Position Estimates by Year

After manually reviewing some of the positions that were listed as "Primary Position", I realized they were mistaken in some (if not many) places. Thus, I need to scrape positions by year and percentage played (through play-by-play analysis).

This is quite easy through basketball-reference.com.

Note: This data also includes :

- +- per 100 possessions,
- BRef's Positions (total, not est), and
- A row for each team a player played on during said season (i.e., a way to tell if a player is traded, etc)
    - Further, if a player plays on two teams in a year, they will also have a "TOT" column with their aggregate statistics

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 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
import warnings
warnings.filterwarnings('ignore')
```

In [ ]:

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

In [ ]:

```
years = np.arange(2000,2024,1)
```

In [ ]:

```
position_files = os.listdir('data/player/play_by_play/')

to_download = []
for year in years:
    for file in position_files:
        if str(year) in file:
            to_download.append(file)

to_download
```

Out[ ]:

```
['2000position_estimates.csv',
 '2001position_estimates.csv',
 '2002position_estimates.csv',
 '2003position_estimates.csv',
 '2004position_estimates.csv',
 '2005position_estimates.csv',
 '2006position_estimates.csv',
 '2007position_estimates.csv',
 '2008position_estimates.csv',
 '2009position_estimates.csv',
 '2010position_estimates.csv',
 '2011position_estimates.csv',
 '2012position_estimates.csv',
 '2013position_estimates.csv',
 '2014position_estimates.csv',
 '2015position_estimates.csv',
 '2016position_estimates.csv',
 '2017position_estimates.csv',
 '2018position_estimates.csv',
 '2019position_estimates.csv',
 '2020position_estimates.csv',
 '2021position_estimates.csv',
 '2022position_estimates.csv']
```

In [ ]:

```
# check to_download files against position_files to see if any are in one but not t
left_to_download = []
for file in to_download:
    if file not in position_files:
        left_to_download.append(file)

left_to_download
```

Out[ ]:

```
[]
```

In [ ]:

```
if left_to_download == []:
    print('All files downloaded')
else:
    print('Files to download:',left_to_download)
    for year in years:
        df = pd.read_html('https://www.basketball-reference.com/leagues/NBA_'+str(y
        df = df[0]
        yar = year-1
        df['season'] = yar
        df.to_csv('data/player/play_by_play/'+str(yar)+'position_estimates.csv')
```

```
All files downloaded
```

In [ ]:

```
appended_data = []

files = os.listdir('data/player/play_by_play/')
for file in files:
    df = pd.read_csv('data/player/play_by_play/'+file)[:]
    appended_data.append(df)

df = pd.concat(appended_data)
df.to_csv('data/player/aggregates/all_position_estimates.csv')
```

In [ ]:

```
df = df.rename(columns={'Unnamed: 0':'na', 'Unnamed: 0_level_0' : 'rank', 'Unnamed:
                        'Unnamed: 2_level_0': 'position', 'Unnamed: 3_level_0': 'ag
                        'Totals': 'G', 'Totals.1': 'MP', 'Position Estimate': 'PG_e
                        'Position Estimate.2': 'SF_est_%', 'Position Estimate.3': '
                        })

df = df.rename(columns={'+/- Per 100 Poss.':'per100poss_+/-_ON_court', '+/- Per 100

df = df.rename(columns={'Turnovers':'BadPass', 'Turnovers.1':'LostBall'})

df.columns
```

Out[ ]:

```
Index(['na', 'rank', 'player', 'position', 'age', 'team', 'G', 'MP',
       'PG_est_%', 'SG_est_%', 'SF_est_%', 'PF_est_%', 'C_est_%',
       'per100poss_+/-_OFF_court', '+/- Per 100 Poss..1', 'BadPass',
       'LostBall', 'Fouls Committed', 'Fouls Committed.1', 'Fouls Drawn',
       'Fouls Drawn.1', 'Misc.', 'Misc..1', 'Misc..2', 'season'],
      dtype='object')
```

In [ ]:

```
# drop all unnamed cols
unnamed = df.columns[df.columns.str.contains('Unnamed')]
df = df.drop(columns=unnamed)

# drop na and rank if they are in the df
if 'na' in df.columns:
    to_drop = ['na']
    df = df.drop(columns=to_drop)
if 'rank' in df.columns:
    to_drop = ['rank']
    df = df.drop(columns=to_drop)

# drop na in season
df = df.dropna(subset = 'season')

# season to int
df['season'] = df['season'].astype(int)
```

In [ ]:

```
# fix the % values
df['PG_est_%'] = df['PG_est_%'].str.replace('%', '')
df['SG_est_%'] = df['SG_est_%'].str.replace('%', '')
df['SF_est_%'] = df['SF_est_%'].str.replace('%', '')
df['PF_est_%'] = df['PF_est_%'].str.replace('%', '')
df['C_est_%'] = df['C_est_%'].str.replace('%', '')
df.head()
```

Out[ ]:

	player	position	age	team	G	MP	PG_est_%	SG_est_%	SF_est_%	PF_est_%	...	BadPas
1	Tariq Abdul-Wahad	SG	25	TOT	61	1578	1	96	3	NaN	...	4
2	Tariq Abdul-Wahad	SG	25	ORL	46	1205	NaN	97	3	NaN	...	3
3	Tariq Abdul-Wahad	SG	25	DEN	15	373	4	93	3	NaN	...	
4	Shareef Abdur-Rahim	SF	23	VAN	82	3223	NaN	NaN	63	35	...	8
5	Cory Alexander	PG	26	DEN	29	329	97	3	NaN	NaN	...	1

5 rows × 23 columns

In [ ]:

```
df['PG_est_%'] = df['PG_est_%'].fillna(0)
df['SG_est_%'] = df['SG_est_%'].fillna(0)
df['SF_est_%'] = df['SF_est_%'].fillna(0)
df['PF_est_%'] = df['PF_est_%'].fillna(0)
df['C_est_%'] = df['C_est_%'].fillna(0)
df.head(2)
```

Out[ ]:

	player	position	age	team	G	MP	PG_est_%	SG_est_%	SF_est_%	PF_est_%	...	BadPass
1	Tariq Abdul-Wahad	SG	25	TOT	61	1578	1	96	3	0	...	44
2	Tariq Abdul-Wahad	SG	25	ORL	46	1205	0	97	3	0	...	36

2 rows × 23 columns

In [ ]:

```
df['PG_est_%'] = df['PG_est_%'].fillna(0)
df['SG_est_%'] = df['SG_est_%'].fillna(0)
df['SF_est_%'] = df['SF_est_%'].fillna(0)
df['PF_est_%'] = df['PF_est_%'].fillna(0)
df['C_est_%'] = df['C_est_%'].fillna(0)
df.head(2)
```

Out[ ]:

	player	position	age	team	G	MP	PG_est_%	SG_est_%	SF_est_%	PF_est_%	...	BadPass
1	Tariq Abdul-Wahad	SG	25	TOT	61	1578	1	96	3	0	...	44
2	Tariq Abdul-Wahad	SG	25	ORL	46	1205	0	97	3	0	...	36

2 rows × 23 columns

In [ ]:

```
df.to_csv('data/player/aggregates_of_aggregates/all_position_estimates.csv')
```