

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
print('Libraries imported successfully')
```

Libraries imported successfully

```
In [3]: df1 = pd.read_csv("C:\\Users\\ABC\\Desktop\\Basketball.csv")
df1.head()
```

```
Out[3]:
```

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames	BasketScored	BasketGiven
0	Team 1	86	4385	2762	1647	552	563	5947	
1	Team 2	86	4262	2762	1581	573	608	5900	
2	Team 3	80	3442	2614	1241	598	775	4534	
3	Team 4	82	3386	2664	1187	616	861	4398	
4	Team 5	86	3368	2762	1209	633	920	4631	

```
In [4]: df1.shape
```

```
Out[4]: (61, 13)
```

```
In [5]: df1.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
```

```
In [6]: df1.describe(include='all')
```

```
Out[6]:
```

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames	BasketScored
count	61	61.000000	61	61	61	61	61	61
unique	61	NaN	61	53	59	57	56	60
top	Team 1	NaN	4385	2762	7	14	37	70
freq	1	NaN	1	3	2	2	3	2
mean	NaN	24.000000	NaN	NaN	NaN	NaN	NaN	NaN
std	NaN	26.827225	NaN	NaN	NaN	NaN	NaN	NaN
min	NaN	1.000000	NaN	NaN	NaN	NaN	NaN	NaN
25%	NaN	4.000000	NaN	NaN	NaN	NaN	NaN	NaN
50%	NaN	12.000000	NaN	NaN	NaN	NaN	NaN	NaN
75%	NaN	38.000000	NaN	NaN	NaN	NaN	NaN	NaN
max	NaN	86.000000	NaN	NaN	NaN	NaN	NaN	NaN

```
In [7]: df1.tail()
```

```
Out[7]:
```

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames	BasketScored	Ba
56	Team 57	1	34	38	8	10	20	38	
57	Team 58	1	22	30	7	8	15	37	
58	Team 59	1	19	30	7	5	18	51	
59	Team 60	1	14	30	5	4	21	34	
60	Team 61	1	-	-	-	-	-	-	

```
In [8]: df1=df1.replace('-',0)  
df1.tail()
```

Out[8]:

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames	BasketScored	BasketGiven
56	Team 57	1	34	38	8	10	20	38	38
57	Team 58	1	22	30	7	8	15	37	37
58	Team 59	1	19	30	7	5	18	51	51
59	Team 60	1	14	30	5	4	21	34	34
60	Team 61	1	0	0	0	0	0	0	0

```
In [9]: df1['TeamLaunch'] = df1['TeamLaunch'].str[:4]
df1['TeamLaunch']
```

```
Out[9]: 0    1929
1    1929
2    1929
3    1931
4    1929
...
56   2009
57   1956
58   1951
59   1955
60   2017
Name: TeamLaunch, Length: 61, dtype: object
```

```
In [10]: basketball = df1.astype("int64", errors='ignore')
basketball.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    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    int64
12  HighestPositionHeld   61 non-null    int64
dtypes: int64(12), object(1)
memory usage: 6.3+ KB
```

```
In [11]: basketball.describe()
```

Out[11]:

	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames	BasketScored
count	61.000000	61.000000	61.000000	61.000000	61.000000	61.000000	61.000000
mean	24.000000	901.426230	796.819672	303.967213	188.934426	303.754098	1140.344262
std	26.827225	1134.899121	876.282765	406.991030	201.799477	294.708594	1506.740211
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	4.000000	96.000000	114.000000	34.000000	24.000000	62.000000	153.000000
50%	12.000000	375.000000	423.000000	123.000000	95.000000	197.000000	430.000000
75%	38.000000	1351.000000	1318.000000	426.000000	330.000000	563.000000	1642.000000
max	86.000000	4385.000000	2762.000000	1647.000000	633.000000	1070.000000	5947.000000

In [12]:

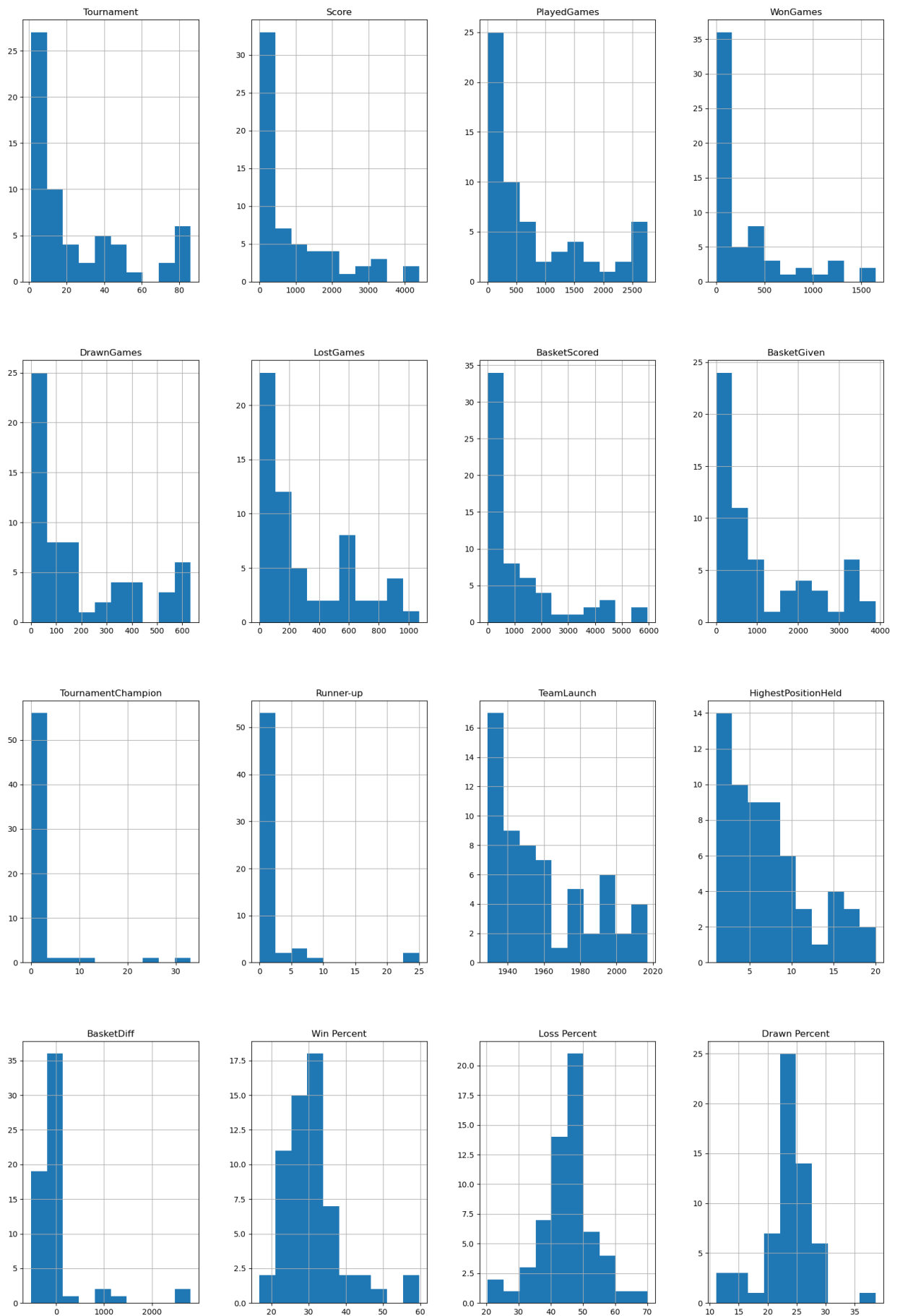
```
basketball['IfWinner']=basketball['TournamentChampion']>0
basketball['BasketDiff']=basketball['BasketScored']-basketball['BasketGiven']
basketball['NetScorer']=basketball['BasketDiff']>0
basketball['Win Percent']=round((basketball['WonGames']/basketball['PlayedGames'])*100)
basketball['Loss Percent']=round((basketball['LostGames']/basketball['PlayedGames'])*100)
basketball['Drawn Percent']=round((basketball['DrawnGames']/basketball['PlayedGames'])*100)
basketball['NetWinner']=(basketball['Win Percent']>basketball['Loss Percent']) & (basketball['NetScorer'])
basketball.head()
```

Out[12]:

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames	BasketScored	BasketGiven
0	Team 1		86	4385	2762	1647	552	563	5947
1	Team 2		86	4262	2762	1581	573	608	5900
2	Team 3		80	3442	2614	1241	598	775	4534
3	Team 4		82	3386	2664	1187	616	861	4398
4	Team 5		86	3368	2762	1209	633	920	4631

In [13]:

```
basketball.hist(figsize=(20,30));
```

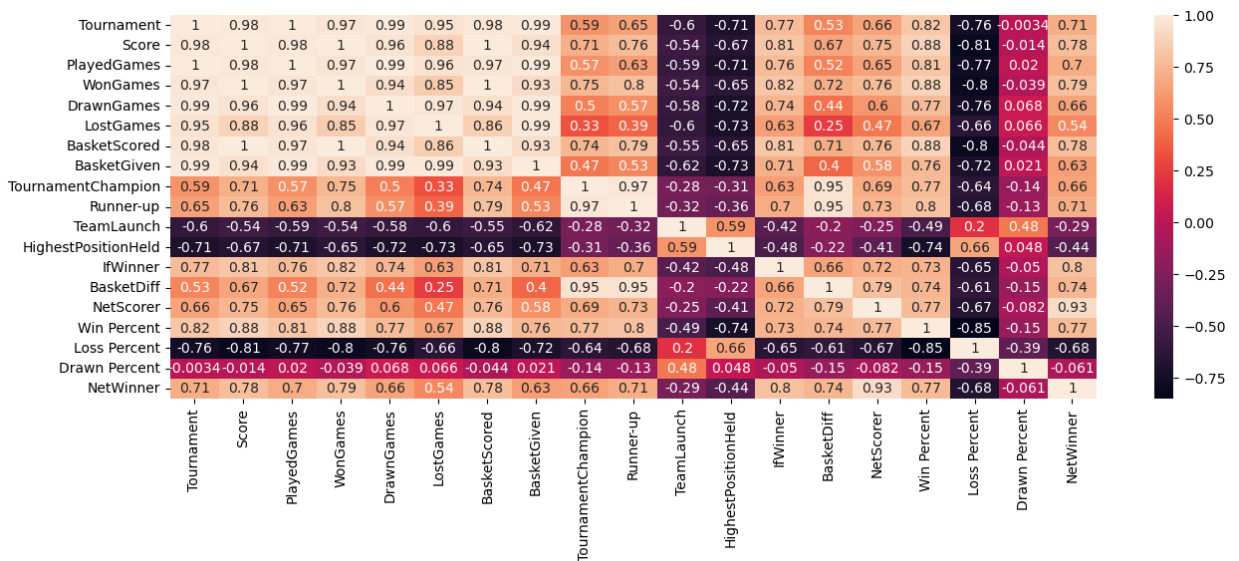


```
In [14]: corr_mat=basketball.corr()
fig, ax = plt.subplots(figsize=(15,5))
```

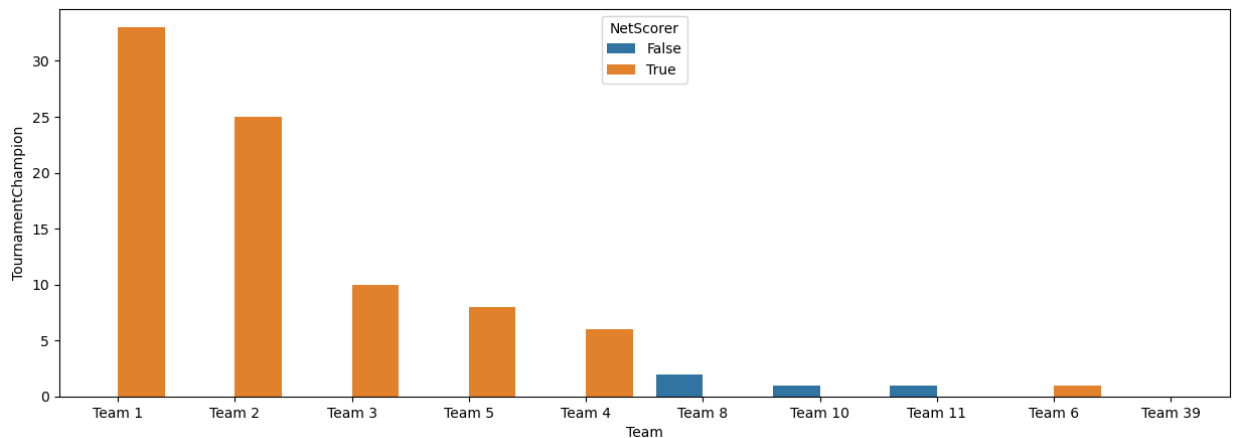
```
sns.heatmap(corr_mat,annot=True);
```

C:\Users\ABC\AppData\Local\Temp\ipykernel_18044\3556082604.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

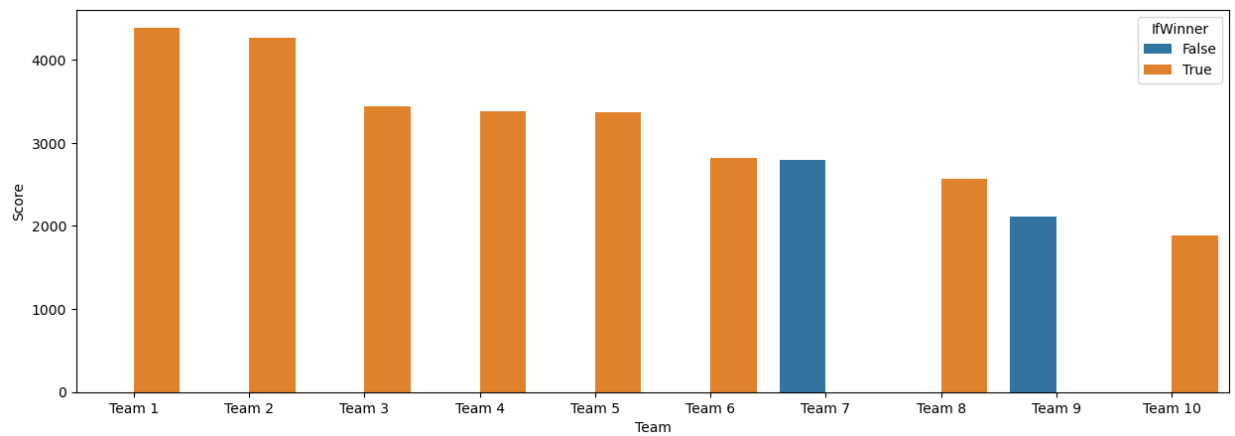
```
corr_mat=basketball.corr()
```



```
In [15]: temp = basketball.sort_values('TournamentChampion',ascending=False).head(10)
fig, ax = plt.subplots(figsize=(15,5))
sns.barplot(x="Team", y = 'TournamentChampion',data=temp,hue='NetScorer');
```



```
In [16]: temp = basketball.sort_values('Score',ascending=False).head(10)
fig, ax = plt.subplots(figsize=(15,5))
sns.barplot(x="Team", y = 'Score',data=temp,hue='IfWinner');
```



```
In [17]: basketball.sort_values('TeamLaunch',ascending=True).head(1)
```

```
Out[17]:
```

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames	BasketScored	Ba
0	Team 1	86	4385	2762	1647	552	563	5947	

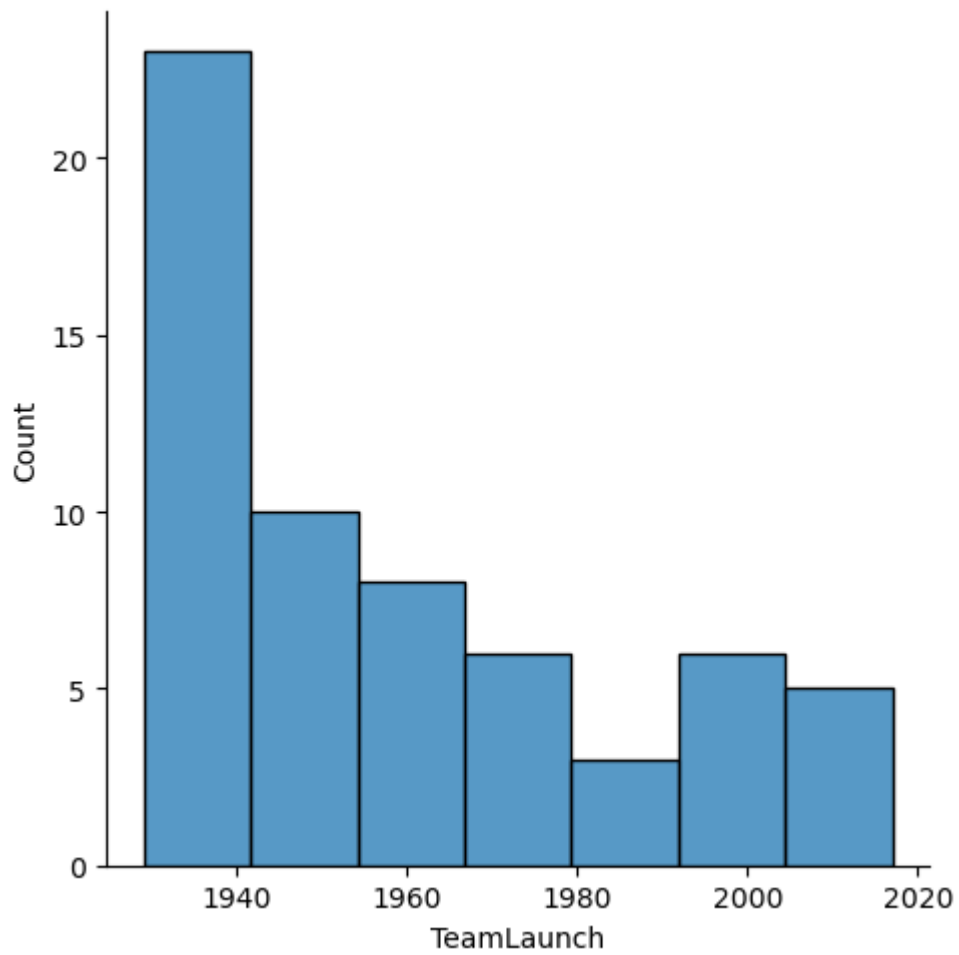
```
In [18]: basketball.sort_values('TeamLaunch',ascending=False).head(1)
```

```
Out[18]:
```

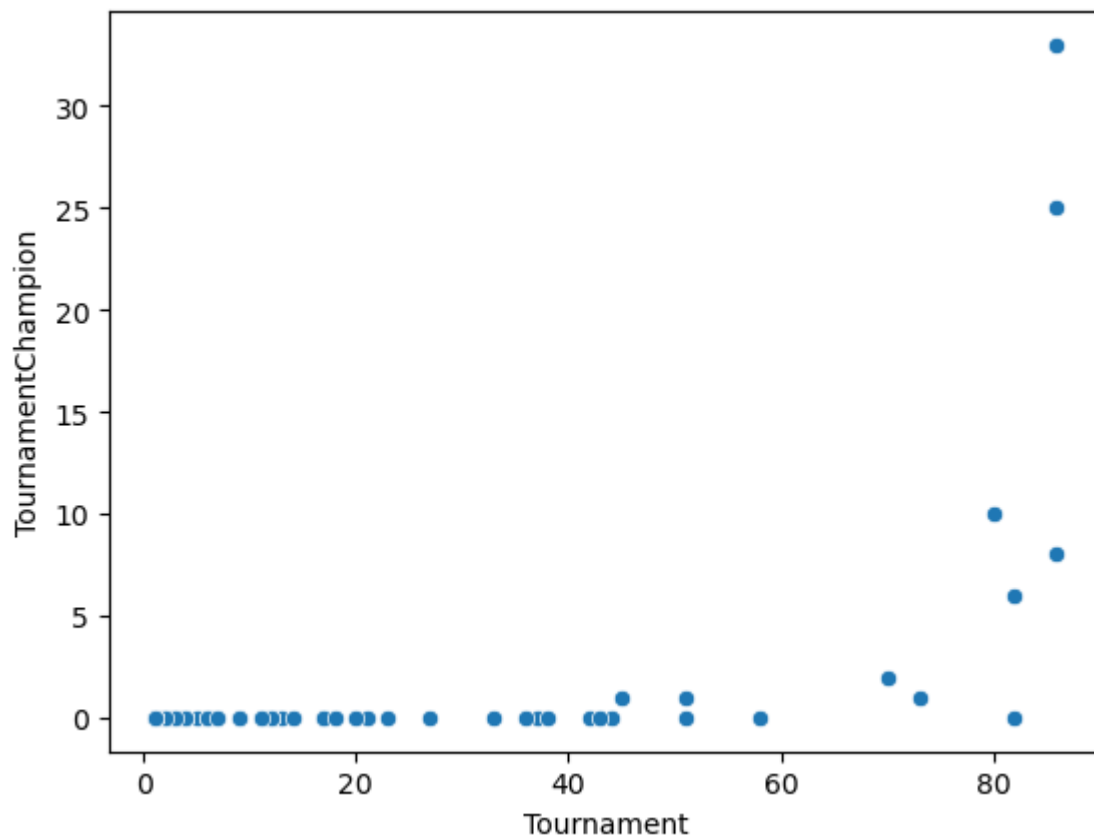
	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames	BasketScored	Ba
60	Team 61	1	0	0	0	0	0	0	

```
In [19]: sns.displot(x='TeamLaunch',data=basketball)
```

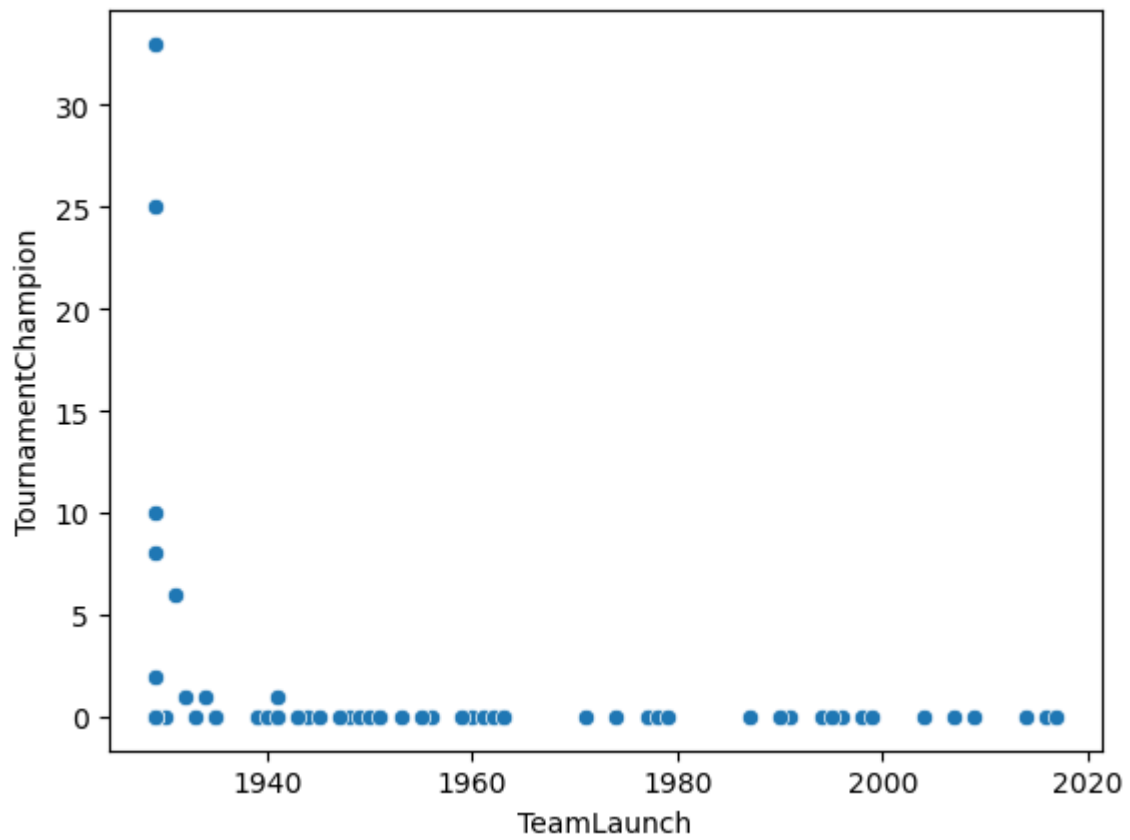
```
Out[19]: <seaborn.axisgrid.FacetGrid at 0x21ce05bd6f0>
```



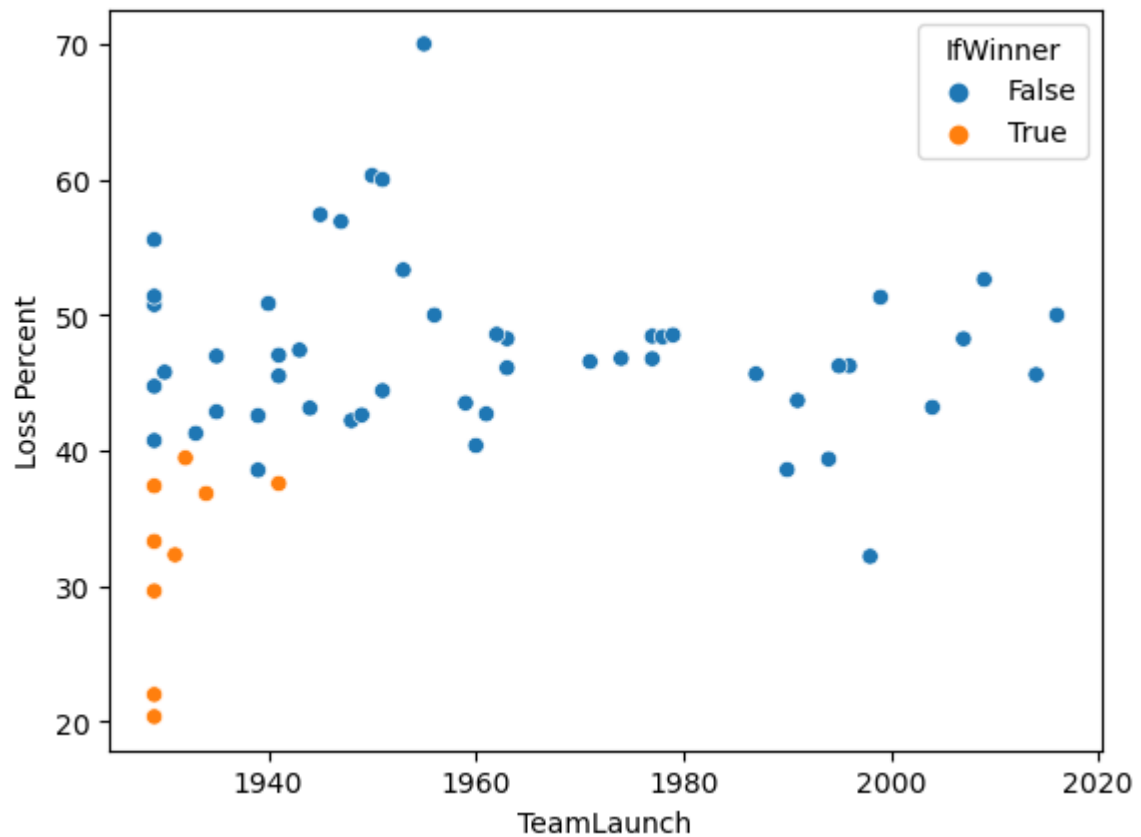
```
In [20]: sns.scatterplot(data=basketball, x="Tournament", y="TournamentChampion");
```



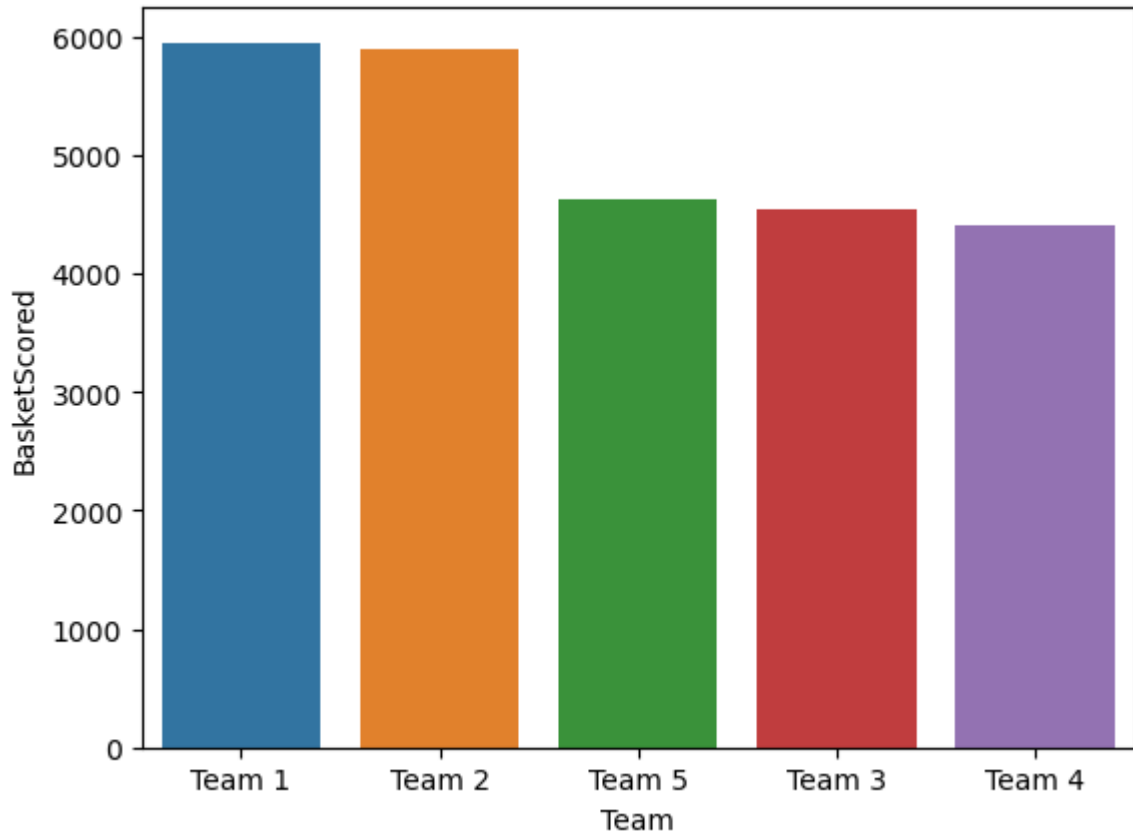

```
In [21]: sns.scatterplot(data=basketball, x="TeamLaunch", y="TournamentChampion");
```



```
In [22]: sns.scatterplot(data=basketball, x="TeamLaunch", y="Loss Percent", hue='IfWinner');
```



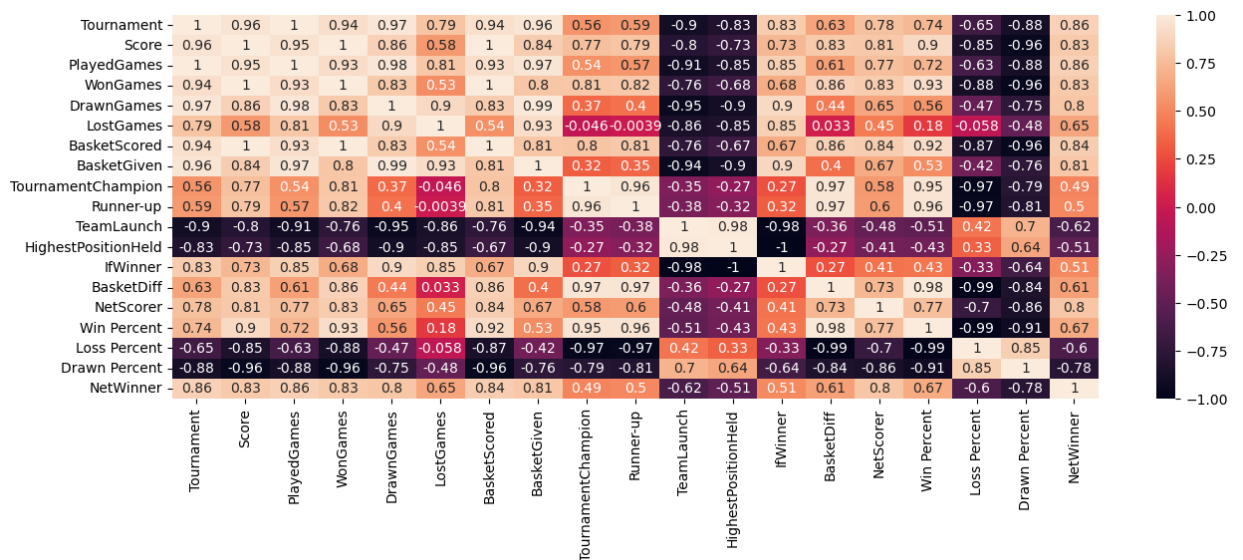
```
In [23]: b = basketball.sort_values('BasketScored',ascending=False).head()
sns.barplot(x="Team", y = 'BasketScored',data=b);
```



```
In [24]: c=basketball.sort_values('TournamentChampion',ascending=False).head(10)
corr_mat_c=c.corr()
fig, ax = plt.subplots(figsize=(15,5))
sns.heatmap(corr_mat_c,annot=True);
```

C:\Users\ABC\AppData\Local\Temp\ipykernel_18044\2762719022.py:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
corr_mat_c=c.corr()
```



```
In [25]: c[['BasketDiff', 'Win Percent', 'Loss Percent', 'Runner-up', 'WonGames']].min()
```

```
Out[25]: BasketDiff      -333.00  
Win Percent      32.50  
Loss Percent     20.38  
Runner-up        0.00  
WonGames         52.00  
dtype: float64
```

```
In [26]: d = basketball.sort_values('TournamentChampion', ascending=False)  
d = d[10:]  
d = d.loc[(d['BasketDiff'] >= -333) & (d['Win Percent'] >= 32.5) & (d['Runner-up'] >= 0) & (d['Loss Percent'] <= 20.38)]  
d.sort_values('TeamLaunch', ascending=False)
```

```
Out[26]:
```

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames	BasketScored	Ba
--	------	------------	-------	-------------	----------	------------	-----------	--------------	----

20	Team 21	17	970	646	266	172	208	892	
18	Team 19	27	1148	988	333	256	399	1182	
8	Team 9	58	2109	1986	698	522	766	2683	
17	Team 18	38	1174	1192	408	292	492	1642	
28	Team 29	12	421	380	125	81	174	458	
6	Team 7	82	2792	2626	948	608	1070	3609	

```
In [27]: d = d.loc[(d['BasketDiff'] >= -333) & (d['Win Percent'] >= 32.5) & (d['Runner-up'] >= 0) & (d['Loss Percent'] <= 20.38)]  
d.sort_values('Loss Percent', ascending=True)
```

```
Out[27]:
```

	Team	Tournament	Score	PlayedGames	WonGames	DrawnGames	LostGames	BasketScored	Ba
--	------	------------	-------	-------------	----------	------------	-----------	--------------	----

20	Team 21	17	970	646	266	172	208	892	
8	Team 9	58	2109	1986	698	522	766	2683	

```
In [ ]: Quality - There aren't enough categorical variables in the data. Region of a particular  
Quantity - What was the team ranking/highest position held in each year since the tournament  
Variety - What factors contribute to winning tournament championships? Is team budget  
Velocity - We know from the above analysis that older teams have won more championships  
Veracity - Source of dataset has not been given for verification. If a source is given  
Age - What is the mean age of teams that have won the championships? Is age a factor in  
Which teams are already in contract with the Company X is not given. This data should
```