

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.gridspec
%matplotlib inline

sns.set(color_codes=True)

df= pd.read_csv("Basketball.csv")
```

```
df
```

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames \
0	Team 1	86	4385	2762	1647	552	
1	Team 2	86	4262	2762	1581	573	
2	Team 3	80	3442	2614	1241	598	
3	Team 4	82	3386	2664	1187	616	
4	Team 5	86	3368	2762	1209	633	
..
56	Team 57	1	34	38	8	10	
57	Team 58	1	22	30	7	8	
58	Team 59	1	19	30	7	5	
59	Team 60	1	14	30	5	4	
60	Team 61	1	-	-	-	-	

	BasketScored	BasketGiven	TournamentChampion	Runner-up	TeamLaunch	\
0	5947	3140	33	23	1929	
1	5900	3114	25	25	1929	
2	4534	3309	10	8	1929	
3	4398	3469	6	6	1931to32	
4	4631	3700	8	7	1929	
..
56	38	66	-	-	2009-10	
57	37	57	-	-	1956-57	
58	51	85	-	-	1951~52	
59	34	65	-	-	1955-56	
60	-	-	-	-	2017~18	

	HighestPositionHeld
0	1
1	1
2	1
3	1
4	1
..	...
56	20
57	16
58	16
59	15
60	9

[61 rows x 13 columns]

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 61 entries, 0 to 60

Data columns (total 13 columns):

#	Column	Non-Null Count	Dtype
0	Team	61 non-null	object
1	Tournament	61 non-null	int64
2	Score	61 non-null	object
3	PlayedGames	61 non-null	object
4	WonGames	61 non-null	object
5	DrawnGames	61 non-null	object
6	LostGames	61 non-null	object
7	BasketScored	61 non-null	object
8	BasketGiven	61 non-null	object
9	TournamentChampion	61 non-null	object
10	Runner-up	61 non-null	object
11	TeamLaunch	61 non-null	object
12	HighestPositionHeld	61 non-null	int64

dtypes: int64(2), object(11)

memory usage: 6.3+ KB

df.isnull().sum()

Team	0
Tournament	0
Score	0
PlayedGames	0
WonGames	0
DrawnGames	0
LostGames	0
BasketScored	0

```

BasketGiven      0
TournamentChampion  0
Runner-up        0
TeamLaunch       0
HighestPositionHeld  0
dtype: int64

```

```
df.isna().sum()
```

```

Team      0
Tournament  0
Score      0
PlayedGames  0
WonGames   0
DrawnGames  0
LostGames  0
BasketScored  0
BasketGiven  0
TournamentChampion  0
Runner-up  0
TeamLaunch  0
HighestPositionHeld  0
dtype: int64

```

```
df[df.Score.str.isdigit()==False]
```

```

      Team  Tournament  Score  PlayedGames  WonGames  DrawnGames
LostGames \
60 Team 61          1    -          -          -          -
-
```

```

      BasketScored  BasketGiven  TournamentChampion  Runner-up  TeamLaunch \
60          -          -          -          -          -  2017~18

```

```

      HighestPositionHeld
60          9

```

```
df['Score'] = df['Score'].replace('-', 0)
```

```
df[df.Score.str.isdigit()==False]
```

```
Empty DataFrame
```

```

Columns: [Team, Tournament, Score, PlayedGames, WonGames, DrawnGames,
LostGames, BasketScored, BasketGiven, TournamentChampion, Runner-up,
TeamLaunch, HighestPositionHeld]
Index: []

```

```
df['Score'] = df['Score'].astype('int64')
```

```
df = df.replace('-', 0)
```

```

cols = df.columns.drop(['Team', 'TeamLaunch'])
cols

Index(['Tournament', 'Score', 'PlayedGames', 'WonGames', 'DrawnGames',
       'LostGames', 'BasketScored', 'BasketGiven',
       'TournamentChampion',
       'Runner-up', 'HighestPositionHeld'],
      dtype='object')

df[cols] = df[cols].apply(pd.to_numeric)

df['DebutYear'] = df['TeamLaunch'].str.slice(0, 4)

df['Defenders'] = df['BasketScored'] - df['BasketGiven']

df['BasketsScoredPerMatch'] = (df['BasketScored'] /
df['PlayedGames']).round(2)

df['BasketsGivenPerMatch'] = (df['BasketGiven'] /
df['PlayedGames']).round(2)

df['DebutYear'] = df['DebutYear'].astype('int64')

df['WinningPercentage'] = (df['WonGames']/df['PlayedGames'])*100
df['LosingPercentage'] = (df['LostGames']/df['PlayedGames'])*100
df

```

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames
0	Team 1	86	4385	2762	1647	552
563						
1	Team 2	86	4262	2762	1581	573
608						
2	Team 3	80	3442	2614	1241	598
775						
3	Team 4	82	3386	2664	1187	616
861						
4	Team 5	86	3368	2762	1209	633
920						
..
...						
56	Team 57	1	34	38	8	10
20						
57	Team 58	1	22	30	7	8
15						
58	Team 59	1	19	30	7	5
18						
59	Team 60	1	14	30	5	4
21						
60	Team 61	1	0	0	0	0

0

	BasketScored	BasketGiven	TournamentChampion	Runner-up
--	--------------	-------------	--------------------	-----------

TeamLaunch \

0	5947	3140	33	23
---	------	------	----	----

1929

1	5900	3114	25	25
---	------	------	----	----

1929

2	4534	3309	10	8
---	------	------	----	---

1929

3	4398	3469	6	6
---	------	------	---	---

1931to32

4	4631	3700	8	7
---	------	------	---	---

1929

..
----	-----	-----	-----	-----	----

.

56	38	66	0	0	2009-
----	----	----	---	---	-------

10

57	37	57	0	0	1956-
----	----	----	---	---	-------

57

58	51	85	0	0
----	----	----	---	---

1951~52

59	34	65	0	0	1955-
----	----	----	---	---	-------

56

60	0	0	0	0
----	---	---	---	---

2017~18

	HighestPositionHeld	DebutYear	Defenders
--	---------------------	-----------	-----------

BasketsScoredPerMatch \

0	1	1929	2807	2.15
---	---	------	------	------

1	1	1929	2786	2.14
---	---	------	------	------

2	1	1929	1225	1.73
---	---	------	------	------

3	1	1931	929	1.65
---	---	------	-----	------

4	1	1929	931	1.68
---	---	------	-----	------

..
----	-----	-----	-----	-----

56	20	2009	-28	1.00
----	----	------	-----	------

57	16	1956	-20	1.23
----	----	------	-----	------

58	16	1951	-34	1.70
----	----	------	-----	------

59	15	1955	-31	1.13
----	----	------	-----	------

60	9	2017	0	NaN
----	---	------	---	-----

	BasketsGivenPerMatch	WinningPercentage	LosingPercentage
0	1.14	59.630702	20.383780
1	1.13	57.241130	22.013034
2	1.27	47.475134	29.648049
3	1.30	44.557057	32.319820
4	1.34	43.772629	33.309196
...
56	1.74	21.052632	52.631579
57	1.90	23.333333	50.000000
58	2.83	23.333333	60.000000
59	2.17	16.666667	70.000000
60	NaN	NaN	NaN

[61 rows x 19 columns]

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 61 entries, 0 to 60

Data columns (total 19 columns):

#	Column	Non-Null Count	Dtype
0	Team	61 non-null	object
1	Tournament	61 non-null	int64
2	Score	61 non-null	int64
3	PlayedGames	61 non-null	int64
4	WonGames	61 non-null	int64
5	DrawnGames	61 non-null	int64
6	LostGames	61 non-null	int64
7	BasketScored	61 non-null	int64
8	BasketGiven	61 non-null	int64
9	TournamentChampion	61 non-null	int64
10	Runner-up	61 non-null	int64
11	TeamLaunch	61 non-null	object
12	HighestPositionHeld	61 non-null	int64
13	DebutYear	61 non-null	int64
14	Defenders	61 non-null	int64
15	BasketsScoredPerMatch	60 non-null	float64
16	BasketsGivenPerMatch	60 non-null	float64
17	WinningPercentage	60 non-null	float64
18	LosingPercentage	60 non-null	float64

dtypes: float64(4), int64(13), object(2)

memory usage: 9.2+ KB

df.describe().transpose()

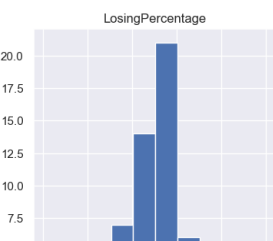
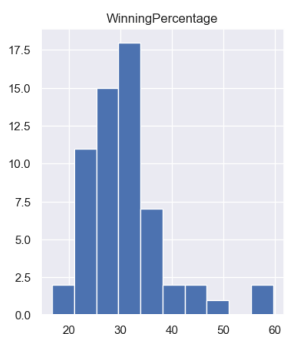
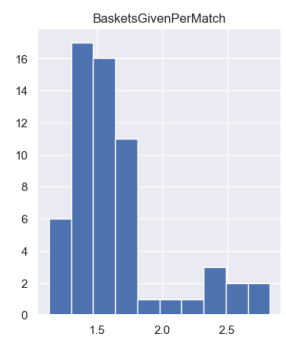
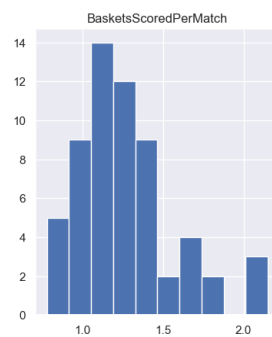
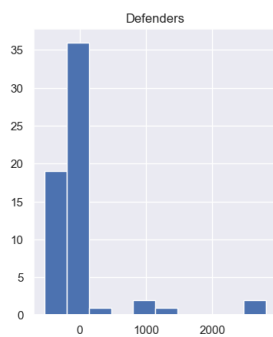
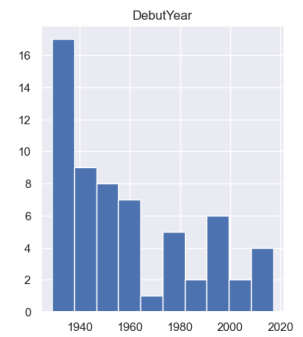
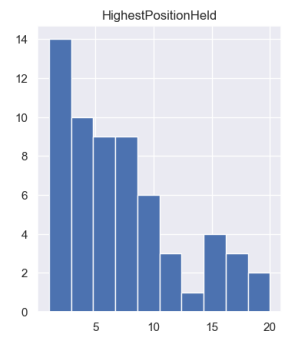
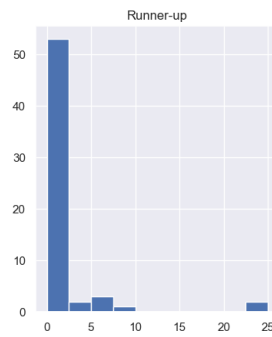
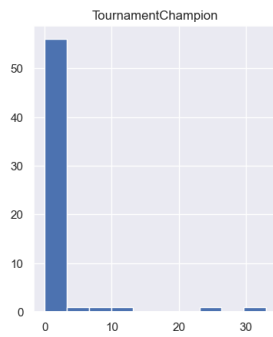
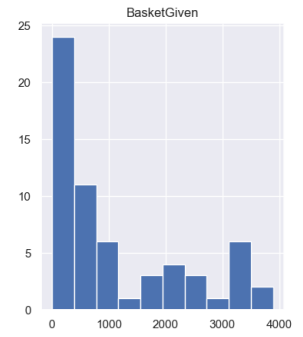
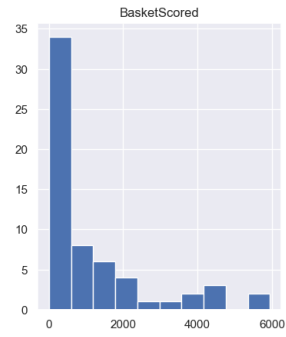
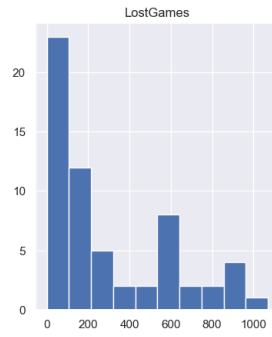
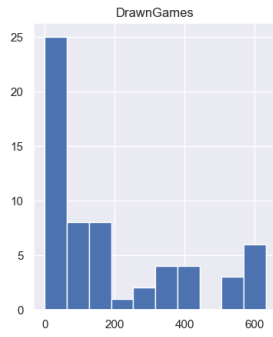
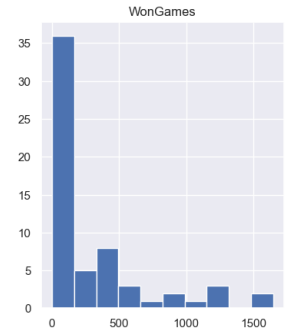
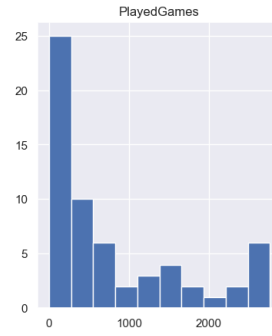
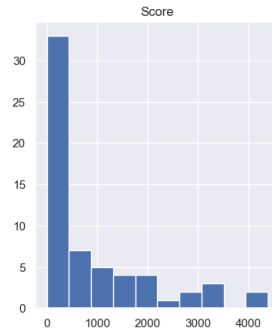
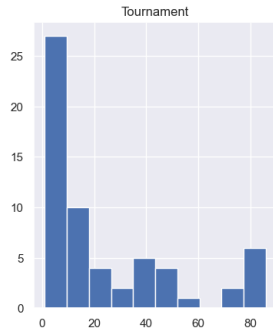
	count	mean	std	min	\
Tournament	61.0	24.000000	26.827225	1.000000	

Score	61.0	901.426230	1134.899121	0.000000
PlayedGames	61.0	796.819672	876.282765	0.000000
WonGames	61.0	303.967213	406.991030	0.000000
DrawnGames	61.0	188.934426	201.799477	0.000000
LostGames	61.0	303.754098	294.708594	0.000000
BasketScored	61.0	1140.344262	1506.740211	0.000000
BasketGiven	61.0	1140.229508	1163.710766	0.000000
TournamentChampion	61.0	1.426230	5.472535	0.000000
Runner-up	61.0	1.409836	4.540107	0.000000
HighestPositionHeld	61.0	7.081967	5.276663	1.000000
DebutYear	61.0	1958.918033	27.484114	1929.000000
Defenders	61.0	0.114754	598.095814	-525.000000
BasketsScoredPerMatch	60.0	1.263333	0.307944	0.780000
BasketsGivenPerMatch	60.0	1.632833	0.383313	1.130000
WinningPercentage	60.0	31.364790	7.831199	16.666667
LosingPercentage	60.0	44.991949	8.400881	20.383780
		25%	50%	75%
max				
Tournament	4.000000	12.000000	38.000000	
86.000000				
Score	96.000000	375.000000	1351.000000	
4385.000000				
PlayedGames	114.000000	423.000000	1318.000000	
2762.000000				
WonGames	34.000000	123.000000	426.000000	
1647.000000				
DrawnGames	24.000000	95.000000	330.000000	
633.000000				
LostGames	62.000000	197.000000	563.000000	
1070.000000				
BasketScored	153.000000	430.000000	1642.000000	
5947.000000				
BasketGiven	221.000000	632.000000	1951.000000	
3889.000000				
TournamentChampion	0.000000	0.000000	0.000000	
33.000000				
Runner-up	0.000000	0.000000	0.000000	
25.000000				
HighestPositionHeld	3.000000	6.000000	10.000000	
20.000000				
DebutYear	1935.000000	1951.000000	1978.000000	
2017.000000				
Defenders	-269.000000	-112.000000	-34.000000	
2807.000000				
BasketsScoredPerMatch	1.067500	1.205000	1.355000	
2.150000				
BasketsGivenPerMatch	1.407500	1.515000	1.675000	
2.830000				

WinningPercentage	27.607494	30.491722	33.540164
59.630702			
LosingPercentage	41.142971	45.727107	48.542449
70.000000			

```
df.hist(figsize=(20,30))
```

```
array([[<Axes: title={'center': 'Tournament'}>,
        <Axes: title={'center': 'Score'}>,
        <Axes: title={'center': 'PlayedGames'}>,
        <Axes: title={'center': 'WonGames'}>],
       [<Axes: title={'center': 'DrawnGames'}>,
        <Axes: title={'center': 'LostGames'}>,
        <Axes: title={'center': 'BasketScored'}>,
        <Axes: title={'center': 'BasketGiven'}>],
       [<Axes: title={'center': 'TournamentChampion'}>,
        <Axes: title={'center': 'Runner-up'}>,
        <Axes: title={'center': 'HighestPositionHeld'}>,
        <Axes: title={'center': 'DebutYear'}>],
       [<Axes: title={'center': 'Defenders'}>,
        <Axes: title={'center': 'BasketsScoredPerMatch'}>,
        <Axes: title={'center': 'BasketsGivenPerMatch'}>,
        <Axes: title={'center': 'WinningPercentage'}>],
       [<Axes: title={'center': 'LosingPercentage'}>, <Axes: >, <Axes:
>,
        <Axes: >]], dtype=object)
```

```

fig, ax = plt.subplots(2, 3)

fig.set_figheight(10)
fig.set_figwidth(20)

sns.distplot(df['DebutYear'], ax = ax[0,0])
sns.distplot(df['WinningPercentage'], ax = ax[0,1])
sns.distplot(df['BasketsScoredPerMatch'], ax = ax[0,2])
sns.distplot(df['BasketsGivenPerMatch'], ax = ax[1,0])
sns.distplot(df['PlayedGames'], ax = ax[1,1])
sns.distplot(df['Tournament'], ax = ax[1,2])

ax[0,0].set_title('Debut Year')
ax[0,1].set_title('Winning Percentage')
ax[0,2].set_title('Baskets Scored Permatch')
ax[1,0].set_title('Baskets Given Permatch')
ax[1,1].set_title('Played Games')
ax[1,2].set_title('Tournament')

plt.show()

```

C:\Users\WACOM\AppData\Local\Temp\ipykernel_14512\3793160979.py:6:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```

sns.distplot(df['DebutYear'], ax = ax[0,0])
C:\Users\WACOM\AppData\Local\Temp\ipykernel_14512\3793160979.py:7:
UserWarning:

```

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```

sns.distplot(df['WinningPercentage'], ax = ax[0,1])

```

```
C:\Users\WACOM\AppData\Local\Temp\ipykernel_14512\3793160979.py:8:  
UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn  
v0.14.0.
```

Please adapt your code to use either ``displot`` (a figure-level function with similar flexibility) or ``histplot`` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['BasketsScoredPerMatch'], ax = ax[0,2])  
C:\Users\WACOM\AppData\Local\Temp\ipykernel_14512\3793160979.py:9:  
UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn  
v0.14.0.
```

Please adapt your code to use either ``displot`` (a figure-level function with similar flexibility) or ``histplot`` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['BasketsGivenPerMatch'], ax = ax[1,0])  
C:\Users\WACOM\AppData\Local\Temp\ipykernel_14512\3793160979.py:10:  
UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn  
v0.14.0.
```

Please adapt your code to use either ``displot`` (a figure-level function with similar flexibility) or ``histplot`` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

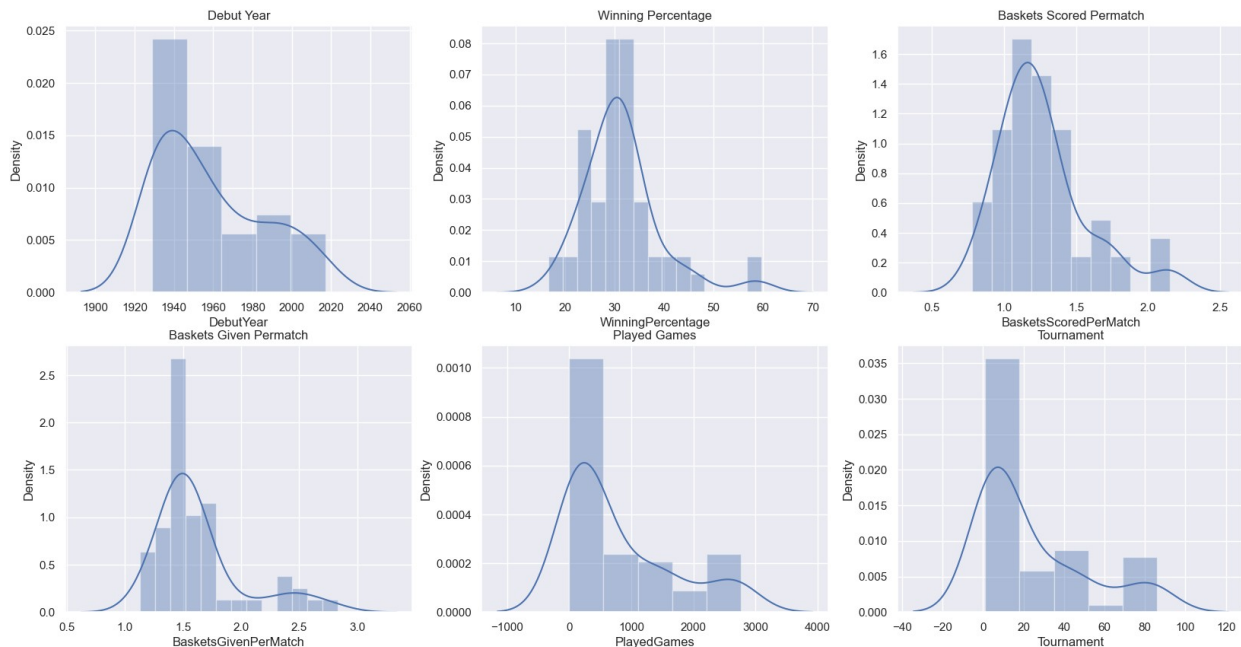
```
sns.distplot(df['PlayedGames'], ax = ax[1,1])  
C:\Users\WACOM\AppData\Local\Temp\ipykernel_14512\3793160979.py:11:  
UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn  
v0.14.0.
```

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['Tournament'], ax = ax[1,2])
```



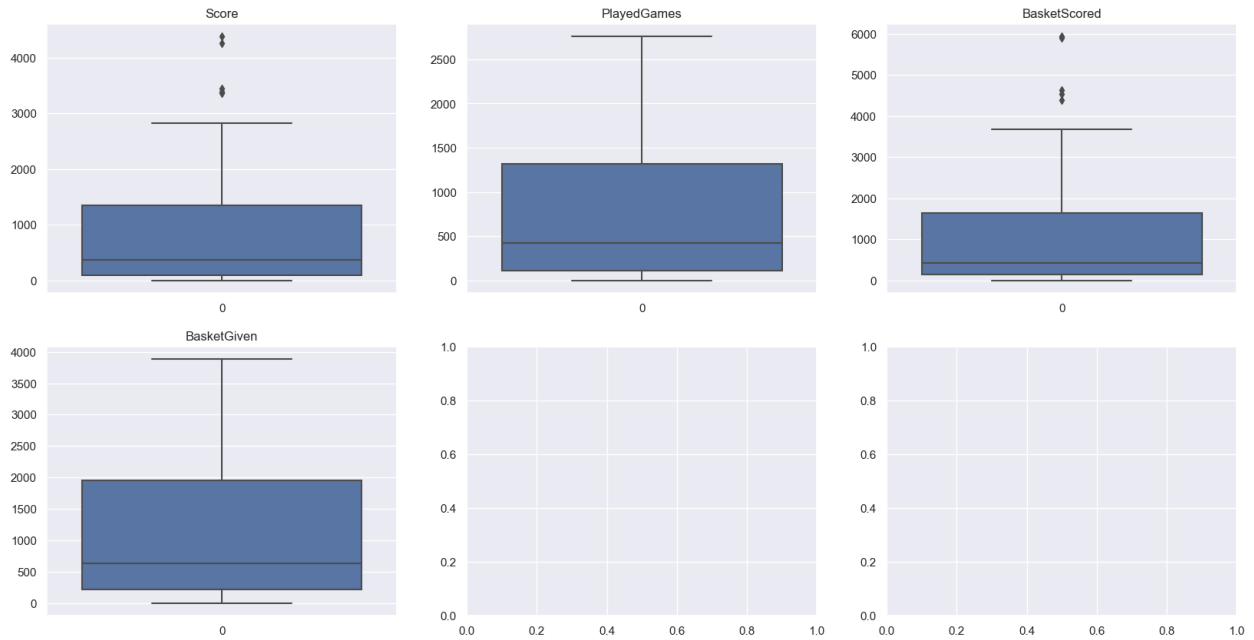
```
fig, ax = plt.subplots(2, 3)

fig.set_figheight(10)
fig.set_figwidth(20)

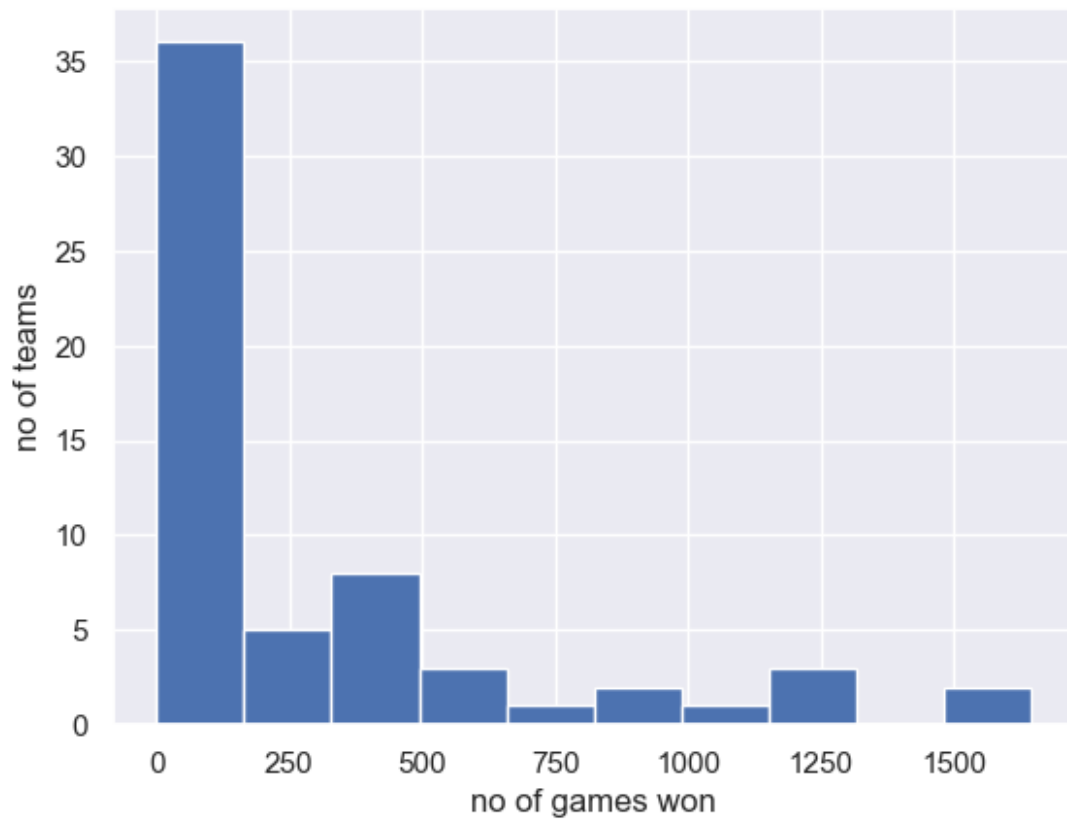
sns.boxplot(df['Score'], ax = ax[0,0])
sns.boxplot(df['PlayedGames'], ax = ax[0,1])
sns.boxplot(df['BasketScored'], ax = ax[0,2])
sns.boxplot(df['BasketGiven'], ax = ax[1,0])

ax[0,0].set_title('Score')
ax[0,1].set_title('PlayedGames')
ax[0,2].set_title('BasketScored')
ax[1,0].set_title('BasketGiven')

plt.show()
```



```
plt.hist(df['WonGames'])
plt.xlabel('no of games won')
plt.ylabel('no of teams')
plt.show()
```



```

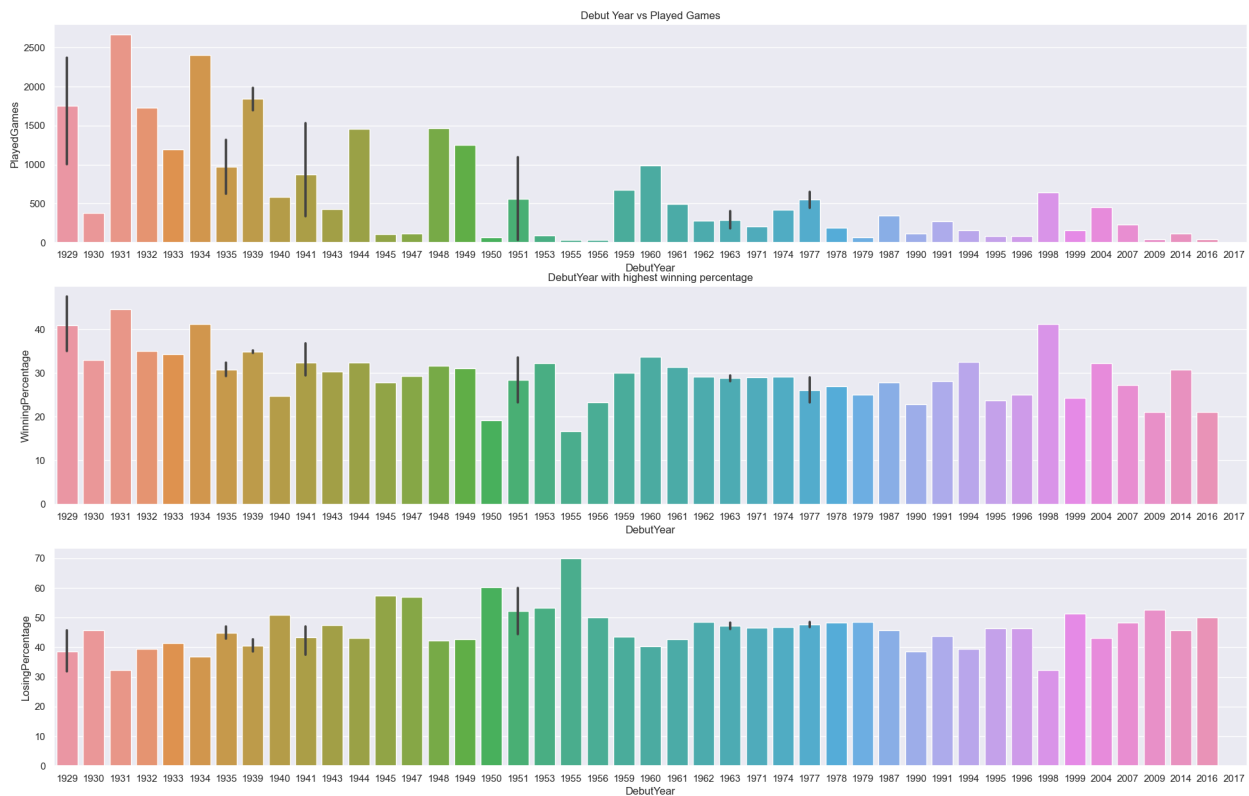
fig, ax = plt.subplots(3,1)

fig.set_figheight(15)
fig.set_figwidth(24)

sns.barplot(x='DebutYear',y='PlayedGames', data=df, ax=ax[0])
sns.barplot(x='DebutYear',y='WinningPercentage', data=df, ax=ax[1])
sns.barplot(x='DebutYear',y='LosingPercentage', data=df, ax=ax[2])

ax[0].set_title('Debut Year vs Played Games')
ax[1].set_title('DebutYear with highest winning percentage')
plt.show()

```



```

fig, ax = plt.subplots(2,1)

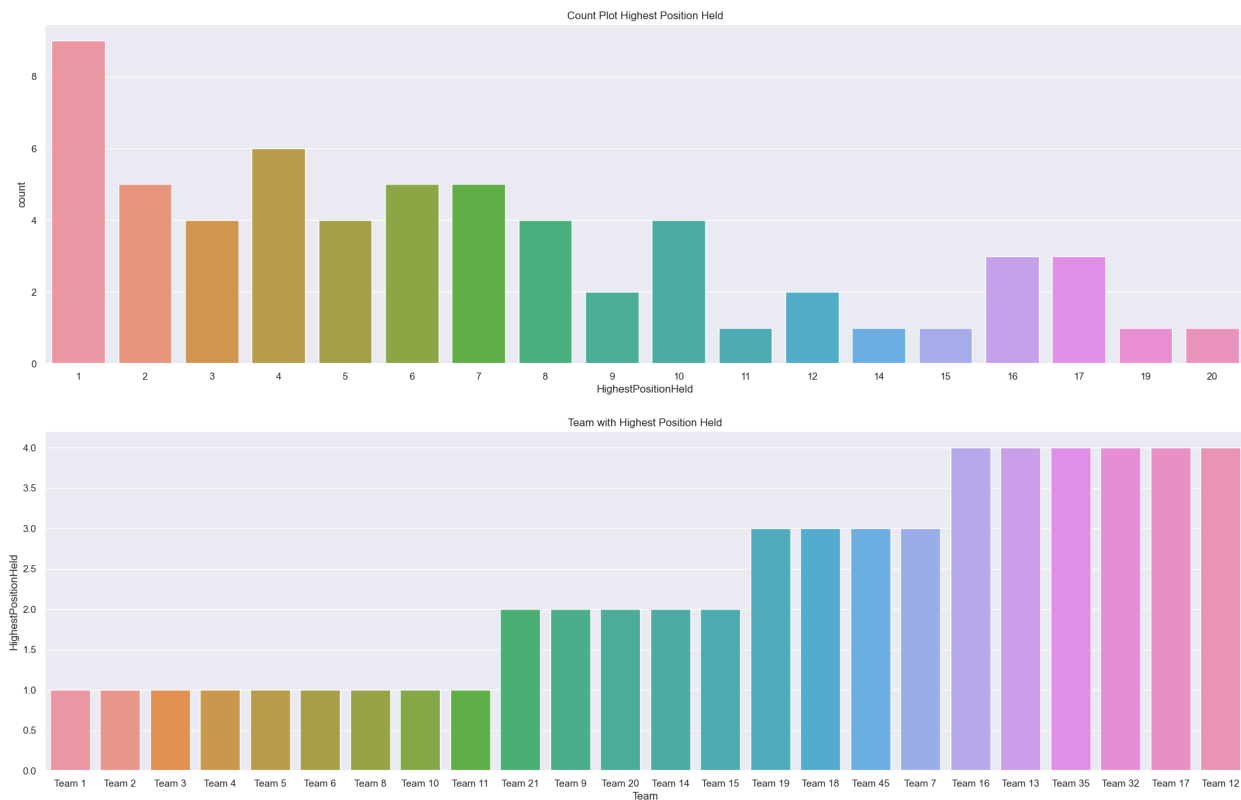
fig.set_figheight(15)
fig.set_figwidth(24)

sns.countplot(x="HighestPositionHeld", data=df, ax=ax[0])
sns.barplot(x='Team',y='HighestPositionHeld',
data=df[df.HighestPositionHeld <
5].sort_values(by='HighestPositionHeld', ascending=True), ax=ax[1])

ax[0].set_title('Count Plot Highest Position Held')

```

```
ax[1].set_title('Team with Highest Position Held')
plt.show()
```



```
fig, ax = plt.subplots(3,2)

fig.set_figheight(15)
fig.set_figwidth(22)

sns.barplot(x='Team',y='Tournament',
data=df.sort_values(by='Tournament', ascending=False).head(10),
ax=ax[0,0])
sns.barplot(x='Team',y='WinningPercentage',
data=df.sort_values(by='WinningPercentage', ascending=False).head(10),
ax=ax[0,1])
sns.barplot(x='Team',y='LosingPercentage',
data=df.sort_values(by='LosingPercentage', ascending=True).head(10),
ax=ax[1,0])
sns.barplot(x='Team',y='BasketsScoredPerMatch',
data=df.sort_values(by='BasketsScoredPerMatch',
ascending=False).head(10), ax=ax[1,1])
sns.barplot(x='Team',y='BasketsGivenPerMatch',
data=df.sort_values(by='BasketsGivenPerMatch',
ascending=False).head(10), ax=ax[2,0])
sns.barplot(x='Team',y='Defenders',
data=df.sort_values(by='Defenders', ascending=False).head(10),
```

```
ax=ax[2,1])
```

```
ax[0,0].set_title('Teams Played No.of Tournments')
```

```
ax[0,1].set_title('Teams with highest winning percentage')
```

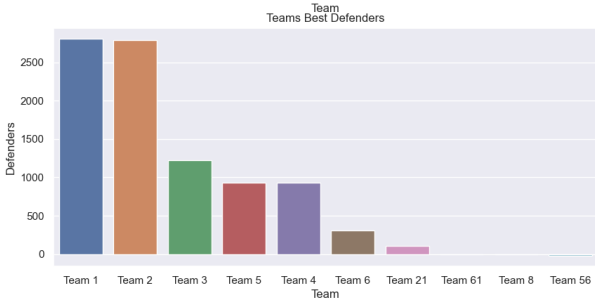
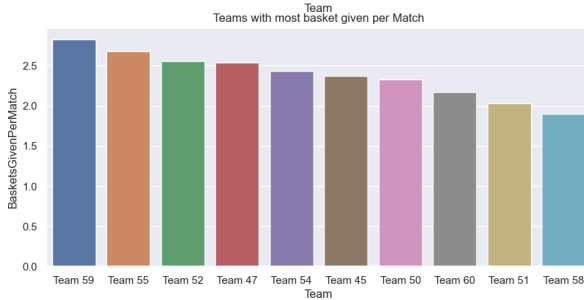
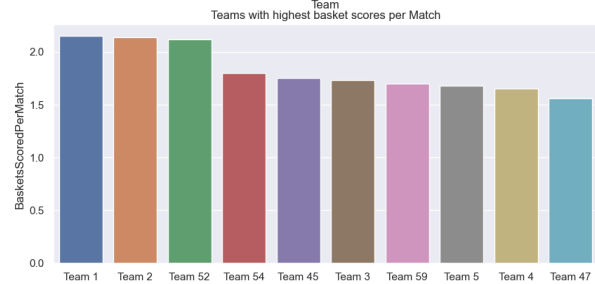
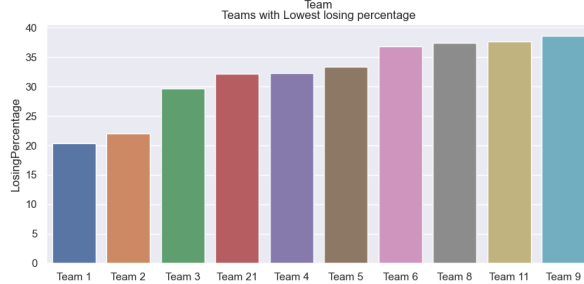
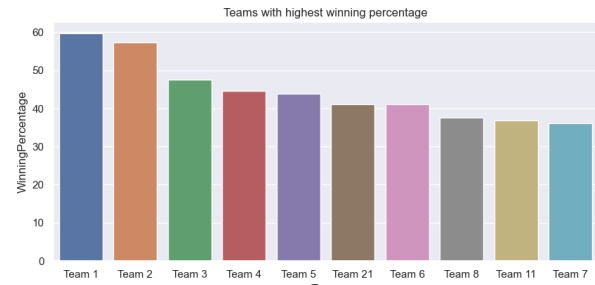
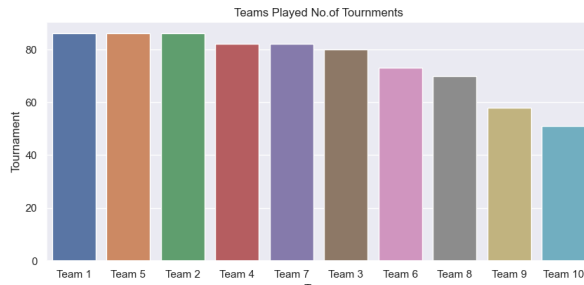
```
ax[1,0].set_title('Teams with Lowest losing percentage')
```

```
ax[1,1].set_title('Teams with highest basket scores per Match')
```

```
ax[2,0].set_title('Teams with most basket given per Match')
```

```
ax[2,1].set_title('Teams Best Defenders')
```

```
plt.show()
```



```
dfOlderTeam=df[df['DebutYear'] < 1950]
```

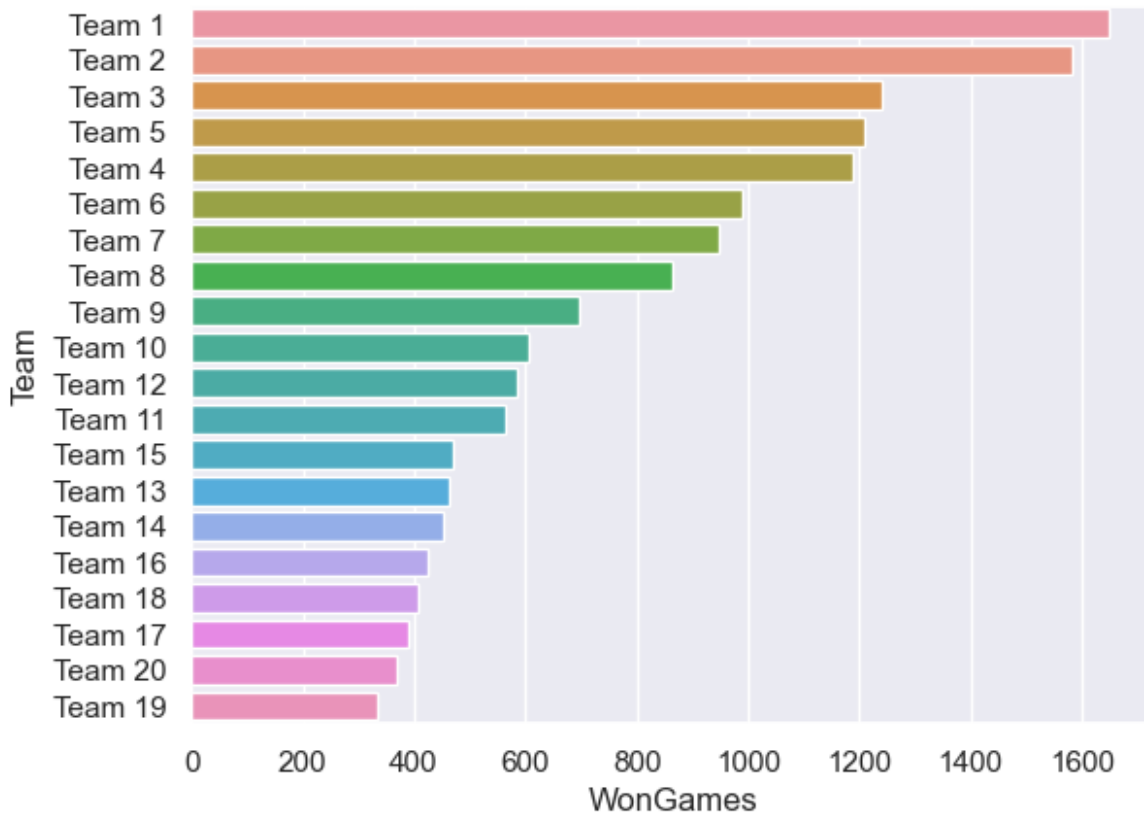
```
dfEmergingTeam=df[df['DebutYear'] >1995]
```

```
#details of the top 20 winning teams, also the teams with highest score
```

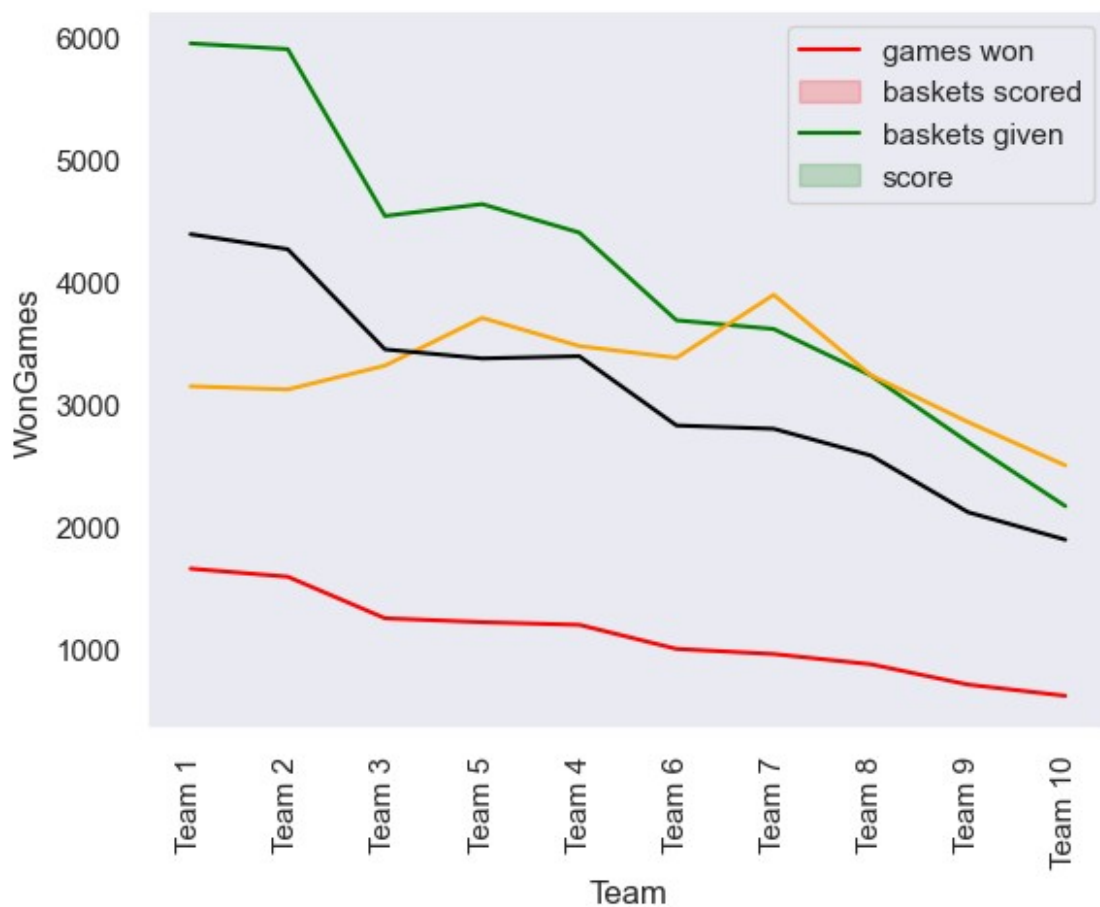
```
df_wongames= df.nlargest(20, ['WonGames'])
```

```
sns.barplot(y=df_wongames['Team'], x= df_wongames['WonGames'])
```

```
plt.show()
```

```
#we will find if the teams with max number of matches won also has the
max number of baskets scored
sns.lineplot(x='Team', y='WonGames', data=df_wongames.head(10),
color='red')
sns.lineplot(x='Team', y='BasketScored', data=df_wongames.head(10),
color='green')
sns.lineplot(x='Team', y='BasketGiven', data=df_wongames.head(10),
color='orange')
sns.lineplot(x='Team', y='Score', data=df_wongames.head(10),
color='black')
plt.grid()
plt.xticks(rotation=90)
plt.legend(labels=['games won', 'baskets scored', 'baskets given',
'score'])
plt.show()
```



```
df[df['BasketsScoredPerMatch'] > 2.0]
```

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames
LostGames \						
0	Team 1	86	4385	2762	1647	552
563						
1	Team 2	86	4262	2762	1581	573
608						
51	Team 52	4	56	72	21	14
37						

	BasketScored	BasketGiven	TournamentChampion	Runner-up
TeamLaunch \				
0	5947	3140	33	23
1929				
1	5900	3114	25	25
1929				
51	153	184	0	0
1929				

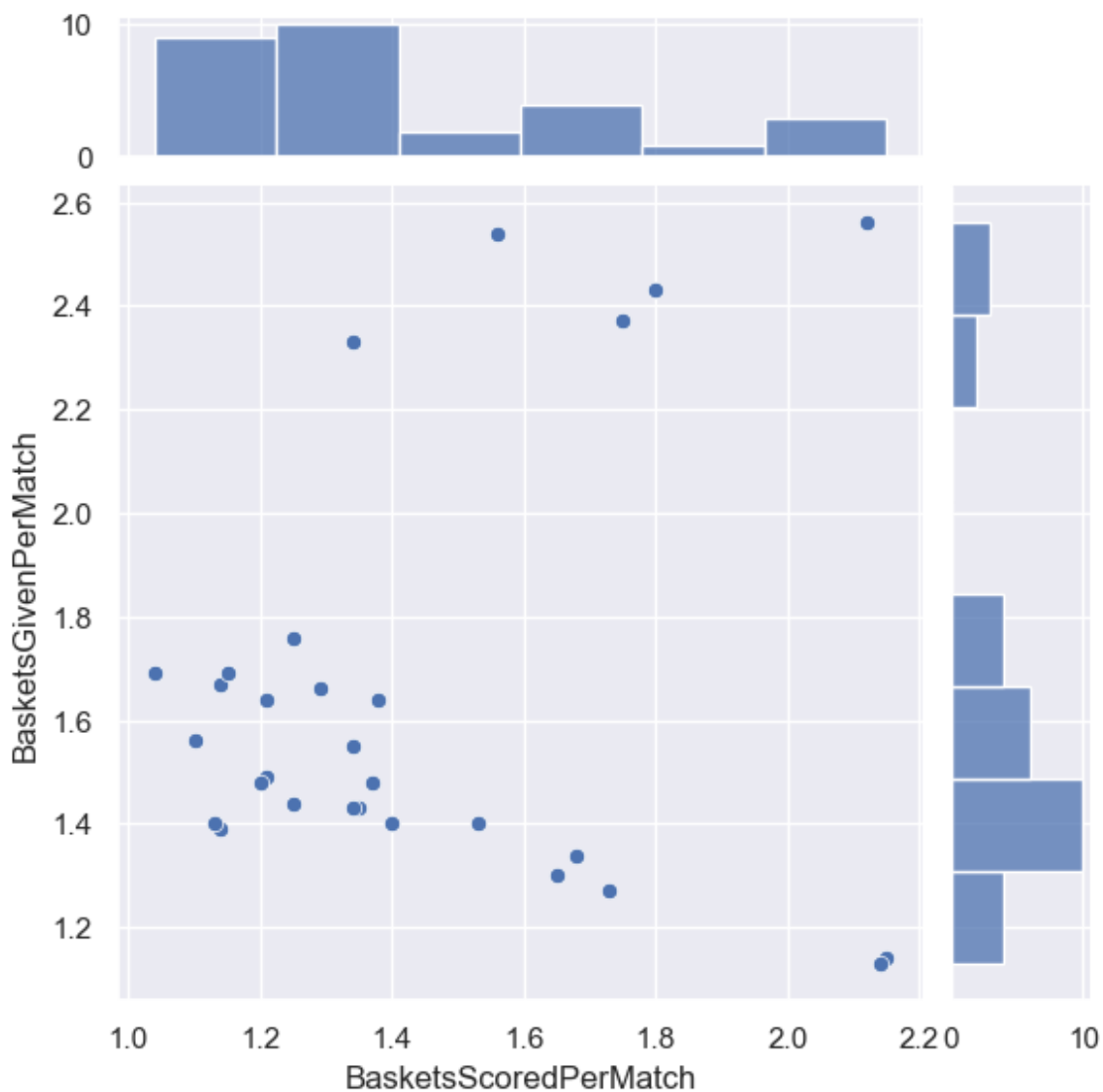
	HighestPositionHeld	DebutYear	Defenders
BasketsScoredPerMatch \			

0	1	1929	2807	2.15
1	1	1929	2786	2.14
51	6	1929	-31	2.12

	BasketsGivenPerMatch	WinningPercentage	LosingPercentage
0	1.14	59.630702	20.383780
1	1.13	57.241130	22.013034
51	2.56	29.166667	51.388889

```
sns.jointplot(data=dfOlderTeam, x='BasketsScoredPerMatch',
y='BasketsGivenPerMatch', marginal_ticks=True)
```

```
<seaborn.axisgrid.JointGrid at 0x1f1a93c6fd0>
```



```
sns.jointplot(data=dfEmergingTeam,x='BasketsScoredPerMatch',
y='BasketsGivenPerMatch', marginal_ticks=True)
```

```
dfEmergingTeam[dfEmergingTeam['BasketsScoredPerMatch'] > 1.1]
```

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames
20	Team 21	17	970	646	266	172	
24	Team 25	12	553	456	147	112	
43	Team 44	3	132	114	35	27	

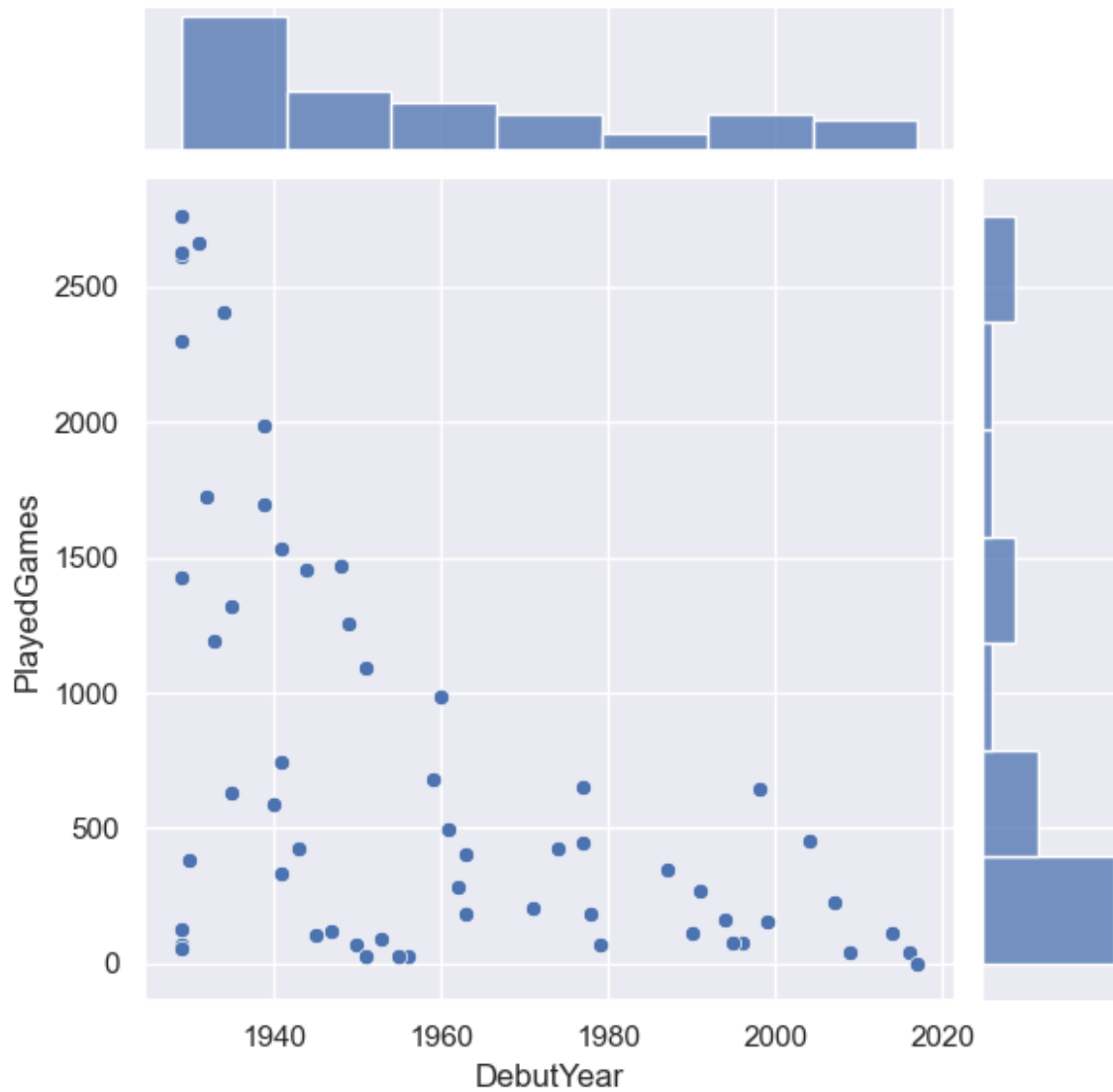
	BasketScored	BasketGiven	TournamentChampion	Runner-up	
20	892	789	0	1	1998-
24	520	633	0	0	
43	139	167	0	0	2014-

	HighestPositionHeld	DebutYear	Defenders	
20	2	1998	103	1.38
24	6	2004	-113	1.14
43	10	2014	-28	1.22

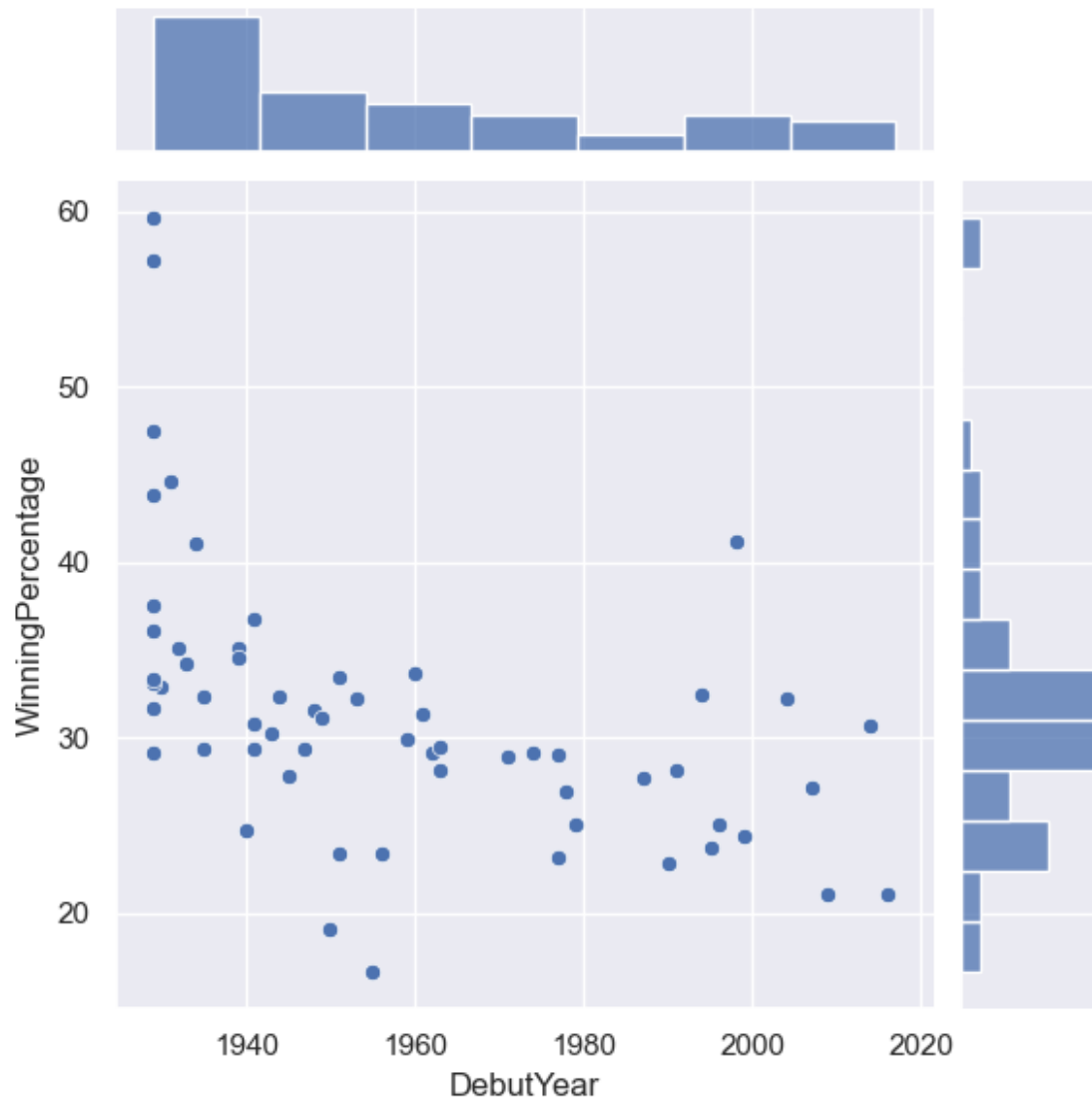
	BasketsGivenPerMatch	WinningPercentage	LosingPercentage
20	1.22	41.176471	32.198142
24	1.39	32.236842	43.201754
43	1.46	30.701754	45.614035

```
sns.jointplot(data=df, x='DebutYear', y='PlayedGames')
```

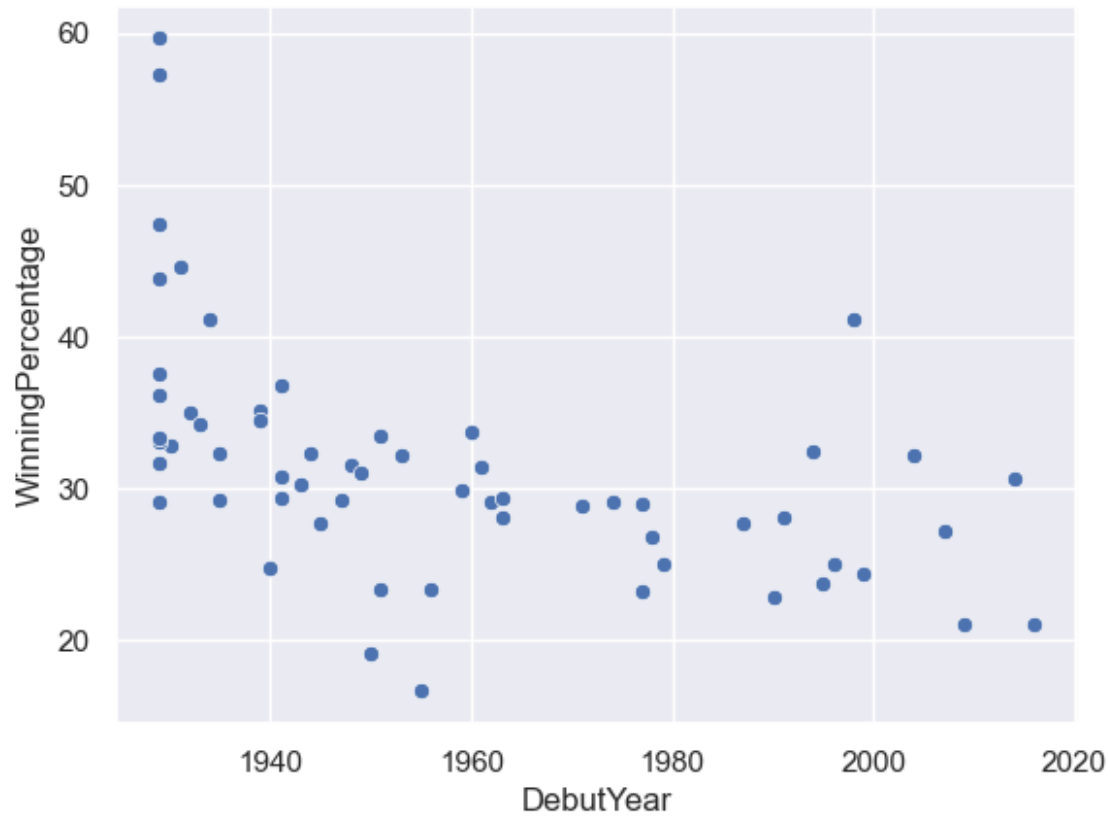
```
<seaborn.axisgrid.JointGrid at 0x1f1a8cf6fd0>
```



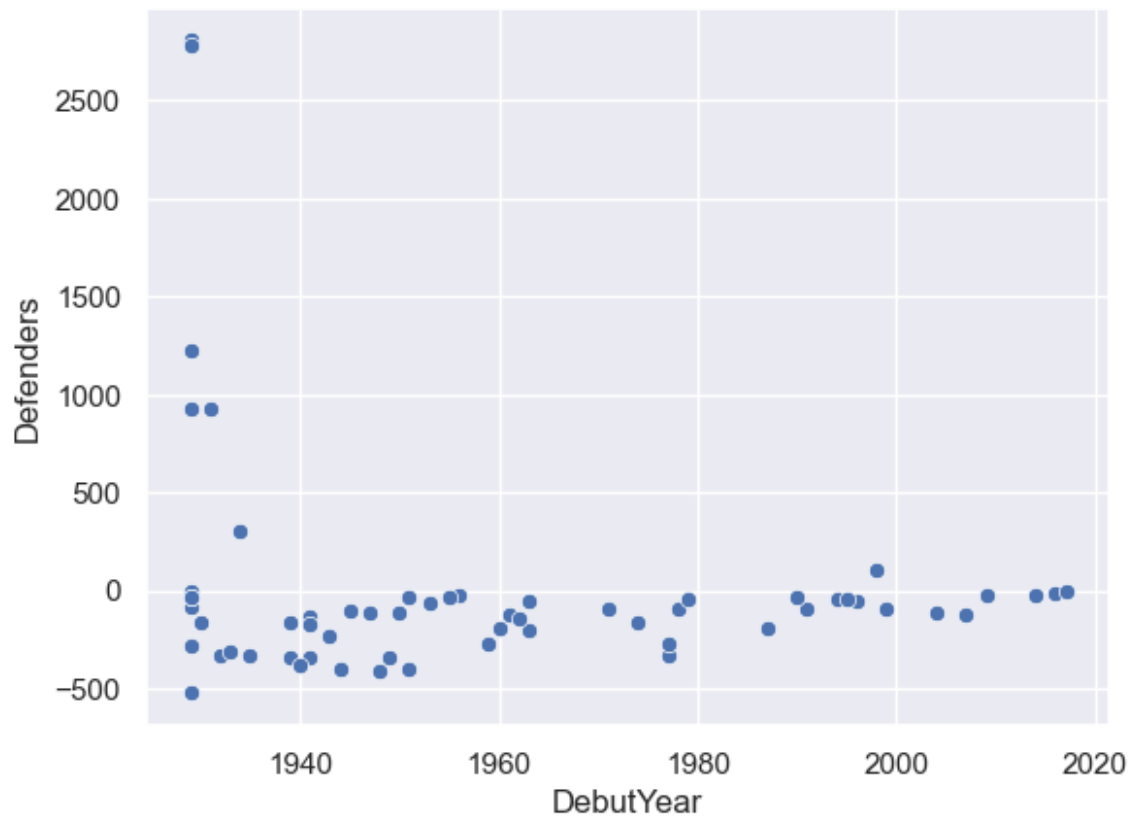
```
sns.jointplot(data=df, x='DebutYear', y='WinningPercentage')  
<seaborn.axisgrid.JointGrid at 0x1f1a94f0990>
```



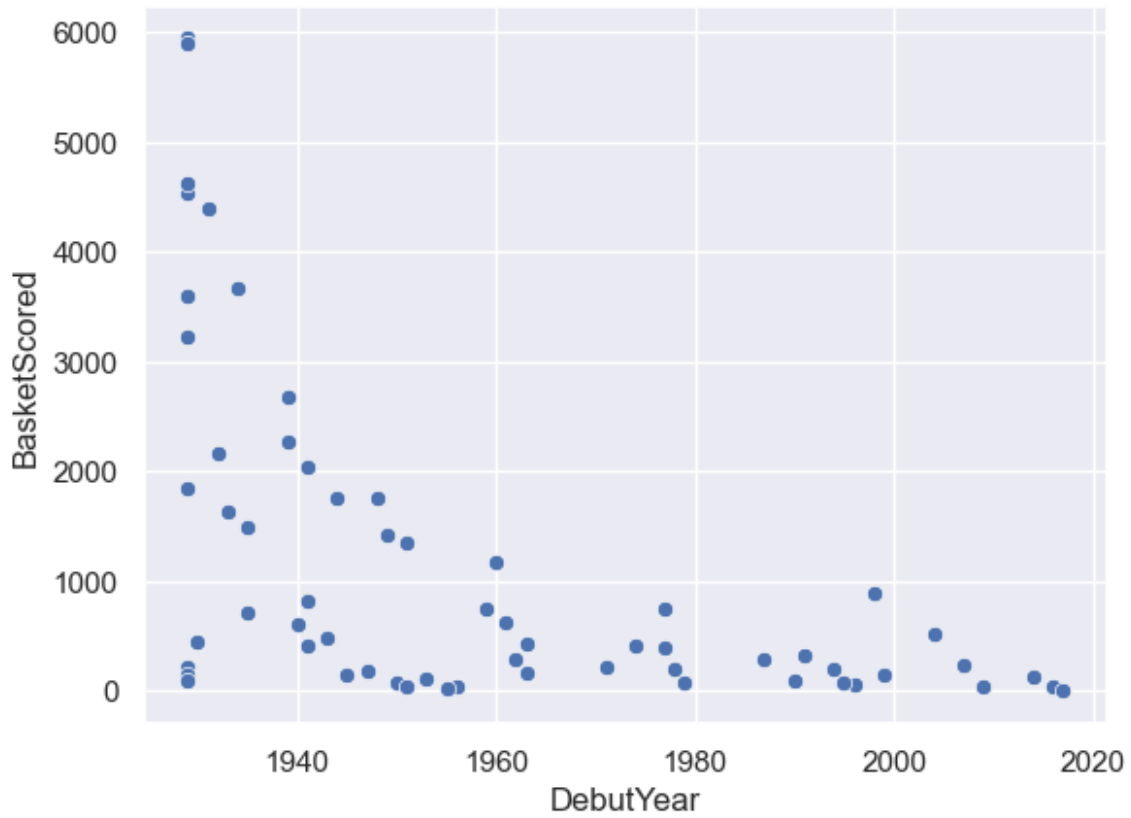
```
sns.scatterplot(data=df,x='DebutYear', y='WinningPercentage')  
<Axes: xlabel='DebutYear', ylabel='WinningPercentage'>
```



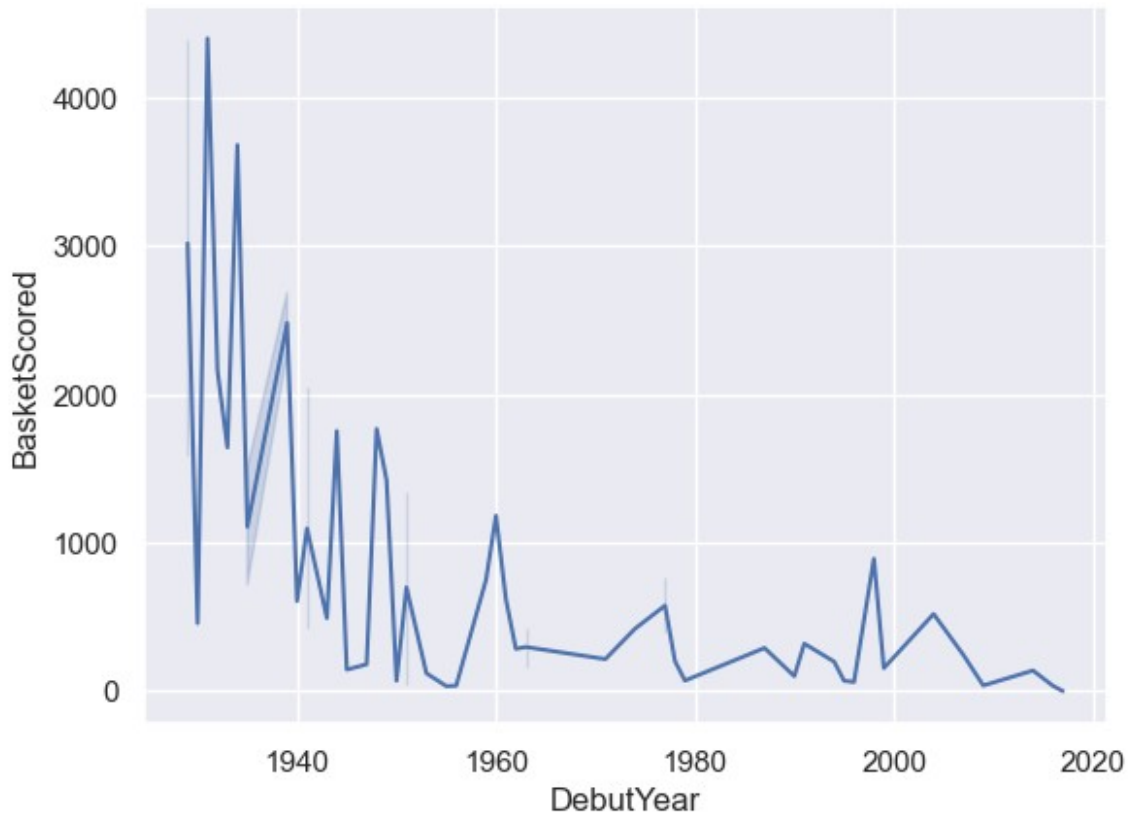
```
sns.scatterplot(data=df,x='DebutYear', y='Defenders')  
<Axes: xlabel='DebutYear', ylabel='Defenders'>
```



```
sns.scatterplot(data=df,x='DebutYear', y='BasketScored')  
<Axes: xlabel='DebutYear', ylabel='BasketScored'>
```

```
sns.lineplot(x='DebutYear',y='BasketScored', data=df)  
<Axes: xlabel='DebutYear', ylabel='BasketScored'>
```



```
(df.loc[:, ~df.columns.isin(['Team', 'TeamLaunch'])]).corr()
```

	Tournament	Score	PlayedGames	WonGames	\
Tournament	1.000000	0.981441	0.998677	0.971954	
Score	0.981441	1.000000	0.979824	0.997240	
PlayedGames	0.998677	0.979824	1.000000	0.967641	
WonGames	0.971954	0.997240	0.967641	1.000000	
DrawnGames	0.989387	0.958452	0.994053	0.939844	
LostGames	0.949863	0.880040	0.956503	0.852785	
BasketScored	0.975059	0.996656	0.970127	0.999318	
BasketGiven	0.987828	0.944263	0.990129	0.926234	
TournamentChampion	0.589072	0.714280	0.574716	0.752204	
Runner-up	0.645876	0.764278	0.632921	0.797350	
HighestPositionHeld	-0.708071	-0.669775	-0.710876	-0.648628	
DebutYear	-0.603151	-0.543432	-0.587451	-0.543854	
Defenders	0.534387	0.673560	0.517486	0.715343	
BasketsScoredPerMatch	0.533277	0.588661	0.511234	0.610586	
BasketsGivenPerMatch	-0.474015	-0.492364	-0.493682	-0.473987	
WinningPercentage	0.819559	0.877385	0.813179	0.884278	
LosingPercentage	-0.761297	-0.809566	-0.767913	-0.802821	
	DrawnGames	LostGames	BasketScored		
BasketGiven \					
Tournament	0.989387	0.949863	0.975059		

0.987828				
Score	0.958452	0.880040	0.996656	
0.944263				
PlayedGames	0.994053	0.956503	0.970127	
0.990129				
WonGames	0.939844	0.852785	0.999318	
0.926234				
DrawnGames	1.000000	0.973156	0.942668	
0.992579				
LostGames	0.973156	1.000000	0.859169	
0.985275				
BasketScored	0.942668	0.859169	1.000000	
0.931548				
BasketGiven	0.992579	0.985275	0.931548	
1.000000				
TournamentChampion	0.503472	0.325462	0.743144	
0.471824				
Runner-up	0.566269	0.393211	0.790532	
0.533814				
HighestPositionHeld	-0.719845	-0.725149	-0.649455	-
0.726610				
DebutYear	-0.577616	-0.600513	-0.553453	-
0.615715				
Defenders	0.443541	0.247401	0.706723	
0.401089				
BasketsScoredPerMatch	0.460700	0.360511	0.615872	
0.465960				
BasketsGivenPerMatch	-0.508538	-0.464752	-0.462492	-
0.462700				
WinningPercentage	0.774416	0.665366	0.879124	
0.757279				
LosingPercentage	-0.757461	-0.655055	-0.795043	-
0.717092				

	TournamentChampion	Runner-up	
HighestPositionHeld \			
Tournament	0.589072	0.645876	-
0.708071			
Score	0.714280	0.764278	-
0.669775			
PlayedGames	0.574716	0.632921	-
0.710876			
WonGames	0.752204	0.797350	-
0.648628			
DrawnGames	0.503472	0.566269	-
0.719845			
LostGames	0.325462	0.393211	-
0.725149			
BasketScored	0.743144	0.790532	-

0.649455			
BasketGiven	0.471824	0.533814	-
0.726610			
TournamentChampion	1.000000	0.971552	-
0.305397			
Runner-up	0.971552	1.000000	-
0.360408			
HighestPositionHeld	-0.305397	-0.360408	
1.000000			
DebutYear	-0.284878	-0.315881	
0.588914			
Defenders	0.954126	0.952892	-
0.222366			
BasketsScoredPerMatch	0.619698	0.640952	-
0.465669			
BasketsGivenPerMatch	-0.303812	-0.331376	
0.390923			
WinningPercentage	0.765351	0.801247	-
0.737288			
LosingPercentage	-0.638581	-0.677121	
0.661382			

	DebutYear	Defenders	BasketsScoredPerMatch	\
Tournament	-0.603151	0.534387	0.533277	
Score	-0.543432	0.673560	0.588661	
PlayedGames	-0.587451	0.517486	0.511234	
WonGames	-0.543854	0.715343	0.610586	
DrawnGames	-0.577616	0.443541	0.460700	
LostGames	-0.600513	0.247401	0.360511	
BasketScored	-0.553453	0.706723	0.615872	
BasketGiven	-0.615715	0.401089	0.465960	
TournamentChampion	-0.284878	0.954126	0.619698	
Runner-up	-0.315881	0.952892	0.640952	
HighestPositionHeld	0.588914	-0.222366	-0.465669	
DebutYear	1.000000	-0.196282	-0.603446	
Defenders	-0.196282	1.000000	0.644764	
BasketsScoredPerMatch	-0.603446	0.644764	1.000000	
BasketsGivenPerMatch	-0.185772	-0.266535	0.221075	
WinningPercentage	-0.491259	0.742541	0.720084	
LosingPercentage	0.201751	-0.609304	-0.379143	

	BasketsGivenPerMatch	WinningPercentage	\
Tournament	-0.474015	0.819559	
Score	-0.492364	0.877385	
PlayedGames	-0.493682	0.813179	
WonGames	-0.473987	0.884278	
DrawnGames	-0.508538	0.774416	
LostGames	-0.464752	0.665366	
BasketScored	-0.462492	0.879124	

BasketGiven	-0.462700	0.757279
TournamentChampion	-0.303812	0.765351
Runner-up	-0.331376	0.801247
HighestPositionHeld	0.390923	-0.737288
DebutYear	-0.185772	-0.491259
Defenders	-0.266535	0.742541
BasketsScoredPerMatch	0.221075	0.720084
BasketsGivenPerMatch	1.000000	-0.442562
WinningPercentage	-0.442562	1.000000
LosingPercentage	0.766499	-0.851444

	LosingPercentage
Tournament	-0.761297
Score	-0.809566
PlayedGames	-0.767913
WonGames	-0.802821
DrawnGames	-0.757461
LostGames	-0.655055
BasketScored	-0.795043
BasketGiven	-0.717092
TournamentChampion	-0.638581
Runner-up	-0.677121
HighestPositionHeld	0.661382
DebutYear	0.201751
Defenders	-0.609304
BasketsScoredPerMatch	-0.379143
BasketsGivenPerMatch	0.766499
WinningPercentage	-0.851444
LosingPercentage	1.000000

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x1f1a95f2810>
```

