

✓ 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

```
import pandas as pd
```

Step 2. Import the dataset from this [address](#).

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

```
user_id = pd.read_csv('https://raw.githubusercontent.com/thieu1995/csv-files/main/data/pandas/u.user', sep='|', index_col='user_id')
user_id
```



	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
...
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

```
user_id.head(25)
```



	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
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	20	M	engineer	55107

▼ Step 5. See the last 10 entries

```
user_id.tail(10)
```

```
↗
```

	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

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

```
user_id.shape[0]
```

```
↗ 943
```

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

```
user_id.shape[1]
```

```
↗ 4
```

▼ Step 8. Print the name of all the columns.

```
user_id.columns
```

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

▼ Step 9. How is the dataset indexed?

```
user_id.index
```

```
↗ Index([ 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?

```
user_id.dtypes
```

```
↗
```

	0
age	int64
gender	object
occupation	object
zip_code	object

▼ Step 11. Print only the occupation column

```
print(user_id['occupation'])
```

```
↗
```

user_id	occupation
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?

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

Step 13. What is the most frequent occupation?

```
user_id['occupation'].value_counts().idxmax()
```



Step 14. Summarize the DataFrame.

```
user_id.describe()
```



	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

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



	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

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



	occupation
count	943
unique	21
top	student
freq	196

Step 17. What is the mean age of users?

```
print(user_id['age'].mean())
```



```
34.05196182396607
```

Step 18. What is the age with least occurrence?

```
print(user_id['age'].value_counts().idxmin())
```



```
7
```

