

Credit Card Banking Data Analytics

In this project, I had learned how to work on a real world project of Data Analysis with Python.

I had used multiple commands to solve the tasks given by the mentor

The commands that I used in this project :

- DF.head(15)
- DF.tail(10)
- segment_spend.idxmax()
- segment_spend.max()
- DF["Column"].mean()
- pd.read_excel(r"path")
- pd.read_excel("card_banking_project.xls",sheet_name="Spend")
- customer_acquisition.sort_values(by="City", ascending=False, ignore_index=True)
- DF.drop("column", axis=1,inplace=True)
- DF.rename(columns = {'Old_Name':'New_Name'},inplace=True)
- DF.loc[DF["Column"] >= 50 , ["Column1","Column2","Column3"]]
- customer_acquisition.loc[customer_acquisition['Age'] >=18, 'Age']
- surplus_data = spend.merge(repayment, on=["Customer","Month"],how="outer",suffixes=("_spend","_repayment"))
- pd.merge(customer_acquisition, spend, on='Customer',how="outer")
- customer_acquisition.merge(spend, on="Customer",how="outer")
- spend["yy-mm-dd"]=spend['Date'].dt.strftime('%y-%m-%d')
- spend.groupby(['Customer', 'month_year'])['Amount'].sum().reset_index()
- spend.groupby(["Customer","month_year"]).agg({"Amount":"sum"})
- customer_spending2.sort_values(by="Amount", ascending=False)
- age_bins=[0,18,25,34,43,50,55,61,67,74,79]
- age_labels=["<18","18-25","25-34","34-43","43-50","50-55","55-61","61-67","67-74","74-79"]
- pd.cut(merged_data1["Age"], bins = age_bins, labels = age_labels)

Tasks

- 1) Monthly spend of each customer.**
- 2) Monthly repayment of each customer.**
- 3) Highest paying 10 customers.**
- 4) People in which segment are spending more money.**
- 5) Which age group is spending more money?**
- 6) Which is the most profitable segment?**
- 7) In which category the customers are spending more money?**
- 8) Monthly profit for the bank.**
- 9) Impose an interest rate of 2.9% for each customer for any due amount.**

```
In [1]: import pandas as pd
```

```
In [2]: customer_acquisition = pd.read_excel(r"C:\Users\PAIDI\Desktop\Project 1 Edulyt\Credit Banking\Customer Acquisi  
customer_acquisition
```

Out[2]:

	SI No:	Customer	Age	City	Credit Card Product	Limit	Company	Segment
0	1	A1	0.928521	BANGALORE	Gold	500000	C1	Self Employed
1	2	A2	35.534551	CALCUTTA	Silver	100000	C2	Salaried_MNC
2	3	A3	11.559307	COCHIN	Platinum	10000	C3	Salaried_Pvt
3	4	A4	45.820278	BOMBAY	Platinum	10001	C4	Govt
4	5	A5	69.663948	BANGALORE	Platinum	10002	C5	Normal Salary
...
95	96	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt
96	97	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt
97	98	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary
98	99	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed
99	100	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC

100 rows × 8 columns

```
In [3]: customer_acquisition.drop("SI No:", axis=1,inplace=True)
```

In [4]: customer_acquisition

Out[4]:

	Customer	Age	City	Credit Card Product	Limit	Company	Segment
0	A1	0.928521	BANGALORE	Gold	500000	C1	Self Employed
1	A2	35.534551	CALCUTTA	Silver	100000	C2	Salaried_MNC
2	A3	11.559307	COCHIN	Platinum	10000	C3	Salaried_Pvt
3	A4	45.820278	BOMBAY	Platinum	10001	C4	Govt
4	A5	69.663948	BANGALORE	Platinum	10002	C5	Normal Salary
...
95	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt
96	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt
97	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary
98	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed
99	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC

100 rows × 7 columns

Data Cleaning

Provide a meaningful treatment to all values where age is less than 18.

```
In [5]: #customer_acquisition[customer_acquisition['Age'] >=18]["Age"]
#customer_acquisition[customer_acquisition.Age>=18].Age
customer_acquisition.loc[customer_acquisition['Age'] >=18]
```

Out[5]:

	Customer	Age	City	Credit Card Product	Limit	Company	Segment
1	A2	35.534551	CALCUTTA	Silver	100000	C2	Salaried_MNC
3	A4	45.820278	BOMBAY	Platinum	10001	C4	Govt
4	A5	69.663948	BANGALORE	Platinum	10002	C5	Normal Salary
5	A6	35.578586	DELHI	Silver	100000	C6	Self Employed
6	A7	52.102217	COCHIN	Silver	100001	C7	Salaried_MNC
...
95	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt
96	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt
97	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary
98	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed
99	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC

78 rows × 7 columns

```
In [6]: mean_age = customer_acquisition.loc[customer_acquisition['Age'] >=18, 'Age'].mean()
mean_age
```

Out[6]: 49.29242405876662

```
In [7]: customer_acquisition.loc[customer_acquisition['Age'] < 18, 'Age'] = mean_age
```

```
In [8]: customer_acquisition.head(12)
```

Out[8]:

	Customer	Age	City	Credit Card Product	Limit	Company	Segment
0	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed
1	A2	35.534551	CALCUTTA	Silver	100000	C2	Salaried_MNC
2	A3	49.292424	COCHIN	Platinum	10000	C3	Salaried_Pvt
3	A4	45.820278	BOMBAY	Platinum	10001	C4	Govt
4	A5	69.663948	BANGALORE	Platinum	10002	C5	Normal Salary
5	A6	35.578586	DELHI	Silver	100000	C6	Self Employed
6	A7	52.102217	COCHIN	Silver	100001	C7	Salaried_MNC
7	A8	77.839595	PATNA	Silver	100002	C8	Salaried_Pvt
8	A9	18.158058	CHENNAI	Silver	100003	C9	Govt
9	A10	49.292424	CALCUTTA	Gold	500000	C10	Normal Salary
10	A11	61.341063	COCHIN	Gold	500000	C11	Normal Salary
11	A12	49.292424	BOMBAY	Gold	500000	C12	Self Employed

Identifying where the repayment is more than the spend to give them a credit of 2% of their surplus amount in the next month billing.

```
In [9]: #now Lets import spend and repayment sheet of excel  
spend = pd.read_excel(r"C:\Users\PAIDI\Desktop\Project 1 Edulyt\Credit Banking\Spend.xlsx")  
repayment = pd.read_excel(r"C:\Users\PAIDI\Desktop\Project 1 Edulyt\Credit Banking\Repayment.xlsx")
```

```
In [10]: spend
```

Out[10]:

	SI No:	Costomer	Month	Type	Amount
0	1	A1	2004-01-12	JEWELLERY	473775.834869
1	2	A1	2004-01-03	PETRO	335578.666019
2	3	A1	2004-01-15	CLOTHES	371040.941912
3	4	A1	2004-01-25	FOOD	141177.813256
4	5	A1	2005-01-17	CAMERA	398403.771812
...
1495	1496	A67	2006-02-04	BUS TICKET	195841.312215
1496	1497	A68	2006-03-25	BUS TICKET	284262.517709
1497	1498	A69	2006-03-31	BUS TICKET	206552.273598
1498	1499	A70	2006-03-23	BUS TICKET	41024.781283
1499	1500	A71	2006-03-24	BUS TICKET	101397.548588

1500 rows × 5 columns

```
In [11]: spend.drop("Sl No:", axis=1,inplace=True)
```

```
In [12]: spend.rename(columns = {'Costomer':'Customer'},inplace=True)
```

In [13]: spend

Out[13]:

	Customer	Month	Type	Amount
0	A1	2004-01-12	JEWELLERY	473775.834869
1	A1	2004-01-03	PETRO	335578.666019
2	A1	2004-01-15	CLOTHES	371040.941912
3	A1	2004-01-25	FOOD	141177.813256
4	A1	2005-01-17	CAMERA	398403.771812
...
1495	A67	2006-02-04	BUS TICKET	195841.312215
1496	A68	2006-03-25	BUS TICKET	284262.517709
1497	A69	2006-03-31	BUS TICKET	206552.273598
1498	A70	2006-03-23	BUS TICKET	41024.781283
1499	A71	2006-03-24	BUS TICKET	101397.548588

1500 rows × 4 columns

```
In [14]: repayment
```

Out[14]:

	SL No:	Costomer	Month	Amount
0	1	A1	2004-01-12	331844.007374
1	2	A1	2004-01-03	441139.807335
2	3	A1	2004-01-15	32480.004009
3	4	A1	2004-01-25	90636.966534
4	5	A1	2005-01-17	1581.969829
...
1495	1496	A67	2005-05-09	200288.874065
1496	1497	A68	2006-05-10	231194.702295
1497	1498	A69	2006-07-11	391195.615836
1498	1499	A70	2005-08-12	357629.618186
1499	1500	A71	2004-09-13	179771.745248

1500 rows × 4 columns

```
In [15]: repayment.drop("SL No:",axis=1,inplace=True)
```

```
In [16]: repayment.rename(columns = {'Costomer':'Customer'},inplace=True)
```

```
In [17]: repayment
```

Out[17]:

	Customer	Month	Amount
0	A1	2004-01-12	331844.007374
1	A1	2004-01-03	441139.807335
2	A1	2004-01-15	32480.004009
3	A1	2004-01-25	90636.966534
4	A1	2005-01-17	1581.969829
...
1495	A67	2005-05-09	200288.874065
1496	A68	2006-05-10	231194.702295
1497	A69	2006-07-11	391195.615836
1498	A70	2005-08-12	357629.618186
1499	A71	2004-09-13	179771.745248

1500 rows × 3 columns

```
In [18]: surplus_data = spend.merge(repayment, on=["Customer", "Month"], how="outer", suffixes=("_spend", "_repayment"))
```

```
In [19]: surplus_data
```

Out[19]:

	Customer	Month	Type	Amount_spend	Amount_repayment
0	A1	2004-01-12	JEWELLERY	473775.834869	331844.007374
1	A1	2004-01-12	JEWELLERY	473775.834869	390074.789989
2	A1	2004-01-12	TRAIN TICKET	52225.052808	331844.007374
3	A1	2004-01-12	TRAIN TICKET	52225.052808	390074.789989
4	A1	2004-01-03	PETRO	335578.666019	441139.807335
...
2576	A67	2005-05-09	NaN	NaN	200288.874065
2577	A68	2006-05-10	NaN	NaN	231194.702295
2578	A69	2006-07-11	NaN	NaN	391195.615836
2579	A70	2005-08-12	NaN	NaN	357629.618186
2580	A71	2004-09-13	NaN	NaN	179771.745248

2581 rows × 5 columns

```
In [20]: surplus_data["Surplus"] = surplus_data["Amount_repayment"] - surplus_data["Amount_spend"]
```

```
In [21]: surplus_data
```

Out[21]:

	Customer	Month	Type	Amount_spend	Amount_repayment	Surplus
0	A1	2004-01-12	JEWELLERY	473775.834869	331844.007374	-141931.827495
1	A1	2004-01-12	JEWELLERY	473775.834869	390074.789989	-83701.044880
2	A1	2004-01-12	TRAIN TICKET	52225.052808	331844.007374	279618.954566
3	A1	2004-01-12	TRAIN TICKET	52225.052808	390074.789989	337849.737181
4	A1	2004-01-03	PETRO	335578.666019	441139.807335	105561.141316
...
2576	A67	2005-05-09		NaN	200288.874065	NaN
2577	A68	2006-05-10		NaN	231194.702295	NaN
2578	A69	2006-07-11		NaN	391195.615836	NaN
2579	A70	2005-08-12		NaN	357629.618186	NaN
2580	A71	2004-09-13		NaN	179771.745248	NaN

2581 rows × 6 columns

```
In [22]: credit=surplus_data.loc[surplus_data["Surplus"]>0,"Surplus"]*0.02
```

```
In [23]: credit
```

Out[23]:

```
2      5592.379091
3      6756.994744
4      2111.222826
12     1640.156307
13     788.131884
...
1685    4161.320892
1686    499.104694
1687    73.702580
1690    3072.118331
1691    313.118252
Name: Surplus, Length: 406, dtype: float64
```

```
In [24]: surplus_data.loc[surplus_data["Surplus"]>0,"Credit"]=credit
```

```
In [25]: surplus_data
```

Out[25]:

	Customer	Month	Type	Amount_spend	Amount_repayment	Surplus	Credit
0	A1	2004-01-12	JEWELLERY	473775.834869	331844.007374	-141931.827495	NaN
1	A1	2004-01-12	JEWELLERY	473775.834869	390074.789989	-83701.044880	NaN
2	A1	2004-01-12	TRAIN TICKET	52225.052808	331844.007374	279618.954566	5592.379091
3	A1	2004-01-12	TRAIN TICKET	52225.052808	390074.789989	337849.737181	6756.994744
4	A1	2004-01-03	PETRO	335578.666019	441139.807335	105561.141316	2111.222826
...
2576	A67	2005-05-09	NaN	NaN	200288.874065	NaN	NaN
2577	A68	2006-05-10	NaN	NaN	231194.702295	NaN	NaN
2578	A69	2006-07-11	NaN	NaN	391195.615836	NaN	NaN
2579	A70	2005-08-12	NaN	NaN	357629.618186	NaN	NaN
2580	A71	2004-09-13	NaN	NaN	179771.745248	NaN	NaN

2581 rows × 7 columns

```
In [26]: surplus_data.drop("Surplus",axis=1,inplace=True)
```

```
In [27]: surplus_data
```

```
Out[27]:
```

	Customer	Month	Type	Amount_spend	Amount_repayment	Credit
0	A1	2004-01-12	JEWELLERY	473775.834869	331844.007374	NaN
1	A1	2004-01-12	JEWELLERY	473775.834869	390074.789989	NaN
2	A1	2004-01-12	TRAIN TICKET	52225.052808	331844.007374	5592.379091
3	A1	2004-01-12	TRAIN TICKET	52225.052808	390074.789989	6756.994744
4	A1	2004-01-03	PETRO	335578.666019	441139.807335	2111.222826
...
2576	A67	2005-05-09		NaN	200288.874065	NaN
2577	A68	2006-05-10		NaN	231194.702295	NaN
2578	A69	2006-07-11		NaN	391195.615836	NaN
2579	A70	2005-08-12		NaN	357629.618186	NaN
2580	A71	2004-09-13		NaN	179771.745248	NaN

2581 rows × 6 columns

Tasks

1) Monthly spend of each customer.

In [28]: spend

Out[28]:

	Customer	Month	Type	Amount
0	A1	2004-01-12	JEWELLERY	473775.834869
1	A1	2004-01-03	PETRO	335578.666019
2	A1	2004-01-15	CLOTHES	371040.941912
3	A1	2004-01-25	FOOD	141177.813256
4	A1	2005-01-17	CAMERA	398403.771812
...
1495	A67	2006-02-04	BUS TICKET	195841.312215
1496	A68	2006-03-25	BUS TICKET	284262.517709
1497	A69	2006-03-31	BUS TICKET	206552.273598
1498	A70	2006-03-23	BUS TICKET	41024.781283
1499	A71	2006-03-24	BUS TICKET	101397.548588

1500 rows × 4 columns

In [29]: `spend['yy-mm'] = spend['Month'].dt.strftime('%y-%m')`

```
In [30]: spend
```

```
Out[30]:
```

	Customer	Month	Type	Amount	yy-mm
0	A1	2004-01-12	JEWELLERY	473775.834869	04-01
1	A1	2004-01-03	PETRO	335578.666019	04-01
2	A1	2004-01-15	CLOTHES	371040.941912	04-01
3	A1	2004-01-25	FOOD	141177.813256	04-01
4	A1	2005-01-17	CAMERA	398403.771812	05-01
...
1495	A67	2006-02-04	BUS TICKET	195841.312215	06-02
1496	A68	2006-03-25	BUS TICKET	284262.517709	06-03
1497	A69	2006-03-31	BUS TICKET	206552.273598	06-03
1498	A70	2006-03-23	BUS TICKET	41024.781283	06-03
1499	A71	2006-03-24	BUS TICKET	101397.548588	06-03

1500 rows × 5 columns

```
In [31]: monthly_spend1 = spend.groupby(["Customer", "yy-mm"]).agg({"Amount": "sum"}) #method1
```

```
In [32]: monthly_spend1
```

Out[32]:

Customer	yy-mm	Amount
A1	04-01	1.511173e+06
	04-02	4.138111e+04
	04-05	1.311966e+05
	05-01	3.984038e+05
	05-02	1.404193e+06
...
A95	04-01	3.478339e+05
A96	04-01	3.203635e+05
A97	04-01	1.643300e+05
A98	04-01	8.748351e+04
A99	04-01	4.760204e+05

807 rows × 1 columns

```
In [33]: monthly_spend2 = spend.groupby(['Customer', 'yy-mm'])['Amount'].sum().reset_index() #method2
```

```
In [34]: monthly_spend2
```

```
Out[34]:
```

	Customer	yy-mm	Amount
0	A1	04-01	1.511173e+06
1	A1	04-02	4.138111e+04
2	A1	04-05	1.311966e+05
3	A1	05-01	3.984038e+05
4	A1	05-02	1.404193e+06
...
802	A95	04-01	3.478339e+05
803	A96	04-01	3.203635e+05
804	A97	04-01	1.643300e+05
805	A98	04-01	8.748351e+04
806	A99	04-01	4.760204e+05

807 rows × 3 columns

2) Monthly repayment of each customer.

```
In [35]: repayment
```

Out[35]:

	Customer	Month	Amount
0	A1	2004-01-12	331844.007374
1	A1	2004-01-03	441139.807335
2	A1	2004-01-15	32480.004009
3	A1	2004-01-25	90636.966534
4	A1	2005-01-17	1581.969829
...
1495	A67	2005-05-09	200288.874065
1496	A68	2006-05-10	231194.702295
1497	A69	2006-07-11	391195.615836
1498	A70	2005-08-12	357629.618186
1499	A71	2004-09-13	179771.745248

1500 rows × 3 columns

```
In [36]: repayment['yy-mm'] = repayment['Month'].dt.strftime('%y-%m')
```

```
In [37]: repayment
```

Out[37]:

	Customer	Month	Amount	yy-mm
0	A1	2004-01-12	331844.007374	04-01
1	A1	2004-01-03	441139.807335	04-01
2	A1	2004-01-15	32480.004009	04-01
3	A1	2004-01-25	90636.966534	04-01
4	A1	2005-01-17	1581.969829	05-01
...
1495	A67	2005-05-09	200288.874065	05-05
1496	A68	2006-05-10	231194.702295	06-05
1497	A69	2006-07-11	391195.615836	06-07
1498	A70	2005-08-12	357629.618186	05-08
1499	A71	2004-09-13	179771.745248	04-09

1500 rows × 4 columns

```
In [38]: monthly_repayment1 = repayment.groupby(["Customer", "yy-mm"]).agg({"Amount": "sum"})
```

```
In [39]: monthly_repayment1
```

Out[39]:

Customer	yy-mm	Amount
A1	04-01	1.362775e+06
	04-02	1.911800e+05
	05-01	1.581970e+03
	05-02	1.199808e+06
	05-07	3.005817e+05
...
A95	04-01	7.510949e+04
A96	04-01	1.101390e+05
A97	04-01	1.746064e+05
A98	04-01	9.780260e+04
A99	04-01	3.585899e+05

798 rows × 1 columns

```
In [40]: monthly_repayment2 = repayment.groupby(["Customer", "yy-mm"]).Amount.sum().reset_index()
```

```
In [41]: monthly_repayment2
```

```
Out[41]:
```

	Customer	yy-mm	Amount
0	A1	04-01	1.362775e+06
1	A1	04-02	1.911800e+05
2	A1	05-01	1.581970e+03
3	A1	05-02	1.199808e+06
4	A1	05-07	3.005817e+05
...
793	A95	04-01	7.510949e+04
794	A96	04-01	1.101390e+05
795	A97	04-01	1.746064e+05
796	A98	04-01	9.780260e+04
797	A99	04-01	3.585899e+05

798 rows × 3 columns

3) Highest paying 10 customers.

```
In [42]: customer_spending1 = spend.groupby("Customer")["Amount"].sum()
customer_spending1
```

```
Out[42]: Customer
A1      4.858294e+06
A10     4.110786e+06
A100    4.225419e+04
A11     4.581924e+06
A12     6.848587e+06
...
A95     3.478339e+05
A96     3.203635e+05
A97     1.643300e+05
A98     8.748351e+04
A99     4.760204e+05
Name: Amount, Length: 100, dtype: float64
```

```
In [43]: customer_spending2 = spend.groupby("Customer")["Amount"].sum().reset_index()
customer_spending2
```

```
Out[43]:
```

	Customer	Amount
0	A1	4.858294e+06
1	A10	4.110786e+06
2	A100	4.225419e+04
3	A11	4.581924e+06
4	A12	6.848587e+06
...
95	A95	3.478339e+05
96	A96	3.203635e+05
97	A97	1.643300e+05
98	A98	8.748351e+04
99	A99	4.760204e+05

100 rows × 2 columns

```
In [44]: highest_paying = customer_spending2.sort_values(by="Amount", ascending=False)
highest_paying.head(10)
```

Out[44]:

	Customer	Amount
15	A22	9.637819e+06
35	A40	9.595040e+06
57	A60	9.317672e+06
58	A61	9.148468e+06
43	A48	9.056652e+06
37	A42	8.665639e+06
36	A41	8.583532e+06
33	A39	8.526678e+06
40	A45	8.498612e+06
5	A13	8.042339e+06

4) People in which segment are spending more money.

In [45]: customer_acquisition

Out[45]:

	Customer	Age	City	Credit Card Product	Limit	Company	Segment
0	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed
1	A2	35.534551	CALCUTTA	Silver	100000	C2	Salaried_MNC
2	A3	49.292424	COCHIN	Platinum	10000	C3	Salaried_Pvt
3	A4	45.820278	BOMBAY	Platinum	10001	C4	Govt
4	A5	69.663948	BANGALORE	Platinum	10002	C5	Normal Salary
...
95	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt
96	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt
97	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary
98	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed
99	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC

100 rows × 7 columns

```
In [46]: spend
```

Out[46]:

	Customer	Month	Type	Amount	yy-mm
0	A1	2004-01-12	JEWELLERY	473775.834869	04-01
1	A1	2004-01-03	PETRO	335578.666019	04-01
2	A1	2004-01-15	CLOTHES	371040.941912	04-01
3	A1	2004-01-25	FOOD	141177.813256	04-01
4	A1	2005-01-17	CAMERA	398403.771812	05-01
...
1495	A67	2006-02-04	BUS TICKET	195841.312215	06-02
1496	A68	2006-03-25	BUS TICKET	284262.517709	06-03
1497	A69	2006-03-31	BUS TICKET	206552.273598	06-03
1498	A70	2006-03-23	BUS TICKET	41024.781283	06-03
1499	A71	2006-03-24	BUS TICKET	101397.548588	06-03

1500 rows × 5 columns

```
In [47]: merged_data1 = pd.merge(customer_acquisition, spend, on='Customer', how="outer")
```

```
In [48]: merged_data1
```

```
Out[48]:
```

	Customer	Age	City	Credit Card Product	Limit	Company	Segment	Month	Type	Amount	yy-mm
0	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-12	JEWELLERY	473775.834869	04-01
1	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-03	PETRO	335578.666019	04-01
2	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-15	CLOTHES	371040.941912	04-01
3	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-25	FOOD	141177.813256	04-01
4	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2005-01-17	CAMERA	398403.771812	05-01
...
1495	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt	2004-01-25	BIKE	320363.535698	04-01
1496	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt	2004-01-12	AUTO	164330.042270	04-01
1497	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary	2004-01-03	SHOPPING	87483.511992	04-01
1498	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed	2004-01-15	AIR TICKET	476020.386647	04-01
1499	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC	2004-01-25	BUS TICKET	42254.191499	04-01

1500 rows × 11 columns

```
In [49]: merged_data2 = customer_acquisition.merge(spend, on="Customer", how="outer" )
```

In [50]: merged_data2

Out[50]:

	Customer	Age	City	Credit Card Product	Limit	Company	Segment	Month	Type	Amount	yy-mm
0	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-12	JEWELLERY	473775.834869	04-01
1	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-03	PETRO	335578.666019	04-01
2	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-15	CLOTHES	371040.941912	04-01
3	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-25	FOOD	141177.813256	04-01
4	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2005-01-17	CAMERA	398403.771812	05-01
...
1495	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt	2004-01-25	BIKE	320363.535698	04-01
1496	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt	2004-01-12	AUTO	164330.042270	04-01
1497	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary	2004-01-03	SHOPPING	87483.511992	04-01
1498	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed	2004-01-15	AIR TICKET	476020.386647	04-01
1499	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC	2004-01-25	BUS TICKET	42254.191499	04-01

1500 rows × 11 columns

In [51]: segment_spend = merged_data1.groupby('Segment')[['Amount']].sum()

```
In [52]: segment_spend
```

```
Out[52]: Segment
Govt           6.732563e+07
Normal Salary  1.077071e+08
Salaried_MNC   6.363949e+07
Salaried_Pvt    7.170431e+07
Self Employed   7.097548e+07
Name: Amount, dtype: float64
```

```
In [53]: max_spend_segment = segment_spend.idxmax()
```

```
In [54]: print("Segment with the highest spend: ", max_spend_segment)
```

```
Segment with the highest spend: Normal Salary
```

5) Which age group is spending more money?

```
In [55]: merged_data1["Age"].max()
```

```
Out[55]: 78.4317555468021
```

```
In [56]: age_bins=[0,18,25,30,35,40,45,50,55,60,65,70,75,79]
age_labels=["<18","18-25","25-30","30-35","35-40","40-45","45-50","50-55","55-60","60-65","65-70","70-75","75"]
```

```
In [57]: merged_data1["Age Group"] = pd.cut(merged_data1["Age"], bins = age_bins, labels = age_labels)
```

In [58]: merged_data1.head(1600)

Out[58]:

	Customer	Age	City	Credit Card Product	Limit	Company	Segment	Month	Type	Amount	yy-mm	Age Group
0	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-12	JEWELLERY	473775.834869	04-01	45-50
1	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-03	PETRO	335578.666019	04-01	45-50
2	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-15	CLOTHES	371040.941912	04-01	45-50
3	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-25	FOOD	141177.813256	04-01	45-50
4	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2005-01-17	CAMERA	398403.771812	05-01	45-50
...
1495	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt	2004-01-25	BIKE	320363.535698	04-01	25-30
1496	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt	2004-01-12	AUTO	164330.042270	04-01	18-25
1497	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary	2004-01-03	SHOPPING	87483.511992	04-01	40-45
1498	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed	2004-01-15	AIR TICKET	476020.386647	04-01	18-25
1499	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC	2004-01-25	BUS TICKET	42254.191499	04-01	18-25

1500 rows × 12 columns



```
In [59]: age_group_spend=merged_data1.groupby("Age Group")["Amount"].sum()  
age_group_spend
```

```
Out[59]: Age Group  
<18      0.000000e+00  
18-25    2.004506e+07  
25-30    2.604367e+07  
30-35    6.349323e+05  
35-40    1.793195e+07  
40-45    2.592249e+07  
45-50    1.097903e+08  
50-55    3.046952e+07  
55-60    1.641891e+07  
60-65    8.068127e+06  
65-70    6.336432e+07  
70-75    3.631383e+07  
75-79    2.634892e+07  
Name: Amount, dtype: float64
```

```
In [60]: max_age_group=age_group_spend.idxmax()
```

```
In [61]: print("Age group spending more money is : ", max_age_group)
```

```
Age group spending more money is : 45-50
```

6) Which is the most profitable segment?

In [62]: merged_data1

Out[62]:

	Customer	Age	City	Credit Card Product	Limit	Company	Segment	Month	Type	Amount	yy-mm	Age Group
0	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-12	JEWELLERY	473775.834869	04-01	45-50
1	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-03	PETRO	335578.666019	04-01	45-50
2	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-15	CLOTHES	371040.941912	04-01	45-50
3	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-25	FOOD	141177.813256	04-01	45-50
4	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2005-01-17	CAMERA	398403.771812	05-01	45-50
...
1495	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt	2004-01-25	BIKE	320363.535698	04-01	25-30
1496	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt	2004-01-12	AUTO	164330.042270	04-01	18-25
1497	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary	2004-01-03	SHOPPING	87483.511992	04-01	40-45
1498	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed	2004-01-15	AIR TICKET	476020.386647	04-01	18-25
1499	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC	2004-01-25	BUS TICKET	42254.191499	04-01	18-25

1500 rows × 12 columns



```
In [63]: profitable_segment = merged_data1.groupby('Segment')['Amount'].sum()
profitable_segment
```

```
Out[63]: Segment
Govt           6.732563e+07
Normal Salary  1.077071e+08
Salaried_MNC   6.363949e+07
Salaried_Pvt    7.170431e+07
Self Employed   7.097548e+07
Name: Amount, dtype: float64
```

```
In [64]: most_profitable_segment = profitable_segment.idxmax()
```

```
In [65]: print("Most profitable segment is : ", most_profitable_segment)
```

```
Most profitable segment is : Normal Salary
```

7) In which category the customers are spending more money?

In [66]: merged_data1.tail(10)

Out[66]:

	Customer	Age	City	Credit Card Product	Limit	Company	Segment	Month	Type	Amount	yy-mm	Age Group
1490	A91	37.750625	BOMBAY	Gold	500000	C14	Govt	2006-04-03	CLOTHES	193571.988519	06-04	35-40
1491	A92	58.059809	BANGALORE	Gold	500000	C15	Govt	2006-04-04	FOOD	296766.523715	06-04	55-60
1492	A93	47.310159	DELHI	Gold	500000	C16	Govt	2004-01-12	CAMERA	366993.643430	04-01	45-50
1493	A94	55.916868	COCHIN	Gold	500000	C17	Govt	2004-01-03	SANDALS	321836.011508	04-01	55-60
1494	A95	44.280682	PATNA	Gold	500000	C18	Salaried_MNC	2004-01-15	CAR	347833.879524	04-01	40-45
1495	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt	2004-01-25	BIKE	320363.535698	04-01	25-30
1496	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt	2004-01-12	AUTO	164330.042270	04-01	18-25
1497	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary	2004-01-03	SHOPPING	87483.511992	04-01	40-45
1498	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed	2004-01-15	AIR TICKET	476020.386647	04-01	18-25
1499	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC	2004-01-25	BUS TICKET	42254.191499	04-01	18-25

```
In [67]: category_spend = merged_data1.groupby('Type')['Amount'].sum()  
category_spend
```

```
Out[67]: Type  
AIR TICKET      3.743547e+07  
AUTO            1.050509e+07  
BIKE            1.315264e+07  
BUS TICKET      2.490590e+07  
CAMERA          4.372101e+07  
CAR              7.018278e+06  
CLOTHES         2.479110e+07  
FOOD             3.829646e+07  
JEWELLERY       2.524795e+07  
MOVIE TICKET    1.878458e+07  
PETRO           5.102258e+07  
RENTAL          2.091467e+07  
SANDALS         6.325017e+06  
SHOPPING         2.741868e+07  
TRAIN TICKET    3.181262e+07  
Name: Amount, dtype: float64
```

```
In [68]: most_category_spend = category_spend.idxmax()
```

```
In [69]: print("Category with highest spend is : ", most_category_spend)
```

```
Category with highest spend is : PETRO
```

8) Monthly profit for the bank.

```
In [70]: surplus_data
```

Out[70]:

	Customer	Month	Type	Amount_spend	Amount_repayment	Credit
0	A1	2004-01-12	JEWELLERY	473775.834869	331844.007374	NaN
1	A1	2004-01-12	JEWELLERY	473775.834869	390074.789989	NaN
2	A1	2004-01-12	TRAIN TICKET	52225.052808	331844.007374	5592.379091
3	A1	2004-01-12	TRAIN TICKET	52225.052808	390074.789989	6756.994744
4	A1	2004-01-03	PETRO	335578.666019	441139.807335	2111.222826
...
2576	A67	2005-05-09		NaN	200288.874065	NaN
2577	A68	2006-05-10		NaN	231194.702295	NaN
2578	A69	2006-07-11		NaN	391195.615836	NaN
2579	A70	2005-08-12		NaN	357629.618186	NaN
2580	A71	2004-09-13		NaN	179771.745248	NaN

2581 rows × 6 columns

```
In [71]: surplus_data["yy-mm"] = surplus_data["Month"].dt.strftime("%y-%m")
```

Correction Here

`surplus_data["Amount_repayment"] - surplus_data["Amount_spend"] -
surplus_data["Credit"].fillna(0)`

```
In [72]: surplus_data["Profit"] = surplus_data["Amount_repayment"] - surplus_data["Amount_spend"] + surplus_data["Cred
```

```
In [73]: surplus_data.head(4)
```

Out[73]:

	Customer	Month	Type	Amount_spend	Amount_repayment	Credit	yy-mm	Profit
0	A1	2004-01-12	JEWELLERY	473775.834869	331844.007374	NaN	04-01	-141931.827495
1	A1	2004-01-12	JEWELLERY	473775.834869	390074.789989	NaN	04-01	-83701.044880
2	A1	2004-01-12	TRAIN TICKET	52225.052808	331844.007374	5592.379091	04-01	285211.333657
3	A1	2004-01-12	TRAIN TICKET	52225.052808	390074.789989	6756.994744	04-01	344606.731925

```
In [74]: bank_monthly_profit = surplus_data.groupby("yy-mm")["Profit"].sum()
```

```
In [75]: bank_monthly_profit
```

```
Out[75]: yy-mm  
04-01    -1.842143e+06  
04-02    -8.167732e+05  
04-03    -6.930477e+05  
04-04    -2.665711e+05  
04-05    -8.967566e+05  
04-09     6.429638e+05  
04-11    -4.555242e+05  
05-01     2.566693e+05  
05-02    -9.877306e+05  
05-04    -2.751795e+05  
05-05    -4.933499e+06  
05-06     6.719868e+05  
05-07     2.952780e+05  
05-08    -7.101375e+05  
05-09    -1.458112e+05  
05-10     7.442726e+04  
05-11     4.868276e+05  
05-12     8.412751e+05  
06-01    -1.125459e+06  
06-02    -1.368432e+05  
06-03     1.366408e+06  
06-04     6.552110e+04  
06-05    -5.994178e+05  
06-06     3.058554e+05  
06-07     3.048222e+05  
06-08     6.934106e+05  
06-09     1.500036e+05  
06-10    -9.787128e+04  
06-11    -3.793046e+05  
06-12     4.914672e+05  
Name: Profit, dtype: float64
```

```
In [76]: bank_monthly_profit[bank_monthly_profit>0].reset_index()
```

Out[76]:

	yy-mm	Profit
0	04-09	6.429638e+05
1	05-01	2.566693e+05
2	05-06	6.719868e+05
3	05-07	2.952780e+05
4	05-10	7.442726e+04
5	05-11	4.868276e+05
6	05-12	8.412751e+05
7	06-03	1.366408e+06
8	06-04	6.552110e+04
9	06-06	3.058554e+05
10	06-07	3.048222e+05
11	06-08	6.934106e+05
12	06-09	1.500036e+05
13	06-12	4.914672e+05

9) Impose an interest rate of 2.9% for each customer for any due amount.

```
In [77]: #lets merge the customer_data and repayment_data  
merge_data = pd.merge(customer_acquisition, repayment, on="Customer", how="outer")
```

```
In [78]: merge_data
```

Out[78]:

	Customer	Age	City	Credit Card Product	Limit	Company	Segment	Month	Amount	yy-mm
0	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-12	331844.007374	04-01
1	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-03	441139.807335	04-01
2	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-15	32480.004009	04-01
3	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-25	90636.966534	04-01
4	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2005-01-17	1581.969829	05-01
...
1495	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt	2004-01-25	110139.014148	04-01
1496	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt	2004-01-12	174606.392283	04-01
1497	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary	2004-01-03	97802.598995	04-01
1498	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed	2004-01-15	358589.924345	04-01
1499	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC	2004-01-25	151141.391939	04-01

1500 rows × 10 columns

```
In [79]: merge_data["Interest"] = merge_data["Amount"]*0.029
```

```
In [80]: merge_data
```

```
Out[80]:
```

	Customer	Age	City	Credit Card Product	Limit	Company	Segment	Month	Amount	yy-mm	Interest
0	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-12	331844.007374	04-01	9623.476214
1	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-03	441139.807335	04-01	12793.054413
2	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-15	32480.004009	04-01	941.920116
3	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2004-01-25	90636.966534	04-01	2628.472029
4	A1	49.292424	BANGALORE	Gold	500000	C1	Self Employed	2005-01-17	1581.969829	05-01	45.877125
...
1495	A96	29.631637	CHENNAI	Silver	100000	C19	Salaried_Pvt	2004-01-25	110139.014148	04-01	3194.031410
1496	A97	20.611833	TRIVANDRUM	Platinum	10000	C20	Govt	2004-01-12	174606.392283	04-01	5063.585376
1497	A98	40.538985	CALCUTTA	Platinum	10001	C21	Normal Salary	2004-01-03	97802.598995	04-01	2836.275371
1498	A99	21.588666	CALCUTTA	Platinum	10002	C22	Self Employed	2004-01-15	358589.924345	04-01	10399.107806
1499	A100	23.607638	COCHIN	Silver	100000	C5	Salaried_MNC	2004-01-25	151141.391939	04-01	4383.100366

1500 rows × 11 columns

```
In [81]: total_interest=merge_data["Interest"].sum()
```

```
In [82]: total_interest
```

```
Out[82]: 10765044.879641727
```

----- * -----

By - P. Yesu Krishna (Data Analyst)