Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Exercise 07 Getting Knowing Your Data with Pandas

Name: Ooi Caaron

IC Number: 990701-07-5837

Date : 27/6/23

Introduction : Learning and use pandas built in functions

Conclusion : Still need to practice more

Ex07 Getting and Knowing your Data with Pandas

This time we are going to pull data directly from the internet. Special thanks to:
https://github.com/justmarkham (https://github.com/justmarkham) for sharing the dataset and materials.

Step 1. Import the necessary libraries

In [1]:

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

Step 2. Import the dataset from this address (https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user).

In [21]:

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

Out[21]:

	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

In [7]:

```

```

```
  File "/var/folders/l5/q27cjv5x7bg68ngcklmljcdr0000gn/T/ipykernel_3134/23
71991143.py", line 1
    df = pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/m
aster/data/u.user' sep = "|")

^
SyntaxError: invalid syntax
```

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

In [30]:

```
# data.rename(columns={
#          '': 'user_id',
#     }, inplace=True)
# data.columns
users = data.set_index('user_id')
users
```

Out[30]:

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 [31]:

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

Out[31]:

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

Step 5. See the last 10 entries

In [32]:

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

Out[32]:

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 [33]:

```
users.shape[0]
```

Out[33]:

```
943
```

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

In [35]:

```
users.shape[1]
```

Out[35]:

```
4
```

Step 8. Print the name of all the columns.

In [36]:

```
users.columns
```

Out[36]:

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

Step 9. How is the dataset indexed?

In [39]:

```
users.index
```

Out[39]:

```
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 [43]:

```
users.dtypes
```

Out[43]:

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

Step 11. Print only the occupation column

In [46]:

```
# df['occupation']
users['occupation']
```

Out[46]:

```
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 [52]:

```
users['occupation'].nunique()
```

Out[52]:

```
21
```

Step 13. What is the most frequent occupation?

In [55]:

```
users['occupation'].value_counts().head(1)
```

Out[55]:

```
student    196
Name: occupation, dtype: int64
```

Step 14. Summarize the DataFrame.

In [56]:

```
users.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 943 entries, 1 to 943
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   age         943 non-null    int64 
 1   gender      943 non-null    object
 2   occupation  943 non-null    object
 3   zip_code    943 non-null    object
dtypes: int64(1), object(3)
memory usage: 36.8+ KB
```

Step 15. Summarize all the columns

In [62]:

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

Out[62]:

	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 [63]:

```
users['occupation'].describe()
```

Out[63]:

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

Step 17. What is the mean age of users?

In [64]:

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

Out[64]:

```
34.05196182396607
```

Step 18. What is the age with least occurrence?

In [70]:

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

Out[70]:

```
73     1
7      1
10     1
11     1
66     1
      ..
27    35
28    36
22    37
25    38
30    39
Name: age, Length: 61, dtype: int64
```

In [ ]:

```

```