Forward
School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Exercise 07 Getting Knowing Your Data with Pandas

Name: Phua Yan Han

IC Number: 050824070059

Date : 27/6/23

Introduction : learning pandas

Conclusion : learn some basic function for pandas

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.u](https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.u)

In [11]:
```
data = pd.read_csv("https://raw.githubusercontent.com/justmarkham/DAT8/master/d
data
```

Out[11]:

	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

In [7]:
```

```

```
  File "/var/folders/l5/q27cjv5x7bg68ngcklmljcdr0000gn/T/ipykernel_3134/23719
91143.py", line 1
    df = pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/mast
er/data/u.user' sep = "|")

^
SyntaxError: invalid syntax
```

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

In [12]:
```
data.set_index('user_id', inplace=True)
```

Step 4. See the first 25 entries

In [13]:

```
data.head(25)
```

Out[13]:

	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
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

Step 5. See the last 10 entries

In [14]:

```
data.tail(10)
```

Out[14]:

	age	gender	occupation	zip_code
user_id				
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 [15]: len(data)
```

```
Out[15]: 943
```

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

```
In [16]: data.shape[1]
```

```
Out[16]: 4
```

Step 8. Print the name of all the columns.

```
In [18]: data.columns
```

```
Out[18]: Index(['age', 'gender', 'occupation', 'zip_code'], dtype='object')
```

Step 9. How is the dataset indexed?

```
In [19]: data.index
```

```
Out[19]: 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 [21]: data.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 11. Print only the occupation column

```
In [22]: data['occupation']
```

```
Out[22]: 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 [26]: len(data['occupation'].unique())
```

```
Out[26]: 21
```

Step 13. What is the most frequent occupation?

```
In [29]: data['occupation'].value_counts().head(1)
```

```
Out[29]: student    196
         Name: occupation, dtype: int64
```

Step 14. Summarize the DataFrame.

```
In [31]: data.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 [35]:
```
data.describe(include=['object', 'int'])
```

Out[35]:

	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 [36]:
```
data["occupation"].describe(include=['object', 'int'])
```

Out[36]:
```
count          943
unique          21
top        student
freq           196
Name: occupation, dtype: object
```

Step 17. What is the mean age of users?

In [37]:
```
data["age"].mean()
```

Out[37]:
```
34.05196182396607
```

Step 18. What is the age with least occurrence?

```
In [49]: data['age'].value_counts().sort_values(ascending=False).loc[::-1]
```

```
Out[49]: 73     1
         10     1
         11     1
         66     1
         7      1
               ..
         27    35
         28    36
         22    37
         25    38
         30    39
         Name: age, Length: 61, dtype: int64
```