TASK - 5

query() for filering

we can use it as

1.single condition filtering

2.multiple condition filtering

Syntax: DataFrame.query(expr, inplace=False, **kwargs)

Parameters:

expr: Expression in string form to filter data.

inplace: Make changes in the original data frame if True

kwargs: Other keyword arguments.

1.single condition filtering

In this example, dataframe has been filtered on multiple conditions. Before applying the query() method, the spaces in column names have been replaced with '_'.

In [3]:

```
import pandas as pd
```

In [9]:

```
euro_data=pd.read_csv('Euro_2012_stats_TEAM.csv')
euro_data
```

Out[9]:

	Team	Goals	Shots on target	Shots off target	Shooting Accuracy	% Goals-to-shots	Total shots (inc. Blocked)	Hit Woodwork	Penalty goals	Penaltin score
0	Croatia	4	13	12	51.9%	16.0%	32	0	0	
1	Czech Republic	4	13	18	41.9%	12.9%	39	0	0	
2	Denmark	4	10	10	50.0%	20.0%	27	1	0	
3	England	5	11	18	50.0%	17.2%	40	0	0	
4	France	3	22	24	37.9%	6.5%	65	1	0	
5	Germany	10	32	32	47.8%	15.6%	80	2	1	
6	Greece	5	8	18	30.7%	19.2%	32	1	1	
7	Italy	6	34	45	43.0%	7.5%	110	2	0	
8	Netherlands	2	12	36	25.0%	4.1%	60	2	0	
9	Poland	2	15	23	39.4%	5.2%	48	0	0	
10	Portugal	6	22	42	34.3%	9.3%	82	6	0	
11	Republic of Ireland	1	7	12	36.8%	5.2%	28	0	0	
12	Russia	5	9	31	22.5%	12.5%	59	2	0	
13	Spain	12	42	33	55.9%	16.0%	100	0	1	
14	Sweden	5	17	19	47.2%	13.8%	39	3	0	
15	Ukraine	2	7	26	21.2%	6.0%	38	0	0	

16 rows × 35 columns

In [10]:

```
euro_filtered=euro_data.loc[:,['Team','Goals','Yellow Cards','Red Cards']]
```

In [13]:

```
euro_filtered.columns = [column.replace(" ", "_") for column in euro_filtered.columns]
```

In [7]:

```
euro_filtered
```

Out[7]:

	Team	Goals	Yellow Cards	Red Cards
0	Croatia	4	9	0
1	Czech Republic	4	7	0
2	Denmark	4	4	0
3	England	5	5	0
4	France	3	6	0
5	Germany	10	4	0
6	Greece	5	9	1
7	Italy	6	16	0
8	Netherlands	2	5	0
9	Poland	2	7	1
10	Portugal	6	12	0
11	Republic of Ireland	1	6	1
12	Russia	5	6	0
13	Spain	12	11	0
14	Sweden	5	7	0
15	Ukraine	2	5	0

In [14]:

```
euro_filtered.query('Yellow_Cards >= 9',inplace=True)
euro_filtered
```

Out[14]:

	Team	Goals	Yellow_Cards	Red_Cards
0	Croatia	4	9	0
6	Greece	5	9	1
7	Italy	6	16	0
10	Portugal	6	12	0
13	Spain	12	11	0

2.Multiple conditions filtering

In this example, dataframe has been filtered on multiple conditions. Before applying the query() method, the spaces in column names have been replaced with '_'.

In [15]:

```
import pandas as pd
euro_filtered.query('Yellow_Cards >= 9' and 'Red_Cards == 1',inplace=True)
euro_filtered
```

Out[15]:

	Team	Goals	Yellow_Cards	Red_Cards
6	Greece	5	9	1

In [ ]:

```
```

In [ ]:

```
```