Module 5 - Extracting a specific column from a data frame

contributed by

Mr. Anik Paul
Intern, R Team, FOSSEE, IIT Bombay
Mathematics Department, IIT Bombay

Ms. Usha Viswanathan
Sr. Project Manager
FOSSEE, IIT Bombay

17 December 2022

Steps to extract a column from a data frame in Python

This module shows the procedure to extract a specific column from a data frame. In this module we shall extract a column from the data used in Module 3. Locate the data file from the working directory. Then follow the instructions given below.

Method- 1: Selecting columns by name

To extract a column, the name of the column is to be passed as an argument inside inverted commas in square brackets after the name of the data file. The command is shown below in Figure 1:

```
pfs=Agriculture_data['PFS']
print(pfs)
```

```
0         LOW
1         LOW
2         LOW
3         LOW
4         LOW
        ...
520    MEDIUM
521    MEDIUM
522    MEDIUM
523    MEDIUM
524    MEDIUM
Name: PFS, Length: 525, dtype: object
```

Figure 1: Extracting a column from the data file

Multiple columns can also be extracted by creating a list with the names of the columns that are to be extracted and passing this list as an argument inside square brackets. The process is shown below in Figure2:

```
columns=['PFS','KFS']
Agriculture_data[columns]
```

	PFS	KFS
0	LOW	MEDIUM
1	LOW	MEDIUM
2	LOW	MEDIUM
3	LOW	MEDIUM
4	LOW	MEDIUM
...	...	...
520	MEDIUM	MEDIUM
521	MEDIUM	MEDIUM
522	MEDIUM	MEDIUM

Figure 2: Extracting multiple columns from the data frame

Method- 2: Extracting columns based on their data types

Data frames can have columns with multiple data types. Columns having the same data type can be extracted using the *dtypes* method. By matching the columns that are of the same data type, the user will get a series of True/False. One can use the *values* method to get just the True/False values and not the index**.**

```
col=Agriculture_data.loc[:,(Agriculture_data.dtypes=='float64').values]
col.head()
```

	Area	Prod.	Prod./Area	SML	SMV	SDN	SDP	SDK
0	391.71	7342.12	18.74	6.84	17.20	91.67	61.93	23.83
1	391.71	7342.12	18.74	3.21	15.76	89.10	55.37	31.51
2	391.71	7342.12	18.74	4.86	13.83	91.66	67.61	29.96
3	391.71	7342.12	18.74	0.49	14.56	0.00	54.91	19.87
4	391.71	7342.12	18.74	5.49	21.54	100.00	56.73	23.33

Figure 3: Extracting columns based on their data types

Method- 3: Selecting columns based on their column name containing a sub-string:

If there are columns in a data frame that are having a similar substring in their column names, then these columns can be extracted following the process shown below in figure 4. Here the substring fetched is 'FS'.

```
col=Agriculture_data.loc[:,['FS' in i for i in Agriculture_data.columns]]
col.head()
```

	NFS	PFS	KFS
0	LOW	LOW	MEDIUM
1	LOW	LOW	MEDIUM
2	LOW	LOW	MEDIUM
3	LOW	LOW	MEDIUM
4	LOW	LOW	MEDIUM

Figure 4: Extraction of columns based on names with 'FS' substring

Method- 4: Selecting columns based on how their name starts with:

Columns with names that start with a certain substring can be extracted by using the *startswith* method. The substring is passed as an argument of the *startswith* method**.** The process is shown below in Figure 5.

```
Agriculture_data.loc[:,Agriculture_data.columns.str.startswith('PF')].head()
```

	PFS
0	LOW
1	LOW
2	LOW
3	LOW
4	LOW

Figure 5: Extraction of columns using the startswith method