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。

python beautifulsoup4 程式碼

```
# 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 = []
for th in table.find_all('th'):   #extract all column names( 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')
```

使用 Bradley-Terry model 分析

資料前處理

```
winlose <- data.table::fread("球隊對戰戰績.csv")
library(dplyr)
library(tidyr)
library(BradleyTerry2)
teams <- winlose$球隊
matches <- winlose[,8:13]

long_format <- matches %>%
  mutate(球隊 = teams) %>%
  pivot_longer(cols = -球隊, names_to = "對戰球隊", values_to = "戰績") %>%
  drop_na()

results <- long_format %>%
  separate(戰績, into = c("勝", "和", "敗"), sep = "-", convert = TRUE)

win_matrix <- results %>%
  mutate(勝隊 = 球隊, 敗隊 = 對戰球隊, 勝數 = 勝) %>%
  select(勝隊, 敗隊, 勝數) %>%
  pivot_wider(names_from = 敗隊, values_from = 勝數, values_fill = 0)

win_matrix <- as.data.frame(win_matrix)
rownames(win_matrix) <- win_matrix$勝隊
win_matrix <- win_matrix[,-which(names(win_matrix)=='勝隊')]

tie_matrix <- results %>%
  mutate(隊伍1 = 球隊, 隊伍2 = 對戰球隊, 和數 = 和) %>%
  select(隊伍1, 隊伍2, 和數) %>%
  pivot_wider(names_from = 隊伍2, values_from = 和數, values_fill = 0)
tie_matrix <- as.data.frame(tie_matrix)
rownames(tie_matrix) <- tie_matrix$隊伍1
tie_matrix <- tie_matrix[,-which(names(tie_matrix) == '隊伍1')]
```

為了針對和局的出現，我們除了一般勝負的6x6矩陣，還額外做出了一個6x6的和局矩陣以此來分析出現和局的狀況。

Bradley-Terry model

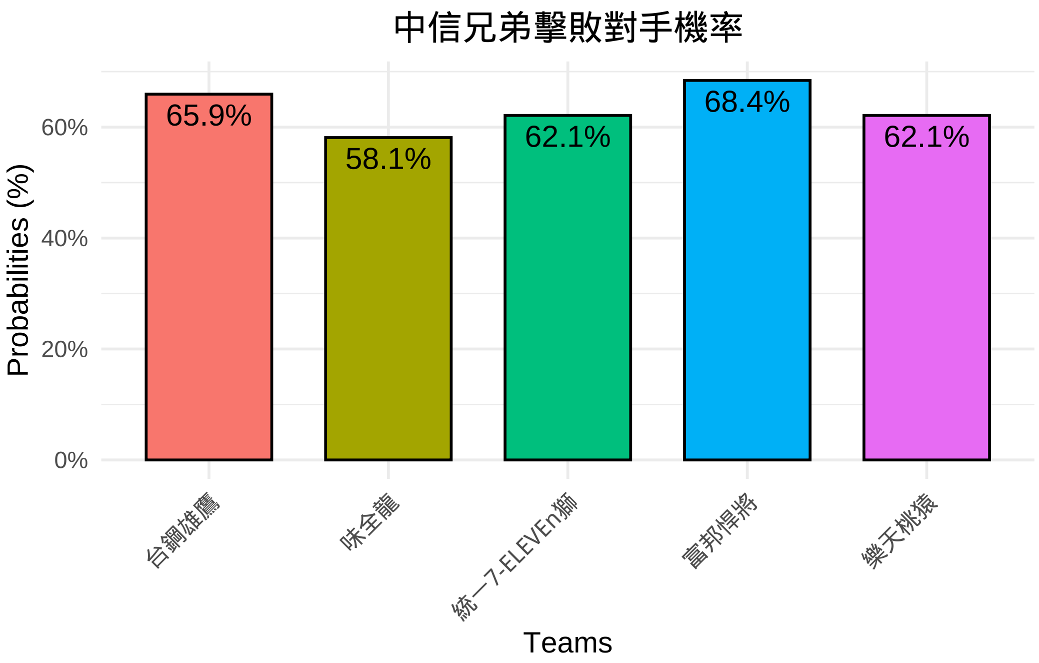
```
library(VGAM)
#fit <- vglm(Brat(as.matrix(win_matrix)) ~1, brat(refgp = 1), trace = FALSE, crit = "coef")
fit_ties <- vglm(Brat(as.matrix(win_matrix), as.matrix(tie_matrix)) ~1, bratt(refgp = 1,refvalue = 1), tra

summary(fit_ties)
```

```
Call:
vglm(formula = Brat(as.matrix(win_matrix), as.matrix(tie_matrix)) ~
    1, family = bratt(refgp = 1, refvalue = 1), trace = FALSE,
    crit = "coef")

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1  -0.3470     0.3419  -1.015   0.3101
(Intercept):2  -0.5170     0.3428  -1.508   0.1314
(Intercept):3  -0.5170     0.3428  -1.508   0.1314
(Intercept):4  -0.6880     0.3451  -1.993   0.0462 *
(Intercept):5  -0.8037     0.3476  -2.312   0.0208 *
(Intercept):6  -4.2693     0.7454  -5.728 1.02e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of linear predictors:  6

Names of linear predictors: loglink(alpha2), loglink(alpha3), loglink(alpha4),
loglink(alpha5), loglink(alpha6), loglink(alpha0)

Log-likelihood: -130.9393 on 0 degrees of freedom

Number of Fisher scoring iterations: 6

Warning: Hauck-Donner effect detected in the following estimate(s):
'(Intercept):6'
```

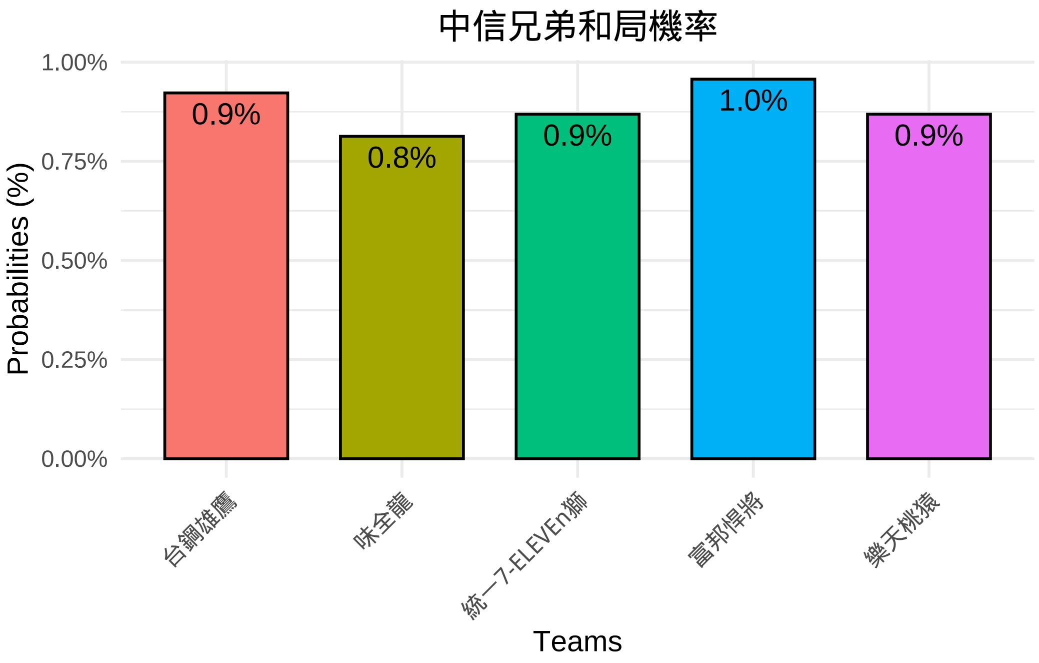
在 Brat 這個 function 當中，他將各隊的對戰情形表示了出來，例如中信兄弟贏統一獅 8 局，那在 Brat 中就會表示為’中信兄弟 > 統一獅’ 為 8，而根據我們 fit 出的模型可以給出以下的解釋。首先分為 intercept1$_{\text{intercept6，其中的 intercept1}}$intercept5 是指 log(αj),j=1…5，並且由 1 到 5 分別代表的球隊為味全龍，統一獅，樂天桃園，台鋼雄鷹以及富邦悍將，至於中信兄弟為 baseline，而最後的 intercept6 是指 log(α0)，他代表的則是平局。在此模型當中，i 打贏 j 的機率為 αi/(αi + αj + α0)，i 與 j 和局的機率為 α0/(αi +αj +α0)。舉例來說，中信兄弟為 baseline，令他為 αi=1(也就是 refvalue=1)，則中信兄弟擊敗味全龍的機率為 1/(1+exp(-0.347)+exp(-4.2693))=0.581，並且和局的機率為 exp(-4.2693)/(1+exp(-0.347)+exp(-4.2693))=0.008。

```
library(ggplot2)
showtext::showtext_auto()
teams <- c(" 味全龍", " 統一 7-ELEVEn 獅", " 樂天桃猿", " 台鋼雄鷹", " 富邦悍將")
loglink_values <- c(-0.3469979, -0.5170282, -0.5170282, -0.6879753, -0.8036663, -4.269289)
names(loglink_values) <- c("alpha2", "alpha3", "alpha4", "alpha5", "alpha6", "alpha0")

alpha0 <- loglink_values["alpha0"]

probabilities <- sapply(loglink_values[1:5], function(loglink_alpha) {
  1 / (1 + exp(loglink_alpha) + exp(alpha0))
})
probabilitiestie<-sapply(loglink_values[1:5], function(loglink_alpha) {
  exp(alpha0) / (1 + exp(loglink_alpha) + exp(alpha0))
})
```

```r
beats <- data.frame(
  Team = teams,
  Probability = probabilities
)

tiess <- data.frame(
  Team = teams,
  Probability = probabilitiestie
)

ggplot(beats, aes(x = Team, y = Probability, fill = Team)) +
  geom_bar(stat = "identity", color = "black", width = 0.7) +
  geom_text(aes(label = scales::percent(Probability, accuracy = 0.1)),
            vjust = 1.5, size = 4, color = "black") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(
    title = "中信兄弟擊敗對手機率",
    x = "Teams",
    y = "Probabilities (%)"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```



```r
ggplot(tiess, aes(x = Team, y = Probability, fill = Team)) +
  geom_bar(stat = "identity", color = "black", width = 0.7) +
  geom_text(aes(label = scales::percent(Probability, accuracy = 0.1)),
            vjust = 1.5, size = 4, color = "black") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(
    title = "中信兄弟和局機率",
    x = "Teams",
    y = "Probabilities (%)"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```



由最後兩張圖片可以很直接地看出中信兄弟贏味全龍的機率為0.581，而贏富邦悍將的機率為0.684，也代表著他在面對其他5隊的時候勝算是相當高的。其中贏味全龍的機率相較於其他隊伍是低的，我認為這是由於味全龍的戰績是第二名，並且中信兄弟在隊上他們僅僅拿到7場勝利，而在面對統一獅、台鋼雄鷹和富邦悍將時各拿下8場勝利。在和局的方面，可以發現中信兄弟面對不同隊伍的和局機率都是偏低的，這可能是由於此筆資料中他未曾拿下平局。