Occupation

Introduction:

Special thanks to: https://github.com/justmarkham for sharing the dataset and materials.

Step 1. Import the necessary libraries

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

Step 2. Import the dataset from this address.

Step 3. Assign it to a variable called users.

In [5]:
```
df = pd.read_csv("https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user", sep="|")
df
```

Out[5]:

	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
...	...	...	...	...	...
938	939	26	F	student	33319
939	940	32	M	administrator	02215
940	941	20	M	student	97229
941	942	48	F	librarian	78209
942	943	22	M	student	77841

943 rows × 5 columns

Step 4. Discover what is the mean age per occupation

In [6]:
```
df.groupby("occupation").age.mean()
```

Out[6]:
```
occupation
administrator    38.746835
artist           31.392857
doctor           43.571429
educator         42.010526
engineer         36.388060
entertainment    29.222222
executive        38.718750
healthcare       41.562500
homemaker        32.571429
lawyer           36.750000
librarian        40.000000
marketing        37.615385
none             26.555556
other            34.523810
programmer       33.121212
retired          63.071429
salesman         35.666667
scientist        35.548387
student          22.081633
technician       33.148148
writer           36.311111
Name: age, dtype: float64
```

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

In [7]:
```
df['is_male'] = df.gender.apply(lambda x: True if x == 'M' else False)
df['is_male']
```

Out[7]:
```
0       True
1      False
2       True
3       True
4      False
       ...
938    False
939     True
940     True
941    False
942     True
Name: is_male, Length: 943, dtype: bool
```

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

In [9]:
```
df.groupby('occupation').age.agg(['min', 'max'])
```

Out[9]:

occupation	min	max
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 [10]:
```
df.groupby(['occupation', 'gender']).age.mean()
```

Out[10]:
```
occupation     gender
administrator  F         40.638889
               M         37.162791
artist         F         30.307692
               M         32.333333
doctor         M         43.571429
educator       F         39.115385
               M         43.101449
engineer       F         29.500000
               M         36.600000
entertainment  F         31.000000
               M         29.000000
executive      F         44.000000
               M         38.172414
healthcare     F         39.818182
               M         45.400000
homemaker      F         34.166667
               M         23.000000
lawyer         F         39.500000
               M         36.200000
librarian      F         40.000000
               M         40.000000
marketing      F         37.200000
               M         37.875000
none           F         36.500000
               M         18.600000
other          F         35.472222
               M         34.028986
programmer     F         32.166667
               M         33.216667
retired        F         70.000000
               M         62.538462
salesman       F         27.000000
               M         38.555556
scientist      F         28.333333
               M         36.321429
student        F         20.750000
               M         22.669118
technician     F         38.000000
               M         32.961538
writer         F         37.631579
               M         35.346154
Name: age, dtype: float64
```

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

In [12]:
```
occup_count = df.groupby(['occupation']).count()
occup_count
```

Out[12]:

occupation	user_id	age	gender	zip_code	is_male
administrator	79	79	79	79	79
artist	28	28	28	28	28
doctor	7	7	7	7	7
educator	95	95	95	95	95
engineer	67	67	67	67	67
entertainment	18	18	18	18	18
executive	32	32	32	32	32
healthcare	16	16	16	16	16
homemaker	7	7	7	7	7
lawyer	12	12	12	12	12
librarian	51	51	51	51	51
marketing	26	26	26	26	26
none	9	9	9	9	9
other	105	105	105	105	105
programmer	66	66	66	66	66
retired	14	14	14	14	14
salesman	12	12	12	12	12
scientist	31	31	31	31	31
student	196	196	196	196	196
technician	27	27	27	27	27
writer	45	45	45	45	45

In [13]:
```
df.describe(include="all")
```

Out[13]:

	user_id	age	gender	occupation	zip_code	is_male
count	943.000000	943.000000	943	943	943	943
unique	NaN	NaN	2	21	795	2
top	NaN	NaN	M	student	55414	True
freq	NaN	NaN	670	196	9	670
mean	472.000000	34.051962	NaN	NaN	NaN	NaN
std	272.364951	12.192740	NaN	NaN	NaN	NaN
min	1.000000	7.000000	NaN	NaN	NaN	NaN
25%	236.500000	25.000000	NaN	NaN	NaN	NaN
50%	472.000000	31.000000	NaN	NaN	NaN	NaN
75%	707.500000	43.000000	NaN	NaN	NaN	NaN
max	943.000000	73.000000	NaN	NaN	NaN	NaN