finalproject

May 11, 2023

1 The Effect of Per Game Stats on the Modern Era NBA Teams Win Percentage

Nick Van, Daniel Taylor, Eric Feng, Winston Tan

2 Introduction

NBA is a professional basketball league where 30 teams compete in a series of 82 games every year with the goal of winning as many games as possible. A team wins when they outscore the other team, so some might be quick to say that stats related to scoring such as shooting percentage are the ones that will dictate if a team is successful or not. In this project, we will take a look at if this really is the case, or if other stats such as the number of rebounds (balls retrieved after a missed shot), fouls per game or even shots blocked have a greater impact on the result of the game. Figuring out which statistics matter the most can help teams scout for players who #excel in those aspects, help fans win in the sports betting industry, or even help teams figure out what they need to change in general.

[253]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

3 Data Collection

[254]:
```
headers = {
'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.
↪36 (KHTML, like Gecko) Chrome/109.0.0.0 Safari/537.36',
    'From': 'randomemail@gmail.com'
}
```

We will grab the advanced stats from the past 10 years (which is the modern era of basketball) which contains all the data, even though some of which are unnecessary for our analysis. We scraped the data from basketball-reference.com which is one of the leading databases for sports statistics.

After obtaining the data we must iterate over the rows and cells of the data to place them into a readable table that is not just HTML.

[165]:
```
# Obtain the data from a table via the URL, using pandas and beautifulsoup
# we are able to extract a table in HTML form which we will later have to␣
↪iterate through
# the rows and cells to place into a dataframe.
url = 'https://www.basketball-reference.com/leagues/NBA_2022.html#advanced-team'
res = requests.get(url)
soup = BeautifulSoup(res.content, 'html.parser')

table22 = soup.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = table22.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in table22.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)

# create a pandas dataframe from the extracted data
df22 = pd.DataFrame(data, columns=columns)

# We will repeat this 9 more times

url = 'https://www.basketball-reference.com/leagues/NBA_2021.html#advanced-team'
res = requests.get(url)
soup = BeautifulSoup(res.content, 'html.parser')

table21 = soup.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = table21.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in table21.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
```

```python
    if row_data:
        data.append(row_data)

# create a pandas dataframe from the extracted data
df21 = pd.DataFrame(data, columns=columns)


url = 'https://www.basketball-reference.com/leagues/NBA_2020.html#advanced-team'
res = requests.get(url)
soup = BeautifulSoup(res.content, 'html.parser')

table20 = soup.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = table20.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in table20.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)

# create a pandas dataframe from the extracted data
df20 = pd.DataFrame(data, columns=columns)


url = 'https://www.basketball-reference.com/leagues/NBA_2019.html#advanced-team'
res = requests.get(url)
soup = BeautifulSoup(res.content, 'html.parser')

table19 = soup.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = table19.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in table19.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
```

```python
        data.append(row_data)

# create a pandas dataframe from the extracted data
df19 = pd.DataFrame(data, columns=columns)


url = 'https://www.basketball-reference.com/leagues/NBA_2018.html#advanced-team'
res = requests.get(url)
soup = BeautifulSoup(res.content, 'html.parser')

table18 = soup.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = table18.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in table18.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)

# create a pandas dataframe from the extracted data
df18 = pd.DataFrame(data, columns=columns)


url = 'https://www.basketball-reference.com/leagues/NBA_2017.html#advanced-team'
res = requests.get(url)
soup = BeautifulSoup(res.content, 'html.parser')

table17 = soup.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = table17.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in table17.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)
```

```python
# create a pandas dataframe from the extracted data
df17 = pd.DataFrame(data, columns=columns)


url = 'https://www.basketball-reference.com/leagues/NBA_2016.html#advanced-team'
res = requests.get(url)
soup = BeautifulSoup(res.content, 'html.parser')

table16 = soup.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = table16.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in table16.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)

# create a pandas dataframe from the extracted data
df16 = pd.DataFrame(data, columns=columns)


url = 'https://www.basketball-reference.com/leagues/NBA_2015.html#advanced-team'
res = requests.get(url)
soup = BeautifulSoup(res.content, 'html.parser')

table15 = soup.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = table15.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in table15.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)
```

```
# create a pandas dataframe from the extracted data
df15 = pd.DataFrame(data, columns=columns)


url = 'https://www.basketball-reference.com/leagues/NBA_2014.html#advanced-team'
res = requests.get(url)
soup = BeautifulSoup(res.content, 'html.parser')

table14 = soup.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = table14.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in table14.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)

# create a pandas dataframe from the extracted data
df14 = pd.DataFrame(data, columns=columns)


url = 'https://www.basketball-reference.com/leagues/NBA_2013.html#advanced-team'
res = requests.get(url)
soup = BeautifulSoup(res.content, 'html.parser')

table13 = soup.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = table13.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in table13.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)

# create a pandas dataframe from the extracted data
```

```
df13 = pd.DataFrame(data, columns=columns)
```

3.1 Combining Dataset

We started by grabbing the advanced stats from the 2022 season and put it in a dataframe. We then repeated this process for the other 9 years, and combined the resulting tables into a single one. This way we can analyze all the stats and trends at once.

[255]:
```
dfs = [df13, df14, df15, df16, df17, df18, df19, df20, df21, df22]
result = pd.concat(dfs, ignore_index=True)
result.head(310)
```

[255]:
```
     Rk                     Team   Age   W   L  PW  PL    MOV    SOS    SRS  \
0     1  Oklahoma City Thunder*  26.0  60  22  64  18   9.21  -0.06   9.15
1     2              Miami Heat*  30.3  66  16  62  20   7.87  -0.84   7.03
2     3   Los Angeles Clippers*  28.8  56  26  59  23   6.45  -0.02   6.43
3     4       San Antonio Spurs*  28.6  58  24  58  24   6.40   0.27   6.67
4     5          Denver Nuggets*  26.1  57  25  55  27   5.09   0.28   5.37
..   ..                      …     …  ..  ..  ..  ..     …      …      …
305  27            Orlando Magic  23.3  22  60  21  61  -8.00   0.33  -7.67
306  28    Oklahoma City Thunder  22.4  24  58  21  61  -8.10   0.20  -7.90
307  29          Houston Rockets  24.1  20  62  21  61  -8.48   0.22  -8.26
308  30  Portland Trail Blazers  25.6  27  55  20  62  -8.88   0.33  -8.55
309               League Average  26.3          41  41   0.00   0.00   0.00

     … FT/FGA    eFG%  TOV%  DRB% FT/FGA                       Arena  \
0    …   .280    .469  13.5  73.4   .197   Chesapeake Energy Arena
1    …   .224    .487  14.8  73.0   .200     AmericanAirlines Arena
2    …   .203    .492  15.4  73.5   .229             STAPLES Center
3    …   .204    .480  13.7  74.9   .179                AT&T Center
4    …   .216    .493  14.3  71.8   .193               Pepsi Center
..   …     … ..    …     …     …   … ..                           …
305  …   .175    .532  11.7  77.2   .196               Amway Center
306  …   .169    .533  11.8  76.1   .169              Paycom Center
307  …   .202    .554  12.3  74.4   .206              Toyota Center
308  …   .188    .559  12.7  76.9   .222                Moda Center
309  …   .192    .532  12.3  76.8   .192

     Attend. Attend./G
0    746,323    18,203
1    819,290    19,983
2    788,293    19,227
3    755,700    18,432
4    730,616    17,820
..         …         …
305  622,881    15,192
306  595,112    14,515
```

```
307  638,977    15,585
308  705,608    17,210
309  695,475    16,963

[310 rows x 31 columns]
```

4 Data Processing

We now need to clean up the Dataframe and organize our data to what we want.

Now we have all the data in one place, however, it is not organized. For this project, we care about which stats affect the win percentage of teams. T The next step we can take to reach this goal is to sort the data from the past 10 years by win percentage.

[256]:
```
result = result.sort_values('W', ascending=False)
result.head(310)
```

[256]:
```
     Rk                      Team   Age   W   L  PW  PL    MOV    SOS    SRS  \
94    2  Golden State Warriors*  27.4  73   9  65  17  10.76  -0.38  10.38
124   1  Golden State Warriors*  28.2  67  15  67  15  11.63  -0.28  11.35
93    1      San Antonio Spurs*  30.3  67  15  67  15  10.63  -0.36  10.28
62    1  Golden State Warriors*  26.6  67  15  65  17  10.10  -0.09  10.01
1     2             Miami Heat*  30.3  66  16  62  20   7.87  -0.84   7.03
..   ..                     …    …  ..  ..  ..   …     …      …
61               League Average  26.6          41  41   0.00   0.00   0.00
185              League Average  26.6          41  41   0.00   0.00   0.00
92               League Average  26.8          41  41   0.00   0.00   0.00
154              League Average  26.6          41  41   0.00   0.00   0.00
309              League Average  26.3          41  41   0.00   0.00   0.00

     … FT/FGA    eFG%  TOV%  DRB% FT/FGA                     Arena  \
94   …   .191    .479  12.6  76.0   .208              Oracle Arena
124  …   .204    .486  13.5  74.9   .198              Oracle Arena
93   …   .197    .477  14.1  79.1   .182               AT&T Center
62   …   .184    .470  14.3  74.5   .217              Oracle Arena
1    …   .224    .487  14.8  73.0   .200   AmericanAirlines Arena
..   …     … ..    …     …     …     … ..                         …
61   …   .215    .501  13.6  74.5   .215
185  …   .193    .521  13.0  77.7   .193
92   …   .205    .496  13.3  74.9   .205
154  …   .209    .514  12.7  76.7   .209
309  …   .192    .532  12.3  76.8   .192

     Attend.  Attend./G
94   803,436    19,596
124  803,436    19,596
93   756,445    18,450
```

```
62   803,436    19,596
1    819,290    19,983
..       …        …
61   713,714    17,407
185  737,485    17,989
92   729,877    17,814
154  733,247    17,880
309  695,475    16,963

[310 rows x 31 columns]
```

Since we want to find what stats correlate to a high win percentage, we should only look at teams that win a lot. Drop rows from the dataframe so that only the top 50 results are present. This represents about the top 3 seeds in each confrence per year.

[257]:
```
result = result.head(50)
result
```

[257]:
```
     Rk                    Team  Age   W   L  PW  PL    MOV    SOS    SRS  \
94    2  Golden State Warriors*  27.4  73   9  65  17  10.76  -0.38  10.38
124   1  Golden State Warriors*  28.2  67  15  67  15  11.63  -0.28  11.35
93    1      San Antonio Spurs*  30.3  67  15  67  15  10.63  -0.36  10.28
62    1  Golden State Warriors*  26.6  67  15  65  17  10.10  -0.09  10.01
1     2             Miami Heat*  30.3  66  16  62  20   7.87  -0.84   7.03
155   1         Houston Rockets*  29.8  65  17  61  21   8.48  -0.27   8.21
280   2            Phoenix Suns*  27.5  64  18  59  23   7.50  -0.56   6.94
31    1      San Antonio Spurs*  28.9  62  20  61  21   7.72   0.28   8.00
125   2      San Antonio Spurs*  29.6  61  21  60  22   7.20  -0.06   7.13
186   1         Milwaukee Bucks*  26.9  60  22  61  21   8.87  -0.82   8.04
65    4           Atlanta Hawks*  27.8  60  22  56  26   5.43  -0.68   4.75
0     1   Oklahoma City Thunder*  26.0  60  22  64  18   9.21  -0.06   9.15
33    3   Oklahoma City Thunder*  26.2  59  23  58  24   6.34   0.32   6.66
156   2         Toronto Raptors*  25.8  59  23  60  22   7.78  -0.49   7.29
188   3         Toronto Raptors*  27.3  58  24  56  26   6.09  -0.60   5.49
3     4      San Antonio Spurs*  28.6  58  24  58  24   6.40   0.27   6.67
157   3  Golden State Warriors*  28.8  58  24  56  26   5.98  -0.19   5.79
96    4    Cleveland Cavaliers*  28.1  57  25  57  25   6.00  -0.55   5.45
32    2   Los Angeles Clippers*  28.1  57  25  59  23   6.98   0.30   7.27
187   2  Golden State Warriors*  28.4  57  25  56  26   6.46  -0.04   6.42
4     5         Denver Nuggets*  26.1  57  25  55  27   5.09   0.28   5.37
282   4      Memphis Grizzlies*  24.0  56  26  55  27   5.68  -0.32   5.37
97    5         Toronto Raptors*  26.3  56  26  53  29   4.50  -0.42   4.08
2     3   Los Angeles Clippers*  28.8  56  26  59  23   6.45  -0.02   6.43
36    6          Indiana Pacers*  27.2  56  26  54  28   4.40  -0.77   3.63
68    7         Houston Rockets*  27.6  56  26  50  32   3.44   0.38   3.82
6     7      Memphis Grizzlies*  27.0  56  26  54  28   4.15   0.18   4.32
63    2   Los Angeles Clippers*  28.8  56  26  58  24   6.59   0.22   6.80
```

217	1	Milwaukee Bucks*	29.2	56	17	57	16	10.08	-0.67	9.41
69	8	Memphis Grizzlies*	29.6	55	27	50	32	3.24	0.38	3.62
160	6	Boston Celtics*	24.7	55	27	51	31	3.59	-0.35	3.23
64	3	San Antonio Spurs*	29.8	55	27	58	24	6.20	0.14	6.34
95	3	Oklahoma City Thunder*	25.8	55	27	59	23	7.28	-0.19	7.09
126	3	Houston Rockets*	27.4	55	27	55	27	5.77	0.08	5.84
38	8	Portland Trail Blazers*	25.8	54	28	52	30	3.99	0.45	4.44
193	8	Denver Nuggets*	24.9	54	28	51	31	3.95	0.24	4.19
5	6	New York Knicks*	30.2	54	28	53	29	4.23	-0.50	3.73
34	4	Miami Heat*	30.6	54	28	54	28	4.76	-0.61	4.15
37	7	Houston Rockets*	25.4	54	28	53	29	4.56	0.50	5.06
98	6	Los Angeles Clippers*	29.7	53	29	53	29	4.28	-0.15	4.13
66	5	Cleveland Cavaliers*	26.9	53	29	53	29	4.48	-0.40	4.08
283	5	Golden State Warriors*	27.6	53	29	55	27	5.54	-0.02	5.52
190	5	Houston Rockets*	29.2	53	29	53	29	4.77	0.19	4.96
192	7	Portland Trail Blazers*	26.2	53	29	51	31	4.20	0.24	4.43
284	6	Miami Heat*	28.2	53	29	53	29	4.45	-0.22	4.23
131	8	Boston Celtics*	25.9	53	29	48	34	2.63	-0.39	2.25
220	4	Toronto Raptors*	26.6	53	19	50	22	6.24	-0.26	5.97
248	1	Utah Jazz*	28.5	52	20	55	17	9.25	-0.29	8.97
285	7	Dallas Mavericks*	26.7	52	30	50	32	3.30	-0.18	3.12
159	5	Philadelphia 76ers*	25.8	52	30	53	29	4.50	-0.20	4.30

	…	FT/FGA	eFG%	TOV%	DRB%	FT/FGA	Arena	\
94	…	.191	.479	12.6	76.0	.208	Oracle Arena	
124	…	.204	.486	13.5	74.9	.198	Oracle Arena	
93	…	.197	.477	14.1	79.1	.182	AT&T Center	
62	…	.184	.470	14.3	74.5	.217	Oracle Arena	
1	…	.224	.487	14.8	73.0	.200	AmericanAirlines Arena	
155	…	.233	.521	13.4	79.9	.171	Toyota Center	
280	…	.176	.510	13.0	77.1	.195	Phoenix Suns Arena	
31	…	.188	.482	12.8	76.4	.184	AT&T Center	
125	…	.210	.492	13.5	77.6	.192	AT&T Center	
186	…	.197	.503	11.5	80.3	.162	Fiserv Forum	
65	…	.201	.492	14.9	73.4	.185	Philips Arena	
0	…	.280	.469	13.5	73.4	.197	Chesapeake Energy Arena	
33	…	.244	.488	13.9	75.6	.221	Chesapeake Energy Arena	
156	…	.198	.501	13.0	77.7	.212	Air Canada Centre	
188	…	.198	.509	13.1	77.1	.190	Scotiabank Arena	
3	…	.204	.480	13.7	74.9	.179	AT&T Center	
157	…	.195	.504	12.6	76.3	.186	Oracle Arena	
96	…	.194	.496	12.6	78.5	.205	Quicken Loans Arena	
32	…	.258	.484	13.8	72.5	.222	STAPLES Center	
187	…	.182	.508	11.7	77.1	.205	Oracle Arena	
4	…	.216	.493	14.3	71.8	.193	Pepsi Center	
282	…	.180	.523	13.3	77.8	.195	FedEx Forum	
97	…	.255	.498	12.7	77.7	.201	Air Canada Centre	

2	…	.203	.492	15.4	73.5	.229	STAPLES Center
36	…	.226	.460	12.9	76.8	.197	Bankers Life Fieldhouse
68	…	.223	.486	14.6	72.9	.208	Toyota Center
6	…	.202	.475	15.2	74.3	.209	FedEx Forum
63	…	.215	.493	13.2	75.7	.231	STAPLES Center
217	…	.201	.489	12.0	81.6	.178	Fiserv Forum
69	…	.214	.492	14.5	75.3	.183	FedEx Forum
160	…	.188	.495	13.0	78.4	.191	TD Garden
64	…	.200	.484	13.3	77.3	.190	AT&T Center
95	…	.228	.484	11.7	76.0	.205	Chesapeake Energy Arena
126	…	.233	.519	13.2	75.8	.194	Toyota Center
38	…	.220	.488	11.0	74.7	.194	Moda Center
193	…	.175	.521	12.3	78.0	.194	Pepsi Center
5	…	.196	.508	14.8	74.7	.216	Madison Square Garden (IV)
34	…	.228	.511	15.8	73.0	.212	AmericanAirlines Arena
37	…	.275	.489	12.5	74.1	.193	Toyota Center
98	…	.220	.480	13.8	73.8	.222	STAPLES Center
66	…	.216	.502	12.6	74.7	.177	Quicken Loans Arena
283	…	.181	.509	13.0	78.7	.201	Chase Center
190	…	.221	.525	13.4	74.4	.210	Toyota Center
192	…	.210	.516	11.0	77.9	.195	Moda Center
284	…	.204	.524	13.8	78.0	.209	FTX Arena
131	…	.220	.503	12.6	75.3	.223	TD Garden
220	…	.210	.502	14.6	76.7	.202	Scotiabank Arena
248	…	.195	.507	10.3	79.3	.159	Vivint Smart Home Arena
285	…	.192	.521	12.2	78.0	.185	American Airlines Center
159	…	.198	.492	12.6	78.6	.218	Wells Fargo Center

	Attend.	Attend./G
94	803,436	19,596
124	803,436	19,596
93	756,445	18,450
62	803,436	19,596
1	819,290	19,983
155	732,722	17,871
280	663,171	16,175
31	755,031	18,415
125	755,347	18,423
186	721,692	17,602
65	713,909	17,412
0	746,323	18,203
33	746,323	18,203
156	813,431	19,840
188	812,822	19,825
3	755,700	18,432
157	803,436	19,596
96	843,042	20,562

```
32   787,692    19,212
187  803,436    19,596
4    730,616    17,820
282  646,785    15,775
97   812,863    19,826
2    788,293    19,227
36   717,542    17,501
68   747,412    18,230
6    681,613    16,625
63   785,892    19,168
217  549,036    17,711
69   710,502    17,329
160  763,584    18,624
64   762,855    18,606
95   746,323    18,203
126  695,903    16,973
38   809,612    19,747
193  756,457    18,450
5    780,353    19,033
34   811,036    19,781
37   743,082    18,124
98   786,910    19,193
66   843,042    20,562
283  740,624    18,064
190  740,392    18,058
192  799,345    19,496
284  804,761    19,628
131  760,690    18,553
220  633,456    19,796
248  151,300     4,203
285  808,037    19,708
159  833,503    20,361

[50 rows x 31 columns]
```

We want to remove unwanted data such as attendance, arena, the loss column, ranks, as well as redicted wins/loss. Attendance and arena is an uncontrollable stat, as it is dependant on a team's location. Also teams performance away and at home are not possible to be accounted for. The loss, rank and prediction columns are already calculated via the wins share column, thus is just repeated data in which we can remove.

[258]:
```
result = result.drop('Rk', axis = 1)
result = result.drop('L', axis = 1)
result = result.drop('PW', axis = 1)
result = result.drop('PL', axis = 1)
result = result.drop('Arena', axis = 1)
result = result.drop('Attend.', axis = 1)
```

```python
result = result.drop('Attend./G', axis = 1)
result
```

[258]:

	Team	Age	W	MOV	SOS	SRS	ORtg	DRtg	\
94	Golden State Warriors*	27.4	73	10.76	-0.38	10.38	114.5	103.8	
124	Golden State Warriors*	28.2	67	11.63	-0.28	11.35	115.6	104.0	
93	San Antonio Spurs*	30.3	67	10.63	-0.36	10.28	110.3	99.0	
62	Golden State Warriors*	26.6	67	10.10	-0.09	10.01	111.6	101.4	
1	Miami Heat*	30.3	66	7.87	-0.84	7.03	112.3	103.7	
155	Houston Rockets*	29.8	65	8.48	-0.27	8.21	114.7	106.1	
280	Phoenix Suns*	27.5	64	7.50	-0.56	6.94	114.8	107.3	
31	San Antonio Spurs*	28.9	62	7.72	0.28	8.00	110.5	102.4	
125	San Antonio Spurs*	29.6	61	7.20	-0.06	7.13	111.1	103.5	
186	Milwaukee Bucks*	26.9	60	8.87	-0.82	8.04	113.8	105.2	
65	Atlanta Hawks*	27.8	60	5.43	-0.68	4.75	108.9	103.1	
0	Oklahoma City Thunder*	26.0	60	9.21	-0.06	9.15	112.4	102.6	
33	Oklahoma City Thunder*	26.2	59	6.34	0.32	6.66	110.5	103.9	
156	Toronto Raptors*	25.8	59	7.78	-0.49	7.29	113.8	105.9	
188	Toronto Raptors*	27.3	58	6.09	-0.60	5.49	113.1	107.1	
3	San Antonio Spurs*	28.6	58	6.40	0.27	6.67	108.3	101.6	
157	Golden State Warriors*	28.8	58	5.98	-0.19	5.79	113.6	107.6	
96	Cleveland Cavaliers*	28.1	57	6.00	-0.55	5.45	110.9	104.5	
32	Los Angeles Clippers*	28.1	57	6.98	0.30	7.27	112.1	104.8	
187	Golden State Warriors*	28.4	57	6.46	-0.04	6.42	115.9	109.5	
4	Denver Nuggets*	26.1	57	5.09	0.28	5.37	110.4	105.1	
282	Memphis Grizzlies*	24.0	56	5.68	-0.32	5.37	114.6	109.0	
97	Toronto Raptors*	26.3	56	4.50	-0.42	4.08	110.0	105.2	
2	Los Angeles Clippers*	28.8	56	6.45	-0.02	6.43	110.6	103.6	
36	Indiana Pacers*	27.2	56	4.40	-0.77	3.63	104.1	99.3	
68	Houston Rockets*	27.6	56	3.44	0.38	3.82	107.0	103.4	
6	Memphis Grizzlies*	27.0	56	4.15	0.18	4.32	104.9	100.3	
63	Los Angeles Clippers*	28.8	56	6.59	0.22	6.80	112.4	105.5	
217	Milwaukee Bucks*	29.2	56	10.08	-0.67	9.41	112.4	102.9	
69	Memphis Grizzlies*	29.6	55	3.24	0.38	3.62	105.7	102.2	
160	Boston Celtics*	24.7	55	3.59	-0.35	3.23	107.6	103.9	
64	San Antonio Spurs*	29.8	55	6.20	0.14	6.34	108.5	102.0	
95	Oklahoma City Thunder*	25.8	55	7.28	-0.19	7.09	113.1	105.6	
126	Houston Rockets*	27.4	55	5.77	0.08	5.84	114.7	109.0	
38	Portland Trail Blazers*	25.8	54	3.99	0.45	4.44	111.5	107.4	
193	Denver Nuggets*	24.9	54	3.95	0.24	4.19	113.0	108.9	
5	New York Knicks*	30.2	54	4.23	-0.50	3.73	111.1	106.3	
34	Miami Heat*	30.6	54	4.76	-0.61	4.15	110.9	105.8	
37	Houston Rockets*	25.4	54	4.56	0.50	5.06	111.0	106.3	
98	Los Angeles Clippers*	29.7	53	4.28	-0.15	4.13	108.3	103.8	
66	Cleveland Cavaliers*	26.9	53	4.48	-0.40	4.08	111.1	106.3	
283	Golden State Warriors*	27.6	53	5.54	-0.02	5.52	112.5	106.9	
190	Houston Rockets*	29.2	53	4.77	0.19	4.96	115.5	110.7	

192	Portland Trail Blazers*	26.2	53	4.20	0.24	4.43	114.7	110.5
284	Miami Heat*	28.2	53	4.45	-0.22	4.23	113.7	109.1
131	Boston Celtics*	25.9	53	2.63	-0.39	2.25	111.2	108.4
220	Toronto Raptors*	26.6	53	6.24	-0.26	5.97	111.1	105.0
248	Utah Jazz*	28.5	52	9.25	-0.29	8.97	117.6	108.3
285	Dallas Mavericks*	26.7	52	3.30	-0.18	3.12	112.8	109.4
159	Philadelphia 76ers*	25.8	52	4.50	-0.20	4.30	109.5	105.0

	NRtg	Pace	…	eFG%	TOV%	ORB%	FT/FGA	eFG%	TOV%	DRB%	FT/FGA
94	+10.7	99.3	…	.563	13.5	23.5	.191	.479	12.6	76.0	.208
124	+11.6	99.8	…	.563	13.2	22.8	.204	.486	13.5	74.9	.198
93	+11.3	93.8	…	.526	12.4	23.0	.197	.477	14.1	79.1	.182
62	+10.2	98.3	…	.540	13.1	24.1	.184	.470	14.3	74.5	.217
1	+8.6	90.7	…	.552	13.7	22.2	.224	.487	14.8	73.0	.200
155	+8.6	97.6	…	.551	12.7	21.3	.233	.521	13.4	79.9	.171
280	+7.5	99.8	…	.549	11.6	22.3	.176	.510	13.0	77.1	.195
31	+8.1	95.0	…	.537	13.5	22.7	.188	.482	12.8	76.4	.184
125	+7.6	94.2	…	.524	12.6	24.0	.210	.492	13.5	77.6	.192
186	+8.6	103.3	…	.550	12.0	20.8	.197	.503	11.5	80.3	.162
65	+5.8	93.9	…	.527	13.5	21.4	.201	.492	14.9	73.4	.185
0	+9.8	93.3	…	.527	14.4	26.7	.280	.469	13.5	73.4	.197
33	+6.6	95.4	…	.520	14.0	26.5	.244	.488	13.9	75.6	.221
156	+7.9	97.4	…	.539	12.1	23.0	.198	.501	13.0	77.7	.212
188	+6.0	100.2	…	.543	12.4	21.9	.198	.509	13.1	77.1	.190
3	+6.7	94.2	…	.531	14.0	20.5	.204	.480	13.7	74.9	.179
157	+6.0	99.6	…	.569	14.1	21.0	.195	.504	12.6	76.3	.186
96	+6.4	93.3	…	.524	12.7	25.1	.194	.496	12.6	78.5	.205
32	+7.3	95.9	…	.526	12.7	25.0	.258	.484	13.8	72.5	.222
187	+6.4	100.9	…	.565	12.6	22.5	.182	.508	11.7	77.1	.205
4	+5.3	95.1	…	.515	13.6	31.4	.216	.493	14.3	71.8	.193
282	+5.6	100.3	…	.522	11.2	30.0	.180	.523	13.3	77.8	.195
97	+4.8	92.9	…	.504	12.3	24.6	.255	.498	12.7	77.7	.201
2	+7.0	91.1	…	.526	13.9	28.8	.203	.492	15.4	73.5	.229
36	+4.8	92.5	…	.490	14.3	24.9	.226	.460	12.9	76.8	.197
68	+3.6	96.5	…	.512	15.0	26.8	.223	.486	14.6	72.9	.208
6	+4.6	88.4	…	.472	13.3	31.0	.202	.475	15.2	74.3	.209
63	+6.9	94.7	…	.533	11.6	22.8	.215	.493	13.2	75.7	.231
217	+9.5	105.1	…	.552	12.9	20.7	.201	.489	12.0	81.6	.178
69	+3.5	92.0	…	.489	12.6	24.7	.214	.492	14.5	75.3	.183
160	+3.7	96.0	…	.518	13.0	21.5	.188	.495	13.0	78.4	.191
64	+6.5	93.8	…	.517	13.1	23.4	.200	.484	13.3	77.3	.190
95	+7.5	96.7	…	.524	14.0	31.1	.228	.484	11.7	76.0	.205
126	+5.7	100.0	…	.545	13.3	24.6	.233	.519	13.2	75.8	.194
38	+4.1	94.9	…	.504	12.4	28.0	.220	.488	11.0	74.7	.194
193	+4.1	97.7	…	.527	11.9	26.6	.175	.521	12.3	78.0	.194
5	+4.8	89.8	…	.515	11.7	25.6	.196	.508	14.8	74.7	.216
34	+5.1	91.2	…	.554	14.6	20.6	.228	.511	15.8	73.0	.212

```
37    +4.7   96.3  …  .531  14.6  27.4   .275    .489  12.5  74.1   .193
98    +4.5   95.8  …  .524  12.1  20.1   .220    .480  13.8  73.8   .222
66    +4.8   92.3  …  .520  13.4  26.8   .216    .502  12.6  74.7   .177
283   +5.6   98.4  …  .552  13.5  22.8   .181    .509  13.0  78.7   .201
190   +4.8   97.9  …  .542  12.0  22.8   .221    .525  13.4  74.4   .210
192   +4.2   99.1  …  .528  12.1  26.6   .210    .516  11.0  77.9   .195
284   +4.6   95.9  …  .547  13.4  23.5   .204    .524  13.8  78.0   .209
131   +2.8   96.8  …  .525  12.2  21.2   .220    .503  12.6  75.3   .223
220   +6.1  100.9  …  .536  13.1  21.3   .210    .502  14.6  76.7   .202
248   +9.3   98.5  …  .563  12.7  24.5   .195    .507  10.3  79.3   .159
285   +3.4   95.4  …  .538  11.7  21.3   .192    .521  12.2  78.0   .185
159   +4.5   99.8  …  .535  14.6  25.3   .198    .492  12.6  78.6   .218

[50 rows x 24 columns]
```

We now see that everything is sorted and organized, but there is still one thing that makes our data a tad confusing. The labels of the table include four offensive and defensive stats that are the exact same name. This includes eFG%, TOV%, and FT/FGA. Let us rename it so that we can better interpret the data.

[259]:
```python
result = result.rename(columns={'eFG%': 'OeFG%'})
result = result.rename(columns={'TOV%': 'OTOV%'})
result = result.rename(columns={'FT/FGA': 'O_FT/FGA', 'FT/FGA': 'D_FT/FGA' })
result
```

[259]:
```
                         Team   Age   W    MOV    SOS    SRS   ORtg   DRtg  \
94    Golden State Warriors*  27.4  73  10.76  -0.38  10.38  114.5  103.8
124   Golden State Warriors*  28.2  67  11.63  -0.28  11.35  115.6  104.0
93        San Antonio Spurs*  30.3  67  10.63  -0.36  10.28  110.3   99.0
62    Golden State Warriors*  26.6  67  10.10  -0.09  10.01  111.6  101.4
1                Miami Heat*  30.3  66   7.87  -0.84   7.03  112.3  103.7
155         Houston Rockets*  29.8  65   8.48  -0.27   8.21  114.7  106.1
280            Phoenix Suns*  27.5  64   7.50  -0.56   6.94  114.8  107.3
31        San Antonio Spurs*  28.9  62   7.72   0.28   8.00  110.5  102.4
125       San Antonio Spurs*  29.6  61   7.20  -0.06   7.13  111.1  103.5
186         Milwaukee Bucks*  26.9  60   8.87  -0.82   8.04  113.8  105.2
65            Atlanta Hawks*  27.8  60   5.43  -0.68   4.75  108.9  103.1
0     Oklahoma City Thunder*  26.0  60   9.21  -0.06   9.15  112.4  102.6
33    Oklahoma City Thunder*  26.2  59   6.34   0.32   6.66  110.5  103.9
156         Toronto Raptors*  25.8  59   7.78  -0.49   7.29  113.8  105.9
188         Toronto Raptors*  27.3  58   6.09  -0.60   5.49  113.1  107.1
3         San Antonio Spurs*  28.6  58   6.40   0.27   6.67  108.3  101.6
157   Golden State Warriors*  28.8  58   5.98  -0.19   5.79  113.6  107.6
96      Cleveland Cavaliers*  28.1  57   6.00  -0.55   5.45  110.9  104.5
32     Los Angeles Clippers*  28.1  57   6.98   0.30   7.27  112.1  104.8
187   Golden State Warriors*  28.4  57   6.46  -0.04   6.42  115.9  109.5
4            Denver Nuggets*  26.1  57   5.09   0.28   5.37  110.4  105.1
```

282	Memphis Grizzlies*	24.0	56	5.68	-0.32	5.37	114.6	109.0
97	Toronto Raptors*	26.3	56	4.50	-0.42	4.08	110.0	105.2
2	Los Angeles Clippers*	28.8	56	6.45	-0.02	6.43	110.6	103.6
36	Indiana Pacers*	27.2	56	4.40	-0.77	3.63	104.1	99.3
68	Houston Rockets*	27.6	56	3.44	0.38	3.82	107.0	103.4
6	Memphis Grizzlies*	27.0	56	4.15	0.18	4.32	104.9	100.3
63	Los Angeles Clippers*	28.8	56	6.59	0.22	6.80	112.4	105.5
217	Milwaukee Bucks*	29.2	56	10.08	-0.67	9.41	112.4	102.9
69	Memphis Grizzlies*	29.6	55	3.24	0.38	3.62	105.7	102.2
160	Boston Celtics*	24.7	55	3.59	-0.35	3.23	107.6	103.9
64	San Antonio Spurs*	29.8	55	6.20	0.14	6.34	108.5	102.0
95	Oklahoma City Thunder*	25.8	55	7.28	-0.19	7.09	113.1	105.6
126	Houston Rockets*	27.4	55	5.77	0.08	5.84	114.7	109.0
38	Portland Trail Blazers*	25.8	54	3.99	0.45	4.44	111.5	107.4
193	Denver Nuggets*	24.9	54	3.95	0.24	4.19	113.0	108.9
5	New York Knicks*	30.2	54	4.23	-0.50	3.73	111.1	106.3
34	Miami Heat*	30.6	54	4.76	-0.61	4.15	110.9	105.8
37	Houston Rockets*	25.4	54	4.56	0.50	5.06	111.0	106.3
98	Los Angeles Clippers*	29.7	53	4.28	-0.15	4.13	108.3	103.8
66	Cleveland Cavaliers*	26.9	53	4.48	-0.40	4.08	111.1	106.3
283	Golden State Warriors*	27.6	53	5.54	-0.02	5.52	112.5	106.9
190	Houston Rockets*	29.2	53	4.77	0.19	4.96	115.5	110.7
192	Portland Trail Blazers*	26.2	53	4.20	0.24	4.43	114.7	110.5
284	Miami Heat*	28.2	53	4.45	-0.22	4.23	113.7	109.1
131	Boston Celtics*	25.9	53	2.63	-0.39	2.25	111.2	108.4
220	Toronto Raptors*	26.6	53	6.24	-0.26	5.97	111.1	105.0
248	Utah Jazz*	28.5	52	9.25	-0.29	8.97	117.6	108.3
285	Dallas Mavericks*	26.7	52	3.30	-0.18	3.12	112.8	109.4
159	Philadelphia 76ers*	25.8	52	4.50	-0.20	4.30	109.5	105.0

	NRtg	Pace	…	OeFG%	OTOV%	ORB%	D_FT/FGA	OeFG%	OTOV%	DRB%	\
94	+10.7	99.3	…	.563	13.5	23.5	.191	.479	12.6	76.0	
124	+11.6	99.8	…	.563	13.2	22.8	.204	.486	13.5	74.9	
93	+11.3	93.8	…	.526	12.4	23.0	.197	.477	14.1	79.1	
62	+10.2	98.3	…	.540	13.1	24.1	.184	.470	14.3	74.5	
1	+8.6	90.7	…	.552	13.7	22.2	.224	.487	14.8	73.0	
155	+8.6	97.6	…	.551	12.7	21.3	.233	.521	13.4	79.9	
280	+7.5	99.8	…	.549	11.6	22.3	.176	.510	13.0	77.1	
31	+8.1	95.0	…	.537	13.5	22.7	.188	.482	12.8	76.4	
125	+7.6	94.2	…	.524	12.6	24.0	.210	.492	13.5	77.6	
186	+8.6	103.3	…	.550	12.0	20.8	.197	.503	11.5	80.3	
65	+5.8	93.9	…	.527	13.5	21.4	.201	.492	14.9	73.4	
0	+9.8	93.3	…	.527	14.4	26.7	.280	.469	13.5	73.4	
33	+6.6	95.4	…	.520	14.0	26.5	.244	.488	13.9	75.6	
156	+7.9	97.4	…	.539	12.1	23.0	.198	.501	13.0	77.7	
188	+6.0	100.2	…	.543	12.4	21.9	.198	.509	13.1	77.1	
3	+6.7	94.2	…	.531	14.0	20.5	.204	.480	13.7	74.9	

157	+6.0	99.6	…	.569	14.1	21.0	.195	.504	12.6	76.3
96	+6.4	93.3	…	.524	12.7	25.1	.194	.496	12.6	78.5
32	+7.3	95.9	…	.526	12.7	25.0	.258	.484	13.8	72.5
187	+6.4	100.9	…	.565	12.6	22.5	.182	.508	11.7	77.1
4	+5.3	95.1	…	.515	13.6	31.4	.216	.493	14.3	71.8
282	+5.6	100.3	…	.522	11.2	30.0	.180	.523	13.3	77.8
97	+4.8	92.9	…	.504	12.3	24.6	.255	.498	12.7	77.7
2	+7.0	91.1	…	.526	13.9	28.8	.203	.492	15.4	73.5
36	+4.8	92.5	…	.490	14.3	24.9	.226	.460	12.9	76.8
68	+3.6	96.5	…	.512	15.0	26.8	.223	.486	14.6	72.9
6	+4.6	88.4	…	.472	13.3	31.0	.202	.475	15.2	74.3
63	+6.9	94.7	…	.533	11.6	22.8	.215	.493	13.2	75.7
217	+9.5	105.1	…	.552	12.9	20.7	.201	.489	12.0	81.6
69	+3.5	92.0	…	.489	12.6	24.7	.214	.492	14.5	75.3
160	+3.7	96.0	…	.518	13.0	21.5	.188	.495	13.0	78.4
64	+6.5	93.8	…	.517	13.1	23.4	.200	.484	13.3	77.3
95	+7.5	96.7	…	.524	14.0	31.1	.228	.484	11.7	76.0
126	+5.7	100.0	…	.545	13.3	24.6	.233	.519	13.2	75.8
38	+4.1	94.9	…	.504	12.4	28.0	.220	.488	11.0	74.7
193	+4.1	97.7	…	.527	11.9	26.6	.175	.521	12.3	78.0
5	+4.8	89.8	…	.515	11.7	25.6	.196	.508	14.8	74.7
34	+5.1	91.2	…	.554	14.6	20.6	.228	.511	15.8	73.0
37	+4.7	96.3	…	.531	14.6	27.4	.275	.489	12.5	74.1
98	+4.5	95.8	…	.524	12.1	20.1	.220	.480	13.8	73.8
66	+4.8	92.3	…	.520	13.4	26.8	.216	.502	12.6	74.7
283	+5.6	98.4	…	.552	13.5	22.8	.181	.509	13.0	78.7
190	+4.8	97.9	…	.542	12.0	22.8	.221	.525	13.4	74.4
192	+4.2	99.1	…	.528	12.1	26.6	.210	.516	11.0	77.9
284	+4.6	95.9	…	.547	13.4	23.5	.204	.524	13.8	78.0
131	+2.8	96.8	…	.525	12.2	21.2	.220	.503	12.6	75.3
220	+6.1	100.9	…	.536	13.1	21.3	.210	.502	14.6	76.7
248	+9.3	98.5	…	.563	12.7	24.5	.195	.507	10.3	79.3
285	+3.4	95.4	…	.538	11.7	21.3	.192	.521	12.2	78.0
159	+4.5	99.8	…	.535	14.6	25.3	.198	.492	12.6	78.6

	D_FT/FGA
94	.208
124	.198
93	.182
62	.217
1	.200
155	.171
280	.195
31	.184
125	.192
186	.162
65	.185

```
0        .197
33       .221
156      .212
188      .190
3        .179
157      .186
96       .205
32       .222
187      .205
4        .193
282      .195
97       .201
2        .229
36       .197
68       .208
6        .209
63       .231
217      .178
69       .183
160      .191
64       .190
95       .205
126      .194
38       .194
193      .194
5        .216
34       .212
37       .193
98       .222
66       .177
283      .201
190      .210
192      .195
284      .209
131      .223
220      .202
248      .159
285      .185
159      .218

[50 rows x 24 columns]
```

5 Exploratory Analysis and Data Visualization:

In this part of tutorial, we will dive into trends and logistical analysis via graphs and sorting through the numbers. There will be plots to show trends we notice, as well as sorting the tables by certain categories that will show which correlate to team wins. Therefore we will further dive into the data and explore these trends.

Before we dive into details and statistics we should look at which teams make it in the top 50 the most and that will lead us to wanna find out the facts and correlations of why these teams win so much

We will plot a pie chart to help visualize which teams win in comparison to others.

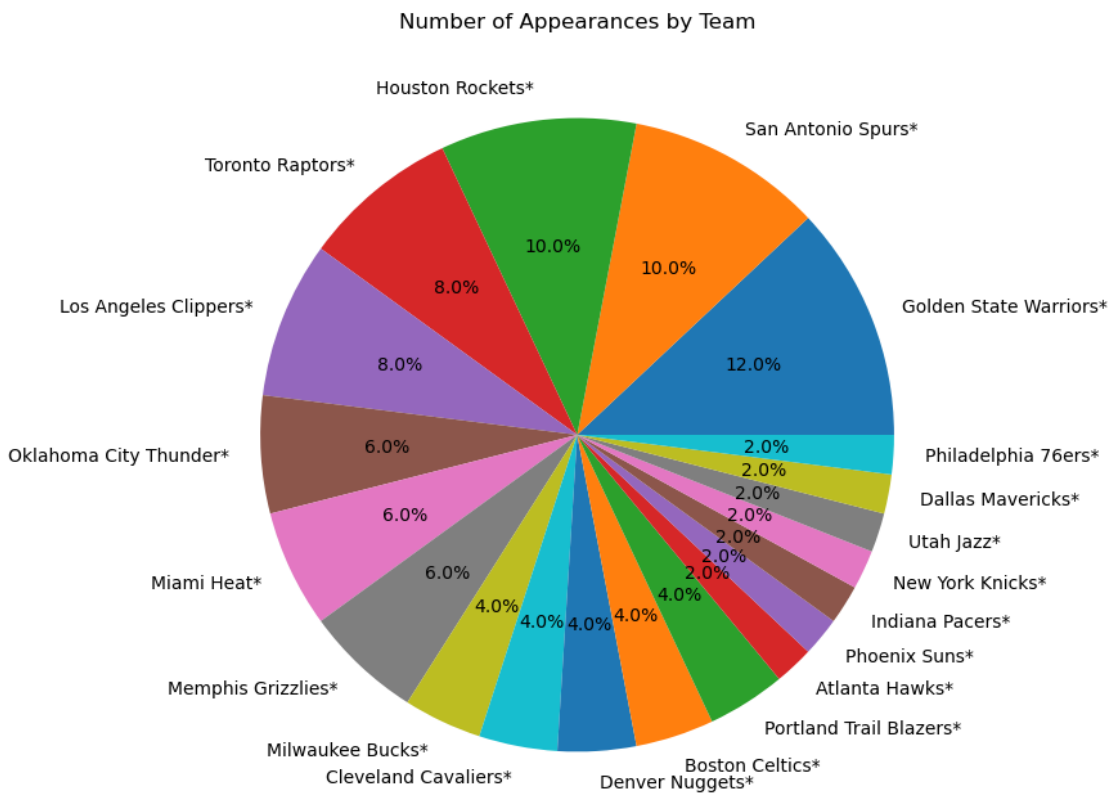
[170]:
```
display(result[['Team','W']])
```

```
                         Team   W
94     Golden State Warriors*  73
124    Golden State Warriors*  67
93         San Antonio Spurs*  67
62     Golden State Warriors*  67
1                 Miami Heat*  66
155          Houston Rockets*  65
280             Phoenix Suns*  64
31         San Antonio Spurs*  62
125        San Antonio Spurs*  61
186          Milwaukee Bucks*  60
65             Atlanta Hawks*  60
0      Oklahoma City Thunder*  60
33     Oklahoma City Thunder*  59
156          Toronto Raptors*  59
188          Toronto Raptors*  58
3          San Antonio Spurs*  58
157    Golden State Warriors*  58
96       Cleveland Cavaliers*  57
32      Los Angeles Clippers*  57
187    Golden State Warriors*  57
4             Denver Nuggets*  57
282        Memphis Grizzlies*  56
97           Toronto Raptors*  56
2       Los Angeles Clippers*  56
36            Indiana Pacers*  56
68           Houston Rockets*  56
6          Memphis Grizzlies*  56
63      Los Angeles Clippers*  56
217          Milwaukee Bucks*  56
69         Memphis Grizzlies*  55
160           Boston Celtics*  55
64         San Antonio Spurs*  55
95     Oklahoma City Thunder*  55
126          Houston Rockets*  55
38    Portland Trail Blazers*  54
193           Denver Nuggets*  54
5           New York Knicks*  54
34                Miami Heat*  54
```

```
37           Houston Rockets*   54
98      Los Angeles Clippers*   53
66       Cleveland Cavaliers*   53
283    Golden State Warriors*   53
190          Houston Rockets*   53
192   Portland Trail Blazers*   53
284               Miami Heat*   53
131          Boston Celtics*    53
220          Toronto Raptors*   53
248                Utah Jazz*   52
285        Dallas Mavericks*    52
159      Philadelphia 76ers*    52
```

[260]:

```python
counts = result['Team'].value_counts()

# create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(counts.values, labels=counts.index, autopct='%1.1f%%')
plt.title('Number of Appearances by Team')
plt.show()
```



We clearly see now that the Golden State Warriors, San Antonio Spurs, and Houston Rockets taken up most of the space on the pie chart. We now will dive into what makes these teams so good and what factors we can take away from this.

We can see from the pie chart that five teams, the Golden State Warriors, San Antonio Spurs, Houston Rockets, Toronto Raptors, and Los Angeles Clippers make up almost half of the appearances. This means that whatever strategies or strengths they have are working very well to get them an above average amount of wins. Lets take a look a couple stats which these teams might have strengths in which may lead them to more wins First, drop all rows from the dataframe where the team is not one of these five top teams

[261]:
```
team_names = ['Golden State Warriors*', 'San Antonio Spurs*', 'Houston␣
↪Rockets*', 'Toronto Raptors*', 'Los Angeles Clippers*']
filtered_results = result[result['Team'].isin(team_names)]
filtered_results
```

[261]:
```
                       Team   Age   W    MOV    SOS    SRS   ORtg   DRtg  \
94   Golden State Warriors*  27.4  73  10.76  -0.38  10.38  114.5  103.8
124  Golden State Warriors*  28.2  67  11.63  -0.28  11.35  115.6  104.0
93       San Antonio Spurs*  30.3  67  10.63  -0.36  10.28  110.3   99.0
62   Golden State Warriors*  26.6  67  10.10  -0.09  10.01  111.6  101.4
155        Houston Rockets*  29.8  65   8.48  -0.27   8.21  114.7  106.1
31       San Antonio Spurs*  28.9  62   7.72   0.28   8.00  110.5  102.4
125      San Antonio Spurs*  29.6  61   7.20  -0.06   7.13  111.1  103.5
156        Toronto Raptors*  25.8  59   7.78  -0.49   7.29  113.8  105.9
188        Toronto Raptors*  27.3  58   6.09  -0.60   5.49  113.1  107.1
3        San Antonio Spurs*  28.6  58   6.40   0.27   6.67  108.3  101.6
157  Golden State Warriors*  28.8  58   5.98  -0.19   5.79  113.6  107.6
32    Los Angeles Clippers*  28.1  57   6.98   0.30   7.27  112.1  104.8
187  Golden State Warriors*  28.4  57   6.46  -0.04   6.42  115.9  109.5
97         Toronto Raptors*  26.3  56   4.50  -0.42   4.08  110.0  105.2
2     Los Angeles Clippers*  28.8  56   6.45  -0.02   6.43  110.6  103.6
68         Houston Rockets*  27.6  56   3.44   0.38   3.82  107.0  103.4
63    Los Angeles Clippers*  28.8  56   6.59   0.22   6.80  112.4  105.5
64       San Antonio Spurs*  29.8  55   6.20   0.14   6.34  108.5  102.0
126        Houston Rockets*  27.4  55   5.77   0.08   5.84  114.7  109.0
37         Houston Rockets*  25.4  54   4.56   0.50   5.06  111.0  106.3
98    Los Angeles Clippers*  29.7  53   4.28  -0.15   4.13  108.3  103.8
283  Golden State Warriors*  27.6  53   5.54  -0.02   5.52  112.5  106.9
190        Houston Rockets*  29.2  53   4.77   0.19   4.96  115.5  110.7
220        Toronto Raptors*  26.6  53   6.24  -0.26   5.97  111.1  105.0

      NRtg  Pace  … OeFG% OTOV%  ORB% D_FT/FGA  OeFG% OTOV%  DRB%  \
94   +10.7  99.3  …  .563  13.5  23.5     .191   .479  12.6  76.0
124  +11.6  99.8  …  .563  13.2  22.8     .204   .486  13.5  74.9
93   +11.3  93.8  …  .526  12.4  23.0     .197   .477  14.1  79.1
62   +10.2  98.3  …  .540  13.1  24.1     .184   .470  14.3  74.5
155   +8.6  97.6  …  .551  12.7  21.3     .233   .521  13.4  79.9
```

31	+8.1	95.0	…	.537	13.5	22.7	.188	.482	12.8	76.4
125	+7.6	94.2	…	.524	12.6	24.0	.210	.492	13.5	77.6
156	+7.9	97.4	…	.539	12.1	23.0	.198	.501	13.0	77.7
188	+6.0	100.2	…	.543	12.4	21.9	.198	.509	13.1	77.1
3	+6.7	94.2	…	.531	14.0	20.5	.204	.480	13.7	74.9
157	+6.0	99.6	…	.569	14.1	21.0	.195	.504	12.6	76.3
32	+7.3	95.9	…	.526	12.7	25.0	.258	.484	13.8	72.5
187	+6.4	100.9	…	.565	12.6	22.5	.182	.508	11.7	77.1
97	+4.8	92.9	…	.504	12.3	24.6	.255	.498	12.7	77.7
2	+7.0	91.1	…	.526	13.9	28.8	.203	.492	15.4	73.5
68	+3.6	96.5	…	.512	15.0	26.8	.223	.486	14.6	72.9
63	+6.9	94.7	…	.533	11.6	22.8	.215	.493	13.2	75.7
64	+6.5	93.8	…	.517	13.1	23.4	.200	.484	13.3	77.3
126	+5.7	100.0	…	.545	13.3	24.6	.233	.519	13.2	75.8
37	+4.7	96.3	…	.531	14.6	27.4	.275	.489	12.5	74.1
98	+4.5	95.8	…	.524	12.1	20.1	.220	.480	13.8	73.8
283	+5.6	98.4	…	.552	13.5	22.8	.181	.509	13.0	78.7
190	+4.8	97.9	…	.542	12.0	22.8	.221	.525	13.4	74.4
220	+6.1	100.9	…	.536	13.1	21.3	.210	.502	14.6	76.7

	D_FT/FGA
94	.208
124	.198
93	.182
62	.217
155	.171
31	.184
125	.192
156	.212
188	.190
3	.179
157	.186
32	.222
187	.205
97	.201
2	.229
68	.208
63	.231
64	.190
126	.194
37	.193
98	.222
283	.201
190	.210
220	.202

[24 rows x 24 columns]

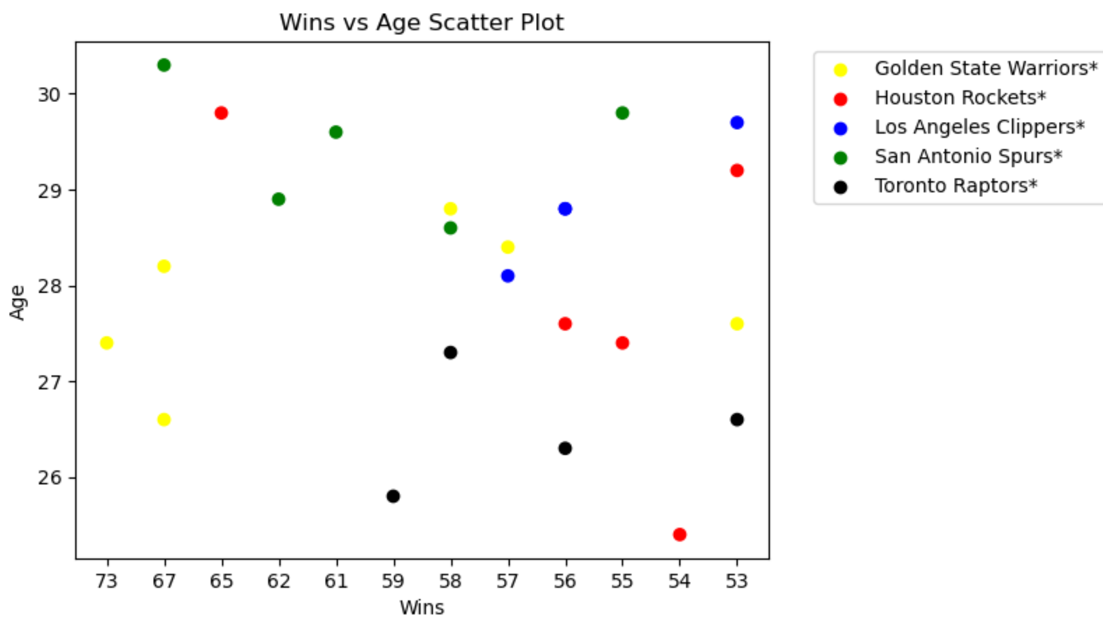
Next pick four stats to look at. Look for trends in the stats, if they have a strong correlation with the win percentage. In addition, look for trends regarding the teams - does any team excel at any one of the stats? Plotting the datapoints in color to match which team the data is from makes the analysis easier.

[177]:
```
#Plot for Age
color_map = {'Golden State Warriors*': 'yellow', 'Houston Rockets*': 'red',␣
↪'Los Angeles Clippers*': 'blue', 'San Antonio Spurs*': 'green', 'Toronto␣
↪Raptors*': 'black'}

filtered_results['Age'] = filtered_results['Age'].astype(float)
colors = filtered_results['Team'].map(color_map)
valid_colors = [color_map.get(team, 'gray') for team in␣
↪filtered_results['Team']]
plt.scatter(filtered_results['W'], filtered_results['Age'], c=valid_colors)
# add a legend
labels = color_map.keys()
handles = [plt.scatter([], [], c=color_map[label], label=label) for label in␣
↪labels]
plt.legend(handles, labels, loc='best')
plt.legend(loc='upper left', bbox_to_anchor=(1.05, 1))

plt.xlabel('Wins')
plt.ylabel('Age')
plt.title('Wins vs Age Scatter Plot')
plt.show()

#Plot for Defensive Rating
color_map = {'Golden State Warriors*': 'yellow', 'Houston Rockets*': 'red',␣
↪'Los Angeles Clippers*': 'blue', 'San Antonio Spurs*': 'green', 'Toronto␣
↪Raptors*': 'black'}

filtered_results['DRtg'] = filtered_results['DRtg'].astype(float)
colors = filtered_results['Team'].map(color_map)
valid_colors = [color_map.get(team, 'gray') for team in␣
↪filtered_results['Team']]
plt.scatter(filtered_results['W'], filtered_results['DRtg'], c=valid_colors)
labels = color_map.keys()
handles = [plt.scatter([], [], c=color_map[label], label=label) for label in␣
↪labels]
plt.legend(handles, labels, loc='best')
plt.legend(loc='upper left', bbox_to_anchor=(1.05, 1))

plt.xlabel('Wins')
plt.ylabel('Defensive Rating')
plt.title('Wins vs Defensive Rating')
plt.show()
```

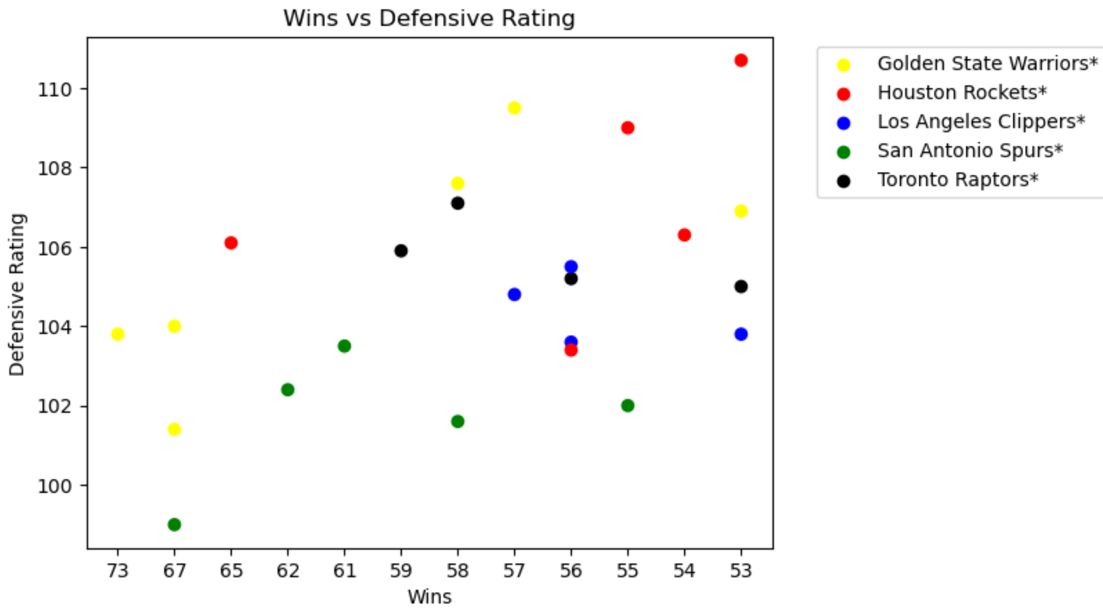
```
#Plot for Offensive Rating
color_map = {'Golden State Warriors*': 'yellow', 'Houston Rockets*': 'red',␣
↪'Los Angeles Clippers*': 'blue', 'San Antonio Spurs*': 'green', 'Toronto␣
↪Raptors*': 'black'}

filtered_results['ORtg'] = filtered_results['ORtg'].astype(float)
colors = filtered_results['Team'].map(color_map)
valid_colors = [color_map.get(team, 'gray') for team in␣
↪filtered_results['Team']]
plt.scatter(filtered_results['W'], filtered_results['ORtg'], c=valid_colors)
labels = color_map.keys()
handles = [plt.scatter([], [], c=color_map[label], label=label) for label in␣
↪labels]
plt.legend(handles, labels, loc='best')
plt.legend(loc='upper left', bbox_to_anchor=(1.05, 1))

plt.xlabel('Wins')
plt.ylabel('Offensive Rating')
plt.title('Wins vs Offensive Rating')
plt.show()

#Plot for Pace
color_map = {'Golden State Warriors*': 'yellow', 'Houston Rockets*': 'red',␣
↪'Los Angeles Clippers*': 'blue', 'San Antonio Spurs*': 'green', 'Toronto␣
↪Raptors*': 'black'}

filtered_results['Pace'] = filtered_results['Pace'].astype(float)
colors = filtered_results['Team'].map(color_map)
valid_colors = [color_map.get(team, 'gray') for team in␣
↪filtered_results['Team']]
plt.scatter(filtered_results['W'], filtered_results['Pace'], c=valid_colors)
labels = color_map.keys()
handles = [plt.scatter([], [], c=color_map[label], label=label) for label in␣
↪labels]
plt.legend(handles, labels, loc='best')
plt.legend(loc='upper left', bbox_to_anchor=(1.05, 1))

plt.xlabel('Wins')
plt.ylabel('Pace')
plt.title('Wins vs Pace')
plt.show()
```

```
/tmp/ipykernel_841/2482424528.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

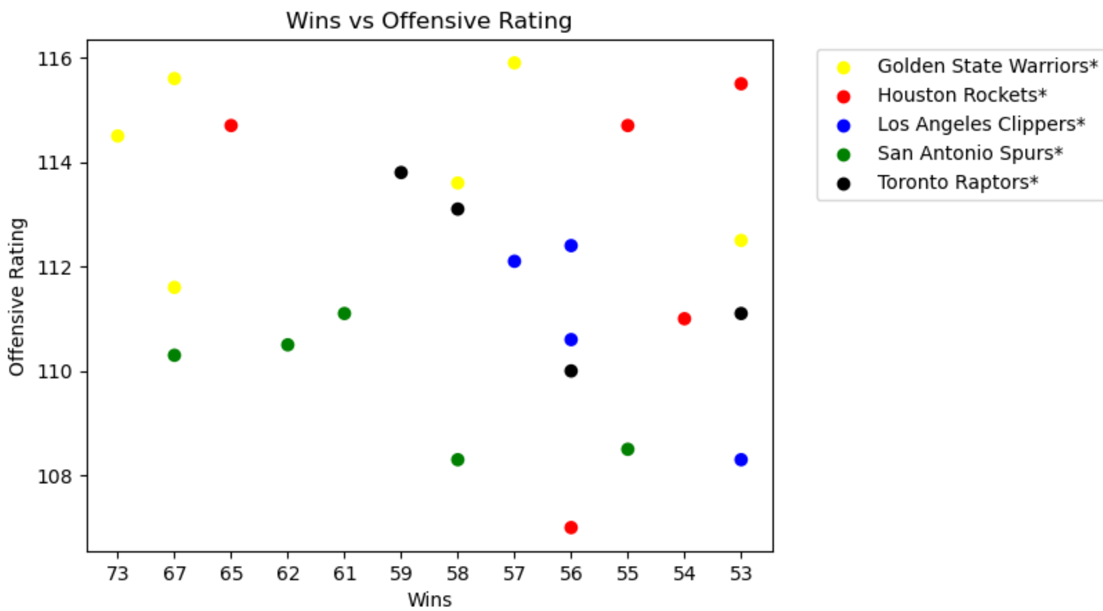
```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_results['Age'] = filtered_results['Age'].astype(float)
```



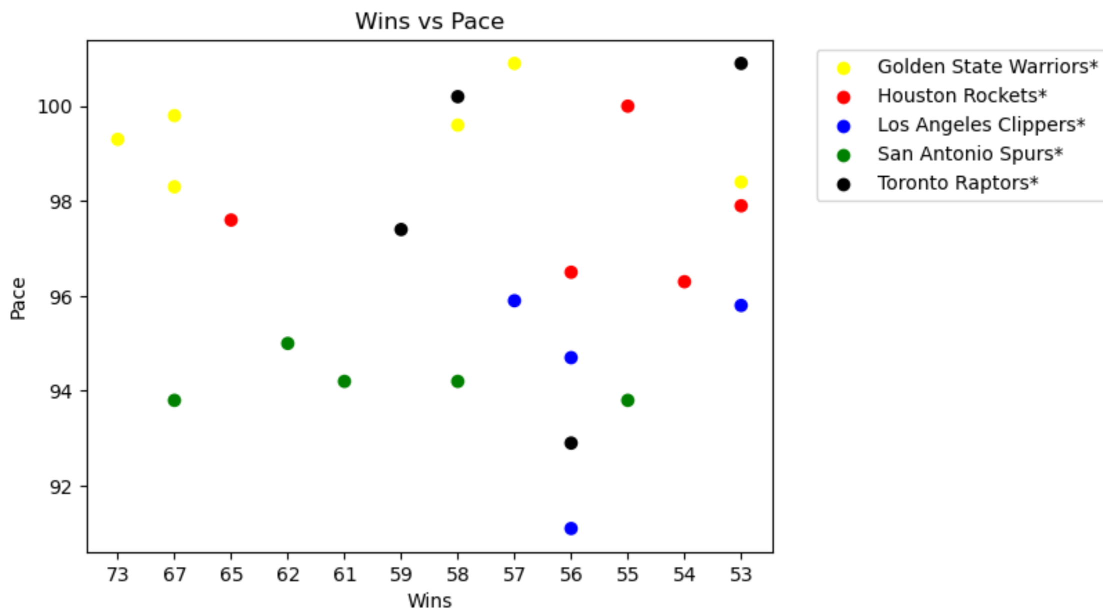

```
/tmp/ipykernel_841/2482424528.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_results['DRtg'] = filtered_results['DRtg'].astype(float)
```




```
/tmp/ipykernel_841/2482424528.py:39: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_results['ORtg'] = filtered_results['ORtg'].astype(float)
```



```
/tmp/ipykernel_841/2482424528.py:56: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_results['Pace'] = filtered_results['Pace'].astype(float)
```



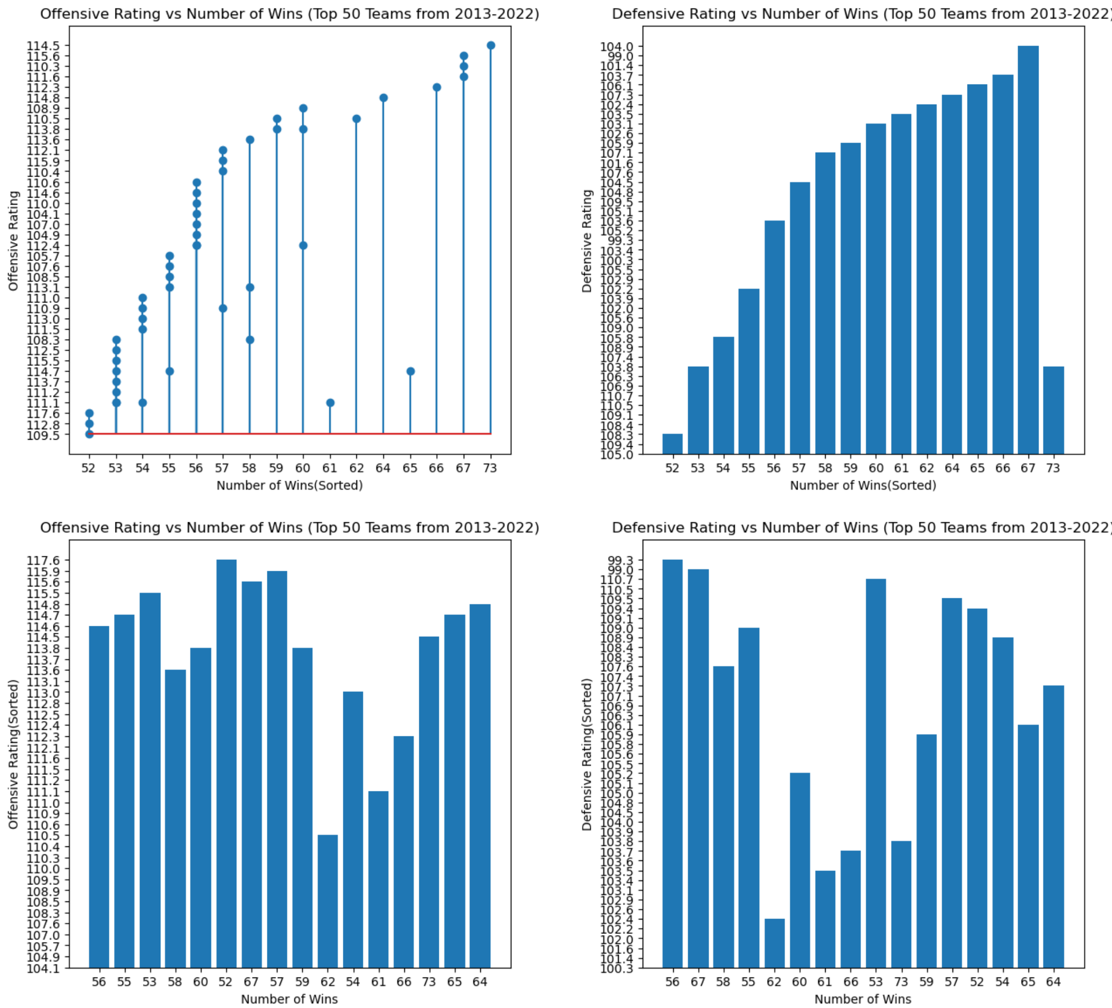
Analysis of the top five teams and their stats. When looking at the plots above, there seems to be no strong correlation between any of the stats and the win percentage. For less obvious stats such as age or pace this makes sense (an older team does not necessarily mean more wins nor does a fast paced team always beat out a slower one who can control the pace). However, it is surprising that defensive rating does not have a strong correlation with wins. Many teams specifically Golden State have poor ratings compared to the other four but still excel in wins. However, after looking at the offensive rating graph, we can see Golden State has a high offensive rating which may counteract the low defensive ratings.

We will explore the offense and defensive ratings of all the teams because that is an essential part of the sport We will see if there are trends and correlations to the number of wins to these ratings. Now let us plot both and compare into two new data frames.

[262]:
```
# create a figure with two subplots, arranged side by side
fig, axs = plt.subplots(ncols=2, nrows=2, figsize=(15, 14))



'''
```

```
    Below are looking at the correlation between Offensive rating vs Number of␣
  ↪Wins
'''
# plot the first subplot
sortedWins = result.sort_values('W')
axs[0][0].stem(sortedWins['W'], sortedWins['ORtg'])
axs[0][0].set_xlabel('Number of Wins(Sorted)')
axs[0][0].set_ylabel('Offensive Rating')
axs[0][0].set_title('Offensive Rating vs Number of Wins (Top 50 Teams from␣
  ↪2013-2022)')


# plot the 2nd plot
sortedORTG = result.sort_values('ORtg')
axs[1][0].bar(sortedORTG['W'], sortedORTG['ORtg'])
axs[1][0].set_xlabel('Number of Wins')
axs[1][0].set_ylabel('Offensive Rating(Sorted)')
axs[1][0].set_title('Offensive Rating vs Number of Wins (Top 50 Teams from␣
  ↪2013-2022)')




'''
    Below are looking at the correlation between Defensive Rating vs Number of␣
  ↪Wins
'''
# plot the first subplot
sortedWins = result.sort_values('W')
axs[0][1].bar(sortedWins['W'], sortedWins['DRtg'])
axs[0][1].set_ylabel('Defensive Rating')
axs[0][1].set_xlabel('Number of Wins(Sorted)')
axs[0][1].set_title('Defensive Rating vs Number of Wins (Top 50 Teams from␣
  ↪2013-2022)')


# # plot the 2nd plot
sortedDRTG = result.sort_values('DRtg')
axs[1][1].bar(sortedDRTG['W'], sortedDRTG['DRtg'])
axs[1][1].set_ylabel('Defensive Rating(Sorted)')
axs[1][1].set_xlabel('Number of Wins')
axs[1][1].set_title('Defensive Rating vs Number of Wins (Top 50 Teams from␣
  ↪2013-2022)')
```

```python
# adjust the spacing between the subplots
plt.subplots_adjust(wspace=0.3)


# display the plot
plt.show()
```



5.1 About the Graphs

Looking at the graphs above, two of them display the offensive rating vs the number of wins(left side) between the top 50 teams from 2013-2022. The other two graphs(right side) displays the defensive rating vs the number of wins between the top 50 teams from 2013-2022. The reason why there are two graphs for both is because the way how the graphs were originally displayed does not help depict the rating vs the wins/losses well. So to fix this, we decided to sort one of the axes in one graph and sort one of the other axes on the other graph. Also notice how the graphs aren't specifically displaying 50 different data points(bars). Each bar represents one of the top 50 playing teams between 2013-2022. If there is a repeat of the team who ranks top 50 between the time period, we take the mean of the offensive rating and number of wins; remember that we're specially looking at the correlation between the ratings and wins therefore we will not display which team is represented by the bar.

5.2 Analysis and Correlation Between Offensive and Defensive Rating

Offensive Looking at the two left graphs, they help us determine if there is a correlation between the offensive rating and the number of wins. So let's first look at the difference between the most victorious team and the least victorious team in the top 50. Looking at the graphs, there is a team who racked up 73 wins and a team who racked up 52 wins. Assume team A is the team with 73 wins and team B is the team with 52 wins. Looking at their offensive ratings, team A has an offensive rating of 114.5 and team B has an offensive rating of 112.8. Looking at both of those alone, it doesn't tell us much since their offensive ratings are very close to each other. So… let's look at the other teams. Is there a correlation? Looking at teams who are in the middle(around the high 50s and mid 60s) there doesn't seem to be a correlation between the offensive ratings. Looking at the team with 67 wins, it has a rating of 117.7, which is the highest between the rest. Then if we were to look at the team with 62 wins, it has a rating of 110.5 which is the lowest. The next lowest rating team is a team who racked up 61 wins. Okay so should the teams in the 50's win mark be lower in rating? No, they're actually one of the top rating teams. The teams who had 53,55,56 wins had some of the highest ratings in comparison to others(ratings above 114). So we can conclude here that the higher the offensive rating, the higher the number of wins a team has.

Defensive Knowing that there is no correlation between a teams offensive rating and defensive rating, is there a correlation between the number of wins and a teams' defensive rating? Looking at teams who have 60+ wins, it looks like their most ratings fall below 108. On the other hand, most teams who are sub-60 wins have some of the highest defensive ratings in comparison to other teams. For example, let's look at the following:

Wins | Defensive Rating 61 103.5 62 102.4 66 103.7

52 109.5 55 109.0 56 99.3 58 107.6

As we can see, looking at 6 of the data points there doesn't seem to be a correlation between the defensive rating and the number of wins as well.

Conclusion Originally before creating these graphs, we thought that a team's rating would be one of the factors to help determine whether a team would be more victorious in its games. After analyzing the graphs of the offensive ratings and defensive ratings, they do not help a teams' ability to win games. There seems to be other factors that fall in play when helping a team to win games.

Now we dive deeper into the offense as we explored that there is not a big difference in the offensive rating in corelation to wins once we reach a certain threshold. It makes us curious on why this is thus Let us explore into that more and see why that is!

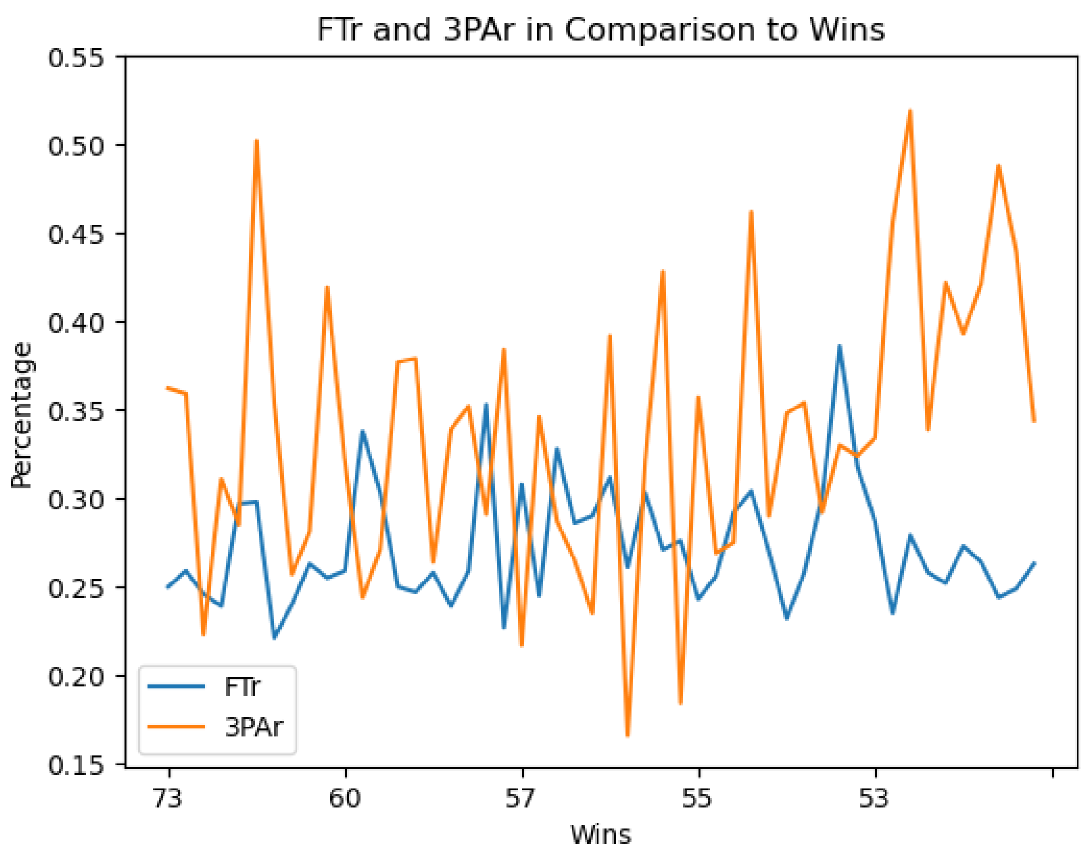
The main part of offense in basketball is how well a team shoots the ball. So we will view the FT and 3PA (free throws and 3 point ratings) and see if the higher they are the more it lines up with wins.

[153]:
```
display(result[['W','FTr', '3PAr', 'TS%']])
```

	W	FTr	3PAr	TS%
94	73	0.250	0.362	0.593
124	67	0.259	0.359	0.597
93	67	0.246	0.223	0.564
62	67	0.239	0.311	0.571
1	66	0.297	0.285	0.588
155	65	0.298	0.502	0.590
280	64	0.221	0.354	0.581
31	62	0.240	0.257	0.571
125	61	0.263	0.281	0.564
186	60	0.255	0.419	0.583
65	60	0.259	0.321	0.563
0	60	0.338	0.244	0.580
33	59	0.303	0.271	0.566
156	59	0.250	0.377	0.575
188	58	0.247	0.379	0.579
3	58	0.258	0.264	0.568
157	58	0.239	0.339	0.603
96	57	0.259	0.352	0.558
32	57	0.353	0.291	0.567
187	57	0.227	0.384	0.596
4	57	0.308	0.217	0.549
282	56	0.245	0.346	0.553
97	56	0.328	0.287	0.552
2	56	0.286	0.265	0.557
36	56	0.290	0.235	0.535
68	56	0.312	0.392	0.548
6	56	0.261	0.166	0.514
63	56	0.303	0.322	0.565
217	56	0.271	0.428	0.583
69	55	0.276	0.184	0.531
160	55	0.243	0.357	0.552
64	55	0.256	0.269	0.555
95	55	0.292	0.275	0.565
126	55	0.304	0.462	0.583
38	54	0.270	0.290	0.548
193	54	0.232	0.348	0.558
5	54	0.258	0.354	0.550
34	54	0.300	0.292	0.590
37	54	0.386	0.330	0.571
98	53	0.318	0.324	0.556
66	53	0.287	0.334	0.557
283	53	0.235	0.456	0.582
190	53	0.279	0.519	0.581
192	53	0.258	0.339	0.568

```
284   53   0.252   0.422   0.584
131   53   0.273   0.393   0.567
220   53   0.264   0.421   0.574
248   52   0.244   0.488   0.597
285   52   0.249   0.439   0.572
159   52   0.263   0.344   0.568
```

[121]:
```
# We change the data in table to numeric values
result['FTr'] = result['FTr'].astype(float)
result['3PAr'] = result['3PAr'].astype(float)

# Plot all three on one graph
result.plot(x='W', y=['FTr', '3PAr'], kind='line')
# set the y-axis tick intervals
plt.yticks(np.arange(.15, .6, 0.05))  # start, stop, step
plt.xlabel('Wins')
plt.ylabel('Percentage')
plt.title('FTr and 3PAr in Comparison to Wins')
plt.show()
```

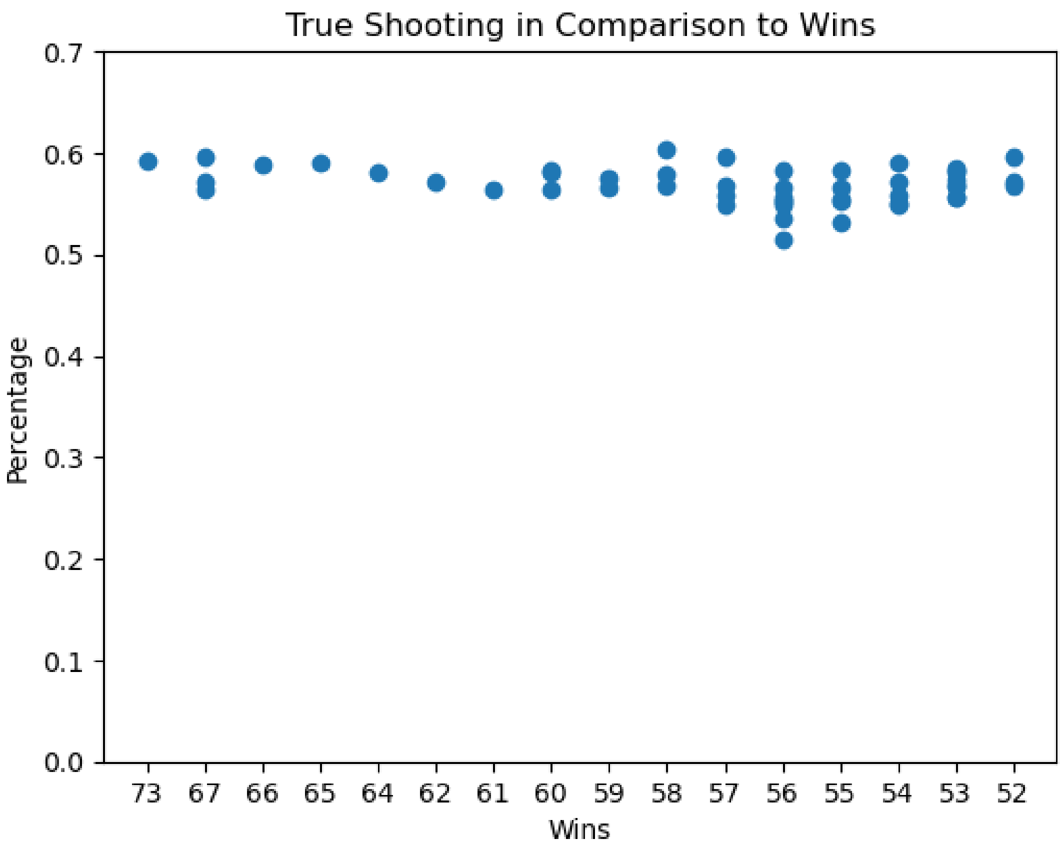


As we can see the 73 win team actually does not have a high FTr and 3PA compared to the others.

In fact, we see that even the lower win teams are up there in these stats. On average, the middle win teams average the highest numbers of these stats. This makes sense with our conclusion earlier, because it shows that the offensive rating does not correlate to number of wins once a team reaches a certain offensive rating threshold.

There is a true shooting % which calculates how a team scores overall, through all methods of scoring We will see if there is any trends or points we notice when we look at the graph.

[124]:
```
result['TS%'] = result['TS%'].astype(float)
plt.scatter(result['W'], result['TS%'])
plt.yticks(np.arange(0, .8, 0.1))  # start, stop, step
plt.xlabel('Wins')
plt.ylabel('Percentage')
plt.title('True Shooting in Comparison to Wins')
plt.show()
```



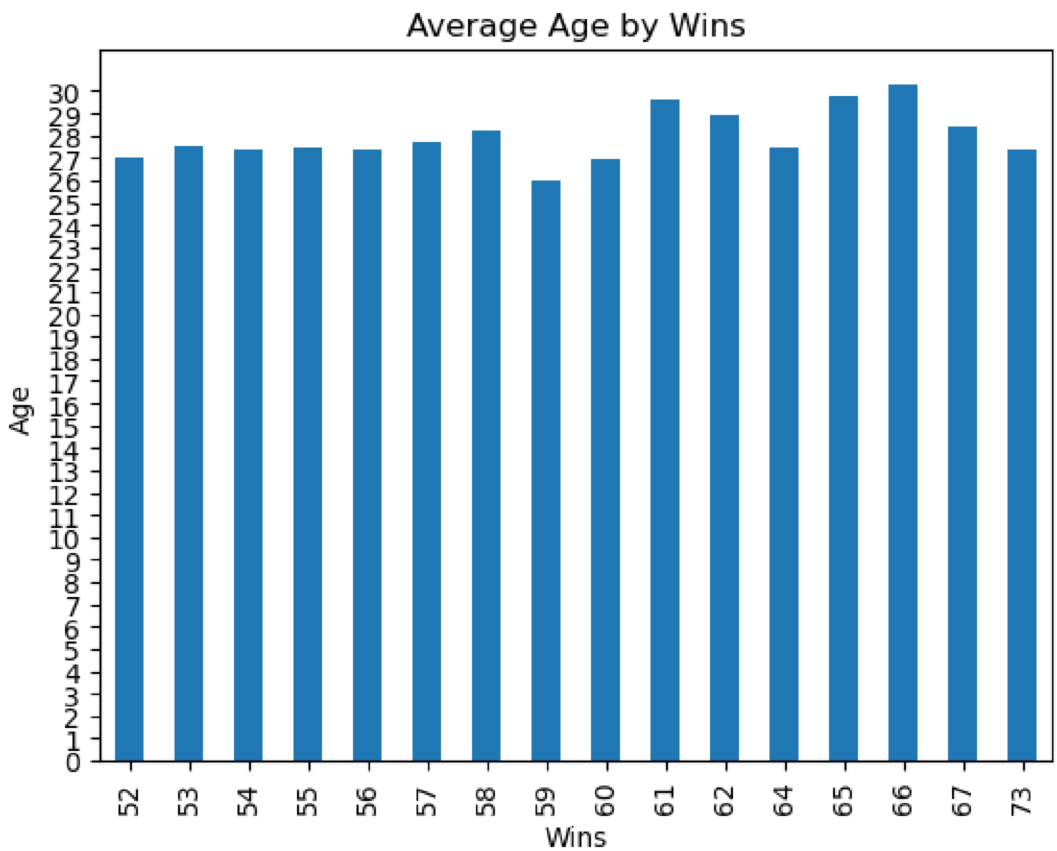
We notice that for the most part, all the teams have a good true shooting. Wheter they have 73 wins or 53 wins for almost all of the top 50 teams have a high and similar true shooting. Each of which are within .05% of each other which shows the commonality amongst all the winning teams.

5.2.1 What about Age?

This is the only non offensive or defensive statistic, and it makes you wonder if it matters if the average age of the team matters on amount of wins? Is it better to have old experienced players or youthful players who can play at a high pace for longer?

We will see if there is any correlation, and if there is, this is one of the statistics a team can always control.

[144]:
```
result['Age'] = result['Age'].astype(float)
age_means = result.groupby('W')['Age'].mean()
age_means.plot(kind='bar')
plt.xlabel('Wins')
plt.ylabel('Age')
plt.title('Average Age by Wins')
plt.yticks(range(0, int(max(age_means))+1, 1))
plt.show()
```



Now let us compare this to what the average age of all the succesful teams are

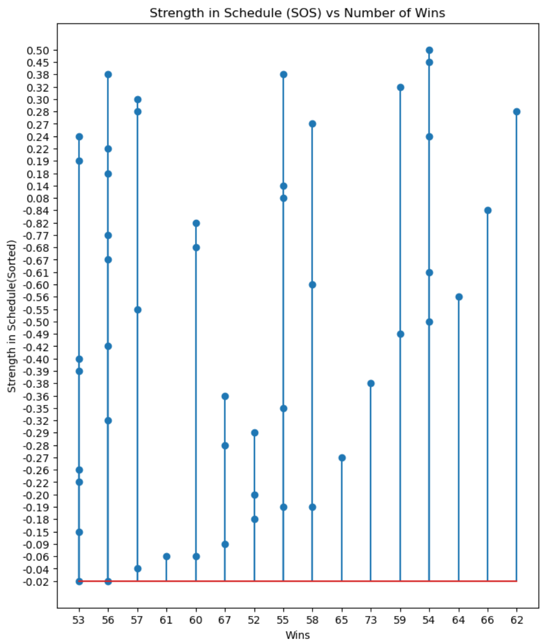
[147]:
```
age_avg = result['Age'].mean()
print("The average age is:", age_avg)
```

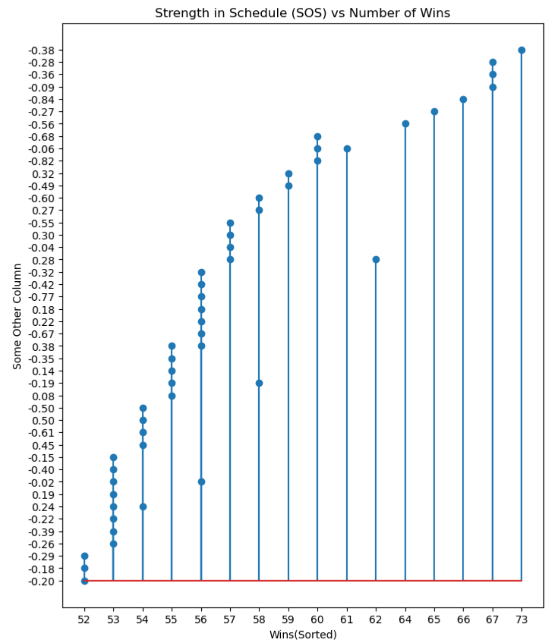
```
The average age is: 27.622
```

So what we can conclude is that the average team age is 27 for the top 50 teams. If we analyze the data table we see that the more successful teams all are above 27 years of age. They are around 27-30 but never under this. This means that experience matters a lot but there is definetly a cap of when the age is too much.

Another thing is there is a stat where a team can not control and it is based on the schedule of the teams they play his is called (SOS) and it is a given score based on how "difficult" a teams schedule is. We are curious to see if this really matters, and if a team with an easier schedule means they win more.

[156]:
```python
sortedSOS = result.sort_values('SOS')


# Create a figure and two subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(18,10))


# Plot the scatter plot on the first subplot (ax1)
ax1.stem(sortedSOS['W'], sortedSOS['SOS'])
ax1.set_title("Strength in Schedule (SOS) vs Number of Wins")
ax1.set_xlabel("Wins")
ax1.set_ylabel("Strength in Schedule(Sorted)")




sortedWins = result.sort_values('W')
# Plot a different type of plot on the second subplot (ax2)
# For example, let's plot a line plot
ax2.stem(sortedWins['W'], sortedWins['SOS'])
ax2.set_title("Strength in Schedule (SOS) vs Number of Wins")
ax2.set_xlabel("Wins(Sorted)")
ax2.set_ylabel("Some Other Column")


# Adjust spacing between subplots
plt.subplots_adjust(hspace=0.5)


# Display the plot
plt.show()
```





So, is there a correlation between the strength of a team's schedule and the number of wins they achieve? Let's examine teams that have 53 wins as an example. In the dataset, there appear to be several teams with 53 wins, represented by individual dots on the graph. When we analyze the line that represents teams with 53 wins (the left graph), we observe that the data points are widely scattered. The range of values extends as low as -0.02 and as high as 0.24.

Now, let's consider other teams with different win totals. If we examine lines representing teams with 56, 60, 54, and so on, we find that they also exhibit scattered patterns. Additionally, when we examine lines with only one data point, such as 61, 62, 64, 65, and 73 wins, they too display scattered distributions.

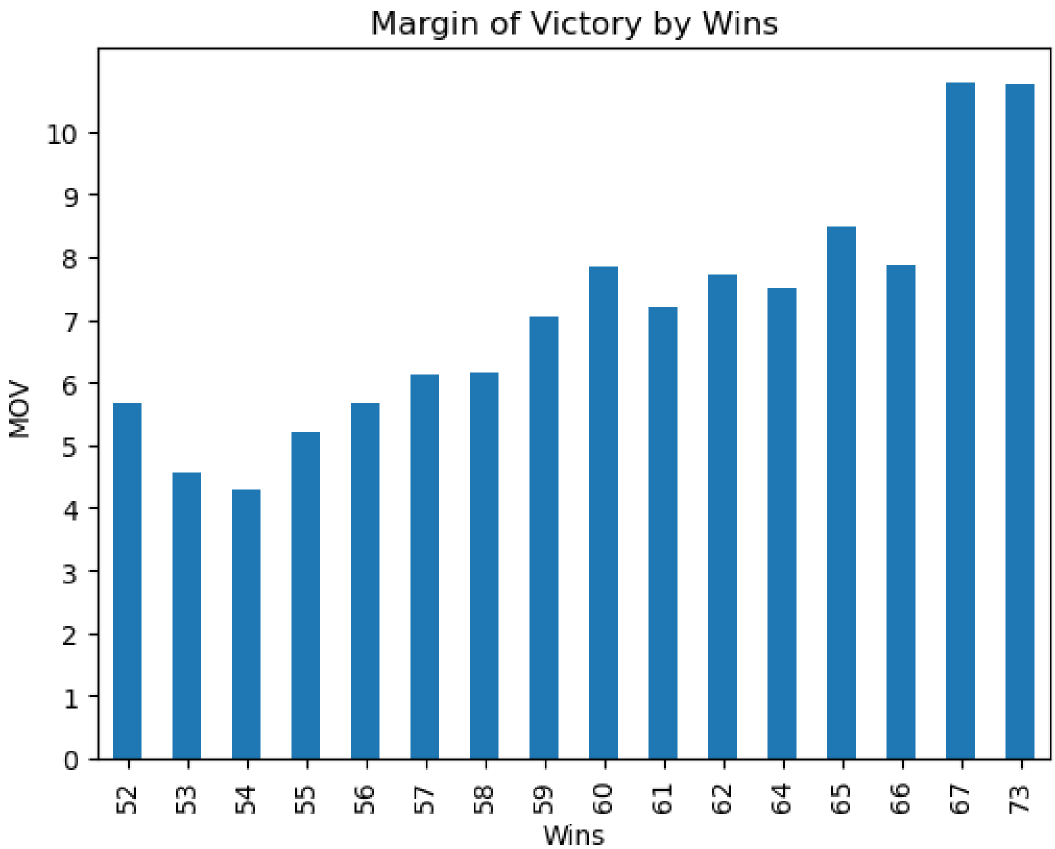
Based on these observations, it appears that there is no clear correlation between the strength of a team's schedule (measured by SOS - Strength of Schedule) and the number of wins they achieve. This conclusion is supported by the fact that the SOS of a team with 66 wins falls between the SOS values of teams with 65 wins and 52 wins, further indicating the lack of a consistent relationship between schedule strength and win totals.

Essentially stating that with the top teams, they are so dominant the the schedule and the teams they play do not play a factor!

We wanna see if the Margin Of Victory (MOV) matters at all because this in summary is everything we have looked at both offense and defense intertwined. We now will compare them via a bar graph because we want to see the trends and if there is a correlation between them.

[150]:
```
result['MOV'] = result['MOV'].astype(float)
movwins = result.groupby('W')['MOV'].mean()
movwins.plot(kind='bar')
plt.xlabel('Wins')
```

```
plt.ylabel('MOV')
plt.title('Margin of Victory by Wins')
plt.yticks(range(0, int(max(age_means))+1, 1))
plt.show()
```



We see a constant trend here! As the wins go up so does the margin of victory. This is expected because that means the teams offense to defense is high which would lead to more wins (theoretically).

We now have explored statistics, team controlled variables and non controllable values in correspondence to team wins! We have seen the trends, common factors and other variables that might affect the teams wins.

A question we had all this time was how do these stats compare to the league average over these years. This will further provide us how much of a skill gap there is between the top teams.

[201]:
```
url2 = 'https://www.basketball-reference.com/leagues/NBA_2015.
 ↪html#advanced-team'
res2 = requests.get(url2)
soup2 = BeautifulSoup(res2.content, 'html.parser')
```

```
tableref = soup2.find('table', {'id': 'advanced-team'})
data = []

# get the column names
headers = tableref.find_all('tr')[1]
columns = [header.text.strip() for header in headers.find_all('th')]

# iterate over the rows and cells of the table
for row in tableref.find_all('tr')[2:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)

# create a pandas dataframe from the extracted data
overall = pd.DataFrame(data, columns=columns)
```

[209]:

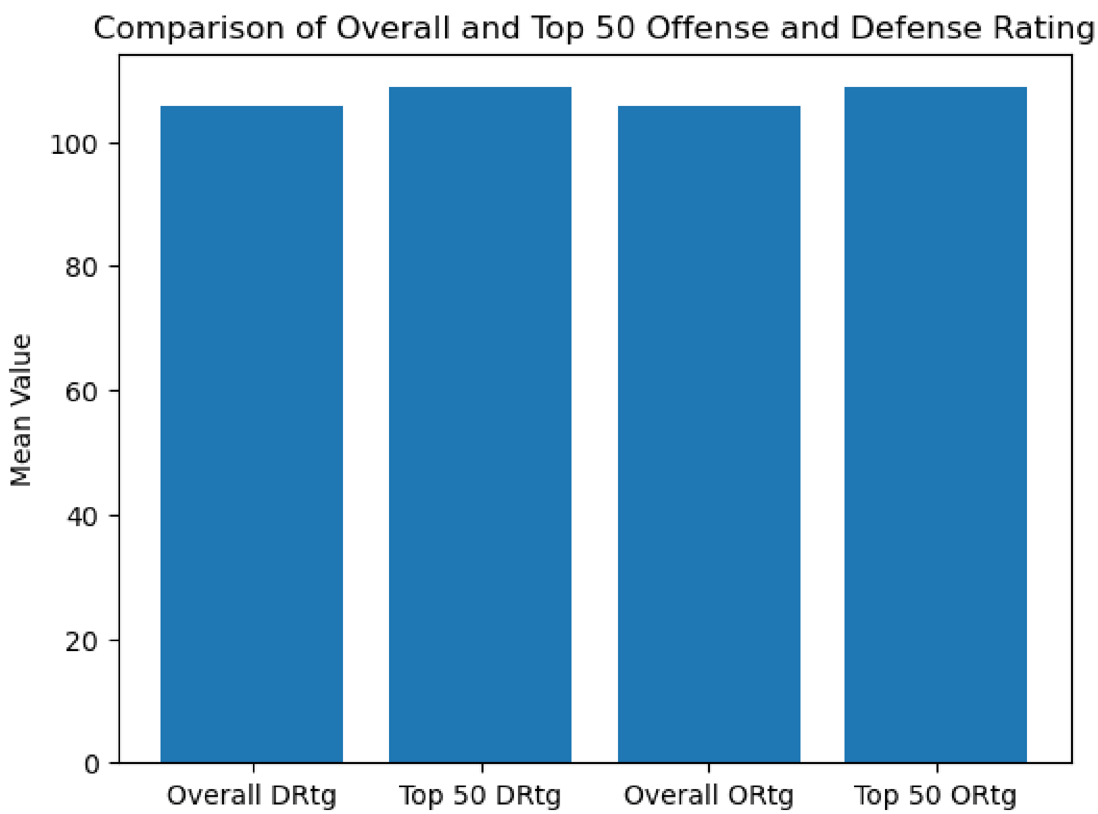
```
overall['TS%'] = overall['TS%'].astype(float)
result['TS%'] = result['TS%'].astype(float)

ovTS = overall['TS%'].mean()
topTS = result['TS%'].mean()
print('This is 2022 Seasons TS%:', ovTS)
print('This is the top 50 teams TS%:', topTS)

overall['ORtg'] = overall['ORtg'].astype(float)
result['ORtg'] = result['ORtg'].astype(float)

ovortg = overall['ORtg'].mean()
toportg = result['ORtg'].mean()
print('This is 2022 Seasons TS%:', ovortg)
print('This is the top 50 teams TS%:', toportg)

overall['DRtg'] = overall['DRtg'].astype(float)
result['DRtg'] = result['DRtg'].astype(float)

ovdef = overall['DRtg'].mean()
topdef = result['DRtg'].mean()
print('This is 2022 Seasons DTrg:', ovdef)
print('This is the top 50 teams DTrg:', topdef)
```

```
This is 2022 Seasons TS%: 0.5341290322580645
This is the top 50 teams TS%: 0.5521516129032257
This is 2022 Seasons TS%: 105.63870967741936
This is the top 50 teams TS%: 108.71774193548387
This is 2022 Seasons DTrg: 105.64838709677417
```

```
This is the top 50 teams DTrg: 108.72806451612902
```

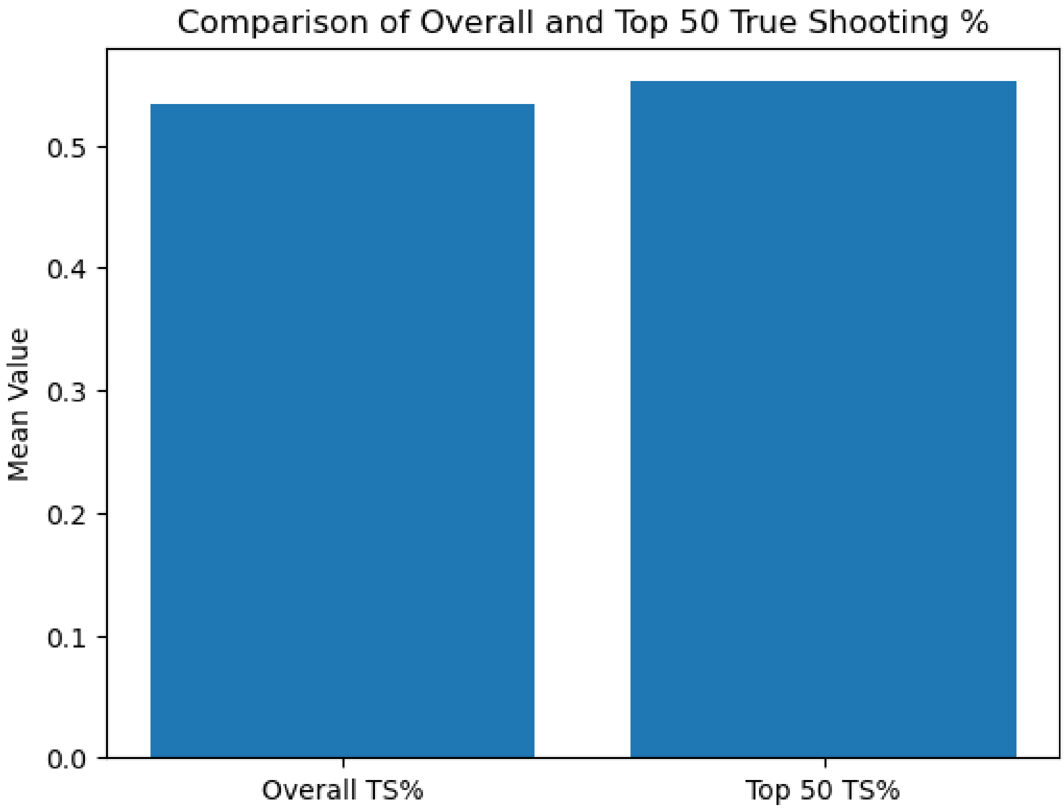
[212]:

```python
x_labels = ['Overall DRtg', 'Top 50 DRtg', 'Overall ORtg', 'Top 50 ORtg']
y_values = [ovdef, topdef, ovortg, toportg]

plt.bar(x_labels, y_values)
plt.title('Comparison of Overall and Top 50 Offense and Defense Rating')
plt.ylabel('Mean Value')
plt.show()
```



[213]:

```python
x_labels = ['Overall TS%', 'Top 50 TS%']
y_values = [ovTS, topTS]

plt.bar(x_labels, y_values)
plt.title('Comparison of Overall and Top 50 True Shooting %')
plt.ylabel('Mean Value')
plt.show()
```



We see here that the top 50 teams compared to a random year in the mdoern era are all better in the main stats It might not seem like a whole lot more but perentages and ratings in sports are very tight and even 1 or 2 points off a rating is very big.

6 Model: Analysis, Hypothesis Testing

We have captured various modeling, statistics and calculations to be used to obtain a predicitive model of our data We will attempt multiple modeling strategies to obtain these models, which will allow us to predict values that we do not have Here we will try and predict how many wins the Golden State Warriors will have this 2023 season.

[238]:
```
urlgsw = 'https://www.basketball-reference.com/teams/GSW/#GSW'
resgsw = requests.get(urlgsw)
soup3 = BeautifulSoup(resgsw.content, 'html.parser')

tablegsw = soup3.find('table')
data = []

# get the column names
headers = tablegsw.find_all('tr')[0]
columns = [header.text.strip() for header in headers.find_all('th')]
```

```
# iterate over the rows and cells of the table
for row in tablegsw.find_all('tr')[1:]:
    row_data = []
    for cell in row.find_all(['th', 'td']):
        row_data.append(cell.text.strip())
    if row_data:
        data.append(row_data)

# create a pandas dataframe from the extracted data
gsw = pd.DataFrame(data, columns=columns)
gsw = gsw.drop(index=0)
gsw = gsw.head(11)
gsw
```

[238]:

```
     Season   Lg                     Team   W   L  W/L%    Finish    SRS  \
1   2021-22  NBA  Golden State Warriors*  53  29  .646  2nd of 5   5.52
2   2020-21  NBA   Golden State Warriors  39  33  .542  4th of 5   1.10
3   2019-20  NBA   Golden State Warriors  15  50  .231  5th of 5  -8.12
4   2018-19  NBA  Golden State Warriors*  57  25  .695  1st of 5   6.42
5   2017-18  NBA  Golden State Warriors*  58  24  .707  1st of 5   5.79
6   2016-17  NBA  Golden State Warriors*  67  15  .817  1st of 5  11.35
7   2015-16  NBA  Golden State Warriors*  73   9  .890  1st of 5  10.38
8   2014-15  NBA  Golden State Warriors*  67  15  .817  1st of 5  10.01
9   2013-14  NBA  Golden State Warriors*  51  31  .622  2nd of 5   5.15
10  2012-13  NBA  Golden State Warriors*  47  35  .573  2nd of 5   1.32
11  2011-12  NBA   Golden State Warriors  23  43  .348  4th of 5  -2.79

     Pace Rel Pace   ORtg Rel ORtg   DRtg Rel DRtg                  Playoffs  \
1    98.4      0.2  112.5      0.5  106.9     -5.1                Won Finals
2   102.2      3.0  111.1     -1.2  110.1     -2.2
3   100.3      0.0  105.2     -5.4  113.8      3.2
4   100.9      0.9  115.9      5.5  109.5     -0.9               Lost Finals
5    99.6      2.3  113.6      5.0  107.6     -1.0                Won Finals
6    99.8      3.4  115.6      6.8  104.0     -4.8                Won Finals
7    99.3      3.5  114.5      8.1  103.8     -2.6               Lost Finals
8    98.3      4.4  111.6      6.0  101.4     -4.2                Won Finals
9    96.2      2.3  107.5      0.8  102.6     -4.1  Lost W. Conf. 1st Rnd.
10   94.5      2.5  106.4      0.5  105.5     -0.4      Lost W. Conf. Semis
11   92.3      1.0  105.4      0.8  109.1      4.5

              Coaches            Top WS
1     S. Kerr (53-29)    S. Curry (8.0)
2     S. Kerr (39-33)    S. Curry (9.0)
3     S. Kerr (15-50)   M. Chriss (3.4)
4     S. Kerr (57-25)  K. Durant (11.5)
5     S. Kerr (58-24)  K. Durant (10.4)
```

```
6       S. Kerr (67-15)   S. Curry (12.6)
7        S. Kerr (73-9)   S. Curry (17.9)
8       S. Kerr (67-15)   S. Curry (15.7)
9    M. Jackson (51-31)   S. Curry (13.4)
10   M. Jackson (47-35)   S. Curry (11.2)
11   M. Jackson (23-43)      D. Lee (5.0)
```

We only need certain columns that match with our data, so we need to filter and modify this dataframe so we can compare data We proved that there is a certain threshold that teams need to be a succesfull team and that these statistics matter to a certain point.

[243]:
```python
tokeep = ['Team', 'W', 'Pace', 'ORtg', 'DRtg']
gsw = gsw.loc[:, tokeep]
gsw
```

[243]:

	Team	W	Pace	ORtg	DRtg
1	Golden State Warriors*	53	98.4	112.5	106.9
2	Golden State Warriors	39	102.2	111.1	110.1
3	Golden State Warriors	15	100.3	105.2	113.8
4	Golden State Warriors*	57	100.9	115.9	109.5
5	Golden State Warriors*	58	99.6	113.6	107.6
6	Golden State Warriors*	67	99.8	115.6	104.0
7	Golden State Warriors*	73	99.3	114.5	103.8
8	Golden State Warriors*	67	98.3	111.6	101.4
9	Golden State Warriors*	51	96.2	107.5	102.6
10	Golden State Warriors*	47	94.5	106.4	105.5
11	Golden State Warriors	23	92.3	105.4	109.1

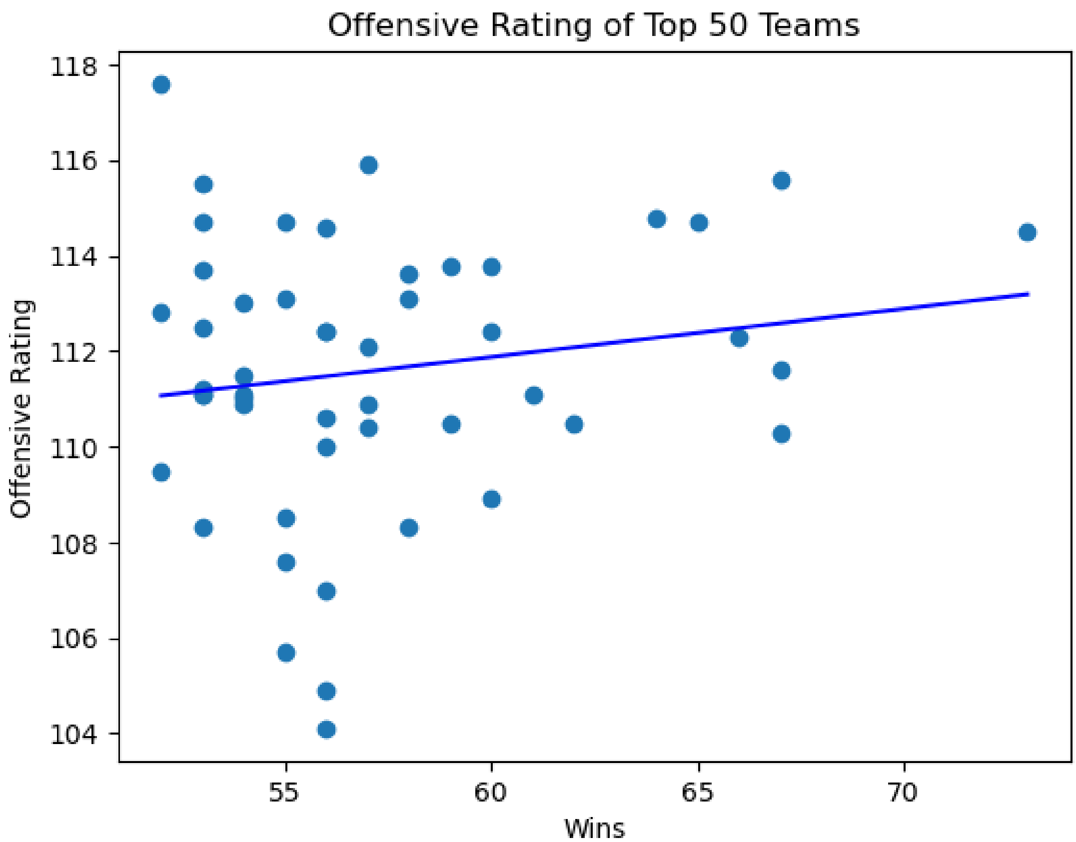
So we will calculate the average of these ratings / statistics and put them in correlation to our top 50 teams!

[263]:
```python
gsw['DRtg'] = gsw['DRtg'].astype(float)
gsw['ORtg'] = gsw['ORtg'].astype(float)
gsw['Pace'] = gsw['Pace'].astype(float)

gswdef = gsw['DRtg'].mean()
gswof = gsw['ORtg'].mean()
gswpace = gsw['Pace'].mean()
print('This is GSW Average DRtf:', gswdef)
print('This is GSW Average OTrg:', gswof)
print('This is GSW Average Pace:', gswpace)
```

```
This is GSW Average DRtf: 106.75454545454544
This is GSW Average OTrg: 110.84545454545456
This is GSW Average Pace: 98.34545454545454
```

We calculated the ORtg plot of the top 50 teams and created a line of best fit

[273]:
```
x = result['W'].astype(float)
y = result['ORtg'].astype(float)
slope, intercept = np.polyfit(x, y, 1)
plt.scatter(x, y)
plt.plot(x, slope * x + intercept, color='blue')
plt.xlabel('Wins')
plt.ylabel('Offensive Rating')
plt.title('Offensive Rating of Top 50 Teams')
plt.show()
```

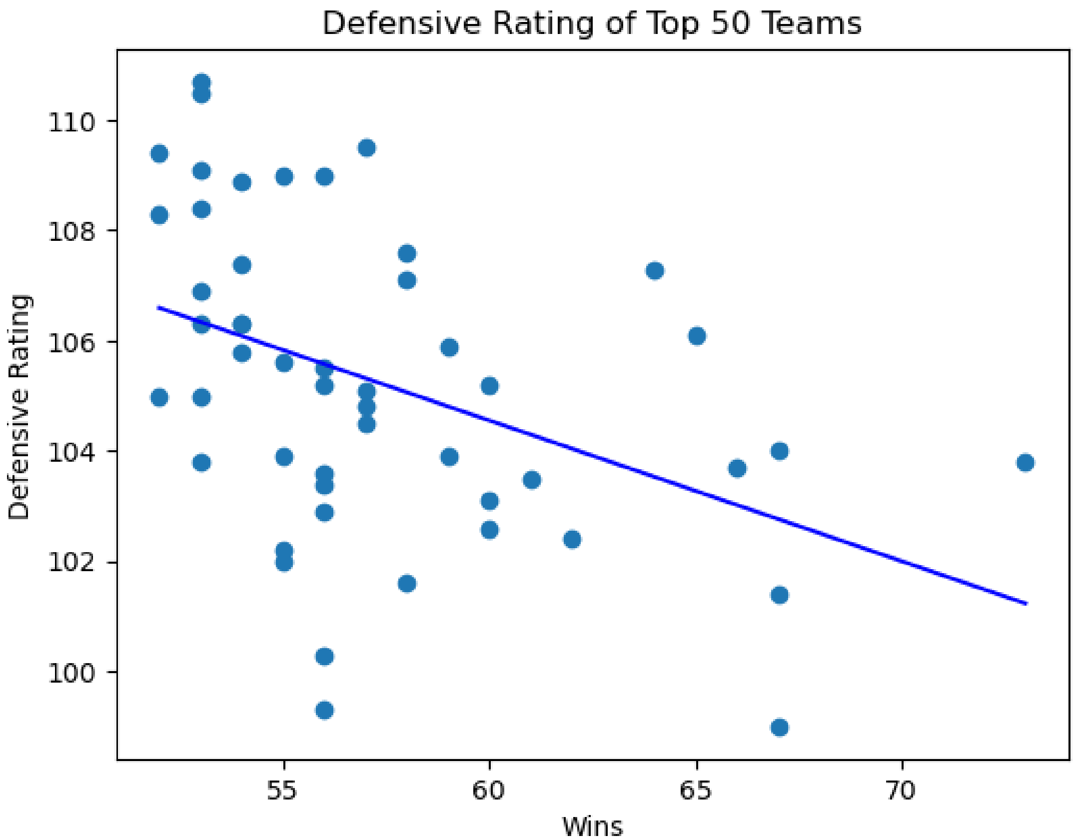


If we were to place the OTrg score of 110 on this chart we will see it equates to around 57 wins

We calculated the DRtg plot of the top 50 teams and created a line of best fit

[274]:
```
x = result['W'].astype(float)
y = result['DRtg'].astype(float)
slope, intercept = np.polyfit(x, y, 1)
plt.scatter(x, y)
plt.plot(x, slope * x + intercept, color='blue')
plt.xlabel('Wins')
plt.ylabel('Defensive Rating')
```

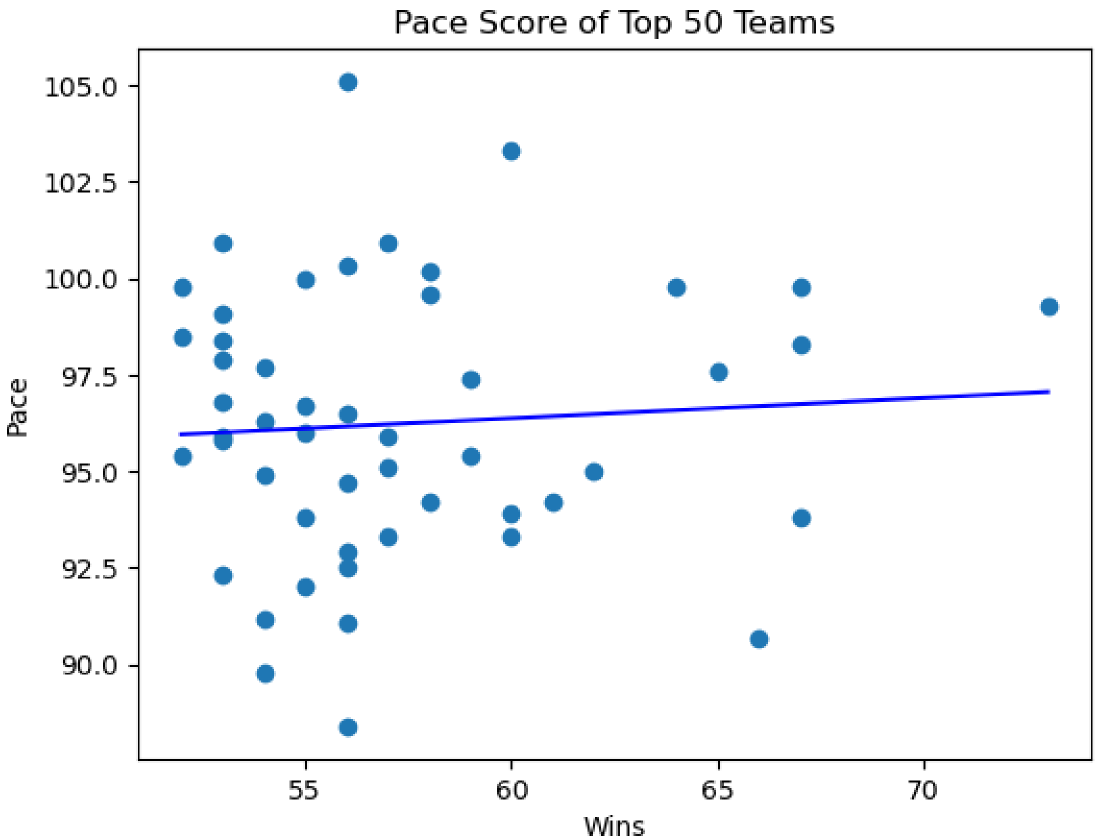
```
plt.title('Defensive Rating of Top 50 Teams')
plt.show()
```



If we were to place the DTrg score of 106 on this chart we will see it equates to around 54 wins

We calculated the Pace plot of the top 50 teams and created a line of best fit

[275]:
```
x = result['W'].astype(float)
y = result['Pace'].astype(float)
slope, intercept = np.polyfit(x, y, 1)
plt.scatter(x, y)
plt.plot(x, slope * x + intercept, color='blue')
plt.xlabel('Wins')
plt.ylabel('Pace')
plt.title('Pace Score of Top 50 Teams')
plt.show()
```



If we were to place the Pace score of 98 on this chart we will see it equates to an average of around 57 wins

If we take all this data, we have 57, 54, 57 which averages to a predicted 55 wins. We can see if that is right in the future.

7 Interpretation and Conclusions

In this portion, we will use the data and the data analysis to draw conclusions.

The main goal of this project was to see if any stats were highly correlated to more wins for an NBA team. To do this, we first pulled 10 years of regular season NBA data. The next step was tidying up the data. After sorting the data by the number of wins, we decided to only keep the top 50 performances from the past 10 years. This would allow us to look at what winning teams had in common, and see if any statistics correlated with them winning a lot. What we found was surprising.

Some statistics, logically, don't correlate to more wins. Some examples we tested were Age (older or younger teams don't necessarily mean better) and Pace (faster or slower pace doesn't mean better). After wrangling with the data and creating plots, we were able to see scattered distributions for both of these factors. There was no clear correlation or correlation at all that these factors helped the successful teams be successful. This part was not surprising to us.

However, what surprised us is the results that came out of studying factors that logically would make a team win more. We studied many of these factors, a few include offensive rating, defensive rating, the strength of schedule (how good/bad the opposing team is), and shooting statistics such as free throws, 3-pointers, and true shooting. None of these statistics showed a correlation to wins - teams with lower offensive ratings would win more than those with higher ones, teams with worse true shooting stats would win just as much as teams with better stats, and the same could be said for the other stats that would logically make a team win more.