNAI	Silver	100000.0	C19	Salaried_Pvt
96	97	A97	58	TRIVANDRUM	Platinum	10000.0	C20	Govt
97	98	A98	51	CALCUTTA	Platinum	10001.0	C21	Normal Salary
98	99	A99	35	CALCUTTA	Platinum	10002.0	C22	Self Employed
99	100	A100	36	COCHIN	Silver	100000.0	C5	Salaried_MNC

[100 rows x 8 columns]

```
In [4]:  
spend.info()
```

<class 'pandas.core.frame.DataFrame'>

```
RangeIndex: 1500 entries, 0 to 1499
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          --          --      
 0   Sl No:      1500 non-null   int64  
 1   Customer    1500 non-null   object  
 2   Month       1500 non-null   object  
 3   Type        1500 non-null   object  
 4   Amount      1500 non-null   float64 
dtypes: float64(1), int64(1), object(3)
memory usage: 41.1+ KB
```

In [6]: `spend.shape`

Out[6]: (1500, 5)

In [8]: `spend.size`

Out[8]: 7500

In [9]: `#drop the column Sl No  
spend.drop('Sl No:',axis=1,inplace=True)`

In [10]: `#Change the type of Month Column from object to datetime  
spend['Month']=spend['Month'].apply(lambda x:pd.to_datetime(x))`

In [11]: `spend.head()`

Out[11]:

	Customer	Month	Type	Amount
0	A1	2004-01-12	JEWELLERY	485470.80
1	A1	2004-01-03	PETRO	410556.13
2	A1	2004-01-15	CLOTHES	23740.46
3	A1	2004-01-25	FOOD	484342.47
4	A1	2005-01-17	CAMERA	369694.07

In [12]: `spend.isnull().sum()`

Out[12]:

Customer	0
Month	0
Type	0
Amount	0
dtype:	int64

In [13]: `repayment.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1523 entries, 0 to 1522
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          --          --      
 0   SL No:      1499 non-null   float64
```

```
1   Customer    1500 non-null  object
2   Month       1500 non-null  object
3   Amount      1500 non-null  float64
4   Unnamed: 4   0 non-null    float64
dtypes: float64(3), object(2)
memory usage: 47.7+ KB
```

In [14]: `repayment.shape`

Out[14]: (1523, 5)

In [15]: `repayment.size`

Out[15]: 7615

In [16]: *#Drop the unnamed columns, SL No from repayment\_data*  
`repayment.drop(['Unnamed: 4', 'SL No'], axis=1, inplace=True)`

In [17]: *#drop the rows which contain missing values*  
`repayment.dropna(axis=0, inplace=True)`

In [18]: *#change the type of month column from object to datetime*  
`repayment['Month']=repayment['Month'].apply(lambda x:pd.to_datetime(x))`

In [19]: `repayment.head()`

Out[19]:

	Customer	Month	Amount
0	A1	2004-01-12	495414.75
1	A1	2004-01-03	245899.02
2	A1	2004-01-15	259490.06
3	A1	2004-01-25	437555.12
4	A1	2005-01-17	165972.88

In [20]: `repayment.isnull().sum()`

Out[20]:

Customer	0
Month	0
Amount	0
dtype:	int64

In [21]: `customer.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype  
 ---  --          --          --    
 0   No          100 non-null   int64  
 1   Customer    100 non-null   object  
 2   Age          100 non-null   int64  
 3   Gender       100 non-null   object  
 4   Education    100 non-null   object  
 5   Marital status 100 non-null  object  
 6   Income       100 non-null   int64  
 7   Loan          100 non-null   int64  
dtypes: int64(4), object(4)
memory usage: 4.8+ KB
```

```

3   City      100 non-null    object
4   Product    100 non-null    object
5   Limit      100 non-null  float64
6   Company    100 non-null    object
7   Segment    100 non-null    object
dtypes: float64(1), int64(2), object(5)
memory usage: 4.4+ KB

```

In [22]: `customer.shape`

Out[22]: (100, 8)

In [23]: `customer.size`

Out[23]: 800

In [24]: `customer.describe()`

	No	Age	Limit
<b>count</b>	100.000000	100.000000	100.000000
<b>mean</b>	50.500000	46.490000	284700.610000
<b>std</b>	29.011492	18.282572	218494.967852
<b>min</b>	1.000000	12.000000	10000.000000
<b>25%</b>	25.750000	33.750000	100000.000000
<b>50%</b>	50.500000	46.500000	300001.500000
<b>75%</b>	75.250000	60.250000	500000.000000
<b>max</b>	100.000000	79.000000	500000.000000

In [25]: `repayment.describe()`

	Amount
<b>count</b>	1500.000000
<b>mean</b>	247890.054420
<b>std</b>	145290.382733
<b>min</b>	55.860000
<b>25%</b>	118604.905000
<b>50%</b>	248008.185000
<b>75%</b>	367236.890000
<b>max</b>	499930.660000

In [26]: `spend.describe()`

	Amount
--	--------

Amount	
<b>count</b>	1500.000000
<b>mean</b>	243096.049460
<b>std</b>	141349.547367
<b>min</b>	720.300000
<b>25%</b>	122606.487500
<b>50%</b>	239210.360000
<b>75%</b>	361617.632500
<b>max</b>	499793.490000

In [31]:

```
#a
customer.loc[(customer.Age<18), 'Age']=customer.Age.mean()
```

In [28]:

```
mean_new = customer["Age"].mean()
print("the new_mean of age coloum is ",mean_new)
```

the new\_mean of age coloum is 48.399399999999986

In [33]:

```
customer.loc[customer["Age"]< 18,"Age"]
```

Out[33]:

Series([], Name: Age, dtype: float64)

In [34]:

```
#b
Customer_Spend=pd.merge(left=customer,right=spend,on='Customer')
```

In [35]:

```
Customer_Spend.head()
```

Out[35]:

No	Customer	Age	City	Product	Limit	Company	Segment	Month	Type
0	1	A1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed	2004-01-12 JEWELLERY 4
1	1	A1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed	2004-01-03 PETRO 4
2	1	A1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed	2004-01-15 CLOTHES
3	1	A1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed	2004-01-25 FOOD 4
4	1	A1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed	2005-01-17 CAMERA 3

◀ ▶

In [39]:

```
Customer_Spend[Customer_Spend["Amount"] > Customer_Spend["Limit"]]
```

Out[39]:

No	Customer	Age	City	Product	Limit	Company	Segment	Month
0	1	A1	BANGALORE	Gold	500000.0	C1	Self Employed	2004-01-12

No	Customer	Age		City	Product	Limit	Company	Segment	Month
18	2	A2	71.0	CALCUTTA	Silver	100000.0	C2	Salaried_MNC	2004-01-03
19	2	A2	71.0	CALCUTTA	Silver	100000.0	C2	Salaried_MNC	2005-02-28
20	2	A2	71.0	CALCUTTA	Silver	100000.0	C2	Salaried_MNC	2004-04-06
21	2	A2	71.0	CALCUTTA	Silver	100000.0	C2	Salaried_MNC	2004-11-23
22	2	A2	71.0	CALCUTTA	Silver	100000.0	C2	Salaried_MNC	2005-09-03 JEV
...	...	...	...	...	...	...	...	...	...
1486	87	A87	27.0	CALCUTTA	Silver	100003.0	C10	Govt	2006-03-30
1496	97	A97	58.0	TRIVANDRUM	Platinum	10000.0	C20	Govt	2004-01-12
1497	98	A98	51.0	CALCUTTA	Platinum	10001.0	C21	Normal Salary	2004-01-03 SH
1498	99	A99	35.0	CALCUTTA	Platinum	10002.0	C22	Self Employed	2004-01-15 AIR
1499	100	A100	36.0	COCHIN	Silver	100000.0	C5	Salaried_MNC	2004-01-25

607 rows × 11 columns



```
In [40]: #In case spend amount is more than the Limit, replace it with 50% of that customer's
Customer_Spend.loc[Customer_Spend["Amount"] > Customer_Spend["Limit"], "Amount"] = (5
```

```
In [41]: # there are no customers left whose spend amount is more than limit
Customer_Spend[Customer_Spend["Amount"] > Customer_Spend["Limit"]]
```

```
Out[41]: No Customer Age City Product Limit Company Segment Month Type Amount
```

```
In [49]: #c
customer_repayment = pd.merge(left = repayment ,right = customer, on = 'Customer', how
```

```
In [50]: customer_repayment[customer_repayment["Amount"] > customer_repayment["Limit"]]
```

No	Customer	Month	Amount	No	Age	City	Product	Limit	Company	Segment
18	A8	2004-02-05	156913.59	8	67.0	PATNA	Silver	100002.0	C8	Salaried_Pvt
19	A8	2004-01-12	350999.98	8	67.0	PATNA	Silver	100002.0	C8	Salaried_Pvt
20	A8	2006-03-31	155642.95	8	67.0	PATNA	Silver	100002.0	C8	Salaried_Pvt

	Customer	Month	Amount	No	Age	City	Product	Limit	Company	Segment
21	A8	2005-05-09	145457.38	8	67.0	PATNA	Silver	100002.0	C8	Salaried_Pvt
22	A8	2006-05-08	361184.89	8	67.0	PATNA	Silver	100002.0	C8	Salaried_Pvt
...	...	...	...	...	...	...	...	...	...	...
1494	A7	2006-05-06	400375.97	7	26.0	COCHIN	Silver	100001.0	C7	Salaried_MNC
1496	A7	2005-12-03	429700.12	7	26.0	COCHIN	Silver	100001.0	C7	Salaried_MNC
1497	A7	2005-02-23	344628.72	7	26.0	COCHIN	Silver	100001.0	C7	Salaried_MNC
1498	A7	2006-03-31	336264.37	7	26.0	COCHIN	Silver	100001.0	C7	Salaried_MNC
1499	A7	2004-01-03	473662.61	7	26.0	COCHIN	Silver	100001.0	C7	Salaried_MNC

600 rows × 10 columns



In [51]:

```
#2 A
Customer_Spend['Customer'].drop_duplicates().count()
```

Out[51]: 100

In [52]:

```
#B
Customer_Spend['Product'].drop_duplicates()
```

Out[52]:

```
0      Gold
18     Silver
29    Platinum
Name: Product, dtype: object
```

In [54]:

```
customer_repayment['Segment'].drop_duplicates()
```

Out[54]:

```
0      Self Employed
18     Salaried_Pvt
36      Govt
54    Normal Salary
121   Salaried_MNC
Name: Segment, dtype: object
```

In [56]:

```
#C
Customer_Spend_Amount=Customer_Spend.groupby(['Customer','Month'])[['Amount']].mean()
```

In [57]:

```
Customer_Spend['Spend_Month']=Customer_Spend['Month'].apply(lambda x:x.month)
```

In [58]:

```
Customer_Spend['Spend_Month']=Customer_Spend['Month'].apply(lambda x:x.month)
```

In [59]:

```
AvgMonthlySpend=Customer_Spend.groupby(['Customer','Spend_Month'])[['Amount']].mean()
```

In [60]:

```
AvgMonthlySpend.head(20)
```

Out[60]:

	Customer	Spend_Month	Amount
0	A1	1	372762.587143
1	A1	2	175235.690000
2	A1	4	261649.250000
3	A1	5	361141.310000
4	A1	8	344372.100000
5	A1	10	244534.680000
6	A1	11	183839.610000
7	A10	1	161144.326667
8	A10	2	194557.667500
9	A10	3	134631.266667
10	A10	4	270721.680000
11	A10	5	189207.392500
12	A10	7	239798.030000
13	A10	8	454827.230000
14	A10	11	459690.690000
15	A100	1	50000.000000
16	A11	1	130298.903333
17	A11	2	237842.337500
18	A11	3	177102.153333
19	A11	4	483603.210000

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [61]:

```
#f
Customer_Spend.groupby('Type').Amount.sum().reset_index().sort_values('Amount', ascending=False)
```

Out[61]:

	Type	Amount
10	PETRO	28597384.98

Type	Amount
4	CAMERA 27690738.44
7	FOOD 20519243.60
0	AIR TICKET 20155847.12
14	TRAIN TICKET 19995825.72
13	SHOPPING 17106660.97
3	BUS TICKET 16279935.35
9	MOVIE TICKET 13001899.29
6	CLOTHES 12302511.36
11	RENTAL 11850628.72
8	JEWELLERY 11507786.00
2	BIKE 7009789.53
5	CAR 5107867.45
1	AUTO 4806060.41
12	SANDALS 3848906.29

In [62]:

```
#g
Customer_Spend.groupby('City').Amount.sum().reset_index().sort_values('Amount', ascending=True)
```

Out[62]:

City	Amount
4	COCHIN 45963513.50
2	CALCUTTA 41108132.30
0	BANGALORE 40461647.02
1	BOMBAY 28603127.38
3	CHENNAI 22128635.65
7	TRIVANDRUM 15337155.87
6	PATNA 15260586.97
5	DELHI 10918286.54

In [64]:

```
#h
Customer_Spend.groupby('Age').Amount.sum()
```

Out[64]:

Age	Amount
19.00	70014.00
21.00	7662247.39
24.00	6948235.19
25.00	9749700.72
26.00	803699.35
27.00	1859124.07
28.00	17365270.23
29.00	8231607.26
31.00	312714.37
32.00	992781.50
33.00	10000.00

```

34.00      1781962.38
35.00      5980154.00
36.00      6491191.06
37.00      9992645.71
38.00      6876389.81
39.00      4891474.75
41.00      3650955.67
42.00      155015.50
43.00      5252438.32
44.00      9002326.39
46.00      4368194.79
46.49      15872551.38
47.00      9157809.12
49.00      1506267.31
50.00      8082803.39
51.00      9670808.79
52.00      5656608.44
53.00      1353183.28
54.00      4458771.82
55.00      757179.90
56.00      382522.09
57.00      169992.36
58.00      5000.00
59.00      902108.49
60.00      6346711.02
61.00      1559253.89
62.00      240401.50
63.00      3528482.07
65.00      175659.72
66.00      30066.44
67.00      7840760.08
69.00      871636.85
70.00      932200.20
71.00      1007945.92
72.00      6051138.76
74.00      1002195.63
75.00      47380.56
76.00      5142702.76
77.00      1346771.77
78.00      12329508.54
79.00      904520.69
Name: Amount, dtype: float64

```

In [66]:

```
# which age group is spending more money
Customer_Spend.loc[Customer_Spend['Amount'].idxmax()]
```

Out[66]:

No	40
Customer	A40
Age	37.0
City	CALCUTTA
Product	Platinum
Limit	500000.0
Company	C22
Segment	Salaried_Pvt
Month	2006-02-04 00:00:00
Type	SHOPPING
Amount	499270.0
Spend_Month	2

Name: 869, dtype: object

In [70]:

```
# i
customer_repayment.groupby('Customer').Amount.sum().reset_index().sort_values('Amon
```

Out[70]:

	Customer	Amount
58	A61	10539142.91
57	A60	9876290.74
5	A13	9572000.66
15	A22	9372561.68
14	A21	9056336.15
38	A43	8489871.46
40	A45	8448334.87
4	A12	8334760.16
43	A48	8292416.75
6	A14	7943268.63

In [71]:

```
#3
Customer_Spend['Yearly_Spend']=Customer_Spend['Month'].apply(lambda x:x.year)
```

In [72]:

```
City_Wise = pd.pivot_table(data = Customer_Spend, index = 'City', columns=['Product']
                           values='Amount', aggfunc='sum')
```

In [73]:

```
City_Wise
```

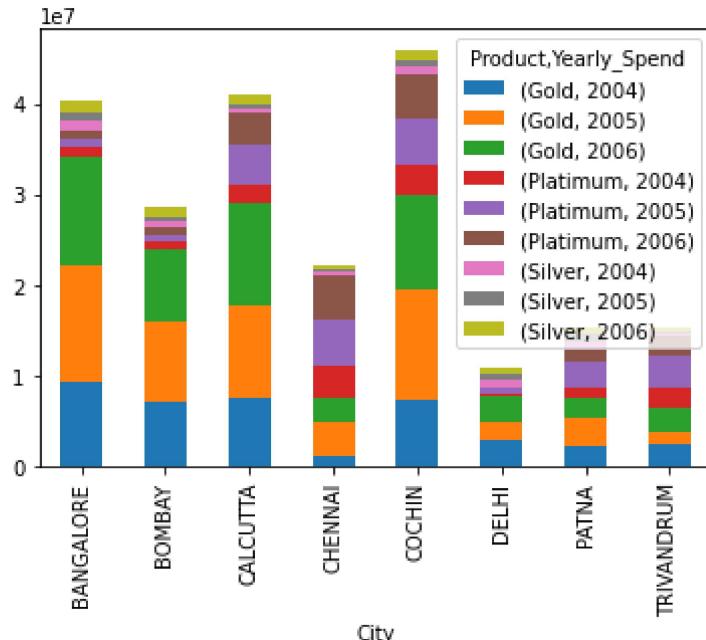
Out[73]:

Yearly_Spend	Product			Gold			Platinum	
	2004	2005	2006	2004	2005	2006	2004	2004
City	BANGALORE	9289878.54	12892362.99	12030611.09	1112732.30	912284.97	685831.27	1291852.58
	BOMBAY	6987853.53	8983018.28	7917144.31	897264.94	711804.94	972130.01	532089.03
	CALCUTTA	7477140.98	10303355.80	11167532.77	2037690.30	4434696.31	3589606.66	500006.00
	CHENNAI	1059618.50	3740945.58	2704288.62	3535652.38	5053956.32	5024923.40	282056.86
	COCHIN	7315850.15	12110613.03	10499142.38	3419904.07	4960921.71	4920592.29	814229.62
	DELHI	2806495.00	1964845.27	2957103.32	283512.89	494157.86	176788.72	768172.30
	PATNA	2072567.90	3252615.77	2276181.69	1113069.60	2883231.07	1256137.65	756684.10
	TRIVANDRUM	2415102.84	1240375.85	2735710.87	2161676.67	3601443.30	2147403.75	209667.48



In [74]:

```
City_Wise.plot(kind='bar', stacked=True)
plt.show()
```



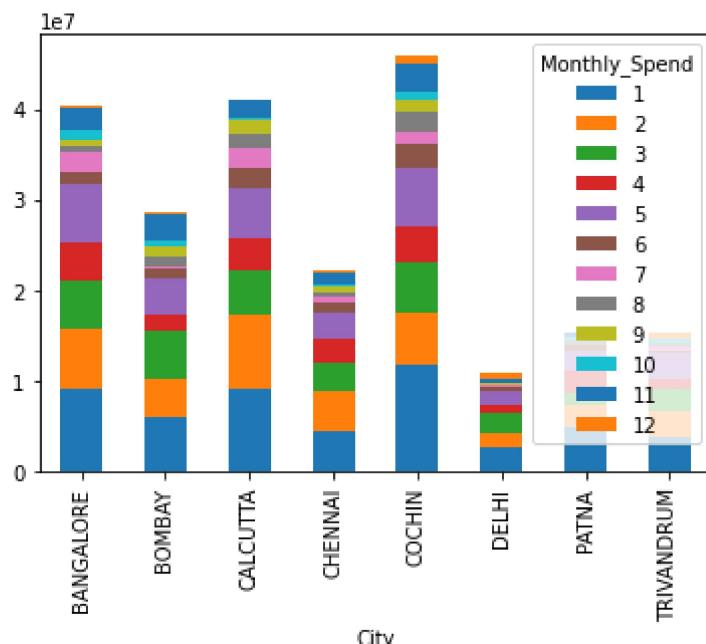
In [75]:

```
#4
Customer_Spend['Monthly_Spend']=Customer_Spend.Month.apply(lambda x:x.month)
```

In [76]:

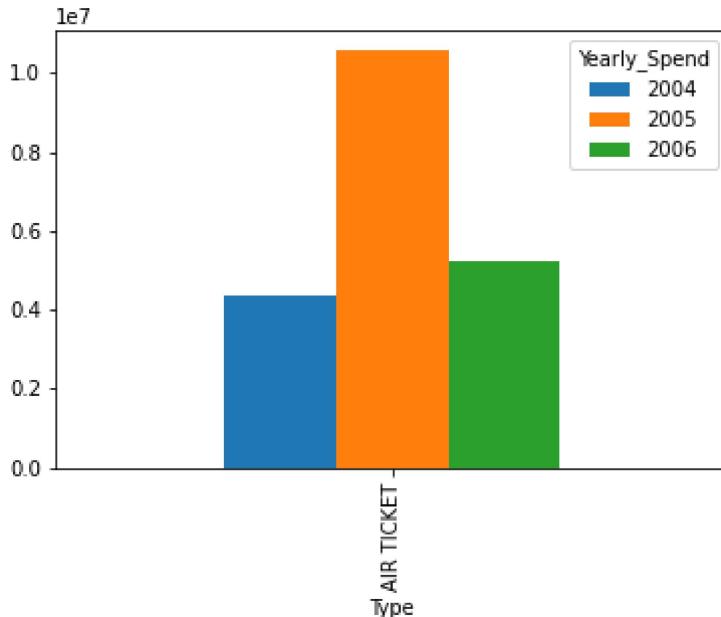
```
pd.pivot_table(data = Customer_Spend, index = 'City', columns='Monthly_Spend',
               values='Amount', aggfunc='sum').plot(kind='bar', stacked=True)
```

Out[76]:



In [79]:

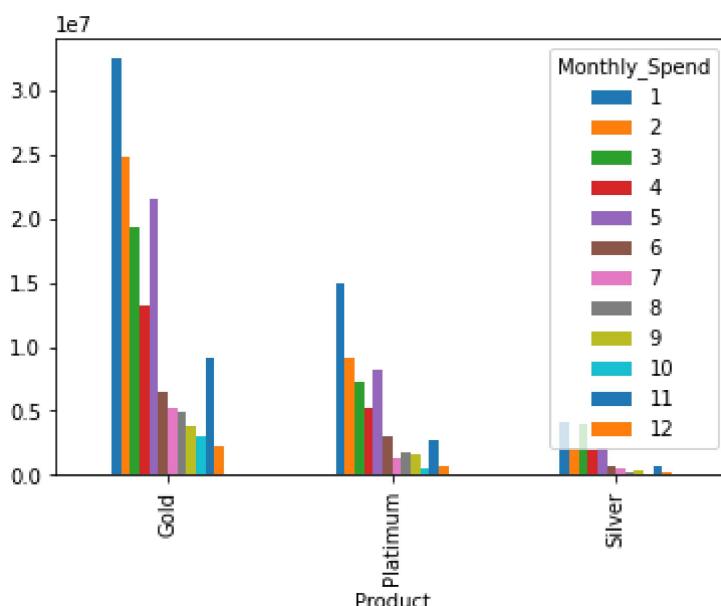
```
# B
pd.pivot_table(data = Customer_Spend[Customer_Spend.Type=='AIR TICKET'], index = 'Type',
               values='Amount', aggfunc='sum').plot(kind='bar')
plt.show()
```



In [80]:

```
#C
pd.pivot_table(data = Customer_Spend, index = 'Product', columns='Monthly_Spend',
               values='Amount', aggfunc='sum').plot(kind='bar')
```

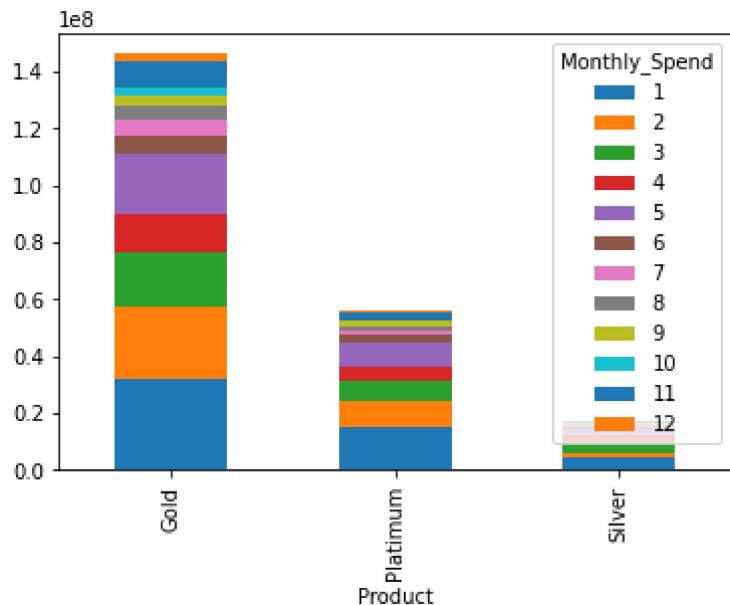
Out[80]:



In [81]:

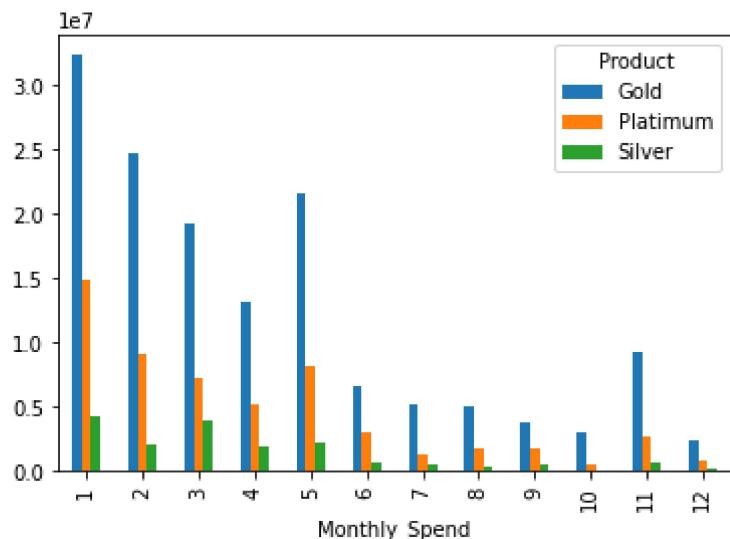
```
pd.pivot_table(data = Customer_Spend, index = 'Product', columns='Monthly_Spend',
               values='Amount', aggfunc='sum').plot(kind='bar', stacked=True)
```

Out[81]:



```
In [82]: pd.pivot_table(data = Customer_Spend, index = 'Monthly_Spend', columns='Product', values='Amount', aggfunc='sum').plot(kind='bar')
```

```
Out[82]: <AxesSubplot:xlabel='Monthly_Spend'>
```



```
In [84]: Customer_Spend.head()
```

No	Customer	Age	City	Product	Limit	Company	Segment	Month	Type
0	1	A1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed	2004-01-12 JEWELLERY 4
1	1	A1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed	2004-01-03 PETRO 4
2	1	A1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed	2004-01-15 CLOTHES
3	1	A1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed	2004-01-25 FOOD 4
4	1	A1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed	2005-01-17 CAMERA 3

In [85]:

```
#df.sort_values(['item', 'value'], ascending=False).groupby('item').head(10)
customer_repayment.head()
```

Out[85]:

	Customer	Month	Amount	No	Age	City	Product	Limit	Company	Segment
0	A1	2004-01-12	495414.75	1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed
1	A1	2004-01-03	245899.02	1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed
2	A1	2004-01-15	259490.06	1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed
3	A1	2004-01-25	437555.12	1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed
4	A1	2005-01-17	165972.88	1	76.0	BANGALORE	Gold	500000.0	C1	Self Employed

In [6]:

```
#5
Customer_Spend.nlargest(['Product', 'Amount']).groupby('City','Amount').head(10)
```

**NameError**

Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel\_3648/2810324009.py in &lt;module&gt;

----&gt; 1 Customer\_Spend.nlargest(['Product', 'Amount']).groupby('City','Amount').head(10)

**NameError:** name 'Customer\_Spend' is not defined

In [ ]: