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.

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

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

```
users = data.set_index('user_id')
```

Step 4. See the first 25 entries

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

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

Next steps: Generate code with users | View recommended plots | New interactive sheet

Step 5. See the last 10 entries

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

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?

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

```
943
```

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

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

```
4
```

Step 8. Print the name of all the columns.

```
users.columns
```

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

Step 9. How is the dataset indexed?

```
users.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?

```
users.dtypes
```

	0
age	int64
gender	object
occupation	object
zip_code	object

dtype: object

Step 11. Print only the occupation column

```
users['occupation']
```

user_id	occupation
1	technician
2	other
3	writer
4	technician
5	other
...	...
939	student
940	administrator
941	student
942	librarian
943	student

943 rows × 1 columns

dtype: object

Step 12. How many different occupations are in this dataset?

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

```
21
```

Step 13. What is the most frequent occupation?

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

```
'student'
```

Step 14. Summarize the DataFrame.

```
users.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 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

```
users.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

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

occupation	count
student	196
other	105
educator	95
administrator	79
engineer	67
programmer	66
librarian	51
writer	45
executive	32
scientist	31
artist	28
technician	27
marketing	26
entertainment	18
healthcare	16
retired	14
lawyer	12
salesman	12
none	9
homemaker	7
doctor	7

dtype: int64

Step 17. What is the mean age of users?

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

```
34.05196182396607
```

Step 18. What is the age with least occurrence?

```
int(users['age'].value_counts().idxmin())
```

```
7
```

Start coding or generate with AI.