loc & iloc_ Function

December 10, 2022

- Difference between loc() and iloc() in Pandas DataFrame
- Python loc() function
- Python iloc() function

[3]:
```
import pandas as pd
```

1 Creating a sample dataframe

[7]:
```
data = pd.DataFrame({'Brand': ['Maruti', 'Hyundai', 'Tata',
                               'Mahindra', 'Maruti', 'Hyundai',
                               'Renault', 'Tata', 'Maruti'],
                     'Year': [2012, 2014, 2011, 2015, 2012,
                              2016, 2014, 2018, 2019],
                     'Kms Driven': [50000, 30000, 60000,
                                    25000, 10000, 46000,
                                    31000, 15000, 12000],
                     'City': ['Gurgaon', 'Delhi', 'Mumbai',
                              'Delhi', 'Mumbai', 'Delhi',
                              'Mumbai', 'Chennai',  'Ghaziabad'],
                     'Mileage':  [28, 27, 25, 26, 28,
                                  29, 24, 21, 24]})
```

1.1 Displaying DataFrame

[10]:
```
display(data)  # Display Method 1
```

	Brand	Year	Kms Driven	City	Mileage
0	Maruti	2012	50000	Gurgaon	28
1	Hyundai	2014	30000	Delhi	27
2	Tata	2011	60000	Mumbai	25
3	Mahindra	2015	25000	Delhi	26
4	Maruti	2012	10000	Mumbai	28
5	Hyundai	2016	46000	Delhi	29
6	Renault	2014	31000	Mumbai	24
7	Tata	2018	15000	Chennai	21
8	Maruti	2019	12000	Ghaziabad	24

[11]:
```
print(data) # Display Method 2
```

```
      Brand  Year  Kms Driven       City  Mileage
0    Maruti  2012       50000    Gurgaon       28
1   Hyundai  2014       30000      Delhi       27
2      Tata  2011       60000     Mumbai       25
3  Mahindra  2015       25000      Delhi       26
4    Maruti  2012       10000     Mumbai       28
5   Hyundai  2016       46000      Delhi       29
6   Renault  2014       31000     Mumbai       24
7      Tata  2018       15000    Chennai       21
8    Maruti  2019       12000  Ghaziabad       24
```

[13]:
```
data # Display Method 3
```

[13]:

	Brand	Year	Kms Driven	City	Mileage
0	Maruti	2012	50000	Gurgaon	28
1	Hyundai	2014	30000	Delhi	27
2	Tata	2011	60000	Mumbai	25
3	Mahindra	2015	25000	Delhi	26
4	Maruti	2012	10000	Mumbai	28
5	Hyundai	2016	46000	Delhi	29
6	Renault	2014	31000	Mumbai	24
7	Tata	2018	15000	Chennai	21
8	Maruti	2019	12000	Ghaziabad	24

2 Python loc() function

- The loc() function is label based data selecting method which means that we have to pass the name of the row or column which we want to select. This method includes the last element of the range passed in it, unlike iloc(). loc() can accept the boolean data unlike iloc(). Many operations can be performed using the loc() method like
- It works on Row & Column Name not on index Number

Excercise

2.0.1 Selecting cars with brand 'Maruti'

[25]:
```
data.loc[(data.Brand == 'Maruti')]
```

[25]:

	Brand	Year	Kms Driven	City	Mileage
0	Maruti	2012	50000	Gurgaon	28
4	Maruti	2012	10000	Mumbai	28
8	Maruti	2019	12000	Ghaziabad	24

[26]:
```

```

[28]:
```
data
```

[28]:

	Brand	Year	Kms Driven	City	Mileage
0	Maruti	2012	50000	Gurgaon	28
1	Hyundai	2014	30000	Delhi	27
2	Tata	2011	60000	Mumbai	25
3	Mahindra	2015	25000	Delhi	26
4	Maruti	2012	10000	Mumbai	28
5	Hyundai	2016	46000	Delhi	29
6	Renault	2014	31000	Mumbai	24
7	Tata	2018	15000	Chennai	21
8	Maruti	2019	12000	Ghaziabad	24

2.0.2 selecting range of rows from 2 to 5

[29]:
```
display(data.loc[2: 5])  # 2 & 5 exactly row & column name/ not index number
```

	Brand	Year	Kms Driven	City	Mileage
2	Tata	2011	60000	Mumbai	25
3	Mahindra	2015	25000	Delhi	26
4	Maruti	2012	10000	Mumbai	28
5	Hyundai	2016	46000	Delhi	29

3 Python iloc() function

- The iloc() function is an indexed-based selecting method which means that we have to pass an integer index in the method to select a specific row/column. This method does not include the last element of the range passed in it unlike loc(). iloc() does not accept the boolean data unlike loc(). Operations performed using iloc() are:

3.1 Excercise

[33]:
```
data
```

[33]:

	Brand	Year	Kms Driven	City	Mileage
0	Maruti	2012	50000	Gurgaon	28
1	Hyundai	2014	30000	Delhi	27
2	Tata	2011	60000	Mumbai	25
3	Mahindra	2015	25000	Delhi	26
4	Maruti	2012	10000	Mumbai	28
5	Hyundai	2016	46000	Delhi	29
6	Renault	2014	31000	Mumbai	24
7	Tata	2018	15000	Chennai	21
8	Maruti	2019	12000	Ghaziabad	24

selecting 0th, 2th, 4th, and 7th index rows

[36]:
```
data.iloc[[0,2,4,7]] # exactly index position not row & column name
```

[36]:

	Brand	Year	Kms Driven	City	Mileage
0	Maruti	2012	50000	Gurgaon	28
2	Tata	2011	60000	Mumbai	25
4	Maruti	2012	10000	Mumbai	28
7	Tata	2018	15000	Chennai	21

3.1.1 # selecting rows from 1 to 4 and columns from 2 to 4

[35]:
```
data.iloc[1: 5, 2: 5]  # skip last index
```

[35]:

	Kms Driven	City	Mileage
1	30000	Delhi	27
2	60000	Mumbai	25
3	25000	Delhi	26
4	10000	Mumbai	28