CMSE202 Final Project Code 2024

September 28, 2024

[4]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
```

[5]:

```
nba = pd.read_csv("nba_team_stats_00_to_21.csv")
```

[6]:

```
nba
```

[6]:

```
     teamstatspk                   TEAM  GP   W   L   WIN%   MIN    PTS   FGM  \
0              0           Phoenix Suns  52  42  10  0.808  48.1  112.7  42.7
1              1  Golden State Warriors  53  40  13  0.755  48.2  110.9  40.4
2              2      Memphis Grizzlies  55  37  18  0.673  48.3  112.7  42.7
3              3             Miami Heat  54  34  20  0.630  48.5  108.7  39.3
4              4          Chicago Bulls  53  33  20  0.623  48.1  111.6  41.6
..           …                    …  ..  ..  ..    …   …    …   …
621          621          Atlanta Hawks  82  25  57  0.305  48.1   91.0  35.1
622          622    Vancouver Grizzlies  82  23  59  0.280  48.2   91.7  35.0
623          623     Washington Wizards  82  19  63  0.232  48.0   93.2  34.5
624          624  Golden State Warriors  82  17  65  0.207  48.2   92.5  35.8
625          625          Chicago Bulls  82  15  67  0.183  48.4   87.6  33.2

      FGA  …   REB   AST   TOV   STL  BLK  BLKA    PF   PFD  +/-   SEASON
0    89.4  …  46.1  26.5  13.3   8.6  4.3   4.0  19.3  19.3  7.8  2020-21
1    86.5  …  46.5  27.5  15.6   9.4  4.9   4.1  20.3  17.7  8.3  2020-21
2    93.4  …  48.6  25.1  13.3  10.1  6.4   6.4  19.1  19.0  4.1  2020-21
3    85.7  …  44.6  25.9  14.9   7.6  3.3   4.4  20.5  20.0  4.2  2020-21
4    87.0  …  43.0  24.5  13.0   7.2  4.6   5.2  18.8  17.8  1.7  2020-21
..    …  …   …   …   …   …  …   …   …    …  …      …
621  81.3  …  42.9  19.0  16.7   7.7  4.7   6.3  22.7   0.1 -5.2  2000-01
622  79.7  …  40.5  23.2  15.7   7.1  4.4   5.8  21.1   0.1 -5.7  2000-01
623  78.7  …  41.3  20.1  17.0   7.7  4.7   6.2  23.3   0.1 -6.7  2000-01
624  87.5  …  45.5  21.8  15.9   9.0  5.0   6.0  21.1   0.1 -9.0  2000-01
625  78.2  …  38.9  22.1  15.8   8.2  4.6   5.2  23.2   0.0 -9.1  2000-01

[626 rows x 29 columns]
```

[ ]:

```

```

[ ]:

```

```

[7]:

```
cols = nba.columns[6:28] # only grabbing certain columns
```

[8]:

```
cols
```

[8]:

```
Index(['MIN', 'PTS', 'FGM', 'FGA', 'FG%', '3PM', '3PA', '3P%', 'FTM', 'FTA',
       'FT%', 'OREB', 'DREB', 'REB', 'AST', 'TOV', 'STL', 'BLK', 'BLKA', 'PF',
       'PFD', '+/-'],
      dtype='object')
```

[9]:

```
def check_rsq3(STAT):

    '''Takes in the name of a stat category (Make sure it is in quotes).
    Computes the adjusted R squared value for the correlation between the given␣
 ↪stat and win percentage'''

    y2 = nba['WIN%']
    x2 = nba[STAT]
    x2 = sm.add_constant(x2)
    model2 = sm.OLS(y2, x2).fit()
    adj_rsquared = model2.rsquared_adj
    return adj_rsquared
```

[10]:

```
def rsq_list(columns):

    '''Takes in a list which should be a list of the names of stat categories␣
 ↪in quotes.
    Outputs a list of the adjusted R squared value for each stat category in␣
 ↪the list.
    Uses the checka_rsq3 function in a for loop to do this.'''

    r_sq_list = []

    for i in columns:
        r_sq_list.append(check_rsq3(i))

    return r_sq_list
```

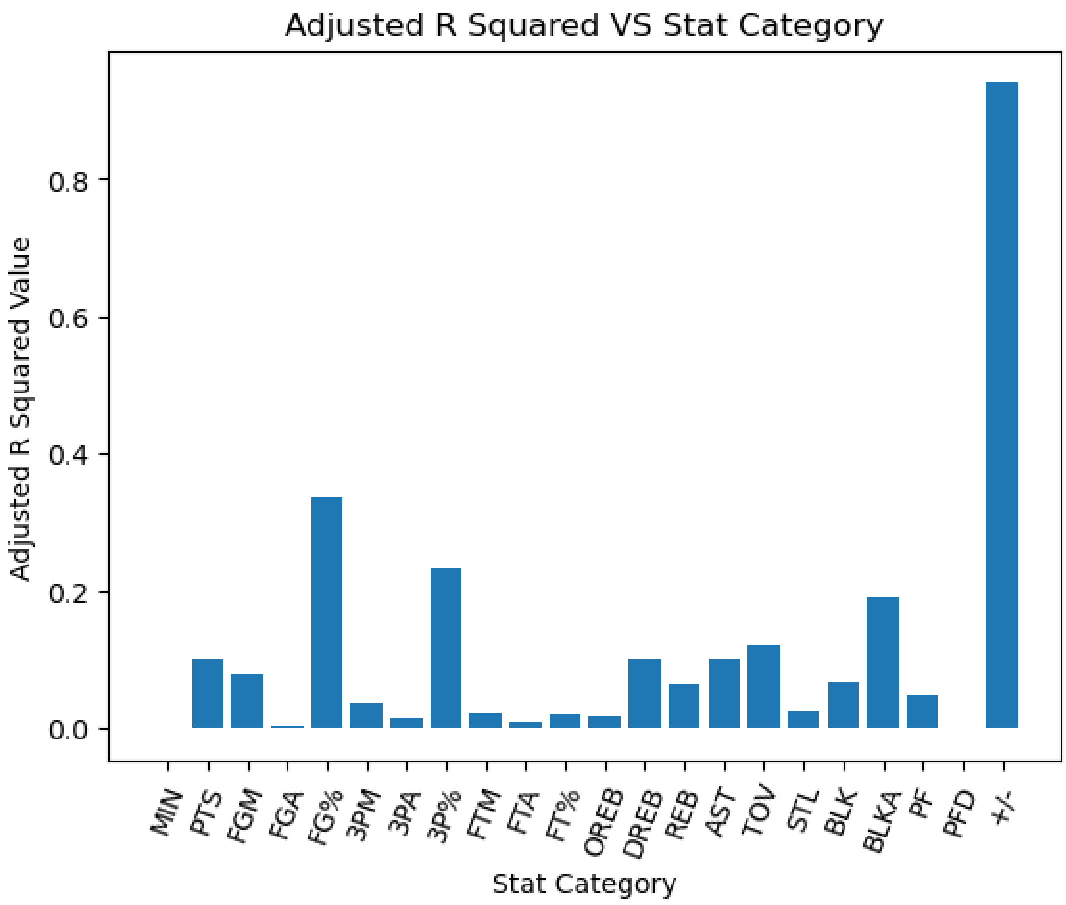
[11]:

```
rsq_list(cols)
```

```
[11]: [-0.0007161034580946968,
       0.10212079214043168,
       0.0797547086494963,
       0.0023059138647639754,
       0.33733407381405933,
       0.035857044015388695,
       0.013919707332485709,
       0.23222456885944753,
       0.021292305592412353,
       0.008555090526928155,
       0.02053293600113315,
       0.017609907814430614,
       0.10048567926123397,
       0.06341328979581184,
       0.10115040845320145,
       0.12023306707490633,
       0.02647656732635184,
       0.06737433637861123,
       0.18983618262447566,
       0.04797677970628644,
       -0.0014313480181882277,
       0.9398605077442408]
```

```
[12]: plt.bar(cols, rsq_list(cols))
      plt.xticks(rotation=70)
      plt.xlabel("Stat Category")
      plt.ylabel("Adjusted R Squared Value")
      plt.title("Adjusted R Squared VS Stat Category")
```

```
[12]: Text(0.5, 1.0, 'Adjusted R Squared VS Stat Category')
```



Chat GPT 3.5 https://chat.openai.com/c/c137f877-dc30-4bf1-b0b6-6c664cbdf3ac. Accessed 4/1/24. The prompt I used was: “I got a histogram but its labels block each other, how can I get it so that I can see all the labels?” This helped me to be able to rotate the labels so they can be seen better using plt.xticks.

[13]:
```
def headtohead(season, team1, team2):

    '''Takes in the year of the season in question in the format (ex. 2018-19)
    and the names of 2 teams in quotes.
    The function outputs which team is more likely to win and in how long the␣
 ↪best of 7 series should take;'''

    df = nba[nba['SEASON'] == season] # creates a new df for the season in␣
 ↪question
    team1 = df[df['TEAM'] == team1] # creates a new df for both teams in␣
 ↪question
    team2 = df[df['TEAM'] == team2]
    rating1 = ((team1['WIN%'] + team1['+/-']/5 + team1['FG%']/50 +  # pulls the␣
 ↪values from certain colums of the teams df
           team1['3P%']/50 + team1['BLK']/10).values[0]) # and adds the values␣
 ↪together to compute a rating for the team

    rating2 = ((team2['WIN%'] + team2['+/-']/5 + team2['FG%']/50 +
           team2['3P%']/50 + team2['BLK']/10).values[0])

    if rating1 > rating2: # Uses if statements to determine which team's rating␣
 ↪is higher and by how much
        if 1 < rating1/rating2 < 1.25: # This is so that it can decide how␣
 ↪close the series will be (how many games)
            print('pick', team1['TEAM'].values[0], 'in 7 games')
        if 1.25 < rating1/rating2 < 1.5:
            print('pick', team1['TEAM'].values[0], 'in 6 games')
        if 1.5 < rating1/rating2 < 1.75:
            print('pick', team1['TEAM'].values[0], 'in 5 games')
        if 1.75 < rating1/rating2:
            print('pick', team1['TEAM'].values[0], 'in 4 games')

    if rating2 > rating1:
        if 1 < rating2/rating1 < 1.25:
            print('pick', team2['TEAM'].values[0], 'in 7 games')
        if 1.25 < rating2/rating1 < 1.5:
            print('pick', team2['TEAM'].values[0], 'in 6 games')
        if 1.5 < rating2/rating1 < 1.75:
            print('pick', team2['TEAM'].values[0], 'in 5 games')
        if 1.75 < rating2/rating1:
            print('pick', team2['TEAM'].values[0], 'in 4 games')
```

[14]:

```
# ROUND 1
```

[91]:

```
headtohead("2014-15", 'Atlanta Hawks', 'Brooklyn Nets') # Correct winner but␣
 ↪went to 6 games
```

```
pick Atlanta Hawks in 4 games
```

[92]:

```
headtohead("2014-15", 'Toronto Raptors', 'Washington Wizards') # Incorrect,␣
 ↪Wizards won in 4 games
```

```
pick Toronto Raptors in 7 games
```

[93]:

```
headtohead("2014-15", 'Chicago Bulls', 'Milwaukee Bucks') # Correct, but went␣
 ↪to 6 games
```

```
pick Chicago Bulls in 7 games
```

```
[94]: headtohead("2014-15", 'Cleveland Cavaliers', 'Boston Celtics') # Correct, but␣
       ↪went to 4 games
```

```
pick Cleveland Cavaliers in 6 games
```

```
[95]: headtohead("2014-15", 'Golden State Warriors', 'New Orleans Pelicans') #␣
       ↪Correct, but went to 4 games
```

```
pick Golden State Warriors in 5 games
```

```
[96]: headtohead("2014-15", 'Portland Trail Blazers', 'Memphis Grizzlies') #␣
       ↪Incorrect, Grizzlies won in 5 games
```

```
pick Portland Trail Blazers in 7 games
```

```
[97]: headtohead("2014-15", 'Los Angeles Clippers', 'San Antonio Spurs') # Correct
```

```
pick Los Angeles Clippers in 7 games
```

```
[98]: headtohead("2014-15", 'Houston Rockets', 'Dallas Mavericks') # Correct, but␣
       ↪went to 5 games
```

```
pick Houston Rockets in 7 games
```

```
[99]: # ROUND 2
```

```
[100]: headtohead("2014-15", 'Atlanta Hawks', 'Toronto Raptors') # Correct, different␣
        ↪matchup
```

```
pick Atlanta Hawks in 7 games
```

```
[101]: headtohead("2014-15", 'Chicago Bulls', 'Cleveland Cavaliers') # Correct, but␣
        ↪went to 6 games
```

```
pick Cleveland Cavaliers in 7 games
```

```
[102]: headtohead("2014-15", 'Golden State Warriors', 'Portland Trail Blazers') #␣
        ↪Correct, different matchup
```

```
pick Golden State Warriors in 6 games
```

```
[103]: headtohead("2014-15", 'Houston Rockets', 'Los Angeles Clippers') # Incorrect,␣
        ↪Rockets won in 7 games
```

```
pick Los Angeles Clippers in 7 games
```

```
[104]: # ROUND 3
```

[105]:

```
headtohead("2014-15", 'Atlanta Hawks', 'Cleveland Cavaliers') # Incorrect,␣
 ↪Cavaliers won in 4 games
```

```
pick Atlanta Hawks in 7 games
```

[106]:

```
headtohead("2014-15", 'Golden State Warriors', 'Los Angeles Clippers') #␣
 ↪Correct, different matchup
```

```
pick Golden State Warriors in 7 games
```

[108]:

```
# FINALS
```

[109]:

```
headtohead("2014-15", 'Golden State Warriors', 'Atlanta Hawks') # Correct,␣
 ↪different matchup
```

```
pick Golden State Warriors in 6 games
```

[ ]:

```
# Filled out a bracket for 2015 using the function and compared the predicted␣
 ↪results to the actual results
# The model correctly predicted the winner of 11 out of 15 series.
```