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
import seaborn as sns
```

Step 2. Import the dataset from this address.

In [2]:
```
'https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user'
```

Out[2]:
```
'https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user'
```

Step 3. 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.
```

Step 4. See the first 25 entries

In [4]:
```
users.head()
```

Out[4]:

	age	gender	occupation	zip_code
user_id				
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

Step 5. See the last 10 entries

In [5]:
```
users.tail(10)
```

Out[5]:

	age	gender	occupation	zip_code
user_id				
934	61	M	engineer	22902
935	42	M	doctor	66221
936	24	M	other	32789

	age	gender	occupation	zip_code
user_id				
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 [6]:
```
users.shape[0]
```

Out[6]:
```
943
```

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

In [7]:
```
users.shape[1]
```

Out[7]:
```
4
```

Step 8. Print the name of all the columns.

In [8]:
```
users.columns
```

Out[8]:
```
Index(['age', 'gender', 'occupation', 'zip_code'], dtype='object')
```

Step 9. How is the dataset indexed?

In [9]:
```
users.index
```

Out[9]:
```
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 [10]:
```
users.dtypes
```

Out[10]:
```
age            int64
gender        object
occupation    object
zip_code      object
dtype: object
```

Step 11. Print only the occupation column

In [11]:
```
users.occupation
```

Out[11]:
```
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
```

In [12]:
```
users['occupation']

#this is also how we can show the results.
```

Out[12]:
```
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 [13]:
```
users.occupation.nunique()
```

Out[13]:
```
21
```

Step 13. What is the most frequent occupation?

In [14]:
```
users.occupation.value_counts().head(1).index[0]
```

Out[14]:
```
'student'
```

Step 14. Summarize the DataFrame.

In [15]:
```
users.describe()
```

Out[15]:

	age
count	943.000000
mean	34.051962
std	12.192740
min	7.000000

	age
25%	25.000000
50%	31.000000
75%	43.000000
max	73.000000

Step 15. Summarize all the columns

In [16]:
```
users.describe(include='all')
```

Out[16]:

	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 [18]:
```
users.occupation.describe()
```

Out[18]:
```
count          943
unique          21
top        student
freq           196
Name: occupation, dtype: object
```

16.a Summarize only the age column

In [19]:
```
users.age.describe()
```

Out[19]:
```
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
```

16.b Summarize only the gender column

In [20]:
```
users.gender.describe()
```

Out[20]:
```
count     943
unique      2
top         M
freq      670
Name: gender, dtype: object
```

16.c Summarize only the zip_code column

In [21]:
```
users.zip_code.describe()
```

Out[21]:
```
count       943
unique      795
top       55414
freq          9
Name: zip_code, dtype: object
```

Step 17. What is the mean age of users?

In [22]:
```
round(users.age.mean())
```

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

In [28]:
```
round(users.age.std())
```

Out[28]:
```
12
```

In [29]:
```
round(users.age.min())
```

Out[29]:
```
7
```

In [30]:
```
round(users.age.max())
```

Out[30]:
```
73
```

In [40]:
```
round(users.age.median())
```

Out[40]:
```
31
```

In [41]:
```
round(users.age.mode())
```

Out[41]:
```
0    30
dtype: int64
```

In [42]:
```
round(users.age.kurtosis())
```

Out[42]:
```
0
```

In [46]:

```
round(users.age.skew())
```

Out[46]:
```
1
```

Step 18. What is the age with least occurrence?

In [55]:
```
users.age.value_counts().tail(10)
```

Out[55]:
```
14    3
62    2
69    2
64    2
68    2
7     1
66    1
10    1
11    1
73    1
Name: age, dtype: int64
```