

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('nbaallelo.csv')
```

```
df
```

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	\
0	1	194611010TRH	NBA	0	1947	11/17/1946	
1	1	194611010TRH	NBA	1	1947	11/1/1946	
2	2	194611020CHS	NBA	0	1947	11/2/1946	
3	2	194611020CHS	NBA	1	1947	11/2/1946	
4	3	194611020DTF	NBA	0	1947	11/2/1946	
...	
126309	63155	201506110CLE	NBA	0	2015	6/11/2015	
126310	63156	201506140GSW	NBA	0	2015	6/14/2015	
126311	63156	201506140GSW	NBA	1	2015	6/14/2015	
126312	63157	201506170CLE	NBA	0	2015	6/16/2015	
126313	63157	201506170CLE	NBA	1	2015	6/16/2015	

opp_id	seasongame	is_playoffs	team_id	fran_id	...	win_equiv
0	1	0	TRH	Huskies	...	40.294830
NYK						
1	1	0	NYK	Knicks	...	41.705170
TRH						
2	1	0	CHS	Stags	...	42.012257
NYK						
3	2	0	NYK	Knicks	...	40.692783
CHS						
4	1	0	DTF	Falcons	...	38.864048
WSC						
...
...						
126309	100	1	CLE	Cavaliers	...	60.309792
GSW						
126310	102	1	GSW	Warriors	...	68.013329
CLE						
126311	101	1	CLE	Cavaliers	...	60.010067
GSW						
126312	102	1	CLE	Cavaliers	...	59.290245
GSW						
126313	103	1	GSW	Warriors	...	68.519516
CLE						

game_result	opp_fran	opp_pts	opp_elo_i	opp_elo_n	game_location
0	Knicks	68	1300.0000	1306.7233	H
L					

1	Huskies	66	1300.0000	1293.2767	A
W					
2	Knicks	47	1306.7233	1297.0712	H
W					
3	Stags	63	1300.0000	1309.6521	A
L					
4	Capitols	50	1300.0000	1320.3811	H
L					
...
...					
126309	Warriors	103	1790.9591	1809.9791	H
L					
126310	Cavaliers	91	1704.3949	1700.7391	H
W					
126311	Warriors	104	1809.9791	1813.6349	A
L					
126312	Warriors	105	1813.6349	1822.2881	H
L					
126313	Cavaliers	97	1700.7391	1692.0859	A
W					

	forecast	notes
0	0.640065	NaN
1	0.359935	NaN
2	0.631101	NaN
3	0.368899	NaN
4	0.640065	NaN
...
126309	0.546572	NaN
126310	0.765565	NaN
126311	0.234435	NaN
126312	0.481450	NaN
126313	0.518550	NaN

[126314 rows x 23 columns]

df.dtypes

gameorder	int64
game_id	object
lg_id	object
_iscopy	int64
year_id	int64
date_game	object
seasongame	int64
is_playoffs	int64
team_id	object
fran_id	object
pts	int64
elo_i	float64

```

elo_n          float64
win_equiv      float64
opp_id         object
opp_fran       object
opp_pts        int64
opp_elo_i      float64
opp_elo_n      float64
game_location  object
game_result    object
forecast       float64
notes         object
dtype: object

```

```
df.describe()
```

	gameorder	_iscopy	year_id	seasongame	\
count	126314.000000	126314.000000	126314.000000	126314.000000	
mean	31579.000000	0.500000	1988.200374	43.533733	
std	18231.927643	0.500002	17.582309	25.375178	
min	1.000000	0.000000	1947.000000	1.000000	
25%	15790.000000	0.000000	1975.000000	22.000000	
50%	31579.000000	0.500000	1990.000000	43.000000	
75%	47368.000000	1.000000	2003.000000	65.000000	
max	63157.000000	1.000000	2015.000000	108.000000	

	is_playoffs	pts	elo_i	elo_n	\
count	126314.000000	126314.000000	126314.000000	126314.000000	
mean	0.063857	102.729982	1495.236055	1495.236055	
std	0.244499	14.814845	112.139945	112.461687	
min	0.000000	0.000000	1091.644500	1085.774400	
25%	0.000000	93.000000	1417.237975	1416.994900	
50%	0.000000	103.000000	1500.945550	1500.954400	
75%	0.000000	112.000000	1576.060000	1576.291625	
max	1.000000	186.000000	1853.104500	1853.104500	

	win_equiv	opp_pts	opp_elo_i	opp_elo_n	\
count	126314.000000	126314.000000	126314.000000	126314.000000	
mean	41.707889	102.729982	1495.236055	1495.236055	
std	10.627332	14.814845	112.139945	112.461687	
min	10.152501	0.000000	1091.644500	1085.774400	
25%	34.103035	93.000000	1417.237975	1416.994900	
50%	42.113357	103.000000	1500.945550	1500.954400	
75%	49.635328	112.000000	1576.060000	1576.291625	
max	71.112038	186.000000	1853.104500	1853.104500	

	forecast
count	126314.000000
mean	0.500000
std	0.215252
min	0.020447

```

25%      0.327989
50%      0.500000
75%      0.672011
max       0.979553

```

```
df.isnull().sum()
```

```

gameorder      0
game_id        0
lg_id          0
_iscopy        0
year_id        0
date_game      0
seasongame     0
is_playoffs    0
team_id        0
fran_id        0
pts            0
elo_i          0
elo_n          0
win_equiv      0
opp_id         0
opp_fran       0
opp_pts        0
opp_elo_i      0
opp_elo_n      0
game_location  0
game_result    0
forecast       0
notes          120890
dtype: int64

```

```
df = df.drop("notes",axis=1)
```

```
df.shape
```

```
(126314, 22)
```

```
df
```

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	\
0	1	194611010TRH	NBA	0	1947	11/1/1946	
1	1	194611010TRH	NBA	1	1947	11/1/1946	
2	2	194611020CHS	NBA	0	1947	11/2/1946	
3	2	194611020CHS	NBA	1	1947	11/2/1946	
4	3	194611020DTF	NBA	0	1947	11/2/1946	
...	
126309	63155	201506110CLE	NBA	0	2015	6/11/2015	
126310	63156	201506140GSW	NBA	0	2015	6/14/2015	
126311	63156	201506140GSW	NBA	1	2015	6/14/2015	
126312	63157	201506170CLE	NBA	0	2015	6/16/2015	

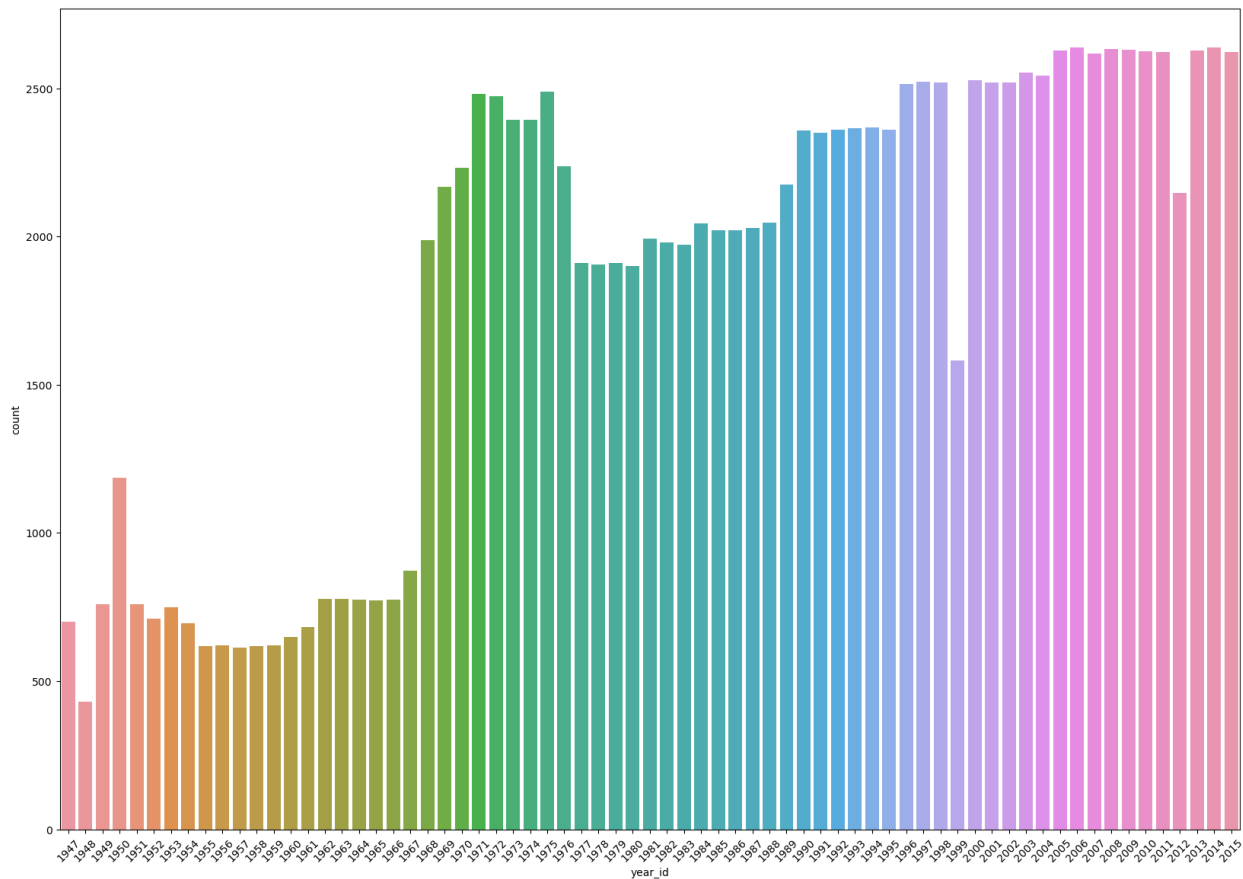
126313	63157	201506170CLE	NBA	1	2015	6/16/2015	
	seasongame	is_playoffs	team_id	fran_id	...	elo_n	
win_equiv \							
0	1	0	TRH	Huskies	...	1293.2767	
40.294830							
1	1	0	NYK	Knicks	...	1306.7233	
41.705170							
2	1	0	CHS	Stags	...	1309.6521	
42.012257							
3	2	0	NYK	Knicks	...	1297.0712	
40.692783							
4	1	0	DTF	Falcons	...	1279.6189	
38.864048							
...	
...							
126309	100	1	CLE	Cavaliers	...	1704.3949	
60.309792							
126310	102	1	GSW	Warriors	...	1813.6349	
68.013329							
126311	101	1	CLE	Cavaliers	...	1700.7391	
60.010067							
126312	102	1	CLE	Cavaliers	...	1692.0859	
59.290245							
126313	103	1	GSW	Warriors	...	1822.2881	
68.519516							
	opp_id	opp_fran	opp_pts	opp_elo_i	opp_elo_n	game_location	
\							
0	NYK	Knicks	68	1300.0000	1306.7233	H	
1	TRH	Huskies	66	1300.0000	1293.2767	A	
2	NYK	Knicks	47	1306.7233	1297.0712	H	
3	CHS	Stags	63	1300.0000	1309.6521	A	
4	WSC	Capitols	50	1300.0000	1320.3811	H	
...	
126309	GSW	Warriors	103	1790.9591	1809.9791	H	
126310	CLE	Cavaliers	91	1704.3949	1700.7391	H	
126311	GSW	Warriors	104	1809.9791	1813.6349	A	
126312	GSW	Warriors	105	1813.6349	1822.2881	H	
126313	CLE	Cavaliers	97	1700.7391	1692.0859	A	

	game_result	forecast
0	L	0.640065
1	W	0.359935
2	W	0.631101
3	L	0.368899
4	L	0.640065
...
126309	L	0.546572
126310	W	0.765565
126311	L	0.234435
126312	L	0.481450
126313	W	0.518550

[126314 rows x 22 columns]

It is a countplot representing number of NBA matches played in this every year
The sudden increase in number of matches played in year 1967 and 1968 is due to expansion of league from 10 teams to 12 teams.
Also for each played match there are two records in dataset, one for winning team and one for losing team.

```
plt.figure(figsize=(20,14))
sns.countplot(data=df, x="year_id")
plt.xticks(rotation=45)
plt.show()
```



```
df[df["date_game"] == '11/1/1946']
```

	gameorder	game_id	lg_id	_iscopy	year_id	date_game
seasongame \						
0	1	194611010TRH	NBA	0	1947	11/1/1946
1						
1	1	194611010TRH	NBA	1	1947	11/1/1946
1						

	is_playoffs	team_id	fran_id	...	elo_n	win_equiv	opp_id
opp_fran \							
0	0	TRH	Huskies	...	1293.2767	40.29483	NYK
Knicks							
1	0	NYK	Knicks	...	1306.7233	41.70517	TRH
Huskies							

	opp_pts	opp_elo_i	opp_elo_n	game_location	game_result	forecast
0	68	1300.0	1306.7233	H	L	0.640065
1	66	1300.0	1293.2767	A	W	0.359935

```
[2 rows x 22 columns]
```

winningGame

\	season	game	is_playoffs	team_id	fran_id	...	elo_n
1		1	0	NYK	Knicks	...	1306.7233
2		1	0	CHS	Stags	...	1309.6521
5		1	0	WSC	Capitols	...	1320.3811
7		1	0	PRO	Steamrollers	...	1305.1542
9		1	0	STB	Bombers	...	1304.6908
...	
126304		98	1	CLE	Cavaliers	...	1716.8708
126307		99	1	CLE	Cavaliers	...	1723.4149
126308		101	1	GSW	Warriors	...	1809.9791
126310		102	1	GSW	Warriors	...	1813.6349
126313		103	1	GSW	Warriors	...	1822.2881

	win_equiv	opp_id	opp_fran	opp_pts	opp_elo_i	opp_elo_n	\
1	41.705170	TRH	Huskies	66	1300.0000	1293.2767	
2	42.012257	NYK	Knicks	47	1306.7233	1297.0712	
5	43.135952	DTF	Falcons	33	1300.0000	1279.6189	
7	41.540619	BOS	Celtics	53	1300.0000	1294.8458	
9	41.492020	PIT	Ironmen	51	1300.0000	1295.3092	
...	

126304	61.312763	GSW	Warriors	93	1806.2035	1797.5032
126307	61.826408	GSW	Warriors	91	1797.5032	1790.9591
126308	67.794983	CLE	Cavaliers	82	1723.4149	1704.3949
126310	68.013329	CLE	Cavaliers	91	1704.3949	1700.7391
126313	68.519516	CLE	Cavaliers	97	1700.7391	1692.0859

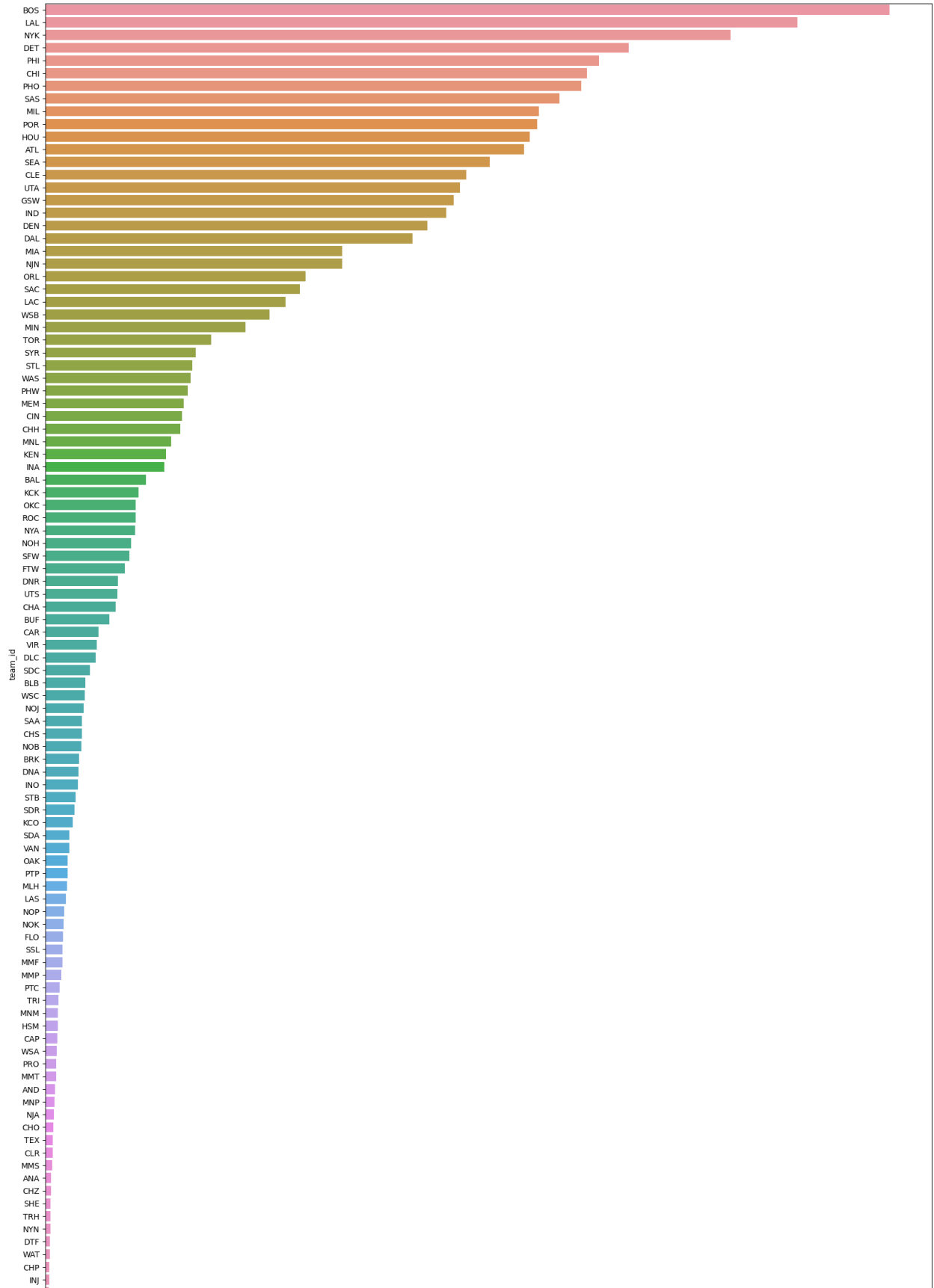
	game_location	game_result	forecast
1	A	W	0.359935
2	H	W	0.631101
5	A	W	0.359935
7	H	W	0.640065
9	H	W	0.640065
...
126304	A	W	0.242326
126307	H	W	0.527843
126308	A	W	0.453428
126310	H	W	0.765565
126313	A	W	0.518550

[63157 rows x 22 columns]

It is a barplot representing number of matches won by each team. The team with highest winnings is BOS and the team with less winnings is SDS

```
id_counts = winningGame['team_id'].value_counts().reset_index()
id_counts.columns = ['team_id', 'count']
sorted_df = id_counts.sort_values(by='count', ascending=False)

plt.figure(figsize=(20,30))
sns.barplot(data=sorted_df, x='count', y='team_id')
plt.show()
```



```
df[df['forecast'] > 0.5]
```

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	\
0	1	194611010TRH	NBA	0	1947	11/1/1946	
2	2	194611020CHS	NBA	0	1947	11/2/1946	
4	3	194611020DTF	NBA	0	1947	11/2/1946	
7	4	194611020PRO	NBA	0	1947	11/2/1946	
9	5	194611020STB	NBA	0	1947	11/2/1946	
...	
126305	63153	201506070GSW	NBA	0	2015	6/7/2015	
126307	63154	201506090CLE	NBA	0	2015	6/9/2015	
126309	63155	201506110CLE	NBA	0	2015	6/11/2015	
126310	63156	201506140GSW	NBA	0	2015	6/14/2015	
126313	63157	201506170CLE	NBA	1	2015	6/16/2015	

	seasongame	is_playoffs	team_id	fran_id	...	elo_n	\
0	1	0	TRH	Huskies	...	1293.2767	
2	1	0	CHS	Stags	...	1309.6521	
4	1	0	DTF	Falcons	...	1279.6189	
7	1	0	PRO	Steamrollers	...	1305.1542	
9	1	0	STB	Bombers	...	1304.6908	
...	
126305	99	1	GSW	Warriors	...	1797.5032	
126307	99	1	CLE	Cavaliers	...	1723.4149	
126309	100	1	CLE	Cavaliers	...	1704.3949	
126310	102	1	GSW	Warriors	...	1813.6349	
126313	103	1	GSW	Warriors	...	1822.2881	

	win_equiv	opp_id	opp_fran	opp_pts	opp_elo_i	opp_elo_n	\
0	40.294830	NYK	Knicks	68	1300.0000	1306.7233	
2	42.012257	NYK	Knicks	47	1306.7233	1297.0712	
4	38.864048	WSC	Capitols	50	1300.0000	1320.3811	
7	41.540619	BOS	Celtics	53	1300.0000	1294.8458	
9	41.492020	PIT	Ironmen	51	1300.0000	1295.3092	
...	
126305	67.029549	CLE	Cavaliers	95	1708.1705	1716.8708	
126307	61.826408	GSW	Warriors	91	1797.5032	1790.9591	

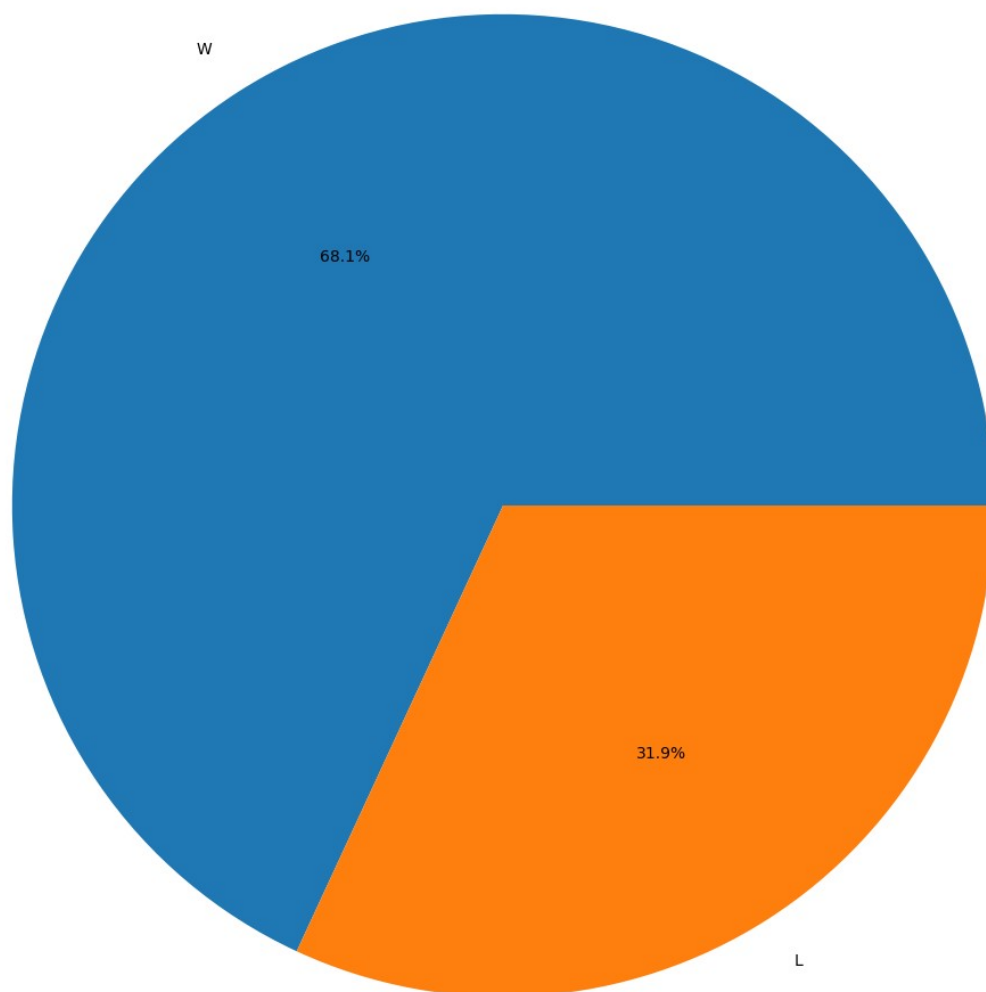
126309	60.309792	GSW	Warriors	103	1790.9591	1809.9791
126310	68.013329	CLE	Cavaliers	91	1704.3949	1700.7391
126313	68.519516	CLE	Cavaliers	97	1700.7391	1692.0859

	game_location	game_result	forecast
0	H	L	0.640065
2	H	W	0.631101
4	H	L	0.640065
7	H	W	0.640065
9	H	W	0.640065
...
126305	H	L	0.757674
126307	H	W	0.527843
126309	H	L	0.546572
126310	H	W	0.765565
126313	A	W	0.518550

[63157 rows x 22 columns]

It is a piechart represnting the accuracy of forcast. It represents number actual wins when forecast was win and number of actual loss even forecast was loss

```
plt.figure(figsize=(20,14))
count = df[df['forecast'] > 0.5]['game_result'].value_counts()
plt.pie(count,labels = count.index,autopct='%.1f%%')
plt.show()
```



df

	gameorder	game_id	lg_id	_iscopy	year_id	date_game	\
0	1	194611010TRH	NBA	0	1947	11/1/1946	
1	1	194611010TRH	NBA	1	1947	11/1/1946	
2	2	194611020CHS	NBA	0	1947	11/2/1946	
3	2	194611020CHS	NBA	1	1947	11/2/1946	
4	3	194611020DTF	NBA	0	1947	11/2/1946	
...	
126309	63155	201506110CLE	NBA	0	2015	6/11/2015	
126310	63156	201506140GSW	NBA	0	2015	6/14/2015	

126311	63156	201506140	GSW	NBA	1	2015	6/14/2015
126312	63157	201506170	CLE	NBA	0	2015	6/16/2015
126313	63157	201506170	CLE	NBA	1	2015	6/16/2015

season	game	is_playoffs	team_id	fran_id	...	elo_n
win_equiv \						
0	1	0	TRH	Huskies	...	1293.2767
40.294830						
1	1	0	NYK	Knicks	...	1306.7233
41.705170						
2	1	0	CHS	Stags	...	1309.6521
42.012257						
3	2	0	NYK	Knicks	...	1297.0712
40.692783						
4	1	0	DTF	Falcons	...	1279.6189
38.864048						
...
...						
126309	100	1	CLE	Cavaliers	...	1704.3949
60.309792						
126310	102	1	GSW	Warriors	...	1813.6349
68.013329						
126311	101	1	CLE	Cavaliers	...	1700.7391
60.010067						
126312	102	1	CLE	Cavaliers	...	1692.0859
59.290245						
126313	103	1	GSW	Warriors	...	1822.2881
68.519516						

opp_id	opp_fran	opp_pts	opp_elo_i	opp_elo_n	game_location
\					
0	NYK	Knicks	68	1300.0000	1306.7233
					H
1	TRH	Huskies	66	1300.0000	1293.2767
					A
2	NYK	Knicks	47	1306.7233	1297.0712
					H
3	CHS	Stags	63	1300.0000	1309.6521
					A
4	WSC	Capitols	50	1300.0000	1320.3811
					H
...
126309	GSW	Warriors	103	1790.9591	1809.9791
					H
126310	CLE	Cavaliers	91	1704.3949	1700.7391
					H
126311	GSW	Warriors	104	1809.9791	1813.6349
					A
126312	GSW	Warriors	105	1813.6349	1822.2881
					H

```
126313    CLE  Cavaliers    97  1700.7391  1692.0859    A
```

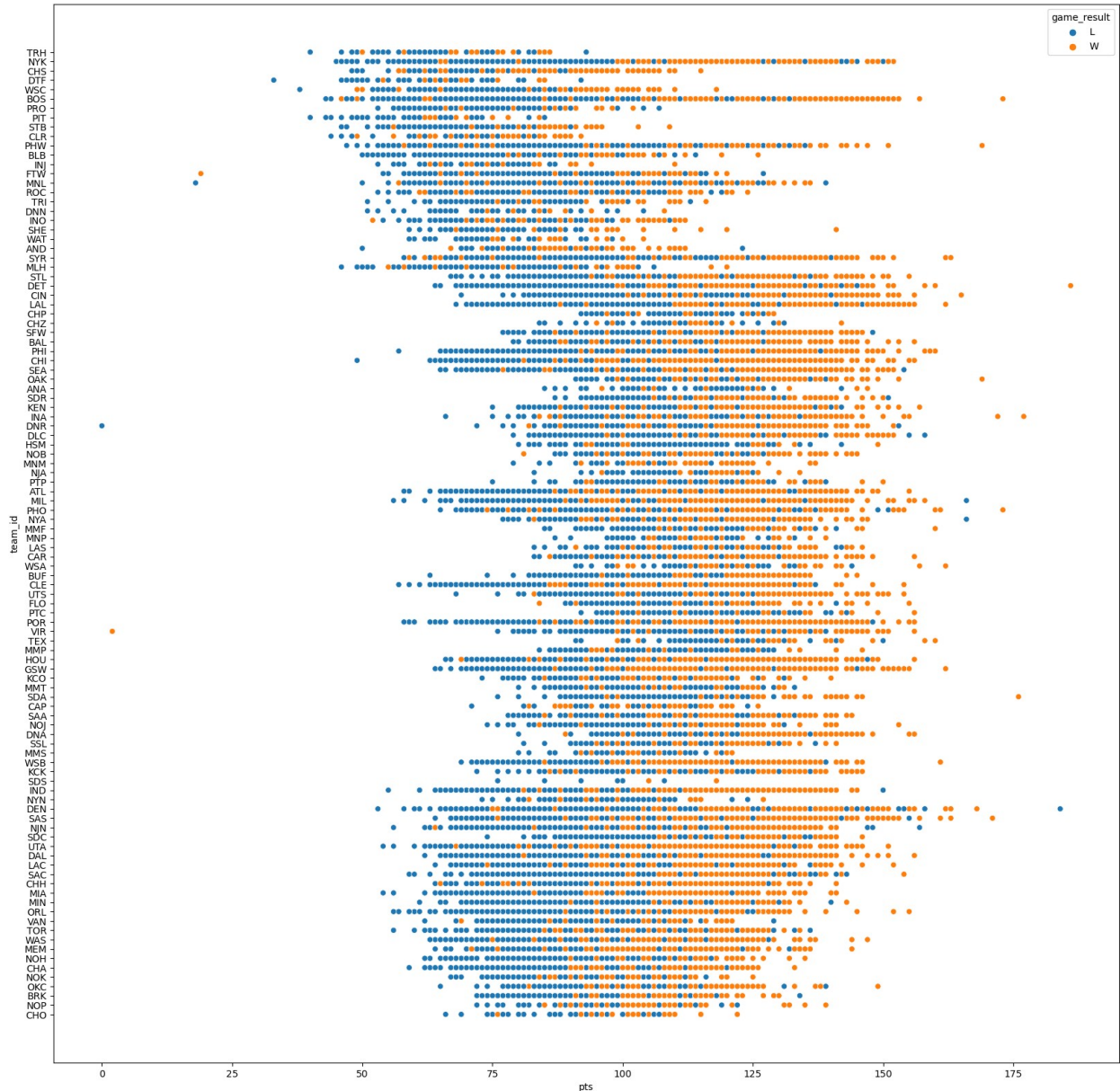
```
      game_result  forecast
0              L  0.640065
1              W  0.359935
2              W  0.631101
3              L  0.368899
4              L  0.640065
...          ...    ...
126309          L  0.546572
126310          W  0.765565
126311          L  0.234435
126312          L  0.481450
126313          W  0.518550
```

```
[126314 rows x 22 columns]
```

```
# It is a scatterplot repring number of points scored by each team  
in different matches
```

```
# A general range of goals scored in match is 50 to 150
```

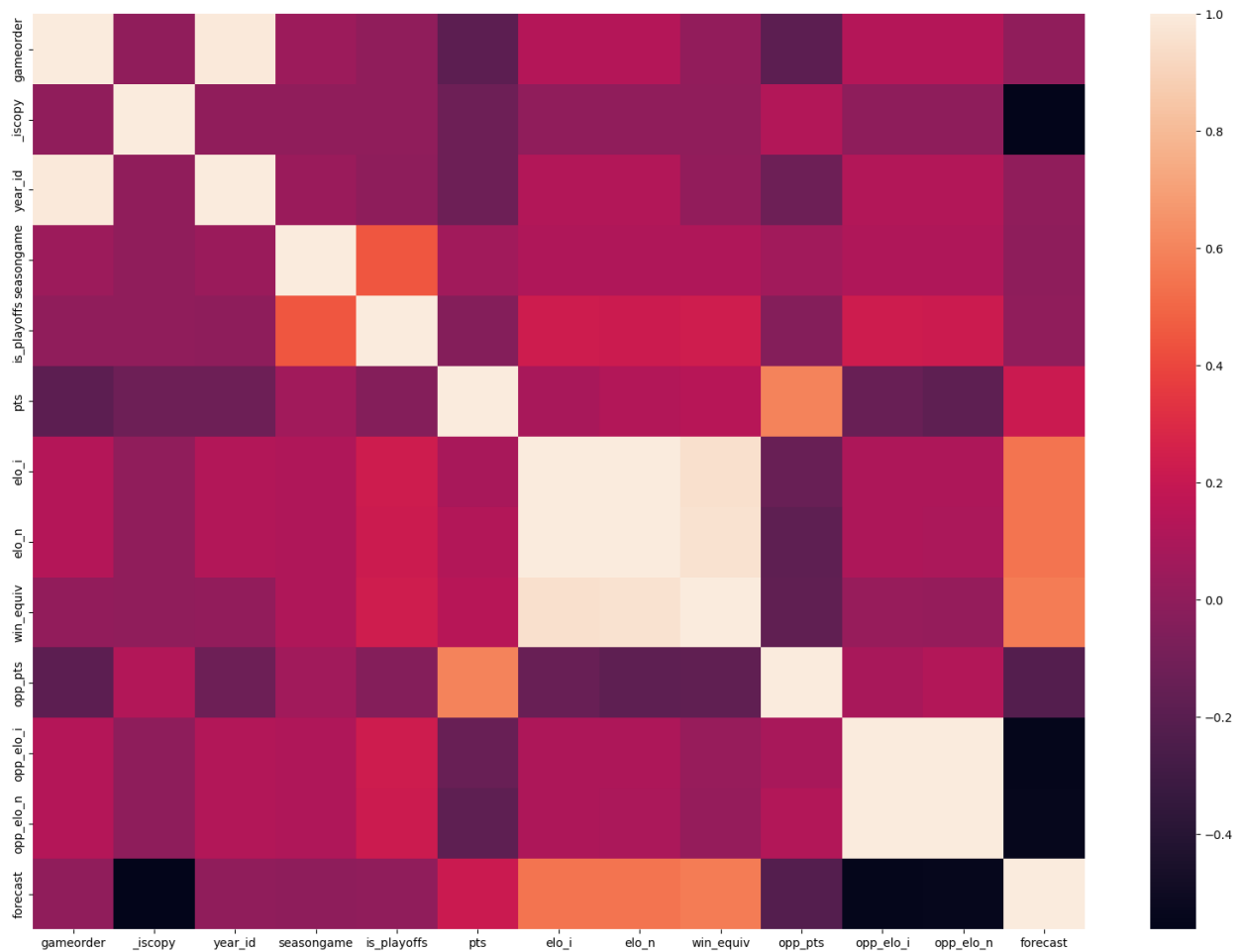
```
plt.figure(figsize=(20,20))
sns.scatterplot(data=df,x='pts',y='team_id',hue='game_result')
plt.show()
```



It is a heatmap representing how the features in dataset are related to each other

```
plt.figure(figsize=(20, 14))
sns.heatmap(df.corr(numeric_only=True))
plt.plot()
```

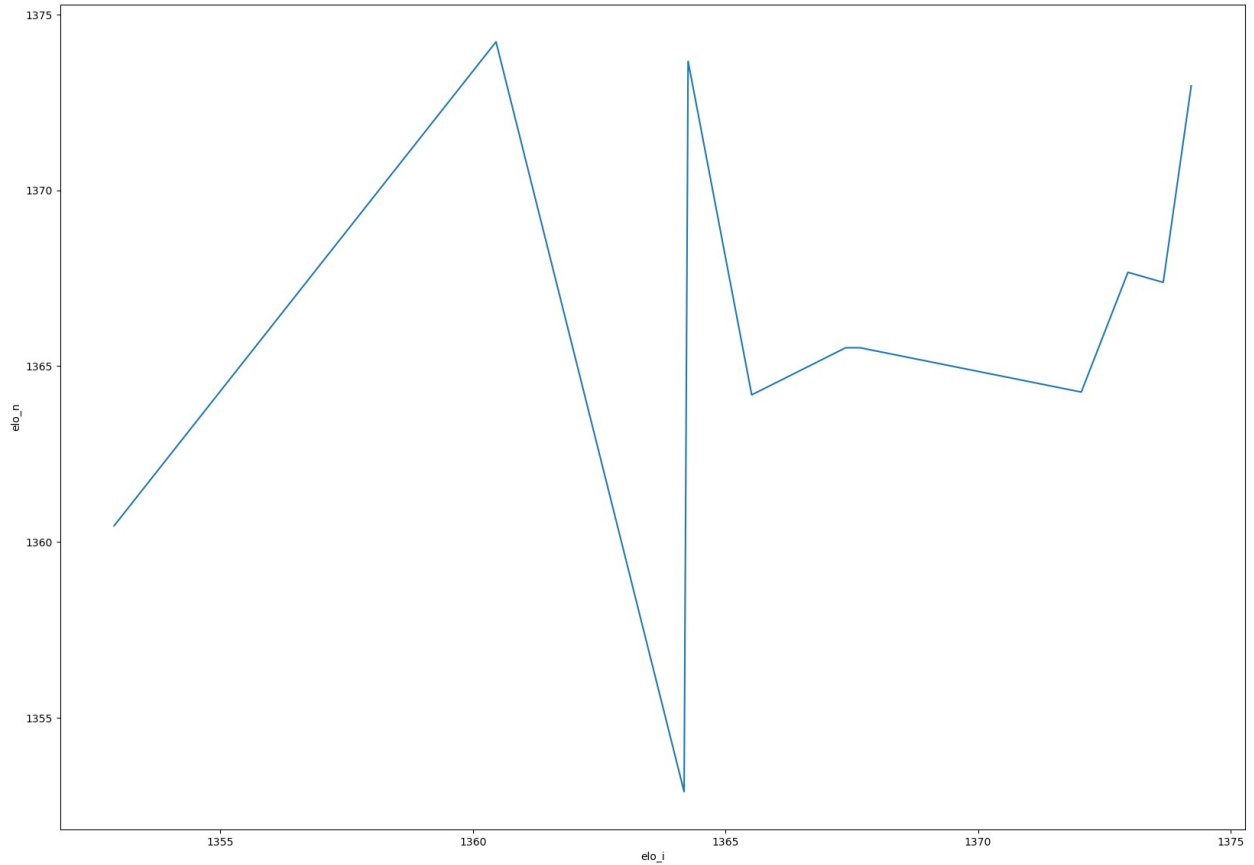
[[



It is a line plot representing the fluctuations of elo rating of team SDS over time

```
plt.figure(figsize=(20, 14))
sns.lineplot(data=df[df['team_id']=='SDS'], x="elo_i", y="elo_n")
plt.plot()
```

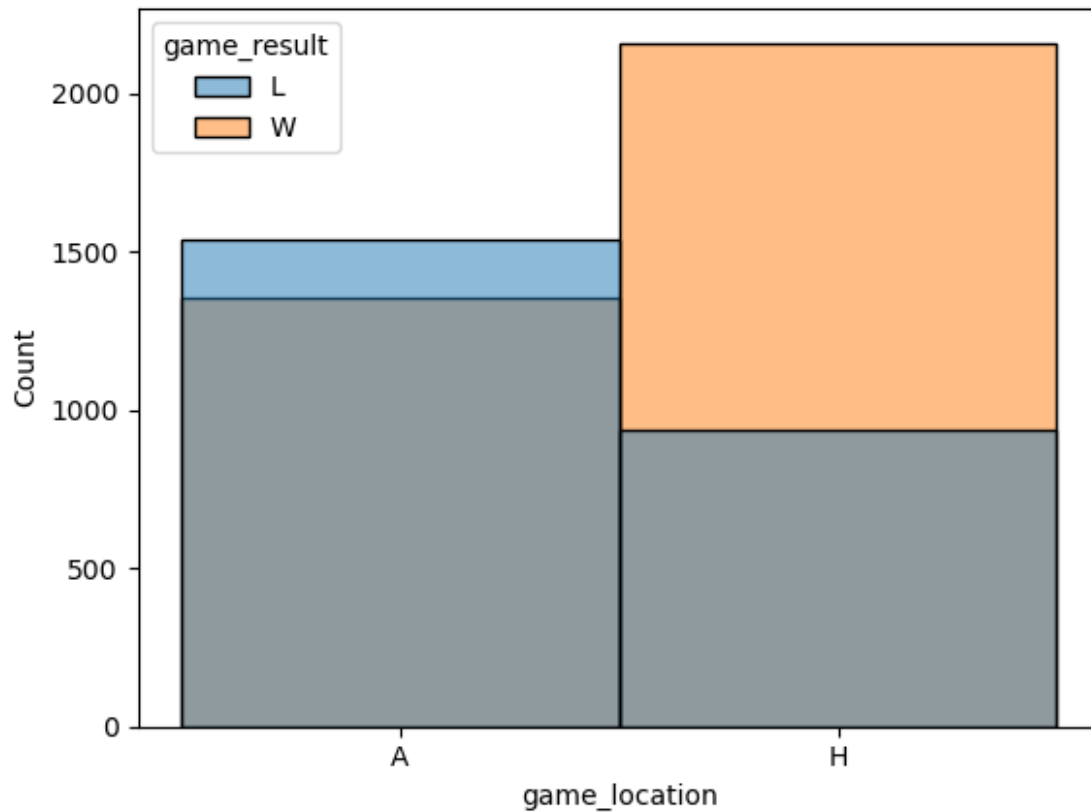
```
[]
```



*#It is a histplot representing the number of matches won and lose by team BOS in home ground and away ground
team has won more matches in home grounds*

```
sns.histplot(data=df[df['team_id']=='BOS'], x="game_location",  
hue="game_result")
```

```
<Axes: xlabel='game_location', ylabel='Count'>
```



it is joint plot represnting realtion between game results, points score and match results
Mostly if team has scored above 100 points, they have won game
and forcast has not much effect on game results

```
plt.figure(figsize=(20, 14))  
sns.jointplot(data=df, x="pts", y="forecast", hue="game_result")  
plt.plot()
```

[]

<Figure size 2000x1400 with 0 Axes>

