```
In [120]: %matplotlib inline
    from selenium import webdriver
    from pandas import *
    import pandas
    import numpy as np
    import matplotlib.pyplot as plt
    from sqlalchemy import *
    import os
    import pymysql
    import time
In [2]: #Data feature I used on stats.nba.com
```

```
In [2]: #Data feature I used on stats.nba.com
#player bio name and team
#team clutch stats
#player clutch stats
#player box score
#shooting stats
```

```
In [69]: #new chrome browser
browser = webdriver.Chrome()
```

```
In [121]: #player bio name and team
          url = 'https://stats.nba.com/players/bio'
          browser.get(url)
          #find right table
          time.sleep(5)
          browser.find element by xpath('/html/body/main/div[2]/div/div[2]/div/div/di
          time.sleep(5)
          browser.find element by xpath('/html/body/main/div[2]/div/div[2]/div/div/di
          time.sleep(5)
          browser.find element by xpath('/html/body/main/div[2]/div/div[2]/div/div/nb
          #get table info
          time.sleep(5)
          table = browser.find_element_by_class_name('nba-stat-table_overflow')
          #glimpse at data
          count = 0
          for line id, lines in enumerate(table.text.split('\n')):
              print (line_id, lines)
              count += 1
              if count >4:
                  break
           1.1.1
          #parse table
          player names = []
          player team = []
          column names = ['PlayerName', 'Team']
          for line id, lines in enumerate(table.text.split('\n')):
              if line id != 0:
                   if line id % 2 == 1:
                      player names.append(lines)
                   if line id % 2 == 0:
                      temp = lines.split(' ')
                      player_team.append(temp[0])
          #create dataframe
          player bio table = pandas.DataFrame({ column names[0]: player names,
                                  column names[1]: player team
                                  }
                                )
          #write into mysql server
          conn = pymysql.connect(
              port=int(3306),
              user="root",
              passwd= "zzh970507",
              database = "NBA DB"
          )
          my cursor = conn.cursor()
          my cursor.execute("CREATE TABLE Players (playName VARCHAR(255), teamName VA
          sqlFormula = "INSERT INTO Players (playName, teamName) VALUES (%s, %s)"
          for index, row in player bio table.iterrows():
              #sequencial compare in general
              cur row = row.tolist()
```

```
cur_stat = (cur_row[0], cur_row[1])
  my_cursor.execute(sqlFormula, cur_stat)

#commit change
conn.commit()
```

```
In [123]: player_bio_table.head()
```

Out[123]:

	PlayerName	Team
0	Aaron Gordon	ORL
1	Aaron Holiday	IND
2	Abdel Nader	OKC
3	Al Horford	BOS
4	Al-Faroug Aminu	POR

```
In [74]: #player box score
         url = 'https://stats.nba.com/players/boxscores-traditional'
         browser.get(url)
         time.sleep(5)
         #find right table
         browser.find element by xpath('/html/body/main/div[2]/div/div[2]/div/div/di
         browser.find element by xpath('/html/body/main/div[2]/div/div[2]/div/div/di
         time.sleep(5)
         #get table info
         table = browser.find element by class name('nba-stat-table overflow')
         #glimpse at data
         count = 0
         for line id, lines in enumerate(table.text.split('\n')):
             print (line id, lines)
             count += 1
             if count >4:
                 print (line_id, lines.split(' '))
                 time = lines.split(' ')[4].split('/')
                 print(time)
                 new\_time = time[-1] + "." + time[-3] + time[-2]
                 new time = float(new time)
                 print(new_time)
                 break
          1.1.1
         #parse table
         Game stat table = pandas.DataFrame()
         for i in range(2,524):
             action = '/html/body/main/div[2]/div/div[2]/div/div/nba-stat-table/div[
             browser.find element by xpath(action).click()
             player_names = []
             opponent team = []
             Points = []
             Dates = []
             Rebounds = []
             Assists = []
             Steals = []
             column_names = ['Player Name', 'Date', 'Points', 'Rebounds', 'Assists',
             for line id, lines in enumerate(table.text.split('\n')):
                  if line id != 0:
                      if line id % 2 == 1:
                          player names.append(lines)
                      if line id % 2 == 0:
                          temp = lines.split(' ')
                          time = temp[4].split('/')
                          new time = time[-1] + time[-3] + time[-2]
                          new time = int(new time)
                          Dates.append(new time)
                          Points.append(temp[7])
                          opponent team.append(temp[3])
```

```
Rebounds.append(temp[-7])
                Assists.append(temp[-6])
                Steals.append(temp[-5])
    temp stat table = pandas.DataFrame({ column_names[0]: player_names,
                                       column_names[1]: Dates,
                                         column_names[2]: Points,
                                       column_names[3]: Rebounds,
                                         column_names[4]: Assists,
                                       column names[5]: Steals,
                                         column_names[6]: opponent_team,
                           }
    Game_stat_table = pandas.concat([Game_stat_table, temp_stat_table], ign
conn = pymysql.connect(
   port=int(3306),
    user="root",
    passwd= "zzh970507",
    database = "NBA DB"
my_cursor = conn.cursor()
my_cursor.execute("CREATE TABLE Game_Stats (playName VARCHAR(255), Date int
sqlFormula = "INSERT INTO Game Stats (playName, Date, Points, Rebound, Assi
my_cursor = conn.cursor()
for index, row in Game_stat_table.iterrows():
    #sequencial compare in general
    cur row = row.tolist()
    cur stat = (cur row[0], int(cur row[1]),cur row[2],int(cur row[3]), int
    my cursor.execute(sqlFormula, cur stat)
conn.commit()
```

In [122]: Game_stat_table.head()

Out[122]:

	Player Name	Date	Points	Rebounds	Assists	Steals	Opponent Team
0	LaMarcus Aldridge	20190410	34	16	1	1	DAL
1	Harrison Barnes	20190410	10	3	3	0	POR
2	Billy Garrett	20190410	6	0	1	0	DET
3	Gorgui Dieng	20190410	18	11	2	3	DEN
4	Jordan Bell	20190410	15	8	1	0	MEM

```
In [110]: ##player clutch stats
          url = 'https://stats.nba.com/players/clutch-traditional'
          browser.get(url)
          #find right table
          time.sleep(5)
          browser.find element by xpath('/html/body/main/div[2]/div/div[2]/div/div/di
          time.sleep(5)
          browser.find element by xpath('/html/body/main/div[2]/div/div[2]/div/div/di
          time.sleep(5)
          browser.find_element_by_xpath('/html/body/main/div[2]/div/div[2]/div/div/di
          browser.find element by xpath('/html/body/main/div[2]/div/div[2]/div/div/nb
          #get table info
          table = browser.find element by class name('nba-stat-table overflow')
          #glimpse at data
          count = 0
          for line id, lines in enumerate(table.text.split('\n')):
              print (line_id, lines)
              count += 1
              if count >4:
                  break
          count = 0
          #parse table
          player_names = []
          ThreePointer P = []
          FG_P = []
          FT P = []
          minutes played = []
          column names = ['PlayerName', '3pointer%', 'Field Goal%', 'Free Throw%', 'Min
          for line id, lines in enumerate(table.text.split('\n')):
              if line id != 0:
                   if line id % 3 == 2:
                       player names.append(lines)
                   if line id % 3 == 0:
                      temp = lines.split(' ')
                       ThreePointer P.append(temp[12])
                      FG P.append(temp[9])
                       FT P.append(temp[15])
                       minutes played.append(temp[5])
          #create dataframe
          Player clutch table = pandas.DataFrame({ column names[0]: player names,
                                                 column names[1]: ThreePointer P,
                                                 column names[2]: FG P,
                                                 column names[3]: FT P,
                                                 column names[4]: minutes played
                                  }
          conn = pymysql.connect(
              port=int(3306),
              user="root",
              passwd= "zzh970507",
              database = "NBA DB"
```

```
my_cursor = conn.cursor()
my_cursor.execute("CREATE TABLE Player_Clutch_Stats (playName VARCHAR(255),
sqlFormula = "INSERT INTO Player_Clutch_Stats (playName, 3pointer_P, Field_
for index, row in Player_clutch_table.iterrows():
    #sequencial compare in general
    cur_row = row.tolist()
    cur_stat = (cur_row[0], cur_row[1], cur_row[2], cur_row[3], cur_row[4])
    my_cursor.execute(sqlFormula, cur_stat)

#commit change
conn.commit()
```

In [115]: Player_clutch_table.head()

Out[115]:

	PlayerName	3pointer%	Field_Goal%	Free_Throw%	Minutes_Played
0	De'Aaron Fox	37.5	44.9	80.0	156
1	D.J. Augustin	31.3	41.7	83.3	153
2	Evan Fournier	22.6	43.9	73.9	152
3	Nikola Vucevic	6.3	37.8	69.2	159
4	Tobias Harris	37.5	52.2	76.7	174

```
In [131]:
          #Team clutch stats
          url = 'https://stats.nba.com/teams/clutch-traditional'
          browser.get(url)
          #find right table
          time.sleep(5)
          browser.find element by xpath('/html/body/main/div[2]/div/div[2]/div/div/di
          time.sleep(5)
          browser.find element by xpath('/html/body/main/div[2]/div/div[2]/div/div/di
          #get table info
          table = browser.find_element_by_class_name('nba-stat-table_overflow')
          1 1 1
          #glimpse at data
          count = 0
          for line id, lines in enumerate(table.text.split('\n')):
              print (line_id, lines)
              count += 1
              if count >4:
                  break
           1.1.1
          nbaTeams = \{\}
          nbaTeams['Atlanta Hawks'] = 'ATL'
          nbaTeams['Brooklyn Nets'] = 'BKN'
          nbaTeams['Boston Celtics'] = 'BOS'
          nbaTeams['Charlotte Hornets'] = 'CHA'
          nbaTeams['Chicago Bulls'] = 'CHI'
          nbaTeams['Cleveland Cavaliers'] = 'CLE'
          nbaTeams['Dallas Mavericks'] = 'DAL'
          nbaTeams['Denver Nuggets'] = 'DEN'
          nbaTeams['Detroit Pistons'] = 'DET'
          nbaTeams['Golden State Warriors'] = 'GSW'
          nbaTeams['Houston Rockets'] = 'HOU'
          nbaTeams['Indiana Pacers'] = 'IND'
          nbaTeams['LA Clippers'] = 'LAC'
          nbaTeams['Los Angeles Lakers'] = 'LAL'
          nbaTeams['Memphis Grizzlies'] = 'MEM'
          nbaTeams['Miami Heat'] = 'MIA'
          nbaTeams['Milwaukee Bucks'] = 'MIL'
          nbaTeams['Minnesota Timberwolves'] = 'MIN'
          nbaTeams['New Orleans Pelicans'] = 'NOP'
          nbaTeams['New York Knicks'] = 'NYK'
          nbaTeams['Oklahoma City Thunder'] = 'OKC'
          nbaTeams['Orlando Magic'] = 'ORL'
          nbaTeams['Philadelphia 76ers'] = 'PHI'
          nbaTeams['Phoenix Suns'] = 'PHX'
          nbaTeams['Portland Trail Blazers'] = 'POR'
          nbaTeams['Sacramento Kings'] = 'SAC'
          nbaTeams['San Antonio Spurs'] = 'SAS'
          nbaTeams['Toronto Raptors'] = 'TOR'
          nbaTeams['Utah Jazz'] = 'UTA'
          nbaTeams['Washington Wizards'] = 'WAS'
          #parse table
          Team names = []
```

```
FG_P = []
column_names = ['TeamName', 'Field_Goal%']
for line_id, lines in enumerate(table.text.split('\n')):
    if line id != 0:
        if line id % 3 == 2:
            Team_names.append(nbaTeams[lines])
        if line id % 3 == 0:
            temp = lines.split(' ')
            FG_P.append(temp[8])
#create dataframe
Team_clutch_table = pandas.DataFrame({ column_names[0]: Team_names,
                                      column names[1]: FG P
                       }
#upload to sql server
conn = pymysql.connect(
    port=int(3306),
    user="root",
    passwd= "zzh970507",
    database = "NBA_DB"
)
my_cursor = conn.cursor()
my cursor.execute("CREATE TABLE Team Clutch Stats (TeamName VARCHAR(255), F
sqlFormula = "INSERT INTO Team Clutch Stats (TeamName, Field Goal P) VALUES
my cursor = conn.cursor()
for index, row in Team clutch table.iterrows():
    #sequencial compare in general
    cur_row = row.tolist()
    cur stat = (cur row[0], cur row[1])
    my_cursor.execute(sqlFormula, cur_stat)
#commit change
conn.commit()
```

In [132]: Team_clutch_table.head()

Out[132]:

	TeamName	Field_Goal%
0	DEN	45.4
1	PHI	44.8
2	LAC	48.2
3	MIL	46.8
4	SAS	47.3