Analyzing NBA Shot Data

COEN 242 - Neil Nguyen

Introduction

- Motivation: I love watching basketball and ESPN’s youtube channel provides many analysis videos of the best players regarding shot location, hit rate, and more. These are very informative and I wanted to try doing it myself.

- Hypothesis: The closer you are to the basket, the more likely it will be to score. Likewise, the farther you are to the basket, the less likely it is to score.

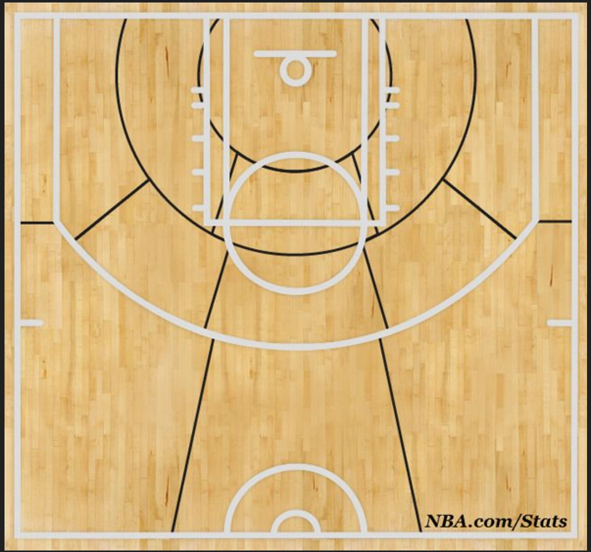
The Players (2020-2021 Regular Season)

Steph Curry:
Long Range
(>22ft)
(>264in)

Chris Paul:
Mid Range
(22ft>x>10ft)
(264in>x>120in)

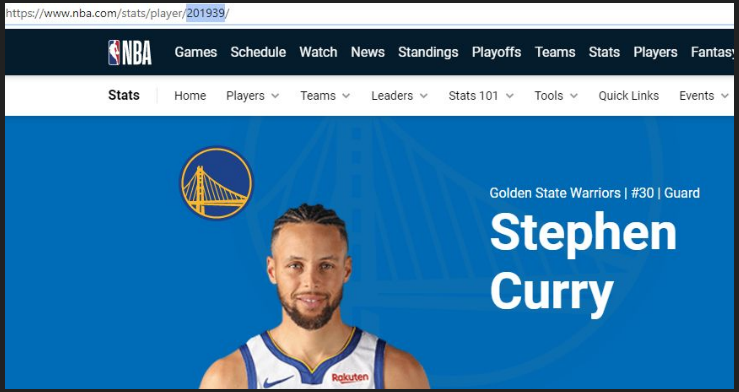
Zion Williamson:
Close Range
(<10ft)
(<120in)

The Ranges



Collecting the Data

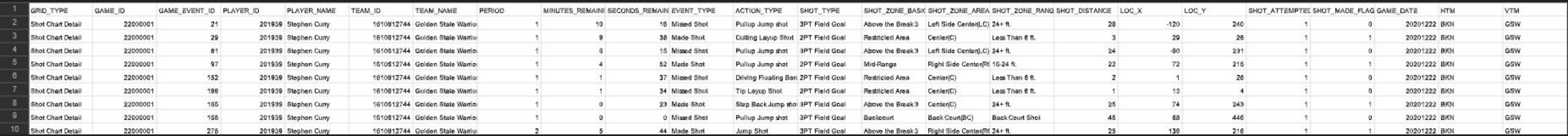
1. Find link for certain player on nba stats
2. Copy the ID# highlighted in the link
3. Run stats_to_txt.py on local pc w/ID#
   - scrapes nba stats site to json
   - converts json to data frame
   - removes irrelevant features
   - converts data frame to list
   - saves as txt file
4. Scp these text files to your design center linux directory
5. Copy to your hdfs directory


https://www.nba.com/stats/player/201939/

Golden State Warriors | #30 | Guard

Stephen Curry

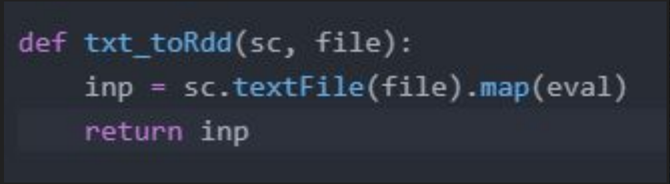
Side Note: “Removing Irrelevant Features”



1	GRID_TYPE	GAME_ID	GAME_EVENT_ID	PLAYER_ID	PLAYER_NAME	TEAM_ID	TEAM_NAME	PERIOD	MINUTES_REMAINI	SECONDS_REMAIN	EVENT_TYPE	ACTION_TYPE	SHOT_TYPE	SHOT_ZONE_BASIC	SHOT_ZONE_AREA	SHOT_ZONE_RANG	SHOT_DISTANCE	LOC_X	LOC_Y	SHOT_ATTEMPTED	SHOT_MADE_FLAG	GAME_DATE	HTM	VTM
2	Shot Chart Detail	22000001	21	201939	Stephen Curry	1610612744	Golden State Warrior	1	10	16	Missed Shot	Pullup Jump shot	3PT Field Goal	Above the Break 3	Left Side Center(LC)	24+ ft.	26	-120	240	1	0	20201222	BKN	GSW
3	Shot Chart Detail	22000001	29	201939	Stephen Curry	1610612744	Golden State Warrior	1	9	38	Made Shot	Cutting Layup Shot	2PT Field Goal	Restricted Area	Center(C)	Less Than 8 ft.	3	29	26	1	1	20201222	BKN	GSW
4	Shot Chart Detail	22000001	81	201939	Stephen Curry	1610612744	Golden State Warrior	1	6	15	Missed Shot	Pullup Jump shot	3PT Field Goal	Above the Break 3	Left Side Center(LC)	24+ ft.	24	-90	231	1	0	20201222	BKN	GSW
5	Shot Chart Detail	22000001	97	201939	Stephen Curry	1610612744	Golden State Warrior	1	4	52	Made Shot	Pullup Jump shot	2PT Field Goal	Mid-Range	Right Side Center(R(	16-24 ft.	22	72	216	1	1	20201222	BKN	GSW
6	Shot Chart Detail	22000001	152	201939	Stephen Curry	1610612744	Golden State Warrior	1	1	37	Missed Shot	Driving Floating Ban	2PT Field Goal	Restricted Area	Center(C)	Less Than 8 ft.	2	1	26	1	0	20201222	BKN	GSW
7	Shot Chart Detail	22000001	196	201939	Stephen Curry	1610612744	Golden State Warrior	1	1	34	Missed Shot	Tip Layup Shot	2PT Field Goal	Restricted Area	Center(C)	Less Than 8 ft.	1	13	4	1	0	20201222	BKN	GSW
8	Shot Chart Detail	22000001	165	201939	Stephen Curry	1610612744	Golden State Warrior	1	0	23	Made Shot	Step Back Jump shot	3PT Field Goal	Above the Break 3	Center(C)	24+ ft.	25	74	243	1	1	20201222	BKN	GSW
9	Shot Chart Detail	22000001	168	201939	Stephen Curry	1610612744	Golden State Warrior	1	0	0	Missed Shot	Pullup Jump shot	3PT Field Goal	Backcourt	Back Court(BC)	Back Court Shot	45	68	446	1	0	20201222	BKN	GSW
10	Shot Chart Detail	22000001	275	201939	Stephen Curry	1610612744	Golden State Warrior	2	5	44	Made Shot	Jump Shot	3PT Field Goal	Above the Break 3	Right Side Center(R(	24+ ft.	25	136	216	1	1	20201222	BKN	GSW

- Lots of features scraped from the NBA stats site
  - In total 24
- For my experiment only 3 are relevant
  - X coordinate
  - Y coordinate
  - hit/miss

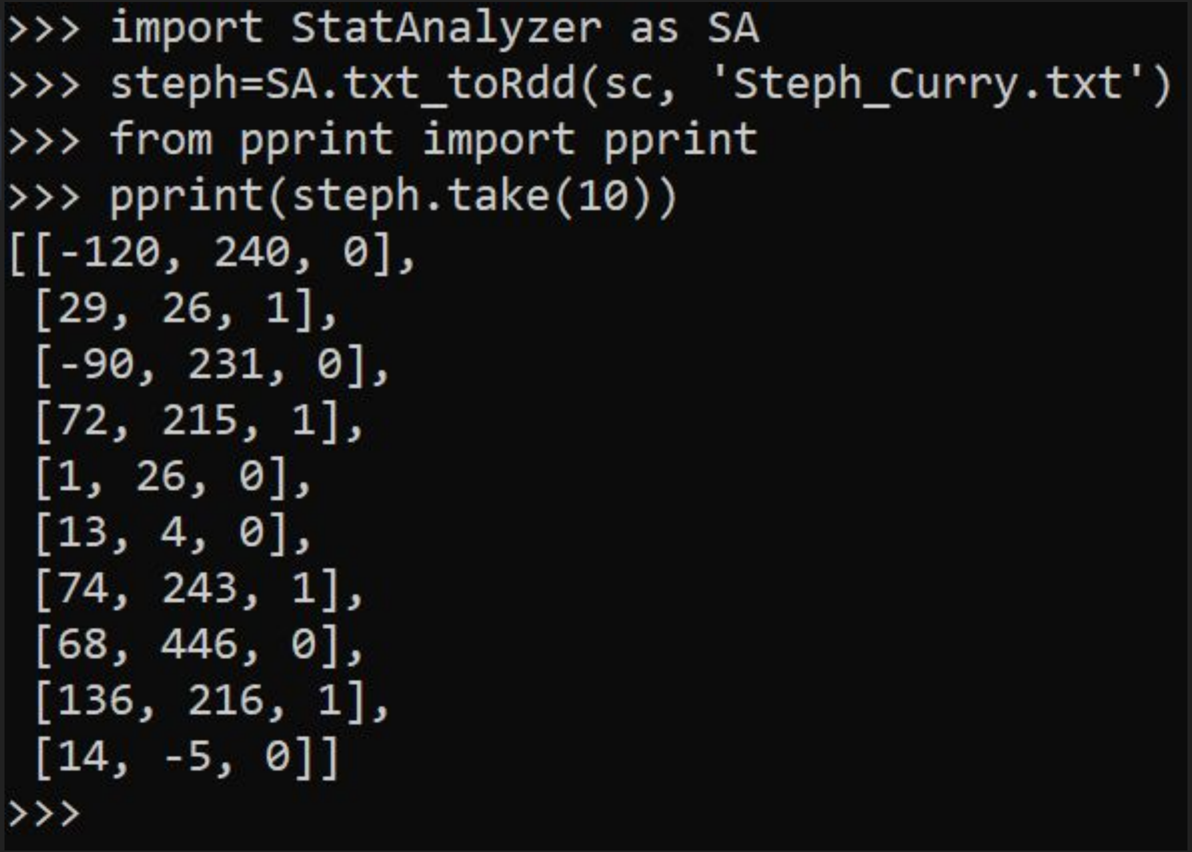
Preparing the Data



```
def txt_toRdd(sc, file):
    inp = sc.textFile(file).map(eval)
    return inp
```

Step 0: Each feature vector in the RDD represents a shot that player has taken. The features are as follows:

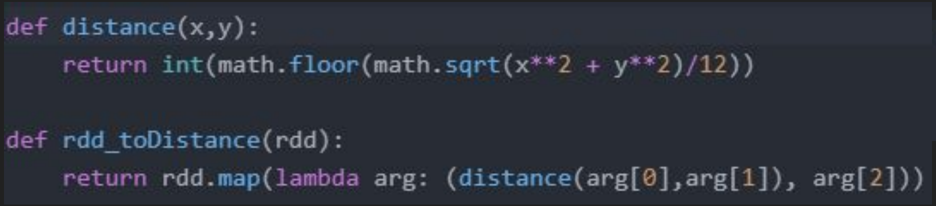
[x coord, y coord, hit/miss]



```
>>> import StatAnalyzer as SA
>>> steph=SA.txt_toRdd(sc, 'Steph_Curry.txt')
>>> from pprint import pprint
>>> pprint(steph.take(10))
[[-120, 240, 0],
 [29, 26, 1],
 [-90, 231, 0],
 [72, 215, 1],
 [1, 26, 0],
 [13, 4, 0],
 [74, 243, 1],
 [68, 446, 0],
 [136, 216, 1],
 [14, -5, 0]]
>>>
```

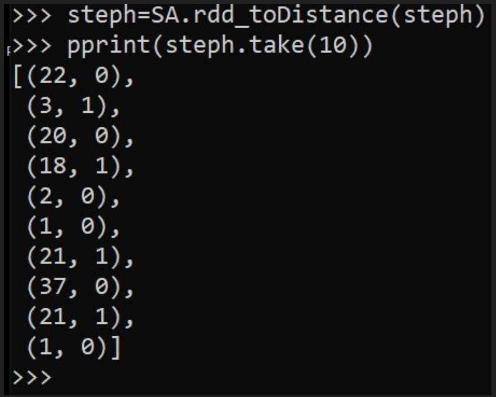
Cleaning the Data

Step 1: Convert x,y coordinates to distances (ft) from (0,0) where the basket is located. Remember the hit/miss data!



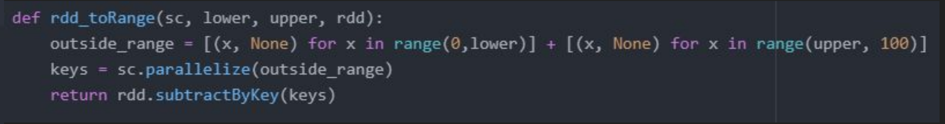
```
def distance(x,y):
    return int(math.floor(math.sqrt(x**2 + y**2)/12))

def rdd_toDistance(rdd):
    return rdd.map(lambda arg: (distance(arg[0],arg[1]), arg[2]))
```

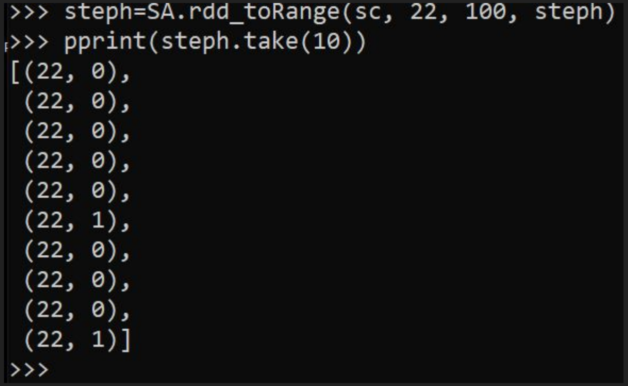


```
>>> steph=SA.rdd_toDistance(steph)
>>> pprint(steph.take(10))
[(22, 0),
 (3, 1),
 (20, 0),
 (18, 1),
 (2, 0),
 (1, 0),
 (21, 1),
 (37, 0),
 (21, 1),
 (1, 0)]
>>>
```

Step 2: Subtract any shots taken outside of the player's given range.



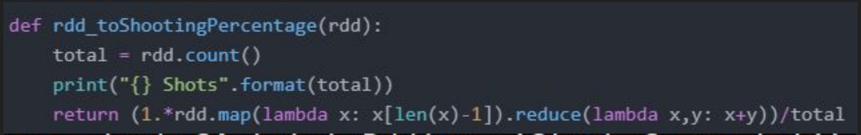
```
def rdd_toRange(sc, lower, upper, rdd):
    outside_range = [(x, None) for x in range(0,lower)] + [(x, None) for x in range(upper, 100)]
    keys = sc.parallelize(outside_range)
    return rdd.subtractByKey(keys)
```



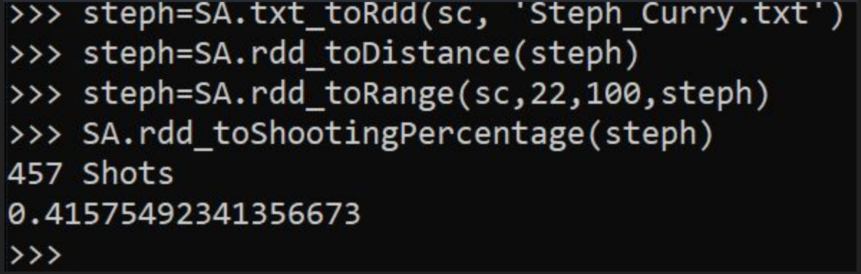
```
>>> steph=SA.rdd_toRange(sc, 22, 100, steph)
>>> pprint(steph.take(10))
[(22, 0),
 (22, 0),
 (22, 0),
 (22, 0),
 (22, 0),
 (22, 1),
 (22, 0),
 (22, 0),
 (22, 0),
 (22, 1)]
>>>
```

Cleaning the Data

Step 3: Calculate shooting percentage of shots in RDD.

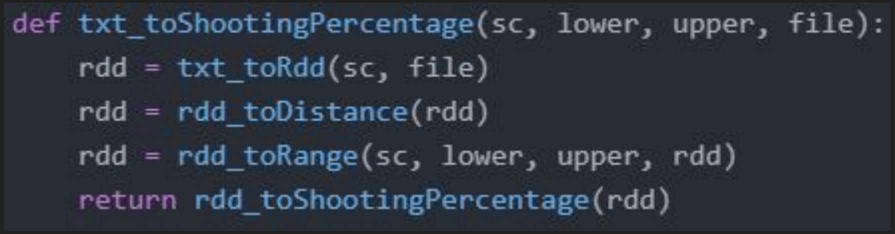


```
def rdd_toShootingPercentage(rdd):
    total = rdd.count()
    print("{} Shots".format(total))
    return (1.*rdd.map(lambda x: x[len(x)-1]).reduce(lambda x,y: x+y))/total
```

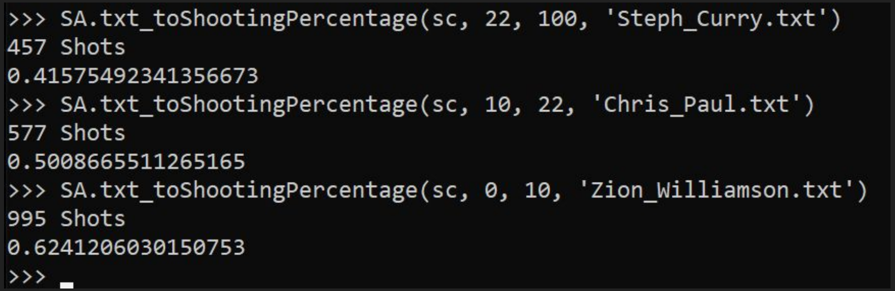


```
>>> steph=SA.txt_toRdd(sc, 'Steph_Curry.txt')
>>> steph=SA.rdd_toDistance(steph)
>>> steph=SA.rdd_toRange(sc,22,100,steph)
>>> SA.rdd_toShootingPercentage(steph)
457 Shots
0.41575492341356673
>>>
```

Step 4: Put all the functions together for convenience.



```
def txt_toShootingPercentage(sc, lower, upper, file):
    rdd = txt_toRdd(sc, file)
    rdd = rdd_toDistance(rdd)
    rdd = rdd_toRange(sc, lower, upper, rdd)
    return rdd_toShootingPercentage(rdd)
```



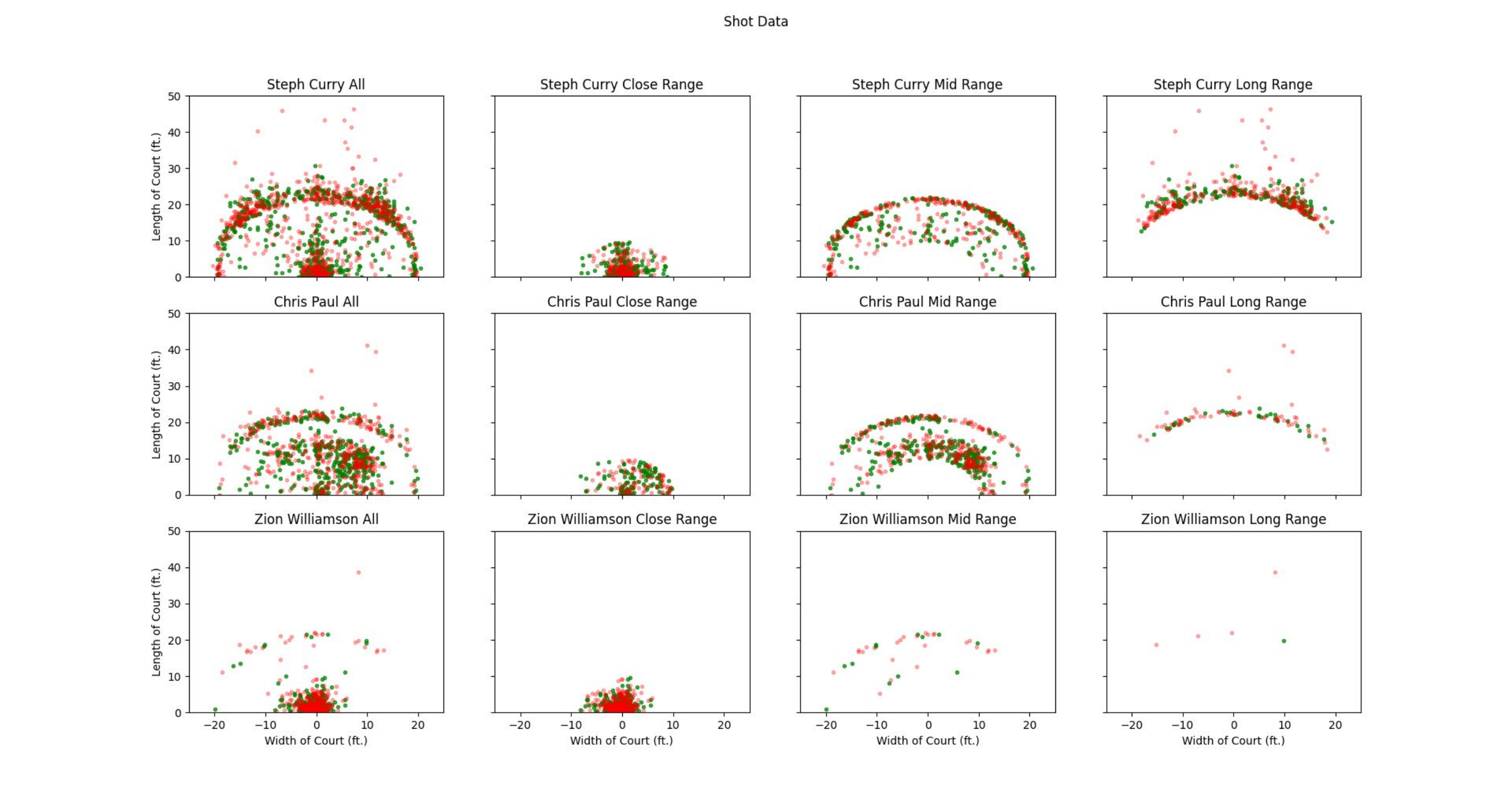
```
>>> SA.txt_toShootingPercentage(sc, 22, 100, 'Steph_Curry.txt')
457 Shots
0.41575492341356673
>>> SA.txt_toShootingPercentage(sc, 10, 22, 'Chris_Paul.txt')
577 Shots
0.5008665511265165
>>> SA.txt_toShootingPercentage(sc, 0, 10, 'Zion_Williamson.txt')
995 Shots
0.6241206030150753
>>> _
```

Analyzing the Data

- The closer these players get to the basket the higher their scoring percentages are
- Zion Williamson has too little mid/long range shots
- Chris Paul has too little long range shots

Player	Range	Count	Percentage
Steph Curry	All	1365	48.21%
Steph Curry	Long	457	41.58%
Steph Curry	Mid	490	43.88%
Steph Curry	Close	418	60.53%
Chris Paul	All	879	49.94%
Chris Paul	Long	93	32.36%
Chris Paul	Mid	577	50.09%
Chris Paul	Close	209	57.42%
Zion Williamson	All	1037	61.13%
Zion Williamson	Long	5	20%
Zion Williamson	Mid	37	32.43%
Zion Williamson	Close	995	62.41%

Graphical Representation of Shot Data



Conclusion

- My hypothesis held true with the data I used
- Despite being the best at their specific ranges, in general these players still shoot better the closer they get to the basket
- How to extrapolate this to the entire NBA?

Insights

- More accurate conclusions can be drawn with more player shot data
  - I.e. The entire NBA
- High volume versatile scorers at every range hold more useful data
  - Steph had 400+ attempts at each range
- Only shot distance was accounted for in this experiment, much more could be addressed
  - Clutch Factor?
  - Defense?
  - Match-ups?

Thank you! Questions?

References:

- https://datavizardry.com/2020/01/28/nba-shot-charts-part-1/