augustine-xu-project_part2

April 24, 2021

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import ipywidgets
```

[2]:
```
df = pd.read_csv("Stephen_curry_shooting.csv")
df
```

[2]:

	name	team_name	game_date	season	…
0	Stephen Curry	Golden State Warriors	2015-12-05	2015	
1	Stephen Curry	Golden State Warriors	2016-04-09	2015	
2	Stephen Curry	Golden State Warriors	2017-11-19	2017	
3	Stephen Curry	Golden State Warriors	2018-02-12	2017	
4	Stephen Curry	Golden State Warriors	2018-01-03	2017	
…	…	…	…	…	
7219	Stephen Curry	Golden State Warriors	2012-12-18	2012	
7220	Stephen Curry	Golden State Warriors	2015-01-14	2014	
7221	Stephen Curry	Golden State Warriors	2013-02-12	2012	
7222	Stephen Curry	Golden State Warriors	2014-11-02	2014	
7223	Stephen Curry	Golden State Warriors	2017-03-21	2016	

	espn_player_id	team_id	espn_game_id	period	minutes_remaining	…
0	3975	1610612744	400828177.0	3	0	
1	3975	1610612744	400829077.0	3	8	
2	3975	1610612744	400974987.0	3	9	
3	3975	1610612744	400975592.0	2	3	
4	3975	1610612744	400975308.0	1	3	
…	…	…	…	…	…	…
7219	3975	1610612744	400278085.0	1	6	
7220	3975	1610612744	400578874.0	1	9	
7221	3975	1610612744	400278494.0	2	3	
7222	3975	1610612744	400578334.0	4	2	
7223	3975	1610612744	400900450.0	3	7	

	seconds_remaining	…	shot_type	shot_distance	…
0	2	…	2PT Field Goal	8	
1	11	…	2PT Field Goal	3	

```
2                    27  …  2PT Field Goal             7
3                    36  …  2PT Field Goal            14
4                    11  …  2PT Field Goal             8
…                   …  …              …           …
7219                 42  …  2PT Field Goal            19
7220                 21  …  2PT Field Goal            14
7221                 18  …  2PT Field Goal            19
7222                 16  …  2PT Field Goal            20
7223                 24  …  2PT Field Goal             8

                    opponent    x    y  dribbles  touch_time    defender_name  \
0            Toronto Raptors   71   46       1.0         2.0     Joseph, Cory
1          Memphis Grizzlies   32   16       0.0         0.0              NaN
2              Brooklyn Nets   67   25       0.0         0.0              NaN
3               Phoenix Suns  130   67       0.0         0.0              NaN
4           Dallas Mavericks  -86   25       0.0         0.0              NaN
…                        …  …  …       …         …              …
7219   New Orleans Pelicans -146  127       NaN         NaN              NaN
7220              Miami Heat  -73  126       1.0         2.0  Napier, Shabazz
7221        Houston Rockets -141  132       NaN         NaN              NaN
7222  Portland Trail Blazers  190   83       7.0         6.0  Lillard, Damian
7223        Dallas Mavericks  -37   72       0.0         0.0              NaN

      defender_distance shot_clock
0                   1.1        0.0
1                   0.0        0.0
2                   0.0        0.0
3                   0.0        0.0
4                   0.0        0.0
…                  …        …
7219                NaN        NaN
7220                4.0       22.1
7221                NaN        NaN
7222                2.9       18.2
7223                0.0        0.0

[7224 rows x 22 columns]
```

[6]:
```
df.columns#check column names
```

[6]:
```
Index(['name', 'team_name', 'game_date', 'season', 'espn_player_id', 'team_id',
       'espn_game_id', 'period', 'minutes_remaining', 'seconds_remaining',
       'shot_made_flag', 'action_type', 'shot_type', 'shot_distance',
       'opponent', 'x', 'y', 'dribbles', 'touch_time', 'defender_name',
       'defender_distance', 'shot_clock'],
      dtype='object')
```

[7]:
```
np.unique(df['season'])#unique seasons
```

[7]:
```
array([2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017], dtype=int64)
```

[3]:
```
season_num = np.unique(df['season'])
season_num = np.append(season_num,'All')
season_2010 = df[df['season'] == 2010]
season_2011 = df[df['season'] == 2011]
season_2012 = df[df['season'] == 2012]
season_2013 = df[df['season'] == 2013]
season_2014 = df[df['season'] == 2014]
season_2015 = df[df['season'] == 2015]
season_2016 = df[df['season'] == 2016]
season_2017 = df[df['season'] == 2017]
season_dict = {'2010' : season_2010,
         '2011' : season_2011,
         '2012' : season_2012,
         '2013' : season_2013,
         '2014' : season_2014,
         '2015' : season_2015,
         '2016' : season_2016,
         '2017' : season_2017,
         'All': df}
```

[8]:
```
np.unique(df['shot_type'])#unique shot types
```

[8]:
```
array(['2PT Field Goal', '3PT Field Goal'], dtype=object)
```

[4]:
```
shot_type = np.unique(df['shot_type'])
shot_type = np.append(shot_type, 'All')
```

[19]:
```
@ipywidgets.interact(Season=season_num,Color=plt.
 ↪colormaps(),Shot_type=shot_type)
#Code from https://towardsdatascience.com/
 ↪make-a-simple-nba-shot-chart-with-python-e5d70db45d0d to draw the basketball␣
 ↪court.
def create_court(Season,Color,Shot_type):
    # Draw basketball court
    fig = plt.figure(figsize=(12.7, 9.4))
    ax = fig.add_axes([0, 0, 1, 1])
    ax.plot([-220, -220], [0, 140], linewidth=2, color='black')
    ax.plot([220, 220], [0, 140], linewidth=2, color='black')
    ax.add_artist(mpl.patches.Arc((0, 140), 440, 315, theta1=0, theta2=180,␣
 ↪facecolor='none', edgecolor='black', lw=2))
    ax.plot([-80, -80], [0, 190], linewidth=2, color='black')
    ax.plot([80, 80], [0, 190], linewidth=2, color='black')
    ax.plot([-60, -60], [0, 190], linewidth=2, color='black')
```

```
    ax.plot([60, 60], [0, 190], linewidth=2, color='black')
    ax.plot([-80, 80], [190, 190], linewidth=2, color='black')
    ax.add_artist(mpl.patches.Circle((0, 190), 60, facecolor='none',␣
↪edgecolor='black', lw=2))
    ax.add_artist(mpl.patches.Circle((0, 60), 15, facecolor='none',␣
↪edgecolor='black', lw=2))
    ax.plot([-30, 30], [40, 40], linewidth=2, color='black')
    ax.set_xticks([])
    ax.set_yticks([])

    # Set axis limits
    ax.set_xlim(-250, 250)
    ax.set_ylim(0, 470)
    #Below is my revision on the code to fit my interactive visualizations
    if Shot_type == 'All':#plot all shots
        plot = season_dict[Season]
        title = 'Stephen Curry Field Goal Chart for '
    else:#plot 2PT or 3PT
        plot = season_dict[Season][season_dict[Season]['shot_type'] ==␣
↪Shot_type]
        title = 'Stephen Curry ' + Shot_type +' Chart for '
    #hexbin plot
    hb = ax.hexbin(plot['x'], plot['y'] + 60, gridsize=(30, 30), extent=(-300,␣
↪300, 0, 940), bins='log', cmap=Color)
    if Season == 'All':#set title for all seasons
        ax.set_title(title + '2010-2011 to 2017-2018 Season' )
    else:#set title for individual season
        temp = int(Season) + 1
        ax.set_title(title + str(Season) + '-'+ str(temp) + ' Season' )
    #hexcbin legends
    cb = fig.colorbar(hb, ax=ax)
    cb.set_label('counts')
    mpl.rcParams['font.family'] = 'Avenir'
    mpl.rcParams['font.size'] = 18
    mpl.rcParams['axes.linewidth'] = 2
    plt.show()

#I used the code from https://towardsdatascience.com/
↪make-a-simple-nba-shot-chart-with-python-e5d70db45d0d to draw the basketball␣
↪court.
#I did some revisions on the code to fit my interactive visualization.
```

```
interactive(children=(Dropdown(description='Season', options=('2010', '2011',␣
↪'2012', '2013', '2014', '2015', …
```

The interactive dashboard is to present the shot chart for NBA player Stephen Curry from 2010-2011 season to 2017-2018 season. The Season is used to select any single season to plot or to plot all seasons from 2010 to 2018. The color is for user to choose the colormaps they like to see. The

colormap is all avaible colormap from Matplotlib library. The Shot_type allows user to choose 2PT or 3PT to plot or all types of shot.

Contextual Variables Description Season 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, All Shot_type 2PT Field Goal, 3PT Field Goal, All Color Accent, Accent_r, Blue, Blue_r, etc.