

```
In [1]: #import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [6]: df=pd.read_csv("championsdata.csv")
df
```

Out[6]:

	Year	Team	Game	Win	Home	MP	FG	FGA	FGP	TP	...	FTP	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
0	1980	Lakers	1	1	1	240	48	89	0.539	0	...	0.867	12	31	43	30	5	9	17	24	109
1	1980	Lakers	2	0	1	240	48	95	0.505	0	...	0.667	15	37	52	32	12	7	26	27	104
2	1980	Lakers	3	1	0	240	44	92	0.478	0	...	0.767	22	34	56	20	5	5	20	25	111
3	1980	Lakers	4	0	0	240	44	93	0.473	0	...	0.737	18	31	49	23	12	6	19	22	102
4	1980	Lakers	5	1	1	240	41	91	0.451	0	...	0.788	19	37	56	28	7	6	21	27	108
...	
215	2017	Warriors	5	1	1	240	46	90	0.511	14	...	0.821	13	29	42	27	8	2	13	24	129
216	2018	Warriors	1	1	1	265	46	90	0.511	13	...	0.950	4	34	38	31	10	6	7	18	124
217	2018	Warriors	2	1	1	240	47	82	0.573	15	...	0.619	7	34	41	28	3	8	12	25	122
218	2018	Warriors	3	1	0	240	42	81	0.519	9	...	0.895	6	31	37	27	6	5	10	20	110
219	2018	Warriors	4	1	0	240	39	86	0.453	14	...	1.000	10	34	44	25	7	13	8	24	108

220 rows × 24 columns

```
In [7]: df.info()#The info method prints information about the DataFrame
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 220 entries, 0 to 219
Data columns (total 24 columns):
 #   Column  Non-Null Count  Dtype  
--- 
 0   Year     220 non-null   int64  
 1   Team    220 non-null   object  
 2   Game    220 non-null   int64  
 3   Win     220 non-null   int64  
 4   Home    220 non-null   int64  
 5   MP      220 non-null   int64  
 6   FG      220 non-null   int64  
 7   FGA     220 non-null   int64  
 8   FGP     220 non-null   float64 
 9   TP      220 non-null   int64  
 10  TPA     220 non-null   int64  
 11  TPP     214 non-null   float64 
 12  FT      220 non-null   int64  
 13  FTA     220 non-null   int64  
 14  FTP     220 non-null   float64 
 15  ORB     220 non-null   int64  
 16  DRB     220 non-null   int64  
 17  TRB     220 non-null   int64  
 18  AST     220 non-null   int64  
 19  STL     220 non-null   int64  
 20  BLK     220 non-null   int64  
 21  TOV     220 non-null   int64  
 22  PF      220 non-null   int64  
 23  PTS     220 non-null   int64  
dtypes: float64(3), int64(20), object(1)
memory usage: 41.4+ KB
```

In [8]: df.describe()#Data Description works only on numeric data

	Year	Game	Win	Home	MP	FG	FGA	FGP	TP	TPA	...	FTP	ORB
count	220.000000	220.000000	220.000000	220.000000	220.000000	220.000000	220.000000	220.000000	220.000000	220.000000	...	220.000000	220.000000
mean	1998.863636	3.400000	0.709091	0.504545	242.386364	37.750000	80.877273	0.466523	5.354545	14.604545	...	0.735568	12.295455
std	11.310865	1.734422	0.455217	0.501120	8.445949	6.323563	9.511753	0.054264	4.035428	9.420321	...	0.105555	4.630549
min	1980.000000	1.000000	0.000000	0.000000	240.000000	25.000000	62.000000	0.289000	0.000000	0.000000	...	0.368000	3.000000
25%	1989.000000	2.000000	0.000000	0.000000	240.000000	33.000000	75.000000	0.429750	2.000000	6.750000	...	0.667000	9.000000
50%	1999.000000	3.000000	1.000000	1.000000	240.000000	37.000000	80.000000	0.467000	5.000000	15.000000	...	0.740000	12.000000
75%	2009.000000	5.000000	1.000000	1.000000	240.000000	42.000000	87.000000	0.500000	8.000000	20.000000	...	0.815750	15.000000
max	2018.000000	7.000000	1.000000	1.000000	315.000000	56.000000	130.000000	0.617000	18.000000	43.000000	...	1.000000	27.000000

8 rows × 23 columns

In [9]: df.head()#The head() function is used to get the first 5 rows

	Year	Team	Game	Win	Home	MP	FG	FGA	FGP	TP	...	FTP	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
0	1980	Lakers	1	1	1	240	48	89	0.539	0	...	0.867	12	31	43	30	5	9	17	24	109
1	1980	Lakers	2	0	1	240	48	95	0.505	0	...	0.667	15	37	52	32	12	7	26	27	104
2	1980	Lakers	3	1	0	240	44	92	0.478	0	...	0.767	22	34	56	20	5	5	20	25	111
3	1980	Lakers	4	0	0	240	44	93	0.473	0	...	0.737	18	31	49	23	12	6	19	22	102
4	1980	Lakers	5	1	1	240	41	91	0.451	0	...	0.788	19	37	56	28	7	6	21	27	108

5 rows × 24 columns

In [10]: df.tail()#The tail() function is used to get the last 5 rows

	Year	Team	Game	Win	Home	MP	FG	FGA	FGP	TP	...	FTP	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
215	2017	Warriors	5	1	1	240	46	90	0.511	14	...	0.821	13	29	42	27	8	2	13	24	129
216	2018	Warriors	1	1	1	265	46	90	0.511	13	...	0.950	4	34	38	31	10	6	7	18	124
217	2018	Warriors	2	1	1	240	47	82	0.573	15	...	0.619	7	34	41	28	3	8	12	25	122
218	2018	Warriors	3	1	0	240	42	81	0.519	9	...	0.895	6	31	37	27	6	5	10	20	110
219	2018	Warriors	4	1	0	240	39	86	0.453	14	...	1.000	10	34	44	25	7	13	8	24	108

5 rows × 24 columns

In [11]: df.isnull().sum()#check is there any null value

```
Out[11]: Year      0
Team      0
Game      0
Win       0
Home      0
MP        0
FG        0
FGA       0
FGP       0
TP        0
TPA       0
TPP       6
FT        0
FTA       0
FTP       0
ORB       0
DRB       0
TRB       0
AST       0
STL       0
BLK       0
TOV       0
PF        0
PTS       0
dtype: int64
```

In [13]: win=df['Win'].sum()
win

Out[13]: 156

```
In [15]: Freethrowpercentage=df['FTP'].sum()
Freethrowpercentage
```

Out[15]: 161.825

```
In [100]: df[df.Year>2000]
```

Out[100]:

	Year	Team	Game	Win	Home	MP	FG	FGA	FGP	TP	...	FTP	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
120	2001	Lakers	1	0	1	265	40	90	0.444	6	...	0.556	16	28	44	23	14	9	19	22	101
121	2001	Lakers	2	1	1	240	38	81	0.469	4	...	0.692	15	37	52	29	7	13	16	24	98
122	2001	Lakers	3	1	0	240	35	75	0.467	4	...	0.880	10	30	40	18	8	6	13	26	96
123	2001	Lakers	4	1	0	240	36	72	0.500	10	...	0.563	12	31	43	24	6	6	14	22	100
124	2001	Lakers	5	1	0	240	32	71	0.451	12	...	0.711	14	33	47	21	6	10	12	22	108
...	
215	2017	Warriors	5	1	1	240	46	90	0.511	14	...	0.821	13	29	42	27	8	2	13	24	129
216	2018	Warriors	1	1	1	265	46	90	0.511	13	...	0.950	4	34	38	31	10	6	7	18	124
217	2018	Warriors	2	1	1	240	47	82	0.573	15	...	0.619	7	34	41	28	3	8	12	25	122
218	2018	Warriors	3	1	0	240	42	81	0.519	9	...	0.895	6	31	37	27	6	5	10	20	110
219	2018	Warriors	4	1	0	240	39	86	0.453	14	...	1.000	10	34	44	25	7	13	8	24	108

100 rows × 24 columns

```
In [32]: df.Team=='Lakers'
```

```
Out[32]: 0      True
1      True
2      True
3      True
4      True
...
215    False
216    False
217    False
218    False
219    False
Name: Team, Length: 220, dtype: bool
```

```
In [33]: df[df.Team=="Lakers"]
```

Out[33]:

	Year	Team	Game	Win	Home	MP	FG	FGA	FGP	TP	...	FTP	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
0	1980	Lakers	1	1	1	240	48	89	0.539	0	...	0.867	12	31	43	30	5	9	17	24	109
1	1980	Lakers	2	0	1	240	48	95	0.505	0	...	0.667	15	37	52	32	12	7	26	27	104
2	1980	Lakers	3	1	0	240	44	92	0.478	0	...	0.767	22	34	56	20	5	5	20	25	111
3	1980	Lakers	4	0	0	240	44	93	0.473	0	...	0.737	18	31	49	23	12	6	19	22	102
4	1980	Lakers	5	1	1	240	41	91	0.451	0	...	0.788	19	37	56	28	7	6	21	27	108
5	1980	Lakers	6	1	0	240	45	92	0.489	0	...	0.943	17	35	52	27	14	4	17	22	123
12	1982	Lakers	1	1	0	240	49	93	0.527	0	...	0.743	19	31	50	34	11	7	22	26	124
13	1982	Lakers	2	0	0	240	35	83	0.422	0	...	0.649	17	22	39	25	11	6	18	21	94
14	1982	Lakers	3	1	1	240	50	91	0.549	1	...	0.596	17	31	48	30	15	5	18	30	129
15	1982	Lakers	4	1	1	240	45	97	0.464	0	...	0.724	16	33	49	35	10	4	12	21	111
16	1982	Lakers	5	0	0	240	47	100	0.470	0	...	0.500	26	23	49	31	5	9	24	29	102
17	1982	Lakers	6	1	1	240	47	87	0.540	0	...	0.800	15	34	49	33	12	11	22	26	114
29	1985	Lakers	1	0	0	240	49	100	0.490	2	...	0.667	8	27	35	28	9	4	12	23	114
30	1985	Lakers	2	1	0	240	44	92	0.478	1	...	0.690	14	35	49	31	9	5	16	32	109
31	1985	Lakers	3	1	1	240	52	96	0.542	4	...	0.778	13	36	49	34	12	6	12	28	136
32	1985	Lakers	4	0	1	240	40	83	0.482	1	...	0.828	9	31	40	32	7	0	14	24	105
33	1985	Lakers	5	1	1	240	50	88	0.568	0	...	0.741	10	29	39	40	8	7	9	19	120
34	1985	Lakers	6	1	0	240	43	84	0.512	0	...	0.694	12	32	44	27	10	2	12	26	111
41	1987	Lakers	1	1	1	240	55	99	0.556	1	...	0.789	17	30	47	32	10	6	13	24	126
42	1987	Lakers	2	1	1	240	56	91	0.615	6	...	0.719	7	26	33	44	8	8	10	25	141
43	1987	Lakers	3	0	0	240	40	81	0.494	3	...	0.741	8	24	32	18	8	4	10	24	103
44	1987	Lakers	4	1	0	240	41	85	0.482	2	...	0.719	15	31	46	17	4	4	11	16	107
45	1987	Lakers	5	0	0	240	43	95	0.453	4	...	0.818	17	23	40	16	7	4	12	26	108
46	1987	Lakers	6	1	1	240	45	93	0.484	0	...	0.696	13	31	44	33	11	9	11	28	106
47	1988	Lakers	1	0	1	240	33	83	0.398	2	...	0.926	13	22	35	21	5	0	10	18	93
48	1988	Lakers	2	1	1	240	35	77	0.455	3	...	0.761	8	35	43	27	6	5	9	25	108
49	1988	Lakers	3	1	0	240	37	72	0.514	1	...	0.706	5	33	38	21	8	6	11	13	99
50	1988	Lakers	4	0	0	240	29	72	0.403	0	...	0.757	8	26	34	11	3	3	16	30	86
51	1988	Lakers	5	0	0	240	37	78	0.474	1	...	0.576	15	16	31	26	7	3	14	27	94
52	1988	Lakers	6	1	1	240	34	72	0.472	0	...	0.814	13	28	41	30	6	3	14	19	103
53	1988	Lakers	7	1	1	240	43	77	0.558	3	...	0.704	10	31	41	30	6	1	18	22	108
114	2000	Lakers	1	1	1	240	45	88	0.511	3	...	0.579	14	34	48	25	9	6	11	19	104
115	2000	Lakers	2	1	1	240	36	75	0.480	7	...	0.561	11	36	47	29	3	8	9	26	111
116	2000	Lakers	3	0	0	240	38	76	0.500	7	...	0.421	8	25	33	23	9	4	16	27	91
117	2000	Lakers	4	1	0	265	48	93	0.516	4	...	0.667	14	28	42	20	6	4	12	31	120
118	2000	Lakers	5	0	0	240	36	90	0.400	4	...	0.524	13	21	34	18	9	2	12	33	87
119	2000	Lakers	6	1	1	240	43	90	0.478	10	...	0.606	13	31	44	25	5	7	5	24	116
120	2001	Lakers	1	0	1	265	40	90	0.444	6	...	0.556	16	28	44	23	14	9	19	22	101
121	2001	Lakers	2	1	1	240	38	81	0.469	4	...	0.692	15	37	52	29	7	13	16	24	98
122	2001	Lakers	3	1	0	240	35	75	0.467	4	...	0.880	10	30	40	18	8	6	13	26	96
123	2001	Lakers	4	1	0	240	36	72	0.500	10	...	0.563	12	31	43	24	6	6	14	22	100
124	2001	Lakers	5	1	0	240	32	71	0.451	12	...	0.711	14	33	47	21	6	10	12	22	108
125	2002	Lakers	1	1	1	240	33	72	0.458	1	...	0.711	17	33	50	21	8	8	16	20	99
126	2002	Lakers	2	1	1	240	39	78	0.500	9	...	0.792	12	35	47	26	6	7	16	21	106
127	2002	Lakers	3	1	0	240	37	68	0.544	8	...	0.686	7	33	40	17	7	10	19	22	106
128	2002	Lakers	4	1	0	240	37	71	0.521	11	...	0.757	7	32	39	28	7	3	9	15	113
163	2009	Lakers	1	1	1	240	41	89	0.461	3	...	0.833	15	40	55	18	4	7	8	23	100
164	2009	Lakers	2	1	1	265	36	78	0.462	5	...	0.857	4	31	35	20	12	6	12	25	101
165	2009	Lakers	3	0	0	240	40	78	0.513	8	...	0.615	11	16	27	16	6	3	13	25	104
166	2009	Lakers	4	1	0	265	38	91	0.418	8	...	0.750	10	29	39	15	8	2	7	28	99
167	2009	Lakers	5	1	0	240	35	80	0.438	8	...	0.750	13	34	47	13	6	8	10	20	99
168	2010	Lakers	1	1	1	240	37	76	0.487	4	...	0.774	12	30	42	18	6	7	12	26	102
169	2010	Lakers	2	0	1	240	29	71	0.408	5	...	0.756	10	29	39	18	8	14	15	29	94
170	2010	Lakers	3	1	0	240	34	76	0.447	2	...	0.875	11	32	43	13	2	7	8	20	91
171	2010	Lakers	4	0	0	240	32	71	0.451	7	...	0.818	8	26	34	13	6	3	15	23	89

Year	Team	Game	Win	Home	MP	FG	FGA	FGP	TP	...	FTP	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	
172	2010	Lakers	5	0	0	240	31	78	0.397	7	...	0.654	16	18	34	12	9	1	13	22	86
173	2010	Lakers	6	1	1	240	33	79	0.418	6	...	0.895	12	40	52	17	13	8	13	17	89
174	2010	Lakers	7	1	1	240	27	83	0.325	4	...	0.676	23	30	53	11	7	3	11	19	83

58 rows × 24 columns

In [21]: df[df.Year>2015]

Year	Team	Game	Win	Home	MP	FG	FGA	FGP	TP	...	FTP	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	
204	2016	Cavaliers	1	0	0	240	32	84	0.381	7	...	0.900	15	32	47	17	7	4	15	17	89
205	2016	Cavaliers	2	0	0	240	28	79	0.354	5	...	0.667	9	25	34	15	15	3	17	19	77
206	2016	Cavaliers	3	1	1	240	48	91	0.527	12	...	0.706	17	35	52	23	8	3	13	25	120
207	2016	Cavaliers	4	0	1	240	38	81	0.469	6	...	0.577	16	24	40	15	5	6	11	24	97
208	2016	Cavaliers	5	1	0	240	44	83	0.530	10	...	0.609	8	33	41	15	11	9	16	22	112
209	2016	Cavaliers	6	1	1	240	40	77	0.519	10	...	0.781	8	37	45	24	12	7	10	25	115
210	2016	Cavaliers	7	1	0	240	33	82	0.402	6	...	0.840	9	39	48	17	7	6	11	15	93
211	2017	Warriors	1	1	1	240	45	106	0.425	12	...	0.688	14	36	50	31	12	3	4	24	113
212	2017	Warriors	2	1	1	240	46	89	0.517	18	...	0.917	10	43	53	34	5	7	20	19	132
213	2017	Warriors	3	1	0	240	40	83	0.482	16	...	0.917	8	36	44	29	8	4	18	28	118
214	2017	Warriors	4	0	0	240	39	87	0.448	11	...	0.750	16	24	40	26	5	6	12	27	116
215	2017	Warriors	5	1	1	240	46	90	0.511	14	...	0.821	13	29	42	27	8	2	13	24	129
216	2018	Warriors	1	1	1	265	46	90	0.511	13	...	0.950	4	34	38	31	10	6	7	18	124
217	2018	Warriors	2	1	1	240	47	82	0.573	15	...	0.619	7	34	41	28	3	8	12	25	122
218	2018	Warriors	3	1	0	240	42	81	0.519	9	...	0.895	6	31	37	27	6	5	10	20	110
219	2018	Warriors	4	1	0	240	39	86	0.453	14	...	1.000	10	34	44	25	7	13	8	24	108

16 rows × 24 columns

In [25]: df[['Year','Team','Win','PTS']]

Year	Team	Win	PTS	
0	1980	Lakers	1	109
1	1980	Lakers	0	104
2	1980	Lakers	1	111
3	1980	Lakers	0	102
4	1980	Lakers	1	108
...	
215	2017	Warriors	1	129
216	2018	Warriors	1	124
217	2018	Warriors	1	122
218	2018	Warriors	1	110
219	2018	Warriors	1	108

220 rows × 4 columns

In [40]: df.iloc[168:]

Out[40]:

	Year	Team	Game	Win	Home	MP	FG	FGA	FGP	TP	...	FTP	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
168	2010	Lakers	1	1	1	240	37	76	0.487	4	...	0.774	12	30	42	18	6	7	12	26	102
169	2010	Lakers	2	0	1	240	29	71	0.408	5	...	0.756	10	29	39	18	8	14	15	29	94
170	2010	Lakers	3	1	0	240	34	76	0.447	2	...	0.875	11	32	43	13	2	7	8	20	91
171	2010	Lakers	4	0	0	240	32	71	0.451	7	...	0.818	8	26	34	13	6	3	15	23	89
172	2010	Lakers	5	0	0	240	31	78	0.397	7	...	0.654	16	18	34	12	9	1	13	22	86
173	2010	Lakers	6	1	1	240	33	79	0.418	6	...	0.895	12	40	52	17	13	8	13	17	89
174	2010	Lakers	7	1	1	240	27	83	0.325	4	...	0.676	23	30	53	11	7	3	11	19	83
175	2011	Mavericks	1	0	0	240	25	67	0.373	9	...	0.781	6	30	36	18	6	8	11	22	84
176	2011	Mavericks	2	1	0	240	36	75	0.480	6	...	0.810	11	30	41	18	8	2	18	20	95
177	2011	Mavericks	3	0	1	240	28	70	0.400	8	...	0.815	12	30	42	18	3	8	14	14	86
178	2011	Mavericks	4	1	1	240	29	73	0.397	4	...	0.800	12	29	41	13	7	2	11	18	86
179	2011	Mavericks	5	1	1	240	39	69	0.565	13	...	0.778	4	22	26	23	8	3	11	20	112
180	2011	Mavericks	6	1	0	240	41	82	0.500	11	...	0.667	10	30	40	19	11	1	14	24	105
181	2012	Heat	1	0	0	240	36	78	0.462	8	...	0.778	7	28	35	20	6	1	10	19	94
182	2012	Heat	2	1	0	240	36	76	0.474	6	...	0.880	11	29	40	13	5	4	13	21	100
183	2012	Heat	3	1	1	240	28	74	0.378	4	...	0.886	14	31	45	13	6	5	12	19	91
184	2012	Heat	4	1	1	240	38	79	0.481	10	...	0.720	9	31	40	19	8	2	9	18	104
185	2012	Heat	5	1	1	240	40	77	0.519	14	...	0.818	8	33	41	25	8	7	13	21	121
186	2012	'Heat'	1	0	1	240	34	78	0.436	8	...	0.706	9	37	46	20	4	2	8	12	88
187	2013	Heat	2	1	1	240	41	83	0.494	10	...	0.786	9	27	36	22	9	6	6	17	103
188	2013	Heat	3	0	0	240	31	76	0.408	8	...	0.700	9	27	36	21	9	8	16	21	77
189	2013	Heat	4	1	0	240	45	85	0.529	4	...	0.882	7	34	41	23	13	7	9	26	109
190	2013	Heat	5	0	0	240	37	86	0.430	11	...	0.826	12	22	34	25	8	3	13	24	104
191	2013	Heat	6	1	1	265	38	81	0.469	11	...	0.762	12	30	42	23	10	6	15	26	103
192	2013	Heat	7	1	1	240	36	82	0.439	12	...	0.688	11	32	43	14	8	4	16	19	95
193	2014	Spurs	1	1	1	240	40	68	0.588	13	...	0.773	5	34	39	30	8	4	22	14	110
194	2014	Spurs	2	0	1	240	36	82	0.439	12	...	0.600	11	26	37	26	7	1	11	20	96
195	2014	Spurs	3	1	0	240	38	64	0.594	9	...	0.813	5	24	29	21	12	4	12	25	111
196	2014	Spurs	4	1	0	240	40	70	0.571	9	...	0.720	12	32	44	25	8	4	14	19	107
197	2014	Spurs	5	1	1	240	37	78	0.474	12	...	0.783	6	34	40	25	5	4	8	21	104
198	2015	Warriors	1	1	1	265	39	88	0.443	10	...	0.909	11	37	48	24	8	5	12	16	108
199	2015	Warriors	2	0	1	265	33	83	0.398	8	...	0.760	10	35	45	16	11	7	18	31	93
200	2015	Warriors	3	0	0	240	36	90	0.400	12	...	0.583	18	28	46	21	6	7	14	25	91
201	2015	Warriors	4	1	0	240	36	77	0.468	12	...	0.704	6	38	44	24	5	5	7	21	103
202	2015	Warriors	5	1	1	240	36	75	0.480	12	...	0.588	11	32	43	25	7	2	16	25	104
203	2015	Warriors	6	1	0	240	37	85	0.435	13	...	0.621	7	32	39	28	11	4	9	27	105
204	2016	Cavaliers	1	0	0	240	32	84	0.381	7	...	0.900	15	32	47	17	7	4	15	17	89
205	2016	Cavaliers	2	0	0	240	28	79	0.354	5	...	0.667	9	25	34	15	15	3	17	19	77
206	2016	Cavaliers	3	1	1	240	48	91	0.527	12	...	0.706	17	35	52	23	8	3	13	25	120
207	2016	Cavaliers	4	0	1	240	38	81	0.469	6	...	0.577	16	24	40	15	5	6	11	24	97
208	2016	Cavaliers	5	1	0	240	44	83	0.530	10	...	0.609	8	33	41	15	11	9	16	22	112
209	2016	Cavaliers	6	1	1	240	40	77	0.519	10	...	0.781	8	37	45	24	12	7	10	25	115
210	2016	Cavaliers	7	1	0	240	33	82	0.402	6	...	0.840	9	39	48	17	7	6	11	15	93
211	2017	Warriors	1	1	1	240	45	106	0.425	12	...	0.688	14	36	50	31	12	3	4	24	113
212	2017	Warriors	2	1	1	240	46	89	0.517	18	...	0.917	10	43	53	34	5	7	20	19	132
213	2017	Warriors	3	1	0	240	40	83	0.482	16	...	0.917	8	36	44	29	8	4	18	28	118
214	2017	Warriors	4	0	0	240	39	87	0.448	11	...	0.750	16	24	40	26	5	6	12	27	116
215	2017	Warriors	5	1	1	240	46	90	0.511	14	...	0.821	13	29	42	27	8	2	13	24	129
216	2018	Warriors	1	1	1	265	46	90	0.511	13	...	0.950	4	34	38	31	10	6	7	18	124
217	2018	Warriors	2	1	1	240	47	82	0.573	15	...	0.619	7	34	41	28	3	8	12	25	122
218	2018	Warriors	3	1	0	240	42	81	0.519	9	...	0.895	6	31	37	27	6	5	10	20	110
219	2018	Warriors	4	1	0	240	39	86	0.453	14	...	1.000	10	34	44	25	7	13	8	24	108

52 rows × 24 columns

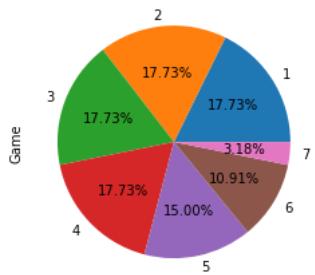
```
In [43]: df1=df[['Year','Team','Win','PTS']].iloc[168:]
df1
```

Out[43]:

	Year	Team	Win	PTS
168	2010	Lakers	1	102
169	2010	Lakers	0	94
170	2010	Lakers	1	91
171	2010	Lakers	0	89
172	2010	Lakers	0	86
173	2010	Lakers	1	89
174	2010	Lakers	1	83
175	2011	Mavericks	0	84
176	2011	Mavericks	1	95
177	2011	Mavericks	0	86
178	2011	Mavericks	1	86
179	2011	Mavericks	1	112
180	2011	Mavericks	1	105
181	2012	Heat	0	94
182	2012	Heat	1	100
183	2012	Heat	1	91
184	2012	Heat	1	104
185	2012	Heat	1	121
186	2012	'Heat'	0	88
187	2013	Heat	1	103
188	2013	Heat	0	77
189	2013	Heat	1	109
190	2013	Heat	0	104
191	2013	Heat	1	103
192	2013	Heat	1	95
193	2014	Spurs	1	110
194	2014	Spurs	0	96
195	2014	Spurs	1	111
196	2014	Spurs	1	107
197	2014	Spurs	1	104
198	2015	Warriors	1	108
199	2015	Warriors	0	93
200	2015	Warriors	0	91
201	2015	Warriors	1	103
202	2015	Warriors	1	104
203	2015	Warriors	1	105
204	2016	Cavaliers	0	89
205	2016	Cavaliers	0	77
206	2016	Cavaliers	1	120
207	2016	Cavaliers	0	97
208	2016	Cavaliers	1	112
209	2016	Cavaliers	1	115
210	2016	Cavaliers	1	93
211	2017	Warriors	1	113
212	2017	Warriors	1	132
213	2017	Warriors	1	118
214	2017	Warriors	0	116
215	2017	Warriors	1	129
216	2018	Warriors	1	124
217	2018	Warriors	1	122
218	2018	Warriors	1	110
219	2018	Warriors	1	108

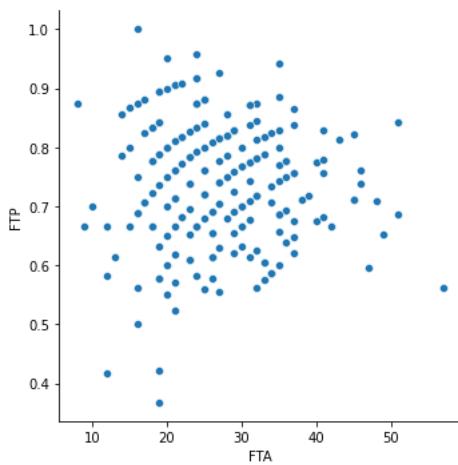
```
In [53]: df['Game'].value_counts().plot(kind='pie', autopct="%1.2f%%")
```

```
Out[53]: <AxesSubplot: ylabel='Game'>
```



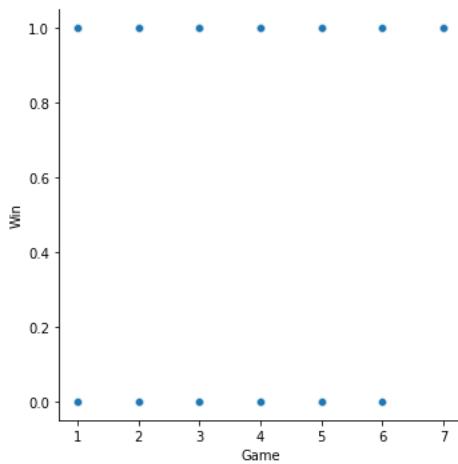
```
In [56]: sns.relplot(x='FTA',y='FTP',data=df)#FTA - Free throws attempted and FTP - Free throw percentage
```

```
Out[56]: <seaborn.axisgrid.FacetGrid at 0x2ae31454580>
```



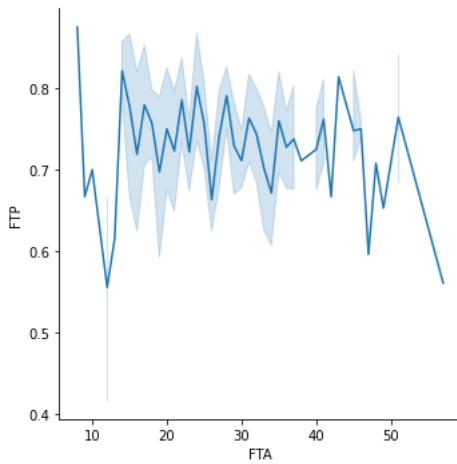
```
In [60]: sns.relplot(x='Game',y='Win',data=df)
```

```
Out[60]: <seaborn.axisgrid.FacetGrid at 0x2ae30d221f0>
```



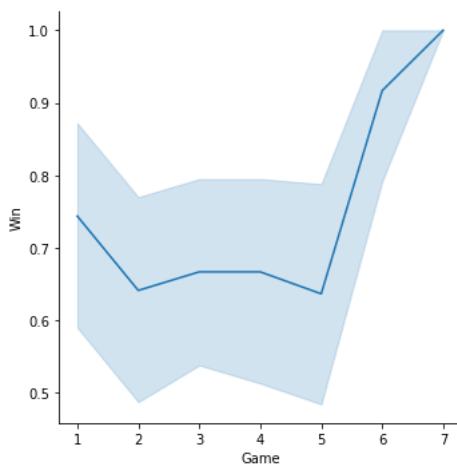
```
In [58]: sns.relplot(x='FTA',y='FTP',data=df,kind='line')
```

```
Out[58]: <seaborn.axisgrid.FacetGrid at 0x2ae340d5610>
```



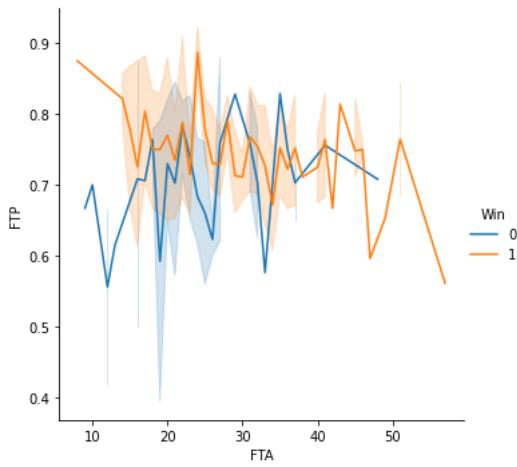
```
In [59]: sns.relplot(x='Game',y='Win',data=df,kind='line')
```

```
Out[59]: <seaborn.axisgrid.FacetGrid at 0x2ae341b7c40>
```



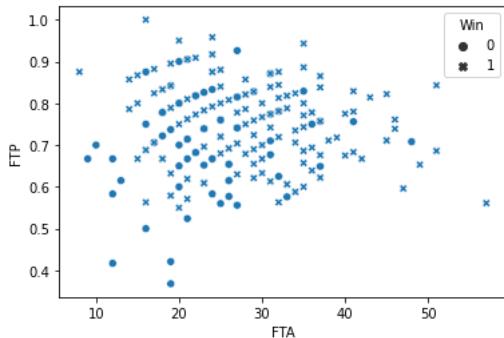
```
In [61]: sns.relplot(x='FTA',y='FTP',data=df,kind='line',hue='Win')
```

```
Out[61]: <seaborn.axisgrid.FacetGrid at 0x2ae31ecf070>
```



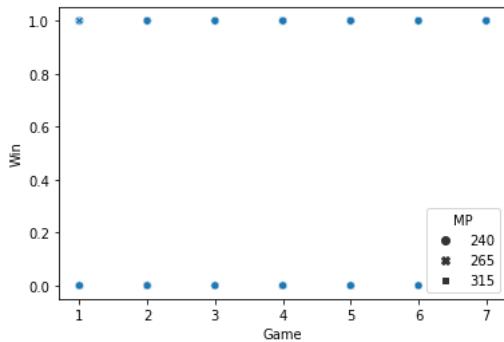
```
In [65]: sns.scatterplot(x='FTA',y='FTP',data=df,style='Win')
```

```
Out[65]: <AxesSubplot:xlabel='FTA', ylabel='FTP'>
```



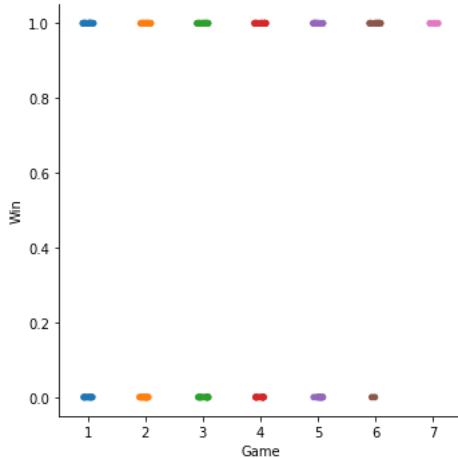
```
In [68]: sns.scatterplot(x='Game',y='Win',data=df,style='MP')
```

```
Out[68]: <AxesSubplot:xlabel='Game', ylabel='Win'>
```



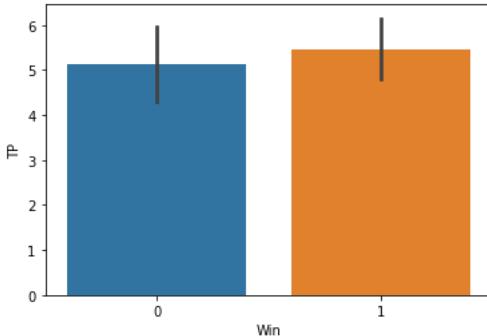
```
In [67]: sns.catplot(x='Game',y='Win',data=df)
```

```
Out[67]: <seaborn.axisgrid.FacetGrid at 0x2ae3480d3d0>
```



```
In [78]: sns.barplot(x='Win',y='TP',data=df)
```

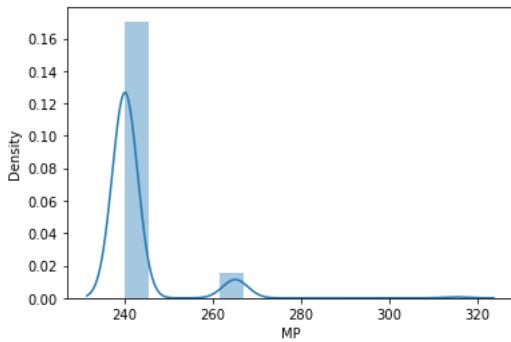
```
Out[78]: <AxesSubplot:xlabel='Win', ylabel='TP'>
```



```
In [79]: sns.distplot(df['MP'])
```

```
C:\Users\Administrator\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
```

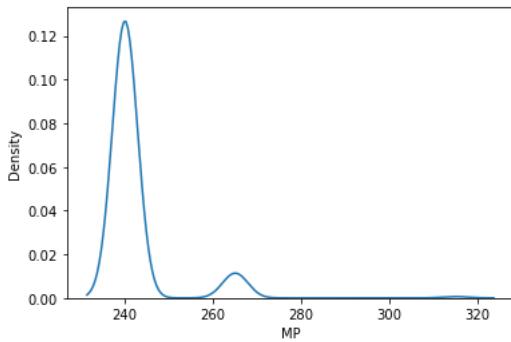
```
Out[79]: <AxesSubplot:xlabel='MP', ylabel='Density'>
```



```
In [80]: sns.distplot(df['MP'],kde=True,hist=False)
```

```
C:\Users\Administrator\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
warnings.warn(msg, FutureWarning)
```

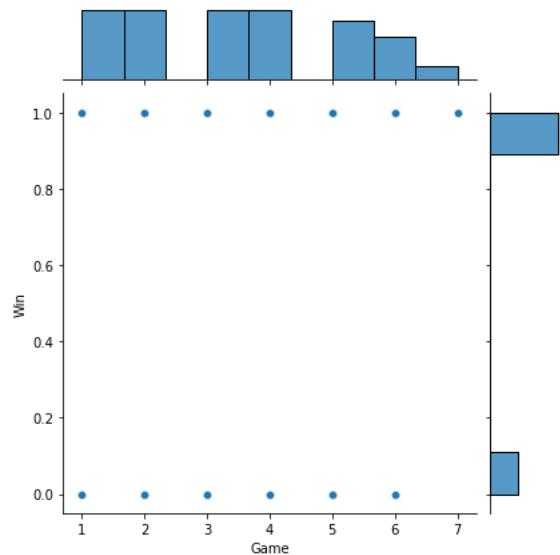
```
Out[80]: <AxesSubplot:xlabel='MP', ylabel='Density'>
```



```
In [81]: sns.jointplot(df['Game'],df['Win'])
```

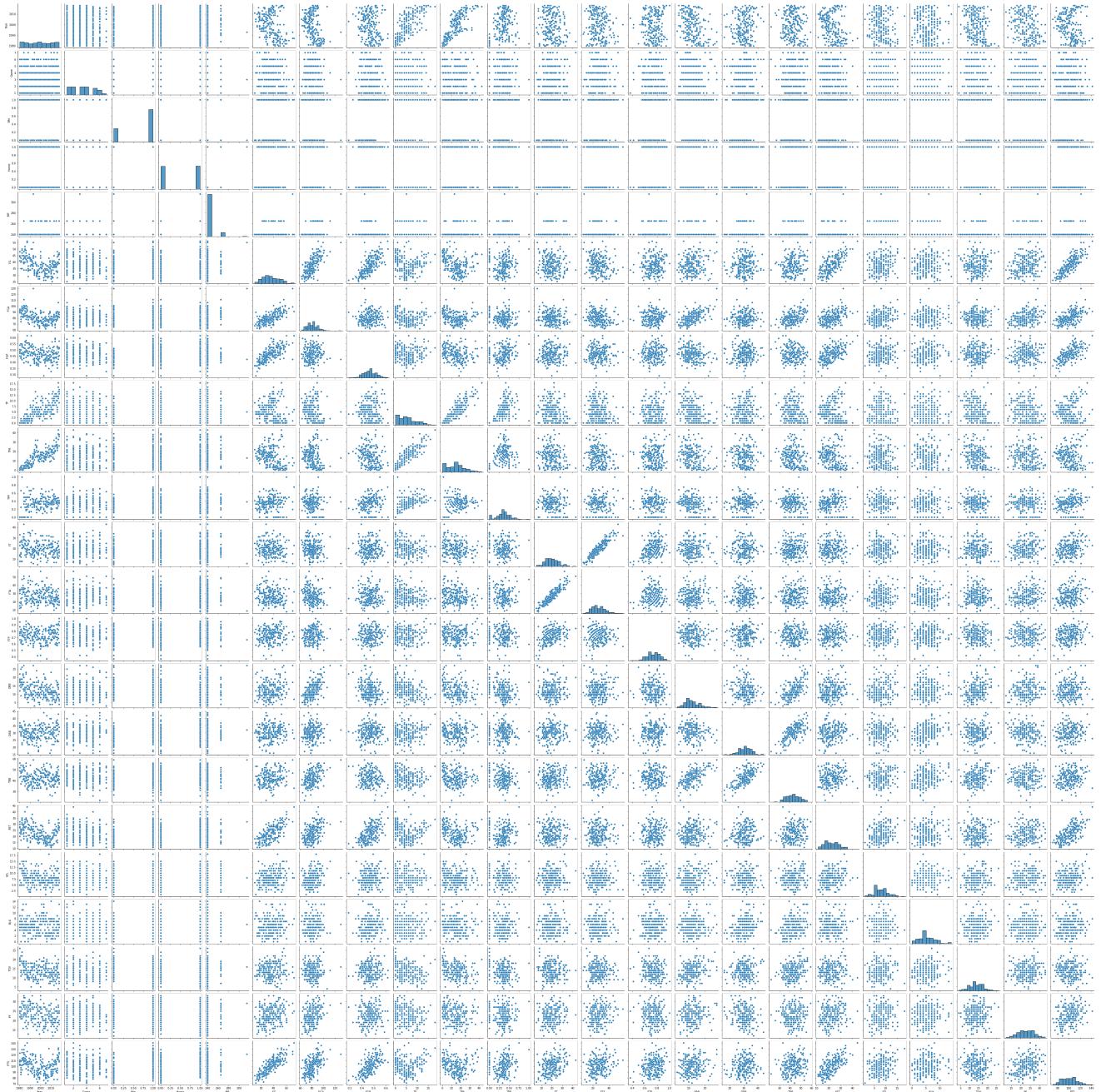
C:\Users\Administrator\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

```
Out[81]: <seaborn.axisgrid.JointGrid at 0x2ae3718f850>
```



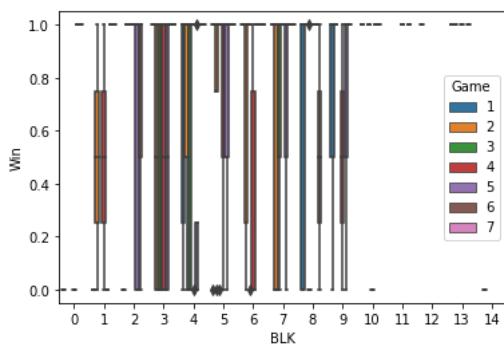
```
In [84]: sns.pairplot(df)
```

```
Out[84]: <seaborn.axisgrid.PairGrid at 0x2ae4f3e7dc0>
```



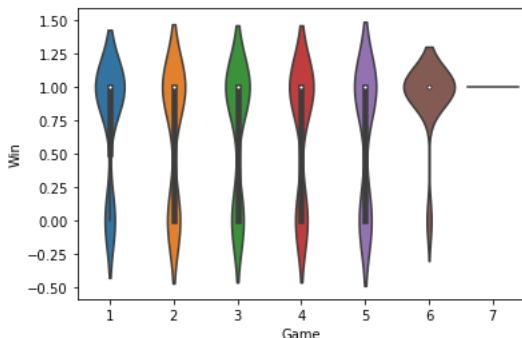
```
In [95]: sns.boxplot(x='BLK',y="Win",data=df,hue='Game')
```

```
Out[95]: <AxesSubplot:xlabel='BLK', ylabel='Win'>
```



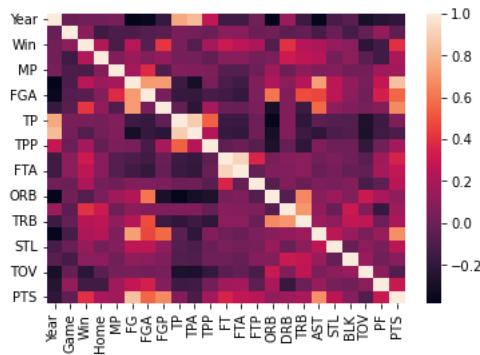
```
In [102]: sns.violinplot(x='Game',y='Win',data=df)
```

```
Out[102]: <AxesSubplot:xlabel='Game', ylabel='Win'>
```



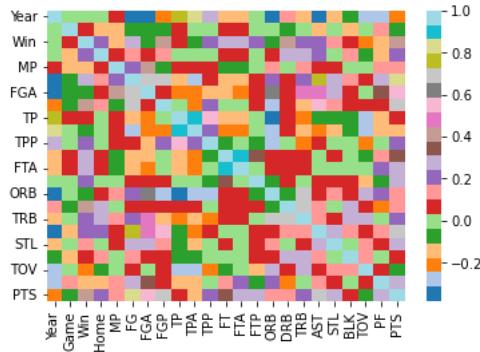
```
In [97]: z=df.corr()
sns.heatmap(z)
```

```
Out[97]: <AxesSubplot:>
```



```
In [98]: sns.heatmap(df.corr(),cmap='tab20')
```

```
Out[98]: <AxesSubplot:>
```



```
In [99]: sns.heatmap(df.corr(),cmap='tab20',annot=True,linewidth=2,linecolor='red')
```

```
Out[99]: <AxesSubplot:>
```

