臺灣 P. LEAGUE+ 職籃聯賽資料分析 (Kent)

此專案旨在從臺灣 P. LEAGUE+ 職籃聯賽官網抓取每個球隊的比賽數據，並分析各球隊的主場和客場勝率，最後在地圖上展示各球隊的主場資訊及客場對戰勝率。

引入必要的套件

首先引入必要的套件：requests 用於發送 HTTP 請求，BeautifulSoup 用於解析 HTML，UserAgent 用於生成隨機的用戶代理，pandas 用於數據處理。

In [1]:
```
import requests
from bs4 import BeautifulSoup
from fake_useragent import UserAgent
import pandas as pd
```

設定隨機用戶代理

生成一個隨機的用戶代理來模擬瀏覽器，防止被網站封鎖。

In [3]:
```
ua = UserAgent()
headers = {'User-Agent': ua.random}
```

抓取球隊比賽數據

透過迴圈抓取各球隊的比賽數據，並解析 HTML 內容以提取表格數據，存儲在 team_data 字典中。

In [7]:
```
# Create an empty dictionary to store the table data for each team
team_data = {}

# Loop to get the table data for teams 1 to 6
for team_number in range(1, 7):
    # Construct the URL
    url = f'https://pleagueofficial.com/team/{team_number}#team_record'
    # Get the webpage content
    response = requests.get(url, headers=headers)
    html_content = response.content
    # Parse the HTML content
    soup = BeautifulSoup(html_content, 'html.parser', from_encoding='utf-8')
    # Find all tables on the page
    tables = soup.find_all('table')
    # Read the table into a pandas DataFrame
    team_df = pd.read_html(str(tables), encoding='utf-8')[1]  # Select the first table, may need adjustment based on w
    # Store the table data of each team in the team_data dictionary
    team_data[f'Team_{team_number}'] = team_df
```

In [9]:
```
team_df.head(5)
```

Out[9]:

	比賽日期	對手	分數	二分	二分%	三分	三分%	罰球	罰球%	得分	籃板	助攻	失誤	抄截	阻攻	犯規
0	2023-11-11	臺北富邦勇士	96W - 89	21 - 48	43.75	14 - 42	33.33	12 - 15	80.00	96	51	25	20	16	6	13
1	2023-11-18	新竹御頂攻城獅	87 - 111W	35 - 56	62.50	8 - 38	21.05	17 - 26	65.38	111	48	26	12	14	2	19
2	2023-11-19	桃園璞園領航猿	113 - 123W	28 - 45	62.22	16 - 43	37.21	19 - 24	79.17	123	48	32	13	12	2	21
3	2023-11-25	桃園璞園領航猿	92 - 98W	28 - 49	57.14	9 - 28	32.14	15 - 19	78.95	98	50	16	20	5	4	22
4	2023-11-26	高雄17直播鋼鐵人	93 - 109W	26 - 50	52.00	16 - 37	43.24	9 - 10	90.00	109	50	26	15	7	2	17

資料處理

將每個球隊的數據存儲在單獨的 DataFrame 中，並組織成一個包含各球隊資訊的列表 teamlist，並且把不需要的欄位給移除。

In [11]:
```
# Store each team's data in separate DataFrames
fubonBrave_df = team_data['Team_1']
taoyuanPilot_df = team_data['Team_2']
hsinchuLioneers_df = team_data['Team_3']
formosaDreamers_df = team_data['Team_4']
kaosiungSteelers_df = team_data['Team_5']
newtaipeiKings_df = team_data['Team_6']
```

```
teamlist = [
    {'隊伍': '臺北富邦勇士', '主客場_df': fubonBrave_df},
    {'隊伍': '桃園璞園領航猿', '主客場_df': taoyuanPilot_df},
    {'隊伍': '新竹御頂攻城獅', '主客場_df': hsinchuLioneers_df},
    {'隊伍': '福爾摩沙夢想家', '主客場_df': formosaDreamers_df},
    {'隊伍': '高雄17直播鋼鐵人', '主客場_df': kaosiungSteelers_df},
    {'隊伍': '新北國王', '主客場_df': newtaipeiKings_df}
]

fubonBrave_df.head(5)
```

Out[11]:

	比賽日期	對手	分數	二分	二分%	三分	三分%	罰球	罰球%	得分	籃板	助攻	失誤	抄截	阻攻	犯規
0	2023-11-11	新北國王	96 - 89L	26 - 54	48.15	10 - 24	41.67	7 - 10	70.00	89	48	17	26	10	4	17
1	2023-11-12	高雄17直播鋼鐵人	94 - 115W	32 - 59	54.24	10 - 30	33.33	21 - 24	87.50	115	48	30	9	11	6	22
2	2023-11-19	福爾摩沙夢想家	86L - 89	21 - 48	43.75	11 - 33	33.33	11 - 17	64.71	86	47	22	16	10	1	20
3	2023-11-26	新竹御頂攻城獅	99W - 84	27 - 48	56.25	11 - 30	36.67	12 - 16	75.00	99	49	26	19	7	5	16
4	2023-12-02	福爾摩沙夢想家	76L - 90	20 - 48	41.67	9 - 38	23.68	9 - 14	64.29	76	48	15	11	5	2	20

In [13]:
```
# Preparing the data for a specific analysis where these statistics are not needed
columns_to_drop = ['二分%', '二分', '三分', '三分%', '罰球', '罰球%', '得分', '籃板', '助攻', '失誤', '抄截', '阻攻', '犯規']
```

定義輔助函數

定義輔助函數以計算比賽結果和勝率，包括 classify_score 用於分類比賽結果，calculate_home_win_rate 和 calculate_away_win_rate 用於計算主場和客場勝率。

In [15]:
```
def classify_score(score):
    # Classify the game result based on the score string
    if score == ' - L':
        return '主場輸' # If the score is ' - L', it indicates a home loss
    elif score == ' - W':
        return '主場贏' # If the score is ' - W', it indicates a home win
    elif score == 'L - ':
        return '客場輸' # If the score is 'L - ', it indicates an away loss
    elif score == 'W - ':
        return '客場贏' # If the score is 'W - ', it indicates an away win
    else:
        return '未知' # For other cases, return 'Unknown'

def calculate_home_win_rate(df):
    # Filter out home games
    home_games = df[df['比賽結果'].isin(['主場贏', '主場輸'])]
    # Count the number of home wins
    home_wins = home_games[home_games['比賽結果'] == '主場贏'].shape[0]
    # Count the total number of home games
    total_home_games = home_games.shape[0]
    # If there are no home games, return 0
    if total_home_games == 0:
        return 0
    # Calculate the home win rate
    return home_wins / total_home_games

def calculate_away_win_rate(df):
    # Filter out away games
    away_games = df[df['比賽結果'].isin(['客場贏', '客場輸'])]
    # Count the number of away wins
    away_wins = away_games[away_games['比賽結果'] == '客場贏'].shape[0]
    # Count the total number of away games
    total_away_games = away_games.shape[0]
    # If there are no away games, return 0
    if total_away_games == 0:
        return 0
    # Calculate the away win rate
    return away_wins / total_away_games
```

分析並且print出勝率

分析各球隊的數據，計算並打印主場和客場勝率，將結果存儲在 data 字典中。

In [17]:
```
# Initialize an empty dictionary to store the data
data = {'球隊名稱': [], '主場總勝率': [], '客場總勝率': []}
```

```
# Iterate through each team in the team list
for team in teamlist:
    df = team['主客場_df']
    df['比賽日期'] = pd.to_datetime(df['比賽日期'], format='%Y-%m-%d', errors='coerce')
    df.drop(columns_to_drop, axis=1, inplace=True)
    df[['客場', '主場']] = df['分數'].str.extract(r'(\d+)\D+(\d+)')
    df['分數'] = df['分數'].str.replace(r'\d+', '', regex=True)
    df['比賽結果'] = df['分數'].str.replace(r'\d+', '').apply(classify_score)
    home_win_rate = calculate_home_win_rate(df)
    print(f"{team['隊伍']}", f"主場總勝率: {home_win_rate:.2%}")
    away_win_rate = calculate_away_win_rate(df)
    print(f"{team['隊伍']}", f"客場總勝率: {away_win_rate:.2%}")
    # Append the results to the data dictionary
    data['球隊名稱'].append(team['隊伍'])
    data['主場總勝率'].append(home_win_rate)
    data['客場總勝率'].append(away_win_rate)
```

```
臺北富邦勇士 主場總勝率: 55.00%
臺北富邦勇士 客場總勝率: 35.00%
桃園璞園領航猿 主場總勝率: 75.00%
桃園璞園領航猿 客場總勝率: 55.00%
新竹御頂攻城獅 主場總勝率: 60.00%
新竹御頂攻城獅 客場總勝率: 45.00%
福爾摩沙夢想家 主場總勝率: 75.00%
福爾摩沙夢想家 客場總勝率: 45.00%
高雄17直播鋼鐵人 主場總勝率: 25.00%
高雄17直播鋼鐵人 客場總勝率: 20.00%
新北國王 主場總勝率: 50.00%
新北國王 客場總勝率: 60.00%
```

In [19]:
```
fubonBrave_df.head(5) # After removing redundant columns
```

Out[19]:

	比賽日期	對手	分數	客場	主場	比賽結果
0	2023-11-11	新北國王	- L	96	89	主場輸
1	2023-11-12	高雄17直播鋼鐵人	- W	94	115	主場贏
2	2023-11-19	福爾摩沙夢想家	L -	86	89	客場輸
3	2023-11-26	新竹御頂攻城獅	W -	99	84	客場贏
4	2023-12-02	福爾摩沙夢想家	L -	76	90	客場輸

進一步分析客場對戰數據

分析各球隊的客場對戰數據，並存儲在相應的 DataFrame 中。

In [21]:
```
for i in range(len(teamlist)):
    home_team = teamlist[i]['隊伍']
    for j in range(len(teamlist)):
        if i != j:  # Ensure it's not the same team
            away_team = teamlist[j]['隊伍']
            # Generate filter condition
            filter_condition = ((teamlist[i]['主客場_df']['比賽結果'] == '客場贏') |
                                (teamlist[i]['主客場_df']['比賽結果'] == '客場輸') &
                                (teamlist[i]['主客場_df']['對手'] == away_team))
            # Filter the data based on the condition
            filtered_df = teamlist[i]['主客場_df'][filter_condition]
            # Store the filtered data in teamlist
            teamlist[i][f"{away_team}_客場"] = filtered_df
```

計算每個對手的客場勝率

計算每個球隊在客場對戰各對手的勝率，並將結果存儲在 data2 字典中。

In [23]:
```
team_names = ["臺北富邦勇士", "桃園璞園領航猿", "新竹御頂攻城獅", "福爾摩沙夢想家", "高雄17直播鋼鐵人", "新北國王"]

data2 = {}

for i, team_data in enumerate(teamlist):
    team_name = team_names[i]
    away_df = team_data['主客場_df'][team_data['主客場_df']['比賽結果'].str.contains('客場')]

    # Calculate the away win rate for each opponent
    grouped = away_df.groupby('對手').agg(
        total_games=('比賽結果', 'size'),
        wins=('比賽結果', lambda x: (x == '客場贏').sum())
    )
```

```
    grouped['win_rate'] = grouped['wins'] / grouped['total_games']

    # Calculate the overall away win rate
    total_away_games = away_df.shape[0]
    total_away_wins = away_df[away_df['比賽結果'] == '客場贏'].shape[0]
    overall_away_win_rate = total_away_wins / total_away_games if total_away_games > 0 else 0

    # Print the away win rate for each opponent and the overall away win rate
    print(f"{team_name} 客場總勝率: {overall_away_win_rate:.2%}")
    for opponent, row in grouped.iterrows():
        print(f"  對手: {opponent}, 客場勝率: {row['win_rate']:.2%}")
    print()

    data2[team_name] = dict(grouped['win_rate'])
```

```
臺北富邦勇士 客場總勝率: 35.00%
  對手: 新北國王, 客場勝率: 75.00%
  對手: 新竹御頂攻城獅, 客場勝率: 25.00%
  對手: 桃園璞園領航猿, 客場勝率: 0.00%
  對手: 福爾摩沙夢想家, 客場勝率: 25.00%
  對手: 高雄17直播鋼鐵人, 客場勝率: 50.00%

桃園璞園領航猿 客場總勝率: 55.00%
  對手: 新北國王, 客場勝率: 25.00%
  對手: 新竹御頂攻城獅, 客場勝率: 75.00%
  對手: 福爾摩沙夢想家, 客場勝率: 25.00%
  對手: 臺北富邦勇士, 客場勝率: 50.00%
  對手: 高雄17直播鋼鐵人, 客場勝率: 100.00%

新竹御頂攻城獅 客場總勝率: 45.00%
  對手: 新北國王, 客場勝率: 75.00%
  對手: 桃園璞園領航猿, 客場勝率: 25.00%
  對手: 福爾摩沙夢想家, 客場勝率: 25.00%
  對手: 臺北富邦勇士, 客場勝率: 50.00%
  對手: 高雄17直播鋼鐵人, 客場勝率: 50.00%

福爾摩沙夢想家 客場總勝率: 45.00%
  對手: 新北國王, 客場勝率: 75.00%
  對手: 新竹御頂攻城獅, 客場勝率: 25.00%
  對手: 桃園璞園領航猿, 客場勝率: 25.00%
  對手: 臺北富邦勇士, 客場勝率: 25.00%
  對手: 高雄17直播鋼鐵人, 客場勝率: 75.00%

高雄17直播鋼鐵人 客場總勝率: 20.00%
  對手: 新北國王, 客場勝率: 0.00%
  對手: 新竹御頂攻城獅, 客場勝率: 0.00%
  對手: 桃園璞園領航猿, 客場勝率: 50.00%
  對手: 福爾摩沙夢想家, 客場勝率: 25.00%
  對手: 臺北富邦勇士, 客場勝率: 25.00%

新北國王 客場總勝率: 60.00%
  對手: 新竹御頂攻城獅, 客場勝率: 75.00%
  對手: 桃園璞園領航猿, 客場勝率: 25.00%
  對手: 福爾摩沙夢想家, 客場勝率: 25.00%
  對手: 臺北富邦勇士, 客場勝率: 75.00%
  對手: 高雄17直播鋼鐵人, 客場勝率: 100.00%
```

可視化球隊數據

在 teams_info 字典中存儲每個球隊的資訊，包括位置、球館、Logo 及勝率。

In [25]:

```
import folium
from folium.plugins import MarkerCluster
from ipywidgets import interact, widgets
from IPython.display import display
import pandas as pd
from folium.plugins import Fullscreen

# Team information
teams_info = {
    '臺北富邦勇士': {'location': (25.021677286559644, 121.54529485749593),
                'arena': '臺北和平籃球館',
                'logo1': 'https://upload.wikimedia.org/wikipedia/zh/7/77/%E8%87%BA%E5%8C%97%E5%AF%8C%E9%82%A6%E5%8B%87%
                'logo2': 'https://i.ytimg.com/vi/Tipu-zyqv9c/maxresdefault.jpg'
               },
    '桃園璞園領航猿': {'location': (24.995359949673343, 121.32290688472271),
                 'arena': '桃園市立綜合體育館',
                 'logo1': 'https://upload.wikimedia.org/wikipedia/zh/4/49/%E6%A1%83%E5%9C%92%E7%92%9E%E5%9C%92%E9%A0%98
                 'logo2':'https://pgw.udn.com.tw/gw/photo.php?u=https://uc.udn.com.tw/photo/2023/11/10/realtime/2704844
                },
```

```
    '新竹御頂攻城獅': {'location': (24.819996484469765, 121.01997695702596),
                'arena': '新竹縣體育館',
                'logo1': 'https://upload.wikimedia.org/wikipedia/zh/1/16/%E6%96%B0%E7%AB%B9%E5%BE%A1%E9%A0%82%E6%94%BE
                'logo2':'https://pgw.udn.com.tw/gw/photo.php?u=https://uc.udn.com.tw/photo/2023/12/20/realtime/2840883
               },
    '福爾摩沙夢想家': {'location': (24.067280946147996, 120.55274162909035),
                'arena': '彰化縣立體育館',
                'logo1': 'https://upload.wikimedia.org/wikipedia/zh/f/fd/%E7%A6%8F%E7%88%BE%E6%91%A9%E6%B2%99%E5%A4%A2
                'logo2':'https://shoplineimg.com/5a409e7f9a76f090d40002d6/655206fb55fc1dfb61bd3a80/2600x.webp?source_f
               },
    '高雄17直播鋼鐵人': {'location': (22.621188413947635, 120.3545428111289),
                  'arena': '高雄市鳳山體育館',
                  'logo1': 'https://upload.wikimedia.org/wikipedia/zh/6/6a/%E9%AB%98%E9%9B%8417%E7%9B%B4%E6%92%AD%E9%8
                  'logo2':'https://cdn2.ettoday.net/images/7292/7292105.jpg'
                 },
    '新北國王': {'location': (25.040589043245653, 121.45183729586284),
             'arena': '新北市立新莊體育館',
             'logo1': 'https://upload.wikimedia.org/wikipedia/zh/f/ff/%E6%96%B0%E5%8C%97%E5%9C%8B%E7%8E%8B.png',
             'logo2':'https://img.sportsv.net/img/article/cover/0/98850/fit-ii3mwKbfNH-945x495.png'
            }
}

win_rates_df = pd.DataFrame(data)

for team in teams_info.keys():
    row = win_rates_df[win_rates_df['球隊名稱'] == team]
    if not row.empty:
        teams_info[team]['home_win_rate'] = row['主場總勝率'].values[0]
        teams_info[team]['away_win_rate'] = row['客場總勝率'].values[0]
```

定義函數 show_team_info，根據選擇的球隊在地圖上顯示其主場及客場資訊。

In [27]:

```
# Create the map
m = folium.Map(location=[23.7, 121], zoom_start=7, scrollWheelZoom=False)

# Create MarkerCluster (to prevent icons from overlapping)
marker_cluster = MarkerCluster().add_to(m)

# Function to display selected team's info
def show_team_info(team):
    m_combined = folium.Map(location=[23.7, 121], zoom_start=7, scrollWheelZoom=False)
    marker_cluster_combined = MarkerCluster().add_to(m_combined)

    # Add home team info
    info = teams_info[team]
    location = info['location']
    popup_content = f"""
    <div style="font-size: 12pt;">
    <img src="{info['logo2']}" width="250"><br>
    <b>球隊:</b> {team}<br>
    <b>球館:</b> {info['arena']}<br>
    <b>主場總勝率:</b> {info['home_win_rate']:.2%}<br>
    <b>客場總勝率:</b> {info['away_win_rate']:.2%}
    """

    icon_html = '''
    <div style="
        background: url({url});
        background-color: white;
        width: 60px; height: 60px;
        background-size: cover;
        border: 4px solid red;
        border-radius: 100%;">
    </div>
    '''.format(url=info['logo1'])

    # Create custom icon
    icon = folium.DivIcon(html=icon_html)

    folium.Marker(location, popup=folium.Popup(popup_content, max_width=500), tooltip=team, icon=icon).add_to(marker_c

    # Add away team info
    for other_team, other_location in teams_info.items():
        if other_team != team:
            away_win_rate = data2[team].get(other_team, None)
            if away_win_rate is not None:
                other_popup_content = f"""
                <div style="font-size: 12pt;">
                <img src="{other_location['logo2']}" width="250"><br>
                <b>球隊:</b> {other_team}<br>
                <b>球館:</b> {other_location['arena']}<br>
```

```
                <b>作客此球場勝率:</b> {away_win_rate:.2%}
                </div>
                """
                other_icon_html = '''
                <div style="
                    background: url({url});
                    background-color: white;
                    width: 60px; height: 60px;
                    background-size: cover;
                    border: 2px solid black;
                    border-radius: 80%;">
                </div>
                '''.format(url=other_location['logo1'])
                other_icon = folium.DivIcon(html=other_icon_html)
                folium.Marker(other_location['location'], popup=folium.Popup(other_popup_content, max_width=500), tool
                              icon=other_icon).add_to(marker_cluster_combined)
    folium.plugins.MiniMap().add_to(m_combined)
    folium.plugins.Fullscreen().add_to(m_combined)
    m_combined.get_root().html.add_child(folium.Element("""
    <div style="position:fixed; top:10px; right:10px; z-index:1000; background:white; padding:6px; border:1px solid gr
        <button onclick="window.location.href=window.location.href">Reset</button>
    </div>
    """))
    display(m_combined)
```

創建一個下拉式選單，讓用戶可以選擇球隊並顯示相應資訊。

In [29]:
```
# Create the interactive widget
team_dropdown = widgets.Dropdown(
    options=list(teams_info.keys()),
    description='選擇球隊:'
)

interact(show_team_info, team=team_dropdown)
```

```
interactive(children=(Dropdown(description='選擇球隊:', options=('臺北富邦勇士', '桃園璞園領航猿', '新竹御頂攻城獅', '福爾摩沙夢想家', '高雄17直…
```

Out[29]:
```
<function __main__.show_team_info(team)>
```

以上程式碼完整實現了從抓取 職籃聯賽數據、分析數據到在地圖上可視化展示結果的過程。

最後，下一頁展示了地圖數據的視覺化結果。

Reference:

1. Python map visualization - using Folium
2. Have fun looking at geospatial data! (4) —Folium interactive map
3. Python Folium: Create Web Maps From Your Data
4. NBA Team Map
5. Jupyter interactive input (ipywidgets control)
6. MarkerCluster


選擇球隊:
✓ 臺北富邦勇士
桃園璞園領航猿
新竹御頂攻城獅
福爾摩沙夢想家
高雄17直播鋼鐵人
新北國王
Reset
球隊: 福爾摩沙夢想家
球館: 彰化縣立體育館
作客此球場勝率: 25.00%
Leaflet | © OpenStreetMap contributors
Reset
球隊: 桃園璞園領航猿
球館: 桃園市立綜合體育館
作客此球場勝率: 0.00%
Leaflet | © OpenStreetMap contributors