棒球勝率分析

高嘉妤、柯堯珹、趙友誠、吳承恩

Table of contents

請利用連結 https://www.cpbl.com.tw/standings/season 中的資料，

利用 Bradley-Terry model 分析各個球隊的戰績。

使用 beautifulsoup4 爬資料

```python
# conda install anaconda::beautifulsoup4
# conda install anaconda::requests
# conda install anaconda::pandas
import requests
from bs4 import BeautifulSoup
import pandas as pd
url = "https://www.cpbl.com.tw/standings/season"
response = requests.get(url,headers = {"User-Agent":"Mozilla/5.0"})
response.encoding = 'utf-8'

soup = BeautifulSoup(response.text, 'html.parser') # parse html

# Because the format of table "球隊對戰戰績" differs from all the others, we handle this seperately.

# find table "球隊對戰戰績" by locating its upper layer first
caption_div = soup.find('div',{'class': 'record_table_caption'}, string="球隊對戰戰績")

# then the table itself
table = caption_div.find_next('div', {'class': 'RecordTable'}).find('table')

# retrieve column names
headers = []

# table.find_all to extract all column names( th )
for th in table.find_all('th'):
    # the structure here differs from the rest of the element( 2 levels )
    if th.find('div', class_='rank'):
        headers.append('排名')
        headers.append('球隊')
    # deal with the second level
    else:
        header = th.get_text(strip=True)
        headers.append(header)

# table.find_all to extract all cell data( tr )
rows = []
for tr in table.find_all('tr')[1:]:  # skip the first row( column name )
    row = []
    # the structure here is 2-level, too
    sticky = tr.find('td', class_='sticky')
    if sticky:
        rank = sticky.find('div', class_='rank').get_text(strip=True)
        team_name = sticky.find('div', class_='team-w-trophy').get_text(strip=True)
        row.append(rank)
        row.append(team_name)
    # handle the rest of the table
    for td in tr.find_all('td')[1:]:  # skip the first row( column name )
        cell = td.get_text(strip=True)
        row.append(cell)
    rows.append(row)
df = pd.DataFrame(rows, columns=headers)
df.to_csv("".join(["球隊對戰戰績",'.csv']), index=False, encoding='utf-8-sig')
# "".join(["球隊對戰戰績",'.csv']) can be replaced by "球隊對戰戰績"+'.csv'

# the rest of the three tables share the same structure

target_list = ['團隊投球成績','團隊打擊成績','團隊守備成績']
target = '團隊投球成績'
for target in target_list:
    caption_div = soup.find('div',{'class': 'record_table_caption'}, string=target)
    table = caption_div.find_next('div', {'class': 'RecordTable'}).find('table')
    headers = []
    for th in table.find_all('th'):
        header = th.get_text(strip=True)
        headers.append(header)

    rows = []
    for tr in table.find_all('tr')[1:]:
        row = []
        sticky = tr.find('td', class_='sticky')
        # slightly different here, the teamname is stored in a "href" hyperlink
        if sticky:
            team_name = sticky.find('a').get_text(strip=True)
            row.append(team_name)
        for td in tr.find_all('td')[1:]:
            cell = td.get_text(strip=True)
            row.append(cell)
        rows.append(row)
    df = pd.DataFrame(rows, columns=headers)
    df.to_csv("".join([target,'.csv']), index=False, encoding='utf-8-sig')
```

使用 Bradley-Terry model 分析

```r
winlose <- data.table::fread("球隊對戰戰績.csv")
pitch <- data.table::fread("團隊投球成績.csv")
combat <- data.table::fread("團隊打擊成績.csv")
defense <- data.table::fread("團隊守備成績.csv")
```

資料前處理

處理-球隊對戰戰績

處理-團隊投球成績

處理-團隊打擊成績

處理-團隊守備成績