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.

```
df = pd.read_csv('https://raw.githubusercontent.com/thieu1995/csv-files/main/data/pandas/u.user', sep='|')
df.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

Step 3. Assign it to a variable called users.

```
users = pd.read_csv('https://raw.githubusercontent.com/thieu1995/csv-files/main/data/pandas/u.user', 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

Step 4. Discover what is the mean age per occupation

```
users.groupby('occupation').age.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

dtype: float64

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

```
def gender_to_numeric(x):
    if x == 'M':
        return 1
    if x == 'F':
        return 0
    else:
        return 0

users['gender_n'] = users['gender'].apply(gender_to_numeric)
a = users.groupby('occupation').gender_n.sum() / users.occupation.value_counts() * 100
a.sort_values(ascending = False)
```

```
                       0
   occupation
       doctor  100.000000
     engineer   97.014925
   technician   96.296296
      retired   92.857143
   programmer   90.909091
    executive   90.625000
    scientist   90.322581
entertainment   88.888889
       lawyer   83.333333
     salesman   75.000000
     educator   72.631579
      student   69.387755
        other   65.714286
    marketing   61.538462
       writer   57.777778
         none   55.555556
administrator   54.430380
       artist   53.571429
    librarian   43.137255
   healthcare   31.250000
    homemaker   14.285714

dtype: float64
```

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

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

```
              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

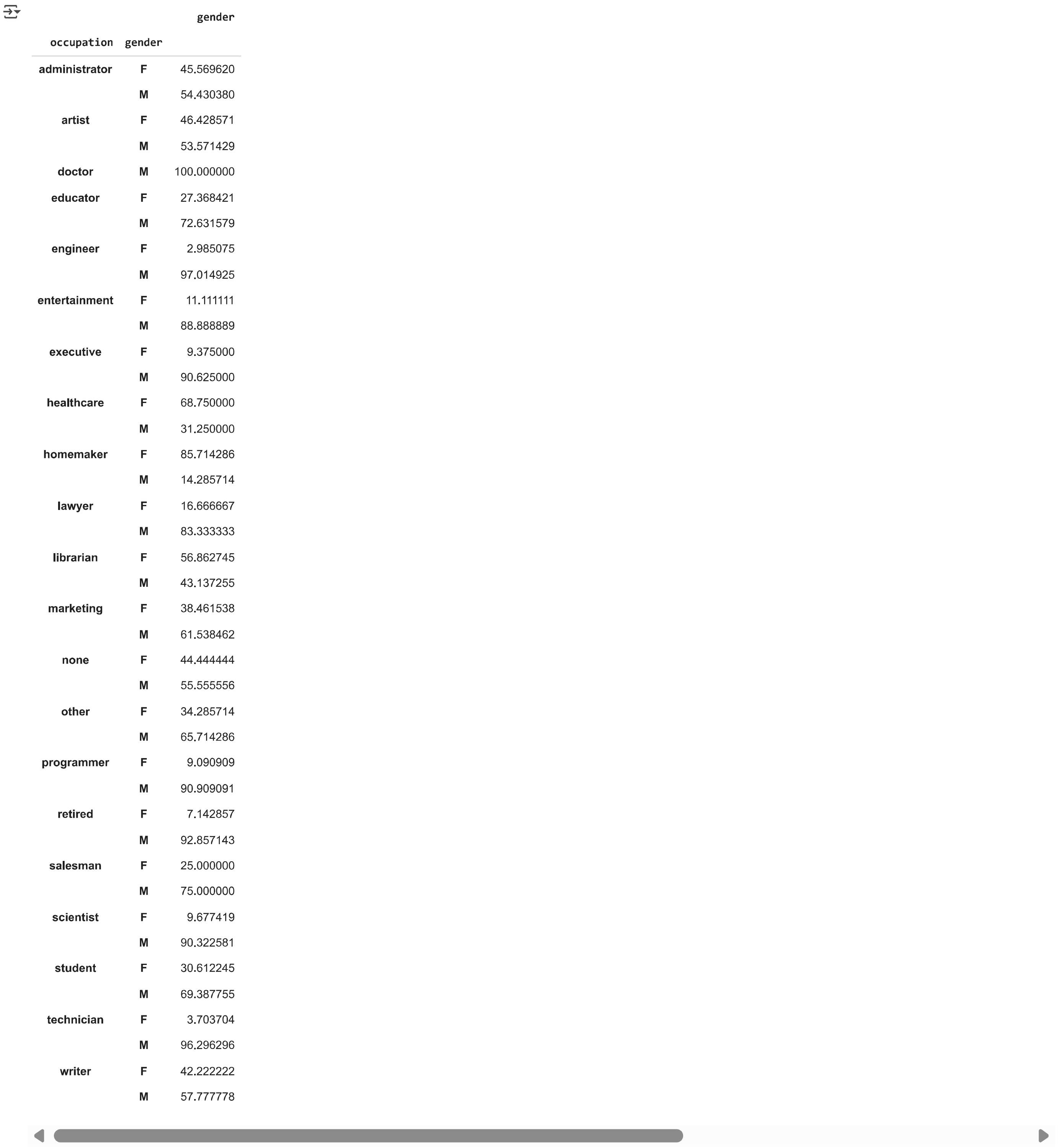
```
users.groupby(['occupation', 'gender']).age.mean()
```

```
                                  age
     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

dtype: float64
```

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

```
gender_ocup = users.groupby(['occupation', 'gender']).agg({'gender': 'count'})
occup_count = users.groupby(['occupation']).agg('count')
occup_gender = gender_ocup.div(occup_count, level = "occupation") * 100
occup_gender.loc[: , 'gender']
```



		gender
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