# Player Boxouts

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import matplotlib.ticker as mtick
import sqlite3
import seaborn as sns
from matplotlib.offsetbox import OffsetImage, AnnotationBbox
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from bs4 import BeautifulSoup
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time
import requests
import shutil
import datetime
from scipy.stats import norm
import os
import winsound

home_folder = 'C:\\Users\\Travis\\OneDrive\\Data Science\\Personal_Projects\\Sports
os.chdir(home_folder)
```

In [ ]:

```
def replace_name_values(filename):
        # replace values with dashes for compatibility
    filename = filename.replace('%','_')
    filename = filename.replace('=','_')
    filename = filename.replace('?','_')
    filename = filename.replace('&','_')
    filename = filename.replace('20Season_','')
    filename = filename.replace('20Season','')
    return filename
```

In [ ]:

```
def grab_player_data(url_list, file_folder):

        # Scrape Season-Level player data from the url_list

        i = 0
        for u in url_list:

                driver.get(u)
                time.sleep(2)

                # if the page does not load, go to the next in the list
                try:
                        xpath = '//*[@id="__next"]/div[2]/div[2]/div[3]/section[2]/
                        elem = WebDriverWait(driver, 30).until(EC.presence_of_eleme
                except:
                        print(f'{u} did not load. Moving to next url.')
                        continue

                # click "all pages"
                xpath_all = '//*[@id="__next"]/div[2]/div[2]/div[3]/section[2]/div/
                elem = WebDriverWait(driver, 30).until(EC.presence_of_element_locat

                driver.find_element(by=By.XPATH, value=xpath_all).click()
                src = driver.page_source
                parser = BeautifulSoup(src, "lxml")
                table = parser.find("table", attrs = {"class":"Crom_table__p1iZz"})
                headers = table.findAll('th')
                headerlist = [h.text.strip() for h in headers[0:]]
                row_names = table.findAll('a')                             # find r
                row_list = [b.text.strip() for b in row_names[0:]]
                rows = table.findAll('tr')[0:]
                player_stats = [[td.getText().strip() for td in rows[i].findAll('td
                tot_cols = len(player_stats[1])                           #set the
                headerlist = headerlist[:tot_cols]
                stats = pd.DataFrame(player_stats, columns = headerlist)

                # assign filename
                filename = file_folder + str(u[34:]).replace('/', '_') + '.csv'
                filename = replace_name_values(filename)
                pd.DataFrame.to_csv(stats, filename)
                i += 1
                lu = len(url_list)
                # close driver
                print(f'{filename} Completed Successfully! {i} / {lu} Complete!')

        winsound.Beep(523, 500)
```

In [ ]:

```
def append_the_data(folder, data_prefix, filename_selector):
    # Appending data together via folder and/or file name

    path = folder
    p = os.listdir(path)
    pf = pd.DataFrame(p)


    # filter for files that contain the filename_selector
    pf_reg = pf.loc[pf[0].astype(str).str.contains(filename_selector)]

    appended_data = []
    for file in pf_reg[0]:
        data = pd.read_csv(folder + '/' + file)
        # if "Season" a column, drop it
        if 'Season' in data.columns:
            data = data.drop(columns = ['Season'])

        data['season'] = file[(file.find('20')):(file.find('20'))+4]
        data['season_type'] = np.where('Regular' in file, 'Regular', 'Playoffs')
        # add prefix to columns
        data = data.add_prefix(data_prefix)
        data.columns = data.columns.str.lower()
        appended_data.append(data)

    appended_data = pd.concat(appended_data)
    return appended_data
```

In [ ]:

```
player_boxouts = 'https://www.nba.com/stats/players/box-outs/'
boxouts_urls = []
years =['2021-22', '2020-21', '2019-20', '2018-19', '2017-18']
season_types = ['Regular%20Season', 'Playoffs']

for year in years:
    for s_types in season_types:
        url = player_boxouts + '?Season=' + year + '&SeasonType=' + s_types
        boxouts_urls.append(str(url))
```

In [ ]:

```
# move the files to the correct folder
for file in os.listdir('data/player/boxouts/'):
    if '.csv' in file:
        if 'Playoffs' in file:
            os.rename('data/player/boxouts/' + file, 'data/player/boxouts/playoffs/
        else:
            os.rename('data/player/boxouts/' + file, 'data/player/boxouts/regular_s
```

In [ ]:

```
boxouts = append_the_data('data/player/boxouts/regular_season', 'boxouts_', 'box-ou
boxouts
```

In [ ]:

```
boxouts.to_csv('data/player/aggregates/All_Boxouts.csv')
```