Ex2 - Filtering and Sorting Data

Step 1. Import the necessary libraries

In [1]:
```
import pandas as pd
```

Step 2. Read from Euro_2012_stats_TEAM.csv and assign it to a variable called euro12.

In [2]:
```
euro12 = pd.read_csv('Euro_2012_stats_TEAM.csv')
euro12
```

Out[2]:

	Team	Goals	Shots on target	Shots off target	Shooting Accuracy	% Goals-to-shots	Total shots (inc. Blocked)	Hit Woodwork	Penalty goals	Pen s
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

Step 3. Select only the "Goals" column.

In [3]:
```
euro12.Goals
```

Out[3]:
```
0      4
1      4
2      4
3      5
4      3
5     10
6      5
7      6
8      2
9      2
10     6
11     1
12     5
13    12
14     5
15     2
Name: Goals, dtype: int64
```

Step 4. How many team participated in the Euro2012?

In [4]:
```
euro12.shape[0]
```

Out[4]:
```
16
```

Step 5. What is the number of columns in the dataset?

In [5]:
```
euro12.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16 entries, 0 to 15
Data columns (total 35 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Team                         16 non-null     object 
 1   Goals                        16 non-null     int64  
 2   Shots on target              16 non-null     int64  
 3   Shots off target             16 non-null     int64  
 4   Shooting Accuracy            16 non-null     object 
 5   % Goals-to-shots             16 non-null     object 
 6   Total shots (inc. Blocked)   16 non-null     int64  
 7   Hit Woodwork                 16 non-null     int64  
 8   Penalty goals                16 non-null     int64  
 9   Penalties not scored         16 non-null     int64  
 10  Headed goals                 16 non-null     int64  
 11  Passes                       16 non-null     int64  
 12  Passes completed             16 non-null     int64  
 13  Passing Accuracy             16 non-null     object 
 14  Touches                      16 non-null     int64  
 15  Crosses                      16 non-null     int64  
 16  Dribbles                     16 non-null     int64  
 17  Corners Taken                16 non-null     int64  
 18  Tackles                      16 non-null     int64  
 19  Clearances                   16 non-null     int64  
 20  Interceptions                16 non-null     int64  
 21  Clearances off line          15 non-null     float64
 22  Clean Sheets                 16 non-null     int64  
 23  Blocks                       16 non-null     int64  
 24  Goals conceded               16 non-null     int64  
 25  Saves made                   16 non-null     int64  
 26  Saves-to-shots ratio         16 non-null     object 
 27  Fouls Won                    16 non-null     int64  
 28  Fouls Conceded               16 non-null     int64  
 29  Offsides                     16 non-null     int64  
 30  Yellow Cards                 16 non-null     int64  
 31  Red Cards                    16 non-null     int64  
 32  Subs on                      16 non-null     int64  
 33  Subs off                     16 non-null     int64  
 34  Players Used                 16 non-null     int64  
dtypes: float64(1), int64(29), object(5)
memory usage: 4.5+ KB
```

Step 6. View only the columns Team, Yellow Cards and Red Cards and assign them to a dataframe called discipline

In [6]:
```
# filter only giving the column names

discipline = euro12[['Team', 'Yellow Cards', 'Red Cards']]
discipline
```

Out[6]:

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

Step 7. Sort the teams by Yellow Cards

In [7]:
```
discipline.sort_values(by = 'Yellow Cards')
```

Out[7]:

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

Step 8. Calculate the mean Yellow Cards given per Team

In [8]:
```
discipline['Yellow Cards'].mean()
```

Out[8]:
```
7.4375
```

Step 9. Filter teams that scored more than 6 goals

In [9]:
```
euro12[euro12.Goals > 6]
```

Out[9]:

	Team	Goals	Shots on target	Shots off target	Shooting Accuracy	% Goals-to-shots	Total shots (inc. Blocked)	Hit Woodwork	Penalty goals	Penalti r scor
5	Germany	10	32	32	47.8%	15.6%	80	2	1	
13	Spain	12	42	33	55.9%	16.0%	100	0	1	

2 rows × 35 columns

Step 10. Select the teams that start with G

In [10]:
```
euro12[euro12.Team.str.startswith('G')]
```

Out[10]:

	Team	Goals	Shots on target	Shots off target	Shooting Accuracy	% Goals-to-shots	Total shots (inc. Blocked)	Hit Woodwork	Penalty goals	Penaltie nc score
5	Germany	10	32	32	47.8%	15.6%	80	2	1	
6	Greece	5	8	18	30.7%	19.2%	32	1	1	

2 rows × 35 columns

Step 11. Select the 5th to 10th rows

In [11]:
```
euro12.iloc[5:11]
```

Out[11]:

	Team	Goals	Shots on target	Shots off target	Shooting Accuracy	% Goals-to-shots	Total shots (inc. Blocked)	Hit Woodwork	Penalty goals	Pen s
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	

6 rows × 35 columns