

Florida Panthers Work

September 14, 2022

```
[2]: import requests
import pandas as pd
from pandasql import sqldf
from datetime import datetime
from datetime import date
import math

pysqldf = lambda q: sqldf(q, globals())
```

```
[2]: def create_panthers_players_table():
    panthers_players_table = pd.DataFrame()
    temp_table = pd.DataFrame()
    count = 0

    season_list = list()
    season_list.append('20142015')
    season_list.append('20152016')
    season_list.append('20162017')
    season_list.append('20172018')
    season_list.append('20182019')
    season_list.append('20192020')
    season_list.append('20202021')
    season_list.append('20212022')
    season_list

    for n in season_list:
        url = 'https://statsapi.web.nhl.com/api/v1/teams/13?expand=team.
↳roster&season='
        url = url + season_list[count]
        r = requests.get(url)
        json = r.json()
        player_id = list()

        for x in json['teams'][0]['roster']['roster']:
            player_id.append(x['person']['id'])
```

```

        temp_table['Player_ID'] = player_id
        for i in player_id:
            temp_table['Season'] = season_list[count]

        panthers_players_table = pd.concat([panthers_players_table,
        ↪temp_table], ignore_index = True)
        panthers_players_table.reset_index()
        count = count + 1
        temp_table = pd.DataFrame()

    player_list = panthers_players_table.Player_ID.tolist()
    player_data_table = pd.DataFrame()
    count = 0
    for n in panthers_players_table.Player_ID:
        url = 'https://statsapi.web.nhl.com/api/v1/people/'
        url = url + str(player_list[count])
        r = requests.get(url)
        json = r.json()
        temp_table = pd.json_normalize(json, record_path = ['people'])
        col_list = ['birthDate', 'height', 'weight']
        temp_table = temp_table[col_list]
        player_data_table = pd.concat([player_data_table, temp_table],
        ↪ignore_index = True)
        temp_table = pd.DataFrame()
        count = count + 1

    panthers_players_table['birthDate'] = player_data_table['birthDate']
    panthers_players_table['height'] = player_data_table['height']
    panthers_players_table['weight'] = player_data_table['weight']

    player_keylist = list()
    url1 = 'https://statsapi.web.nhl.com/api/v1/people/8448208/stats?
    ↪stats=statsSingleSeason&season=20142015'
    r1 = requests.get(url1)
    json1 = r1.json()
    for key in json1['stats'][0]['splits'][0]['stat'].keys():
        player_keylist.append(key)

    count = 0
    for n in player_keylist:
        player_keylist[count] = 'stat.' + player_keylist[count]
        count = count + 1

    goalie_keylist = list()
    url2 = 'https://statsapi.web.nhl.com/api/v1/people/8481519/stats?
    ↪stats=statsSingleSeason&season=20212022'

```

```

r2 = requests.get(url2)
json2 = r2.json()
for key in json2['stats'][0]['splits'][0]['stat'].keys():
    goalie_keylist.append(key)

count = 0
for n in goalie_keylist:
    goalie_keylist[count] = 'stat.' + goalie_keylist[count]
    count = count + 1

keylist = player_keylist + goalie_keylist

new_keylist = list()
count = 0
for n in keylist:
    new_keylist.append(keylist[count][5:])
    count = count + 1

player_data_table = pd.DataFrame()
for index, entry in panthers_players_table.iterrows():
    url = 'https://statsapi.web.nhl.com/api/v1/people/'
    url = url + str(entry['Player_ID']) + '/stats?
↳ stats=statsSingleSeason&season=' + str(entry['Season'])
    r = requests.get(url)
    json = r.json()
    if len(json['stats'][0]['splits']) == 0:
        for n in keylist:
            temp_table[n] = None
    else:
        temp_table = pd.json_normalize(json, record_path=
↳ ['stats', 'splits'])
        player_data_table = pd.concat([player_data_table, temp_table],
↳ ignore_index = True)

    for n in keylist:
        panthers_players_table[n] = player_data_table[n]

datelist = list()
cutofflist = list()
agelist = list()
for index, entry in panthers_players_table.iterrows():
    birthD = datetime.strptime(str(entry['birthDate']), '%Y-%m-%d')
    datelist.append(birthD.date())
    cutoffD = datetime.strptime(str(entry['Season'])[4:] +
↳ '-01-31', '%Y-%m-%d')
    cutofflist.append(cutoffD.date())

```

```

        ageYears = cutoffD.date().year - birthD.date().year - ((cutoffD.date().
↪month, cutoffD.date().day) < (birthD.date().month, birthD.date().day))
        if (cutoffD.date().month, cutoffD.date().day) > (birthD.date().month,
↪birthD.date().day):
            nextBirthD = datetime.strptime(str(entry['Season'])[4:] +
↪str(birthD.date().month) + str(birthD.date().day), '%Y%m%d')
            diff = abs(nextBirthD - cutoffD)
            ageDays = diff.days
        else:
            lastBirthD = datetime.strptime(str(entry['Season'])[:4] +
↪str(birthD.date().month) + str(birthD.date().day), '%Y%m%d')
            diff = abs(cutoffD - lastBirthD)
            ageDays = diff.days
        agelist.append(str(ageYears) + ' years, ' + str(ageDays) + ' days')

panthers_players_table['Age_in_Season'] = agelist

list_2014 = 0
list_2015 = 0
list_2016 = 0
list_2017 = 0
list_2018 = 0
list_2019 = 0
list_2020 = 0
list_2021 = 0

for n in season_list:
    url = 'https://statsapi.web.nhl.com/api/v1/schedule?teamId=13&season='
    url = url + n
    r = requests.get(url)
    json = r.json()
    count = 0
    for n in json['dates']:
        if json['dates'][count]['totalItems'] == 1:
            if str(json['dates'][count]['games'][0]['gamePk'])[:6] ==
↪'201402':
                list_2014 = list_2014 + 1
            elif str(json['dates'][count]['games'][0]['gamePk'])[:6] ==
↪'201502':
                list_2015 = list_2015 + 1
            elif str(json['dates'][count]['games'][0]['gamePk'])[:6] ==
↪'201602':
                list_2016 = list_2016 + 1
            elif str(json['dates'][count]['games'][0]['gamePk'])[:6] ==
↪'201702':
                list_2017 = list_2017 + 1

```

```

        elif str(json['dates'][count]['games'][0]['gamePk'])[:6] ==_
↪ '201802':
            list_2018 = list_2018 + 1
        elif str(json['dates'][count]['games'][0]['gamePk'])[:6] ==_
↪ '201902':
            list_2019 = list_2019 + 1
        elif str(json['dates'][count]['games'][0]['gamePk'])[:6] ==_
↪ '202002':
            list_2020 = list_2020 + 1
        elif str(json['dates'][count]['games'][0]['gamePk'])[:6] ==_
↪ '202102':
            list_2021 = list_2021 + 1

        count = count + 1
    fullseason = list()
    for index, entry in panthers_players_table.iterrows():
        if entry['Season'] == season_list[0]:
            if entry['stat.games'] == list_2014:
                fullseason.append('Yes')
            else:
                fullseason.append('No')
        elif entry['Season'] == season_list[1]:
            if entry['stat.games'] == list_2015:
                fullseason.append('Yes')
            else:
                fullseason.append('No')
        elif entry['Season'] == season_list[2]:
            if entry['stat.games'] == list_2016:
                fullseason.append('Yes')
            else:
                fullseason.append('No')
        elif entry['Season'] == season_list[3]:
            if entry['stat.games'] == list_2017:
                fullseason.append('Yes')
            else:
                fullseason.append('No')
        elif entry['Season'] == season_list[4]:
            if entry['stat.games'] == list_2018:
                fullseason.append('Yes')
            else:
                fullseason.append('No')
        elif entry['Season'] == season_list[5]:
            if entry['stat.games'] == list_2019:
                fullseason.append('Yes')
            else:
                fullseason.append('No')
        elif entry['Season'] == season_list[6]:

```

```

        if entry['stat.games'] == list_2020:
            fullseason.append('Yes')
        else:
            fullseason.append('No')
    elif entry['Season'] == season_list[7]:
        if entry['stat.games'] == list_2021:
            fullseason.append('Yes')
        else:
            fullseason.append('No')

panthers_players_table['Full_Season'] = fullseason

s = panthers_players_table['stat.pim']
s2 = s.astype('Int32')
panthers_players_table['stat.pim'] = s2
return panthers_players_table

```

```

[3]: def create_panthers_game_boxscores_table():
    panthers_game_boxscores_table = pd.DataFrame()
    game_id = list()

    season_list = list()
    season_list.append('20142015')
    season_list.append('20152016')
    season_list.append('20162017')
    season_list.append('20172018')
    season_list.append('20182019')
    season_list.append('20192020')
    season_list.append('20202021')
    season_list.append('20212022')
    season_list

    for n in season_list:
        url = 'https://statsapi.web.nhl.com/api/v1/schedule?teamId=13&season='
        url = url + n
        r = requests.get(url)
        json = r.json()
        count = 0

        for n in json['dates']:
            if json['dates'][count]['totalItems'] == 1:
                game_id.append(json['dates'][count]['games'][0]['gamePk'])
            else:
                game_id.append(json['dates'][count]['games'][0]['gamePk'])
                game_id.append(json['dates'][count]['games'][1]['gamePk'])

        count = count + 1

```

```

panthers_game_boxscores_table['Game_ID'] = game_id

home_or_away = list()
for index, entry in panthers_game_boxscores_table.iterrows():
    url = 'https://statsapi.web.nhl.com/api/v1/game/' +
    ↪str(entry['Game_ID']) + '/boxscore'
    r = requests.get(url)
    json = r.json()
    if json['teams']['away']['team']['id'] == 13:
        home_or_away.append('Away')
    else:
        home_or_away.append('Home')

panthers_game_boxscores_table['Home_or_Away'] = home_or_away

result = list()
for index, entry in panthers_game_boxscores_table.iterrows():
    url = 'https://statsapi.web.nhl.com/api/v1/game/' +
    ↪str(entry['Game_ID']) + '/boxscore'
    r = requests.get(url)
    json = r.json()
    if entry['Home_or_Away'] == 'Home':
        panthersScore =
    ↪json['teams']['home']['teamStats']['teamSkaterStats']['goals']
        opponentScore =
    ↪json['teams']['away']['teamStats']['teamSkaterStats']['goals']
        scoreDiff = panthersScore - opponentScore
        if scoreDiff > 0:
            result.append('Win')
        else:
            url2 = 'https://statsapi.web.nhl.com/api/v1/game/' +
    ↪str(entry['Game_ID']) + '/linescore'
            r2 = requests.get(url2)
            json2 = r2.json()
            if json2['currentPeriod'] > 3:
                if json2['teams']['home']['goals'] >
    ↪json2['teams']['away']['goals']:
                    result.append('Win')
                else:
                    result.append('OT Loss')
            else:
                result.append('Loss')
        else:
            opponentScore =
    ↪json['teams']['home']['teamStats']['teamSkaterStats']['goals']

```

```

panthersScore =
↪json['teams']['away']['teamStats']['teamSkaterStats']['goals']
scoreDiff = panthersScore - opponentScore
if scoreDiff > 0:
    result.append('Win')
else:
    url2 = 'https://statsapi.web.nhl.com/api/v1/game/' +
↪str(entry['Game_ID']) + '/linescore'
    r2 = requests.get(url2)
    json2 = r2.json()
    if json2['currentPeriod'] > 3:
        if json2['teams']['home']['goals'] <
↪json2['teams']['away']['goals']:
            result.append('Win')
        else:
            result.append('OT Loss')
    else:
        result.append('Loss')

panthers_game_boxscores_table['Result'] = result

panthers_goals_list = list()
opponent_goals_list = list()
for index, entry in panthers_game_boxscores_table.iterrows():
    url2 = 'https://statsapi.web.nhl.com/api/v1/game/' +
↪str(entry['Game_ID']) + '/linescore'
    r2 = requests.get(url2)
    json2 = r2.json()
    if json2['teams']['home']['team']['id'] == 13:
        panthers_goals_list.append(json2['teams']['home']['goals'])
        opponent_goals_list.append(json2['teams']['away']['goals'])
    else:
        panthers_goals_list.append(json2['teams']['away']['goals'])
        opponent_goals_list.append(json2['teams']['home']['goals'])

panthers_game_boxscores_table['Panthers_Goals'] = panthers_goals_list
panthers_game_boxscores_table['Opponent_Goals'] = opponent_goals_list

panthers_game_boxscore = pd.DataFrame()
temp_table2 = pd.DataFrame()

```



```

col_list = ['teams.away.teamStats.teamSkaterStats.pim', 'teams.away.
↳teamStats.teamSkaterStats.shots', 'teams.away.teamStats.teamSkaterStats.
↳powerPlayPercentage', 'teams.away.teamStats.teamSkaterStats.
↳powerPlayGoals', 'teams.away.teamStats.teamSkaterStats.
↳powerPlayOpportunities', 'teams.away.teamStats.teamSkaterStats.
↳faceOffWinPercentage', 'teams.away.teamStats.teamSkaterStats.blocked', 'teams.
↳away.teamStats.teamSkaterStats.takeaways', 'teams.away.teamStats.
↳teamSkaterStats.giveaways', 'teams.away.teamStats.teamSkaterStats.
↳hits', 'teams.home.teamStats.teamSkaterStats.pim', 'teams.home.teamStats.
↳teamSkaterStats.shots', 'teams.home.teamStats.teamSkaterStats.
↳powerPlayPercentage', 'teams.home.teamStats.teamSkaterStats.
↳powerPlayGoals', 'teams.home.teamStats.teamSkaterStats.
↳powerPlayOpportunities', 'teams.home.teamStats.teamSkaterStats.
↳faceOffWinPercentage', 'teams.home.teamStats.teamSkaterStats.blocked', 'teams.
↳home.teamStats.teamSkaterStats.takeaways', 'teams.home.teamStats.
↳teamSkaterStats.giveaways', 'teams.home.teamStats.teamSkaterStats.hits']

for index, entry in panthers_game_boxscores_table.iterrows():
    url = 'https://statsapi.web.nhl.com/api/v1/game/' + ␣
↳str(entry['Game_ID']) + '/boxscore'
    r = requests.get(url)
    json = r.json()
    if entry['Home_or_Away'] == 'Home':
        temp_table = pd.json_normalize(json)
        temp_table = temp_table[col_list]
        temp_table2['Panthers_PIM'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.pim']
        temp_table2['Panthers_Shots'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.shots']
        temp_table2['Panthers_Power_Play_%'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.powerPlayPercentage']
        temp_table2['Panthers_Power_Play_Goals'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.powerPlayGoals']
        temp_table2['Panthers_Power_Play_Opportunities'] = ␣
↳temp_table['teams.home.teamStats.teamSkaterStats.powerPlayOpportunities']
        temp_table2['Panthers_FOW%'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.faceOffWinPercentage']
        temp_table2['Panthers_Blocked_Shots'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.blocked']
        temp_table2['Panthers_Takeaways'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.takeaways']
        temp_table2['Panthers_Giveaways'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.giveaways']
        temp_table2['Panthers_Hits'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.hits']
        temp_table2['Opponent_PIM'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.pim']

```

```

        temp_table2['Opponent_Shots'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.shots']
        temp_table2['Opponent_Power_Play_%'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.powerPlayPercentage']
        temp_table2['Opponent_Power_Play_Goals'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.powerPlayGoals']
        temp_table2['Opponent_Power_Play_Opportunities'] =
↳temp_table['teams.away.teamStats.teamSkaterStats.powerPlayOpportunities']
        temp_table2['Opponent_FOW%'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.faceOffWinPercentage']
        temp_table2['Opponent_Blocked_Shots'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.blocked']
        temp_table2['Opponent_Takeaways'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.takeaways']
        temp_table2['Opponent_Giveaways'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.giveaways']
        temp_table2['Opponent_Hits'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.hits']
    else:
        temp_table = pd.json_normalize(json)
        temp_table = temp_table[col_list]
        temp_table2['Panthers_PIM'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.pim']
        temp_table2['Panthers_Shots'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.shots']
        temp_table2['Panthers_Power_Play_%'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.powerPlayPercentage']
        temp_table2['Panthers_Power_Play_Goals'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.powerPlayGoals']
        temp_table2['Panthers_Power_Play_Opportunities'] =
↳temp_table['teams.away.teamStats.teamSkaterStats.powerPlayOpportunities']
        temp_table2['Panthers_FOW%'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.faceOffWinPercentage']
        temp_table2['Panthers_Blocked_Shots'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.blocked']
        temp_table2['Panthers_Takeaways'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.takeaways']
        temp_table2['Panthers_Giveaways'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.giveaways']
        temp_table2['Panthers_Hits'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.hits']
        temp_table2['Opponent_PIM'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.pim']
        temp_table2['Opponent_Shots'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.shots']

```

```

        temp_table2['Opponent_Power_Play_%'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.powerPlayPercentage']
        temp_table2['Opponent_Power_Play_Goals'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.powerPlayGoals']
        temp_table2['Opponent_Power_Play_Opportunities'] =
↳temp_table['teams.home.teamStats.teamSkaterStats.powerPlayOpportunities']
        temp_table2['Opponent_FOW%'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.faceOffWinPercentage']
        temp_table2['Opponent_Blocked_Shots'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.blocked']
        temp_table2['Opponent_Takeaways'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.takeaways']
        temp_table2['Opponent_Giveaways'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.giveaways']
        temp_table2['Opponent_Hits'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.hits']

    panthers_game_boxscore = pd.
↳concat([panthers_game_boxscore,temp_table2], ignore_index = True)

    panthers_game_boxscore['Panthers_Goals'] = panthers_goals_list
    panthers_game_boxscore['Opponent_Goals'] = opponent_goals_list

    table_col_list = panthers_game_boxscore.columns.values.tolist()

    for n in table_col_list:
        panthers_game_boxscores_table[n] = panthers_game_boxscore[n]

    return panthers_game_boxscores_table

```

```

[4]: def create_panthers_player_boxscores_table():
    panthers_player_boxscores_table = pd.DataFrame()
    game_id = list()
    player_id = list()

    season_list = list()
    season_list.append('20142015')
    season_list.append('20152016')
    season_list.append('20162017')
    season_list.append('20172018')
    season_list.append('20182019')
    season_list.append('20192020')
    season_list.append('20202021')

```

```

season_list.append('20212022')
season_list

for n in season_list:
    url = 'https://statsapi.web.nhl.com/api/v1/schedule?teamId=13&season='
    url = url + n
    r = requests.get(url)
    json = r.json()
    count = 0

    for n in json['dates']:
        if json['dates'][count]['totalItems'] == 1:
            game_id.append(json['dates'][count]['games'][0]['gamePk'])
        else:
            game_id.append(json['dates'][count]['games'][0]['gamePk'])
            game_id.append(json['dates'][count]['games'][1]['gamePk'])

        count = count + 1

url = 'https://statsapi.web.nhl.com/api/v1/game/2014010029/boxscore'
r = requests.get(url)
json = r.json()
goalie_keys = list()
player_keys = list()
for key in json['teams']['home']['players']['ID8468540']['stats']['goalieStats'].keys():
    goalie_keys.append(key)
for key in json['teams']['home']['players']['ID8475153']['stats']['skaterStats'].keys():
    player_keys.append(key)

game_keylist = player_keys + goalie_keys
game_keylist = list(dict.fromkeys(game_keylist))

player_data_table = pd.DataFrame()
long_list_game_id = list()
for n in game_id:
    url = 'https://statsapi.web.nhl.com/api/v1/game/' + str(n) + '/boxscore'
    r = requests.get(url)
    json = r.json()
    if json['teams']['home']['team']['id'] == 13:
        for key in json['teams']['home']['players'].keys():
            if len(json['teams']['home']['players'][key]['stats']) != 0:
                player_id.append(key[2:])
                long_list_game_id.append(n)

```

```

        if_
        ↪json['teams']['home']['players'][key]['position']['code'] == 'G':
            statlist =_
        ↪list(json['teams']['home']['players'][key]['stats']['goalieStats'].values())
            temp_table = pd.DataFrame(statlist).T
            temp_table.columns =_
        ↪list(json['teams']['home']['players'][key]['stats']['goalieStats'].keys())
            player_data_table = pd.
        ↪concat([player_data_table,temp_table], ignore_index = True)

        elif_
        ↪json['teams']['home']['players'][key]['position']['code'] == 'D' or_
        ↪json['teams']['home']['players'][key]['position']['code'] == 'L' or_
        ↪json['teams']['home']['players'][key]['position']['code'] == 'C' or_
        ↪json['teams']['home']['players'][key]['position']['code'] == 'R':
            statlist =_
        ↪list(json['teams']['home']['players'][key]['stats']['skaterStats'].values())
            temp_table = pd.DataFrame(statlist).T
            temp_table.columns =_
        ↪list(json['teams']['home']['players'][key]['stats']['skaterStats'].keys())
            player_data_table = pd.
        ↪concat([player_data_table,temp_table], ignore_index = True)

    else:
        for key in json['teams']['away']['players'].keys():
            if len(json['teams']['away']['players'][key]['stats']) != 0:
                player_id.append(key[2:])
                long_list_game_id.append(n)
            if_
        ↪json['teams']['away']['players'][key]['position']['code'] == 'G':
            statlist =_
        ↪list(json['teams']['away']['players'][key]['stats']['goalieStats'].values())
            temp_table = pd.DataFrame(statlist).T
            temp_table.columns =_
        ↪list(json['teams']['away']['players'][key]['stats']['goalieStats'].keys())
            player_data_table = pd.
        ↪concat([player_data_table,temp_table], ignore_index = True)

        elif_
        ↪json['teams']['away']['players'][key]['position']['code'] == 'D' or_
        ↪json['teams']['away']['players'][key]['position']['code'] == 'L' or_
        ↪json['teams']['away']['players'][key]['position']['code'] == 'C' or_
        ↪json['teams']['away']['players'][key]['position']['code'] == 'R':
            statlist =_
        ↪list(json['teams']['away']['players'][key]['stats']['skaterStats'].values())
            temp_table = pd.DataFrame(statlist).T
            temp_table.columns =_
        ↪list(json['teams']['away']['players'][key]['stats']['skaterStats'].keys())

```

```

        player_data_table = pd.
↳concat([player_data_table,temp_table], ignore_index = True)

    panthers_player_boxscores_table['Player_ID'] = player_id
    panthers_player_boxscores_table['Game_ID'] = long_list_game_id
    panthers_player_boxscores_table['timeOnIce'] =_
↳player_data_table['timeOnIce']
    panthers_player_boxscores_table['evenTimeOnIce'] =_
↳player_data_table['evenTimeOnIce']
    panthers_player_boxscores_table['goals'] = player_data_table['goals']
    panthers_player_boxscores_table['assists'] = player_data_table['assists']
    panthers_player_boxscores_table['shots'] = player_data_table['shots']
    panthers_player_boxscores_table['hits'] = player_data_table['hits']
    panthers_player_boxscores_table['powerPlayTimeOnIce'] =_
↳player_data_table['powerPlayTimeOnIce']
    panthers_player_boxscores_table['powerPlayGoals'] =_
↳player_data_table['powerPlayGoals']
    panthers_player_boxscores_table['powerPlayAssists'] =_
↳player_data_table['powerPlayAssists']
    panthers_player_boxscores_table['penaltyMinutes'] =_
↳player_data_table['penaltyMinutes']
    panthers_player_boxscores_table['pim'] = player_data_table['pim']
    panthers_player_boxscores_table['faceOffPct'] =_
↳player_data_table['faceOffPct']
    panthers_player_boxscores_table['faceOffWins'] =_
↳player_data_table['faceOffWins']
    panthers_player_boxscores_table['faceoffTaken'] =_
↳player_data_table['faceoffTaken']
    panthers_player_boxscores_table['takeaways'] =_
↳player_data_table['takeaways']
    panthers_player_boxscores_table['giveaways'] =_
↳player_data_table['giveaways']
    panthers_player_boxscores_table['shortHandedTimeOnIce'] =_
↳player_data_table['shortHandedTimeOnIce']
    panthers_player_boxscores_table['shortHandedGoals'] =_
↳player_data_table['shortHandedGoals']
    panthers_player_boxscores_table['shortHandedAssists'] =_
↳player_data_table['shortHandedAssists']
    panthers_player_boxscores_table['blocked'] = player_data_table['blocked']
    panthers_player_boxscores_table['plusMinus'] =_
↳player_data_table['plusMinus']
    panthers_player_boxscores_table['savePercentage'] =_
↳player_data_table['savePercentage']
    panthers_player_boxscores_table['saves'] = player_data_table['saves']

```

```

    panthers_player_boxscores_table['evenSaves'] =_
    ↪player_data_table['evenSaves']
    panthers_player_boxscores_table['evenShotsAgainst'] =_
    ↪player_data_table['evenShotsAgainst']
    panthers_player_boxscores_table['evenStrengthSavePercentage'] =_
    ↪player_data_table['evenStrengthSavePercentage']
    panthers_player_boxscores_table['powerPlaySaves'] =_
    ↪player_data_table['powerPlaySaves']
    panthers_player_boxscores_table['powerPlayShotsAgainst'] =_
    ↪player_data_table['powerPlayShotsAgainst']
    panthers_player_boxscores_table['powerPlaySavePercentage'] =_
    ↪player_data_table['powerPlaySavePercentage']
    panthers_player_boxscores_table['shortHandedSaves'] =_
    ↪player_data_table['shortHandedSaves']
    panthers_player_boxscores_table['shortHandedShotsAgainst'] =_
    ↪player_data_table['shortHandedShotsAgainst']

    short_save_pct_list = list()
    for index, entry in panthers_player_boxscores_table.iterrows():
        if entry['shortHandedShotsAgainst'] > 0:
            short_save_pct_list = entry['shortHandedSaves']/
            ↪entry['shortHandedShotsAgainst']
            short_save_pct_list.append(short_save_pct)
        else:
            short_save_pct_list.append(math.nan)

    panthers_player_boxscores_table['shortHandedSavePercentage'] =_
    ↪short_save_pct_list
    return panthers_player_boxscores_table

```

```

[4]: panthers_players_table = pd.DataFrame()
temp_table = pd.DataFrame()
count = 0

season_list = list()
season_list.append('20142015')
season_list.append('20152016')
season_list.append('20162017')
season_list.append('20172018')
season_list.append('20182019')
season_list.append('20192020')
season_list.append('20202021')
season_list.append('20212022')
season_list

for n in season_list:

```

```

url = 'https://statsapi.web.nhl.com/api/v1/teams/13?expand=team.
↳roster&season='
url = url + season_list[count]
r = requests.get(url)
json = r.json()
player_id = list()

for x in json['teams'][0]['roster']['roster']:
    player_id.append(x['person']['id'])

temp_table['Player_ID'] = player_id
for i in player_id:
    temp_table['Season'] = season_list[count]

panthers_players_table = pd.concat([panthers_players_table, temp_table],
↳ignore_index = True)
panthers_players_table.reset_index()
count = count + 1
temp_table = pd.DataFrame()

player_list = panthers_players_table.Player_ID.tolist()
player_data_table = pd.DataFrame()
count = 0
for n in panthers_players_table.Player_ID:
    url = 'https://statsapi.web.nhl.com/api/v1/people/'
    url = url + str(player_list[count])
    r = requests.get(url)
    json = r.json()
    temp_table = pd.json_normalize(json, record_path = ['people'])
    col_list = ['birthDate', 'height', 'weight']
    temp_table = temp_table[col_list]
    player_data_table = pd.concat([player_data_table, temp_table], ignore_index,
↳= True)
    temp_table = pd.DataFrame()
    count = count + 1

panthers_players_table['birthDate'] = player_data_table['birthDate']
panthers_players_table['height'] = player_data_table['height']
panthers_players_table['weight'] = player_data_table['weight']

player_keylist = list()
url1 = 'https://statsapi.web.nhl.com/api/v1/people/8448208/stats?
↳stats=statsSingleSeason&season=20142015'
r1 = requests.get(url1)
json1 = r1.json()

```



```

for key in json1['stats'][0]['splits'][0]['stat'].keys():
    player_keylist.append(key)

count = 0
for n in player_keylist:
    player_keylist[count] = 'stat.' + player_keylist[count]
    count = count + 1

goalie_keylist = list()
url2 = 'https://statsapi.web.nhl.com/api/v1/people/8481519/stats?
↳stats=statsSingleSeason&season=20212022'
r2 = requests.get(url2)
json2 = r2.json()
for key in json2['stats'][0]['splits'][0]['stat'].keys():
    goalie_keylist.append(key)

count = 0
for n in goalie_keylist:
    goalie_keylist[count] = 'stat.' + goalie_keylist[count]
    count = count + 1

keylist = player_keylist + goalie_keylist

new_keylist = list()
count = 0
for n in keylist:
    new_keylist.append(keylist[count][5:])
    count = count + 1

player_data_table = pd.DataFrame()
for index, entry in panthers_players_table.iterrows():
    url = 'https://statsapi.web.nhl.com/api/v1/people/'
    url = url + str(entry['Player_ID']) + '/stats?
↳stats=statsSingleSeason&season=' + str(entry['Season'])
    r = requests.get(url)
    json = r.json()
    if len(json['stats'][0]['splits']) == 0:
        for n in keylist:
            temp_table[n] = None
    else:
        temp_table = pd.json_normalize(json, record_path=['stats','splits'])
    player_data_table = pd.concat([player_data_table,temp_table], ignore_index=
↳ True)

for n in keylist:
    panthers_players_table[n] = player_data_table[n]

```

```

datelist = list()
cutofflist = list()
agelist = list()
for index, entry in panthers_players_table.iterrows():
    birthD = datetime.strptime(str(entry['birthDate']), '%Y-%m-%d')
    datelist.append(birthD.date())
    cutoffD = datetime.strptime(str(entry['Season'])[4:] + '-01-31', '%Y-%m-%d')
    cutofflist.append(cutoffD.date())
    ageYears = cutoffD.date().year - birthD.date().year - ((cutoffD.date().
↪ month, cutoffD.date().day) < (birthD.date().month, birthD.date().day))
    if (cutoffD.date().month, cutoffD.date().day) > (birthD.date().month,
↪ birthD.date().day):
        nextBirthD = datetime.strptime(str(entry['Season'])[4:] + str(birthD.
↪ date().month) + str(birthD.date().day), '%Y%m%d')
        diff = abs(nextBirthD - cutoffD)
        ageDays = diff.days
    else:
        lastBirthD = datetime.strptime(str(entry['Season'])[:4] + str(birthD.
↪ date().month) + str(birthD.date().day), '%Y%m%d')
        diff = abs(cutoffD - lastBirthD)
        ageDays = diff.days
    agelist.append(str(ageYears) + ' years, ' + str(ageDays) + ' days')

panthers_players_table['Age_in_Season'] = agelist

list_2014 = 0
list_2015 = 0
list_2016 = 0
list_2017 = 0
list_2018 = 0
list_2019 = 0
list_2020 = 0
list_2021 = 0

for n in season_list:
    url = 'https://statsapi.web.nhl.com/api/v1/schedule?teamId=13&season='
    url = url + n
    r = requests.get(url)
    json = r.json()
    count = 0
    for n in json['dates']:
        if json['dates'][count]['totalItems'] == 1:
            if str(json['dates'][count]['games'][0]['gamePk'])[:6] == '201402':
                list_2014 = list_2014 + 1
            elif str(json['dates'][count]['games'][0]['gamePk'])[:6] ==
↪ '201502':

```

```

        list_2015 = list_2015 + 1
        elif str(json['dates'][count]['games'][0]['gamePk'])[:6] == '
↪'201602':
            list_2016 = list_2016 + 1
            elif str(json['dates'][count]['games'][0]['gamePk'])[:6] == '
↪'201702':
                list_2017 = list_2017 + 1
                elif str(json['dates'][count]['games'][0]['gamePk'])[:6] == '
↪'201802':
                    list_2018 = list_2018 + 1
                    elif str(json['dates'][count]['games'][0]['gamePk'])[:6] == '
↪'201902':
                        list_2019 = list_2019 + 1
                        elif str(json['dates'][count]['games'][0]['gamePk'])[:6] == '
↪'202002':
                            list_2020 = list_2020 + 1
                            elif str(json['dates'][count]['games'][0]['gamePk'])[:6] == '
↪'202102':
                                list_2021 = list_2021 + 1

        count = count + 1
fullseason = list()
for index, entry in panthers_players_table.iterrows():
    if entry['Season'] == season_list[0]:
        if entry['stat.games'] == list_2014:
            fullseason.append('Yes')
        else:
            fullseason.append('No')
    elif entry['Season'] == season_list[1]:
        if entry['stat.games'] == list_2015:
            fullseason.append('Yes')
        else:
            fullseason.append('No')
    elif entry['Season'] == season_list[2]:
        if entry['stat.games'] == list_2016:
            fullseason.append('Yes')
        else:
            fullseason.append('No')
    elif entry['Season'] == season_list[3]:
        if entry['stat.games'] == list_2017:
            fullseason.append('Yes')
        else:
            fullseason.append('No')
    elif entry['Season'] == season_list[4]:
        if entry['stat.games'] == list_2018:
            fullseason.append('Yes')

```

```

        else:
            fullseason.append('No')
    elif entry['Season'] == season_list[5]:
        if entry['stat.games'] == list_2019:
            fullseason.append('Yes')
        else:
            fullseason.append('No')
    elif entry['Season'] == season_list[6]:
        if entry['stat.games'] == list_2020:
            fullseason.append('Yes')
        else:
            fullseason.append('No')
    elif entry['Season'] == season_list[7]:
        if entry['stat.games'] == list_2021:
            fullseason.append('Yes')
        else:
            fullseason.append('No')

panthers_players_table['Full_Season'] = fullseason

s = panthers_players_table['stat.pim']
s2 = s.astype('Int32')
panthers_players_table['stat.pim'] = s2

display(panthers_players_table)

```

	Player_ID	Season	birthDate	height	weight	stat.timeOnIce	\
0	8448208	20142015	1972-02-15	6' 3"	230	1352:54	
1	8465185	20142015	1977-04-23	6' 3"	210	1430:48	
2	8465978	20142015	1977-07-23	6' 2"	217	440:52	
3	8466285	20142015	1979-05-23	5' 10"	192	1903:05	
4	8468001	20142015	1981-06-11	5' 11"	177	1020:58	
..	
270	8482113	20212022	2001-10-03	6' 1"	185	1022:55	
271	8482641	20212022	1998-04-14	6' 0"	181	140:27	
272	8475683	20212022	1988-09-20	6' 2"	182	3082:10	
273	8477992	20212022	1995-09-19	6' 5"	220	486:31	
274	8481519	20212022	2001-04-19	6' 3"	192	1740:14	

	stat.assists	stat.goals	stat.pim	stat.shots	...	stat.savePercentage	\
0	30.0	17.0	48	169.0	...	NaN	
1	5.0	3.0	25	78.0	...	NaN	
2	4.0	1.0	50	53.0	...	NaN	
3	24.0	3.0	22	118.0	...	NaN	
4	6.0	5.0	45	72.0	...	NaN	
..	
270	26	18	18	125	...	NaN	

271	1	1	2	11	...	NaN
272	NaN	NaN	<NA>	NaN	...	0.913
273	NaN	NaN	<NA>	NaN	...	0.856
274	NaN	NaN	<NA>	NaN	...	0.908

	stat.goalAgainstAverage	stat.gamesStarted	stat.shotsAgainst	\
0	NaN	NaN	NaN	
1	NaN	NaN	NaN	
2	NaN	NaN	NaN	
3	NaN	NaN	NaN	
4	NaN	NaN	NaN	
..	
270	NaN	NaN	NaN	
271	NaN	NaN	NaN	
272	2.667	53	1566	
273	4.6864	8	264	
274	2.7927	27	876	

	stat.goalsAgainst	stat.powerPlaySavePercentage	\
0	NaN	NaN	
1	NaN	NaN	
2	NaN	NaN	
3	NaN	NaN	
4	NaN	NaN	
..	
270	NaN	NaN	
271	NaN	NaN	
272	137	89.883268	
273	38	82.608696	
274	81	84.662577	

	stat.shortHandedSavePercentage	stat.evenStrengthSavePercentage	\
0	NaN	NaN	
1	NaN	NaN	
2	NaN	NaN	
3	NaN	NaN	
4	NaN	NaN	
..	
270	NaN	NaN	
271	NaN	NaN	
272	89.361702	91.600634	
273	100.0	84.716157	
274	86.363636	92.329957	

	Age_in_Season	Full_Season
0	42 years, 350 days	No
1	37 years, 283 days	No
2	37 years, 192 days	No

3	35 years, 253 days	Yes
4	33 years, 234 days	Yes
..
270	20 years, 120 days	No
271	23 years, 292 days	No
272	33 years, 133 days	No
273	26 years, 134 days	No
274	20 years, 287 days	No

[275 rows x 54 columns]

```
[3]: panthers_game_boxscores_table = pd.DataFrame()
game_id = list()

season_list = list()
season_list.append('20142015')
season_list.append('20152016')
season_list.append('20162017')
season_list.append('20172018')
season_list.append('20182019')
season_list.append('20192020')
season_list.append('20202021')
season_list.append('20212022')
season_list

for n in season_list:
    url = 'https://statsapi.web.nhl.com/api/v1/schedule?teamId=13&season='
    url = url + n
    r = requests.get(url)
    json = r.json()
    count = 0

    for n in json['dates']:
        if json['dates'][count]['totalItems'] == 1:
            game_id.append(json['dates'][count]['games'][0]['gamePk'])
        else:
            game_id.append(json['dates'][count]['games'][0]['gamePk'])
            game_id.append(json['dates'][count]['games'][1]['gamePk'])

        count = count + 1

panthers_game_boxscores_table['Game_ID'] = game_id

home_or_away = list()
for index, entry in panthers_game_boxscores_table.iterrows():
    url = 'https://statsapi.web.nhl.com/api/v1/game/' + str(entry['Game_ID']) +
    ↪ '/boxscore'
```

```

r = requests.get(url)
json = r.json()
if json['teams']['away']['team']['id'] == 13:
    home_or_away.append('Away')
else:
    home_or_away.append('Home')

panthers_game_boxscores_table['Home_or_Away'] = home_or_away

result = list()
for index, entry in panthers_game_boxscores_table.iterrows():
    url = 'https://statsapi.web.nhl.com/api/v1/game/' + str(entry['Game_ID']) +
    ↪ '/boxscore'
    r = requests.get(url)
    json = r.json()
    if entry['Home_or_Away'] == 'Home':
        panthersScore =
    ↪ json['teams']['home']['teamStats']['teamSkaterStats']['goals']
        opponentScore =
    ↪ json['teams']['away']['teamStats']['teamSkaterStats']['goals']
        scoreDiff = panthersScore - opponentScore
        if scoreDiff > 0:
            result.append('Win')
        else:
            url2 = 'https://statsapi.web.nhl.com/api/v1/game/' +
    ↪ str(entry['Game_ID']) + '/linescore'
            r2 = requests.get(url2)
            json2 = r2.json()
            if json2['currentPeriod'] > 3:
                if json2['teams']['home']['goals'] >
    ↪ json2['teams']['away']['goals']:
                    result.append('Win')
                else:
                    result.append('OT Loss')
            else:
                result.append('Loss')
        else:
            opponentScore =
    ↪ json['teams']['home']['teamStats']['teamSkaterStats']['goals']
            panthersScore =
    ↪ json['teams']['away']['teamStats']['teamSkaterStats']['goals']
            scoreDiff = panthersScore - opponentScore
            if scoreDiff > 0:
                result.append('Win')
            else:

```

```

        url2 = 'https://statsapi.web.nhl.com/api/v1/game/' + str(entry['Game_ID']) + '/linescore'
        r2 = requests.get(url2)
        json2 = r2.json()
        if json2['currentPeriod'] > 3:
            if json2['teams']['home']['goals'] < json2['teams']['away']['goals']:
                result.append('Win')
            else:
                result.append('OT Loss')
        else:
            result.append('Loss')

panthers_game_boxscores_table['Result'] = result

panthers_goals_list = list()
opponent_goals_list = list()
for index, entry in panthers_game_boxscores_table.iterrows():
    url2 = 'https://statsapi.web.nhl.com/api/v1/game/' + str(entry['Game_ID']) + '/linescore'
    r2 = requests.get(url2)
    json2 = r2.json()
    if json2['teams']['home']['team']['id'] == 13:
        panthers_goals_list.append(json2['teams']['home']['goals'])
        opponent_goals_list.append(json2['teams']['away']['goals'])
    else:
        panthers_goals_list.append(json2['teams']['away']['goals'])
        opponent_goals_list.append(json2['teams']['home']['goals'])

panthers_game_boxscores_table['Panthers_Goals'] = panthers_goals_list
panthers_game_boxscores_table['Opponent_Goals'] = opponent_goals_list

panthers_game_boxscore = pd.DataFrame()
temp_table2 = pd.DataFrame()

```



```

col_list = ['teams.away.teamStats.teamSkaterStats.pim', 'teams.away.teamStats.
↳teamSkaterStats.shots', 'teams.away.teamStats.teamSkaterStats.
↳powerPlayPercentage', 'teams.away.teamStats.teamSkaterStats.
↳powerPlayGoals', 'teams.away.teamStats.teamSkaterStats.
↳powerPlayOpportunities', 'teams.away.teamStats.teamSkaterStats.
↳faceOffWinPercentage', 'teams.away.teamStats.teamSkaterStats.blocked', 'teams.
↳away.teamStats.teamSkaterStats.takeaways', 'teams.away.teamStats.
↳teamSkaterStats.giveaways', 'teams.away.teamStats.teamSkaterStats.
↳hits', 'teams.home.teamStats.teamSkaterStats.pim', 'teams.home.teamStats.
↳teamSkaterStats.shots', 'teams.home.teamStats.teamSkaterStats.
↳powerPlayPercentage', 'teams.home.teamStats.teamSkaterStats.
↳powerPlayGoals', 'teams.home.teamStats.teamSkaterStats.
↳powerPlayOpportunities', 'teams.home.teamStats.teamSkaterStats.
↳faceOffWinPercentage', 'teams.home.teamStats.teamSkaterStats.blocked', 'teams.
↳home.teamStats.teamSkaterStats.takeaways', 'teams.home.teamStats.
↳teamSkaterStats.giveaways', 'teams.home.teamStats.teamSkaterStats.hits']
for index, entry in panthers_game_boxscores_table.iterrows():
    url = 'https://statsapi.web.nhl.com/api/v1/game/' + str(entry['Game_ID']) +
↳'/boxscore'
    r = requests.get(url)
    json = r.json()
    if entry['Home_or_Away'] == 'Home':
        temp_table = pd.json_normalize(json)
        temp_table = temp_table[col_list]
        temp_table2['Panthers_PIM'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.pim']
        temp_table2['Panthers_Shots'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.shots']
        temp_table2['Panthers_Power_Play_%'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.powerPlayPercentage']
        temp_table2['Panthers_Power_Play_Goals'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.powerPlayGoals']
        temp_table2['Panthers_Power_Play_Opportunities'] = temp_table['teams.
↳home.teamStats.teamSkaterStats.powerPlayOpportunities']
        temp_table2['Panthers_FOW%'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.faceOffWinPercentage']
        temp_table2['Panthers_Blocked-Shots'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.blocked']
        temp_table2['Panthers_Takeaways'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.takeaways']
        temp_table2['Panthers_Giveaways'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.giveaways']
        temp_table2['Panthers_Hits'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.hits']
        temp_table2['Opponent_PIM'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.pim']

```

```

temp_table2['Opponent_Shots'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.shots']
temp_table2['Opponent_Power_Play_%'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.powerPlayPercentage']
temp_table2['Opponent_Power_Play_Goals'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.powerPlayGoals']
temp_table2['Opponent_Power_Play_Opportunities'] = temp_table['teams.
↳away.teamStats.teamSkaterStats.powerPlayOpportunities']
temp_table2['Opponent_FOW%'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.faceOffWinPercentage']
temp_table2['Opponent_Blocked_Shots'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.blocked']
temp_table2['Opponent_Takeaways'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.takeaways']
temp_table2['Opponent_Giveaways'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.giveaways']
temp_table2['Opponent_Hits'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.hits']
else:
temp_table = pd.json_normalize(json)
temp_table = temp_table[col_list]
temp_table2['Panthers_PIM'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.pim']
temp_table2['Panthers_Shots'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.shots']
temp_table2['Panthers_Power_Play_%'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.powerPlayPercentage']
temp_table2['Panthers_Power_Play_Goals'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.powerPlayGoals']
temp_table2['Panthers_Power_Play_Opportunities'] = temp_table['teams.
↳away.teamStats.teamSkaterStats.powerPlayOpportunities']
temp_table2['Panthers_FOW%'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.faceOffWinPercentage']
temp_table2['Panthers_Blocked_Shots'] = temp_table['teams.away.
↳teamStats.teamSkaterStats.blocked']
temp_table2['Panthers_Takeaways'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.takeaways']
temp_table2['Panthers_Giveaways'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.giveaways']
temp_table2['Panthers_Hits'] = temp_table['teams.away.teamStats.
↳teamSkaterStats.hits']
temp_table2['Opponent_PIM'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.pim']
temp_table2['Opponent_Shots'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.shots']

```

```

        temp_table2['Opponent_Power_Play_%'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.powerPlayPercentage']
        temp_table2['Opponent_Power_Play_Goals'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.powerPlayGoals']
        temp_table2['Opponent_Power_Play_Opportunities'] = temp_table['teams.
↳home.teamStats.teamSkaterStats.powerPlayOpportunities']
        temp_table2['Opponent_FOW%'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.faceOffWinPercentage']
        temp_table2['Opponent_Blocked_Shots'] = temp_table['teams.home.
↳teamStats.teamSkaterStats.blocked']
        temp_table2['Opponent_Takeaways'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.takeaways']
        temp_table2['Opponent_Giveaways'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.giveaways']
        temp_table2['Opponent_Hits'] = temp_table['teams.home.teamStats.
↳teamSkaterStats.hits']

    panthers_game_boxscore = pd.concat([panthers_game_boxscore,temp_table2],
↳ignore_index = True)

panthers_game_boxscore['Panthers_Goals'] = panthers_goals_list
panthers_game_boxscore['Opponent_Goals'] = opponent_goals_list

table_col_list = panthers_game_boxscore.columns.values.tolist()

for n in table_col_list:
    panthers_game_boxscores_table[n] = panthers_game_boxscore[n]

display(panthers_game_boxscores_table)

```

	Game_ID	Home_or_Away	Result	Panthers_Goals	Opponent_Goals	\
0	2014010029	Home	OT Loss	3	4	
1	2014010051	Away	OT Loss	1	2	
2	2014010052	Away	Loss	1	4	
3	2014010065	Away	Loss	4	5	
4	2014010088	Home	Loss	0	3	
..	
685	2021030116	Away	Win	4	3	
686	2021030211	Home	Loss	1	4	
687	2021030212	Home	Loss	1	2	
688	2021030213	Away	Loss	1	5	
689	2021030214	Away	Loss	0	2	

Panthers_PIM Panthers_Shots Panthers_Power_Play_% \

0	17	25	12.5
1	2	24	25.0
2	18	23	0.0
3	10	24	0.0
4	12	21	0.0
..
685	10	31	0.0
686	14	34	0.0
687	6	36	0.0
688	4	35	33.3
689	8	49	0.0

	Panthers_Power_Play_Goals	Panthers_Power_Play_Opportunities	...	\
0	1.0	8.0	...	
1	1.0	4.0	...	
2	0.0	2.0	...	
3	0.0	1.0	...	
4	0.0	3.0	...	
..	
685	0.0	2.0	...	
686	0.0	3.0	...	
687	0.0	4.0	...	
688	1.0	3.0	...	
689	0.0	3.0	...	

	Opponent_PIM	Opponent_Shots	Opponent_Power_Play_%	\
0	21	31	16.7	
1	8	20	0.0	
2	18	25	0.0	
3	2	28	40.0	
4	12	33	0.0	
..	
685	6	37	25.0	
686	8	36	50.0	
687	8	28	33.3	
688	6	36	0.0	
689	6	26	0.0	

	Opponent_Power_Play_Goals	Opponent_Power_Play_Opportunities	\
0	1.0	6.0	
1	0.0	1.0	
2	0.0	2.0	
3	2.0	5.0	
4	0.0	3.0	
..	
685	1.0	4.0	
686	3.0	6.0	
687	1.0	3.0	

688	0.0	2.0
689	0.0	4.0

	Opponent_FOW%	Opponent_Blocked-Shots	Opponent_Takeaways \
0	55.9	14	2
1	73.1	14	4
2	40.4	8	5
3	51.4	4	9
4	52.5	12	1
..
685	48.7	15	9
686	49.2	16	3
687	43.2	24	4
688	44.6	19	10
689	58.2	18	5

	Opponent_Giveaways	Opponent_Hits
0	2	17
1	10	15
2	6	13
3	12	20
4	0	27
..
685	17	51
686	16	30
687	9	35
688	5	33
689	2	29

[690 rows x 25 columns]

```
[6]: panthers_player_boxscores_table = pd.DataFrame()
game_id = list()
player_id = list()

season_list = list()
season_list.append('20142015')
season_list.append('20152016')
season_list.append('20162017')
season_list.append('20172018')
season_list.append('20182019')
season_list.append('20192020')
season_list.append('20202021')
season_list.append('20212022')
season_list

for n in season_list:
```

```

url = 'https://statsapi.web.nhl.com/api/v1/schedule?teamId=13&season='
url = url + n
r = requests.get(url)
json = r.json()
count = 0

for n in json['dates']:
    if json['dates'][count]['totalItems'] == 1:
        game_id.append(json['dates'][count]['games'][0]['gamePk'])
    else:
        game_id.append(json['dates'][count]['games'][0]['gamePk'])
        game_id.append(json['dates'][count]['games'][1]['gamePk'])

    count = count + 1

url = 'https://statsapi.web.nhl.com/api/v1/game/2014010029/boxscore'
r = requests.get(url)
json = r.json()
goalie_keys = list()
player_keys = list()
for key in json['teams']['home']['players']['ID8468540']['stats']['goalieStats'].keys():
    goalie_keys.append(key)
for key in json['teams']['home']['players']['ID8475153']['stats']['skaterStats'].keys():
    player_keys.append(key)

game_keylist = player_keys + goalie_keys
game_keylist = list(dict.fromkeys(game_keylist))

player_data_table = pd.DataFrame()
long_list_game_id = list()
for n in game_id:
    url = 'https://statsapi.web.nhl.com/api/v1/game/' + str(n) + '/boxscore'
    r = requests.get(url)
    json = r.json()
    if json['teams']['home']['team']['id'] == 13:
        for key in json['teams']['home']['players'].keys():
            if len(json['teams']['home']['players'][key]['stats']) != 0:
                player_id.append(key[2:])
                long_list_game_id.append(n)
                if json['teams']['home']['players'][key]['position']['code'] == 'G':
                    statlist = list(json['teams']['home']['players'][key]['stats']['goalieStats'].values())
                    temp_table = pd.DataFrame(statlist).T

```

```

        temp_table.columns =_
        list(json['teams']['home']['players'][key]['stats']['goalieStats'].keys())
        player_data_table = pd.
        concat([player_data_table,temp_table], ignore_index = True)
        elif json['teams']['home']['players'][key]['position']['code']_
        == 'D' or json['teams']['home']['players'][key]['position']['code'] == 'L'_
        or json['teams']['home']['players'][key]['position']['code'] == 'C' or_
        json['teams']['home']['players'][key]['position']['code'] == 'R':
            statlist =_
            list(json['teams']['home']['players'][key]['stats']['skaterStats'].values())
            temp_table = pd.DataFrame(statlist).T
            temp_table.columns =_
            list(json['teams']['home']['players'][key]['stats']['skaterStats'].keys())
            player_data_table = pd.
            concat([player_data_table,temp_table], ignore_index = True)

    else:
        for key in json['teams']['away']['players'].keys():
            if len(json['teams']['away']['players'][key]['stats']) != 0:
                player_id.append(key[2:])
                long_list_game_id.append(n)
                if json['teams']['away']['players'][key]['position']['code'] ==_
        'G':
            statlist =_
            list(json['teams']['away']['players'][key]['stats']['goalieStats'].values())
            temp_table = pd.DataFrame(statlist).T
            temp_table.columns =_
            list(json['teams']['away']['players'][key]['stats']['goalieStats'].keys())
            player_data_table = pd.
            concat([player_data_table,temp_table], ignore_index = True)
            elif json['teams']['away']['players'][key]['position']['code']_
            == 'D' or json['teams']['away']['players'][key]['position']['code'] == 'L'_
            or json['teams']['away']['players'][key]['position']['code'] == 'C' or_
            json['teams']['away']['players'][key]['position']['code'] == 'R':
                statlist =_
                list(json['teams']['away']['players'][key]['stats']['skaterStats'].values())
                temp_table = pd.DataFrame(statlist).T
                temp_table.columns =_
                list(json['teams']['away']['players'][key]['stats']['skaterStats'].keys())
                player_data_table = pd.
                concat([player_data_table,temp_table], ignore_index = True)

panthers_player_boxscores_table['Player_ID'] = player_id
panthers_player_boxscores_table['Game_ID'] = long_list_game_id
panthers_player_boxscores_table['timeOnIce'] = player_data_table['timeOnIce']

```

```

panthers_player_boxscores_table['evenTimeOnIce'] =_
↳player_data_table['evenTimeOnIce']
panthers_player_boxscores_table['goals'] = player_data_table['goals']
panthers_player_boxscores_table['assists'] = player_data_table['assists']
panthers_player_boxscores_table['shots'] = player_data_table['shots']
panthers_player_boxscores_table['hits'] = player_data_table['hits']
panthers_player_boxscores_table['powerPlayTimeOnIce'] =_
↳player_data_table['powerPlayTimeOnIce']
panthers_player_boxscores_table['powerPlayGoals'] =_
↳player_data_table['powerPlayGoals']
panthers_player_boxscores_table['powerPlayAssists'] =_
↳player_data_table['powerPlayAssists']
panthers_player_boxscores_table['penaltyMinutes'] =_
↳player_data_table['penaltyMinutes']
panthers_player_boxscores_table['pim'] = player_data_table['pim']
panthers_player_boxscores_table['faceOffPct'] = player_data_table['faceOffPct']
panthers_player_boxscores_table['faceOffWins'] =_
↳player_data_table['faceOffWins']
panthers_player_boxscores_table['faceoffTaken'] =_
↳player_data_table['faceoffTaken']
panthers_player_boxscores_table['takeaways'] = player_data_table['takeaways']
panthers_player_boxscores_table['giveaways'] = player_data_table['giveaways']
panthers_player_boxscores_table['shortHandedTimeOnIce'] =_
↳player_data_table['shortHandedTimeOnIce']
panthers_player_boxscores_table['shortHandedGoals'] =_
↳player_data_table['shortHandedGoals']
panthers_player_boxscores_table['shortHandedAssists'] =_
↳player_data_table['shortHandedAssists']
panthers_player_boxscores_table['blocked'] = player_data_table['blocked']
panthers_player_boxscores_table['plusMinus'] = player_data_table['plusMinus']
panthers_player_boxscores_table['savePercentage'] =_
↳player_data_table['savePercentage']
panthers_player_boxscores_table['saves'] = player_data_table['saves']
panthers_player_boxscores_table['evenSaves'] = player_data_table['evenSaves']
panthers_player_boxscores_table['evenShotsAgainst'] =_
↳player_data_table['evenShotsAgainst']
panthers_player_boxscores_table['evenStrengthSavePercentage'] =_
↳player_data_table['evenStrengthSavePercentage']
panthers_player_boxscores_table['powerPlaySaves'] =_
↳player_data_table['powerPlaySaves']
panthers_player_boxscores_table['powerPlayShotsAgainst'] =_
↳player_data_table['powerPlayShotsAgainst']
panthers_player_boxscores_table['powerPlaySavePercentage'] =_
↳player_data_table['powerPlaySavePercentage']
panthers_player_boxscores_table['shortHandedSaves'] =_
↳player_data_table['shortHandedSaves']

```



```

panthers_player_boxscores_table['shortHandedShotsAgainst'] =
↳player_data_table['shortHandedShotsAgainst']

short_save_pct_list = list()
for index, entry in panthers_player_boxscores_table.iterrows():
    if entry['shortHandedShotsAgainst'] > 0:
        short_save_pct = entry['shortHandedSaves']/
↳entry['shortHandedShotsAgainst']
        short_save_pct_list.append(short_save_pct)
    else:
        short_save_pct_list.append(math.nan)

panthers_player_boxscores_table['shortHandedSavePercentage'] =
↳short_save_pct_list

display(panthers_player_boxscores_table)

```

	Player_ID	Game_ID	timeOnIce	evenTimeOnIce	goals	assists	shots	hits	\
0	8468540	2014010029	39:51	NaN	0	0	11	NaN	
1	8474625	2014010029	19:03	14:38	0	0	1	1	
2	8470105	2014010029	16:48	10:34	0	1	0	0	
3	8475755	2014010029	21:38	17:54	0	0	1	3	
4	8475760	2014010029	15:39	9:19	0	0	4	1	
...	
13166	8477986	2021030214	15:28	13:20	0	0	2	2	
13167	8475279	2021030214	14:24	12:13	0	0	2	5	
13168	8479553	2021030214	10:56	10:54	0	0	0	1	
13169	8477493	2021030214	21:25	14:32	0	0	2	1	
13170	8482113	2021030214	11:50	9:59	0	0	1	1	

	powerPlayTimeOnIce	powerPlayGoals	...	saves	evenSaves	evenShotsAgainst	\
0	NaN	NaN	...	9	7	8	
1	3:46	0	...	NaN	NaN	NaN	
2	5:01	0	...	NaN	NaN	NaN	
3	1:56	0	...	NaN	NaN	NaN	
4	4:50	0	...	NaN	NaN	NaN	
...	
13166	2:02	0	...	NaN	NaN	NaN	
13167	0:00	0	...	NaN	NaN	NaN	
13168	0:02	0	...	NaN	NaN	NaN	
13169	4:04	0	...	NaN	NaN	NaN	
13170	0:00	0	...	NaN	NaN	NaN	

	evenStrengthSavePercentage	powerPlaySaves	powerPlayShotsAgainst	\
0	87.5	1	2	
1	NaN	NaN	NaN	
2	NaN	NaN	NaN	

3	NaN	NaN	NaN
4	NaN	NaN	NaN
...
13166	NaN	NaN	NaN
13167	NaN	NaN	NaN
13168	NaN	NaN	NaN
13169	NaN	NaN	NaN
13170	NaN	NaN	NaN

	powerPlaySavePercentage	shortHandedSaves	shortHandedShotsAgainst	\
0	50.0	1	1	
1	NaN	NaN	NaN	
2	NaN	NaN	NaN	
3	NaN	NaN	NaN	
4	NaN	NaN	NaN	
...	
13166	NaN	NaN	NaN	
13167	NaN	NaN	NaN	
13168	NaN	NaN	NaN	
13169	NaN	NaN	NaN	
13170	NaN	NaN	NaN	

	shortHandedSavePercentage
0	1.0
1	NaN
2	NaN
3	NaN
4	NaN
...	...
13166	NaN
13167	NaN
13168	NaN
13169	NaN
13170	NaN

[13171 rows x 34 columns]

[7]: *#Which five players have played the most full seasons for the Panthers in the time frame we pulled?*

```
q = """
SELECT Player_ID, COUNT(*) as 'Number of Full Seasons'
FROM panthers_players_table
WHERE Full_Season = 'Yes'
GROUP BY Player_ID
ORDER BY COUNT(*)DESC
LIMIT 5
```

```
"""
```

```
q1_answer = pysqldf(q)
print(q1_answer)
```

	Player_ID	Number of Full Seasons
0	8471735	5
1	8476456	3
2	8478366	2
3	8477932	2
4	8476389	2

[8]: *#Which player had the highest plus-minus in an individual game and what was the result for the Panthers?*

```
q = """
SELECT panthers_player_boxscores_table.
    ↪Player_ID,panthers_player_boxscores_table.Game_ID,
    ↪panthers_player_boxscores_table.plusMinus,panthers_game_boxscores_table.
    ↪Result
FROM panthers_player_boxscores_table
JOIN panthers_game_boxscores_table
ORDER BY plusMinus DESC
LIMIT 1
"""
q2_answer = pysqldf(q)
print(q2_answer)
```

	Player_ID	Game_ID	plusMinus	Result
0	8477407	2021020661	6	OT Loss

[9]: *#Who were the youngest players to play for the Panthers in this time period, that were also on the team in 2021-22, and what were the first games they played for the Panthers?*

```
q = """
SELECT panthers_players_table.Player_ID,panthers_players_table.Season,
    ↪panthers_players_table.Age_In_Season, panthers_player_boxscores_table.Game_ID
FROM panthers_players_table
JOIN panthers_player_boxscores_table
WHERE panthers_player_boxscores_table.Game_ID LIKE '2017%' AND
    ↪panthers_player_boxscores_table.Player_ID = '8480015'
GROUP BY panthers_players_table.Player_ID
HAVING COUNT(DISTINCT Season) > 1
ORDER BY Age_In_Season ASC
LIMIT 1
"""
```

```
q3_answer = pysqldf(q)
print(q3_answer)
```

	Player_ID	Season	Age_in_Season	Game_ID
0	8480015	20172018	18 years, 349 days	2017010017

[10]: *#Who had the most total penalty minutes in a season for the Panthers, and what season did this take place in?*

```
q = """
SELECT Player_ID, MAX([stat.pim]), Season
FROM panthers_players_table
ORDER BY [stat.pim] DESC
"""
q4_answer = pysqldf(q)
print(q4_answer)
```

	Player_ID	MAX([stat.pim])	Season
0	8474230	212	20172018

[5]: *#Rank Panthers seasons by number of total home shutouts*

```
q = """
SELECT COUNT(Home_or_Away = 'Home') AS 'Home_Shutouts', '20142015' AS 'Season'
FROM panthers_game_boxscores_table
WHERE Game_ID LIKE '2014%' AND Opponent_Goals = 0 AND Home_or_Away = 'Home'
UNION
SELECT COUNT(Home_or_Away = 'Home') AS 'Home_Shutouts', '20152016' AS 'Season'
FROM panthers_game_boxscores_table
WHERE Game_ID LIKE '2015%' AND Opponent_Goals = 0 AND Home_or_Away = 'Home'
UNION
SELECT COUNT(Home_or_Away = 'Home') AS 'Home_Shutouts', '20162017' AS 'Season'
FROM panthers_game_boxscores_table
WHERE Game_ID LIKE '2016%' AND Opponent_Goals = 0 AND Home_or_Away = 'Home'
UNION
SELECT COUNT(Home_or_Away = 'Home') AS 'Home_Shutouts', '20172018' AS 'Season'
FROM panthers_game_boxscores_table
WHERE Game_ID LIKE '2017%' AND Opponent_Goals = 0 AND Home_or_Away = 'Home'
UNION
SELECT COUNT(Home_or_Away = 'Home') AS 'Home_Shutouts', '20182019' AS 'Season'
FROM panthers_game_boxscores_table
WHERE Game_ID LIKE '2018%' AND Opponent_Goals = 0 AND Home_or_Away = 'Home'
UNION
SELECT COUNT(Home_or_Away = 'Home') AS 'Home_Shutouts', '20192020' AS 'Season'
FROM panthers_game_boxscores_table
WHERE Game_ID LIKE '2019%' AND Opponent_Goals = 0 AND Home_or_Away = 'Home'
UNION
```

```

SELECT COUNT(Home_or_Away = 'Home') AS 'Home_Shutouts', '20202021' AS 'Season'
FROM panthers_game_boxscores_table
WHERE Game_ID LIKE '2020%' AND Opponent_Goals = 0 AND Home_or_Away = 'Home'
UNION
SELECT COUNT(Home_or_Away = 'Home') AS 'Home_Shutouts', '20212022' AS 'Season'
FROM panthers_game_boxscores_table
WHERE Game_ID LIKE '2021%' AND Opponent_Goals = 0 AND Home_or_Away = 'Home'
ORDER BY [Home_Shutouts]DESC
"""
q1_answer = pysqldf(q)
print(q1_answer)

```

	Home_Shutouts	Season
0	4	20172018
1	3	20152016
2	3	20202021
3	2	20162017
4	2	20192020
5	2	20212022
6	1	20142015
7	1	20182019

[]: