Occupation

Introduction:

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

Step 1. Import the necessary libraries

```
import pandas as pd
```

Step 2. Import the dataset from this address.

```
url = "https://raw.githubusercontent.com/thieu1995/csv-files/main/data/pandas/u.user"
```

Step 3. Assign it to a variable called users.

```
users = pd.read_csv(url, sep="|")
users.head()
```

	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

Next steps: Generate code with users | View recommended plots | New interactive sheet

Step 4. Discover what is the mean age per occupation

```
mean_age = users.groupby("occupation")["age"].mean()

print(mean_age)
```

```
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

```
male_ratio = users[users['gender'] == 'M']

male_ratio = male_ratio.groupby('occupation')['gender'].count() / users.groupby('occupation')['gender'].count()

male_ratio.head()
```

occupation	gender
administrator	0.544304
artist	0.535714
doctor	1.000000
educator	0.726316
engineer	0.970149

dtype: float64

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

```
age_stats = users.groupby("occupation")["age"].agg(['min', 'max'])

print(age_stats)
```

```
               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
```

```
min = users.groupby("occupation")['age'].min()

print(min)
```

```
occupation
administrator    21
artist           19
doctor           28
educator         23
engineer         22
entertainment    15
executive        22
healthcare       22
homemaker        20
lawyer           21
librarian        23
marketing        24
none             11
other            13
programmer       20
retired          51
salesman         18
scientist        23
student           7
technician       21
writer           18
Name: age, dtype: int64
```

```
max = users.groupby("occupation")['age'].max()

print(max)
```

```
occupation
administrator    70
artist           48
doctor           64
educator         63
engineer         70
entertainment    50
```

```
executive        69
healthcare       62
homemaker        50
lawyer           53
librarian        69
marketing        55
none             55
other            64
programmer       63
retired          73
salesman         66
scientist        55
student          42
technician       55
writer           60
Name: age, dtype: int64
```

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

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

print(combination)
```

```
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

```
gender_counts = users.groupby(['occupation', 'gender']).size()

total_counts = users.groupby('occupation').size()

gender_percentage = gender_counts / total_counts * 100

print(gender_percentage)
```

```
occupation     gender
administrator  F          45.569620
               M          54.430380
artist         F          46.428571
               M          53.571429
doctor         M         100.000000
educator       F          27.368421
```

```
                  M         72.631579
engineer          F          2.985075
                  M         97.014925
entertainment     F         11.111111
                  M         88.888889
executive         F          9.375000
                  M         90.625000
healthcare        F         68.750000
                  M         31.250000
homemaker         F         85.714286
                  M         14.285714
lawyer            F         16.666667
                  M         83.333333
librarian         F         56.862745
                  M         43.137255
marketing         F         38.461538
                  M         61.538462
none              F         44.444444
                  M         55.555556
other             F         34.285714
                  M         65.714286
programmer        F          9.090909
                  M         90.909091
retired           F          7.142857
                  M         92.857143
salesman          F         25.000000
                  M         75.000000
scientist         F          9.677419
                  M         90.322581
student           F         30.612245
                  M         69.387755
technician        F          3.703704
                  M         96.296296
writer            F         42.222222
                  M         57.777778
dtype: float64
```

Start coding or generate with AI.