Final Report133

November 27, 2016

Group members: Seung Soo Kim, Zimei Yuan, Jiyoon Jeong, and Jin Kweon

Research Question

In the 2015-2016 season, how do the skills of a player relate to his salary?

Abstract & Motivation

This report analyzes what National Basketball Association (NBA) players' salaries have to do with their performance (performance including many aspects, and we are going to talk about these as well). The motivation is to think of and predict how each team/the NBA sets the salary for each player, based on indicators (we say skills based on how I defined for our project. We defined skills later in our report): games played, age, positions, goals, rebounds, blocks, etc. Throughout the work of the project, our group tried to find out which variables are affecting players' salaries the most. At the end, we also analyze players' value and see if players are getting paid more than their value or less than their value. Our group is formed of four students: Seung Soo, Zimei, Jiyoon, and Jin. We have used R language the most, to come up with the answers. All the files and results are uploaded to Open Science Framework (OSF), with the link following: osf. The types of file extensions are ".R," ".Rmd," ".md," ".txt," ".csv," and ".pdf."

README File

Before we talk about our project, we want to introduce README File beforehand. We want to include a README file to show the general structure of our working directory. The README.md file includes the title (Stat133 Final Project Fall 2016), the authors' names, and a description of this project and the directory-files structure. This README file will let readers know about other files in the directory and therefore see our work clearly.

Introduction

Our analysis are focusing on the following five questions:

- How can we evaluate the performance of the players?
- Which skills are more correlated with salary?
- Are there any differences in skills and salary depending on the players' position?
- Are the players really worth the amounts of money clubs pay for?
- Are there any undervalued or undervalued players?

From these five analytical / exploratory research questions above, our group came up with an answer for the main question of our project: "In the 2015 - 2016 season, how do the skills of a player relate to his salary?"

Our group divided our work based on our skills. Seung Soo worked on "efficiency" statistics (EFF), Zimei worked on Readme, data dictionaries for each data file, and slides (Jerome also contributed on Readme, data dictionaries, and slides, as well), Clover worked on data acquisition, cleaning, exploratory data analysis (EDA), and Shiny Apps, and Jin worked on Shiny Apps, ggplots, and the report (all other members contributing on the report, as well).

Data

First of all, NBA is the abbreviation of the National Basketball Association: men's professional basketball league in North America. All the players' performances and salaries are based on the 2015 - 2016 season. And, the main source is the NBA official website: basketball website. In that season, there are thirty teams in the league. So, in total, we acquired 90 raw data files.
We developed all the raw data tables for the following three types: roster, totals, and salaries, for each of the thirty teams in the NBA. The names of the teams are following: "Atlanta Hawks (ATL)," "Boston Celtics (BOS),"Brooklyn Nets (BRK)," "Charlotte Hornets (CHO)," "Chicago Bulls (CHI),"Cleveland Cavaliers (CLE)," "Dallas Mavericks (DAL)," "Denver Nuggets (DEN)," "Detroit Pistons (DET)," "Golden State Warriors (GSW)," "Houston Rockets (HOU)," "Indiana Pacers (IND)," "Los Angeles Clippers (LAC)," "Los Angeles Lakers (LAL)," "Memphis Grizzlies (MEM)," "Miami Heat (MIA)," "Milwaukee Bucks (MIL)," "Minnesota Timberwolves (MIN)," "New Orleans Pelicans (NOP)," "New York Knicks (NYK)," "Oklahoma City Thunder (OKC)," "Orland Magic (ORL)," "Philadelphia 76ers (PHI)," "Phoenix Suns (PHO)," "Portland Trail Blazers (POR)," "Sacramento Kings (SAC)," "San Antonio Spurs (SAS)," "Toronto Raptors (TOR)," "Utah Jazz (UTA)," and "Washington Wizards (WAS)."
All the rosters tables include the information about roster number, players' names, positions, heights, weights, birth dates, countries born, NBA years of experiences, and colleges attended. All the totals tables include the information about total rank, players' names, ages, games, games started, minutes played, field goals, field goal attempts, field goal percentages, 3-point field goals, 3-point field goal attempts, 3-point field goal percentages, 2-point field goals, 2-point field goal attempts, 2-point field goal percentages, effective field goal percentages, free throws, free throw attempts, free throw percentages, offensive rebounds, defensive rebounds, total rebounds, assists, steals, blocks, turnovers, personal fouls, and points. All the salaries tables include the information about salary ranks, roster players' names, and salaries.

	roster.No.	roster.Player	roster.Pos	roster.Ht	roster.Wt	roster.Birth.Date	roster	roster.Exp	roster.College
1	9	Jared Cunningham	SG	04-6	195	May 22, 1991	us	3	Oregon State University
2	8	Matthew Dellavedova	PG	04-6	198	September 8, 1990	au	2	Saint Mary's College of California
3	9	Channing Frye	C	11-6	255	May 17, 1983	us	9	University of Arizona
4	12	Joe Harris	SG	06-6	219	September 7, 1991	us	1	University of Virginia
5	2	Kyrie Irving	PG	03-6	193	March 23, 1992	au	4	Duke University
6	23	LeBron James	SF	08-6	250	December 30, 1984	us	12	
7	24	Richard Jefferson	SF	07-6	233	June 21, 1980	us	14	University of Arizona
8	30	Dahntay Jones	SF	06-6	225	December 27, 1980	us	11	Duke University
9	1	James Jones	SF	08-6	218	October 4, 1980	us	12	University of Miami
10	14	Sasha Kaun	C	11-6	260	May 8, 1985	ru	R	University of Kansas
11	0	Kevin Love	PF	10-6	251	September 7, 1988	us	7	University of California, Los Angeles
12	12	Jordan McRae	PG	06-6	185	March 28, 1991	us	R	University of Tennessee
13	20	Timofey Mozgov	C	01-7	275	July 16, 1986	ru	5	
14	4	Iman Shumpert	SG	05-6	220	June 26, 1990	us	4	Georgia Institute of Technology
15	5	J.R. Smith	SG	06-6	225	September 9, 1985	us	11	
16	13	Tristan Thompson	PF	09-6	238	March 13, 1991	ca	4	University of Texas at Austin
17	17	Anderson Varejao	C	10-6	273	September 28, 1982	br	11	
18	52	Mo Williams	PG	01-6	198	December 19, 1982	us	12	University of Alabama

Figure 1. This is one of the roster table (csv) we acquired from the link

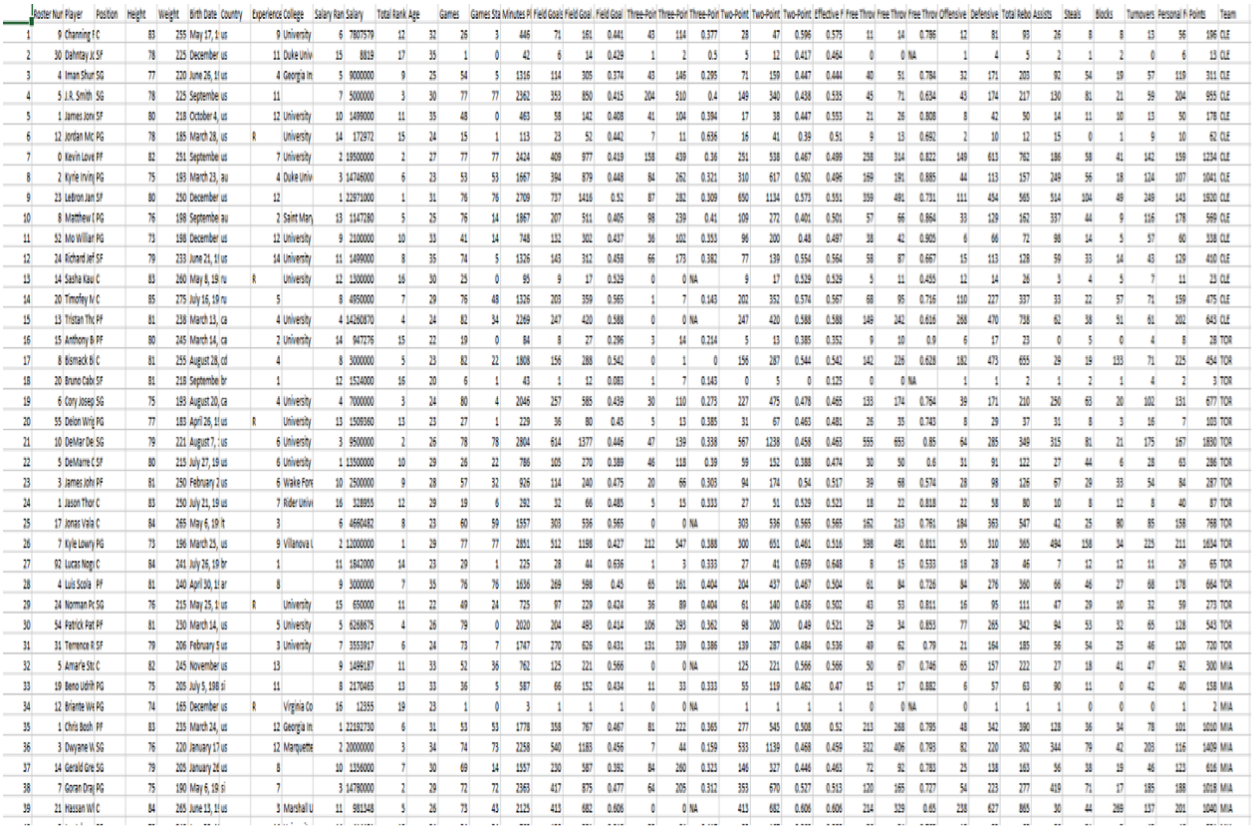
	salaries.Rk	salaries.Player	salaries.Salary
1	1	LeBron James	$22,971,000
2	2	Kevin Love	$19,500,000
3	3	Kyrie Irving	$14,746,000
4	4	Tristan Thompson	$14,260,870
5	5	Iman Shumpert	$9,000,000
6	6	Channing Frye	$7,807,579
7	7	J.R. Smith	$5,000,000
8	8	Timofey Mozgov	$4,950,000
9	9	Mo Williams	$2,100,000
10	10	James Jones	$1,499,000
11	11	Richard Jefferson	$1,499,000
12	12	Sasha Kaun	$1,300,000
13	13	Matthew Dellavedova	$1,147,280
14	14	Jordan McRae	$172,972
15	15	Dahntay Jones	$8,819

Figure 2. This is one of the salary table (csv) we acquired from the link

	totals.Rk	totals.Player	totals.Age	totals.G	totals.GS	totals.MP	totals.FG	totals.FGA	totals.FG.	totals.3P	totals.3PA	totals.3P.	totals.2P	totals.2PA	totals.2P.	totals.eFG.	totals.FT	totals.FTA	totals.FT.	totals.ORB	totals.DRB	totals.TRB	totals.AST	totals.STL	totals.BLK	totals.TOV	totals.PF	totals.PTS
1	1	LeBron James	31	76	76	2709	737	1416	0.52	87	282	0.309	650	1134	0.573	0.551	359	491	0.731	111	454	565	514	104	49	249	143	1920
2	2	Kevin Love	27	77	77	2424	409	977	0.419	158	439	0.36	251	538	0.467	0.499	258	314	0.822	149	613	762	186	58	41	142	159	1234
3	3	J.R. Smith	30	77	77	2362	353	850	0.415	204	510	0.4	149	340	0.438	0.535	45	71	0.634	43	174	217	130	81	21	59	204	955
4	4	Tristan Thompson	24	82	34	2269	247	420	0.588	0	0		247	420	0.588	0.588	149	242	0.616	268	470	738	62	38	51	61	202	643
5	5	Matthew Dellavedova	25	76	14	1867	207	511	0.405	98	239	0.41	109	272	0.401	0.501	57	66	0.864	33	129	162	337	44	9	116	178	569
6	6	Kyrie Irving	23	53	53	1667	394	879	0.448	84	262	0.321	310	617	0.502	0.496	169	191	0.885	44	113	157	249	56	18	124	107	1041
7	7	Timofey Mozgov	29	76	48	1326	203	359	0.565	1	7	0.143	202	352	0.574	0.567	68	95	0.716	110	227	337	33	22	57	71	159	475
8	8	Richard Jefferson	35	74	5	1326	143	312	0.458	66	173	0.382	77	139	0.554	0.564	58	87	0.667	15	113	128	59	33	14	43	129	410
9	9	Iman Shumpert	25	54	5	1316	114	305	0.374	43	146	0.295	71	159	0.447	0.444	40	51	0.784	32	171	203	92	54	19	57	119	311
10	10	Mo Williams	33	41	14	748	132	302	0.437	36	102	0.353	96	200	0.48	0.497	38	42	0.905	6	66	72	98	14	5	57	60	338
11	11	James Jones	35	48	0	463	58	142	0.408	41	104	0.394	17	38	0.447	0.553	21	26	0.808	8	42	50	14	11	10	13	50	178
12	12	Channing Frye	32	26	3	446	71	161	0.441	43	114	0.377	28	47	0.596	0.575	11	14	0.786	12	81	93	26	8	8	13	56	196
13	13	Jared Cunningham	24	40	3	355	32	91	0.352	10	32	0.313	22	59	0.373	0.407	30	48	0.625	3	26	29	19	12	2	18	37	104
14	14	Anderson Varejao	33	31	0	310	32	76	0.421	0	1	0	32	75	0.427	0.421	16	21	0.762	24	67	91	20	11	5	16	35	80
15	15	Jordan McRae	24	15	1	113	23	52	0.442	7	11	0.636	16	41	0.39	0.51	9	13	0.692	2	10	12	15	0	1	9	10	62
16	16	Sasha Kaun	30	25	0	95	9	17	0.529	0	0		9	17	0.529	0.529	5	11	0.455	12	14	26	3	4	5	7	11	23
17	17	Dahntay Jones	35	1	0	42	6	14	0.429	1	2	0.5	5	12	0.417	0.464	0	0		1	4	5	2	1	2	0	6	13
18	18	Joe Harris	24	5	0	15	1	4	0.25	1	4	0.25	0	0		0.375	0	0		0	3	3	2	0	0	1	1	3

Figure 3. This is one of the totals (stats) table (csv) we acquired from the link

After we obtained the raw data of each team's records, we removed unnecessary data, edited the data structure and measurements of each variable, and added certain necessary data to the new data frame, to generate 'roster-salary-stats.csv.' The methods we used to acquire cleaned table are described under the section, Methodology.



	Roster Nur	Player	Position	Height	Weight	Birth Date	Country	Experience	College	Salary Ran	Salary	Total Rank	Age	Games	Games Sta	Minutes P	Field Goals	Field Goal	Field Goal	Three-Poin	Three-Poin	Three-Poin	Two-Point	Two-Point	Two-Point	Effective F	Free Throw	Free Throw	Free Throw	Offensive	Defensive	Total Rebo	Assists	Steals	Blocks	Turnovers	Personal F	Points	Team
1	9	Channing F	C	83	255	May 17, 1	us	9	University	6	7807579	12	32	26	3	446	71	161	0.441	43	114	0.377	28	47	0.596	0.575	11	14	0.786	12	81	93	26	8	8	13	96	196	CLE
2	30	Dahntay Jo	SF	78	225	December	us	11	Duke Univ	15	8819	17	35	1	0	42	6	14	0.429	1	2	0.5	5	12	0.417	0.464	0	0	NA	1	4	5	2	1	2	0	6	13	CLE
3	4	Iman Shum	SG	77	220	June 26, 1	us	4	Georgia In	5	9000000	9	25	54	5	1316	114	305	0.374	43	146	0.295	71	159	0.447	0.444	40	51	0.784	32	171	203	92	54	19	57	119	311	CLE
4	5	J.R. Smith	SG	78	225	Septembe	us	11		7	5000000	3	30	77	77	2362	353	850	0.415	204	510	0.4	149	340	0.438	0.535	45	71	0.634	43	174	217	130	81	21	59	204	955	CLE
5	1	James Jon	SF	80	218	October 4,	us	12	University	10	1499000	11	35	48	0	463	58	142	0.408	41	104	0.394	17	38	0.447	0.553	21	26	0.808	8	42	50	14	11	10	13	90	178	CLE
6	12	Jordan Mc	PG	78	185	March 28,	us	R	University	14	172972	15	24	15	1	113	23	52	0.442	7	11	0.636	16	41	0.39	0.51	9	13	0.692	2	10	12	15	0	1	9	10	62	CLE
7	0	Kevin Love	PF	82	251	Septembe	us	7	University	2	19500000	2	27	77	77	2424	409	977	0.419	158	439	0.36	251	538	0.467	0.499	268	314	0.822	149	613	762	186	58	41	142	159	1234	CLE
8	2	Kyrie Irving	PG	75	193	March 23,	au	4	Duke Univ	3	14766000	6	23	53	53	1667	394	879	0.448	84	262	0.321	310	617	0.502	0.496	169	191	0.885	44	113	157	249	56	18	124	107	1041	CLE
9	23	Lebron Jam	SF	80	250	December	us	12		1	22971000	1	31	76	76	2709	737	1416	0.52	87	282	0.309	650	1134	0.573	0.551	359	491	0.731	111	454	565	514	104	49	249	143	1930	CLE
10	8	Matthew F	PG	76	198	Septembe	au	2	Saint Mary	13	1147780	5	25	76	14	1867	207	511	0.405	98	239	0.41	109	272	0.401	0.501	57	66	0.864	33	129	162	337	44	9	116	178	569	CLE
11	52	Mo Williar	PG	73	198	December	us	12	University	9	2100000	10	33	41	14	748	132	302	0.437	36	102	0.353	96	200	0.48	0.497	38	42	0.905	6	66	72	98	14	5	57	60	338	CLE
12	24	Richard Jef	SF	79	233	June 21, 1	us	14	University	11	1499000	8	35	74	5	1326	143	312	0.458	66	173	0.382	77	139	0.554	0.564	58	87	0.667	15	113	128	59	33	14	43	120	410	CLE
13	14	Sasha Kau	C	83	260	May 8, 19	ru	R	University	12	1300000	16	30	25	0	95	9	17	0.529	0	0	NA	9	17	0.529	0.529	5	11	0.455	12	14	26	3	4	5	7	11	23	CLE
14	20	Timofey M	C	85	275	July 16, 19	ru	5		8	4950000	7	29	76	48	1326	203	359	0.565	1	7	0.143	202	352	0.574	0.567	68	95	0.716	110	227	337	33	22	57	71	159	475	CLE
15	13	Tristan Thc	PF	81	238	March 13,	ca	4	University	4	14260870	4	24	82	34	2269	247	420	0.588	0	0	NA	247	420	0.588	0.588	148	242	0.616	268	470	738	62	38	51	61	202	643	CLE
16	15	Anthony B	PF	80	245	March 14,	ca	2	University	14	947276	15	22	19	0	84	8	27	0.296	3	14	0.214	5	13	0.385	0.352	9	10	0.9	6	17	23	0	5	0	4	8	28	TOR
17	8	Bismack Bi	C	81	255	August 28,	cd	4		8	3000000	5	23	82	22	1808	156	288	0.542	0	1	0	156	287	0.544	0.542	142	226	0.628	182	473	655	29	19	133	71	225	454	TOR
18	20	Bruno Cabi	SF	81	218	Septembe	br	1		12	1524000	16	20	6	1	43	1	12	0.083	1	7	0.143	0	5	0	0.125	0	0	NA	1	1	2	1	2	1	4	2	3	TOR
19	6	Cory Joseph	SG	75	193	August 20,	ca	4	University	4	7000000	3	24	80	4	2046	257	585	0.439	30	110	0.273	227	475	0.478	0.465	133	174	0.764	39	171	210	290	63	20	102	131	677	TOR
20	55	Delon Wrig	PG	77	183	April 26, 1	us	R	University	13	1500360	13	23	27	1	229	36	80	0.45	5	13	0.385	31	67	0.463	0.481	26	35	0.743	8	29	37	31	8	3	16	7	103	TOR
21	10	DeMar DeI	SG	79	221	August 7,	us	6	University	3	9500000	2	26	78	78	2804	614	1377	0.446	47	139	0.338	567	1238	0.458	0.463	555	653	0.85	64	285	349	315	81	21	175	167	1830	TOR
22	5	DeMarre C	SF	80	215	July 27, 19	us	6	University	1	13500000	10	29	26	22	786	105	270	0.389	46	118	0.39	59	152	0.388	0.474	30	50	0.6	31	91	122	27	44	6	28	63	286	TOR
23	3	James Johi	PF	81	250	February 2	us	6	Wake Fore	10	2500000	9	28	57	32	926	114	240	0.475	20	66	0.303	94	174	0.54	0.517	39	68	0.574	28	98	126	67	29	33	54	84	287	TOR
24	1	Jason Thor	C	83	250	July 21, 19	us	7	Rider Univ	16	328955	12	29	19	6	292	32	66	0.485	3	13	0.333	27	51	0.529	0.523	18	22	0.818	22	58	80	10	8	12	8	40	87	TOR
25	17	Jonas Vala	C	84	265	May 6, 19	lt	3		6	4866482	8	23	60	59	1557	303	536	0.565	0	0	NA	303	536	0.565	0.565	162	213	0.761	184	363	547	42	25	80	85	158	788	TOR
26	7	Kyle Lowry	PG	73	196	March 25,	us	9	Villanova	2	12000000	1	29	77	77	2851	512	1198	0.427	212	547	0.388	300	651	0.461	0.516	398	491	0.811	55	310	365	484	158	34	225	211	1634	TOR
27	92	Lucas Nog	C	84	241	July 26, 19	br	1		11	1842000	14	23	29	1	225	28	44	0.636	1	3	0.333	27	41	0.659	0.648	8	15	0.533	18	28	46	7	12	12	11	29	65	TOR
28	4	Luis Scola	PF	81	240	April 30, 1	ar	8		9	3000000	7	35	76	76	1636	269	598	0.45	65	161	0.404	204	437	0.467	0.504	61	84	0.726	84	276	360	66	46	27	68	178	664	TOR
29	24	Norman Pc	SG	76	215	May 25, 1	us	R	University	15	650000	11	22	49	24	725	97	229	0.424	36	89	0.404	61	140	0.436	0.502	43	53	0.811	16	95	111	47	29	10	32	59	273	TOR
30	54	Patrick Pat	PF	81	230	March 14,	us	5	University	5	6268675	4	26	79	0	2020	204	493	0.414	106	293	0.362	98	200	0.49	0.521	29	34	0.853	77	265	342	94	53	32	65	128	543	TOR
31	31	Terrence R	SF	79	206	February 5	us	3	University	7	3553917	6	24	73	7	1747	270	626	0.431	131	339	0.386	139	287	0.484	0.536	49	62	0.79	21	164	185	56	54	25	46	120	720	TOR
32	5	Amar'e Stc	C	82	245	November	us	13		9	1499187	11	33	52	36	762	125	221	0.566	0	0	NA	125	221	0.566	0.566	50	67	0.746	65	157	222	27	18	41	47	92	300	MIA
33	19	Beno Udrih	PG	75	205	July 5, 198	si	11		8	2170465	13	33	36	5	587	66	152	0.434	11	33	0.333	55	119	0.462	0.47	15	17	0.882	6	57	63	90	11	0	42	40	158	MIA
34	12	Briante Wil	PG	74	165	December	us	R	Virginia Co	16	12355	19	23	1	0	3	1	1	1	0	0	NA	1	1	1	1	0	0	NA	0	1	1	1	0	0	0	1	2	MIA
35	1	Chris Bosh	PF	83	235	March 24,	us	12	Georgia In	1	22192730	6	31	53	53	1778	358	767	0.467	81	222	0.365	277	545	0.508	0.52	213	268	0.795	48	342	390	128	36	34	78	101	1010	MIA
36	3	Dwyane W	SG	76	220	January 17	us	12	Marquette	2	20000000	3	34	74	73	2258	540	1183	0.456	7	44	0.159	533	1139	0.468	0.459	322	406	0.793	82	220	302	344	79	42	203	116	1409	MIA
37	14	Gerald Gre	SG	79	205	January 26	us	8		10	1356000	7	30	69	14	1557	230	587	0.392	84	260	0.323	146	327	0.446	0.463	72	92	0.783	25	138	163	56	38	19	46	123	616	MIA
38	7	Goran Dra	PG	75	190	May 6, 19	si	7		3	14780000	2	29	72	72	2393	417	875	0.477	64	205	0.312	353	670	0.527	0.513	120	165	0.727	54	223	277	419	71	17	185	188	1018	MIA
39	21	Hassan Whi	C	84	265	June 13, 1	us	3	Marshall U	11	981348	5	26	73	43	2125	413	682	0.606	0	0	NA	413	682	0.606	0.606	214	329	0.65	238	627	865	30	44	269	137	201	1040	MIA

Figure 4. This is the part of a cleaned csv (roster-salary-stats.csv). This table includes 39 columns in total: roster numbers, roster players' names, positions, heights, weights, birth dates, countries of birth, NBA years of experiences, colleges attended, salary ranks, salaries, total ranks, ages, games, games started, minutes played, field goals, field goal attempts, field goal percentages, 3-point field goals, 3-point field goal attempts, 3-point field goal percentages, 2-point field goals, 2-point field goal attempts, 2-point field goal percentages, effective field goal percentages, free throws, free throw attempts, free throw percentages, offensive rebounds, defensive rebounds, total rebounds, assists, steals, blocks, turnovers, personal fouls, points, and team names.

Methodology & Analysis

We modified the provided code using loops and creating our own function to avoid repetitive types of codes, and to shorten the lengths of codes. The Data Scraping part is composed of two R scripts, which are 'download-data-script.R' and 'download-data-functions.R' inside of the functions and scripts files, respectively. First, this is the methodology of the data scraping. We extracted each team's information, such as href attributes, from the anchor tags, and teams' abbreviations from the specified URL. This part of the code is included in an R script named 'download-data-functions.R', since we need the data such as 'team_names' in 'clean-data-script.R' to load each team's raw data files by the team's name. Merely Sourcing 'download-data-functions.R' enables us to get the NBA team's name in 'clean-data-script.R' without writing the same code to get the team's name. After we obtained each team's name and URL for designated data from the previous code, we had to get the information on the roster, salary, and stat table in that URL by recognizing the initial line position of roster, salary, and stat html table. The last line where those three html tables end was obtained by the variable 'line_counter', where it counts the number of lines before

.

After we obtain the first and last line position on the desired html table, we read the information

on rosters, salaries, and stats html tables as data.frame. Finally, we export the acquired data frame as a csv file, and name it in the form of 'the type of data (roster, salaries, or totals)'-'each team's name'. The purpose of this code is to obtain each team's csv file, and it is more efficient run by functions, since the same logic (getting the initial and last line of the html table and exporting the table as a csv file) with different arguments (the different file path and name of the different type of tables) applies to all three html tables (roster, salary, and stats). Therefore, function 'exportcsv' is created, and this function is run by 'download-data-script.R' to generate each team's csv file for three types of data tables.

Second, this is the methodology of the data cleaning.

-Removed and edited data from raw data 1. In salaries. Salary column in Salary csv, $(dollar sign) in front of each player's salary is removed and the salary data is changed into numeric vectors. 2. In roster csv, the original roster.Ht columns' values is in inches and feet, but we removed dash(-) and changed the measurement to feet. We also changed this column vectors in numeric vectors to make the graph easier to understand, in the Analysis part. 3. When we merged all the columns of each data table, we obtained duplicated players, belonging to different teams. Those players' rows were removed, except for the first row of each player. For example, we have 'Alex Stepheson' who belongs to LAC, and 'Alex Stepheson' who belongs to MEM, and we just choose the 'Alex Stepheson' who belongs to LAC, since it is the first row that shows data for 'Alex Stepheson.'
4. We removed the previous column names and added more descriptive column names, in order for us to be able to recognize the data easily.

-Added data 1. We added a new column, called "Team", since we need to arrange players by the names of their teams, to draw the graph in Shiny App (team-salaries) and obtain information in "eda-script.R."

-Joined data 1. First, we merged roster table data and salaries table data, which are ordered by players' names. We saved this data into the variable called 'a.' 2. Secondly, we merged the data 'a' with the stats table data, which are also ordered by players' names, and we got the merged data of the three tables.

For exploratory data analysis (in eda-script), we divide up quantitative and qualitative variables. The qualitative variables in our csv file are four in total: "player name," "position," "birth date," "country origin," "colleges players graduated," and "teams." The other 33 are all quantitative, such as roster number, height, weight, salary, etc. With quantitative variables, there are some special cases we have to modify for some columns. For example, the experience column has "R", and the three-point field goal percentage has "NA," and to make R program consider it numeric, we decided to make "R" and "NA" equal to zero (0) - there is either no information, or players have no experience in the columns.

Here is an example below:

Experience
9
11
4
11
12
R
7

Figure 5. This picture shows one of the examples that some columns include a character

while most of the elements are numeric.

> In this R script, we used 'sink' (function in R) eda-output.txt to save minimum, median, tenth / twenty-five / seventy-five / ninety quartiles, inter-quartiles, maximum, minimum, median, standard deviation, range, difference (between the biggest and smallest numbers), and sums for quantitative variables in eda-output.txt. Most of the time, we used for-loop to get summaries for each column.

Here is one of the examples for for-loop below:

```
# Get minimum for each column
for (i in 1:33) {
  minimum[i] = min(data[,quant[i]], na.rm = TRUE)
}
```

Figure 6. This picture is part of the codes how we used for-loop to get statistics for some variables.

Ranges

```
 [1]        0       99       69       87      161      307        0       20        1       23     8819
[12] 25000000        1       27       19       39        1       82        0       82        3     3125
[23]        0      805        0     1617        0        1        0      402        0      886        0
[34]        1        0      650        0     1238        0        1        0        1        0      720
[45]        0      837        0        1        0      395        0      803        0     1198        0
[56]      839        0      169        0      269        0      374        0      258        0     2376
```

Difference

```
 [1]     99.0     18.0    146.0     20.0     22.0 24991181.0     26.0     20.0     81.0
[10]     82.0   3122.0    805.0   1617.0      1.0    402.0    886.0      1.0    650.0
[19]   1238.0      1.0      1.5    720.0    837.0      1.0    395.0    803.0   1198.0
[28]    839.0    169.0    269.0    374.0    258.0   2376.0
```

Sum

```
 [1]    8439   37261  104241    2295    4323 2173709567    4190   12533   24668
[10]   11840  569488   90642  199882      NA      20040   56502      NA   70602
[19]  143380      NA      NA   42023   55546         NA   24748   79076  103824
[28]   52514   18553   11855   32523   47687     243347
```

Figure 7. This is part of the 'eda-output.txt,' including all the statistics.

Also, we made some graphs, but there are some variables that we did not use in the graphs: roster number, player name, birth date, salary rank, and total rank. We skipped them because those variables have nothing to do with the skills of the players. Quantitative variables are interpreted with histograms and boxplots (and summary statistics in eda-output.txt), and qualitative variables are interpreted with bar-charts (and frequency tables in eda-output.txt). For every bar-chart, we displayed them horizontally, for visual purposes.

Here are some of the ggplots we got below:

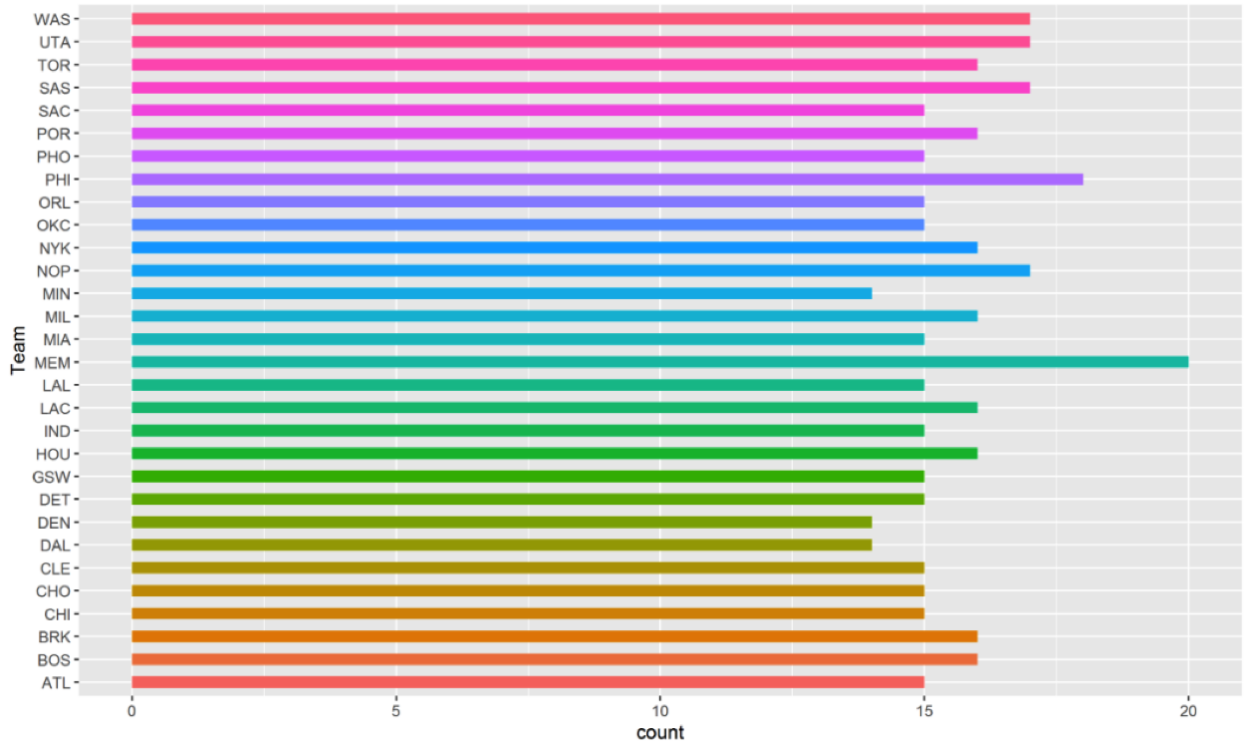


Figure 8. Here is one of the bar charts, displaying the counts of the teams.

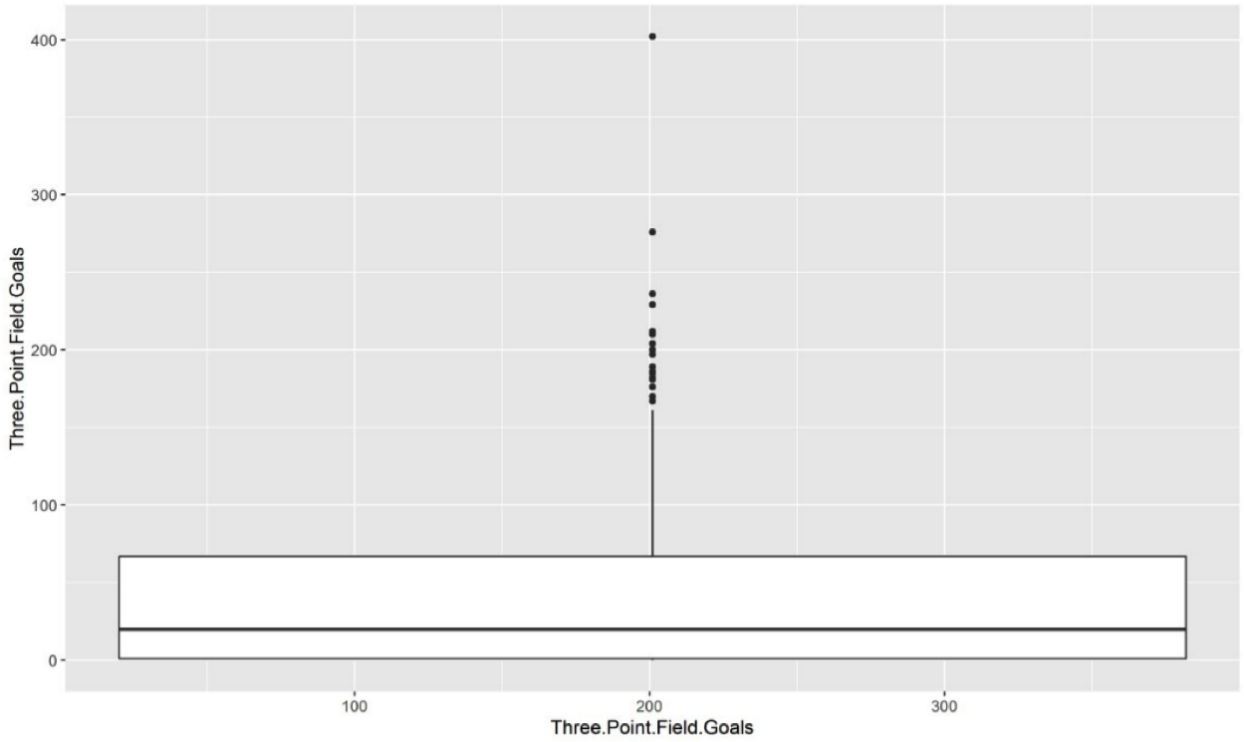


Figure 9. Here is one of the box plots, displaying the data (IQR, quartiles, media, etc) for three point field goals.

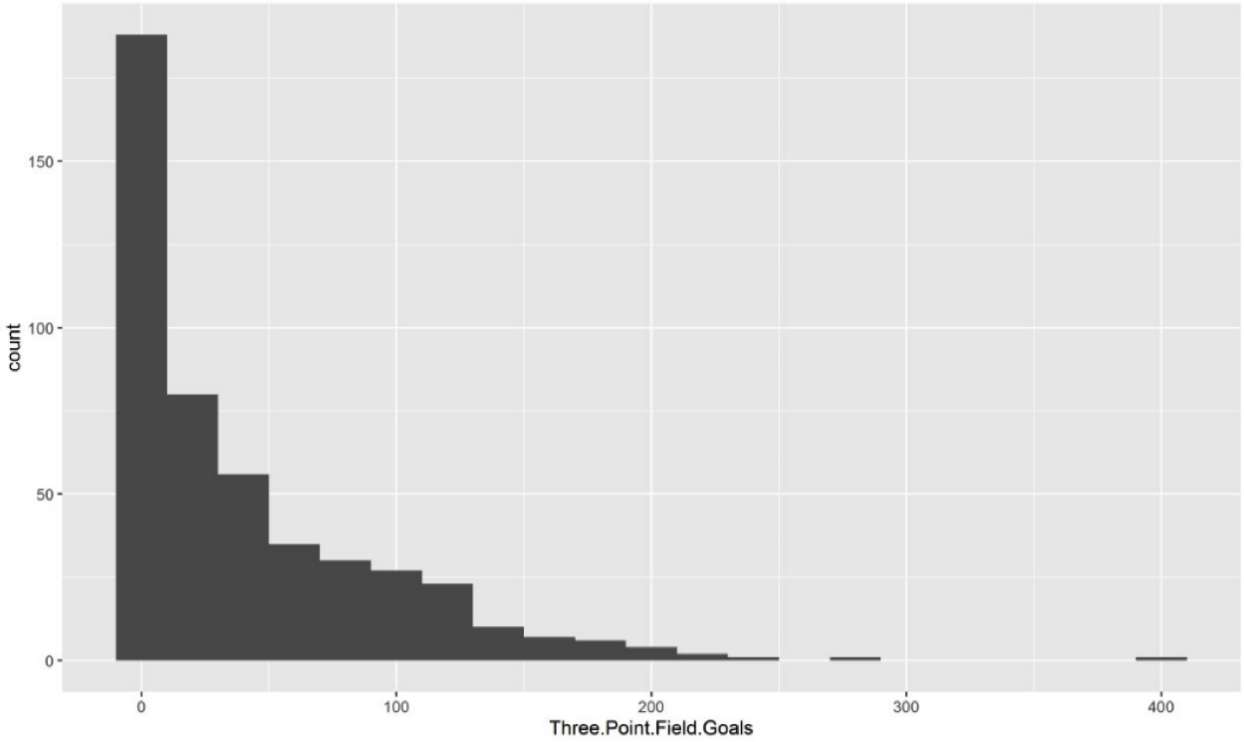


Figure 10. Here is one of the histograms, displaying the counts of three point field goals.

PCA The original efficiency formula is offense oriented, so defensive players might seem less efficient compared to those who have offensive position. In order to balance the efficiency between offensive and defensive players, we applied Principal Component Analysis(PCA) to the original efficiency.

Modified EFF In order to calculate the efficiency for each position, we classified the cleandata by players' position, and added missing records(missed free throws, missed field goals, turnovers). Also, I divided records(points, assists, steals etc) by the number of played game to see the efficiency per game. The modified efficiency was obtained by multiplying each variable with coefficient that is derived by PCA. To standardize the coefficient(weight), we divided the coefficient by a standard deviation of each position. So the modified EFF formula is the sum of coefficient * variables(per game) / standard deviation.

"eff-stats-salary.csv" consists of 14 columns (refer the picture below). The efficiency index is the modified efficiency we obtained by using PCA.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		Player's name	Total Points	Total Rebounds	Assists	Steals	Blocks	Bissed Field goals	Missed Free Throws	Turnovers	Games Played	Efficiency Index	Salary	Position
2	1	Channing Frye	196	93	26	8	8	90	3	13	26	2.816584877	7807579	C
3	2	Sasha Kaun	23	26	3	4	5	8	6	7	25	0.814241467	1300000	C
4	3	Timofey Mozgov	475	337	33	22	57	156	27	71	76	2.912380347	4950000	C
5	4	Bismack Biyombo	454	655	29	19	133	132	84	71	82	3.758825301	3.00E+06	C
6	5	Jason Thompson	87	80	10	8	12	34	4	8	19	2.458519905	328955	C
7	6	Jonas Valanciunas	768	547	42	25	80	233	51	85	60	5.31272371	4660482	C
8	7	Lucas Nogueira	65	46	7	12	12	16	7	11	29	1.495782421	1842000	C
9	8	Amar'e Stoudemire	300	222	27	18	41	96	17	47	52	2.896153675	1499187	C
10	9	Hassan Whiteside	1040	865	30	44	269	269	115	137	73	7.516716132	981348	C
11	10	Al Horford	1249	596	263	68	121	519	26	107	82	6.838641081	1.20E+07	C
12	11	Mike Muscala	195	117	34	13	27	76	8	27	60	1.684619196	947276	C
13	12	Tiago Splitter	201	120	30	20	12	74	9	24	36	2.7330762	8500000	C
14	13	Walter Tavares	25	21	3	1	6	8	5	5	11	1.436109759	1.00E+06	C
15	14	Kelly Olynyk	687	281	105	52	33	303	32	74	69	4.420886752	2165160	C
16	15	Tyler Zeller	364	178	29	10	22	152	20	46	60	2.395425992	2616975	C
17	16	Al Jefferson	562	301	70	30	41	260	39	34	47	5.060941729	13833333	C
18	17	Cody Zeller	638	455	71	57	63	206	57	68	73	4.365449173	4204200	C
19	18	Frank Kaminsky	606	335	98	38	43	310	40	58	81	3.557108973	2612500	C
20	19	Ian Mahinmi	660	507	104	65	75	184	93	100	71	5.356540724	4.00E+06	C

Figure 11. This is part of the "eff-stats-salary.csv." We added an extra column, 'position' for Shiny app "stat-salaries."

Shiny App shows the correlation between two variables easily. For example, in the Shiny App image below, we can see how total points and salary are positively correlated. Also, we see that most of the players are distributed between 0 and 300 points, regardless of their positions.

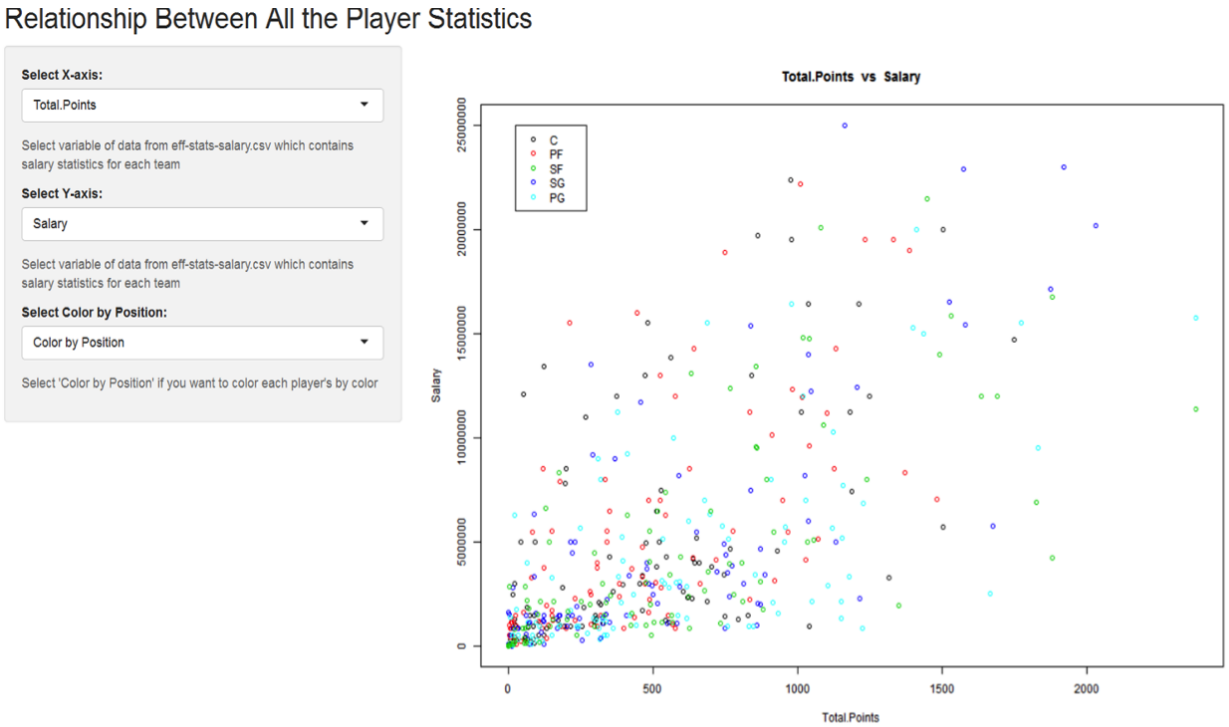


Figure 12. This is one of the Shiny pictures: showing the relationship between total points and salary.

"best-worst-value-players.txt" consists of 5 columns(index number, player's name, position, value, identify). We picked top 20 and bottom 20 players among all players by ordering the value. Value of a player is obtained by a formula that is efficiency divided by salary.

	Player's name	Position	Value	Identify
1	Dahntay Jones	SF	0.0006640627169495	most valuable
2	Jordan Hamilton	SF	0.0000462826969291	most valuable
3	Xavier Munford	SG	0.0000462591842282	most valuable
4	Jared Cunningham	SG	0.0000427049529131	most valuable
5	Briante Weber	PG	0.0000425215533947	most valuable
6	Orlando Johnson	SG	0.0000304638403307	most valuable
7	Alan Williams	PF	0.0000247057141562	most valuable
8	Chuck Hayes	C	0.0000237752114430	most valuable
9	Henry Sims	C	0.0000230695486776	most valuable
10	Jordan Farmar	PG	0.0000201699600456	most valuable
11	Phil Pressey	PG	0.0000181760698908	most valuable
12	Jimmer Fredette	SG	0.0000171761621858	most valuable
13	Lorenzo Brown	PG	0.0000164425652586	most valuable
14	Marcus Thornton	SG	0.0000162035168349	most valuable
15	Nate Robinson	PG	0.0000160879282800	most valuable
16	Elliot Williams	PG	0.0000157343501652	most valuable
17	Michael Beasley	SF	0.0000150545786738	most valuable
18	Axel Toupane	SF	0.0000148240847353	most valuable
19	Alex Stepheson	PF	0.0000147834979193	most valuable
20	Bryce Dejean-Jones	SG	0.0000146335270101	most valuable
21	Nikola Pekovic	C	0.0000001656909312	worst value
22	Sam Dekker	SF	0.0000001846189651	worst value
23	Roy Hibbert	C	0.0000002423578677	worst value
24	Omer Asik	C	0.0000002465642030	worst value
25	Chris Kaman	C	0.0000002590056923	worst value
26	Enes Kanter	C	0.0000002674877260	worst value
27	Kobe Bryant	SF	0.0000002725500483	worst value
28	David Lee	PF	0.0000002772035116	worst value
29	Derrick Rose	PG	0.0000003001176206	worst value
30	Joel Anthony	C	0.0000003159084722	worst value
31	Wesley Matthews	SG	0.0000003214292266	worst value
32	Tiago Splitter	C	0.0000003215383765	worst value
33	Kirk Hinrich	PG	0.0000003357659578	worst value
34	Jodie Meeks	SG	0.0000003383357354	worst value
35	Tyson Chandler	C	0.0000003396614614	worst value
36	Brian Roberts	PG	0.0000003415218146	worst value
37	Chris Bosh	PF	0.0000003486950934	worst value
38	Tony Parker	PG	0.0000003490283122	worst value
39	Tristan Thompson	PF	0.0000003512295323	worst value
40	Dwight Howard	C	0.0000003565347428	worst value

Figure 13, 14. This is picture of the "best-worst-value-players.txt," showing the 20 most and worst value players.

From now on, we answered the five main questions below for analysis:

1. How can we evaluate the performance of the players?

We calculated applied modified efficiency by using PCA to standardize the indicator, because the original formula is offensive position oriented. Formula = (w1PTS + $w2$REB + w3AST + , , + w8*TO) / GP, (w1~w8 are weights)

Efficiency of center

Nazr Mohammed has the lowest efficiency (0.6529) among all center players. DeMarcus Cousins has the highest efficiency (12.294) among all center players. The median efficiency of centers is 3.6299, and the mean is 4.0088.

Efficiency of power forward

Branden Dawson has the lowest efficiency (0.3678) among all power forward players. Draymond Green has the highest efficiency (11.5831) among all power forward players. The median efficiency of power forwards is 3.6971, and the mean is 4.0242.

Efficiency of small forward

Sam Dekker has the lowest efficiency(0.304) among all small forward players. LeBron James has the highest efficiency (11.149) among all small forward players. The median efficiency of power forwards is 3.474, and the mean is 3.878.

Efficiency of shooting guard

Luis Montero has the lowest efficiency (0.6448) among all shooting guard players. James Harden has the highest efficiency (13.7412) among all shooting guard players. The median efficiency of power forwards is 4.1755, and the mean is 4.5777.

Efficiency of point guard

Bryce Cotton has the lowest efficiency (0.0477) among all point guard players. Russell Westbrook has the highest efficiency (10.7583) among all point guard players. The median efficiency of power forwards is 3.67313, and the mean is 4.14635.

2. Which skills are more correlated with salary?

Our group defines skills as all the nine elements that consist of EFF. From the Shiny App "stat-salaries," we can find some skills that have something to do with salaries, as follows: total points, total rebounds, assists, steals, missed field goals, and turnovers. However, some skills barely have relationships with salaries, as follows: blocks, missed free throws, and games played. These decisions are purely made through our observation (filtering out the skills that are likely to correlate with salary, before finding the correlations) of the graphs on Shiny App "stat-salaries." For example, when you see the relationship between blocks and salaries below, most of the dots are concentrated near the origin, and many players have big gaps in salary, even though they have similar blocking numbers.

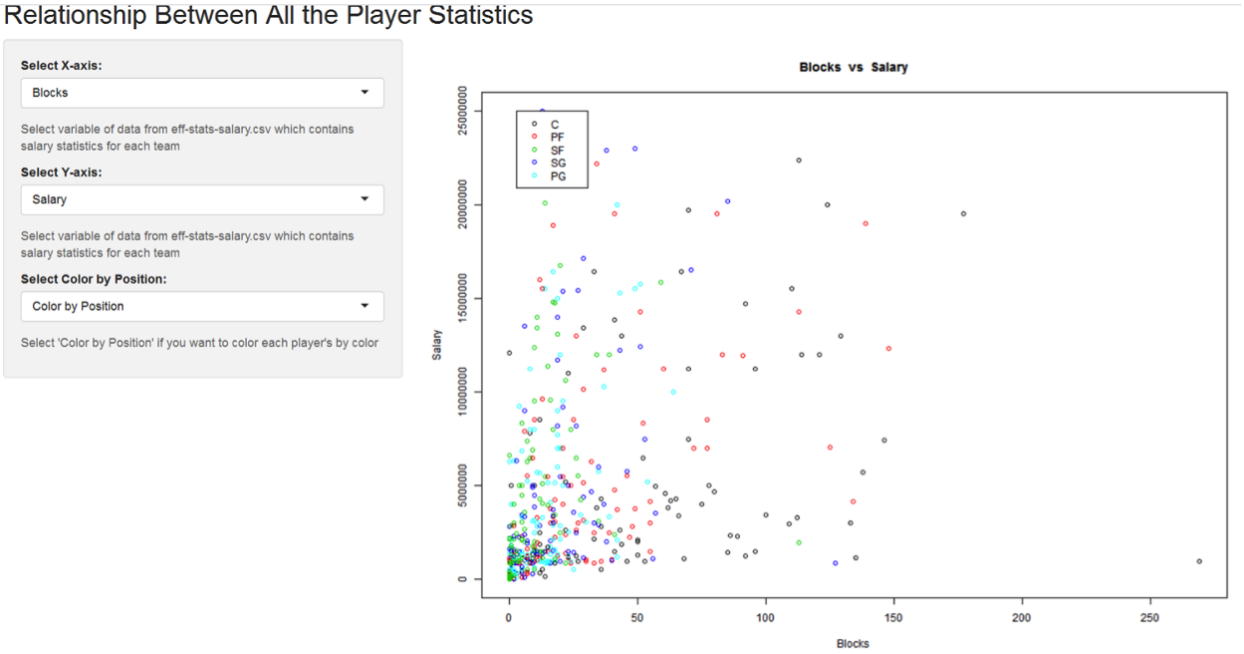


Figure 15. This is one of the Shiny app showing the relationship between blocks and salary. (show how each skill is correlated with salary)

After, our group tried to make relationships between skills (including the unlikely related skills, as well, to make sure our visual predictions were correct) and salaries. We concluded that cor_ point most correlated with salary, with the value of 0.6400 rounded to. And, the least correlated skill with salary was cor_blocks, with the value of 0.3589 rounded to.

We concluded that cor_point most correlated with salary, with the value of 0.6400 rounded to. And, the least correlated skill with salary was cor_blocks, with the value of 0.3589 rounded to.

```
> cor_point
[1] 0.6400331773
> cor_rebounds
[1] 0.5298865676
> cor_assists
[1] 0.5133871695
> cor_steals
[1] 0.4857321402
> cor_blocks
[1] 0.3588732683
> cor_missed_field_goals
[1] 0.5936087256
> cor_missed_free_throws
[1] 0.4496767807
> cor_turnovers
[1] 0.5828069091
> cor_games
[1] 0.3640731545
```

Figure 16. This is the table of the correlation for every skill.

Most correlated with salary, in ascending order, are the following:

- cor_blocks (blocks)
- cor_games (games played)
- cor_ missed_ free_throws (missed free throws)
- cor_steals (steals)
- cor_assists (assists)
- cor_rebounds (rebounds)
- cor_turnovers (turnovers)
- cor_ missed_ field_goals (missed field goals)
- cor_point(point)

Figure 17. These are some of the regressions between salary and other skills.

3. Are there any differences in skills and salary depending on the players' position?

We came up with the most and least correlated skills with the salaries, for each position. By seeing how each position is related to skill, we could tell how each position requires different skills. First of all, the center's salary has the most correlation with points, and the least correlation with steals. Second, the point forward's salary has the most correlation with points, and the least correlation with games. Third, the small forward's salary has the most correlation with assists, and the least correlation with games. Fourth, the shooting guard's salary has the most correlation with assists, and the least correlation with games. Last, the point guard's salary has the most correlation with assists, and the least correlation with blocks. From this analysis, definitely, points and assists are the important factors for every player in every position, for earning a higher salary, and the number of games they played is not that significant a factor.

```
> # Try to find the best and least correlated skills with the salaries for center
> print(y_center[which.max(x_center)])
[1] "cor_point_center"
> print(y_center[which.min(x_center)])
[1] "cor_steals_center"
>
> # Try to find the best and least correlated skills with the salaries for point forward
> print(y_power_forward[which.max(x_power_forward)])
[1] "cor_point_power_forward"
> print(y_power_forward[which.min(x_power_forward)])
[1] "cor_games_power_forward"
>
> # Try to find the best and least correlated skills with the salaries for small forward
> print(y_small_forward[which.max(x_small_forward)])
[1] "cor_assists_small_forward"
> print(y_small_forward[which.min(x_small_forward)])
[1] "cor_games_small_forward"
>
> # Try to find the best and least correlated skills with the salaries for shooting guard
> print(y_shooting_guard[which.max(x_shooting_guard)])
[1] "cor_assists_shooting_guard"
> print(y_shooting_guard[which.min(x_shooting_guard)])
[1] "cor_games_shooting_guard"
>
> # Try to find the best and least correlated skills with the salaries for point guard
> print(y_point_guard[which.max(x_point_guard)])
[1] "cor_assists_point_guard"
> print(y_point_guard[which.min(x_point_guard)])
[1] "cor_blocks_point_guard"
```

Figure 18. It shows the most and least correlated skill of each position.

4. Are the players really worth the amount of money clubs pay for?

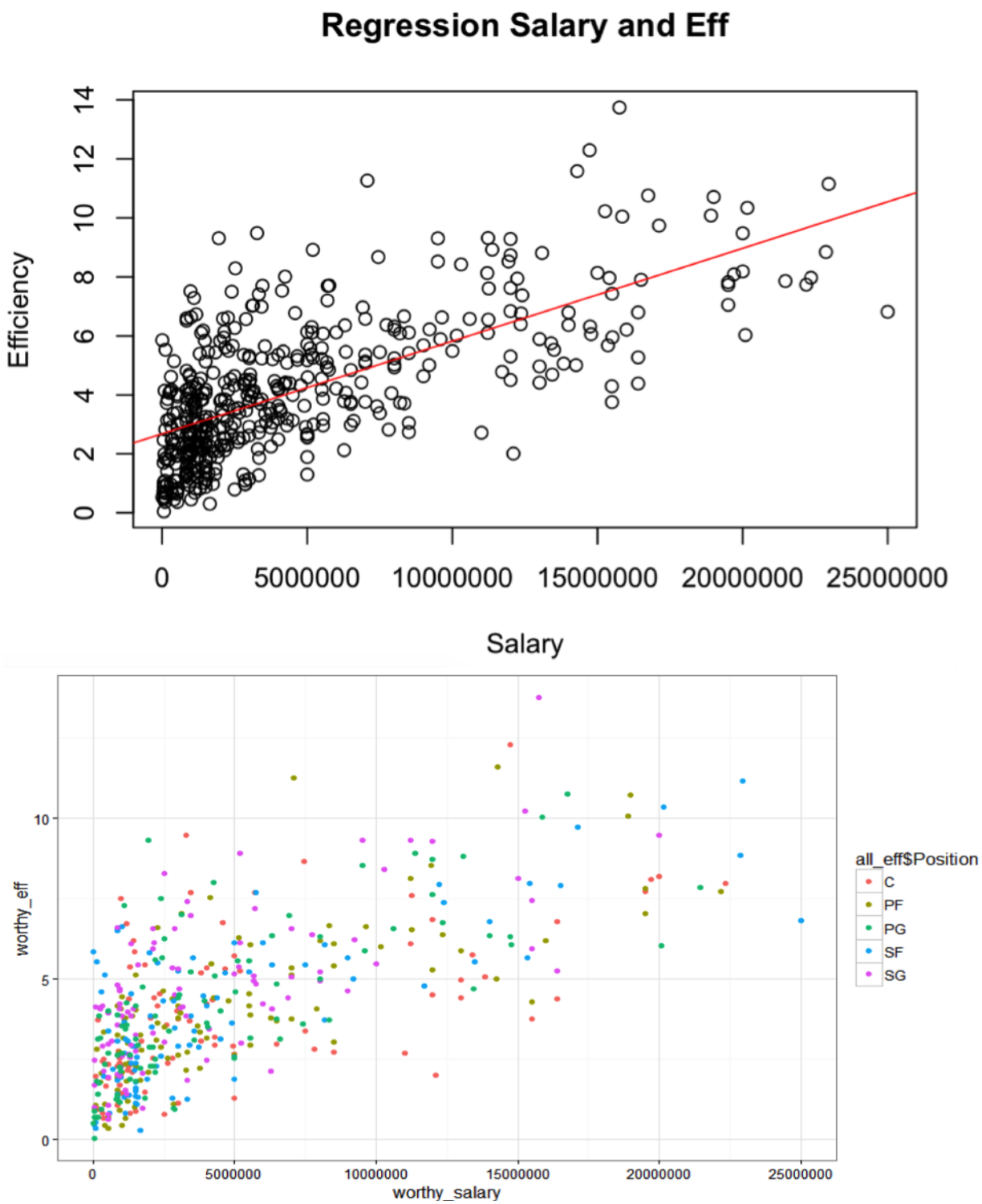


Figure 19, 20. These two show regression between EFF and salary to check worthiness of salary. The first graph includes the regression line in the plot graphs, and the second graph distinguish each plots based on their positions.

We made two graphs to see if players are worthy of their salary: a plot graph with a regression line, and a colored plot graph based on positions. We assume that the regression line indicates

the adequate line between salary and efficiency. Although most of the players are positioned close to the regression line, there are some outliers. As the above graph shows, the numbers above and below the regression line are similar. The points above the line are considered "worthier" players than the clubs pay for, and the points below the line are considered "less worthy" players than the clubs pay for. It is easily shown in the position-colored graph that many centers and small forwards (and slightly the point forwards) are less worthy, and many shooting guards (and slightly the point guards) are worthier. This graph indirectly shows that most of the forwards are earning much higher salaries than the guards who have similar stats.

5. Are there any undervalued or overvalued players?

The value of player is defined by the efficiency and salary. If a player gets paid more than his efficiency, we can consider him as an overvalued player. Conversely, if a player gets paid less than he should, we can consider him as an undervalued player. We have already calculated the value that is efficiency divided with the salary, and the volume of values represent whether the player is getting too much money or too less money. In order to sort undervalued and overvalued players, I ordered all players by the their value and extracted 30 players with top 30 values and 30 players with bottom 30 values.

```
> under_valued_players
   under_valued_30_players.Player under_valued_30_players.EFF under_valued_30_players.Salary under_valued_30_players.value
1                   Dahntay Jones                5.8563691008                          8819                0.000664062716950
2                  Jordan Hamilton                5.5290235405                        119462                0.000046282696929
3                   Xavier Munford                4.1442677966                         89588                0.000046259184228
4                 Jared Cunningham                2.4651861119                         57726                0.000042704952913
5                    Briante Weber                0.5253537922                         12355                0.000042521553395
6                  Orlando Johnson                1.6975061109                         55722                0.000030463840331
7                    Alan Williams                2.8234925423                        114285                0.000024705714156
8                      Chuck Hayes                1.9872034980                         83583                0.000023775211443
9                       Henry Sims                3.7285465964                        161622                0.000023069548678
10                   Jordan Farmar                3.8616598705                        191456                0.000020169960046
11                    Phil Pressey                3.1397161369                        172739                0.000018176069891
12                 Jimmer Fredette                1.0259493435                         59731                0.000017176162186
13                   Lorenzo Brown                1.8324252427                        111444                0.000016442565259
14                 Marcus Thornton                4.0688165019                        251107                0.000016203516835
15                   Nate Robinson                0.7078688443                         44000                0.000016087928280
16                 Elliot Williams                0.8767494599                         55722                0.000015734350165
17                 Michael Beasley                4.6146348372                        306527                0.000015054578674
18                    Axel Toupane                1.8315453172                        123552                0.000014824084735
19                  Alex Stepheson                0.9132653675                         61776                0.000014783497919
```

Figure 21. It is the list of under-valued players.

```
> over_valued_players
   over_valued_30_players.Player over_valued_30_players.EFF over_valued_30_players.Salary over_valued_30_players.value
1                  Nikola Pekovic                2.0048602675                      12100000                0.0000001656909312
2                      Sam Dekker                0.3039566641                       1646400                0.0000001846189651
3                     Roy Hibbert                3.7565469487                      15500000                0.0000002423578677
4                       Omer Asik                2.7122062334                      11000000                0.0000002465642030
5                     Chris Kaman                1.2950284616                       5000000                0.0000002590056923
6                     Enes Kanter                4.3867987068                      16400000                0.0000002674877260
7                     Kobe Bryant                6.8137512085                      25000000                0.0000002725500483
8                       David Lee                4.2949025036                      15493680                0.0000002772035116
9                    Derrick Rose                6.0302822585                      20093063                0.0000003001176206
10                   Joel Anthony                0.7897711805                       2500000                0.0000003159084722
11                 Wesley Matthews                5.2714393162                      16400000                0.0000003214292266
12                  Tiago Splitter                2.7330762001                       8500000                0.0000003215383765
13                    Kirk Hinrich                0.9636482988                       2870000                0.0000003357659578
14                    Jodie Meeks                2.1213650612                       6270000                0.0000003383357354
15                  Tyson Chandler                4.4155989983                      13000000                0.0000003396614614
```

Figure 22. It is the list of over-valued-players.

Note: All the codes for the analysis are belong to the compute-efficiency-script.R script.

Result

The interesting thing we found is that most of the positions' salaries have the least correlation with the number of games they played. It is interesting that the games they played barely have nothing to do with their salaries. The games they played are great proof, from the players in any sport, that they are important in the team; thus, we would expect the more frequently they play, the higher salaries they earn. However, the result turned out to be opposite.

The main result is that players with a relatively high efficiency have a higher salary than other players with a lower efficiency do. In general, salary is determined by a player's performances. One interesting and surprising thing is among the 20 worst valuable players, 9 of them are position "center". (We can see this fact from the best-worst-value-players.txt). Generally, one of the obvious patterns is that players with high salaries have higher efficiency and better performance. The PCA and EFF formula we used to calculate the efficiency are the core of this project. They contribute to help us analyzing data comprehensively.

Also we were surprised that the most related factor was total point among all positions, although three out of five positions showed that the number of assists was the most related factor. Thus, we assume that the total point of power forward and center has a higher influence the correlation than the number of assist does. When we do data analysis in the future, we may still need to use methods similar to what we have used in this project.

Conclusion

We found that points, missed field goals, and rebounds are the top three correlation factors for salary; however, blocks, games played, and missed free throws are the bottom three correlation factors for salary in general. Although we do not consider position a skill, we did find that position is one of the biggest factors that affects the salary of the players. Each player's position has different skills that are most / least correlated with their salaries. Here, we found that points and assists are important factors for their salaries, but the number of games played is not that related to salary. We found that shooting guards have the highest efficiency, and point guards have the lowest efficiency. Finally, there are many shooting forwards who are worthier than what they are paid, and many centers and small forwards that are less worthy than what they are paid. As a result, our analysis would be beneficial to the owner of basketball team. The analysis will provide the value of each player, so that the owner can set the amount of salary strategically. However, we are wondering if top three skills that are highly related to the amount of salary are also related to the percentage of victories. In the future, we would like to investigate the factors that influence the percentage of winning, because the ultimate goal of a game would be winning the game. Also, the owner of a team will be able to reward players properly through our analysis.

From our analysis, although there is a positive relation between players' salaries and their performance and skills, i.e. usually players who get higher salaries have better performances than other players do, such as more total points, and fewer missed field goals and missed free throws, there do exist some exceptions. In the pictures in Shiny App we have shown before, there are some players who have a high salary but who have missed a lot of field goals, and some players who earn an average or low salary actually perform well. So basketball teams should use an accurate statistical method to analyze players' performances as a standard to determine their salaries. They need to see which players are more or less worthy than the others, and distribute salaries more properly.

Reference

link for reference
@bball_ref. “Basketball Statistics and History | Basketball-Reference.com.”
Basketball-Reference.com. SPORTS REFERENCE LLC, 2016. Web. 27 Nov. 2016.