

Explore Query function in pandas

Lambda function with filter and map function

In []:

Query() function

- 1.The query() method allows you to query the DataFrame.
- 2.It is used to filter rows in DataFrame
- 3.Query function

In [1]:

```
import pandas as pd
```

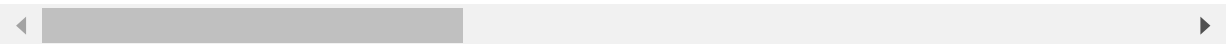
In [7]:

```
data= pd.read_csv("D:\ProITBridge\Beginner\Day6\student-mat.csv")
pd.set_option("display.max_columns", None)
data
```

Out[7]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother
1	GP	F	17	U	GT3	T	1	1	at_home	other	course	father
2	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother
3	GP	F	15	U	GT3	T	4	2	health	services	home	mother
4	GP	F	16	U	GT3	T	3	3	other	other	home	father
...
390	MS	M	20	U	LE3	A	2	2	services	services	course	other
391	MS	M	17	U	LE3	T	3	1	services	services	course	mother
392	MS	M	21	R	GT3	T	1	1	other	other	course	other
393	MS	M	18	R	LE3	T	3	2	services	other	course	mother
394	MS	M	19	U	LE3	T	1	1	other	at_home	course	father

395 rows × 33 columns



In [20]:

```
df2=data.query("sex == 'F'").head(10)
df2
```

Out[20]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother
1	GP	F	17	U	GT3	T	1	1	at_home	other	course	father
2	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother
3	GP	F	15	U	GT3	T	4	2	health	services	home	mother

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian
4	GP	F	16	U	GT3	T	3	3	other	other	home	father
7	GP	F	17	U	GT3	A	4	4	other	teacher	home	mother
10	GP	F	15	U	GT3	T	4	4	teacher	health	reputation	mother
11	GP	F	15	U	GT3	T	2	1	services	other	reputation	father
15	GP	F	16	U	GT3	T	4	4	health	other	home	mother
16	GP	F	16	U	GT3	T	4	4	services	services	reputation	mother

```
In [25]: df3=data.query("Mjob == 'at_home']").head(10)
df3
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother
1	GP	F	17	U	GT3	T	1	1	at_home	other	course	father
2	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother
39	GP	F	15	R	GT3	T	2	2	at_home	other	reputation	mother
79	GP	F	16	U	GT3	T	3	4	at_home	other	course	mother
86	GP	F	16	U	LE3	T	2	2	at_home	other	course	mother
95	GP	F	15	R	GT3	T	1	1	at_home	other	home	mother
112	GP	F	16	U	GT3	T	2	2	at_home	other	home	mother
120	GP	F	15	U	GT3	T	1	2	at_home	services	course	mother
127	GP	F	19	U	GT3	T	0	1	at_home	other	course	other

```
In [29]: df4=data.query("absences >= 10").head(5)
df4
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian
2	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother
5	GP	M	16	U	LE3	T	4	3	services	other	reputation	mother
18	GP	M	17	U	GT3	T	3	2	services	services	course	mother
25	GP	F	16	U	GT3	T	2	2	services	services	home	mother
29	GP	M	16	U	GT3	T	4	4	teacher	teacher	home	mother

Lambda function

In [33]: `data["Fjob_details"]=data["Fjob"].apply(lambda x: 1 if x=='teacher' else 0)`

In [34]: `data.head(10)`

Out[34]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother
1	GP	F	17	U	GT3	T	1	1	at_home	other	course	father
2	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother
3	GP	F	15	U	GT3	T	4	2	health	services	home	mother
4	GP	F	16	U	GT3	T	3	3	other	other	home	father
5	GP	M	16	U	LE3	T	4	3	services	other	reputation	mother
6	GP	M	16	U	LE3	T	2	2	other	other	home	mother
7	GP	F	17	U	GT3	A	4	4	other	teacher	home	mother
8	GP	M	15	U	LE3	A	3	2	services	other	home	mother
9	GP	M	15	U	GT3	T	3	4	other	other	home	mother

In [35]: `data["mother_G"]=data["guardian"].apply(lambda x: 1 if x=='mother' else 0)`
`data.head(10)`

Out[35]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother
1	GP	F	17	U	GT3	T	1	1	at_home	other	course	father
2	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother
3	GP	F	15	U	GT3	T	4	2	health	services	home	mother
4	GP	F	16	U	GT3	T	3	3	other	other	home	father
5	GP	M	16	U	LE3	T	4	3	services	other	reputation	mother
6	GP	M	16	U	LE3	T	2	2	other	other	home	mother
7	GP	F	17	U	GT3	A	4	4	other	teacher	home	mother
8	GP	M	15	U	LE3	A	3	2	services	other	home	mother
9	GP	M	15	U	GT3	T	3	4	other	other	home	mother

In [36]: `square= lambda x:x*x`

In [37]: `square(20)`

Out[37]: 400

In []:

Filter and Map function

1.filter() creates a new iterable by applying a filtering function to each element of the input iterable, returning only the elements that satisfy a given condition.

2.map() applies a transformation function to each element of the input iterable, creating a new iterable with the transformed values.

In [43]:

```
df= pd.DataFrame({'Name': ['albin', 'harry', 'Chipmunks',"potter"], 'Age': [28, 10, 23, 32]})
df
```

Out[43]:

	Name	Age
0	albin	28
1	harry	10
2	Chipmunks	23
3	potter	32

In [47]:

```
filtered_df = df.filter( 'Age <= 23')
filtered_df
```

Out[47]:

0
1
2
3

In []:

In [2]:

```
lst= [1,2,23,24,3,34,37,39,58,59,60]
```

In [8]:

```
result = list(filter(lambda x: x %2 == 0, lst))
```

In [9]:

```
print(result)
```

```
[2, 24, 34, 58, 60]
```

In []:

map()

In [10]:

```
#It Returns a list of the results after applying the given function to each item of a
```

```
In [11]: incremented_value= list(map(lambda x: x+2, lst))
```

```
In [12]: incremented_value
```

```
Out[12]: [3, 4, 25, 26, 5, 36, 39, 41, 60, 61, 62]
```

```
In [ ]:
```