

# Women's Super League

## Data Analysis

COURTNEY SWEENEY

Can any  
team stats  
**predict** where  
a team will  
end on the  
table?



# XML Overview

- Transformed data from CSV file into an XML file
- “Standings” element overall
- “Team” elements with statistical elements

```
<team>
<name>Chelsea</name>
<record>
    <wins>18</wins>
    <draws>1</draws>
    <losses>3</losses>
</record>
<goals_for>71</goals_for>
<goals_against>18</goals_against>
<goal_differential>53</goal_differential>
<points>55</points>
<points_per_match>2.5</points_per_match>
<expected_goals>57.9</expected_goals>
<expected_goals_against>20.1</expected_goals_against>
<expected_goals_per_90_minutes>1.72</expected_goals_per_90_minutes>
<expected_goal_differential>37.9</expected_goal_differential>
</team>
```

Example “**team**” element with its statistic elements

# Data Analysis

- Created method to evaluate how accurate a statistic is to the actual standings



```
def standingsComp(statStandings):

    standings = ["Chelsea", "Manchester City", "Arsenal",
    statAccuracy = 0

    for i in standings:
        teamInd = standings.index(i)
        team = standings[teamInd]
        for j in statStandings:
            if j["name"] == team:
                statInd = statStandings.index(j)
                break
            teamAccuracy = abs(teamInd - statInd)
            statAccuracy -= teamAccuracy

    statAccuracy /= 12
    statAccuracy = round(statAccuracy, 3)
return(statAccuracy)
```

# Stat Accuracy

All of the stats are given an accuracy score from 0 to -6. The most accurate stats are closer to 0.



# Stat Evaluations

## ACCURACY OF EACH STATISTIC

Goals Against: -0.667

Goals For: -0.5

Goal Differential: -0.5

Expected Goals: -1.167

Expected Goals Against: -1.167

Expected Goals Per 90: -1.0

Expected Goal Differential: -1.0

Expected Goal vs Goal Highest Difference: -2.333

Expected Goal vs Goal Lowest Difference: -5.0

Expected GA vs GA Highest Difference: -5.667

Expected GA vs GA Lowest Difference: -1.5

Expected GD vs GD Highest Difference: -1.5

Expected GD vs GD Lowest Difference: -5.333

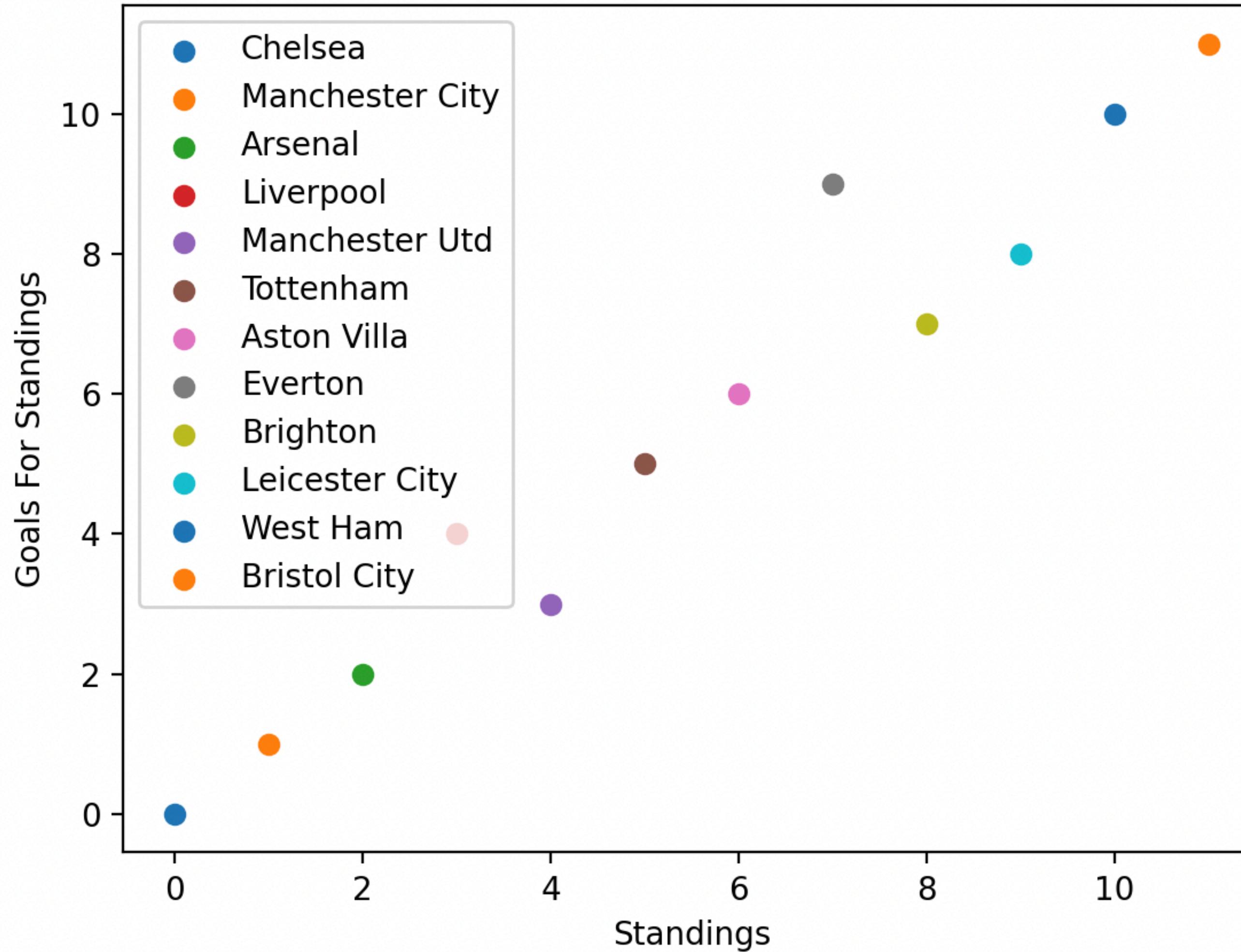
- These stats were taken from the XML file using **XPATH**
- Used a function to create a list of dictionaries that contained the team's name and stat information
- **Sorted** the list items by the value of their stats
- **Compared** this list with the standings to get the statistic's ranking accuracy

# My Algorithm

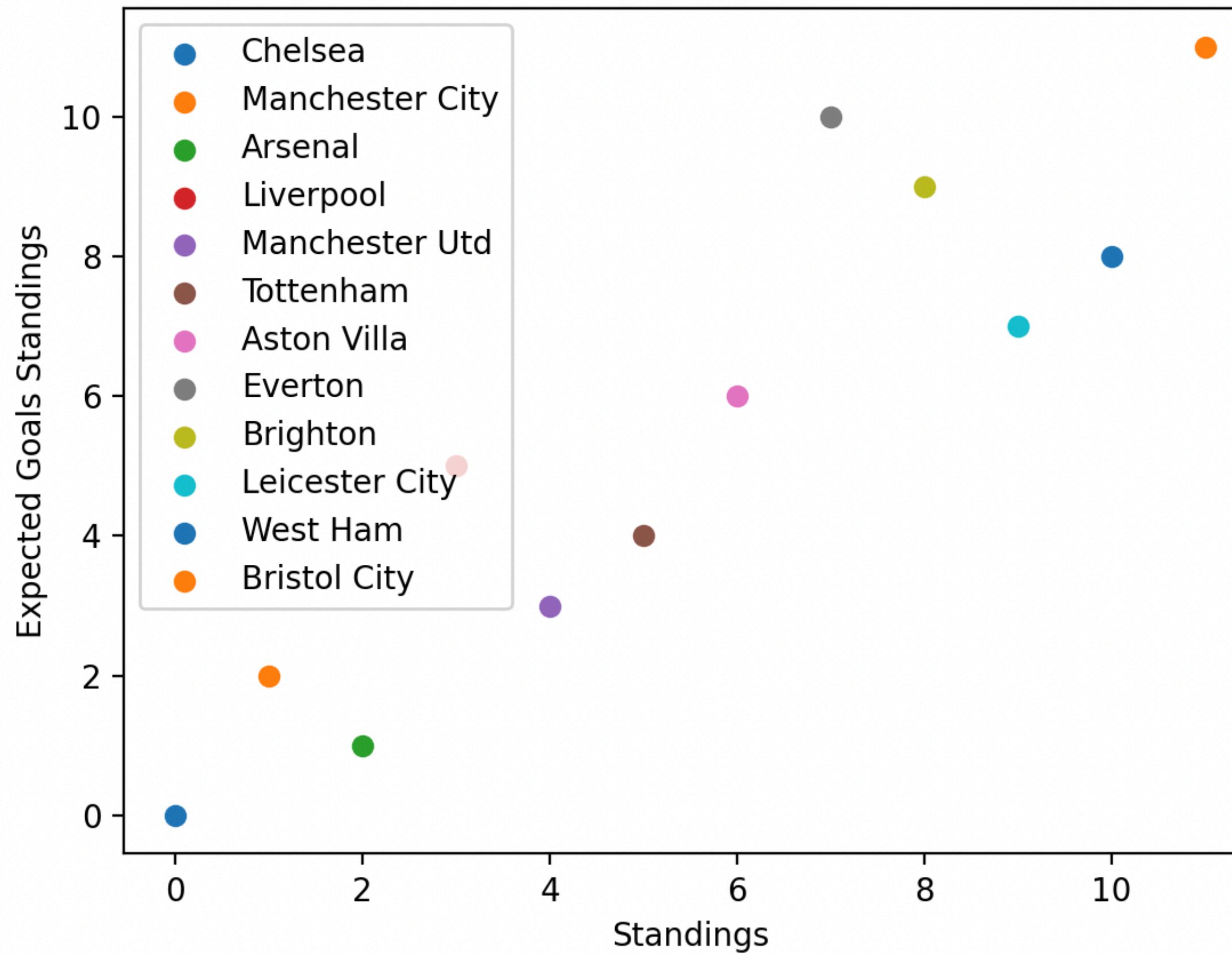
- This awards points based on where a team lands for a particular statistic
- The points ranking takes **every statistic** from the previous slide into account

```
def pointAssignor(statList, pointsDict):  
  
    pointTotalDict = {0:12, 1:11, 2:10, 3:9, 4:8}  
  
    for i in statList:  
        team = i["name"]  
        index = statList.index(i)  
        points = pointTotalDict[index]  
        for j in pointsDict:  
            if j["name"] == team:  
                j["points"] += points  
            break  
  
    return(pointsDict)
```

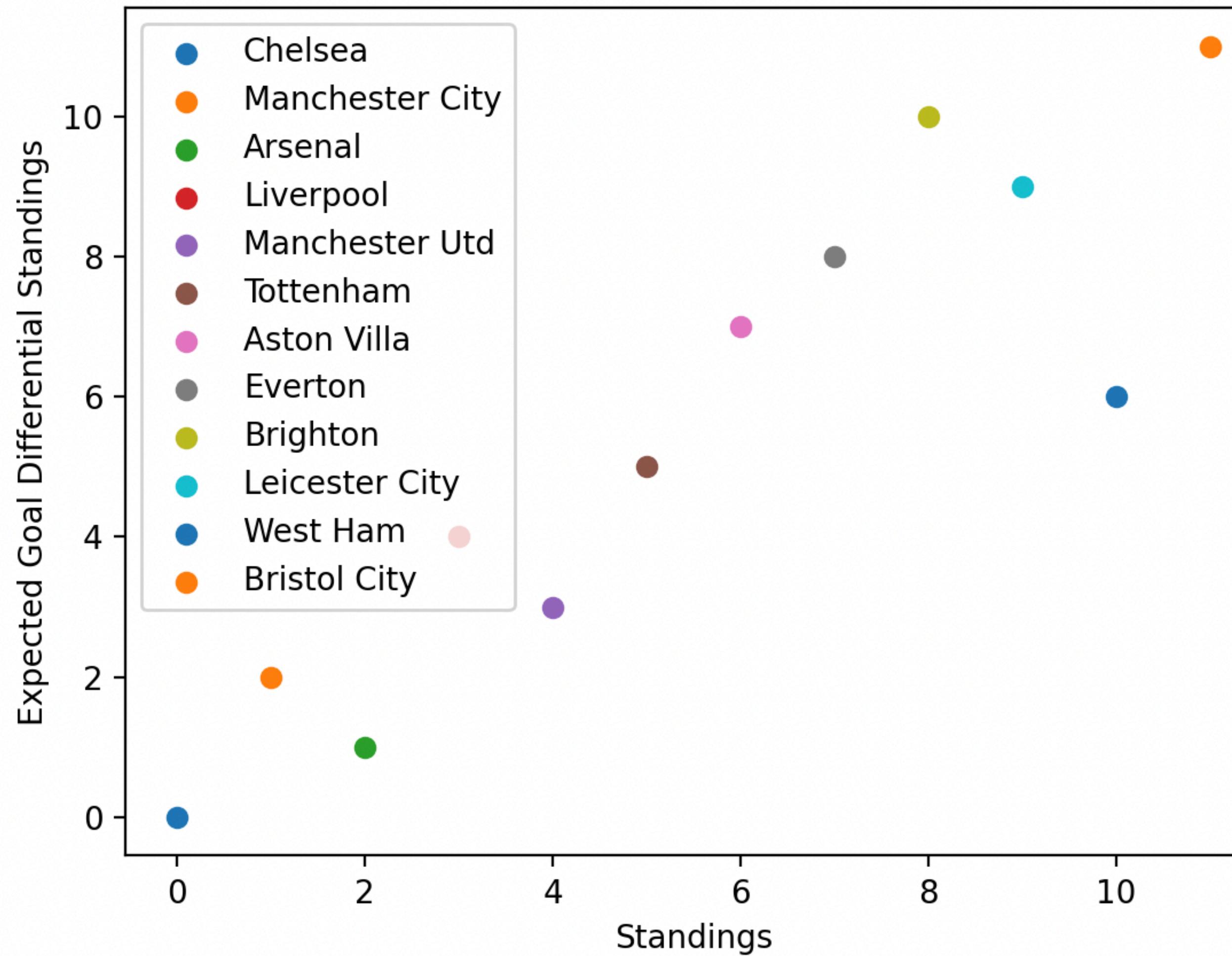
# Standings vs Goals For Standings



## Standings vs Expected Goals Standings



## Standings vs Expected Goal Differential Standings



# Conclusion

- Apply this process to other leagues
- Make my algorithm more accurate by giving certain statistics more weight, etc.

# Thank You!