Scraping

May 4, 2020

1 Data Scraping

[ ]:
```
from bs4 import BeautifulSoup
import urllib.request
import pandas as pd
import numpy as np
import requests
```

1.1 Roster

Read in the roster from masters.com. This includes amatuers, so we need to take those out.
Source: https://www.masters.com/en_US/players/invitees_2020.html
We copied and pasted the table into an excel sheet and did some initial cleaning.

[ ]:
```
roster = pd.read_excel('players.xlsx')
```

[ ]:
```
def reverse_name(name):
    '''Goes from `last, first` to `first last`
        Also removes "#" from the name
    '''
    name = name.replace('#', "")
    name_lst = name.split(',')
    name_lst = [name.strip() for name in name_lst[::-1]]
    return " ".join(name_lst)
```

[ ]:
```
players_2020 = []
for row in roster.itertuples():
    if "*" not in row.Name:
        players_2020.append(reverse_name(row.Name))
```

[ ]:
```
s = pd.Series(players_2020)
```

Save the cleaned data to a csv file.

[ ]:
```
s.to_csv('2020_players.csv')
```

1.2 Scraping Tournment Data

1.2.1 Get links to all of the tournements in the past 10 years

[ ]:
```
base_link = "https://www.espn.com/golf/schedule/_/season/"
main_links = [
    'https://www.espn.com/golf/schedule/_/season/2010',
    'https://www.espn.com/golf/schedule/_/season/2011',
    'https://www.espn.com/golf/schedule/_/season/2012',
    'https://www.espn.com/golf/schedule/_/season/2013',
    'https://www.espn.com/golf/schedule/_/season/2014',
    'https://www.espn.com/golf/schedule/_/season/2015',
    'https://www.espn.com/golf/schedule/_/season/2016',
    'https://www.espn.com/golf/schedule/_/season/2017',
    'https://www.espn.com/golf/schedule/_/season/2018',
    'https://www.espn.com/golf/schedule/_/season/2019',
    'https://www.espn.com/golf/schedule/_/season/2020'
]
```

1.2.2 Extract all tournement links from the table

There are tournement links and player links. Tournement links do not have "player" in the path.

[ ]:
```
tournement_links = {}
# For each year
for link in main_links:
    year = link.split('/')[-1]
    tournement_links[year] = []
    source = urllib.request.urlopen(link).read()
    soup = BeautifulSoup(source,'lxml')
    table_titles = soup.findAll("section", {"class" : "ResponsiveTable"})
    # Find the completed tournements table
    for table in table_titles:
        title = table.find("div", {"class" : "Table__Title"})
        if title.text == "Completed Tournaments":
            # This is the one that we want
            # Still saved in table
            break
    links = table.findAll('a', {'class' : "AnchorLink"})
    # get all tournements in the table
    for link in links:
        href = link.attrs['href']
        if "player" not in href.split('/'):
            tournement_links[year].append(href)
```

1.2.3 Parse tournement results

This saves the table from each tournmemnt in a pandas dataframe and saves it to a csv file in the data folder. The data folder has folder for each year which contains a the csv file for each tournement.

[ ]:
```
def get_tournement_results(link):
    source = urllib.request.urlopen(link).read()
    soup = BeautifulSoup(source,'lxml')

    compet_table = soup.find("div", {"class" : "competitors"})
    tables = compet_table.find_all("section", {"class" : "ResponsiveTable"})
    for table in tables:
        #Get headings
        headings = []
        headings_tag = table.find('thead')
        head_cells = headings_tag.findAll("th")
        if len(head_cells) < 8:
            continue

        for heading in headings_tag.findAll("th"):
                headings.append(heading.find('a').text)

        body = table.find("tbody")
        rows = body.findAll('tr')
        player_data = []
        for row in rows:
            current_row = []
            for text in row.findAll("td"):
                current_row.append(text.text)

            player_data.append(current_row)
        return [headings] + player_data
```

[ ]:
```
data = {}
failed_links = []
for year in tournement_links:
    for link in tournement_links[year]:
        data[year] = []
        try:
            results = get_tournement_results(link)

            df = pd.DataFrame(results[1:], columns=results[0]).
 ↪set_index("PLAYER")
            df[['R1', 'R2', 'R3', 'R4']] = df[['R1', 'R2', 'R3', 'R4']].
 ↪replace("--", np.nan).astype(float)

            df.to_csv('data/' + str(year) + '/' + link.split('=')[-1])
            data[year].append(df)
```

```
        except Exception as err:
            failed_links.append(link)
            print(link)
            print(err)
```

[ ]: