FINAL REPORT

AUTHORS

- Jake Lawson (jml496)
- Geoff Miller (gom6)
- Evan McGowan (egm68)

Introduction

The hot hand fallacy is the bias to believe that previous success predicts future performance. Coined by researchers at Cornell and Stanford in 1985, there has been continuous analysis of if there is a hot hand in sports and of the cognitive biases experienced in other truly random activities like gambling. Yet on a larger season scale good teams are there any trends in performance? Does performance earlier on in the season of the four major American, professional sports tell us anything about the rest of their season? Do teams that perform well at the beginning of the season choke or gain confidence as the year progresses?

With the recent increase in sports gambling across the country, it is of particular importance to analyze the trends in sports (for both sides of the bet).

Description of Dataset

We acquired our data set from FiveThirtyEight.com a poll, politics, economics, and sports media company focusing on data journalism. The aggregate sports data for all the major sports leagues. We used four of their datasets for each of the major American sports leagues: MLB, NBA, NFL, and NHL. Each of these datasets contains every game that is publicly available; the MLB dataset starts in 1871, the NBA dataset starts in 1946, the NFL dataset starts in 1920, and the NHL dataset starts in 1917. To deal with the changing landscape of the games, we limited the dates of the games to occur between 1990 and 2020.

Next, we filtered and categorized the dataset to include what type of game each record was (playoff, regular season, championship, etc.) and when in the regular season a game occurred.

Hypothesis

Hypothesis 1

A team's difference in performance between the first and second half of the regular season predicts a team's performance in the postseason. If a team performs better in the second half of the regular season, that team will perform better in the postseason, vice versa, if a team

performs worse in the second half of the regular season, that team will perform worse in the postseason (if they make it).

To analyze this we will create a linear model Second Half Performance ~ First Half Performance (without a constant). Is there a significant relationship between the two halves? What is the coefficient? Is it positive, negative, or neutral? Make a model with a constant, i.e. what does not fixing 0% to 0% change the model? Our null hypothesis is B_first = 0.

Hypothesis 2

There is no difference between the four leagues in terms of first half and second half of season performance for all of the teams.

To analyze this create a linear model Second Half Performance ~ First Half Performance for each of the leagues. Is there a difference between the leagues, are these significant? We will use bootstrap hypothesis testing to take multiple bootstrao samples each league to construct hypothesis tests for each pair of leagues. We will have multiple null hypothesis, one for each pair of leagues, eg. B_first_NHL - B_first_NBA = 0.

Analysis of Data

Analysis 1

Using pandas stats module's OLS function, we created a linear model of Second Half Performance ~ First Half Performance, our first hypothesis we wish to test.

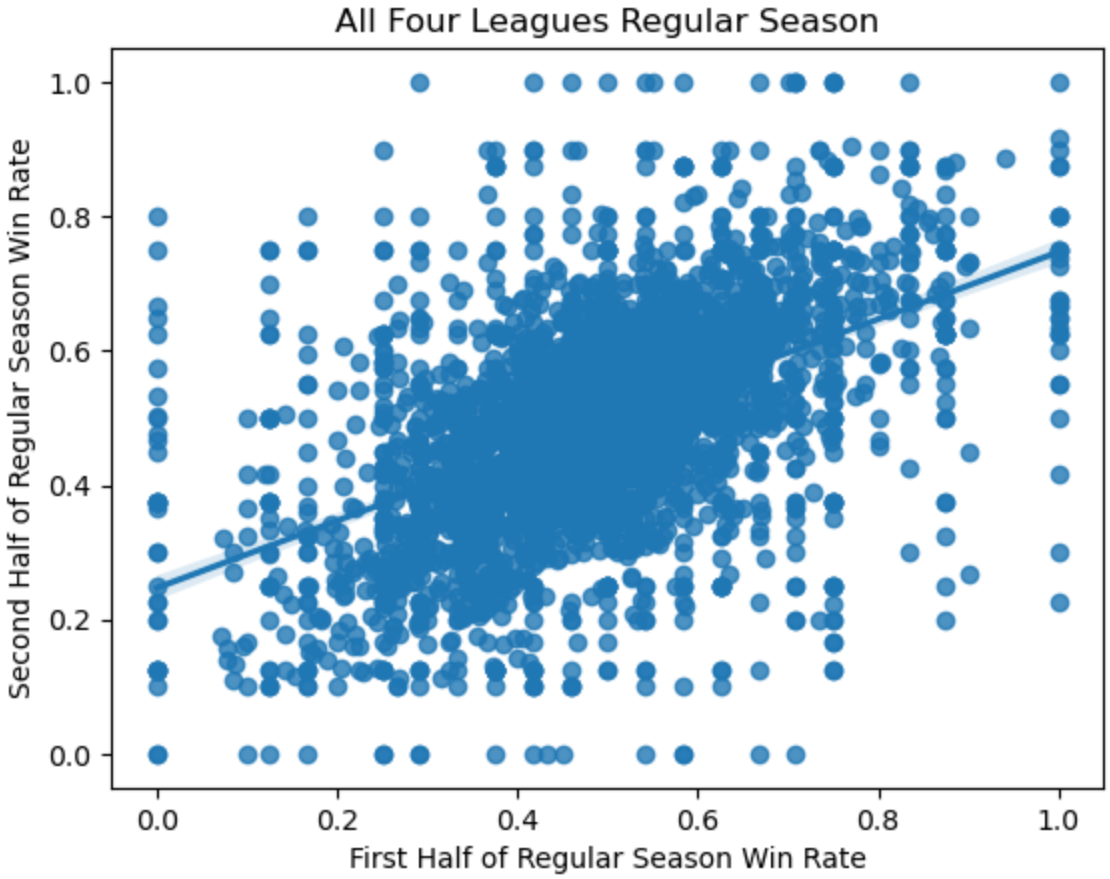
OLS Regression Results

Dep. Variable:	y	R-squared (uncentered):	0.903
Model:	OLS	Adj. R-squared (uncentered):	0.903
Method:	Least Squares	F-statistic:	3.135e+04
Date:	Thu, 16 Nov 2023	Prob (F-statistic):	0.00
Time:	09:00:41	Log-Likelihood:	1356.1
No. Observations:	3357	AIC:	-2710.
Df Residuals:	3356	BIC:	-2704.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.9513	0.005	177.068	0.000	0.941	0.962

Omnibus:	199.617	Durbin-Watson:	1.921
Prob(Omnibus):	0.000	Jarque-Bera (JB):	855.787
Skew:	0.026	Prob(JB):	1.47e-186
Kurtosis:	5.473	Cond. No.	1.00

We also produced a regplot to explore the distribution.



From this we can determine a few things:

1. We can reject the null hypothesis. The p value on the coefficient in the OLS printout is 0. Further exploration is needed to examine the alternative hypothesis.

2. If a team performed poorly in the first half of the season, they will tend to perform better in the second half. We can see this at the far left of the reg plot — if a team wins no games in the first half of the season, our model predicts they will win about 1 in 4 in the second half. Conversely, if a team wins every game in the first half of the season our model predicts they will only win in about 7 of every 10 games in the second half of the season.

3. There are two patterns in the data. One are these discrete values we can see some of the points taking while the other (more an absence of a pattern) is the typical heteroskedastic/independent cloud in the middle. More investigation is needed into this.

Analysis 2

Next, we explore each league individually.

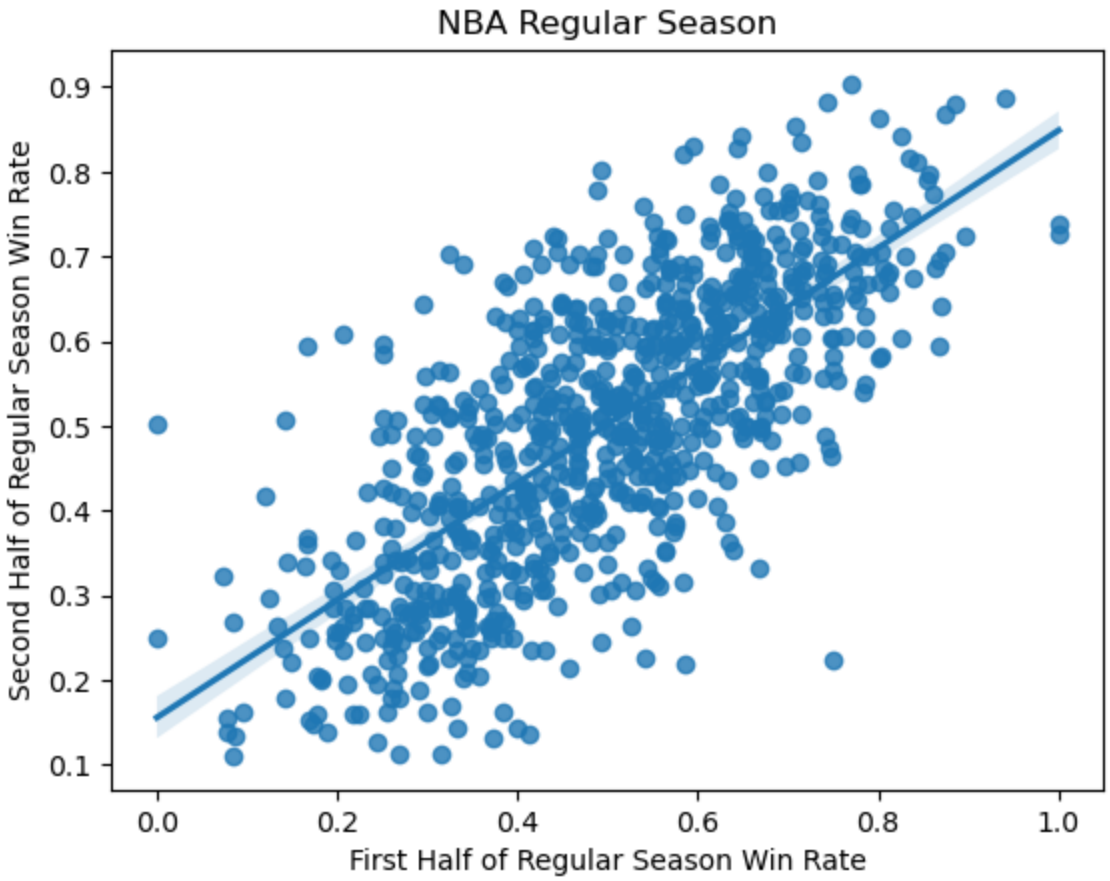
THE NBA

OLS Regression Results

Dep. Variable:	y	R-squared (uncentered):	0.942
Model:	OLS	Adj. R-squared (uncentered):	0.942
Method:	Least Squares	F-statistic:	1.327e+04
Date:	Thu, 16 Nov 2023	Prob (F-statistic):	0.00
Time:	09:01:29	Log-Likelihood:	525.24
No. Observations:	812	AIC:	-1048.
Df Residuals:	811	BIC:	-1044.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.9717	0.008	115.205	0.000	0.955	0.988

Omnibus:	9.962	Durbin-Watson:	1.901
Prob(Omnibus):	0.007	Jarque-Bera (JB):	12.122
Skew:	0.163	Prob(JB):	0.00233
Kurtosis:	3.502	Cond. No.	1.00


NBA Regular Season
Second Half of Regular Season Win Rate
First Half of Regular Season Win Rate
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.0
0.2
0.4
0.6
0.8
1.0

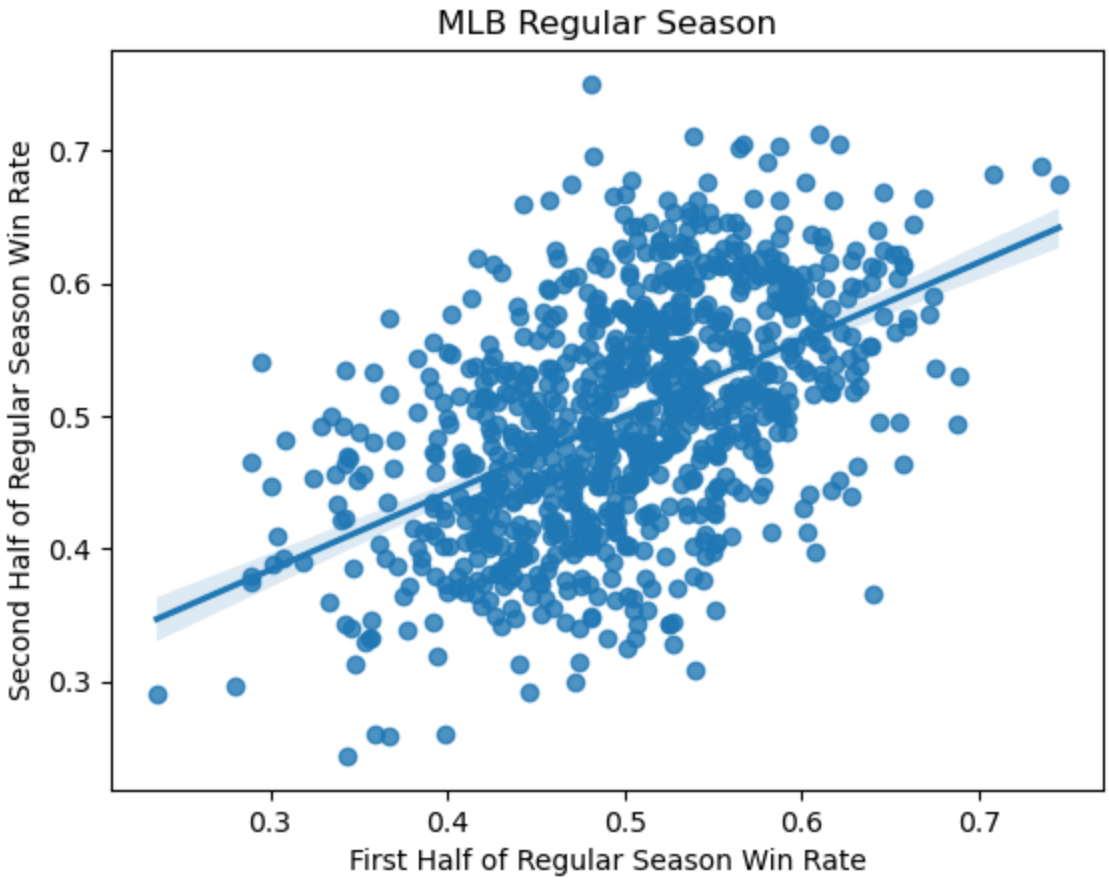
THE MLB

OLS Regression Results

Dep. Variable:	y	R-squared (uncentered):	0.974
Model:	OLS	Adj. R-squared (uncentered):	0.974
Method:	Least Squares	F-statistic:	3.294e+04
Date:	Thu, 16 Nov 2023	Prob (F-statistic):	0.00
Time:	09:01:49	Log-Likelihood:	953.35
No. Observations:	878	AIC:	-1905.
Df Residuals:	877	BIC:	-1900.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.9900	0.005	181.493	0.000	0.979	1.001

Omnibus:	0.472	Durbin-Watson:	1.945
Prob(Omnibus):	0.790	Jarque-Bera (JB):	0.536
Skew:	0.051	Prob(JB):	0.765
Kurtosis:	2.936	Cond. No.	1.00


MLB Regular Season
Second Half of Regular Season Win Rate
First Half of Regular Season Win Rate
0.3
0.4
0.5
0.6
0.7

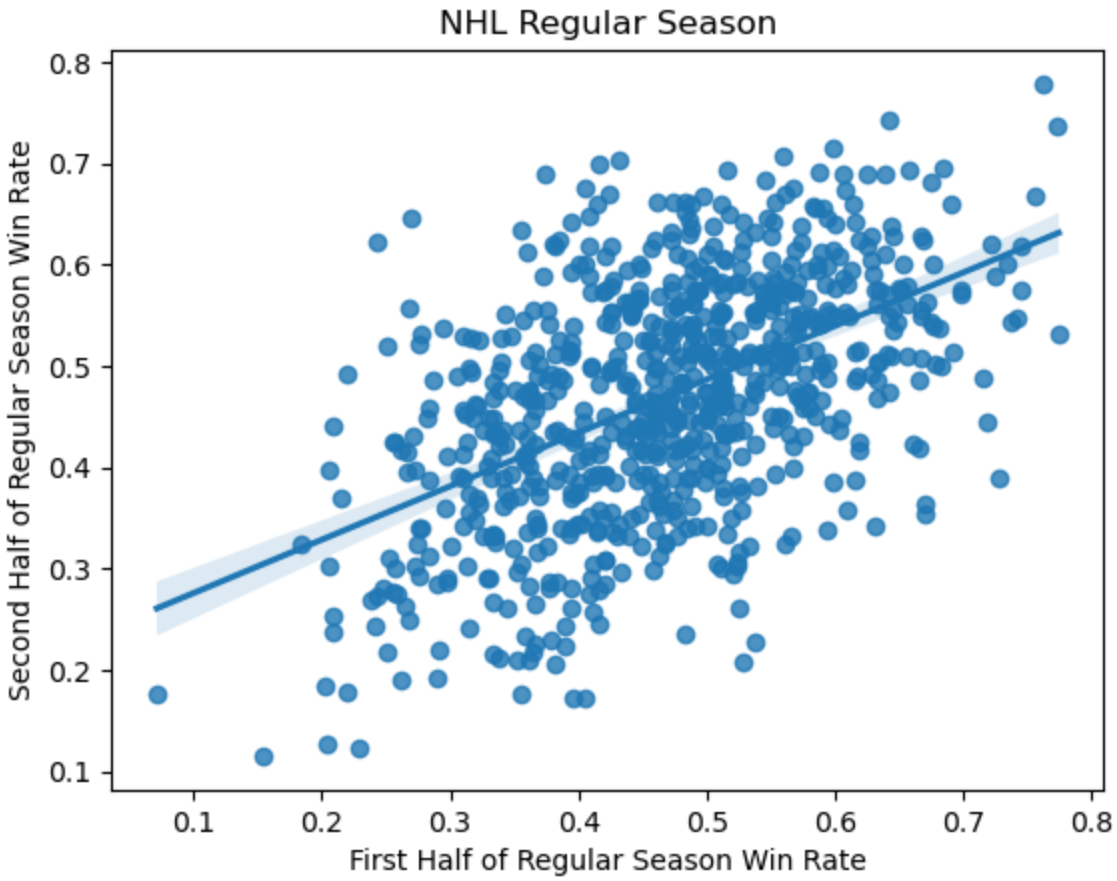
THE NHL

OLS Regression Results

Dep. Variable:	y	R-squared (uncentered):	0.945
Model:	OLS	Adj. R-squared (uncentered):	0.945
Method:	Least Squares	F-statistic:	1.256e+04
Date:	Thu, 16 Nov 2023	Prob (F-statistic):	0.00
Time:	09:03:09	Log-Likelihood:	554.36
No. Observations:	738	AIC:	-1107.
Df Residuals:	737	BIC:	-1102.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.9765	0.009	112.082	0.000	0.959	0.994

Omnibus:	0.638	Durbin-Watson:	1.933
Prob(Omnibus):	0.727	Jarque-Bera (JB):	0.564
Skew:	0.066	Prob(JB):	0.754
Kurtosis:	3.033	Cond. No.	1.00


NHL Regular Season
Second Half of Regular Season Win Rate
First Half of Regular Season Win Rate
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

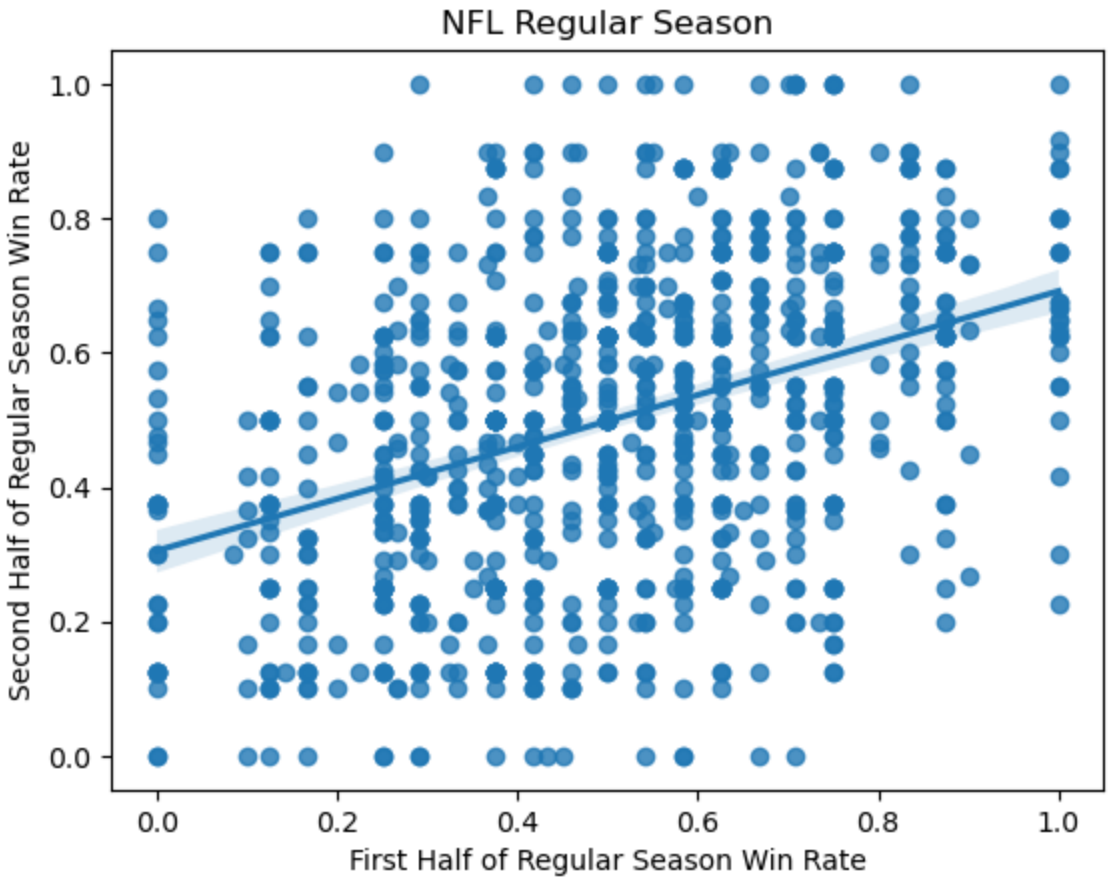
THE NFL

OLS Regression Results

Dep. Variable:	y	R-squared (uncentered):	0.795
Model:	OLS	Adj. R-squared (uncentered):	0.795
Method:	Least Squares	F-statistic:	3598.
Date:	Thu, 16 Nov 2023	Prob (F-statistic):	1.43e-321
Time:	09:03:31	Log-Likelihood:	-25.464
No. Observations:	929	AIC:	52.93
Df Residuals:	928	BIC:	57.76
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.8888	0.015	59.984	0.000	0.860	0.918

Omnibus:	0.029	Durbin-Watson:	1.884
Prob(Omnibus):	0.986	Jarque-Bera (JB):	0.066
Skew:	0.010	Prob(JB):	0.968
Kurtosis:	2.964	Cond. No.	1.00



Now we can see the patterns emerging. Because of the NFL's smaller playing schedule, there are more discrete percentages — the other three leagues also have fixed win rates but because they have more games it appears that they can take any value.

The NFL also has the weakest relationship of the four leagues, although all four are significant with a p-value 0.

Now we want to explore these differences. As we laid out before we took bootstrap samples of the four leagues and took the differences of the mean win rates of these samples. Using this we can create a bootstrap estimate of a hypothesis test. These are the results:

```
Significance tests
sig -------test--------    pvalue
    First Half NBA MLB -> 0.497
    Second Half NBA MLB -> 0.5
    First Half NBA NHL -> 0.505
    Second Half NBA NHL -> 0.472
    First Half NBA NFL -> 0.486
    Second Half NBA NFL -> 0.518
    First Half MLB NHL -> 0.507
    Second Half MLB NHL -> 0.503
    First Half MLB NFL -> 0.516
    Second Half MLB NFL -> 0.511
    First Half NHL NFL -> 0.506
    Second Half NHL NFL -> 0.469
```

From this, we accept a null hypothesis: there is no difference between the leagues in terms of win rates during the first or second halves of the seasons.

Even though the NFL appears to be different, indeed the coefficient is the most different out of all the leagues, the difference is not significant.

Evaluation of Significance

To reiterate, we reject the null hypothesis for our first hypothesis; it is false to say that there is no relationship between the second half of a teams performance in a season and their performance in the first half of the season.

We accept the null hypothesis in our second hypothesis: there is no significant difference between any of the leagues in what we measured.

Conclusion

Even though we found no significant difference in the four leagues in terms of performance, we did find a significant relationship. The relationship between Second and First half of season

performance is positive but according to the model (using all four leagues as input), for every 1% increase in win rate in the first half of the season, we predict that the team will only increase their second half of season win rate by 0.9513%. Now this might be because of the bounds our data takes: you cannot have a win rate above 100% nor below 0%. As a result the LSRL might be rotated slightly clockwise. This is an interesting quirk in the model that elicits further examination.

Looking back to our original question of choking versus excelling in the second half of the season we do see some *observable* differences between the two leagues. When observing the NFL graph, people performing relatively well, winning 75% of their games in the early season, can fall hard. There are many occurrences of these teams winning 0 games in the late season. On the other hand, terrible teams can find their stride in the late season occasionally. Now this might be due to the limited number of games the NFL plays compared to the other three leagues: every game matters when you play so few and teams can get into a rut. But these patterns do not appear to exist in the other leagues (which again might be due to how many games they play).

Looking at the R^2 statistics, they can be very high. The NBA, MLB, and NHL are in the mid to high 0.9s with the MLB even hitting 0.974. This leads us to believe there might be a high degree of collinearity in the data for some of these leagues.

The most interesting result we found is how bad teams, on average, perform better in the late season while good teams perform worse. This might be enough to give you a slight edge at the OTB. But when you are the worst in the league the only way is up and when you are the best it is a long way down so we would not bet on it.

Apendix 1: Final Analysis

In [1]:
```
import pandas as pd
from statsmodels.regression.linear_model import OLS
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
mlb = pd.read_csv('data/cleaned/mlb.csv')
nfl = pd.read_csv('data/cleaned/nfl.csv')
nhl = pd.read_csv('data/cleaned/nhl.csv')
nba = pd.read_csv('data/cleaned/nba.csv')
```

Hypothesis 1

In [3]:
```
mlb
```

Out[3]:

	Unnamed: 0	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_
0	8316	2019-10-30	2019	0	w	HOU	WSN	1599.542804	1584.363378	0.5
1	8317	2019-10-29	2019	0	w	HOU	WSN	1605.069000	1578.837182	0.5

	Unnamed: 0	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_
2	8318	2019-10-27	2019	0	w	WSN	HOU	1584.005206	1599.900976	0.5
3	8319	2019-10-26	2019	0	w	WSN	HOU	1589.985555	1593.920627	0.5
4	8320	2019-10-25	2019	0	w	WSN	HOU	1593.827376	1590.078806	0.5
...	...	...	...	...	...	...	...	...	...	
71083	79399	1990-04-09	1990	0	r1	KCR	BAL	1520.125000	1500.548000	0.5
71084	79400	1990-04-09	1990	0	r1	HOU	CIN	1504.135000	1492.159000	0.5
71085	79401	1990-04-09	1990	0	r1	CHW	MIL	1491.470000	1516.772000	0.4
71086	79402	1990-04-09	1990	0	r1	BOS	DET	1519.201000	1462.610000	0.6
71087	79403	1990-04-09	1990	0	r1	ANA	SEA	1515.923000	1490.516000	0.5

71088 rows × 29 columns

In [4]:
```
mlb_first_half = mlb[mlb.playoff == 'r1'][['playoff','team1','team2','win1', 'wi
mlb_second_half = mlb[mlb.playoff == 'r2'][['playoff','team1','team2','win1', 'w

team1_first = mlb_first_half[['team1','win1', 'win2', 'season']].groupby(by=['te

team2_first = mlb_first_half[['team2','win1', 'win2', 'season']].groupby(by=['te

team1_second = mlb_second_half[['team1','win1', 'win2', 'season']].groupby(by=['

team2_second = mlb_second_half[['team2','win1', 'win2', 'season']].groupby(by=['

mlb_first_half = []
mlb_second_half = []

for entry in team2_second.index:
  mlb_first_half.append(
    (team1_first.loc[entry][0] + team2_first.loc[entry][0])/2)
  mlb_second_half.append(
    (team1_second.loc[entry][0] + team2_second.loc[entry][0])/2)
```

In [5]:
```
nfl
```

Out[5]:

	Unnamed: 0	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_
0	8772	1990-	1989	0	d	SF	MIN	1741.063000	1595.514000	0.77

	Unnamed: 0	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_
		01-06								
1	8773	1990-01-06	1989	0	d	CLE	BUF	1565.876000	1576.806000	0.5
2	8774	1990-01-07	1989	0	d	DEN	PIT	1599.653000	1556.821000	0.65
3	8775	1990-01-07	1989	0	d	NYG	LAR	1623.766000	1643.646000	0.56
4	8776	1990-01-14	1989	0	c	SF	LAR	1755.160000	1666.078000	0.70
...	...	...	...	...	...	...	...	...	...	
7753	16525	2019-12-29	2019	0	r2	HOU	TEN	1584.215388	1542.863113	0.64
7754	16526	2019-12-29	2019	0	r2	DEN	OAK	1487.042014	1407.432240	0.69
7755	16527	2019-12-29	2019	0	r2	DAL	WSH	1536.023522	1309.885760	0.84
7756	16528	2019-12-29	2019	0	r2	NYG	PHI	1351.254830	1551.327497	0.31
7757	16529	2019-12-29	2019	0	r2	SEA	SF	1570.662276	1609.709252	0.53

7758 rows × 36 columns

In [6]:
```
nfl_first_half = nfl[nfl.playoff == 'r1'][['playoff','team1','team2','win1', 'wi
nfl_second_half = nfl[nfl.playoff == 'r2'][['playoff','team1','team2','win1', 'w

team1_first = nfl_first_half[['team1','win1', 'win2', 'season']].groupby(by=['te

team2_first = nfl_first_half[['team2','win1', 'win2', 'season']].groupby(by=['te

team1_second = nfl_second_half[['team1','win1', 'win2', 'season']].groupby(by=['

team2_second = nfl_second_half[['team2','win1', 'win2', 'season']].groupby(by=['
```

In [7]:
```
intersection = team2_second.index.intersection(team1_second.index)\
  .intersection(team2_first.index)\
  .intersection(team1_first.index)
```

In [8]:
```
nfl_first_half = []
nfl_second_half = []

for entry in intersection:
  nfl_first_half.append(
    (team1_first.loc[entry][0] + team2_first.loc[entry][0])/2)
  nfl_second_half.append(
    (team1_second.loc[entry][0] + team2_second.loc[entry][0])/2)
```

In [9]:
```
nba
```

Out[9]:

	Unnamed: 0	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_
0	31335	1990-01-02	1990	0	r2	NYK	PHO	1620.015700	1565.457600	0.7
1	31336	1990-01-02	1990	0	r2	WSB	NJN	1466.478800	1325.255500	0.8
2	31337	1990-01-02	1990	0	r2	POR	MIA	1527.168300	1263.916600	0.8
3	31338	1990-01-02	1990	0	r2	ATL	MIL	1545.971100	1527.273700	0.6
4	31339	1990-01-02	1990	0	r2	DAL	IND	1457.774900	1528.762500	0.5
...	...	...	...	...	...	...	...	...	...	
37569	68904	2019-12-31	2020	0	r1	SAC	LAC	1439.142942	1631.787446	0.3
37570	68905	2019-12-31	2020	0	r1	HOU	DEN	1629.924907	1618.388168	0.6
37571	68906	2019-12-31	2020	0	r1	TOR	CLE	1642.844707	1341.079149	0.9
37572	68907	2019-12-31	2020	0	r1	SAS	GSW	1484.142262	1412.580576	0.
37573	68908	2019-12-31	2020	0	r1	OKC	DAL	1550.631521	1605.977237	0.5

37574 rows × 30 columns

In [10]:
```
nba_first_half = nba[nba.playoff == 'r1'][['playoff','team1','team2','win1', 'wi
nba_second_half = nba[nba.playoff == 'r2'][['playoff','team1','team2','win1', 'w

team1_first = nba_first_half[['team1','win1', 'win2', 'season']].groupby(by=['te

team2_first = nba_first_half[['team2','win1', 'win2', 'season']].groupby(by=['te

team1_second = nba_second_half[['team1','win1', 'win2', 'season']].groupby(by=['

team2_second = nba_second_half[['team2','win1', 'win2', 'season']].groupby(by=['
```

In [11]:
```
intersection = team2_second.index.intersection(team1_second.index)\
  .intersection(team2_first.index)\
  .intersection(team1_first.index)
```

In [12]:
```
nba_first_half = []
nba_second_half = []

for entry in intersection:
  nba_first_half.append(
    (team1_first.loc[entry][0] + team2_first.loc[entry][0])/2)
  nba_second_half.append(
    (team1_second.loc[entry][0] + team2_second.loc[entry][0])/2)
```

In [13]:
```
nhl
```

Out[13]:

	Unnamed: 0	season	date	playoff	neutral	status	ot	home_team	away_team	home_t
0	26609	1990	1990-01-01	r2	0	post	NaN	Washington Capitals	Los Angeles Kings	
1	26610	1990	1990-01-02	r2	0	post	NaN	St. Louis Blues	Edmonton Oilers	
2	26611	1990	1990-01-02	r2	0	post	OT	Calgary Flames	Philadelphia Flyers	
3	26612	1990	1990-01-02	r2	0	post	NaN	New Jersey Devils	Buffalo Sabres	
4	26613	1990	1990-01-02	r2	0	post	NaN	New York Islanders	Los Angeles Kings	
...	...	...	...	...	...	...	...	...	...	
35164	61773	2020	2019-12-31	r1	0	post	NaN	Calgary Flames	Chicago Blackhawks	
35165	61774	2020	2019-12-31	r1	0	post	NaN	Carolina Hurricanes	Montreal Canadiens	
35166	61775	2020	2019-12-31	r1	0	post	NaN	Buffalo Sabres	Tampa Bay Lightning	
35167	61776	2020	2019-12-31	r1	0	post	NaN	Arizona Coyotes	St. Louis Blues	
35168	61777	2020	2019-12-31	r1	0	post	NaN	Columbus Blue Jackets	Florida Panthers	

35169 rows × 27 columns

In [14]:
```
nhl_first_half = nhl[nhl.playoff == 'r1'][['playoff','home_team','away_team','wi
nhl_second_half = nhl[nhl.playoff == 'r2'][['playoff','home_team','away_team','w

team1_first = nhl_first_half[['home_team','win1', 'win2', 'season']].groupby(by=

team2_first = nhl_first_half[['away_team','win1', 'win2', 'season']].groupby(by=

team1_second = nhl_second_half[['home_team','win1', 'win2', 'season']].groupby(b

team2_second = nhl_second_half[['away_team','win1', 'win2', 'season']].groupby(b
```

In [15]:
```
intersection = team2_second.index.intersection(team1_second.index)\
  .intersection(team2_first.index)\
  .intersection(team1_first.index)
```

In [16]:
```
nhl_first_half = []
nhl_second_half = []

for entry in intersection:
  nhl_first_half.append(
```

```
    (team1_first.loc[entry][0] + team2_first.loc[entry][0])/2)
  nhl_second_half.append(
    (team1_second.loc[entry][0] + team2_second.loc[entry][0])/2)
```

In [25]:
```
second_half = mlb_second_half + nfl_second_half + nba_second_half + nhl_second_h
first_half = mlb_first_half + nfl_first_half + nba_first_half + nhl_first_half
```

In [26]:
```
sns.regplot(y=second_half, x=first_half)
plt.xlabel('First Half of Regular Season Win Rate')
plt.ylabel('Second Half of Regular Season Win Rate')
plt.title('All Four Leagues Regular Season')
plt.plot()

model = OLS(endog=second_half, exog=first_half, hasconst=False)
results = model.fit()
results.summary()
```

Out[26]:

OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	0.903
Model:	OLS	**Adj. R-squared (uncentered):**	0.903
Method:	Least Squares	**F-statistic:**	3.135e+04
Date:	Thu, 16 Nov 2023	**Prob (F-statistic):**	0.00
Time:	09:00:41	**Log-Likelihood:**	1356.1
No. Observations:	3357	**AIC:**	-2710.
Df Residuals:	3356	**BIC:**	-2704.
Df Model:	1		
Covariance Type:	nonrobust		

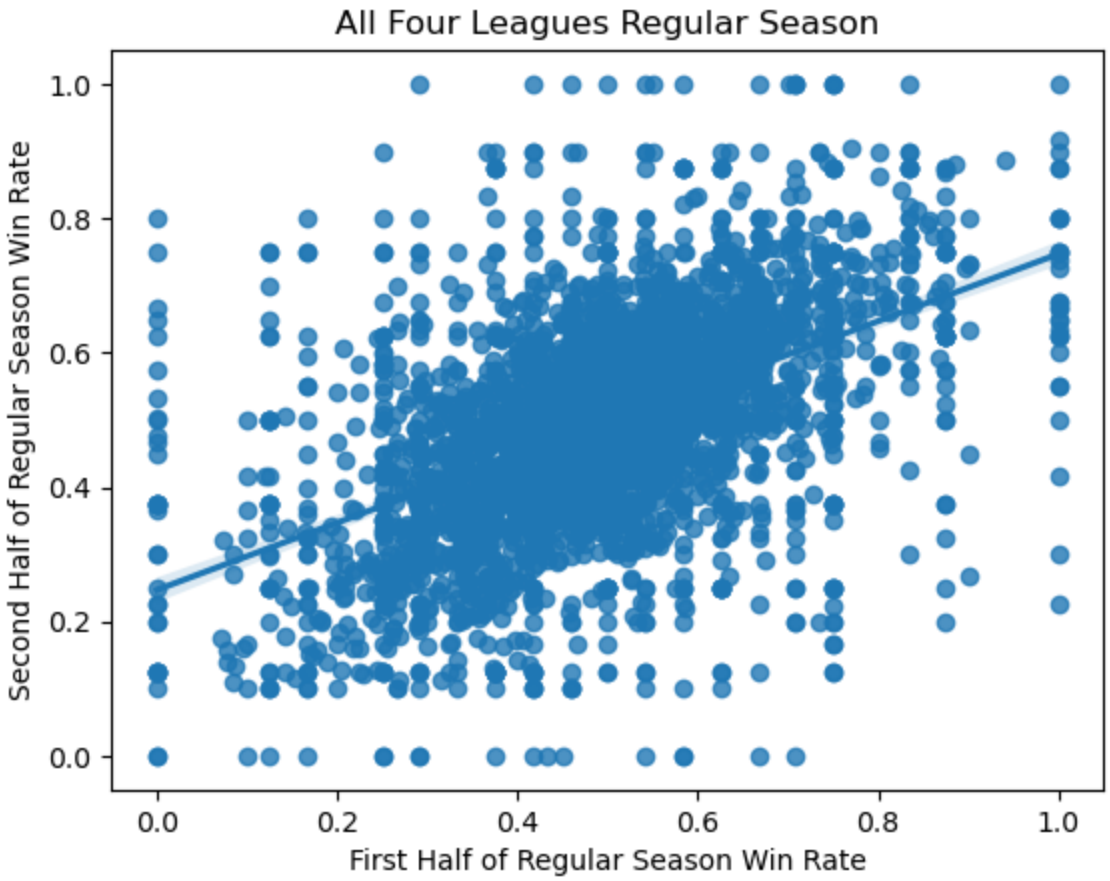
	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.9513	0.005	177.068	0.000	0.941	0.962

Omnibus:	199.617	**Durbin-Watson:**	1.921
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	855.787
Skew:	0.026	**Prob(JB):**	1.47e-186
Kurtosis:	5.473	**Cond. No.**	1.00

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.



In [27]:
```
sns.regplot(y=nba_second_half, x=nba_first_half)
plt.xlabel('First Half of Regular Season Win Rate')
plt.ylabel('Second Half of Regular Season Win Rate')
plt.title('NBA Regular Season')
plt.plot()

model = OLS(endog=nba_second_half, exog=nba_first_half, hasconst=False)
results = model.fit()
results.summary()
```

Out[27]:

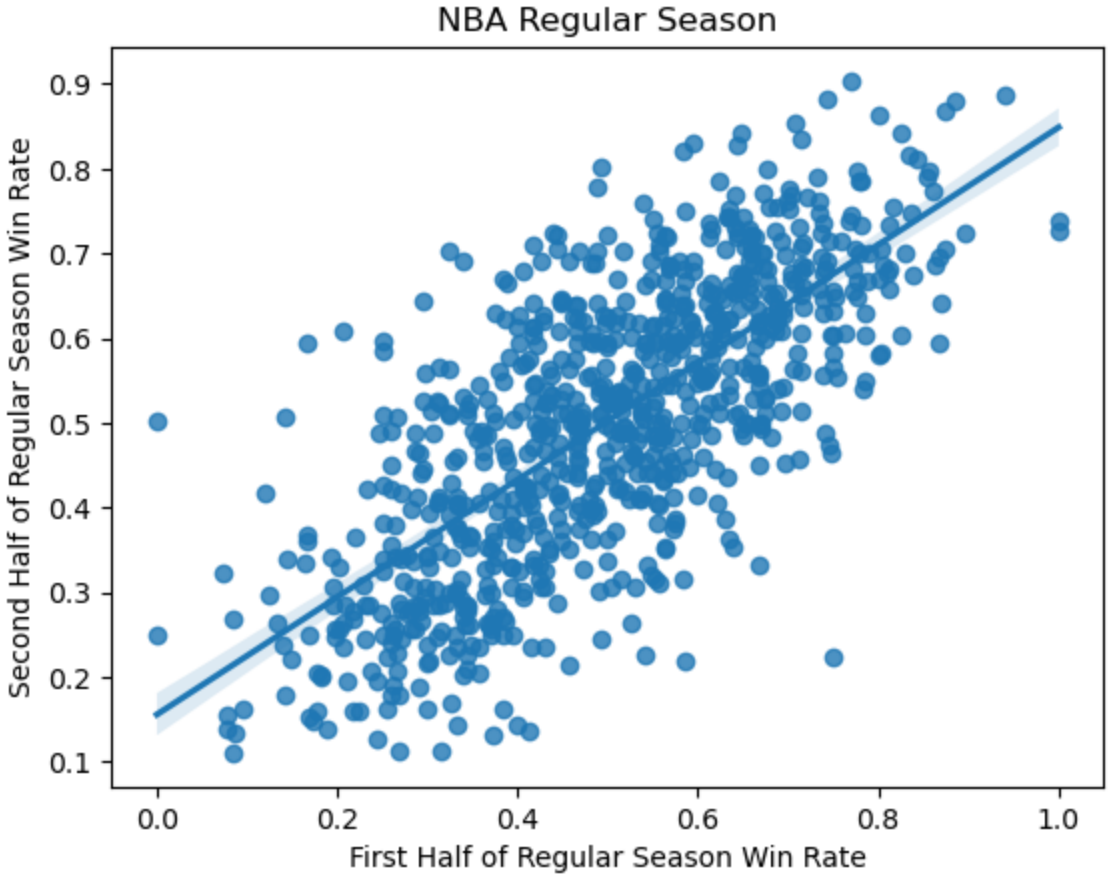
OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	0.942
Model:	OLS	**Adj. R-squared (uncentered):**	0.942
Method:	Least Squares	**F-statistic:**	1.327e+04
Date:	Thu, 16 Nov 2023	**Prob (F-statistic):**	0.00
Time:	09:01:29	**Log-Likelihood:**	525.24
No. Observations:	812	**AIC:**	-1048.
Df Residuals:	811	**BIC:**	-1044.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.9717	0.008	115.205	0.000	0.955	0.988

Omnibus:	9.962	**Durbin-Watson:**	1.901
Prob(Omnibus):	0.007	**Jarque-Bera (JB):**	12.122
Skew:	0.163	**Prob(JB):**	0.00233
Kurtosis:	3.502	**Cond. No.**	1.00

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.



In [28]:
```
sns.regplot(y=mlb_second_half, x=mlb_first_half)
plt.xlabel('First Half of Regular Season Win Rate')
plt.ylabel('Second Half of Regular Season Win Rate')
plt.title('MLB Regular Season')
plt.plot()

model = OLS(endog=mlb_second_half, exog=mlb_first_half, hasconst=False)
results = model.fit()
results.summary()
```

Out[28]:

OLS Regression Results

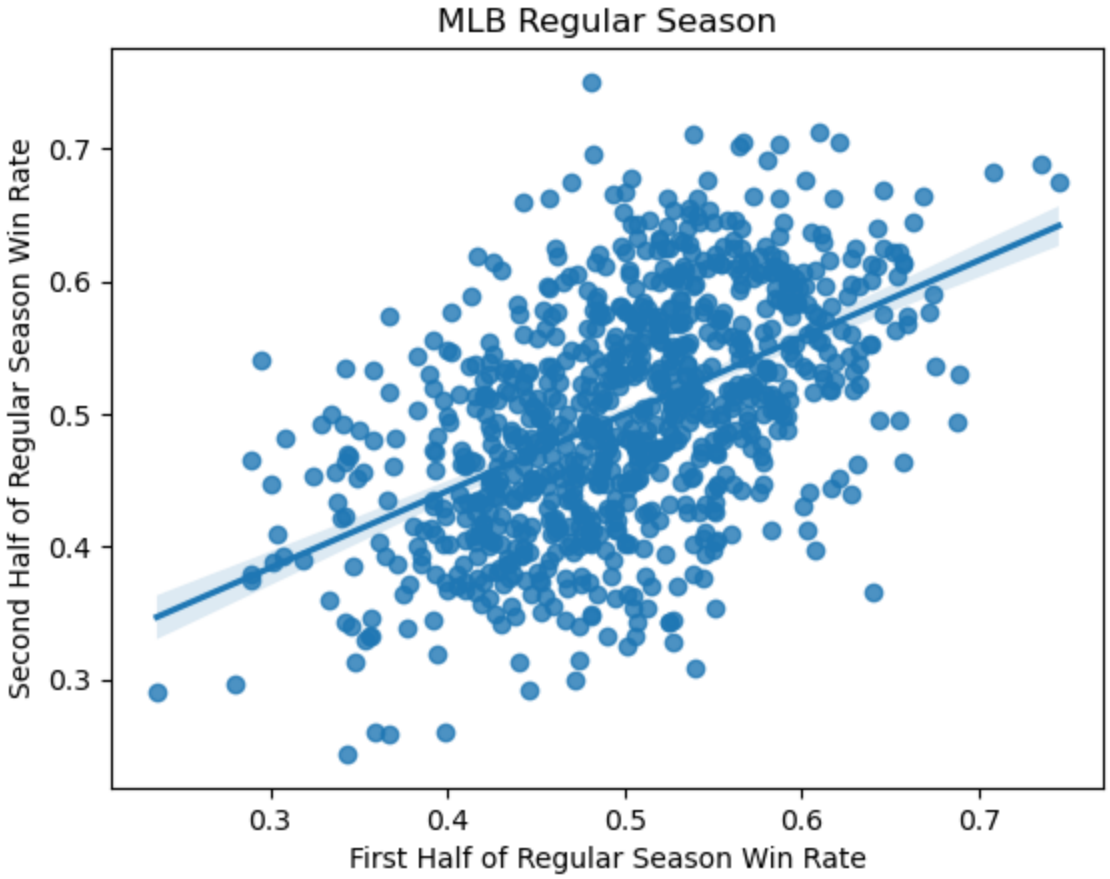
Dep. Variable:	y	**R-squared (uncentered):**	0.974
Model:	OLS	**Adj. R-squared (uncentered):**	0.974
Method:	Least Squares	**F-statistic:**	3.294e+04

Date:	Thu, 16 Nov 2023	**Prob (F-statistic):**	0.00
Time:	09:01:49	**Log-Likelihood:**	953.35
No. Observations:	878	**AIC:**	-1905.
Df Residuals:	877	**BIC:**	-1900.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.9900	0.005	181.493	0.000	0.979	1.001

Omnibus:	0.472	**Durbin-Watson:**	1.945
Prob(Omnibus):	0.790	**Jarque-Bera (JB):**	0.536
Skew:	0.051	**Prob(JB):**	0.765
Kurtosis:	2.936	**Cond. No.**	1.00

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.



In [29]:
```
sns.regplot(y=nhl_second_half, x=nhl_first_half)
plt.xlabel('First Half of Regular Season Win Rate')
```

```
plt.ylabel('Second Half of Regular Season Win Rate')
plt.title('NHL Regular Season')
plt.plot()

model = OLS(endog=nhl_second_half, exog=nhl_first_half, hasconst=False)
results = model.fit()
results.summary()
```

Out[29]:

OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	0.945
Model:	OLS	**Adj. R-squared (uncentered):**	0.945
Method:	Least Squares	**F-statistic:**	1.256e+04
Date:	Thu, 16 Nov 2023	**Prob (F-statistic):**	0.00
Time:	09:03:09	**Log-Likelihood:**	554.36
No. Observations:	738	**AIC:**	-1107.
Df Residuals:	737	**BIC:**	-1102.
Df Model:	1		
Covariance Type:	nonrobust		

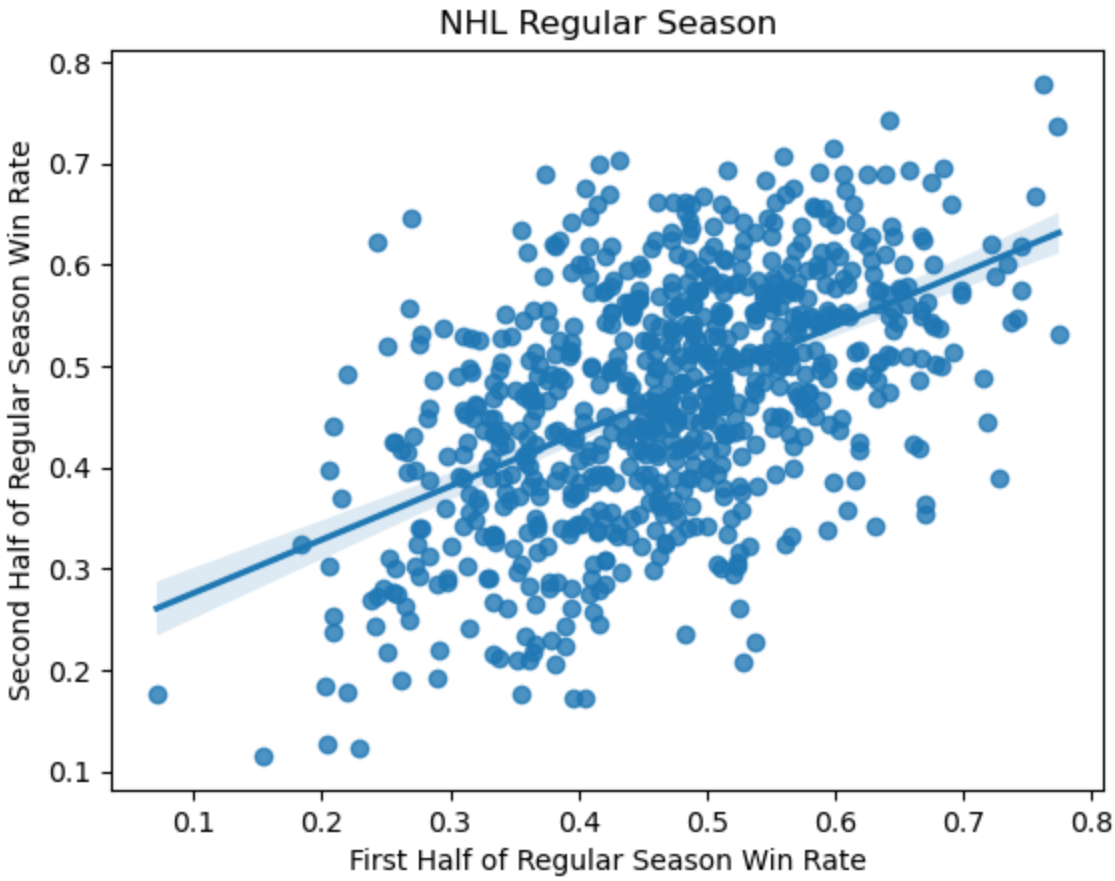
	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.9765	0.009	112.082	0.000	0.959	0.994

Omnibus:	0.638	**Durbin-Watson:**	1.933
Prob(Omnibus):	0.727	**Jarque-Bera (JB):**	0.564
Skew:	0.066	**Prob(JB):**	0.754
Kurtosis:	3.033	**Cond. No.**	1.00

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.



In [30]:
```
sns.regplot(y=nfl_second_half, x=nfl_first_half)
plt.xlabel('First Half of Regular Season Win Rate')
plt.ylabel('Second Half of Regular Season Win Rate')
plt.title('NFL Regular Season')
plt.plot()

model = OLS(endog=nfl_second_half, exog=nfl_first_half, hasconst=False)
results = model.fit()
results.summary()
```

Out[30]:

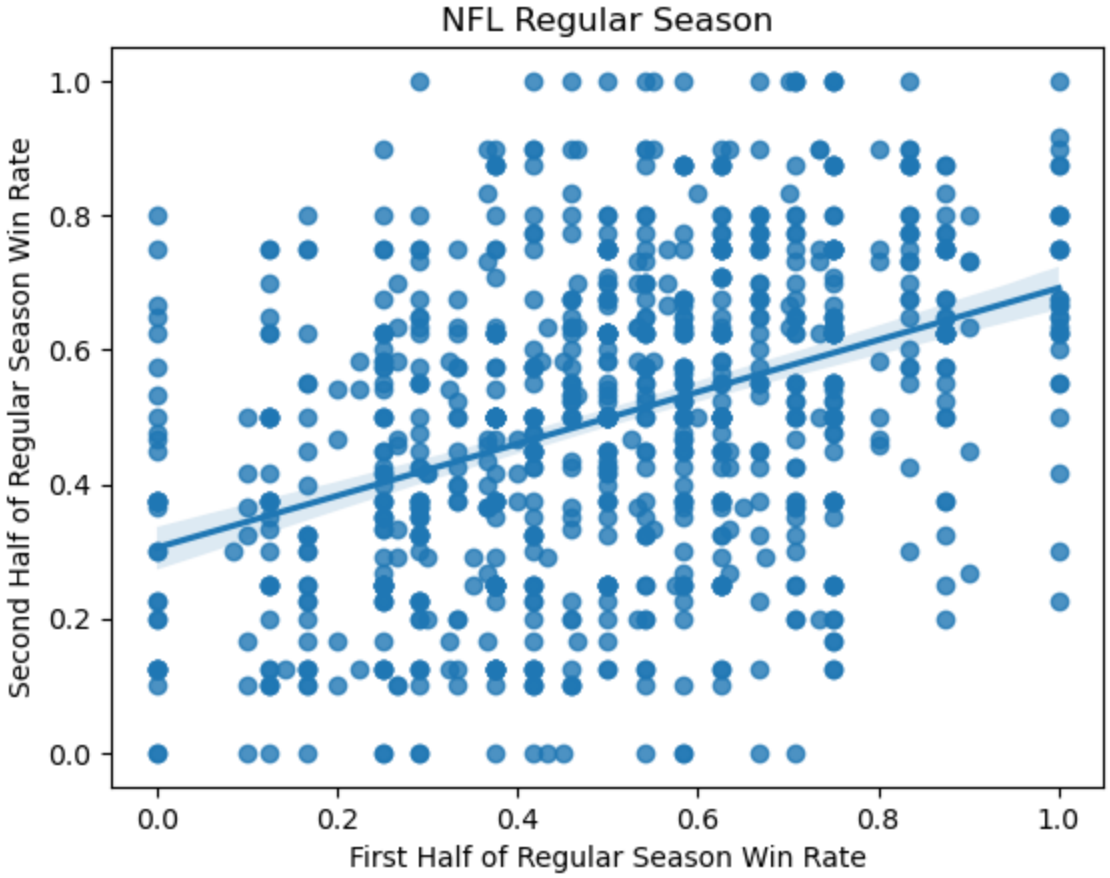
OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	0.795
Model:	OLS	**Adj. R-squared (uncentered):**	0.795
Method:	Least Squares	**F-statistic:**	3598.
Date:	Thu, 16 Nov 2023	**Prob (F-statistic):**	1.43e-321
Time:	09:03:31	**Log-Likelihood:**	-25.464
No. Observations:	929	**AIC:**	52.93
Df Residuals:	928	**BIC:**	57.76
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.8888	0.015	59.984	0.000	0.860	0.918

Omnibus:	0.029	**Durbin-Watson:**	1.884
Prob(Omnibus):	0.986	**Jarque-Bera (JB):**	0.066
Skew:	0.010	**Prob(JB):**	0.968
Kurtosis:	2.964	**Cond. No.**	1.00

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.



In [34]:
```
nba_first_half = np.array(nba_first_half)
nba_second_half = np.array(nba_second_half)

mlb_first_half = np.array(mlb_first_half)
mlb_second_half = np.array(mlb_second_half)

nhl_first_half = np.array(nhl_first_half)
nhl_second_half = np.array(nhl_second_half)

nfl_first_half = np.array(nfl_first_half)
nfl_second_half = np.array(nfl_second_half)
```

In [51]:
```
def single_bootstrap(l1, l2):
  boot1 = np.zeros(1000)
  boot2 = np.zeros(1000)
  for i in range(1000):
```

```
    boot1[i] = np.random.choice(l1, size=len(l1), replace=True).mean()
    boot2[i] = np.random.choice(l2, size=len(l2), replace=True).mean()
  return  boot1 - boot2

nba_mlb_first = single_bootstrap(nba_first_half, mlb_first_half)
nba_mlb_second = single_bootstrap(nba_second_half, mlb_second_half)

nba_nhl_first  = single_bootstrap(nba_first_half, nhl_first_half)
nba_nhl_second = single_bootstrap(nba_second_half, nhl_second_half)


nba_nfl_first = single_bootstrap(nba_first_half, nfl_first_half)
nba_nfl_second = single_bootstrap(nba_second_half, nfl_second_half)

mlb_nhl_first = single_bootstrap(mlb_first_half, nhl_first_half)
mlb_nhl_second = single_bootstrap(mlb_second_half, nhl_second_half)

mlb_nfl_first = single_bootstrap(mlb_first_half, nfl_first_half)
mlb_nfl_second = single_bootstrap(mlb_second_half, nfl_second_half)

nhl_nfl_first = single_bootstrap(nhl_first_half, nfl_first_half)
nhl_nfl_second = single_bootstrap(nhl_second_half, nfl_second_half)
```

In [49]:

```
true_nba_mlb_first = nba_first_half.mean() - mlb_first_half.mean()
true_nba_mlb_second = nba_second_half.mean() - mlb_second_half.mean()

true_nba_nhl_first  = nba_first_half.mean() - nhl_first_half.mean()
true_nba_nhl_second = nba_second_half.mean() - nhl_second_half.mean()


true_nba_nfl_first = nba_first_half.mean() - nfl_first_half.mean()
true_nba_nfl_second = nba_second_half.mean() - nfl_second_half.mean()

true_mlb_nhl_first = mlb_first_half.mean() - nhl_first_half.mean()
true_mlb_nhl_second = mlb_second_half.mean() - nhl_second_half.mean()

true_mlb_nfl_first = mlb_first_half.mean() - nfl_first_half.mean()
true_mlb_nfl_second = mlb_second_half.mean() - nfl_second_half.mean()

true_nhl_nfl_first = nhl_first_half.mean() - nfl_first_half.mean()
true_nhl_nfl_second = nhl_second_half.mean() - nfl_second_half.mean()
```

In [69]:

```
def significance_test(boot, true, name):
  pvalue = (boot > true).mean()
  sig = '*** ' if pvalue < 0.001 else \
        '**  ' if pvalue < 0.01 else \
        '*   ' if pvalue < 0.05 else '    '
  print(f'{sig} {name} -> {pvalue.round(5)}')
  return None
```

In [75]:

```
tests = [
  (nba_mlb_first, true_nba_mlb_first, 'First Half NBA MLB'),
  (nba_mlb_second, true_nba_mlb_second, 'Second Half NBA MLB'),
  (nba_nhl_first, true_nba_nhl_first, 'First Half NBA NHL'),
  (nba_nhl_second, true_nba_nhl_second, 'Second Half NBA NHL'),
  (nba_nfl_first, true_nba_nfl_first, 'First Half NBA NFL'),
  (nba_nfl_second, true_nba_nfl_second, 'Second Half NBA NFL'),

  (mlb_nhl_first, true_mlb_nhl_first, 'First Half MLB NHL'),
```

```
  (mlb_nhl_second, true_mlb_nhl_second, 'Second Half MLB NHL'),
  (mlb_nfl_first, true_mlb_nfl_first, 'First Half MLB NFL'),
  (mlb_nfl_second, true_mlb_nfl_second, 'Second Half MLB NFL'),

  (nhl_nfl_first, true_nhl_nfl_first, 'First Half NHL NFL'),
  (nhl_nfl_second, true_nhl_nfl_second, 'Second Half NHL NFL'),
]
print('Significance tests')
print('sig -------test-------    pvalue')
[significance_test(boot, true, name) for (boot, true, name) in tests]
```

```
Significance tests
sig -------test-------    pvalue
    First Half NBA MLB -> 0.497
    Second Half NBA MLB -> 0.5
    First Half NBA NHL -> 0.505
    Second Half NBA NHL -> 0.472
    First Half NBA NFL -> 0.486
    Second Half NBA NFL -> 0.518
    First Half MLB NHL -> 0.507
    Second Half MLB NHL -> 0.503
    First Half MLB NFL -> 0.516
    Second Half MLB NFL -> 0.511
    First Half NHL NFL -> 0.506
    Second Half NHL NFL -> 0.469
```

Out[75]: [None, None, None, None, None, None, None, None, None, None, None, None]

Appendix 2: Preliminary Analysis and Phase 2

AUTHORS: Jake Lawson; Geoff Miller; Evan McGowan

RESEARCH QUESTIONS

How does the first half of season performance predict the second half of the season and playoff performance within the four leagues considering wins, losses, scorings, and elo ratings?

In this project we want to see if a team's first season performance could be used to predict the teams next season and playoff performance using the 1990-2020 seasons of the NFL, NHL, NBA and MLB.

We were originally intending to compare totals wins and losses across all teams regardless of sport. However, after examining our data and comparing these numbers, it became apparent that there is a large difference in total games played across these four sports. This led us to shift our original idea of comparing statistics between sports towards predicting future success given past success both within and between seasons. We were also curious about the effect of a large home team advantage for each of the four sports. By looking at team performance at home/away/neutral sites we hope to observe and analyze trends in these advantages/disadvantages.

Looking at the final Graph, our research question can be summed up as: how well do these two lines track? Does past performance predict future output or will teams choke at the last minute?

DESCRIPTION OF DATA

We used one data table for each of the four leagues of focus: the NFL, NHL, NBA and MLB. The ranges of time in these tables are highly varied, with some going back all the way to the 1800s and others, only the early 1900s. To solve the inconsistent time periods, we limited all data to years 1990-2020 to ensure our data is relevant and can be used for comparison. Each column of the tables has data of seasons, team names, team ratings, key player names, player ratings, probability of wins, dates and scores.

The ratings and game value are all complex values, derived from mathematical equations, that help to describe participant teams and their likelihood of success. Elo ratings– which are a greater-than-zero numerical representation of a team's relative skill– are calculated using a system of point allocation from wins and losses that allocate larger amounts of points when a weaker team wins and vice versa.

There is the additional issue of differing scoring structures and point allocations. For example, in the NFL, teams earn 6 points per touchdown and an extra 1 or 2 points depending on if they kick or go for a conversion. In the NHL and MLB, there is one point allocated to a team for a given run/goal. In the NBA the points vary by shot distance, with three points being allocated for a shot behind the 3-point line, two point from within the arc, and 1 point for a stationary, penalty shot at a closer distance (this distance is known creatively as the "free-throw line")

As of right now we include all provided data fields from these sources. We expect there to be high colinarity between these features, particularly the elo ratings assigned to each of these teams. However it might be worthwhile to examine trends in these features in future phases of the project so we do not omit them at this time.

- Source for all data tables: https://data.fivethirtyeight.com/
- Source for NBA data table:https://github.com/fivethirtyeight/data/tree/master/nba-forecasts
- Source for MLB data table:https://github.com/fivethirtyeight/data/tree/master/mlb-elo
- Source for NHL data table:https://github.com/fivethirtyeight/data/tree/master/nhl-forecasts
- Source for NFL data table:https://github.com/fivethirtyeight/data/tree/master/nfl-elo

Data Clean UP

The first step was to convert the original CSV file to pandas dataframes that could be manipulated and used for additional calculations. We then began to clean our data, standardizing dates, fixing NaNs for playoff categories by creating qualitative variables, and shortening the regular season variable values. After our data was clean, we reorganized the dataframe to display target variables that would be of specific usefulness to us in our later calculations and analysis. With our data organized, we broke the larger data frame into four smaller tables that represented each independent sport (NFL, NBA, MLB, NHL). In order to draw comparisons between the first and second halves of a season, we needed to divide each season into halves with regard for each individual sport's varying season dates and duration. For the NBA and NHL we defined the first half as the months October to December and the second half

as months January to April. For the MLB we defined the first half as months March to June and the second half as months July to September.

For the playoff columns, we defined variable r1 as the first half of the regular season and variable r2 as the second half of the regular season. Descriptions of how we mapped this values is below and is specific to each league.

IMPORTS

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
mlb = pd.read_csv('data/mlb-elo/mlb_elo.csv')
nba = pd.read_csv('data/nba-forecasts/nba_elo.csv')
nfl = pd.read_csv('data/nfl-elo/nfl_elo.csv')
nhl = pd.read_csv('data/nhl-forecasts/nhl_elo.csv')
```

DATES

Restrict to

In [ ]:
```
mlb.date = pd.to_datetime(mlb.date)
nba.date = pd.to_datetime(nba.date)
nfl.date = pd.to_datetime(nfl.date)
nhl.date = pd.to_datetime(nhl.date)
```

In [ ]:
```
def year_range(df):
    dt_index = pd.DatetimeIndex(df.date)
    cond = (dt_index.year >= 1990) & (dt_index.year < 2020)
    return df[cond]

mlb = year_range(mlb)
nba = year_range(nba)
nfl = year_range(nfl)
nhl = year_range(nhl)
```

In [ ]:
```
mlb.head()
```

	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_prob1	elo_p
8316	2019-10-30	2019	0	w	HOU	WSN	1599.542804	1584.363378	0.574617	0.42
8317	2019-10-29	2019	0	w	HOU	WSN	1605.069000	1578.837182	0.595209	0.40
8318	2019-10-27	2019	0	w	WSN	HOU	1584.005206	1599.900976	0.515546	0.48
8319	2019-10-26	2019	0	w	WSN	HOU	1589.985555	1593.920627	0.538425	0.46
8320	2019-10-25	2019	0	w	WSN	HOU	1593.827376	1590.078806	0.553044	0.44

5 rows × 26 columns

In [ ]:
```
nba.head()
```

	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_prob1	elo_prob
31335	1990-01-02	1990	0	NaN	NYK	PHO	1620.0157	1565.4576	0.708830	0.29117
31336	1990-01-02	1990	0	NaN	WSB	NJN	1466.4788	1325.2555	0.800368	0.19963
31337	1990-01-02	1990	0	NaN	POR	MIA	1527.1683	1263.9166	0.890030	0.10997
31338	1990-01-02	1990	0	NaN	ATL	MIL	1545.9711	1527.2737	0.664470	0.33553
31339	1990-01-02	1990	0	NaN	DAL	IND	1457.7749	1528.7625	0.541655	0.45834

5 rows × 27 columns

In [ ]:
```
nfl.head()
```

	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_prob1	elo_prob2
8772	1990-01-06	1989	0	d	SF	MIN	1741.063	1595.514	0.770656	0.229344
8773	1990-01-06	1989	0	d	CLE	BUF	1565.876	1576.806	0.577191	0.422809
8774	1990-01-07	1989	0	d	DEN	PIT	1599.653	1556.821	0.650385	0.349615
8775	1990-01-07	1989	0	d	NYG	LAR	1623.766	1643.646	0.564570	0.435430
8776	1990-01-14	1989	0	c	SF	LAR	1755.160	1666.078	0.708264	0.291736

5 rows × 33 columns

In [ ]:
```
nhl.head()
```

	season	date	playoff	neutral	status	ot	home_team	away_team	home_team_abbr
26609	1990	1990-01-01	0	0	post	NaN	Washington Capitals	Los Angeles Kings	WSH
26610	1990	1990-01-02	0	0	post	NaN	St. Louis Blues	Edmonton Oilers	STL
26611	1990	1990-01-02	0	0	post	OT	Calgary Flames	Philadelphia Flyers	CGY
26612	1990	1990-01-02	0	0	post	NaN	New Jersey Devils	Buffalo Sabres	NJD

	season	date	playoff	neutral	status	ot	home_team	away_team	home_team_abbr
26613	1990	1990-01-02	0	0	post	NaN	New York Islanders	Los Angeles Kings	NYI

5 rows × 24 columns

Playoffs

In [ ]:
```
print('NHL', nhl.playoff.unique())
print('NBA', nba.playoff.unique())
print('MLB', mlb.playoff.unique())
print('NFL', nfl.playoff.unique())
```

```
NHL [0 1]
NBA [nan 't' 'q' 's' 'c' 'f']
MLB ['w' 'l' 'd' 'c' nan]
NFL ['d' 'c' 's' nan 'w']
```

NHL:

- 0 -> not playoff
- 1 -> playoff

NBA:

- f -> final
- s -> semi
- q -> quarter
- c -> confernce
- t -> tournaments
- r1 -> 1st half regular season
- r2 -> 2st half regular season

MLB:

- w -> world series
- l -> league conference
- d -> division
- c -> wild card
- r1 -> 1st half regular season
- r2 -> 2st half regular season

NFL

- s -> super bowl
- d -> division
- c -> conference
- w -> wildcard
- r1 -> 1st half regular season
- r2 -> 2st half regular season

In [ ]:
```
mlb.playoff = mlb.playoff.fillna('r')
nfl.playoff = nfl.playoff.fillna('r')
nba.playoff = nba.playoff.fillna('r')
```

Divide regular season

NFL

- 1st half -> September, October
- 2nd Half -> November, December, January

NBA

- 1st half -> October - December
- 2nd Half -> January - April

NHL

- 1st half -> October - December
- 2nd Half -> January - April

MLB

- 1st half -> March - June
- 2nd Half -> July - September

In [ ]:
```
nfl_month = pd.DatetimeIndex(nfl.date).month
nba_month = pd.DatetimeIndex(nba.date).month
nhl_month = pd.DatetimeIndex(nhl.date).month
mlb_month = pd.DatetimeIndex(mlb.date).month
```

NFL

In [ ]:
```
nfl.loc[((nfl_month == 9) | (nfl_month == 10)),\
        'playoff'] = 'r1'
nfl.loc[((nfl_month == 11) | (nfl_month == 12) \
         | (nfl_month == 13)), 'playoff'] = 'r2'
nfl
```

	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_prob1	elo_
8772	1990-01-06	1989	0	d	SF	MIN	1741.063000	1595.514000	0.770656	0.2
8773	1990-01-06	1989	0	d	CLE	BUF	1565.876000	1576.806000	0.577191	0.4
8774	1990-01-07	1989	0	d	DEN	PIT	1599.653000	1556.821000	0.650385	0.3
8775	1990-01-07	1989	0	d	NYG	LAR	1623.766000	1643.646000	0.564570	0.4
8776	1990-01-14	1989	0	c	SF	LAR	1755.160000	1666.078000	0.708264	0.2
...	...	...	...	...	...	...	...	...	...	...

	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_prob1	elo_
16525	2019-12-29	2019	0	r2	HOU	TEN	1584.215388	1542.863113	0.648445	0.3
16526	2019-12-29	2019	0	r2	DEN	OAK	1487.042014	1407.432240	0.696871	0.3
16527	2019-12-29	2019	0	r2	DAL	WSH	1536.023522	1309.885760	0.842364	0.1
16528	2019-12-29	2019	0	r2	NYG	PHI	1351.254830	1551.327497	0.314850	0.6
16529	2019-12-29	2019	0	r2	SEA	SF	1570.662276	1609.709252	0.537280	0.4

7758 rows × 33 columns

In [ ]:
```
len((nba_month == 10))
```

37574

In [ ]:
```
nba.loc[((nba_month == 10) | (nba_month == 11) \
         | (nba_month == 12)), 'playoff'] = 'r1'
nba.loc[((nba_month == 1) | (nba_month == 2) \
         | (nba_month == 3)), 'playoff'] = 'r2'
nba
```

	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_prob1	elo_
31335	1990-01-02	1990	0	r2	NYK	PHO	1620.015700	1565.457600	0.708830	0.
31336	1990-01-02	1990	0	r2	WSB	NJN	1466.478800	1325.255500	0.800368	0.1
31337	1990-01-02	1990	0	r2	POR	MIA	1527.168300	1263.916600	0.890030	0.1
31338	1990-01-02	1990	0	r2	ATL	MIL	1545.971100	1527.273700	0.664470	0.3
31339	1990-01-02	1990	0	r2	DAL	IND	1457.774900	1528.762500	0.541655	0.4
...	...	...	...	...	...	...	...	...	...	
68904	2019-12-31	2020	0	r1	SAC	LAC	1439.142942	1631.787446	0.369746	0.6
68905	2019-12-31	2020	0	r1	HOU	DEN	1629.924907	1618.388168	0.655218	0.3
68906	2019-12-31	2020	0	r1	TOR	CLE	1642.844707	1341.079149	0.909927	0.0
68907	2019-12-31	2020	0	r1	SAS	GSW	1484.142262	1412.580576	0.728611	0.2
68908	2019-12-31	2020	0	r1	OKC	DAL	1550.631521	1605.977237	0.563911	0.4

37574 rows × 27 columns

In [ ]:
```
nhl.loc[((nhl_month == 10) | (nhl_month == 11) \
         | (nhl_month == 12)), 'playoff'] = 'r1'
nhl.loc[((nhl_month == 1) | (nhl_month == 2) \
         | (nhl_month == 3)), 'playoff'] = 'r2'
nhl
```

	season	date	playoff	neutral	status	ot	home_team	away_team	home_team_abbr
26609	1990	1990-01-01	r2	0	post	NaN	Washington Capitals	Los Angeles Kings	WSH
26610	1990	1990-01-02	r2	0	post	NaN	St. Louis Blues	Edmonton Oilers	STL
26611	1990	1990-01-02	r2	0	post	OT	Calgary Flames	Philadelphia Flyers	CGY
26612	1990	1990-01-02	r2	0	post	NaN	New Jersey Devils	Buffalo Sabres	NJD
26613	1990	1990-01-02	r2	0	post	NaN	New York Islanders	Los Angeles Kings	NYI
...	...	...	...	...	...	...	...	...	...
61773	2020	2019-12-31	r1	0	post	NaN	Calgary Flames	Chicago Blackhawks	CGY
61774	2020	2019-12-31	r1	0	post	NaN	Carolina Hurricanes	Montreal Canadiens	CAR
61775	2020	2019-12-31	r1	0	post	NaN	Buffalo Sabres	Tampa Bay Lightning	BUF
61776	2020	2019-12-31	r1	0	post	NaN	Arizona Coyotes	St. Louis Blues	ARI
61777	2020	2019-12-31	r1	0	post	NaN	Columbus Blue Jackets	Florida Panthers	CBJ

35169 rows × 24 columns

In [ ]:
```
mlb.loc[((mlb_month == 3) | (mlb_month == 4) | \
         (mlb_month == 5) | (mlb_month == 6)), 'playoff'] = 'r1'
mlb.loc[((mlb_month == 7) | (mlb_month == 8) | \
         (mlb_month == 9)), 'playoff'] = 'r2'
mlb
```

	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_prob1	elo_
8316	2019-10-30	2019	0	w	HOU	WSN	1599.542804	1584.363378	0.574617	0.4
8317	2019-10-29	2019	0	w	HOU	WSN	1605.069000	1578.837182	0.595209	0.4
8318	2019-10-27	2019	0	w	WSN	HOU	1584.005206	1599.900976	0.515546	0.4
8319	2019-	2019	0	w	WSN	HOU	1589.985555	1593.920627	0.538425	0.4

	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_prob1	elo_
	10-26									
8320	2019-10-25	2019	0	w	WSN	HOU	1593.827376	1590.078806	0.553044	0.4
...	...	...	...	...	...	...	...	...	...	
79399	1990-04-09	1990	0	r1	KCR	BAL	1520.125000	1500.548000	0.562386	0.4
79400	1990-04-09	1990	0	r1	HOU	CIN	1504.135000	1492.159000	0.551589	0.4
79401	1990-04-09	1990	0	r1	CHW	MIL	1491.470000	1516.772000	0.498126	0.5
79402	1990-04-09	1990	0	r1	BOS	DET	1519.201000	1462.610000	0.613943	0.3
79403	1990-04-09	1990	0	r1	ANA	SEA	1515.923000	1490.516000	0.570627	0.4

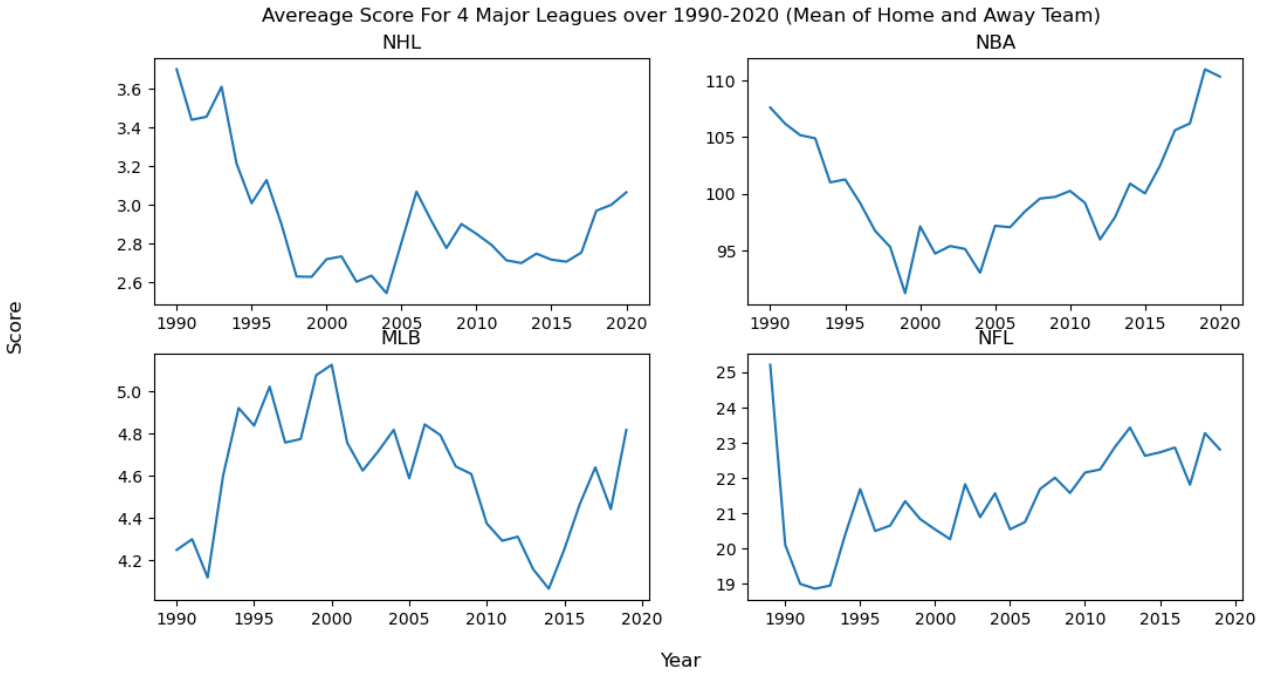
71088 rows × 26 columns

Data Visulazations

The four line graphs of team average scores for each specific sport allows us to observe the manner in which the scoring system differs and affects our ability to compare raw wins and losses between sports. With the use of box plots and further analysis, we can proportionally (i.e. comparing the win/loss rate and total score with regard to total number of games played) compare each sport and observe that the win/loss ratio is relatively constant (with a few notable outliers). We can, additionally, use these plots to observe the relationship between the first and second halves of seasons, e.g. does success in the first half predict success in the second half. We concluded that there is, in fact, a strong correlation between a team's likelihood to succeed in the 2nd half of their season given their success in the first

In [ ]:
```
mlb_score = (mlb.groupby(by='season').score1.mean()\
             + mlb.groupby(by='season').score2.mean()) / 2
nba_score = (nba.groupby(by='season').score1.mean()\
             + nba.groupby(by='season').score2.mean()) / 2
nhl_score = (nhl.groupby(by='season').home_team_score.mean()\
             + nhl.groupby(by='season').away_team_score.mean()) / 2
nfl_score = (nfl.groupby(by='season').score1.mean()\
             + nfl.groupby(by='season').score2.mean()) / 2

fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(12,6))
fig.suptitle('''
Avereage Score For 4 Major Leagues over 1990-2020 (Mean of Home and Away Team)''')
fig.supxlabel('Year')
fig.supylabel('Score')
```

```
axs[0,0].plot(nhl_score)
axs[0,0].set_title('NHL')

axs[0,1].plot(nba_score)
axs[0,1].set_title('NBA')

axs[1,0].plot(mlb_score)
axs[1,0].set_title('MLB')

axs[1,1].plot(nfl_score)
axs[1,1].set_title('NFL')

plt.show()
```



TRENDS

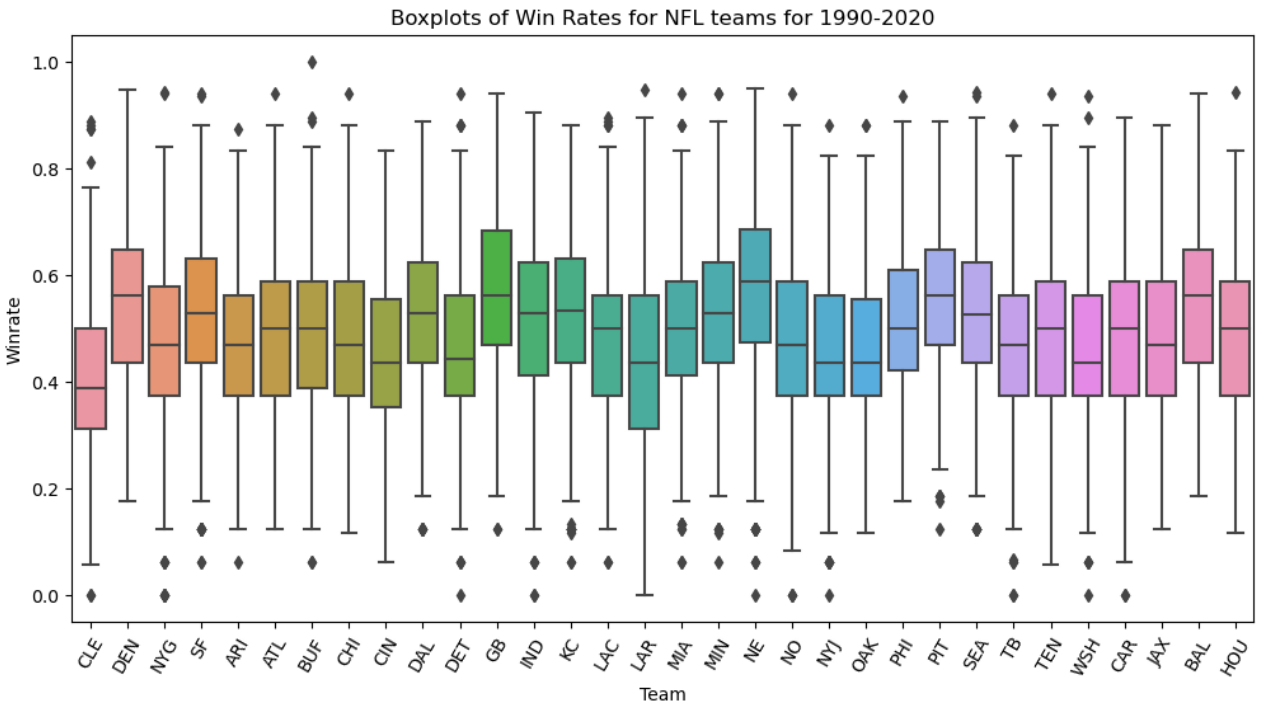
Average scores for a team has decrease over time in the NHL. Scores decreased in the NBA until about the year 2000 and have been increasing ever since. MLB scores were relativly stable until the late 2000s where they took a sharp decline for a decade and have been recovering ever since. There was a sharp decline in socre in the NFL followed by a steady yet slow rise since 1995.

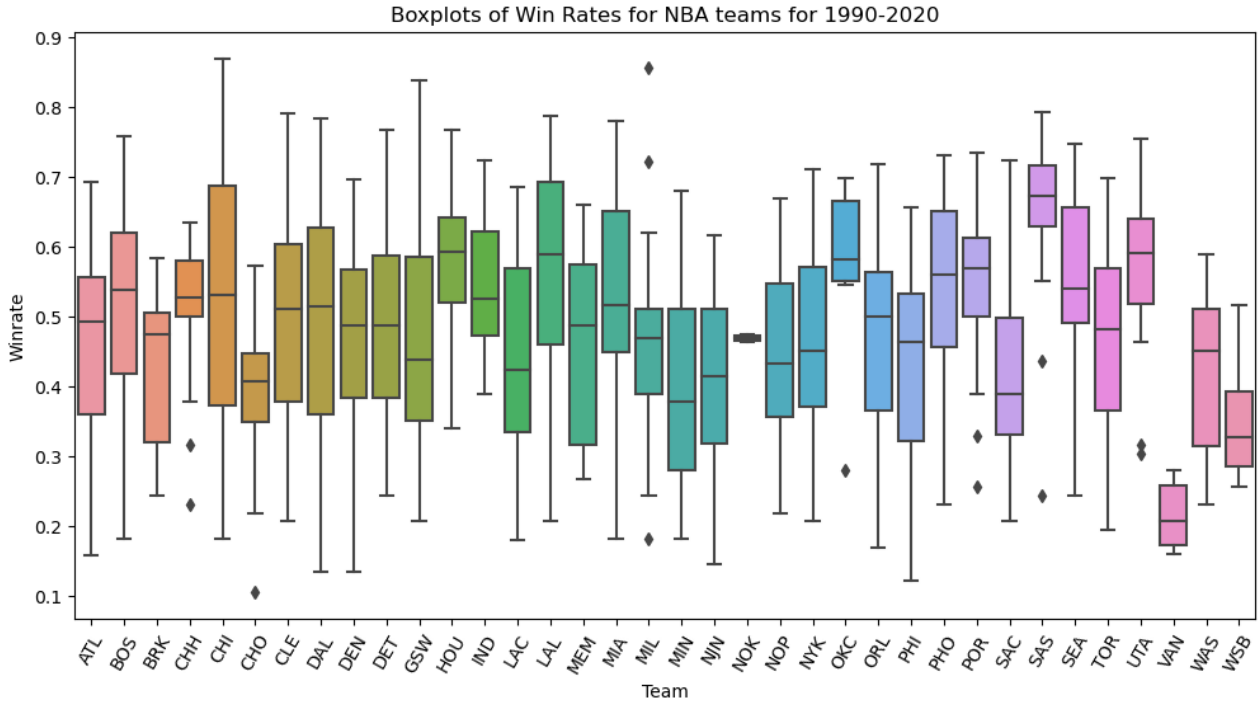
In [ ]:
```
def boxplots(df, name):
    df['win1'] = (df.score1 > df.score2)
    df['win2'] = (df.score1 < df.score2)
    wins = df.groupby(by=['team1', 'season']).win1.sum()\
    + df.groupby(by=['team2', 'season']).win2.sum()
    games_played = df.groupby(by=['team1', 'season']).count().date\
    + df.groupby(by=['team2', 'season']).count().date
    winrate = wins/games_played

    plt.figure(figsize=(12,6))
    sns.boxplot(winrate.reset_index(), x='team1', y=0)
    plt.xticks(rotation=60)
    plt.xlabel('Team')
```

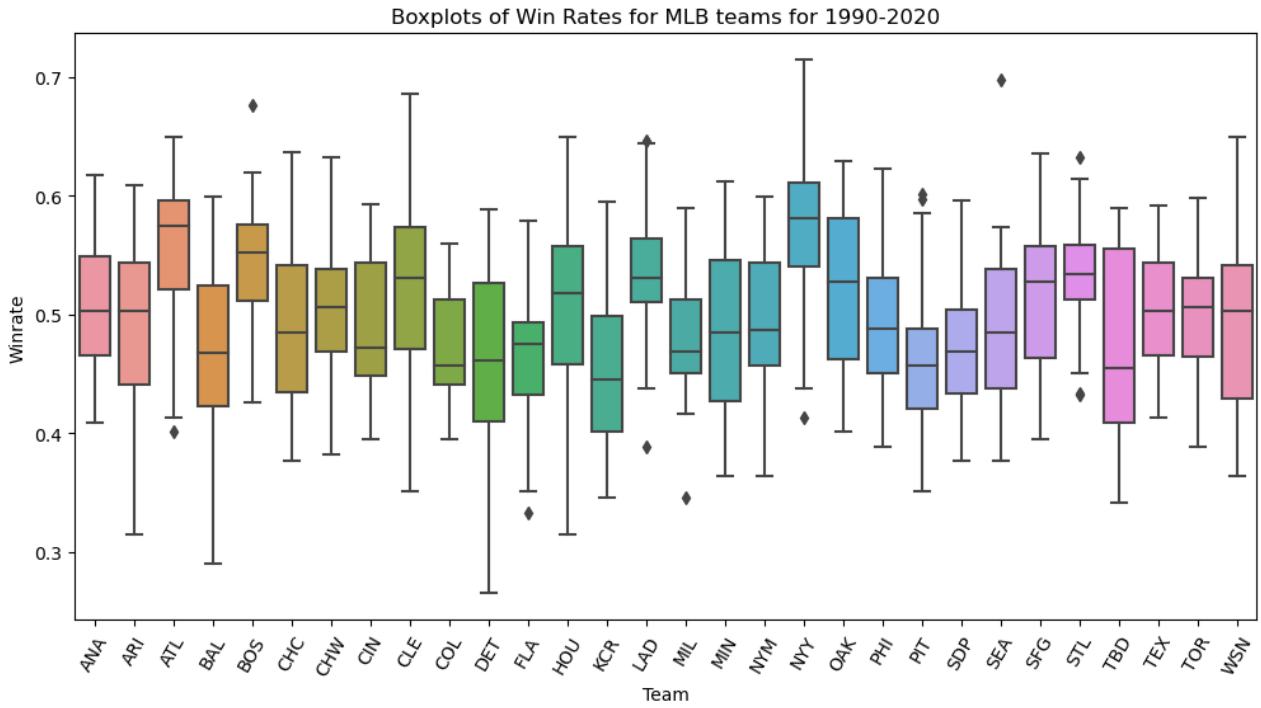
```
    plt.ylabel('Winrate')
    plt.title(f"Boxplots of Win Rates for {name} teams for 1990-2020")
    plt.show()
```

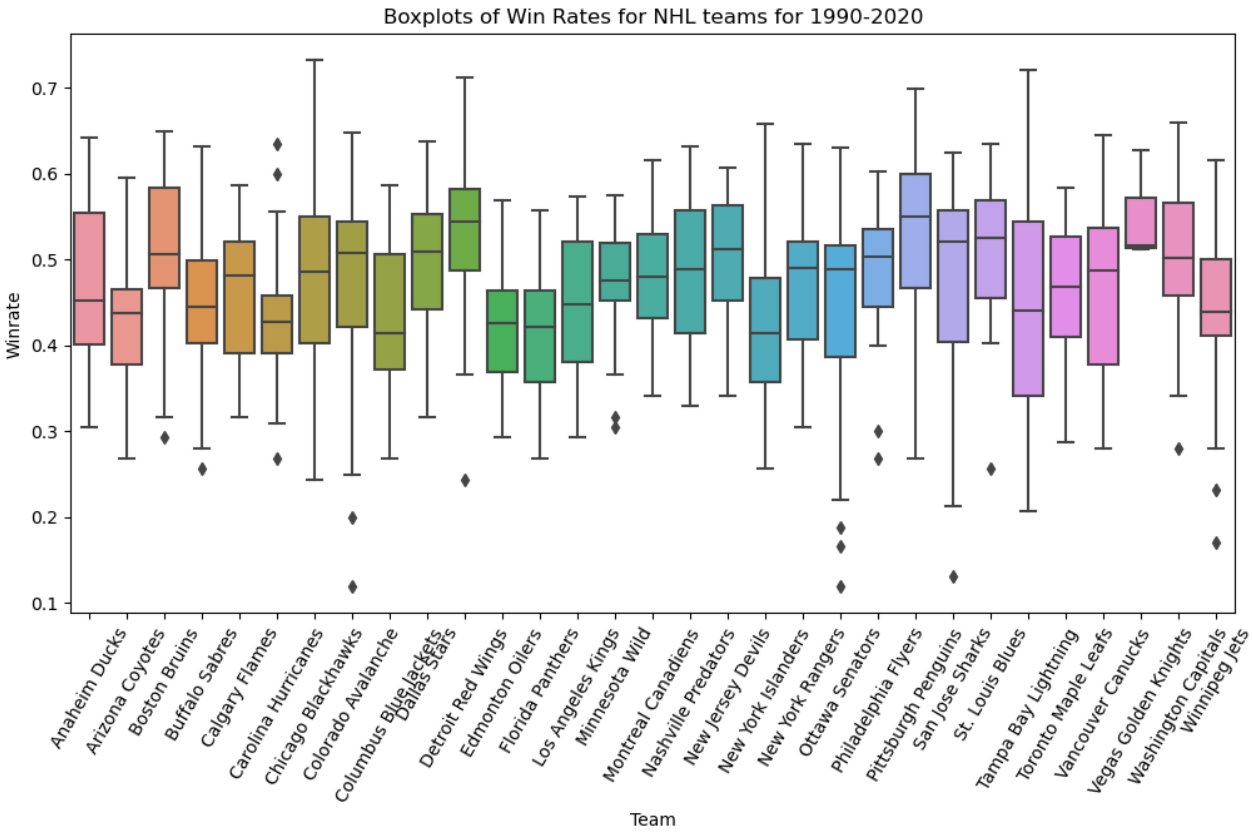
In [ ]:

```
boxplots(nfl, 'NFL')
boxplots(nba, 'NBA')
boxplots(mlb, 'MLB')
nhl['win1'] = (nhl.home_team_score > nhl.away_team_score)
nhl['win2'] = (nhl.home_team_score < nhl.away_team_score)
wins = nhl.groupby(by=['home_team', 'season']).win1.sum()\
+ nhl.groupby(by=['away_team', 'season']).win2.sum()
games_played = nhl.groupby(by=['home_team', 'season']).count().date \
+ nhl.groupby(by=['away_team', 'season']).count().date
winrate = wins/games_played

plt.figure(figsize=(12,6))
sns.boxplot(winrate.reset_index(), x='home_team', y=0)
plt.xticks(rotation=60)
plt.xlabel('Team')
plt.ylabel('Winrate')
plt.title(f"Boxplots of Win Rates for NHL teams for 1990-2020")
plt.show()
```




Boxplots of Win Rates for NBA teams for 1990-2020
Winrate
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
ATL BOS BRK CHH CHI CHO CLE DAL DEN DET GSW HOU IND LAC LAL MEM MIA MIL MIN NJN NOK NOP NYK OKC ORL PHI PHO POR SAC SAS SEA TOR UTA VAN WAS WSB
Team


Boxplots of Win Rates for MLB teams for 1990-2020
Winrate
0.7
0.6
0.5
0.4
0.3
ANA ARI ATL BAL BOS CHC CHW CIN CLE COL DET FLA HOU KCR LAD MIL MIN NYM NYY OAK PHI PIT SDP SEA SFG STL TBD TEX TOR WSN
Team



TRENDS

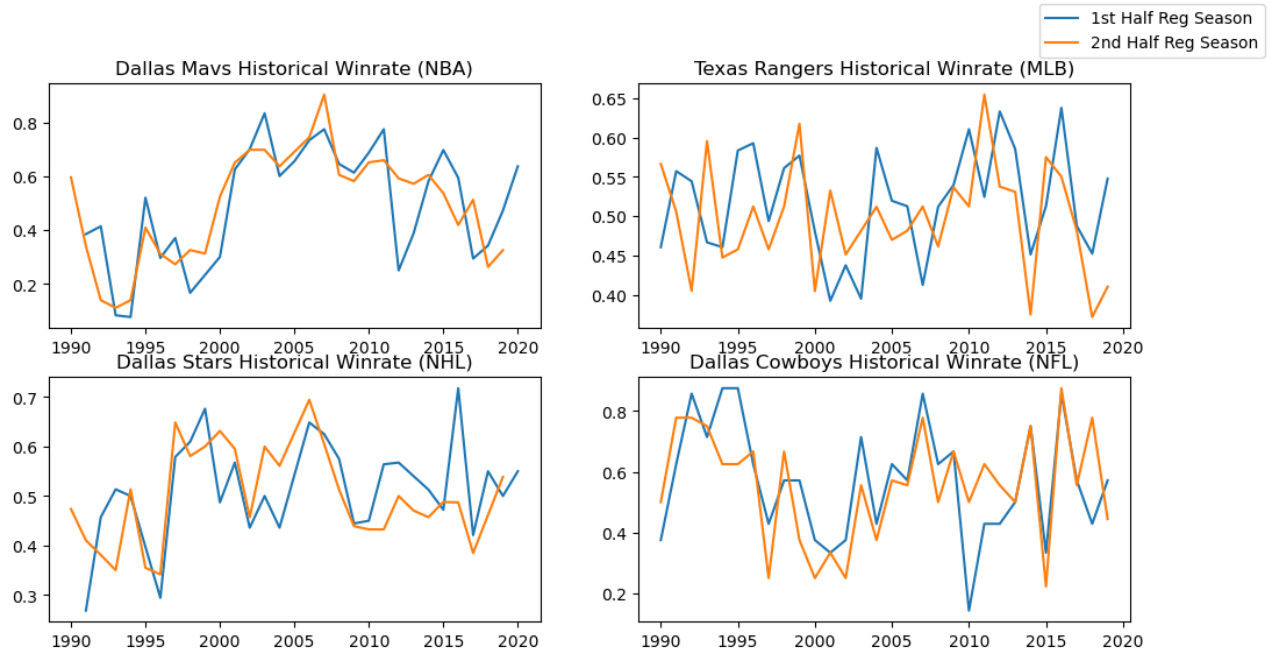
For the past thirty years, there have been goods seasons and bad seasons for almost every team, but there are a few constants in the data. Except for a few teams that always preform well (or, rather never preform poorly), m,ost teams win as much as they loose on average. Every teams has had very good seasons winning with the Chicago Bulls winning almost .900 of their games at their peak. But this is balanced out by the bad teams in those years loosing a commensurate ammount of games. NHL and MLB teams have the tightest spread in distribution followed by NBA and then NFL with the most spread. This comes from the fact that NBA, NHL, and MLB teams play 82 games in the regular season while NFL teams play 17.

Comparing first and second half of regular season win rates

In [ ]:
```
fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(12,6))

dalr1 = nba[((nba.team1 == 'DAL') | (nba.team2 == 'DAL')) & (nba.playoff == 'r1'
dalr2 = nba[((nba.team1 == 'DAL') | (nba.team2 == 'DAL')) & (nba.playoff == 'r2'

winr1 = dalr1[dalr1.team1 == 'DAL'].groupby(by='season').win1.sum() \
    + dalr1[dalr1.team2 == 'DAL'].groupby(by='season').win2.sum()
gamesr1 = dalr1.groupby(by='season').count().date

winr2 = dalr2[dalr2.team1 == 'DAL'].groupby(by='season').win1.sum() \
    + dalr2[dalr2.team2 == 'DAL'].groupby(by='season').win2.sum()
gamesr2 = dalr2.groupby(by='season').count().date

winrater1 = winr1/gamesr1
```

```
winrater2 = winr2/gamesr2
axs[0,0]
axs[0,0].set_title('Dallas Mavs Historical Winrate (NBA)')
axs[0,0].plot(winrater1, label='1st Half Reg Season')
axs[0,0].plot(winrater2, label='2nd Half Reg Season')



dalr1 = mlb[((mlb.team1 == 'TEX') | (mlb.team2 == 'TEX')) & (mlb.playoff == 'r1'
dalr2 = mlb[((mlb.team1 == 'TEX') | (mlb.team2 == 'TEX')) & (mlb.playoff == 'r2'

winr1 = dalr1[dalr1.team1 == 'TEX'].groupby(by='season').win1.sum() \
    + dalr1[dalr1.team2 == 'TEX'].groupby(by='season').win2.sum()
gamesr1 = dalr1.groupby(by='season').count().date

winr2 = dalr2[dalr2.team1 == 'TEX'].groupby(by='season').win1.sum() \
    + dalr2[dalr2.team2 == 'TEX'].groupby(by='season').win2.sum()
gamesr2 = dalr2.groupby(by='season').count().date

winrater1 = winr1/gamesr1
winrater2 = winr2/gamesr2

axs[0,1].set_title('Texas Rangers Historical Winrate (MLB)')
axs[0,1].plot(winrater1, label='1st Half Reg Season')
axs[0,1].plot(winrater2, label='2nd Half Reg Season')



dalr1 = nfl[((nfl.team1 == 'DAL') | (nfl.team2 == 'DAL')) & (nfl.playoff == 'r1'
dalr2 = nfl[((nfl.team1 == 'DAL') | (nfl.team2 == 'DAL')) & (nfl.playoff == 'r2'

winr1 = dalr1[dalr1.team1 == 'DAL'].groupby(by='season').win1.sum() \
    + dalr1[dalr1.team2 == 'DAL'].groupby(by='season').win2.sum()
gamesr1 = dalr1.groupby(by='season').count().date

winr2 = dalr2[dalr2.team1 == 'DAL'].groupby(by='season').win1.sum() \
    + dalr2[dalr2.team2 == 'DAL'].groupby(by='season').win2.sum()
gamesr2 = dalr2.groupby(by='season').count().date

winrater1 = winr1/gamesr1
winrater2 = winr2/gamesr2

axs[1,1].set_title('Dallas Cowboys Historical Winrate (NFL)')
axs[1,1].plot(winrater1, label='1st Half Reg Season')
axs[1,1].plot(winrater2, label='2nd Half Reg Season')



dalr1 = nhl[((nhl.home_team == 'Dallas Stars') | (nhl.away_team == 'Dallas Stars
dalr2 = nhl[((nhl.home_team == 'Dallas Stars') | (nhl.away_team == 'Dallas Stars

winr1 = dalr1[dalr1.home_team == 'Dallas Stars'].groupby(by='season').win1.sum()
    + dalr1[dalr1.away_team == 'Dallas Stars'].groupby(by='season').win2.sum()
gamesr1 = dalr1.groupby(by='season').count().date

winr2 = dalr2[dalr2.home_team == 'Dallas Stars'].groupby(by='season').win1.sum()
    + dalr2[dalr2.away_team == 'Dallas Stars'].groupby(by='season').win2.sum()
gamesr2 = dalr2.groupby(by='season').count().date

winrater1 = winr1/gamesr1
```

```
winrater2 = winr2/gamesr2

axs[1,0].set_title('Dallas Stars Historical Winrate (NHL)')
l1, = axs[1,0].plot(winrater1, label='1st Half Reg Season')
l2, = axs[1,0].plot(winrater2, label='2nd Half Reg Season')

fig.legend(handles=[l1, l2])
plt.show()
```



OUR MAIN QUESTION

As mentioned above, our main question lies in this graph: does early season success predict late season success? How good is this over all the teams, all the leagues?

Difficulties faced during data collection

During the data collection, we faced issues on accurately calculating specific metrics from each of the four tables. When creating our initial line graphs, we faced the issue of fitting multiple teams of different sports on the same graph due to large differences in games played. Each record is a game in all four tables where there is a home and an away team. If we want to do group and joins on these tables wth the names of the teams we need to deal with two columns of names instead of one.

Limitations

A NFL team will play 17 games in a calendar season making each game crucial in their chances at winning the super bowl. For contrast, a NHL or NBA team will play 82 games in a season and will have lower win rates, on average, than NFL teams. This will result in different measures of success– as defined by win/loss ratios– for different teams, and we will not be able to predict future outcomes using models created for a different sport with less games per season.

Each sport has different average point spreads as a result of varied scoring structures. This will have to be accounted for in order to meaningfully compare sports where point allocations are different.

If there were a significant roster chance due to mid-game injury or some unforeseen event, the ELO rating would not be useful in determining a team's likelihood to succeed. Additionally, this statistic is based on historical analysis and may not account for specific matchup dynamics that could affect the outcome of a game.

Questions for reviewers

Do you think our dataset is large enough such that we can draw accurate conclusions and create a model using a multivariable regression?

Did we properly illustrate the data of trends and outcomes with the charts and graphs present?

Was our description of how all four leagues match our research question correct enough for someone to understand sports without a background on such?

In [ ]:

	date	season	neutral	playoff	team1	team2	elo1_pre	elo2_pre	elo_prob1	elo_
8316	2019-10-30	2019	0	w	HOU	WSN	1599.542804	1584.363378	0.574617	0.4
8317	2019-10-29	2019	0	w	HOU	WSN	1605.069000	1578.837182	0.595209	0.4
8318	2019-10-27	2019	0	w	WSN	HOU	1584.005206	1599.900976	0.515546	0.4
8319	2019-10-26	2019	0	w	WSN	HOU	1589.985555	1593.920627	0.538425	0.4
8320	2019-10-25	2019	0	w	WSN	HOU	1593.827376	1590.078806	0.553044	0.4
...	...	...	...	...	...	...	...	...	...	
79399	1990-04-09	1990	0	r1	KCR	BAL	1520.125000	1500.548000	0.562386	0.4
79400	1990-04-09	1990	0	r1	HOU	CIN	1504.135000	1492.159000	0.551589	0.4
79401	1990-04-09	1990	0	r1	CHW	MIL	1491.470000	1516.772000	0.498126	0.5
79402	1990-04-09	1990	0	r1	BOS	DET	1519.201000	1462.610000	0.613943	0.3
79403	1990-04-09	1990	0	r1	ANA	SEA	1515.923000	1490.516000	0.570627	0.4

71088 rows × 28 columns

In [ ]:
```
mlb.to_csv('data/cleaned/mlb.csv')
nba.to_csv('data/cleaned/nba.csv')
nfl.to_csv('data/cleaned/nfl.csv')
nhl.to_csv('data/cleaned/nhl.csv')
```

In [ ]: