In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
data=pd.read_csv('bank_csv.csv')
data.head()
```

Out[2]:

	id	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaig
0	1	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79	
1	2	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220	
2	3	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185	
3	4	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199	
4	5	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226	

In [3]:

```
data.shape
```

Out[3]:

```
(4521, 18)
```

In [39]:

```
# group by job status
#data["job"].unique()
jb=data.groupby(by="job")["balance"].sum()
jb.reset_index().sort_values(by="balance", ascending =False)
```

Out[39]:

	job	balance
4	management	1712154
1	blue-collar	1026563
9	technician	1022205
0	admin.	586380
5	retired	533414
7	services	460350
2	entrepreneur	276381
6	self-employed	254811
3	housemaid	233386
10	unemployed	139446
8	student	129681
11	unknown	57065

In [38]:

```
# the mean balance by month
mb=data.groupby(by="month")["balance"].mean()
mb.reset_index().sort_values(by="balance", ascending =False)
```

Out[38]:

	month	balance
2	dec	3567.150000
10	oct	2738.650000
9	nov	2603.154242
7	mar	2049.530612
6	jun	1884.792844
0	apr	1658.061433
11	sep	1589.346154
1	aug	1454.559242
3	feb	1319.040541
8	may	1104.173820
4	jan	976.364865
5	jul	789.467422

In [36]:

```python
# the average age grouped by job and martial status and sort them accordingly
jmb = data.groupby(by=["job","marital"])["age"].mean()
jmb.reset_index().sort_values(by="age",ascending=True)
```

Out[36]:

	job	marital	age
25	student	single	25.864865
32	unknown	divorced	31.000000
23	services	single	32.621849
8	entrepreneur	single	32.900000
31	unemployed	single	32.935484
5	blue-collar	single	33.477011
24	student	married	33.900000
28	technician	single	34.100746
2	admin.	single	34.552448
20	self-employed	single	35.268293
14	management	single	35.334471
34	unknown	single	37.285714
22	services	married	40.377119
11	housemaid	single	40.733333
1	admin.	married	41.484962
4	blue-collar	married	41.516595
27	technician	married	42.296837
30	unemployed	married	42.466667
13	management	married	42.488330
26	technician	divorced	42.584270
19	self-employed	married	42.637795
7	entrepreneur	married	42.856061
3	blue-collar	divorced	42.936709

	job	marital	age
21	services	divorced	43.112903
0	admin.	divorced	43.362319
12	management	divorced	44.243697
6	entrepreneur	divorced	46.437500
29	unemployed	divorced	46.818182
9	housemaid	divorced	47.538462
17	retired	single	47.818182
18	self-employed	divorced	48.333333
10	housemaid	married	48.488095
33	unknown	married	51.200000
16	retired	married	62.227273
15	retired	divorced	64.000000

In [56]:

```
# mean age of people with house loan
with_housing_loan = data[(data["housing"]=="yes") & (data["loan"]=="yes")]
housing_loan =with_housing_loan.groupby(by=["housing","loan","job"])["age"].mean()
housing_loan =housing_loan.reset_index().sort_values(by="age")
housing_loan.drop(columns=["housing","loan"])
```

Out[56]:

	job	age
1	blue-collar	37.281553
7	services	37.439024
8	technician	38.109589
9	unemployed	38.111111
0	admin.	38.642857
2	entrepreneur	38.724138
4	management	39.000000
6	self-employed	41.933333
3	housemaid	46.200000
10	unknown	50.000000
5	retired	54.538462

In [55]:

```
# mean balance of people with house loan
hl_balance= with_housing_loan.groupby(by=["housing","loan","job"])["balance"].mean()
h1_balance= hl_balance.reset_index().sort_values(by="balance", ascending=False)
h1_balance.drop(columns=["housing","loan"])
```

Out[55]:

	job	balance
4	management	1632.229508
5	retired	1461.692308
2	entrepreneur	1288.586207
3	housemaid	1144.600000
1	blue-collar	884.135922
0	admin.	708.517857
6	self-employed	610.066667
7	services	568.682927
9	unemployed	561.111111
8	technician	532.452055
10	unknown	341.000000