Aggregations

Step 1. Import the necessary libraries

In [41]:

```
import pandas as pd
import numpy as np
import seaborn as sb
```

Step 2. Import the dataset occupation.csv from the folder

In [42]:

```
A=pd.read_csv("occupation.csv",sep="|")
A.head()
```

Out[42]:

	user_id	age	gender	occupation	zip_code
0	1	24	M	technician	85711
1	2	53	F	other	94043
2	3	23	M	writer	32067
3	4	24	M	technician	43537
4	5	33	F	other	15213

Step 3. Assign it to a variable called users.

In [43]:

```
users=A
```

Step 4. Discover what is the mean age per occupation

In [44]:

```
user=users.groupby('occupation')
user.mean()
```

Out[44]:

	user_id	age
occupation		
administrator	430.949367	38.746835
artist	451.892857	31.392857
doctor	533.714286	43.571429
educator	466.905263	42.010526
engineer	456.328358	36.388060
entertainment	398.000000	29.222222
executive	422.312500	38.718750
healthcare	501.437500	41.562500
homemaker	443.000000	32.571429
lawyer	359.083333	36.750000
librarian	486.588235	40.000000
marketing	437.807692	37.615385
none	368.666667	26.555556
other	542.733333	34.523810
programmer	435.530303	33.121212
retired	515.714286	63.071429
salesman	494.916667	35.666667
scientist	465.129032	35.548387
student	484.954082	22.081633
technician	497.629630	33.148148
writer	495.711111	36.311111

Step 5. Discover the Male ratio per occupation and sort it from the most to the least.

Use numpy.where() to encode gender column.

In [ ]:

```
```

Step 6. For each occupation, calculate the minimum and maximum ages

In [52]:

```
users.groupby('occupation')['age'].aggregate([min,max])
```

Out[52]:

	min	max
occupation		
administrator	21	70
artist	19	48
doctor	28	64
educator	23	63
engineer	22	70
entertainment	15	50
executive	22	69
healthcare	22	62
homemaker	20	50
lawyer	21	53
librarian	23	69
marketing	24	55
none	11	55
other	13	64
programmer	20	63
retired	51	73
salesman	18	66
scientist	23	55
student	7	42
technician	21	55
writer	18	60

Step 7. For each combination of occupation and gender, calculate the mean age

In [50]:

```
users.groupby(['gender','occupation'])['age'].mean()
```

Out[50]:

```
gender  occupation
F       administrator    40.638889
        artist           30.307692
        educator         39.115385
        engineer         29.500000
        entertainment    31.000000
        executive        44.000000
        healthcare       39.818182
        homemaker        34.166667
        lawyer           39.500000
        librarian        40.000000
        marketing        37.200000
        none             36.500000
        other            35.472222
        programmer       32.166667
        retired          70.000000
        salesman         27.000000
        scientist        28.333333
        student          20.750000
        technician       38.000000
        writer           37.631579
M       administrator    37.162791
        artist           32.333333
        doctor           43.571429
        educator         43.101449
        engineer         36.600000
        entertainment    29.000000
        executive        38.172414
        healthcare       45.400000
        homemaker        23.000000
        lawyer           36.200000
        librarian        40.000000
        marketing        37.875000
        none             18.600000
        other            34.028986
        programmer       33.216667
        retired          62.538462
        salesman         38.555556
        scientist        36.321429
        student          22.669118
        technician       32.961538
        writer           35.346154
Name: age, dtype: float64
```

Step 8. For each occupation present the percentage of women and men

In [ ]:

```

```