Ex3 - Getting and Knowing your Data

In [2]:
```
#import necessary libraries.

import pandas as pd
import numpy as np
```

In [3]:
```
#import necessary data by given link.

users = ('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user')
```

In [4]:
```
#assign it as users.

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

In [5]:
```
#check users all data.

users
```

Out[5]:

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

In [6]:
```
#check data starting from25 string.

users.head(25)
```

Out[6]:

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
10	53	M	lawyer	90703
11	39	F	other	30329
12	28	F	other	06405
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

In [7]:
```
#check data of lowest 10 values.

users.tail(10)
```

Out[7]:

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

In [12]:
```
#what is the number of observations in the dataset?

users.shape[0]
```

Out[12]:
```
943
```

In [11]:
```
#what is the number of columns in the dataset?

users.shape[1]
```

Out[11]:
```
4
```

In [9]:
```
#Print the name of all the columns.

users.columns
```

Out[9]:
```
Index(['age', 'gender', 'occupation', 'zip_code'], dtype='object')
```

In [13]:
```
#How is the dataset indexed?

users.index
```

Out[13]:
```
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)
```

In [14]:
```
#what is the data type of each column?

users.dtypes
```

Out[14]:
```
age           int64
gender       object
occupation   object
zip_code     object
dtype: object
```

In [15]:
```
#Print only the age column

users.age
```

Out[15]:
```
user_id
1      24
2      53
3      23
4      24
5      33
       ..
939    26
940    32
941    20
942    48
943    22
Name: age, Length: 943, dtype: int64
```

In [15]:
```
#to find the particular number of data which can run as zip code.

users.zip_code[456:675]
```

Out[15]:
```
user_id
457    30011
458    Y1A6B
459    29201
460    60630
461    98102
       ...
671    91919
672    90095
673    22906
674    55337
675    28814
Name: zip_code, Length: 219, dtype: object
```

In [16]:
```
#How many different occupations are in this dataset?

users.occupation.unique()
```

Out[16]:
```
array(['technician', 'other', 'writer', 'executive', 'administrator',
       'student', 'lawyer', 'educator', 'scientist', 'entertainment',
       'programmer', 'librarian', 'homemaker', 'artist', 'engineer',
       'marketing', 'none', 'healthcare', 'retired', 'salesman', 'doctor'},
      dtype=object)
```

In [26]:
```
#summarize the data set

users.describe()
```

Out[26]:

	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

In [29]:
```
#summarize the data set

users.describe()
```

Out[29]:

	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

In [18]:
```
#what is the most frequent occupation?

users.occupation.value_counts().head(1).index[0]

#Because "most" is asked
#users.occupation.value_counts().head(1).index[0]

#or
#to have the top 5

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

Out[18]:
```
'student'
```

In [20]:
```
#Summarize all the columns

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

Out[20]:

	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

In [21]:
```
#Summarize only the occupation column

users.occupation.describe()
```

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

In [22]:
```
#what is the mean age of users?

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

Out[22]:
```
34
```

In [23]:
```
#what is the age with least occurrence?

users.age.value_counts().tail() #7, 10, 11, 66 and 73 years -> only 1 occurrence
```

Out[23]:
```
7     1
66    1
10    1
11    1
73    1
Name: age, dtype: int64
```

In [ ]:
```

```