NBA Shot Charts with Excel and Python

By: Shawn Niederriter

I would like to showcase a project I am currently working on to use Python and Excel to create and analyze NBA shot charts, as well as construct a MySQL Database to store this data. Part of my inspiration for this project comes from a post by Savvas Tjortjoglou (http://savvastjortjoglou.com/nba-shot-sharts.html). His post does a great job of showcasing how to build and visualize a shot chart with Python, but I have found further filtering can make these charts extremely impactful. For example, the original post shows only all shots taken, and does not account for makes, misses, or shot type. I am going to show numerous visualizations I created to make these shot charts more useful.

Creating a DataFrame and "Slicing" the Data

I am going to use a CSV file containing Robert Covington's shot data from the 2015-2016 season up to March 11, 2016. The CSV file has 22 columns which have detailed information about the shot. The columns were the base for the "Shot" table in the MySQL database we will discuss later. However, we will only be using 4 of these columns to create charts; "shot_made_flag" (if shot was made "1" or missed "0"), x and y values (represent shot location on the court), and period (what quarter it was). It contains 531 rows, which are shots.

I first created a Pandas DataFrame containing all the information from the CSV file. I then "sliced" the data into three separate frames to make it easier to access. Here is the code I used to create the slices:

```python
import requests
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.colors

# Create DataFrame
df = pd.read_csv('/Users/shawnniederriter/Desktop/NBA/RobertCovington.csv')
# Create Slice of Makes and Misses from DataFrame
shot_data= df[['shot_made_flag','x','y']]
#Create Slice of which Quarter the shot was taken in
quarter_data= df[['period','x','y']]
```

I then recycled some of Savvas Tjortjoglou's code to create a basketball court to plot the data. This court is a coordinate system representing an accurate scale of an NBA court. The X and Y values mentioned above match the actual location a shot was taken.

First, I am going to visualize makes and misses. Using the shot_data variable above, I created a scatter plot using matplotlib to show all shots taken:
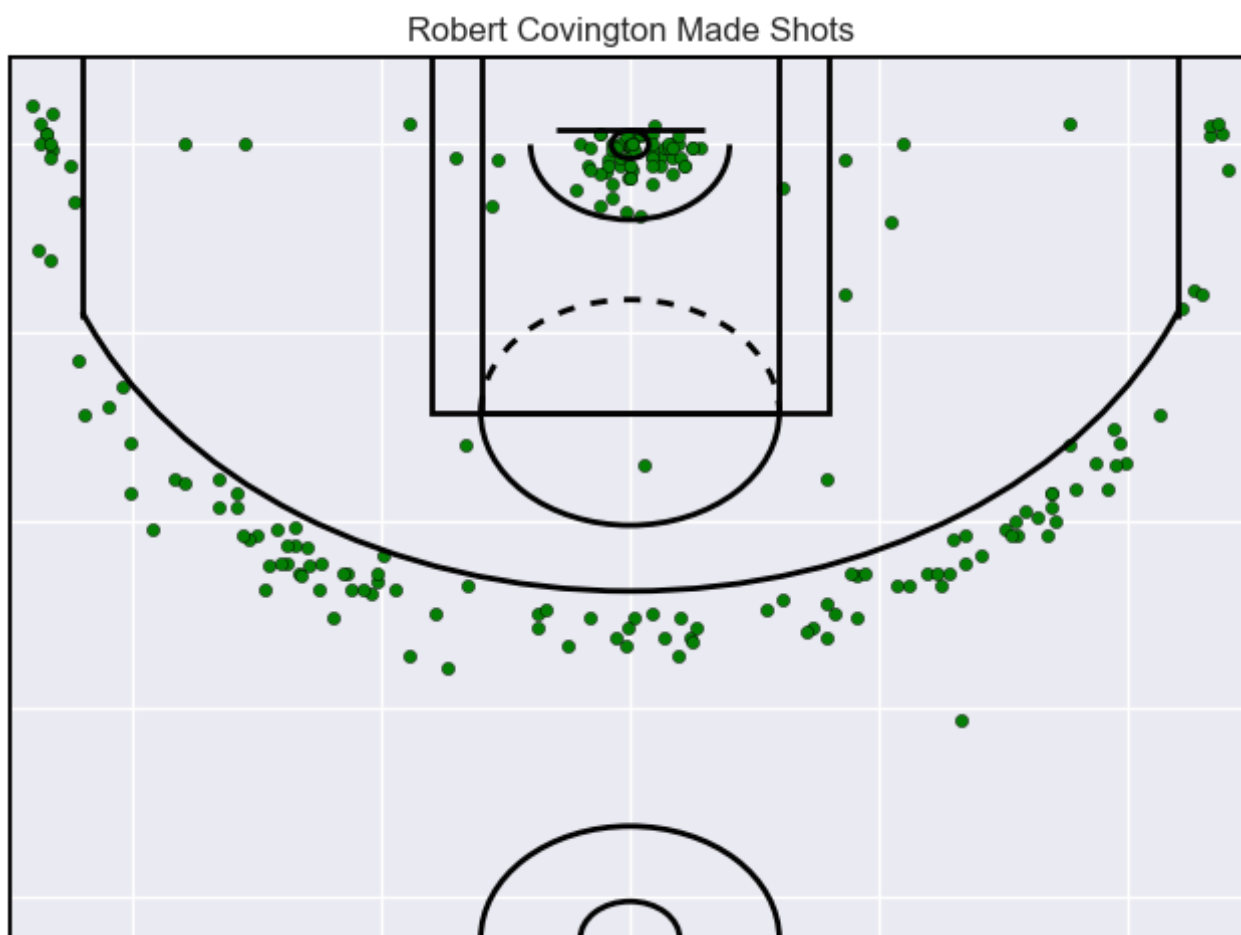

Robert Covington All Shots
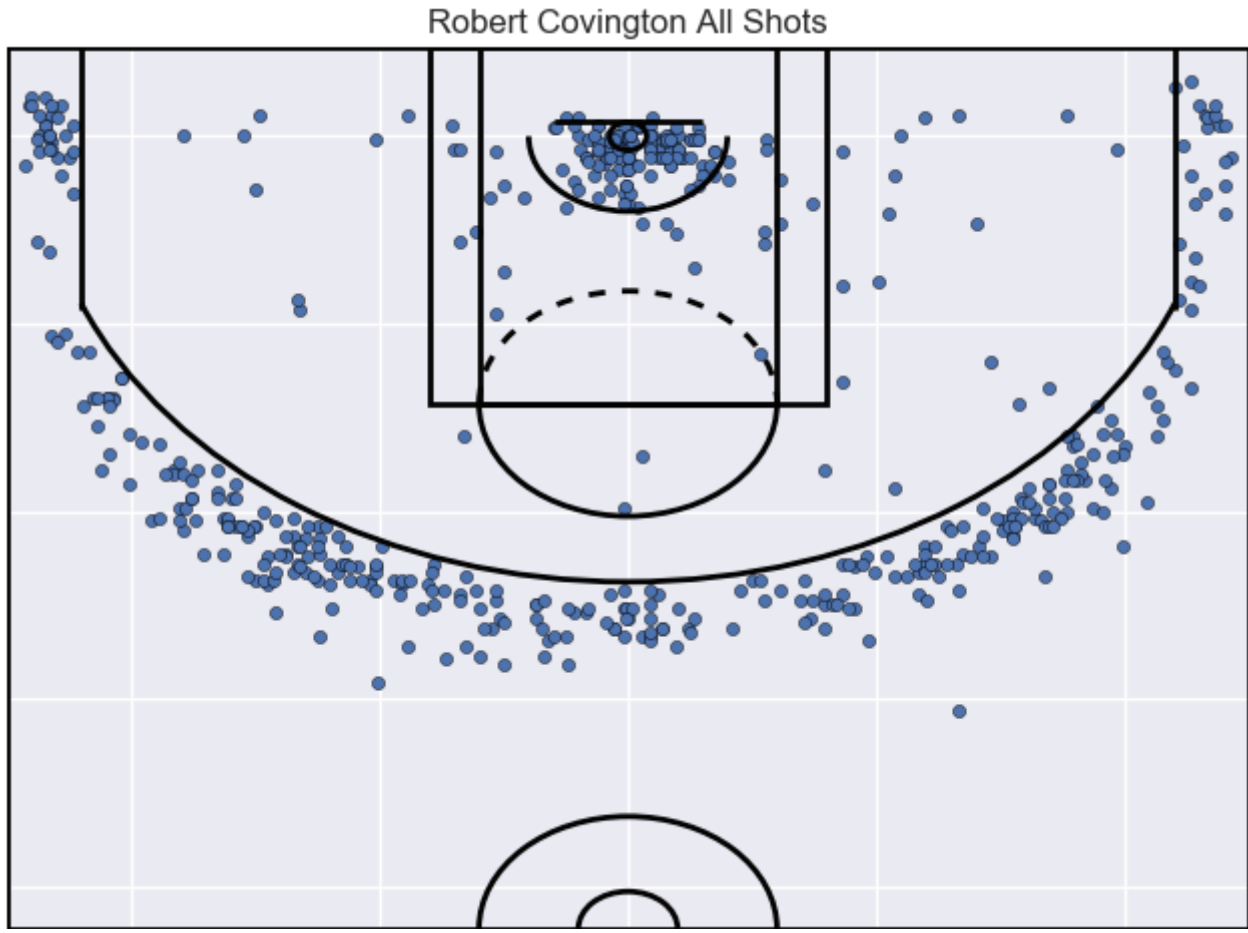
This chart doesn't do much for detailed analysis, but it does show how Covington's shot selection is consistent with the Sixers analytical approach. Notice how he takes virtually no long range two-point shots. The "long two" is often considered the worst analytical shot in basketball. Taking a 19-22 foot two-point shot makes little sense when a player can simply take a step back to attempt a three-pointer.

After this, I wanted to sort the data into makes and misses to give us a better look at Covington's strong and weak shots. I wrote a short function to help sort the data and applied it to the "shot_made_flag" ("1" for yes, "0" for no) column:

```
def mask(shot_data, key, value):
    return shot_data[shot_data[key] == value]

pd.DataFrame.mask = mask

makes= shot_data.mask('shot_made_flag', 1)
misses= shot_data.mask('shot_made_flag', 0)
```

I then plotted out makes and misses separately:

Robert Covington Made Shots

Robert Covington Missed Shots
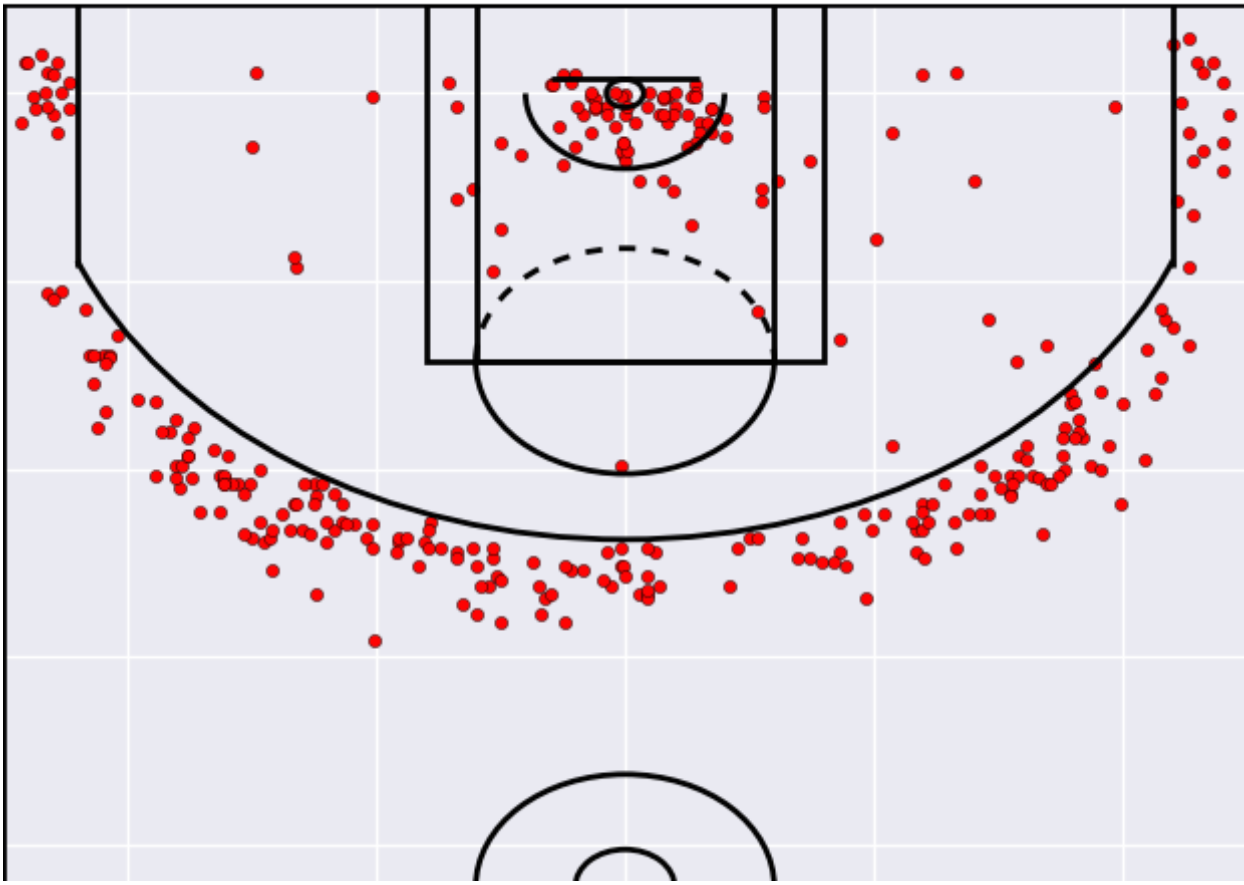
Finally, I plotted makes and misses together. I added a transparency to the shots to help better see clustered data by using the "alpha" parameter of matplotlib's scatter plots. Here is the unified chart, as well as the scatter plot code:

```
ax = makes.plot(kind='scatter', x='x',y='y', c='green', alpha=0.5)
misses.plot(kind='scatter', x='x',y='y', c='red', alpha=0.5, ax=ax)
draw_court(outer_lines=True)
plt.title('Robert Covington Makes and Misses')
ax.set_xlabel('')
ax.set_ylabel('')
ax.tick_params(labelbottom='off', labelleft='off')
plt.xlim(-250,250)
plt.ylim(422.5,-47.5)
plt.show()
```

Robert Covington Makes and Misses
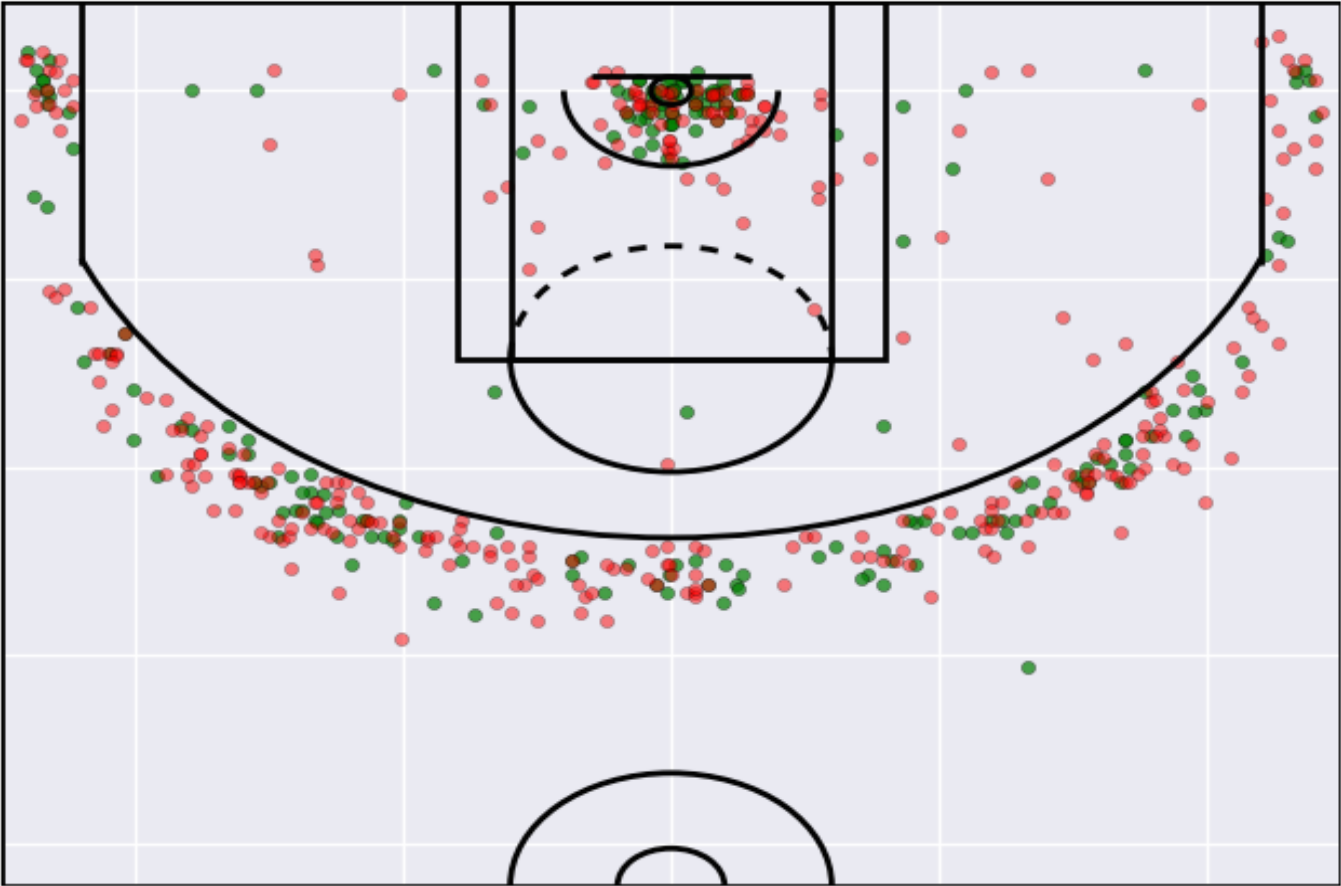
One observation I note from this chart is that Covington doesn't have much success shooting three-pointers from the right-corner. One factor behind this could be the impact of the post player on that side of the court. If he is not getting good passes from the post, it could make shooting more difficult. Nonetheless, I view this as a potential shot area to work on for him.

HexBin Charts

The above shot charts show us a lot about makes and misses. The only problem is, the data can appear clustered at certain points which makes it a little difficult to assess number of shots made by location. One solution for this is to create a Hexagonal Bin chart to show this data in a more summarized form. Luckily, we have already applied a filter function to easily sort the data. To put it on a HexBin chart, we first break the data into individual columns by made, missed, and all shots:

```
x_all = shot_data.x
y_all = shot_data.y

x_made = makes.x
y_made = makes.y

x_miss = misses.x
y_miss = misses.y
```

I then created a "Makes" chart using a Red/Orange/Yellow color scheme. The color bar on the side of the chart shows the number of makes per bin. The lighter the color, the more shots made:

```
madebin = plt.hexbin(x_made, y_made, gridsize=40, cmap=plt.cm.YlOrRd_r, extent=(-250,250,425,-50))
```


Robert Covington Makes - Hexbin
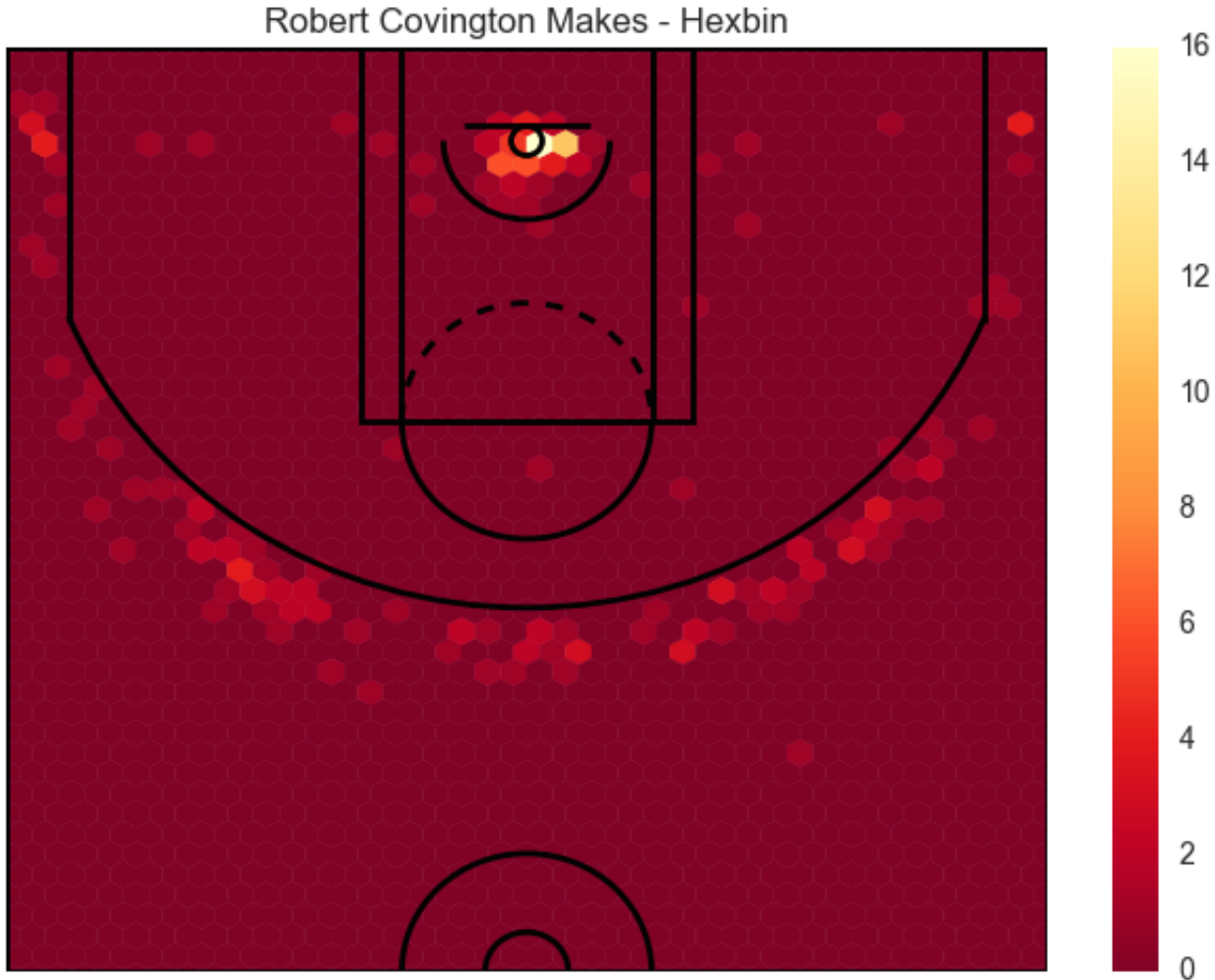
I then chose a Blue/White/Red color scheme for the misses chart. The color bar shows that more red equals more misses per bin:

```
missbin = plt.hexbin(x_miss, y_miss, gridsize=40, cmap=plt.cm.RdBu_r, extent=(-250,250,425,-50))
```


Robert Covington Misses - Hexbin
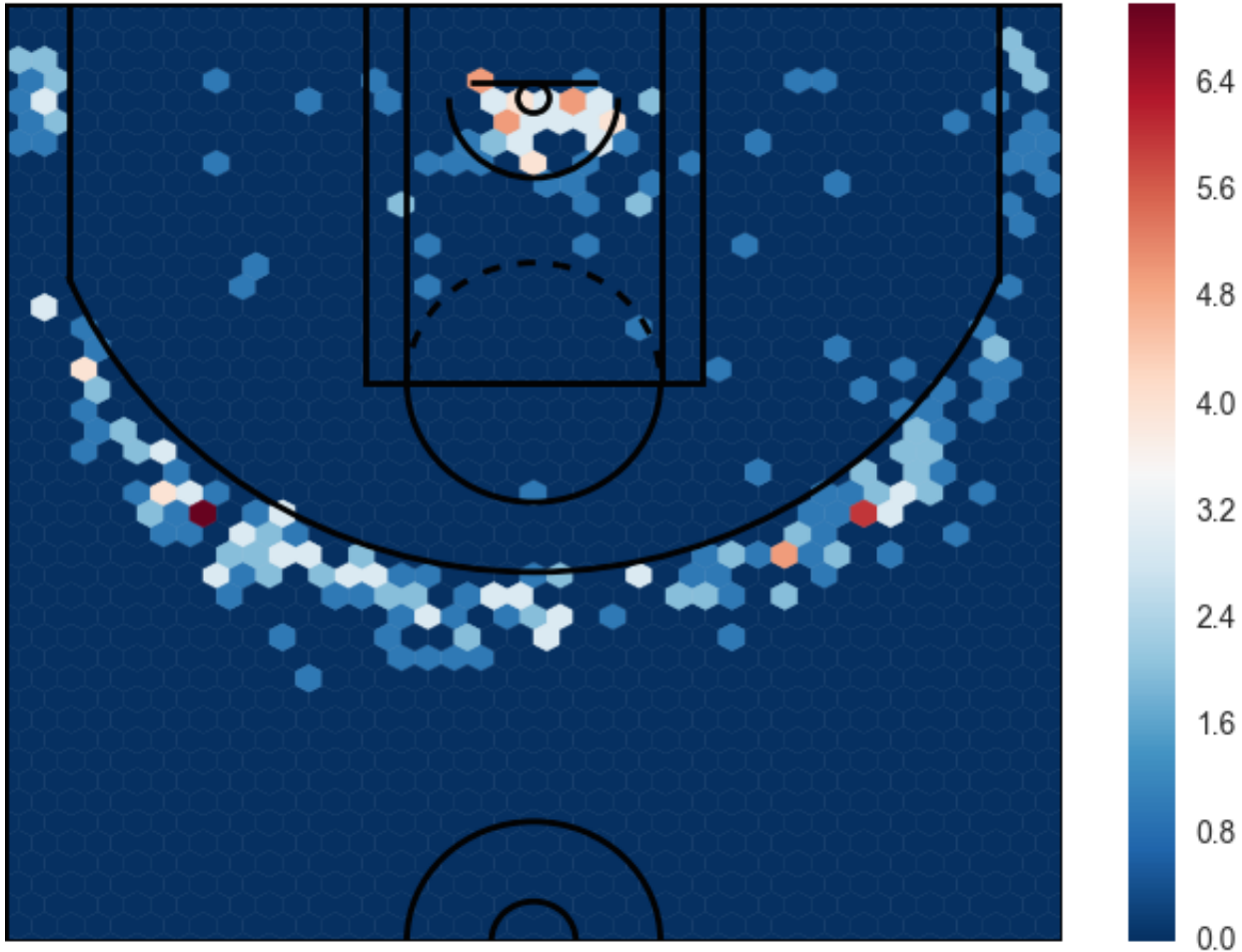
Finally, I wanted to plot this data together to show an effective measure of each bin involving all shots. First, I made sure each bin was measured by the amount of shots per bin. I then put both makes and misses in separate arrays to line up the data, and divided makes by misses to give us a measurement. I used an orange color scheme. The darker spots are Covington's best shot locations:

```
shotbin = plt.hexbin(x_all, y_all, gridsize=40, cmap=plt.cm.Oranges, extent=(-250,250,425,-50))
shotbin.set_array(madebin.get_array()-missbin.get_array())
```

Robert Covington All Shots - Hexbin
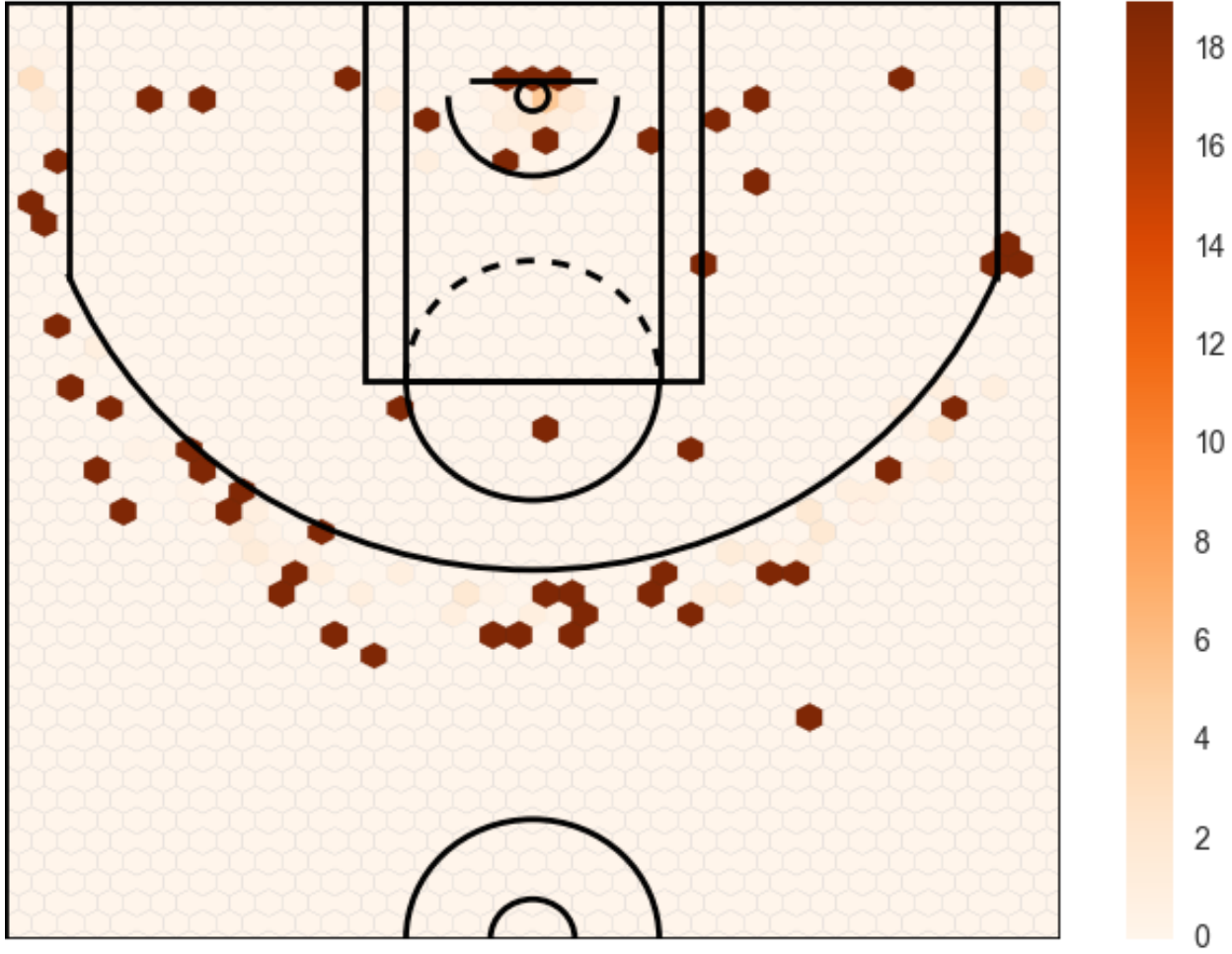
Quarter Shot Charts

I made one final shot chart to show Covington's shots by quarter. I used the "Mask" function we created earlier to filter data by quarter. Here is my code and Shot Chart:

```
first_quarter=quarter_data.mask('period', 1)
second_quarter=quarter_data.mask('period', 2)
third_quarter=quarter_data.mask('period', 3)
fourth_quarter=quarter_data.mask('period', 4)
```

Robert Covington Shots By Quarter

● First Quarter - 137
● Second Quarter - 139
● Third Quarter - 115
● Fourth Quarter - 134
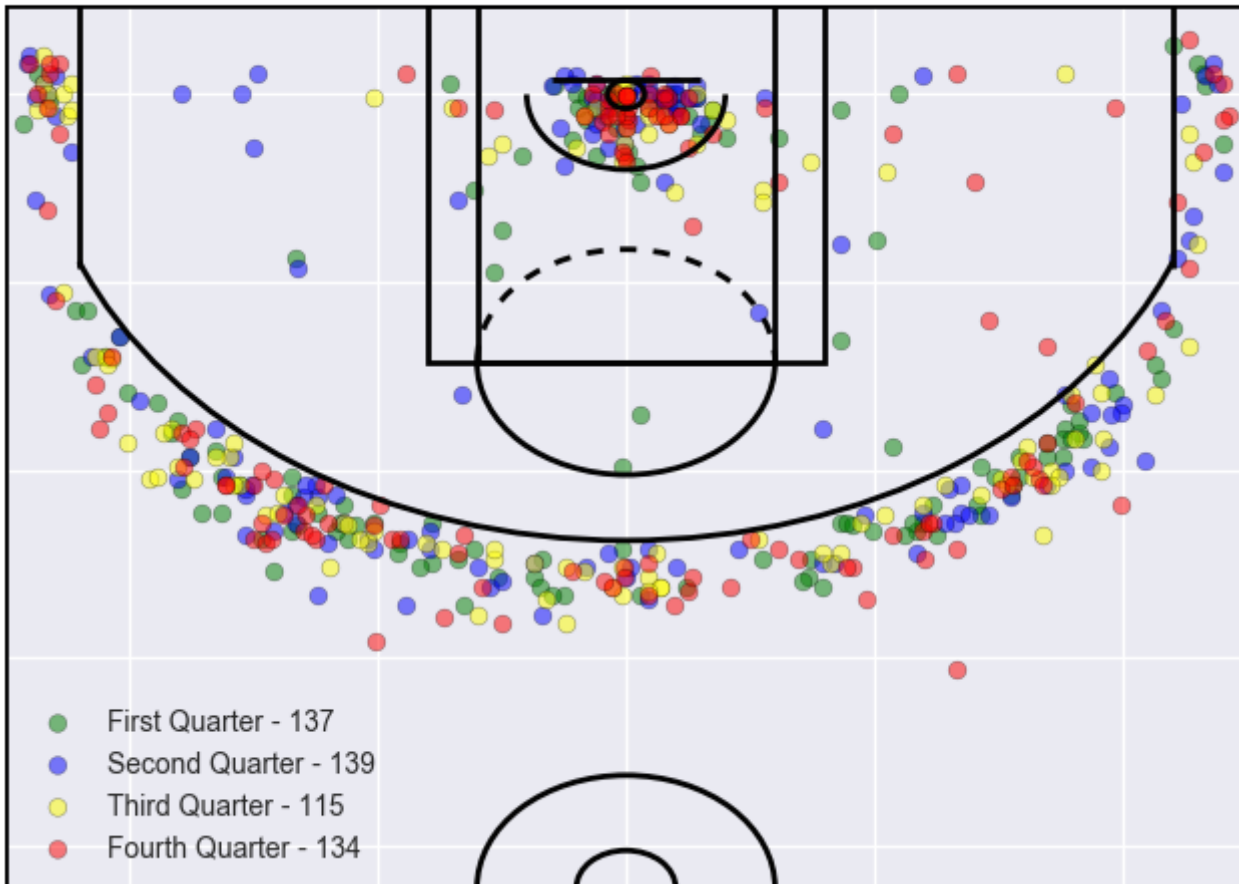
I found it very interesting that Covington has taken 27 more shots in the second half. This chart also shows he takes a lot of close-range shots in the fourth quarter. At the same time, his perimeter shooting is pretty spread out per quarter.

MySQL Database
This CSV file we used for the above shot charts is the base for a table I am building into a MySQL database. I believe this database will be useful so I can easily insert new shot data, as well as easily update and filter the data for more in-depth analysis. Here is the schema:

SixersDB Schema

SixersDB.GameStats_Basic
- GameStats_BasicID INT
- GameID INT
- PlayerID INT
- Minutes_Played DECIMAL(4,2)
- Field_Goals INT(2)
- Field_Goals_Attempted INT(2)
- Three_Pointers INT(2)
- Three_Pointers_Attempted INT(2)
- Free_Throws INT(2)
- Free_Throws_Attempted INT(2)
- O_Rebounds INT(2)
- D_Rebounds INT(2)
- Assists INT(2)
- Steals INT(2)
- Blocks INT(2)
- Fouls INT(1)
- Plus_Minus INT(2)
- Indexes

SixersDB.Game
- GameID INT
- Home_Away VARCHAR(1)
- Opponent VARCHAR(3)
- Win_Loss VARCHAR(1)
- Date DATETIME
- Sixers_Score INT(3)
- Opponent_Score INT(3)
- Indexes

SixersDB.Shots
- ShotID INT
- PlayerID INT
- GameID INT
- Quarter INT(1)
- Mins_Remaining INT(2)
- Secs_Remaining INT(2)
- Shot_Made_Flag BOOLEAN
- Action_Type VARCHAR(45)
- Shot_Type VARCHAR(45)
- Shot_Distance DECIMAL(4,1)
- X_Range INT(4)
- Y_Range INT(4)
- Dribbles INT(2)
- Touch_Time DECIMAL(3,1)
- Defender_Distance DECIMAL(3,1)
- Shot_Clock DECIMAL(3,1)
- Indexes

SixersDB.Player
- PlayerID INT
- First_Name VARCHAR(45)
- Last_Name VARCHAR(45)
- Height INT(2)
- Weight INT(3)
- Birthdate DATETIME
- Position VARCHAR(2)
- Jersey_Number INT(2)
- Experience INT(2)
- College VARCHAR(45)
- Indexes

SixersDB.GameStats_Advanced
- GameStats_AdvancedID INT
- GameID INT
- PlayerID INT
- True_Shoot_Percent DECIMAL(3,3)
- Effective_Field_Goal_Percent DECIMAL(3,3)
- Three_Point_Attempt_Rate DECIMAL(3,3)
- Free_Throw_Attempt_Rate DECIMAL(3,3)
- O_Rebound_Percent DECIMAL(4,1)
- D_Rebound_Percent DECIMAL(4,1)
- Total_Rebound_Percent DECIMAL(4,1)
- Assist_Percent DECIMAL(4,1)
- Steal_Percent DECIMAL(4,1)
- Block_Percent DECIMAL(4,1)
- Turnover_Percent DECIMAL(4,1)
- Usage_Percent DECIMAL(4,1)
- O_Rating INT(3)
- D_Rating INT(3)
- Value_Over_Replacement DECIMAL(4,2)
- Player_Efficiency_Rating DECIMAL(4,2)
- Indexes

GameID

PlayerID

PlayerID

GameID

GameID

PlayerID

This schema has three tables, GameStats_Basic, GameStats_Advanced, and Shots, that hold and track detailed statistics about a player. Player and Game are reference tables, containing general information about a player or game. GameID and PlayerID are foreign keys in all three statistics tables. This allows the statistics tables to be easily queried for a specific player or game by a simple join.

Final Thoughts and Future Plans

I had a lot of fun working on this project, and it taught me a lot about visualizing data. I will continue to add and improve on this project. For example, I plan on using these shot charts as templates so I can easily show data for different players, and do even more visualizations of the shots for aspects like, shot clock time remaining, shot type, and player touch time. Following the completion of the SixersShotDB, I am going to implement a Python script to run these shot charts every time new data is added by a player or game. My final and long-term goal for this project is to expand the database to include all 30 teams and their players. From there, I can create a web app that will allow fans to easily search, view, and compare shot charts of many different players.