Exercise - Getting and Knowing your Data-Occupation Dataset

This time we are going to pull data directly from the internet.

Step 1. Import the necessary libraries

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

Step 2. Import the dataset from this address.

```
In [3]: pd.read_csv("https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user",sep='|')
```

Out[3]:

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

In [4]:

```
user_id=pd.read_csv("https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user",sep='|')
```

Step 4. See the first 25 entries

In [5]:

```
user_id.head(25)
```

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
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
10	11	39	F	other	30329
11	12	28	F	other	06405
12	13	47	M	educator	29206
13	14	45	M	scientist	55106
14	15	49	F	educator	97301
15	16	21	M	entertainment	10309
16	17	30	M	programmer	06355
17	18	35	F	other	37212
18	19	40	M	librarian	02138
19	20	42	F	homemaker	95660
20	21	26	M	writer	30068
21	22	25	M	writer	40206
22	23	30	F	artist	48197
23	24	21	F	artist	94533

	user_id	age	gender	occupation	zip_code
24	25	39	M	engineer	55107

Step 5. See the last 10 entries

In [6]:

```
user_id.tail(10)
```

Out[6]:

	user_id	age	gender	occupation	zip_code
933	934	61	M	engineer	22902
934	935	42	M	doctor	66221
935	936	24	M	other	32789
936	937	48	M	educator	98072
937	938	38	F	technician	55038
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

Step 6. What is the number of observations in the dataset?

In [7]:

```
user_id.shape[0]
```

Out[7]:

```
943
```

Step 7. What is the number of columns in the dataset?

In [8]:

```
user_id.shape[1]
```

```
Out[8]: 5
```

Step 8. Print the name of all the columns.

```
In [11]: user_id.columns
```

```
Out[11]: Index(['user_id', 'age', 'gender', 'occupation', 'zip_code'], dtype='object')
```

Step 9. How is the dataset indexed?

```
In [20]: user_id.loc
```

```
Out[20]: <pandas.core.indexing._LocIndexer at 0x26adc9af860>
```

Step 10. What is the data type of each column?

```
In [13]: user_id.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 943 entries, 0 to 942
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   user_id     943 non-null    int64 
 1   age         943 non-null    int64 
 2   gender      943 non-null    object
 3   occupation  943 non-null    object
 4   zip_code    943 non-null    object
dtypes: int64(2), object(3)
memory usage: 37.0+ KB
```

```
In [21]: df.dtypes
```

```
Out[21]: user_id        int64
         age            int64
         gender        object
         occupation    object
         zip_code      object
         dtype: object
```

Step 11. Print only the occupation column

```
In [22]: df.occupation
```

```
Out[22]: 0         technician
         1              other
         2             writer
         3         technician
         4              other
                    ...
         938          student
         939    administrator
         940          student
         941        librarian
         942          student
         Name: occupation, Length: 943, dtype: object
```

Step 12. How many different occupations are in this dataset?

```
In [25]: df['occupation'].unique()
```

```
Out[25]: array(['technician', 'other', 'writer', 'executive', 'administrator',
                'student', 'lawyer', 'educator', 'scientist', 'entertainment',
                'programmer', 'librarian', 'homemaker', 'artist', 'engineer',
                'marketing', 'none', 'healthcare', 'retired', 'salesman', 'doctor'],
               dtype=object)
```

Step 13. What is the most frequent occupation?

```
In [26]: max(df['occupation'])
```

```
Out[26]: 'writer'
```

Step 14. Summarize the DataFrame.

```
In [32]: df.describe()
```

```
Out[32]:
          user_id         age
count  943.000000  943.000000
mean   472.000000   34.051962
std    272.364951   12.192740
min      1.000000    7.000000
25%    236.500000   25.000000
50%    472.000000   31.000000
75%    707.500000   43.000000
max    943.000000   73.000000
```

Step 15. Summarize all the columns

```
In [33]: df.describe(include='all')
```

```
Out[33]:
           user_id         age  gender  occupation  zip_code
count   943.000000  943.000000     943         943       943
unique         NaN         NaN       2          21       795
top            NaN         NaN       M     student     55414
freq           NaN         NaN     670         196         9
mean    472.000000   34.051962     NaN         NaN       NaN
std     272.364951   12.192740     NaN         NaN       NaN
min       1.000000    7.000000     NaN         NaN       NaN
25%     236.500000   25.000000     NaN         NaN       NaN
50%     472.000000   31.000000     NaN         NaN       NaN
75%     707.500000   43.000000     NaN         NaN       NaN
max     943.000000   73.000000     NaN         NaN       NaN
```

Step 16. Summarize only the occupation column

In [35]:

```
df.occupation.describe(include='all')
```

Out[35]:

```
count         943
unique         21
top       student
freq          196
Name: occupation, dtype: object
```

Step 17. What is the mean age of users?

In [40]:

```
df.age.describe(include='all')
```

Out[40]:

```
count    943.000000
mean      34.051962
std       12.192740
min        7.000000
25%       25.000000
50%       31.000000
75%       43.000000
max       73.000000
Name: age, dtype: float64
```

In [41]:

```
df['age'].mean()
```

Out[41]:

```
34.05196182396607
```

Step 18. What is the age with least occurrence?

In [38]:

```
df['age'].min()
```

Out[38]:

```
7
```

In [ ]: