

```
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
        #player bio name and team
        #team clutch stats
        #player clutch stats
        #player box score
        #shooting stats
```

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



```
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-Farouq Aminu	POR



```

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], ignore_index=True)

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(11), Points int(11), Rebounds int(11), Assists int(11), Steals int(11), OpponentTeam VARCHAR(255))")
sqlFormula = "INSERT INTO Game_Stats (playName, Date, Points, Rebound, Assists, Steals, OpponentTeam) VALUES (%s, %s, %s, %s, %s, %s, %s)"
my_cursor = conn.cursor()
for index, row in Game_stat_table.iterrows():
    #sequential compare in general
    cur_row = row.tolist()
    cur_stat = (cur_row[0], int(cur_row[1]),cur_row[2],int(cur_row[3]), int(cur_row[4]),int(cur_row[5]),cur_row[6])
    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():
    #sequential 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





```

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():
    #sequential 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