Exercise 2

Let's try this exercise

Import the necessary libraries

In [1]:

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

Get the dataset from this address (https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user) and import the data

In [2]:

```
data=pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user',se
data
```

Out[2]:

	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
5	6	42	M	executive	98101
6	7	57	M	administrator	91344
7	8	36	M	administrator	05201
8	9	29	M	student	01002
9	10	53	M	lawyer	90703

Assign it to a variable called users and use the 'user_id' as index

In [3]:

```
users=pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user',s
users
```

Out[3]:

user_id	age	gender	occupation	zip_code
1	24	M	technician	85711
2	53	F	other	94043
3	23	M	writer	32067
4	24	M	technician	43537
5	33	F	other	15213
6	42	M	executive	98101
7	57	M	administrator	91344
8	36	M	administrator	05201
9	29	M	student	01002

See the first 25 entries

In [4]:

```
users.head(25)
```

Out[4]:

user_id	age	gender	occupation	zip_code
1	24	M	technician	85711
2	53	F	other	94043
3	23	M	writer	32067
4	24	M	technician	43537
5	33	F	other	15213
6	42	M	executive	98101
7	57	M	administrator	91344
8	36	M	administrator	05201
9	29	M	student	01002

See the last 10 entries

In [5]:

```
users.tail(10)
```

Out[5]:

user_id	age	gender	occupation	zip_code
934	61	M	engineer	22902
935	42	M	doctor	66221
936	24	M	other	32789
937	48	M	educator	98072
938	38	F	technician	55038
939	26	F	student	33319
940	32	M	administrator	02215
941	20	M	student	97229
942	48	F	librarian	78209
943	22	M	student	77841

What is the number of observations in the dataset?

In [6]:

```
users.shape ## to observe number of rows and column
```

Out[6]:

```
(943, 4)
```

In [7]:

```
len(users) ## to observe number or rows
```

Out[7]:

```
943
```

What is the number of columns in the dataset?

In [8]:

```
len(users.columns)
```

Out[8]:

```
4
```

Print the name of all the columns.

In [9]:

```
list(users)
```

Out[9]:

```
['age', 'gender', 'occupation', 'zip_code']
```

How is the dataset indexed?

In [10]:

```
users.index
```

Out[10]:

```
Int64Index([  1,   2,   3,   4,   5,   6,   7,   8,   9,  10,
            ...
            934, 935, 936, 937, 938, 939, 940, 941, 942, 943],
           dtype='int64', name='user_id', length=943)
```

What is the data type of each column?

In [11]:

```
users.dtypes
```

Out[11]:

```
age            int64
gender        object
occupation    object
zip_code      object
dtype: object
```

Print only the occupation column

In [12]:

```
users.iloc[:,[2]]
```

Out[12]:

user_id	occupation
1	technician
2	other
3	writer
4	technician
5	other
6	executive
7	administrator
8	administrator
9	student

How many different occupations there are in this dataset?

In [13]:

```
len(users.occupation.unique())
```

Out[13]:

```
21
```

What are the five most frequent occupation?

In [14]:

```
users.occupation.value_counts()
```

Out[14]:

```
student          196
other            105
educator          95
administrator     79
engineer          67
programmer        66
librarian         51
writer            45
executive         32
scientist         31
artist            28
technician        27
marketing         26
entertainment     18
healthcare        16
retired           14
lawyer            12
salesman          12
```

In [15]:

```
users.occupation.value_counts().head(5)
```

Out[15]:

```
student          196
other            105
educator          95
administrator     79
engineer          67
Name: occupation, dtype: int64
```

Summarize the quantitative variable in the DataFrame.

In [16]:

```
users.describe()
```

Out[16]:

	age
count	943.000000
mean	34.051962
std	12.192740
min	7.000000
25%	25.000000
50%	31.000000
75%	43.000000
max	73.000000

Summarize all the columns

In [17]:

```
users.describe(include='all')
```

Out[17]:

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

Summarize the qualitative variables in the DataFrame.

In [18]:

```
users.describe(include=[np.object])
```

Out[18]:

	gender	occupation	zip_code
count	943	943	943
unique	2	21	795
top	M	student	55414
freq	670	196	9

What is the mean age of users?

In [19]:

```
users.age.mean()
```

Out[19]:

```
34.05196182396607
```

What is the age with least occurrence?

In [20]:

```
age_least=pd.DataFrame(users.age)
age_least['TotalNumber']=age_least.groupby('age')['age'].transform('count')
age_least[age_least.TotalNumber==1]
```

Out[20]:

user_id	age	TotalNumber
30	7	1
211	66	1
289	11	1
471	10	1
481	73	1

What is the occupation of the person with user id equal to 19?

In [21]:

```
users.occupation[19]
```

Out[21]:

```
'librarian'
```

Sort the data according to age.

In [22]:

```
users.sort_values(['age'])
```

Out[22]:

user_id	age	gender	occupation	zip_code
30	7	M	student	55436
471	10	M	student	77459
289	11	M	none	94619
880	13	M	student	83702
609	13	F	student	55106
142	13	M	other	48118
674	13	F	student	55337
628	13	M	none	94306
813	14	F	student	02136

Find out the oldest person in this data.

In [23]:

```
oldest_person=users[users.age==users.age.max()]
oldest_person
```

Out[23]:

user_id	age	gender	occupation	zip_code
481	73	M	retired	37771

Find out the youngest engineer in this data.

In [24]:

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

Out[24]:

user_id	age	gender	occupation	zip_code
216	22	M	engineer	02215
487	22	M	engineer	92121
493	22	M	engineer	60090
844	22	M	engineer	95662

Find out the occupation of people age between 27-32 years old.

In [25]:

```
newdata=users.occupation[(users.age>27) & (users.age<32)]
newdata
```

Out[25]:

```
user_id
9            student
12             other
17        programmer
23            artist
32           student
38             other
42     administrator
43         librarian
45        programmer
51          educator
63         marketing
77        technician
95     administrator
109            other
112         salesman
115         engineer
125           lawyer
```

In [26]:

```
newdata1=users.occupation[(users.age>27) & (users.age<32)]

list (newdata1.drop_duplicates())

## to list occupation description for people age between 27 and 32 (both age 27 and 32 excl
```

Out[26]:

```
['student',
 'other',
 'programmer',
 'artist',
 'administrator',
 'librarian',
 'educator',
 'marketing',
 'technician',
 'salesman',
 'engineer',
 'lawyer',
 'entertainment',
 'scientist',
 'writer',
 'executive',
 'doctor',
 'healthcare',
 'none']
```

In [27]:

```
newdata2=users[(users.age>27) & (users.age<32)]
newdata2.occupation.unique()
newdata2.drop_duplicates()

## to list down all the records that full fill the condition of occupation with age between
```

Out[27]:

user_id	age	gender	occupation	zip_code
9	29	M	student	01002
12	28	F	other	06405
17	30	M	programmer	06355
23	30	F	artist	48197
32	28	F	student	78741
38	28	F	other	54467
42	30	M	administrator	17870
43	29	F	librarian	20854
45	29	M	programmer	50233

Show the data for people age more than 60 years old.

In [28]:

```
users[users.age>60]
```

Out[28]:

user_id	age	gender	occupation	zip_code
106	61	M	retired	55125
211	66	M	salesman	32605
266	62	F	administrator	78756
318	65	M	retired	06518
349	68	M	retired	61455
351	61	M	educator	49938
364	63	M	engineer	01810
423	64	M	other	91606
481	73	M	retired	37771

Present only the age of homemakers in this data.

In [29]:

```
homemaker=users[users.occupation=='homemaker']
homemaker.age
```

Out[29]:

```
user_id
20     42
35     20
356    32
362    35
708    26
722    50
898    23
Name: age, dtype: int64
```

What is the average age of each occupation in the DataFrame?

In [30]:

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

Out[30]:

```
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
```

In [ ]:

```

```