Giancarlo Bambaren
Gorav Gupta
Joshua Lugo
Daniel Muniz

Interconference Parity in the NBA

Introduction and Background

Currently, the NBA is composed of two conferences, East and West, with 15 teams in each. The East and West teams play against each other a set number of times per season. By summing up the wins and losses, we can obtain the interconference record for that season. Arguably, the league exhibits parity between conferences if this win-loss record is close to even.

This parity is something that the NBA should care about, because if one conference is significantly weaker than another, then it'll create less exciting games. This affects the ratings of their regular season and postseason games, hurting the league's current and future revenue. In fact, this season the NBA's TV ratings on ESPN and TNT have dropped from last season, 6% and 26% respectively, with reporters speculating that it's due to the player movement of LeBron James, who, after playing for the Eastern conference for 15 years, was signed by the Los Angeles Lakers, a West team. Parity is also important for the communal and historical discussion of basketball; comparison across eras is tricky when the parity is very uneven.

But LeBron isn't the only player who switched conferences. We're interested in how the total player movement shapes the two conferences and affects their inter-conference record. If our findings are significant, the NBA might want to modify rules about player movement to encourage more parity between conferences.

Project Methodology

Our goals are to create a predictive model based on the player movement between conferences, and to accurately quantify the aggregate impact of all player movement on the interconference record. In order to measure how much player movement impacts a conference's strength, we took a look at 4 different advanced statistics. These stats are important to general managers as well as NBA coaches, as they each try to capture a player's impact on a game in a different way.

These are:

- Player Efficiency Rating (PER): a measure of a player's per-minute production. It is standardized such that the league average is equivalent to 15.
- Box Plus/Minus (BPM): A box score estimate of the points a player contributed above a league-average player per 100 possessions. This is translated to an average team.

- Value Over Replacement Player (VORP): A box score estimate of the points that a player contributes above a “replacement-level” (-2.0) player per 100 team possessions. This is translated to an average team and prorated to an 82-game season.
- Win Shares (WS): This is an estimate of the number of wins that were contributed by a player.

We summed the previous season’s stats for all the players moving to a new conference. We then compared these aggregated stats to the interconference record of the next season. As an example, to examine the impact of new players on the 1977-1978 season, we find the aggregate stats of the 1976-1977 for those players, and compare them to the interconference record for the 1977-1978 season.

We wish to see how much effect this local variance has on season to season record compared to more global trends. Some of these global trends included the players skill growth and decay throughout their career, any potential injuries as some of these could derail their skill growth or even end their career, the draft picks for a team, and team chemistry. We took a look at the last 41 seasons of the NBA starting with the ABA/NBA merger in the 1976-1977 season and ending with the 2017-2018 season (last season).

With all this in mind, our primary aim was to determine the extent to which player movement immediately affected interconference record, using both Time Series Analysis with ARIMA, and Linear Regression with the Ordinary Least Squares Method. Our secondary aim was to build a predictive model that could estimate the future interconference record, also using Time Series Analysis with ARIMA.

Data Pre-Processing

When it came to this project, the data pre-processing stage was very critical for our analysis and regression model to be successful. Because we were interested in finding out the correlation between player movement from one conference to the other and interconference parity, we knew we would need the list of players that played for each conference per season, the players that changed conferences, the summary statistics for the season, and the win-loss record of each conference against the other. We would need this data from the 1976-1977 season (when the ABA and NBA merged), all the way to the 2017-2018 season (a total of 41 seasons). With this in mind, we were able to find all of our raw data from basketball-reference.com.

In this website, we were able to successfully find the advanced statistics per player (PER, BPM, VORP, WS) that are required in our analysis, the full list of teams and their records during each season, and the full list of every single player, including every team they played with during each season. Using these, we would be able to meet our pre-processing goals, and have the necessary components for our analysis.

Classifying Players into Conferences

Unfortunately, the player data-sets for basketball-reference.com did not include which conference each player belonged to during a season. In order to approach this problem, we used the “Tm” (Team) attribute to determine which conference a player belonged to. If the team played in the eastern conference, then all the players with that team attribute automatically belonged in the eastern conference and vice versa. However, throughout the seasons, it was not a rare occurrence for teams to relocate with a name change to the other conference or to another city in the same conference. Being cautious of this was important in order to not misclassify the players into the wrong conference. The season’s where a team changed conference were as follows: 79-80, 80-81, 81-82, 88-89, 89-90, 90-91, 91-92, 95-96, 97-98, 02-03, 04-05, 12-13, 14-15.

To pull this off first we downloaded and stored all of the player statistics into csv files for all of the 41 seasons we were taking a look at. In python, using the numpy library, we added all of the eastern team abbreviations as conditions, and added “East” as a choice for all of the eastern teams (Figure 1). If a player had a “TOT” team abbreviation, that meant that row was the player’s cumulative statistics for the season, so we gave “NaN” as a choice. If the player’s “Tm” attribute was not in the conditions, then they were automatically defaulted as belonging to the “West”. The conditions were edited based on which seasons we were classifying; as such, we used multiple python files.

When running the files, we would store the csv containing all of the players and their statistics into a DataFrame, and then that DataFrame would be manipulated with

numpy to create the new “Conference” attribute. Finally each updated DataFrame would be stored into a new csv file for further analysis and pre-processing.

Figure 1

```
#02-04 Player Conference Classifier
import numpy as np
import pandas as pd
```

```
#Storing 03-04 Player Statistics into a dataframe for manipulation
df = pd.read_csv("C:\\Users\\Giancarlo\\Desktop\\Intro to Data Science Player Ranks\\Player Ranks No Conferences\
```

```
#Replacing CHH to NOH due to relocation.
conditions = [
    (df['Tm'] == 'TOT'), (df['Tm'] == 'PHI'), (df['Tm'] == 'BOS'), (df['Tm'] == 'NYK'), (df['Tm'] == 'BUF'),
    (df['Tm'] == 'NYN'), (df['Tm'] == 'MIL'), (df['Tm'] == 'WAS'), (df['Tm'] == 'CHI'), (df['Tm'] == 'CLE'),
    (df['Tm'] == 'IND'), (df['Tm'] == 'ATL'), (df['Tm'] == 'NJN'), (df['Tm'] == 'DET'), (df['Tm'] == 'MIA'),
    (df['Tm'] == 'NOH'), (df['Tm'] == 'ORL'), (df['Tm'] == 'TOR')]

choices = ['NaN', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'East',
         'East', 'East', 'East', 'East', 'East']
```

```
df['Conference'] = np.select(conditions, choices, default = 'West')
```

Listing the Players that Switched Conferences

Our next stage in pre-processing the data was finding which players had switched conferences from one season to the next. Using python once again, we now compared two consecutive seasons in order to find which players had switched from one conference to the other. One issue we had to figure out was those players who had the “TOT” attribute under their team. Since that meant that player had played for multiple teams within the same season (sometimes for two different conferences), we needed to decide on an attribute that would determine which team we would count that player for. What we noticed was that a lot of these players who had played for multiple teams, often played a very small amount for one team, and a much larger one for the other team, so we decided to go by the “G” attribute (number of games played). We would count that player as a member of the team he had played the most games for.

When it came to the python program, we ran a script (Figure 2) that would pull the first player csv file for the first season, the second player csv file for the second consecutive season and these would then be stored into DataFrames. It then eliminated all rows with the “Tm” attribute equal to “TOT”. The rows would then be sorted in descending order by the player Name (“Player” attribute) and Games (“G” attribute). Then a loop was used in order to pick the first entry for each player (i.e. with largest games played) and stored each player in a new DataFrame. This eliminated the

duplicate players from the datasets and allowed us to find out which players had changed conferences in between the seasons.

To find which players moved from one conference to the other, we found the intersection from the first season DataFrame to the second season DataFrame based on the “Conference” attribute of each player (Figure 3). So if a player belonged to the “West” class under first season, and in the second season belonged to the “East” class, that player had changed conferences from “West” to “East” in between the two seasons. Similar approach was applied to find the players that moved from “East” to “West”.

Finally this was done for each csv file and we successfully approximated which players had switched conferences in between each season (Figure 4).

Figure 2

```
def onesession (file_name):
 a = 'C:\\\\Users\\\\goravg\\\\Downloads\\\\Player Ranks with Conferences\\\\'
 b = file_name
 inputfile = a + b
 df = pd.read_csv(inputfile)
 df = df.drop(df[df['Tm'] == 'TOT'].index)
 df = df.sort_values(by=['Player','G'], ascending=False)
 df = df.loc[:,['Player','G','Conference']]
 intermediate = []
 finallist = []
 location = 0
 for num,x in enumerate(df['Player']):
     value = num + 1
     try:
         intermediate.index(x)

     except ValueError:
         intermediate.insert(location,x)
         finallist.insert(location,[x,df['G'].iloc[num],df['Conference'].iloc[num]])
         location += 1
     else:
         continue

 df = pd.DataFrame(finallist,columns = ['Player','G','Conference'])


 West = set(df[df['Conference'] == 'West']['Player'])
 East = set(df[df['Conference'] == 'East']['Player'])
 return (West, East)
```

Figure 3

```
def dumplist(file1,file2):
 a = str(file1) + '_Player_Conferences.csv'
 b = str(file2) + '_Player_Conferences.csv'
 Westp, Eastp = onesession(a)
 Westc, Eastc = onesession(b)

 Move_West_to_East = Westp.intersection(Eastc)
 Move_East_to_West = Eastp.intersection(Westc)

 listing = []
 for values in Move_West_to_East:
     #print (values)
     listing.append([values, 'East'])
 #print(listing)
 data = pd.DataFrame(listing,columns = ['Values','Moved_to'])

 for values in Move_East_to_West:
     #print (values)
     listing.append([values,'West'])
 data = pd.DataFrame(listing,columns = ['Values','Moved_to'])
 target_file = str('C:\\\\Users\\\\goravg\\\\Downloads\\\\Output\\\\') + str(file1) + '-TO-' + str(file2) + '.csv'
 data.to_csv(target_file, index = False)

dumplist('76-77','77-78')
```

Figure 4

	A	B
1	Values	Moved_to
2	Matt Bullard\bullama01	East
3	Carlos Rogers\rogerca01	East
4	Jaren Jackson\jacksja01	East
5	Shandon Anderson\andersh01	East
6	Howard Eisley\eisleho01	East
7	Greg Foster\fostegr01	East
8	Damon Jones\jonesda01	East
9	Othella Harrington\harriot01	East
10	Corie Blount\blounco01	East
11	Tyronn Lue\luety01	East
12	Vonteego Cummings\cummivo01	East
13	Patrick Ewing*\ewingpa01	East
14	Christian Laettner\laettch01	East
15	Hakeem Olajuwon*\olajuha01	East

Getting Summary Data for Season:

In the previous steps we created two files

- Player stats for each season having their corresponding conference names
- List of players switched conferences between two seasons.

Now, the next step was to identify the stats related to player movement i.e. find how much was the contribution of moved player to their respective teams (before movement), measured by the following attributes - 'PER', 'WS', 'OBPM', 'DBPM', 'BPM', 'VORP' .

When it came to the python program, we ran a script that would combine the above mentioned two files and pick statistics for only players changed conferences. Then, with the help of pivot-table, we obtained the aggregated impact of player movement (Figure 5).

Figure 5

```
def looping(file_name_1,file_name_2,input_year,final,ind):
 a = 'C:\\Users\\gorav\\Downloads\\Adv Stats Conference\\'
 b = file_name_1
 inputfile = a + b

 stats_data = pd.read_csv(inputfile)
 picked_stats = pd.DataFrame(stats_data, columns = ['Player', 'PER',
                                                    'WS', 'OBPM', 'DBPM',
                                                   'BPM', 'VORP'])

 a = 'C:\\Users\\gorav\\Downloads\\Player Conference Changes\\'
 b = file_name_2
 inputfile = a + b

 players=pd.read_csv(inputfile)

 players.rename(columns={'Values': 'Player'}, inplace=True)


 third=pd.merge(picked_stats, players, on='Player', how='inner')


 third['PER']=np.subtract(third["PER"], 15)


 conf_names=['West', 'East']
 DataFrameDict = {elem : pd.DataFrame for elem in conf_names}

 for key in DataFrameDict.keys():
     DataFrameDict[key] = third[:][third.Moved_to == key]
 west=DataFrameDict['West']
 east=DataFrameDict['East']

 sum_data_East = third[third['Moved_to'] == 'East'].pivot_table(values=['PER', 'WS', 'OBPM', 'DBPM', 'BPM', 'VORP'], index=['Mov
ed_to'],
                            aggfunc='sum')

 sum_data_West = third[third['Moved_to'] == 'West'].pivot_table(values=['PER', 'WS', 'OBPM', 'DBPM', 'BPM', 'VORP'], index=['Mov
ed_to'],
                            aggfunc='sum')

 year =  input_year
 sum_data_West ['Year'] = year
 sum_data_East ['Year'] = year
 sum_data_West ['Moved_to'] = 'West'
 sum_data_East ['Moved_to'] = 'East'

 if ind == 'first':
    final=pd.concat([sum_data_West, sum_data_East],axis=0,sort=False)
 else:
    final=pd.concat([sum_data_West, sum_data_East,final],axis=0,sort=False)

 return (final)
```

This step generated two output files- one for each conference. Within each file, it had deltas of statistics impacted for player movement for each conference (Figure 6).

Figure 6

A	B	C	D	E	F	G	H
BPM	DBPM	OBPM	PER	VORP	WS	Year	Moved_to
-141.6	-56.1	-85.8	-165.6	8.5	75	2018	West
-79.1	-54.5	-24.5	-68.9	5.8	73.9	2017	West
-136.7	-84.9	-51.5	-133.7	19.7	125.9	2016	West
-117.6	-23.4	-93.9	-182.8	22.4	93.5	2015	West
-183.1	-74.3	-108.9	-166.5	18.8	93	2014	West
-62.2	30.2	-92.2	-134.8	28.4	105	2013	West
-101	-43	-58.3	-129.3	24.9	105.5	2012	West
-206.6	-40.4	-166.4	-262.1	8.8	109.1	2011	West
-136.9	-47.7	-89.6	-121.9	15.2	97.9	2010	West
-138.9	-21.8	-117.4	-193.4	18.1	88.9	2009	West
-42	-18	-24.2	-40.1	13.1	66.1	2008	West
-98.8	-22.3	-76.6	-117	13.7	90.2	2007	West
-92.8	-11.7	-81.4	-143.3	22.7	107.4	2006	West
-209.1	-37.6	-171.2	-230.4	15	101	2005	West
-156.6	-41.9	-114.9	-167.9	10.6	86.3	2004	West
-55.7	-19	-36.8	-68.5	10	61.5	2003	West
-69.8	-21.3	-48.3	-94.4	17.9	96.4	2002	West
-145.3	-40.8	-104.3	-135.9	0.6	58.9	2001	West
-138	-49.4	-88.3	-118.7	1.4	40.3	2000	West
-126.2	-46.2	-80.2	-75.7	12.4	97.7	1999	West
-181.8	-50.4	-131	-175.1	16.3	117.7	1998	West
-130.4	-30.6	-99.6	-155.1	28.6	131.1	1997	West
-129	-38.8	-90.2	-122.6	7	58.3	1996	West
-80.2	-18.7	-61.7	-81.6	13.3	65.9	1995	West
-57.6	-10.7	-46.6	-60.1	6.7	56	1994	West
-99.6	-28.1	-71.3	-105.1	13	74.7	1993	West
-89.1	-37.5	-51.8	-86.5	9.2	64.9	1992	West
-134.8	-16.3	-118.4	-153.8	6.7	56.9	1991	West
-79.5	-4.9	-75.1	-91.8	9	73.6	1990	West
-189.3	-49.8	-139.8	-205.8	3.1	68	1989	West

Calculating Interconference Win-Loss Record

Our last step was to calculate the interconference win-loss record. Basketball-reference.com provides the seasonal record for each team. One of the attributes in their dataset is the respective team's record again the Eastern or Western conference (Figure 7). Using this, we were able to easily find the interconference win-loss record. For each team in the Eastern conference, we took their win-loss record against Western teams, and aggregated them, thus giving us the Eastern conference's win-loss record against the West. Since interconference record is zero sum, we can find the West's record by flipping the win-loss record.

Figure 7

			Place		Conference		Division						All-Star		Margin		Month					
Rk	Team	Overall	Home	Road	E	W ▼	A	C	SE	NW	P	SW	Pre	Post	≤3	≥10	Nov	Dec	Jan	Feb	Mar	Apr
1	Detroit Pistons	54-28	32-9	22-19	35-17	19-11	14-4	8-8	13-5	5-5	8-2	6-4	32-19	22-9	8-8	26-14	7-7	8-6	11-5	9-1	9-8	10-1
2	Miami Heat	59-23	35-6	24-17	41-11	18-12	14-4	12-6	15-1	5-5	8-2	5-5	40-14	19-9	7-8	33-6	10-6	14-1	9-6	9-3	12-3	5-4
3	Washington Wizards	45-37	29-12	16-25	28-24	17-13	9-9	9-9	10-6	7-3	5-5	5-5	30-22	15-15	14-6	16-16	7-5	8-8	11-4	5-7	9-6	5-7
4	Cleveland Cavaliers	42-40	29-12	13-28	26-26	16-14	7-11	7-9	12-6	6-4	5-5	5-5	30-21	12-19	4-5	27-20	9-5	8-7	9-5	5-7	6-9	5-7
5	Boston Celtics	45-37	27-14	18-23	30-22	15-15	8-8	8-10	14-4	6-4	5-5	4-6	27-26	18-11	11-7	16-14	5-8	8-8	8-8	8-4	9-5	7-4
6	Chicago Bulls	47-35	27-14	20-21	32-20	15-15	11-7	8-8	13-5	8-2	3-7	4-6	26-23	21-12	7-5	15-15	1-10	8-7	13-3	7-4	11-7	7-4
7	Indiana Pacers	44-38	25-16	19-22	29-23	15-15	8-10	9-7	12-6	6-4	3-7	6-4	25-26	19-12	9-9	18-12	10-4	5-9	5-10	8-4	9-7	7-4
8	Orlando Magic	36-46	24-17	12-29	21-31	15-15	9-9	6-12	6-10	6-4	4-6	5-5	28-24	8-22	10-5	10-22	8-5	7-7	9-8	4-7	6-10	2-9
9	Toronto Raptors	33-49	22-19	11-30	20-32	13-17	7-9	5-13	8-10	5-5	4-6	4-6	21-32	12-17	6-9	16-21	7-9	3-12	8-6	6-5	5-10	4-7
10	New York Knicks	33-49	22-19	11-30	21-31	12-18	6-10	6-12	9-9	5-5	3-7	4-6	21-32	12-17	9-11	13-17	7-6	9-7	2-13	6-7	5-8	4-8
11	New Jersey Nets	42-40	24-17	18-23	31-21	11-19	11-5	11-7	9-9	4-6	5-5	2-8	23-30	19-10	7-4	19-26	3-11	7-7	8-8	7-6	9-6	8-2
12	Philadelphia 76ers	43-39	25-16	18-23	33-19	10-20	8-8	13-5	12-6	4-6	4-6	2-8	26-27	17-12	10-4	13-15	6-7	7-8	8-8	5-6	9-7	8-3
13	Charlotte Bobcats	18-64	14-27	4-37	11-41	7-23	2-16	2-16	7-9	3-7	1-9	3-7	11-39	7-25	5-16	6-29	3-10	4-9	2-13	2-10	4-13	3-9
14	Milwaukee Bucks	30-52	23-18	7-34	24-28	6-24	7-11	8-8	9-9	3-7	1-9	2-8	20-30	10-22	8-8	10-21	4-8	5-9	6-10	8-4	4-13	3-8
15	Atlanta Hawks	13-69	9-32	4-37	8-44	5-25	4-14	2-16	2-14	1-9	1-9	3-7	10-41	3-28	2-8	3-43	2-12	3-11	4-11	1-11	1-15	2-9

Lastly, we found the difference between wins and losses. This differential is what we'll use in our analysis. Again, since the record is zero sum, much of our analysis for one conference will be symmetrical but flipped along the x-axis. Putting this into a .csv file with the summed stats (Figure 8), we're ready to being the analysis.

Figure 8

BPM	DBPM	OBPM	PER	VORP	WS	Year	Moved_to	West +/-
-78.1	-13.3	-65.2	-50.9	29.1	104.5	2018	West	24
-40.3	1.5	-41.8	-46.1	18.6	85.3	2017	West	42
-123.2	-31.8	-91.5	-151.8	12.1	104.8	2016	West	14
-117.4	-30.3	-87.5	-138.3	10	89.6	2015	West	76
-132.2	-51.8	-79.9	-130.4	11.8	95.8	2014	West	118
-69.6	33.2	-102.7	-157.3	21.4	87.1	2013	West	74
-150.5	-53.8	-96.5	-158.1	11.3	101.5	2012	West	42
-204.4	-35.2	-168.3	-228	17.1	130.8	2011	West	72
-180.4	-37.5	-143.1	-220.1	9.3	84	2010	West	42
-103	-10.1	-93.3	-140.6	16.6	90.4	2009	West	-12

Data Breakdown

Overall, the data pre-processing stage was very intensive. The raw data alone is 3.27 MB, split across 41 different .csv files. These 41 files cover the 41 seasons from 1976-1977 through 2017-2018. The number of rows varies from season to season, and is greater than the total roster size for that season. This is due to the fact that if a player plays for more than one team in a season, he gets a row entry for each team.

Earlier seasons tend to have less rows than later seasons. I won't list every single season, but as a rough estimate, the earliest season (1976-1977) has 296 rows, while the latest season (2017-2018) has 665 rows.

All raw data can be found on basketball-reference.com. There is a site option to display the data as csv, which allows for easy transfer to a .csv file through copy-paste.

Experiments

The machine used to run our 2 experiments was configured to run on Windows 7 64-bit. Experiments were run in Python 3.7, using Jupyter notebook run through Anaconda. For both experiments, the size of the outputs, and time taken to run scripts are negligible.

Aim 1: Time Series Analysis with ARIMA

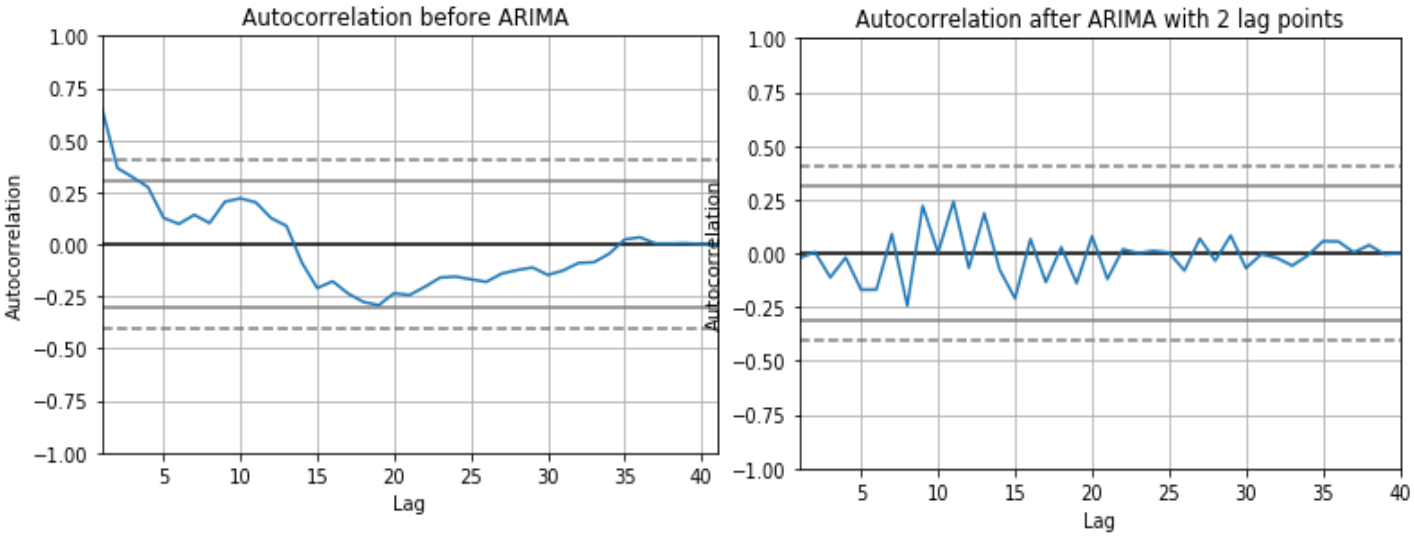
Our first experiment was to analyze the differential of the interconference record, using Time Series analysis with ARIMA in Python. We wished to examine which previous season was most relevant to predicting future interconference record, as well as attempting to build a predictive model. It's useful to note that since we're looking at the differential of the record between conferences, the total +/- for each conference is 0 sum. This means the graphs are symmetrical, and as such the results of our analysis will be identical (and flipped as necessary).

We imported pandas and autocorrelation_plot from pandas.tools.plotting 0.23.4, numpy 1.15.1, ARIMA from statsmodels.tsa.arima_model 0.9.0, matplotlib.pyplot 2.2.3, and mean_squared_error from sklearn.metrics 0.19.2.

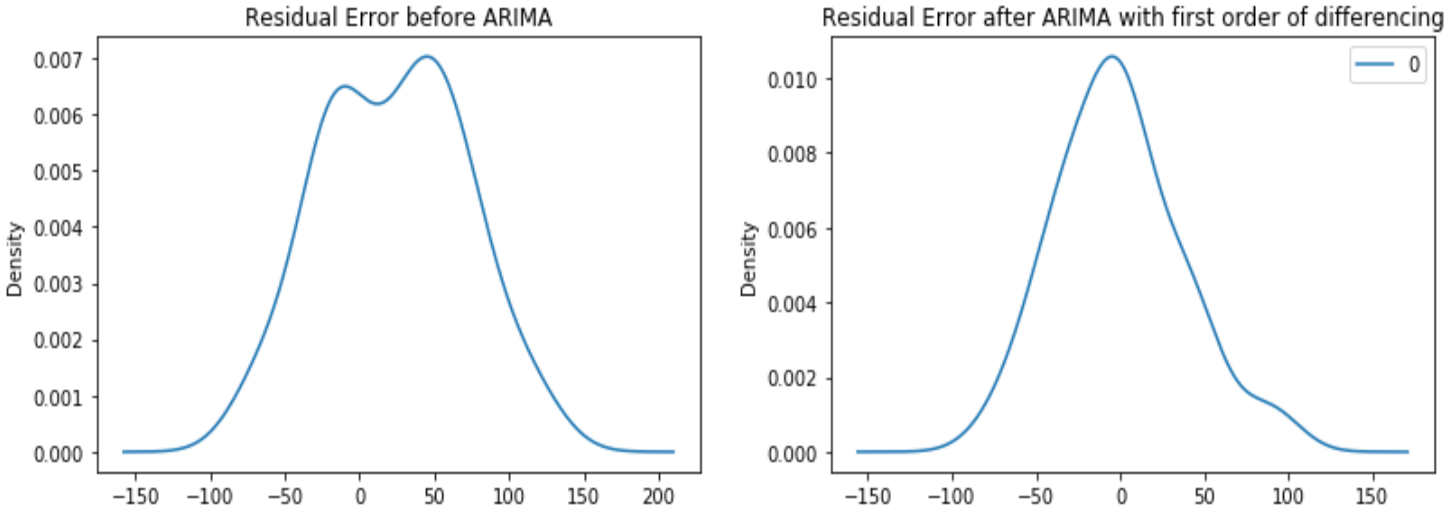
After analysis of the autocorrelation and residuals plot, we set order = (2, 1, 0), with no constant term added(trend = 'nc'). Further differencing did not improve the mean of the residuals, and no other significant lag points appeared in our autocorrelation plot. For our predictive model, we used ⅔ of the seasons for the training set, and ⅓ for the testing set.

Our results are as follows. The ARIMA model showed that the second lag point (i.e. 2 seasons back) is statistically significant in predicting in future season record. For this second lag point, p-value = 0.026 and std error = 0.149. For the first lag point, p-value = 0.424 and std error = 0.147. The residual error for this model has a mean of -0.8428, suggesting that further differcing will not improve the scores much. For our predictive model, the MSE is 1376.758.

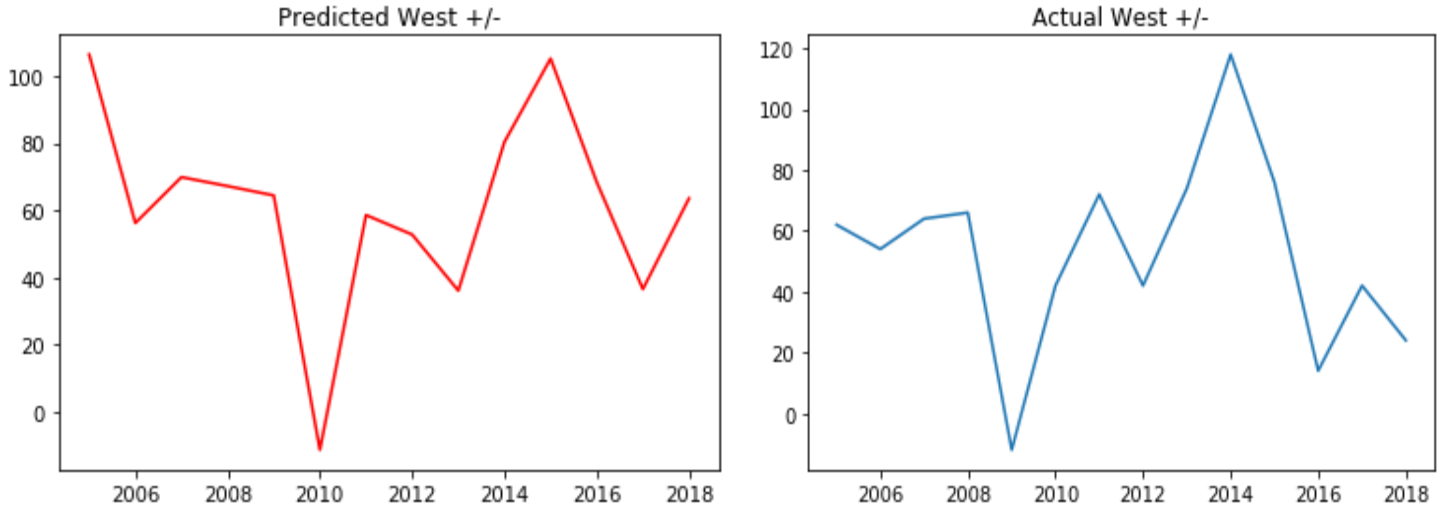
Autocorrelation Plots



Residual Error Plots



Predictive Model Plots



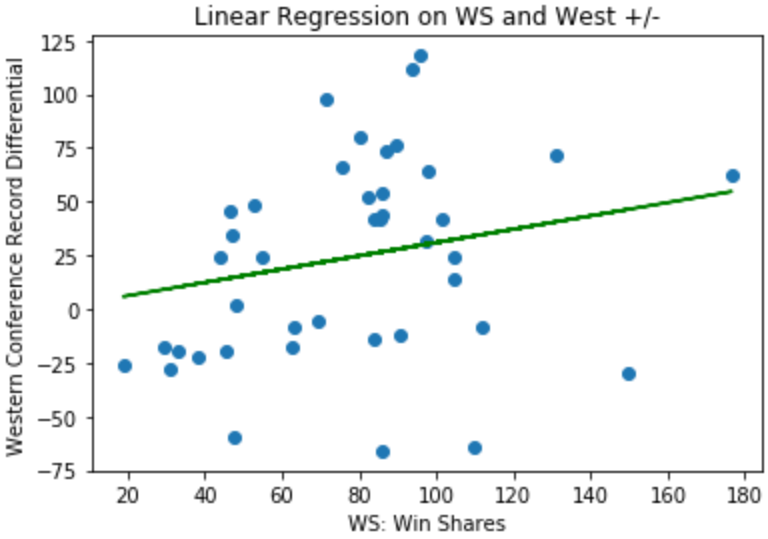
Aim 2: Linear Regression Analysis

Our second experiment was to analyze the relationship between the interconference record and various advanced player statistics, using Ordinary Least Squares (OLS). The player statistics are summed across all players who changed conference in a given season, and are paired with the interconference record of that season. It's important to note that the statistics are taken from the season before the player switched conferences. While interconference record is zero sum, these stats are not. So it's possible for the analysis to be different for each conference.

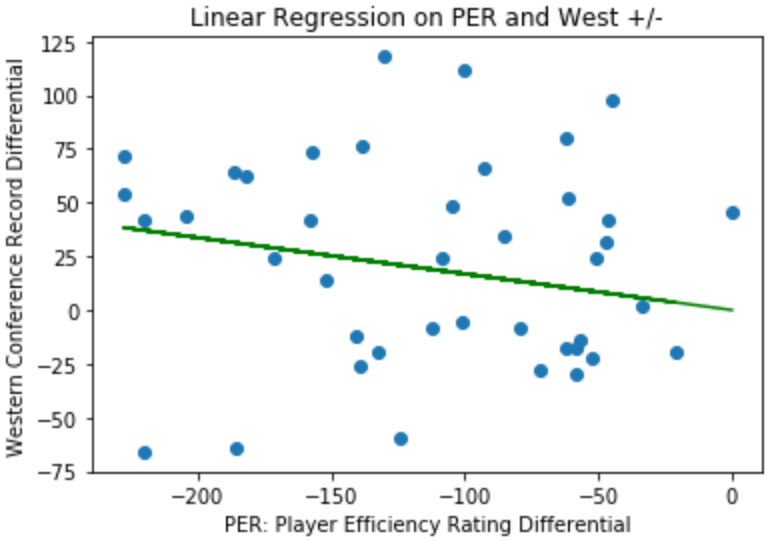
We imported pandas 0.23.4, numpy 1.15.1, matplotlib.pyplot 2.2.3, LinearRegression from sklearn.linear_model 0.19.2., and statsmodels.api 0.9.0.

The statistics we will look at to compare with the record are ‘PER’, ‘WS’, ‘VORP’, and ‘BPM’. We will examine both each stat individually, and multiple stats together. For our Linear Regression, we almost always found better results without adding a constant.

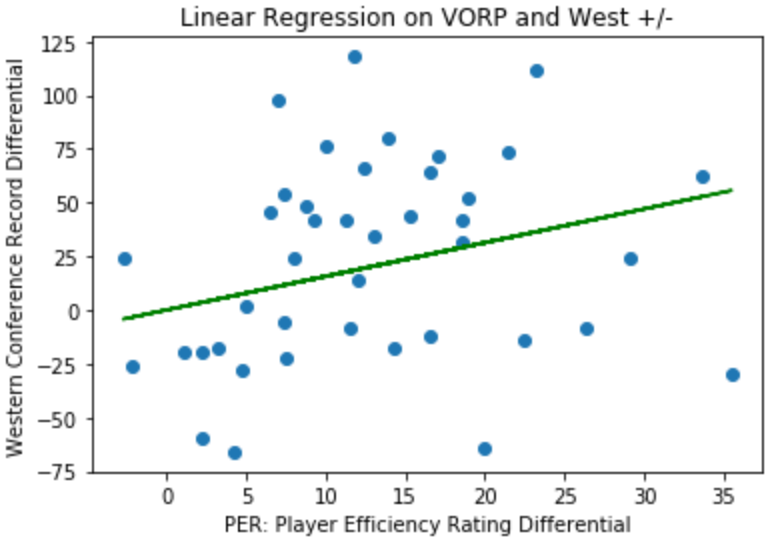
The Linear Regression model found that WS was the best predictor, accounting for 25.6% of the variance. It has a p-value = 0.001, and a std err = 0.083.



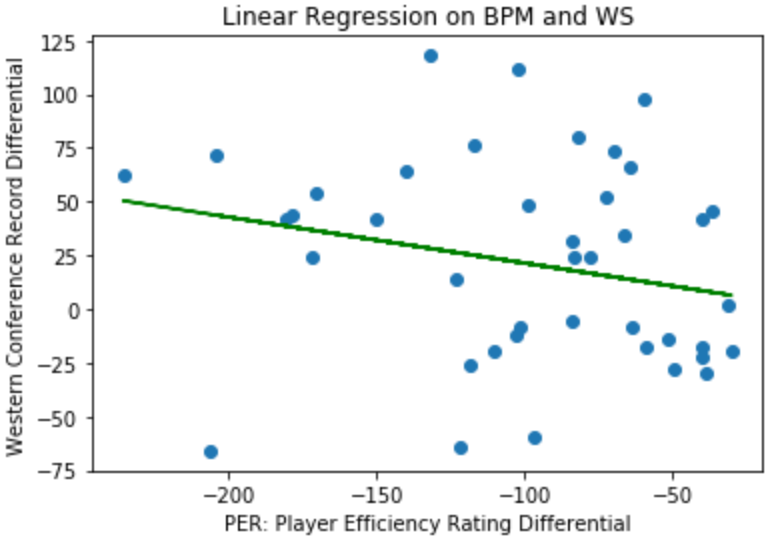
PER accounted for 17.5% of the variance with a p-value = 0.006 and std err = 0.058.



VORP accounted for 22.3% of the variance, with p-value = 0.002 and std err = 0.463.



BPM accounted for 21.8% of the data, with p-value = 0.002 and std err = 0.064



Generally, the multiple statistic regression ended up performing worse. Either one or both of the statistics would show to be statistically insignificant (p-value greater than 0.05).

Discussion & Conclusion

Aim One: Time Series Analysis with ARIMA

First, we were able to create a predictive model using ARIMA. We obtained a MSE of 1376.758; although this is most useful as a comparison to other models, this does seem a little high. Looking at the graph, we can see that the general trends are captured (which includes very big swings both up and down), but the model seems to be off by a year. Perhaps with more tuning we will be able to get the years lined up correctly.

Second, we found that the +/- record (interconference record) from 2 seasons back is the most relevant for predicting the current season +/- record. This suggests that interconference record is more dependent on the global trends noted before (injuries, team chemistry, player growth and decay), and less on the local variance of player movement. Nonetheless, it's important to see how much of the variance we can explain with player movement.

Unfortunately, the ARIMA model recommends at least 50 data points for optimal test conditions. Due to the nature of the league at the time, the best season to start analysis is after the NBA-ABA merger in 1976. This left us with only 41 data points. Presumably, the model will be more accurate in about 10 years or so.

Aim Two: Linear Regression Analysis

For our second experiment, we found that WS was the best predictor, being able to account for 25.6% of the variance in the +/- record. However, not only is this somewhat low, it does not perform significantly better than the other statistics, except for maybe PER.

First of all, this suggests that WS, VORP, and BPM are all somewhat equally effective in judging a player's game impact. Secondly, this is in line with the conclusion that the +/- record (and presumably other factors) from two seasons back is more important for current interconference parity. Teams that improve due so for a multitude of factors: their players improve over time (or their veterans don't regress as much due to age); the players become more familiar with their coach's systems for defense and offense; teams draft good young talent; and teams stay healthy through the 82 game season.

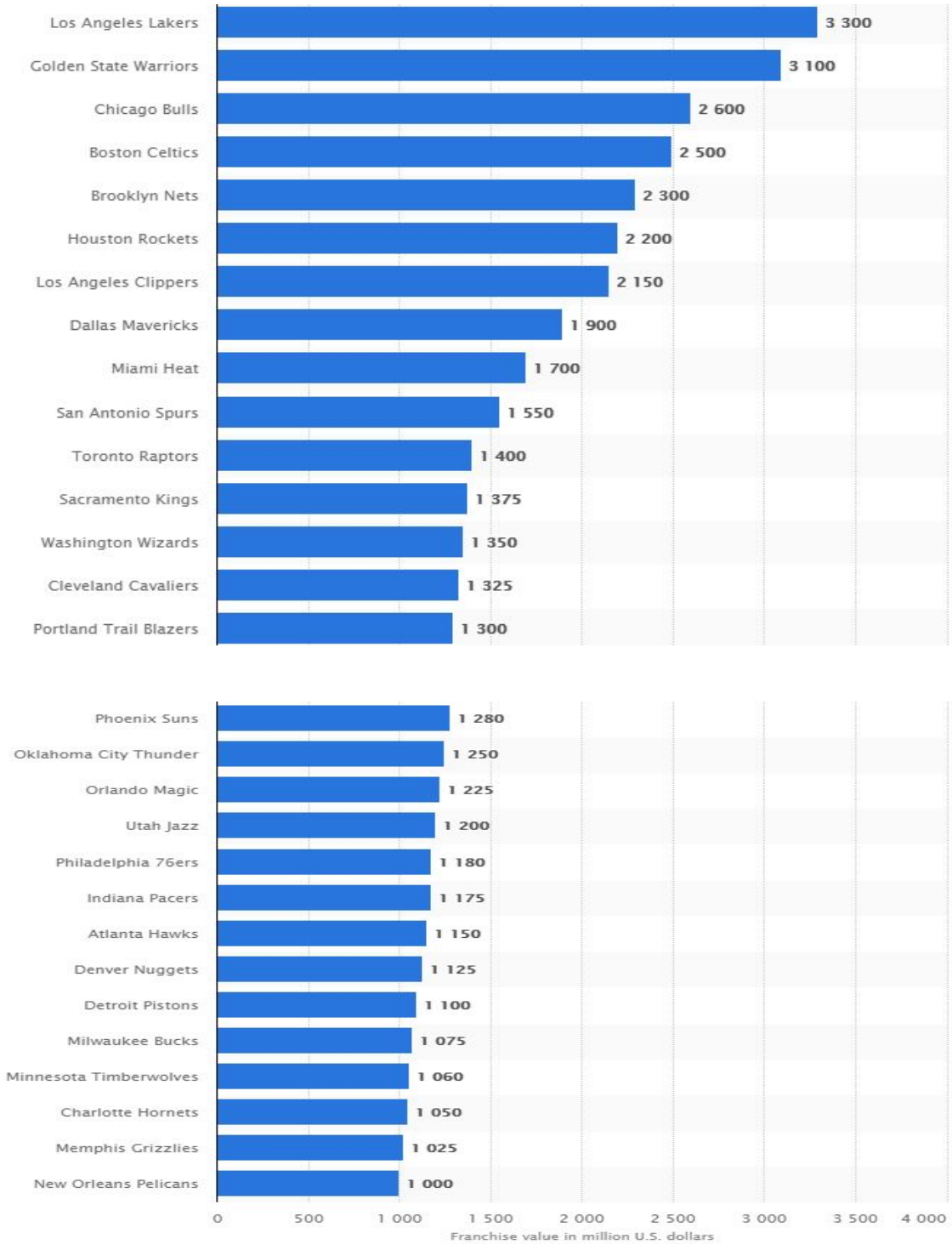
Final Comments and Future Improvements

An improvement to help limit any noise from this data set would be to limit the data frame to players that played at least five minutes per game. This would allow us to only look at the advanced stats for players who were able to contribute on the court.

A factor that affects player movement that merits analysis is the combination of market size and financial constraints. Figure 9 (source) depicts the franchise value of all 30 NBA teams in 2018. Players like to play in bigger markets, and these are considered prime free agent

destinations. Along with market size, all teams have to play by the NBA's salary cap rules. The NBA's choice of a soft cap with luxury tax means that some teams can choose to pay their players more as long as they're willing to pay a penalty fee for going over the cap. In fact, most teams tend to go over this cap at least once every few seasons, but it could be interesting to see if one conference spends significantly more on players than the other conference, and the effect this might have on parity. (https://www.basketball-reference.com/contracts/).

Figure 9



To conclude, our project had some limitations and assumptions required in order to properly analyze the data. However, we were still able to predict the trends in the interconference record and verified that ultimately record is determined by more than just player movement. Given more time, we could implement some of the suggestions mentioned in this section. In the future, it may also be interesting to add data on “Games Played” to the data frame to see if the players that moved were healthy/relevant enough to meaningfully contribute during the season, as well as to look only at above average player movement instead of all player movement.