Ex3 - Getting and Knowing your Data

This time we are going to pull data directly from the internet. 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 and use the 'user_id' as index

In [25]:
```
df = pd.read_table("https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user", sep='|', index_col='user_id')
df
```

Out[25]:

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
...	...	...	...	...
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

943 rows × 4 columns

Step 4. See the first 25 entries

In [26]:
```
df.head(25)
```

user_id	age	gender	occupation	zip_code
13	47	M	educator	29206
14	45	M	scientist	55106
15	49	F	educator	97301
16	21	M	entertainment	10309
17	30	M	programmer	06355
18	35	F	other	37212
19	40	M	librarian	02138
20	42	F	homemaker	95660
21	26	M	writer	30068
22	25	M	writer	40206
23	30	F	artist	48197
24	21	F	artist	94533
25	39	M	engineer	55107

Step 5. See the last 10 entries

In [27]:
```
df.tail(10)
```

Out[27]:

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

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

In [28]:
```
df.shape[0]
```

Out[28]: 943

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

In [29]:
```
df.shape[1]
```

Out[29]: 4

Step 8. Print the name of all the columns.

In [30]:
```
df.columns
```

Out[30]: Index(['age', 'gender', 'occupation', 'zip_code'], dtype='object')

Step 9. How is the dataset indexed?

In [31]:
```
df.index
```

Out[31]:
```
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)
```

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

In [32]:
```
df.dtypes
```

Out[32]:
```
age           int64
gender       object
occupation   object
zip_code     object
dtype: object
```

Step 11. Print only the occupation column

In [35]:
```
df.occupation
```

Out[35]:
```
user_id
1         technician
2              other
3             writer
4         technician
5              other
            ...
939          student
940    administrator
941          student
942        librarian
943          student
Name: occupation, Length: 943, dtype: object
```

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

In [38]:
```
df.occupation.nunique()
```

Out[38]: 21

Step 13. What is the most frequent occupation?

In [45]:
```
df.occupation.value_counts().head(3)
```

Out[45]:
```
student     196
other       105
educator     95
Name: occupation, dtype: int64
```

Step 14. Summarize the DataFrame.

In [48]:
```
df.describe()
```

Out[48]:

	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

Step 15. Summarize all the columns

In [50]:
```
df.describe(include="all")
```

Out[50]:

	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

Step 16. Summarize only the occupation column

In [53]:
```
df.occupation.describe()
```

Out[53]:
```
count         943
unique         21
top       student
freq          196
Name: occupation, dtype: object
```

In [54]:
```
df.age.describe()
```

Out[54]:
```
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
```

Step 17. What is the mean age of users?

In [57]:
```
round(df.age.mean())
```

Out[57]: 34

Step 18. What is the age with least occurrence?

In [58]:
```
df.age.value_counts().tail()
```

Out[58]:
```
7     1
66    1
11    1
10    1
73    1
Name: age, dtype: int64
```