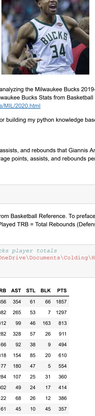
Analysis of the Milwaukee Bucks 2019-20 season and the impact of Giannis By Joe Sliwinski



For my first basketball python project, I will dive into analyzing the Milwaukee Bucks 2019-20 season along with Giannis Antetokounmpo's impact on the season. I will be using the 2019-20 Milwaukee Bucks Stats from Basketball Reference, specifically the Totals table. You can find this: https://www.basketball-reference.com/teams/MIL/2020.html

The analyzation will be simple to lay the foundation for building my python knowledge base and skillset. It will include data strictly from the 2019-20 season. The steps include:

1. computing the average age of the Bucks roster
2. calculating the percentage of the team's points, assists, and rebounds that Giannis Antetokounmpo accounted for
3. creating a new data frame consisting of the average points, assists, and rebounds per game played of everyone on the roster

For this analysis, I will need the following packages:

In [1]:
```
import pandas as pd
import numpy as np
```

My first step will be to load the data that I extracted from Basketball Reference. To preface column header meanings are as follow: GP = Games Played GS = Games Started MP = Minutes Played TRB = Total Rebounds (Defensive and Offensive) AST = Assists STL = Steals BLK = Blocks PTS = Points

In [2]:
```
#importing csv of 2019-20 Milwaukee Bucks player totals
bucks = pd.read_csv (r'C:\Users\sliwi\OneDrive\Documents\Coluding\NoopaData\Projects\bucks2020season\Bucks 2019-20 totals.csv')
bucks
```

Out[2]:

	Player	Age	GP	GS	MP	TRB	AST	STL	BLK	PTS
0	Giannis Antetokounmpo	25	63	63	1917	856	354	61	66	1857
1	Khris Middleton	28	62	59	1853	382	265	53	7	1297
2	Brook Lopez	31	68	67	1817	312	99	46	163	813
3	Eric Bledsoe	30	61	61	1646	282	328	57	26	911
4	Wesley Matthews	33	67	67	1635	166	92	38	9	494
5	Donte DiVincenzo	23	66	24	1520	318	154	85	20	610
6	George Hill	33	59	2	1271	177	180	47	5	554
7	Pat Connaughton	27	67	4	1243	284	107	25	31	360
8	Ersan Ilyasova	32	63	8	986	302	49	24	17	414
9	Kyle Korver	38	58	1	960	122	68	26	12	386
10	Robin Lopez	31	66	5	958	161	45	10	45	357
11	Sterling Brown	24	52	1	767	183	51	29	7	266
12	D.J. Wilson	23	37	1	363	92	25	3	5	133
13	Marvin Williams	33	17	0	321	74	19	10	8	68
14	Thanasis Antetokounmpo	27	20	2	129	24	15	7	2	55
15	Frank Mason	25	9	0	118	19	29	5	1	62
16	Dragan Bender	22	7	0	91	20	9	0	5	26

After pulling in the raw data, the first data segment that stands out is age. With the Bucks being in title contention, having veterans on their roster is key. I suspect that the age of the players will be relatively old. I will now find out the Bucks average age of their roster.

In [3]:
```
# average age of Bucks players rounded to the nearest tenth
averageage = bucks['Age'].mean()
round(averageage, 1)
```

Out[3]: 28.5

An average age of 28.5 seems relatively old for a roster. To put that into perspective, this would be in the top 5 oldest rosters in the NBA during the 2019-20 season. The Houston Rockets had the oldest roster averaging an age of 30.2 per player, while the Phoenix Suns were the youngest at 24.4 per Lineups.com/nbarosters.

My next step is to find the total points, assists, and rebounds accumulated by the roster. This will help me with my final analysis of finding the percentage of points, assists, and rebounds that Giannis accounted for.

In [4]:
```
pts_total = bucks['PTS'].sum()
ast_total = bucks['AST'].sum()
trb_total = bucks['TRB'].sum()
```

In [5]:
```
# Bucks point total for 2019-20 season
pts_total
```

Out[5]: 8663

In [6]:
```
# Bucks assist total for 2019-20 season
ast_total
```

Out[6]: 1889

In [7]:
```
# Bucks rebound total for 2019-20 season
trb_total
```

Out[7]: 3774

Now that I have the total points, assists, and rebounds, I will dive into finding the impact Giannis had on this team. Below you will see just data from Giannis. If you did not notice, Giannis led the team in total points, assists and rebounds, quite an accomplishment and fitting for his unicorn status.

In [8]:
```
# show 1st row of data frame (Giannis)
bucks.iloc[[1]]
```

Out[8]:

	Player	Age	GP	GS	MP	TRB	AST	STL	BLK	PTS
0	Giannis Antetokounmpo	25	63	63	1917	856	354	61	66	1857

I can now use the total points, assists, and rebounds from the team that I calculated earlier. The first equation will be for finding the percent of the Bucks points that Giannis scored.

In [9]:
```
# percent of Bucks 2019-20 total points scored from Giannis
round(1857/8663*100, 2)
```

Out[9]: 21.44

Next is finding the percent of Bucks assists that Giannis accounted for.

In [10]:
```
# percent of Bucks 2019-20 total assists from Giannis
round(354/1889*100, 2)
```

Out[10]: 18.74

Lastly, I will calculate the percent of Bucks rebounds that Giannis had.

In [11]:
```
# percent of Bucks 2019-20 total rebounds from Giannis
round(856/3774*100, 2)
```

Out[11]: 22.68

Without diving into how Giannis ranks across the league in these categories, it is simply impressive that he led his team in all three categories.

For Giannis not being a shooter, he led the team in points showing his dominance in the paint.

For Giannis not being the team's point guard, he led the team in assists showing his playmaking ability.

For Giannis not being the team's center, he led the team in rebounds showing his ability to control the glass.

Giannis nearly accounts for 20% of the team's status for points, assists, and rebounds. This analysis showcases the massive dependence that the Bucks have for Giannis.

The final part of my project includes creating a new data frame of the players on the roster and each of their average points per game, assists per game, and rebounds per game.

In [12]:
```
pergame = pd.DataFrame(bucks, columns = ['Player'])
pergame
```

Out[12]:

	Player
0	Giannis Antetokounmpo
1	Khris Middleton
2	Brook Lopez
3	Eric Bledsoe
4	Wesley Matthews
5	Donte DiVincenzo
6	George Hill
7	Pat Connaughton
8	Ersan Ilyasova
9	Kyle Korver
10	Robin Lopez
11	Sterling Brown
12	D.J. Wilson
13	Marvin Williams
14	Thanasis Antetokounmpo
15	Frank Mason
16	Dragan Bender

In [13]:
```
# ppg = points per game
PTS_GP = bucks['PTS']/bucks['GP']
pergame.insert(1, 'ppg', PTS_GP, True)
pergame
```

Out[13]:

	Player	ppg
0	Giannis Antetokounmpo	29.476190
1	Khris Middleton	20.919355
2	Brook Lopez	11.955882
3	Eric Bledsoe	14.934426
4	Wesley Matthews	7.373134
5	Donte DiVincenzo	9.242424
6	George Hill	9.389831
7	Pat Connaughton	5.373134
8	Ersan Ilyasova	6.571429
9	Kyle Korver	6.655172
10	Robin Lopez	5.409091
11	Sterling Brown	5.115385
12	D.J. Wilson	3.594595
13	Marvin Williams	4.000000
14	Thanasis Antetokounmpo	2.750000
15	Frank Mason	6.888889
16	Dragan Bender	3.714286

In [14]:
```
# apg = assists per game
AST_GP = bucks['AST']/bucks['GP']
pergame.insert(2, 'apg', AST_GP, True)
pergame
```

Out[14]:

	Player	ppg	apg
0	Giannis Antetokounmpo	29.476190	5.619048
1	Khris Middleton	20.919355	4.274194
2	Brook Lopez	11.955882	1.455882
3	Eric Bledsoe	14.934426	5.377049
4	Wesley Matthews	7.373134	1.373134
5	Donte DiVincenzo	9.242424	2.333333
6	George Hill	9.389831	3.050847
7	Pat Connaughton	5.373134	1.597015
8	Ersan Ilyasova	6.571429	0.777778
9	Kyle Korver	6.655172	1.172414
10	Robin Lopez	5.409091	0.681818
11	Sterling Brown	5.115385	0.980769
12	D.J. Wilson	3.594595	0.675676
13	Marvin Williams	4.000000	1.117647
14	Thanasis Antetokounmpo	2.750000	0.750000
15	Frank Mason	6.888889	3.222222
16	Dragan Bender	3.714286	1.285714

In [15]:
```
# rpg = rebounds per game
TRB_GP = bucks['TRB']/bucks['GP']
pergame.insert(3, 'rpg', TRB_GP, True)
pergame
```

Out[15]:

	Player	ppg	apg	rpg
0	Giannis Antetokounmpo	29.476190	5.619048	13.587302
1	Khris Middleton	20.919355	4.274194	6.161290
2	Brook Lopez	11.955882	1.455882	4.588235
3	Eric Bledsoe	14.934426	5.377049	4.622951
4	Wesley Matthews	7.373134	1.373134	2.477612
5	Donte DiVincenzo	9.242424	2.333333	4.818182
6	George Hill	9.389831	3.050847	3.000000
7	Pat Connaughton	5.373134	1.597015	4.238806
8	Ersan Ilyasova	6.571429	0.777778	4.793651
9	Kyle Korver	6.655172	1.172414	2.103448
10	Robin Lopez	5.409091	0.681818	2.439394
11	Sterling Brown	5.115385	0.980769	3.519231
12	D.J. Wilson	3.594595	0.675676	2.486486
13	Marvin Williams	4.000000	1.117647	4.352941
14	Thanasis Antetokounmpo	2.750000	0.750000	1.200000
15	Frank Mason	6.888889	3.222222	2.111111
16	Dragan Bender	3.714286	1.285714	2.857143

In [16]:
```
pergame.round({'ppg':2, 'apg':2, 'rpg':2})
```

Out[16]:

	Player	ppg	apg	rpg
0	Giannis Antetokounmpo	29.48	5.62	13.59
1	Khris Middleton	20.92	4.27	6.16
2	Brook Lopez	11.96	1.46	4.59
3	Eric Bledsoe	14.93	5.38	4.62
4	Wesley Matthews	7.37	1.37	2.48
5	Donte DiVincenzo	9.24	2.33	4.82
6	George Hill	9.39	3.05	3.00
7	Pat Connaughton	5.37	1.60	4.24
8	Ersan Ilyasova	6.57	0.78	4.79
9	Kyle Korver	6.66	1.17	2.10
10	Robin Lopez	5.41	0.68	2.44
11	Sterling Brown	5.12	0.98	3.52
12	D.J. Wilson	3.59	0.68	2.49
13	Marvin Williams	4.00	1.12	4.35
14	Thanasis Antetokounmpo	2.75	0.75	1.20
15	Frank Mason	6.89	3.22	2.11
16	Dragan Bender	3.71	1.29	2.86

The complete dataframe shows that Giannis led the team in scoring at 29.48 points per game, assists at 5.62 per game, and rebounds at 13.59 per game. There were 6 players that averaged more than 9 points per game. 3 players averaged more than 4 assists per game. Only 1 player averaged more than 10 rebounds per game, with the next closest coming at 6.16.

After completing the analysis, I came away in awe of the impact of Giannis. As an avid Bucks fan, I readily see what he does on the court and for the organization, but to see it in numbers further defends how much the Bucks rely on him.

Some improvements that I thought of at the conclusion of this project include:

1. A deeper dive into how stats compare around a division or conference. A bigger analysis would include a league-wide comparison.
2. Build ranking and sorting functions into the analysis
3. Put parameters on who can be included in the analysis such as, X amount of games would need to be played in order to be included.
4. See how the departures and additions of rosters impact the team.

I hope this analysis was clear and helpful. I would certainly appreciate any comments, tips, or suggestions. This project will be available on my website https://joesliwinski.wixsite.com/home with hopefully more to come!

In [ ]:
```

```