```
#3.2 Perform Exploratory Data Analysis (EDA) to answer questionsTake Sports␣ ↪dataset that shows the results from NCAA basketball games from 1985 to 2016
#8)     Display all object columns in dataset(  iterate for loop on columns)
#9)     find all of the games where the winning team scored more than 150 points
#10) find out when the winning team scores more than 150 points and when the losing team scores below 100.
#11)  How many games were played in each season?
#12)  What is the average winning score for each team?
#13)  How many games were won at home, away, and at neutral locations?

import pandas as pd

data = pd.read_csv('BasketballDataset.csv')

data
```

	Season	DayNum	WTeamID	WScore	LTeamID	LScore	WLoc	NumOT
0	1985	20	1228	81	1328	64	N	0
1	1985	25	1106	77	1354	70	H	0
2	1985	25	1112	63	1223	56	H	0
3	1985	25	1165	70	1432	54	H	0
4	1985	25	1192	86	1447	74	H	0
...	...	...	...	...	...	...	...	...
161547	2019	132	1153	69	1222	57	N	0
161548	2019	132	1209	73	1426	64	N	0
161549	2019	132	1277	65	1276	60	N	0
161550	2019	132	1387	55	1382	53	N	0
161551	2019	132	1463	97	1217	85	H	0

```
[161552 rows x 8 columns]

data.reset_index(drop = True, inplace = True)
data
```

	Season	DayNum	WTeamID	WScore	LTeamID	LScore	WLoc	NumOT
0	1985	20	1228	81	1328	64	N	0
1	1985	25	1106	77	1354	70	H	0
2	1985	25	1112	63	1223	56	H	0
3	1985	25	1165	70	1432	54	H	0
4	1985	25	1192	86	1447	74	H	0
...	...	...	...	...	...	...	...	...
161547	2019	132	1153	69	1222	57	N	0
161548	2019	132	1209	73	1426	64	N	0
161549	2019	132	1277	65	1276	60	N	0
161550	2019	132	1387	55	1382	53	N	0
161551	2019	132	1463	97	1217	85	H	0

```
[161552 rows x 8 columns]
```

```
#Display all objects columns from dataset
for col in data.columns:
    if data[col].dtypes == "object":
        print(col)
```

```
WLoc
```

```
#All the games where winning is greater than 150
data[data['WScore'] > 150 ]
```

	Season	DayNum	WTeamID	WScore	LTeamID	LScore	WLoc	NumOT
5269	1986	75	1258	151	1109	107	H	0
12046	1988	40	1328	152	1147	84	H	0
12355	1988	52	1328	151	1173	99	N	0
16040	1989	40	1328	152	1331	122	H	0
16853	1989	68	1258	162	1109	144	A	0
17867	1989	92	1258	181	1109	150	H	0
19653	1990	30	1328	173	1109	101	H	0
19971	1990	38	1258	152	1109	137	A	0
20022	1990	40	1116	166	1109	101	H	0
22145	1990	97	1258	157	1362	115	H	0
23582	1991	26	1318	152	1258	123	N	0
24341	1991	47	1328	172	1258	112	H	0
24970	1991	68	1258	186	1109	140	H	0
25656	1991	84	1106	151	1212	97	H	0
28687	1992	54	1261	159	1319	86	H	0
35023	1993	112	1380	155	1341	91	A	0
40060	1995	32	1375	156	1341	114	H	0
52600	1998	33	1395	153	1410	87	H	0

```
#games where wining team score >150  and losing team score<100
data[(data['WScore'] > 150 ) & (data['LScore'] < 100)]
```

	Season	DayNum	WTeamID	WScore	LTeamID	LScore	WLoc	NumOT
12046	1988	40	1328	152	1147	84	H	0
12355	1988	52	1328	151	1173	99	N	0
25656	1991	84	1106	151	1212	97	H	0
28687	1992	54	1261	159	1319	86	H	0
35023	1993	112	1380	155	1341	91	A	0
52600	1998	33	1395	153	1410	87	H	0

```
#Number of matchs played in each season
data['Season'] .value_counts()
```

```
Season
2019    5463
2018    5405
2017    5395
2016    5369
2014    5362
2015    5354
```

```
2013    5320
2010    5263
2012    5253
2009    5249
2011    5246
2008    5163
2007    5043
2006    4757
2005    4675
2003    4616
2004    4571
2002    4555
2000    4519
2001    4467
1999    4222
1998    4167
1997    4155
1992    4127
1991    4123
1996    4122
1995    4077
1994    4060
1990    4045
1989    4037
1993    3982
1988    3955
1987    3915
1986    3783
1985    3737
Name: count, dtype: int64
```

```
#Average wiining score for each teams ### important remember doubt
data.groupby("WTeamID")["WScore"].mean()
```

```
WTeamID
1101    75.500000
1102    70.675439
1103    75.989492
1104    75.834586
1105    74.788360
          ...
1462    80.028075
1463    72.997758
1464    74.514469
1465    80.083333
1466    71.000000
Name: WScore, Length: 366, dtype: float64
```

```
#Number of loactaions Home,awaay and neutral
data['WLoc'].value_counts()
```

```
WLoc
H    95878
A    49260
N    16414
Name: count, dtype: int64
```