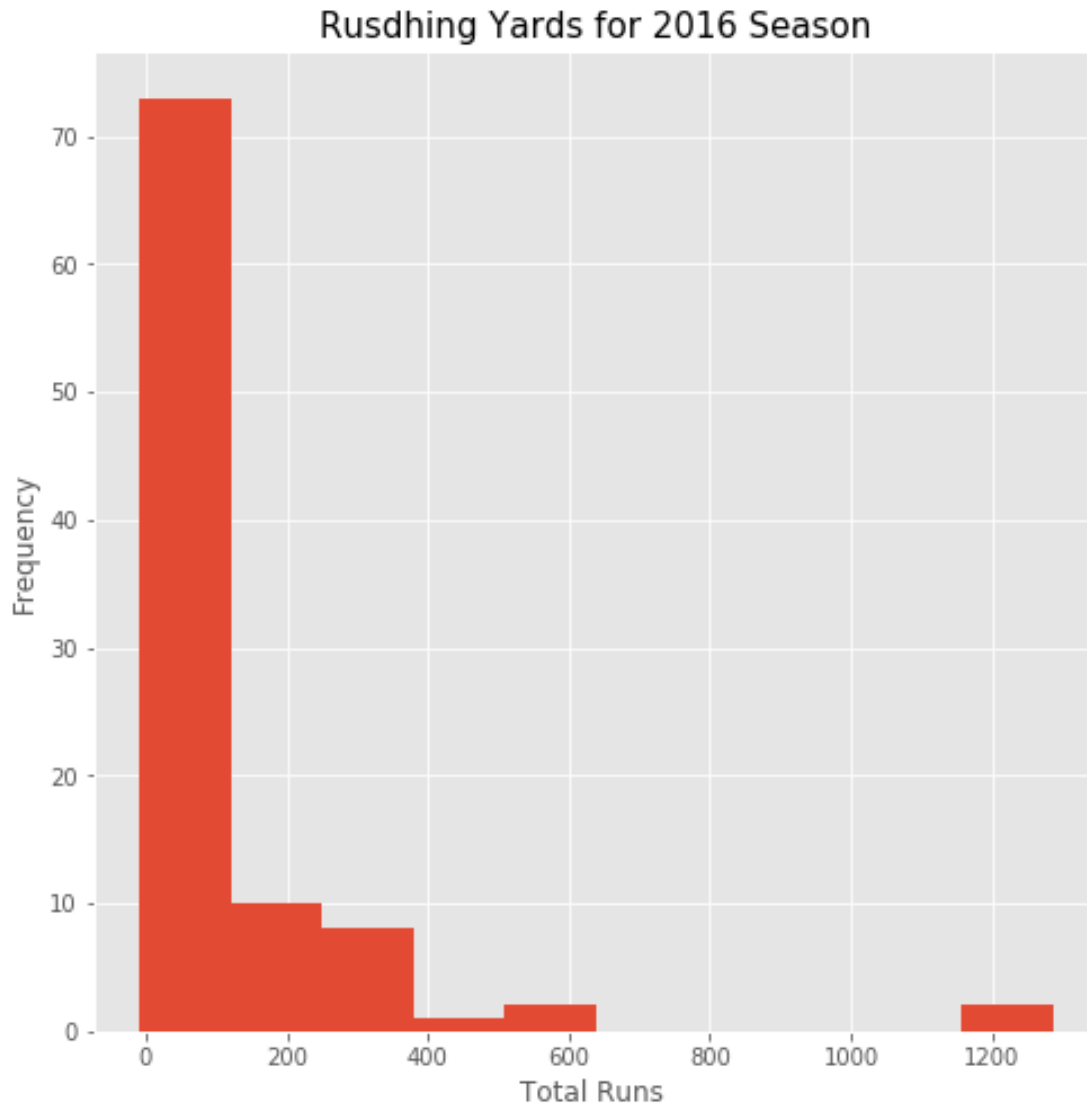


Untitled6

December 11, 2018

```
In [54]: import nflgame
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime
import plotly
from plotly.offline import plot
from plotly.graph_objs import Scatter, Marker, Line
import seaborn as sns
from matplotlib import pyplot as plt

In [11]: all_games = nflgame.games(2016)
all_players = nflgame.combine_game_stats(all_games)
total_runs = pd.Series([player.rushing_yds for player in all_players.passing()])
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111)
ax.set_title('Rusdhing Yards for 2016 Season', fontsize=15, color='black')
ax.set_ylabel('Frequency')
ax.set_xlabel('Total Runs ')
plt.style.use('ggplot')
plt.hist(total_runs)
plt.show()
```



```
In [34]: # Tom Brady statistics
brady_yards = []
brady_touchdowns = []
brady_interceptions = []
# Alex Smith statistics
smith_yards = []
smith_touchdowns = []
smith_interceptions = []
# Russell Wilson statistics
wilson_yards = []
wilson_touchdowns = []
wilson_interceptions = []
for year in range(2013,2016):
```

```

for week in range(1,18):
    game = nflgame.games(year, week=week)
    players = nflgame.combine_game_stats(game)
    for player in players:
        if player.guess_position == 'QB':
            if player.name == 'T.Brady':
                brady_yards.append(player.passing_yds)
                brady_touchdowns.append(player.passing_tds)
                brady_interceptions.append(player.passing_ints)
            elif player.name == 'A.Smith':
                smith_yards.append(player.passing_yds)
                smith_touchdowns.append(player.passing_tds)
                smith_interceptions.append(player.passing_ints)
            elif player.name == 'R.Wilson':
                wilson_yards.append(player.passing_yds)
                wilson_touchdowns.append(player.passing_tds)
                wilson_interceptions.append(player.passing_ints)

bradyTouchdowns = Scatter(
    x = brady_yards,
    y = brady_touchdowns,
    name = 'Tom Brady',
    mode = 'markers',
    marker = Marker(
        symbol = 'diamond',
        color = 'blue'
    )
)

smithTouchdowns = Scatter(
    x = smith_yards,
    y = smith_touchdowns,
    name = 'Alex Smith',
    mode = 'markers',
    marker = Marker(
        symbol = 'diamond',
        color = 'red'
    )
)

wilsonTouchdowns = Scatter(
    x = wilson_yards,
    y = wilson_touchdowns,
    name = 'Russell Wilson',
    mode = 'markers',
    marker = Marker(
        symbol = 'diamond',
        color = 'green'
    )
)

combined_data = [smithTouchdowns, wilsonTouchdowns, bradyTouchdowns]

```

```

#layout
layout = dict(title = 'Alex Smith vs Russell Wilson vs Tom Brady . (2013-2016)',
              xaxis = dict(title = 'Passing Yards'),
              yaxis = dict(title = 'Passing Touchdowns')
            )

# Plot and embed in ipython notebook!
fig = dict(data=combined_data, layout=layout)
plot(fig, filename='smith-vs-wilson-brady-winning-qbs')

```

Out[34]: 'file:///C:\\Users\\Hitesh_Bahar\\smith-vs-wilson-brady-winning-qbs.html'

In [59]: *# Tom Brady stats*

```

brady_yardsPassAttempt = []
brady_yardsRushAttempt = []
brady_touchdowns = []
brady_turnovers = []
brady_games = []

```

Alex Smith stats

```

smith_yardsPassAttempt = []
smith_yardsRushAttempt = []
smith_touchdowns = []
smith_turnovers = []
smith_games = []

```

```

bradyGameCount = 1;

```

```

smithGameCount = 1;

```

```

for year in range(2013,2016):

```

```

    for week in range(1,18):

```

```

        games = nflgame.games(year, week=week)

```

```

        players = nflgame.combine_game_stats(games)

```

```

        for player in players:

```

```

            if player.guess_position == 'QB':

```

```

                if player.name == 'T.Brady':

```

```

                    brady_yardsPassAttempt.append(round(float(player.passing_yds) / f

```

```

                    brady_yardsRushAttempt.append(round(float(player.passing_yds) / f

```

```

                    brady_touchdowns.append(player.passing_tds + player.rushing_tds)

```

```

                    brady_turnovers.append(p.passing_ints + p.fumbles_lost)

```

```

                    brady_games.append(bradyGameCount)

```

```

                    bradyGameCount += 1

```

```

                elif player.name == 'A.Smith':

```

```

                    smith_yardsPassAttempt.append(round(float(player.passing_yds) / f

```

```

                    brady_yardsRushAttempt.append(round(float(player.passing_yds) / f

```

```

                    brady_touchdowns.append(player.passing_tds + player.rushing_tds)

```

```

                    smith_turnovers.append(player.passing_ints + player.fumbles_lost)

```

```

                    brady_games.append(smithGameCount)

```

```

                    smithGameCount += 1

```

Yards per completion plot

```

brady_Ypc = Scatter(
    x = brady_games,
    y = brady_yardsRushAttempt,
    name = 'Brady YardsRushAttempt',
    mode = 'markers',
    marker = Marker(
        color = 'blue'
    )
)

smith_Ypc = Scatter(
    x = smith_games,
    y = smith_yardsRushAttempt,
    name = 'Smith YardsRushAttempt',
    mode = 'markers',
    marker = Marker(
        color = 'red'
    )
)

# Tds plot
brady_Tds = Scatter(
    x = brady_games,
    y = brady_touchdowns,
    name = 'Brady TouchDowns',
    mode = 'markers',
    marker = Marker(
        symbol = 'diamond',
        color = 'blue'
    )
)

smith_Tds = Scatter(
    x = smith_games,
    y = smith_touchdowns,
    name = 'Smith TouchDowns',
    mode = 'markers',
    marker = Marker(
        symbol = 'diamond',
        color = 'red'
    )
)

# Turnovers plot
brady_Turnovers = Scatter(
    x = brady_games,
    y = brady_turnovers,
    name = 'Brady Turnovers',
    mode = 'markers',
    marker = Marker(
        symbol = 'diamond',

```

```

        color = 'blue'
    )
)
smith_Turnovers = Scatter(
    x = smith_games,
    y = smith_turnovers,
    name = 'Smith Turnovers',
    mode = 'markers',
    marker = Marker(
        symbol = 'diamond',
        color = 'red'
    )
)
data_Ypc = [brady_Ypc, smith_Ypc]
data_Ypc_Turnovers = [brady_Ypc, smith_Ypc, brady_Turnovers, smith_Turnovers]
data_Ypc_TDs = [brady_Ypc, smith_Ypc, brady_Tds, smith_Tds]
# Edit the layout
layout_Ypc = dict(title = 'Alex Smith vs. Tom Brady Yards Per Rushing (2013-2015)',
                  xaxis = dict(title = 'Games'),
                  yaxis = dict(title = 'Yards per Completion')
                 )
fig = dict(data=data_Ypc, layout=layout_Ypc)
plot(fig, filename='Brady Vs Smith Yards Per Rushing')

```

Out[59]: 'file:///C:/Users/Hitesh_Bahar/Brady Vs Smith Yards Per Rushing.html'

```

In [53]: score_list = np.zeros((10,10))
game_count = 0
for year in range(2009, 2016):
    for week in range(1, 18):
        games = nflgame.games(year, week=week)
        for game in games:
            score_list[game.score_home%10][game.score_away%10] +=1
            game_count +=1
for i in range(10):
    for j in range(10):
        score_list[i][j] = score_list[i][j]/game_count

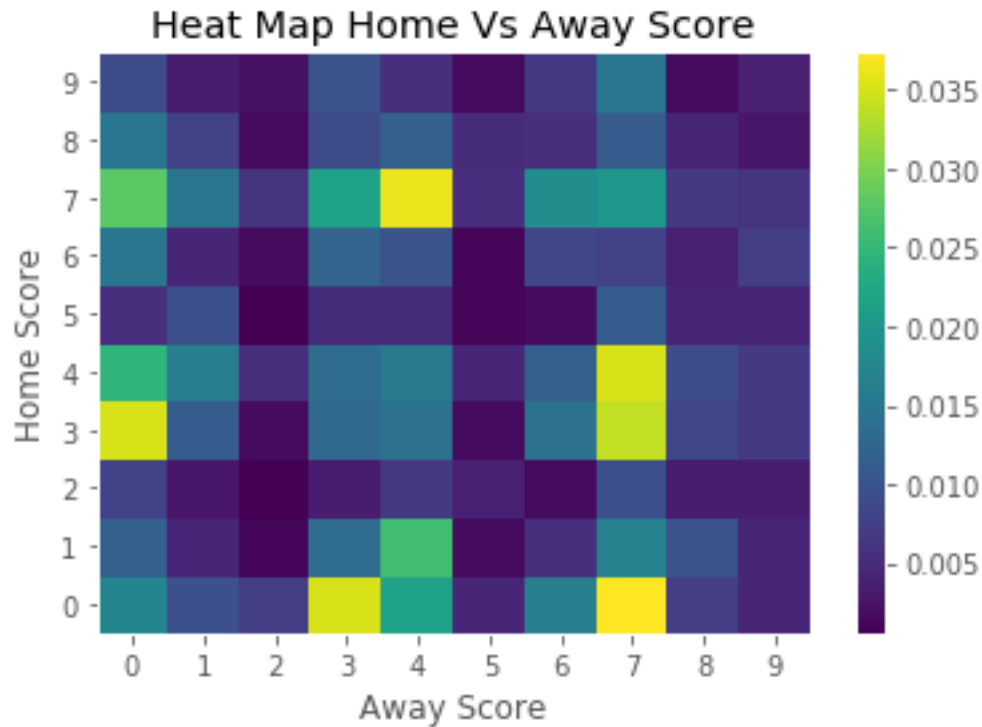
column_labels = list('0123456789')
row_labels = list('0123456789')
fig, ax = plt.subplots()
heatmap = ax.pcolor(score_list)
plt.colorbar(heatmap, ax=ax)
ax.set_title("Heat Map Home Vs Away Score")
ax.set_ylabel("Home Score")
ax.set_xlabel("Away Score")
# put the major ticks at the middle of each cell, notice "reverse" use of dimension
ax.set_yticks(np.arange(score_list.shape[0])+0.5, minor=False)

```

```

ax.set_xticks(np.arange(score_list.shape[1])+0.5, minor=False)
ax.set_xticklabels(row_labels, minor=False)
ax.set_yticklabels(column_labels, minor=False)
plt.show()

```



```

In [55]: col = ['Name', 'Rushing Yards']
rushing_stat = pd.DataFrame(columns= col )
pname = []
prushing_yds = []
games = nflgame.games(2012, week=[15, 16, 17])
players = nflgame.combine(games)
for p in players.rushing().sort("rushing_yds").limit(10):
    pname.append(str(p.name))
    prushing_yds.append(p.rushing_yds)
plt.figure(figsize=(15,8))
ax = sns.barplot(x=pname,y=prushing_yds)
ax.set(xlabel='Player Name', ylabel='Rushing Yards')
plt.title("Top 10 Rushing Yards Player Names")
plt.show()

```

