

Introduction

My project, “NBA Shot Wowy”, is a web application that allows users to view basketball shot charts based on what players were on the floor when the shot was taken. The project was inspired by <http://nbawowy.com/>, a site that allows users to view statistics based on the players on the floor. I combined this site’s idea with basic shot charts, which are a staple in basketball. The result I believe is something very unique.

Methods

I used the NBA’s hidden API that they use to populate their statistics website. I had to use a HTTP monitor to figure out the endpoints the API contained, as the NBA does not provide documentation. Once I knew the endpoints, I used the Python Scrapy library to scrape the needed data into a PostgreSQL database. I found that the data containing what players were playing at a particular time for each game on <http://statsnba.s3-website-us-east-1.amazonaws.com/>. This site contained large json files, which I parsed and inserted in my database using Scrapy. The tables I created are listed at the end of this report as well as in the github README. The technologies I used were Flask, Scrapy, SQLAlchemy, PostgreSQL and Twitter Bootstrap.

Functions My System Supports

The home page of my application allows for filtering by the date of the game:

NBA Shot Wowy

Choose Date Games for Wednesday, October 29, 2008

Date	Home Team	Away Team	Action
2008-10-29	Orlando Magic 	Atlanta Hawks 	* View Game
2008-10-29	Philadelphia 76ers 	Toronto Raptors 	* View Game
2008-10-29	Detroit Pistons 	Indiana Pacers 	* View Game
2008-10-29	Washington Wizards 	Brooklyn Nets 	* View Game
2008-10-29	Minnesota Timberwolves 	Sacramento Kings 	* View Game
2008-10-29	Oklahoma City Thunder 	Milwaukee Bucks 	* View Game
2008-10-29	Houston Rockets 	Memphis Grizzlies 	* View Game
2008-10-29	Golden State Warriors 	New Orleans Pelicans 	* View Game

Upon selecting a game you are able to view the statistics players gathered during the game:

NBA Shot Wowy

Friday, October 31, 2008

[View Game Shots](#)

New York Knicks Philadelphia 76ers

87  116 

Philadelphia 76ers

NAME	MIN	FG	3PT	FT	OREB	DREB	REB	AST	STL	BLK	TO	PF	+/-	PTS
A. Iguodala	37:01	3-7	0-1	1-2	1	5	6	7	1	0	1	0	26	7
A. Miller	33:17	8-12	0-0	4-4	1	6	7	8	0	0	4	3	19	20
S. Dalember	32:36	4-8	0-0	0-2	1	8	9	0	0	4	2	3	22	8
E. Brand	30:40	12-19	0-0	0-2	2	12	14	2	0	3	3	2	22	24
W. Green	28:37	8-10	2-2	0-0	0	2	2	1	2	0	0	0	22	18
L. Williams	27:05	6-14	0-1	3-5	1	3	4	4	0	0	2	1	21	15
T. Young	23:50	6-10	0-1	1-2	3	3	6	1	0	0	2	4	13	13
R. Evans	12:57	1-3	0-0	1-2	3	6	9	1	0	0	2	1	-1	3
K. Rush	8:06	1-2	0-0	0-0	0	2	2	0	1	0	2	2	2	2
M. Speights	5:51	3-4	0-0	0-0	1	2	3	0	0	0	1	1	-1	6
R. Ivey	0:00													
D. Marshall	0:00													

New York Knicks

NAME	MIN	FG	3PT	FT	OREB	DREB	REB	AST	STL	BLK	TO	PF	+/-	PTS
C. Duham	34:05	1-8	0-3	3-4	2	4	6	7	3	0	1	2	-23	5
D. Lee	33:05	5-13	0-0	3-4	5	6	11	3	3	1	1	3	-14	13
N. Robinson	31:07	4-14	2-8	2-3	2	4	6	2	3	0	2	3	-15	12
J. Crawford	30:09	5-12	2-6	2-2	1	0	1	1	1	0	1	0	-18	14
Z. Randolph	29:51	5-19	0-1	2-2	4	3	7	2	0	0	1	1	-28	12
W. Chandler	24:22	5-13	1-3	0-0	1	3	4	1	1	0	1	4	-2	11
Q. Richardson	20:10	3-8	2-5	0-0	0	2	2	1	2	0	1	0	-21	8
M. Collins	13:28	3-7	0-2	1-2	0	2	2	0	0	0	0	0	-3	7
M. Rose	8:15	0-1	0-0	1-2	1	2	3	1	0	0	1	2	-5	1
D. Gallinari	7:51	0-1	0-0	2-2	0	1	1	1	0	0	1	0	-14	2
A. Roberson	7:07	1-2	0-0	0-0	0	0	0	0	0	0	0	0	-2	2
E. Curry	0:00													

Upon choosing to view shots for a game you are able to view all the shots taken by each team in the game with options to filter my player and lineup:

Friday, October 31, 2008
Boxscore

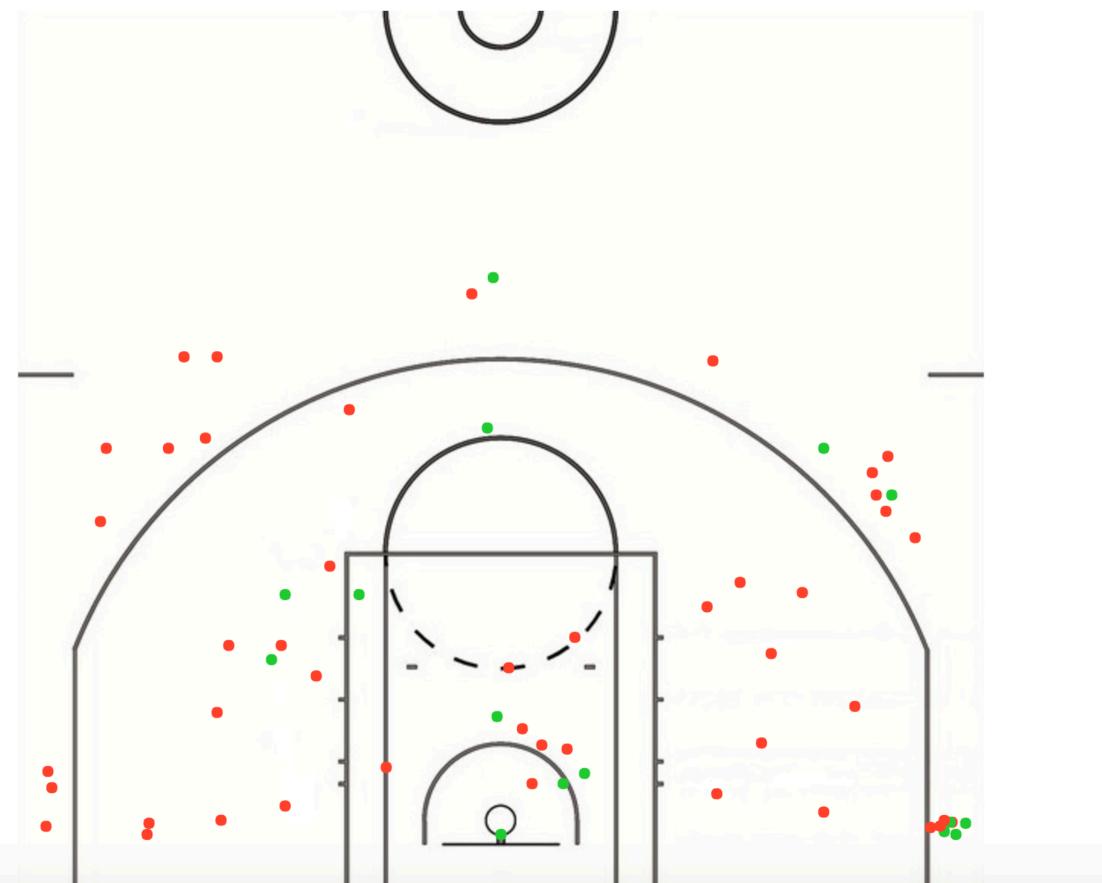
New York Knicks 
87

Philadelphia 76ers 
116

[Change Teams](#)

[All Players](#)

[All Offensive Lineups](#)



Here, we can see an example of how a coach could put my system to use. We are able to cycle through the different lineups David Lee was a part of during the game and reconstruct the seven shots he took. A coach may then notice that, for whatever reason, David shot his 3 longest shots while playing with Crawford, Richardson, Randolph, and Duhon. He missed all three shots.

Friday, October 31, 2008
Boxscore

New York Knicks 
87

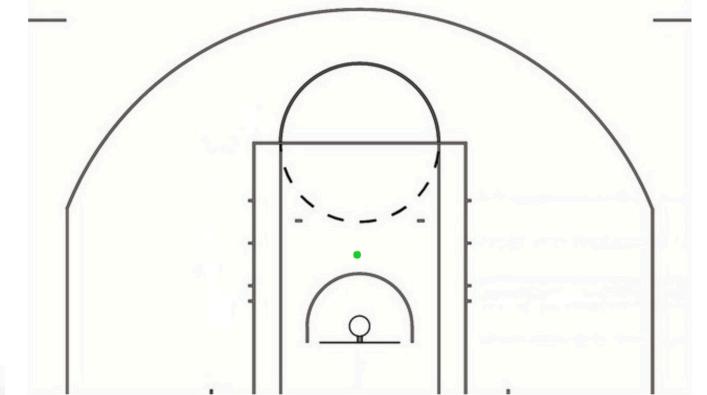
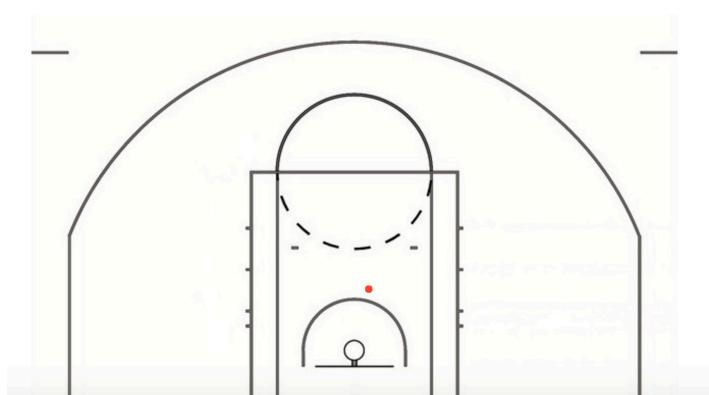
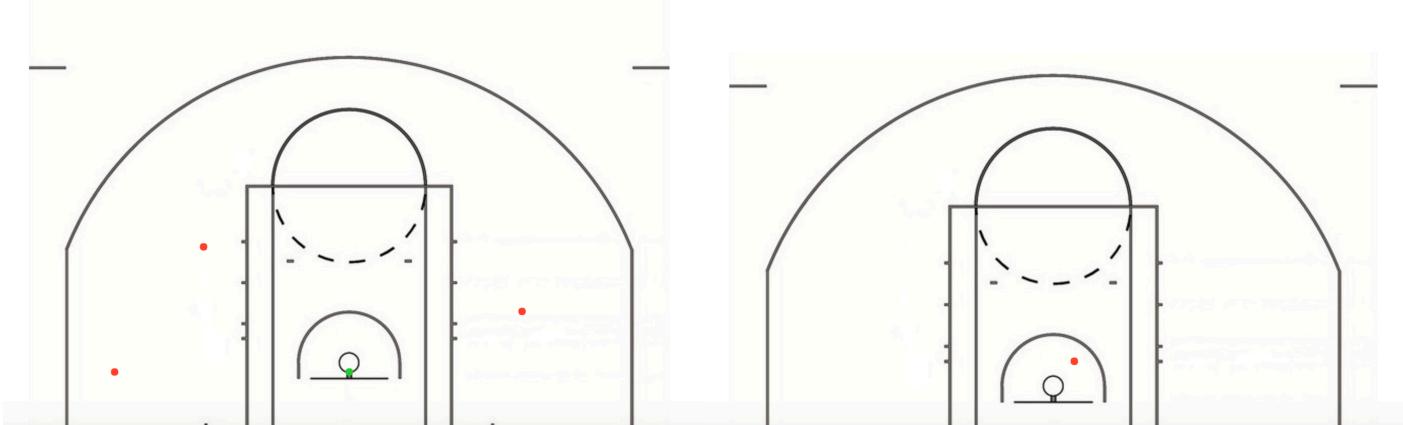
Philadelphia 76ers 
116

Change Teams

David Lee

All Offensive Lineups

J. Crawford, Q. Richardson, Z. Randolph, C. Duhon, D. Lee
J. Crawford, C. Duhon, N. Robinson, D. Lee, W. Chandler
M. Rose, Q. Richardson, Z. Randolph, N. Robinson, D. Lee
J. Crawford, Q. Richardson, Z. Randolph, N. Robinson, D. Lee
N. Robinson, D. Lee, A. Roberson, M. Collins, W. Chandler
N. Robinson, D. Lee, A. Roberson, M. Collins, D. Gallinari



Limitations

My web application has limitations in that you can only filter shot charts by the five offensive players on the floor, however, given the data I have assembled, extending this functionality would be straightforward. The next steps of this project are to extend the shot chart options to include filtering based on a subset of five players as well as filtering based on the defensive players on the floor.

Difficulties

The most difficult parts of this project were figuring out the NBA API's endpoints, scraping the extremely large amount of data, and, most of all, designing my database tables. While designing my tables I really came to appreciate the application of the things I learned in this class. Many of the different calls to the API I used returned duplicate information. Deciding the best place to store each piece of information was a real challenge.

Conclusion

Having followed the basketball analytics community for years, I believe this is the first time anyone in the public space has pieced together lineup and shot data in the way that my system does. In the past people have assembled shot data and people have assembled lineup data, but I do not believe that information has been previously paired. My system does this pairing. Additionally, my database design follows BCNF meaning maintaining and utilizing this data going forward will be straightforward. Additionally, this sort of data is really similar to more sports than just basketball - meaning my system could easily be ported to store and display soccer or any similar sport's data as well.

Tables

players - 4146 rows – (All players who ever played in NBA)

Table "public.players"
Modifiers

Column	Type	Modifiers
player_id	integer	not null default nextval('players_player_id_seq'::regclass)
first_name	character varying	
last_name	character varying	
birthdate	date	
school	character varying	
country	character varying	
last_affiliation	character varying	
height	character varying	
weight	character varying	
season_exp	character varying	
jersey	character varying	
position	character varying	
dleague_flag	character varying	
draft_year	character varying	
draft_round	character varying	
draft_number	character varying	
roster_status	character varying	
from_year	character varying	
to_year	character varying	
team_id	character varying	

Indexes:

"players_pkey" PRIMARY KEY, btree (player_id)

games – 9567 rows (All games 2008-2016)

Table "public.games"				
Column	Type	Modifiers		
game_id	integer	not null default nextval('games_game_id_seq'::regclass)		
game_date	date			
periods	integer			
home_team_id	integer			
visitor_team_id	integer			

Indexes:

- "games_pkey" PRIMARY KEY, btree (game_id)

teams – 30 rows

Table "public.teams"				
Column	Type	Modifiers		
team_id	integer	not null default nextval('teams_team_id_seq'::regclass)		
abbreviation	character varying			
name	character varying			

Indexes:

- "teams_pkey" PRIMARY KEY, btree (team_id)

game_rosters – 238649 rows - (All games 2008-2016)

Table "public.game_rosters"					
Column	Type	Modifiers	Storage	Stats target	Description
game_id	integer	not null	plain		
team_id	integer	not null	plain		
player_id	integer	not null	plain		

Indexes:

- "game_rosters_pkey" PRIMARY KEY, btree (player_id, game_id)

matchups – 283495 rows - (All games 2008-2016)

Table "public.matchups"					
Column	Type	Modifiers	Storage	Stats target	
game_id	integer	not null	plain		
matchup_id	integer	not null	plain		
home_player1_id	integer		plain		
home_player2_id	integer		plain		
home_player3_id	integer		plain		
home_player4_id	integer		plain		
home_player5_id	integer		plain		
away_player1_id	integer		plain		
away_player2_id	integer		plain		
away_player3_id	integer		plain		
away_player4_id	integer		plain		
away_player5_id	integer		plain		

Indexes:

- "matchups_pkey" PRIMARY KEY, btree (game_id, matchup_id)

shots – 3822008 rows - (All shots 1996-2016)

Table "public.shots"					
Column	Type	Modifiers	Storage	Stats target	
game_id	integer	not null	plain		
game_event_id	integer	not null	plain		
player_id	integer		plain		
loc_x	integer		plain		
loc_y	integer		plain		
shot_made_flag	integer		plain		
shot_attempted_flag	integer		plain		

Indexes:

"shots_pkey" PRIMARY KEY, btree (game_id, game_event_id)

event_players – 4944549 rows - (All games 2008-2016)

Table "public.event_players"

Column	Type	Modifiers	Storage	Stats target	Description
game_id	integer	not null	plain		
game_event_id	integer	not null	plain		
matchup_id	integer		plain		

Indexes:

"event_players_pkey" PRIMARY KEY, btree (game_id, game_event_id)