assignment4

May 4, 2021

1 Assignment 4

1.1 Description

In this assignment you must read in a file of metropolitan regions and associated sports teams from assets/wikipedia_data.html and answer some questions about each metropolitan region. Each of these regions may have one or more teams from the "Big 4": NFL (football, in assets/nfl.csv), MLB (baseball, in assets/mlb.csv), NBA (basketball, in assets/nba.csv or NHL (hockey, in assets/nhl.csv). Please keep in mind that all questions are from the perspective of the metropolitan region, and that this file is the "source of authority" for the location of a given sports team. Thus teams which are commonly known by a different area (e.g. "Oakland Raiders") need to be mapped into the metropolitan region given (e.g. San Francisco Bay Area). This will require some human data understanding outside of the data you've been given (e.g. you will have to hand-code some names, and might need to google to find out where teams are)!

For each sport I would like you to answer the question: **what is the win/loss ratio's correlation with the population of the city it is in?** Win/Loss ratio refers to the number of wins over the number of wins plus the number of losses. Remember that to calculate the correlation with pearsonr, so you are going to send in two ordered lists of values, the populations from the wikipedia_data.html file and the win/loss ratio for a given sport in the same order. Average the win/loss ratios for those cities which have multiple teams of a single sport. Each sport is worth an equal amount in this assignment (20%*4=80%) of the grade for this assignment. You should only use data **from year 2018** for your analysis – this is important!

1.2 Notes

1. Do not include data about the MLS or CFL in any of the work you are doing, we're only interested in the Big 4 in this assignment.
2. I highly suggest that you first tackle the four correlation questions in order, as they are all similar and worth the majority of grades for this assignment. This is by design!
3. It's fair game to talk with peers about high level strategy as well as the relationship between metropolitan areas and sports teams. However, do not post code solving aspects of the assignment (including such as dictionaries mapping areas to teams, or regexes which will clean up names).
4. There may be more teams than the assert statements test, remember to collapse multiple teams in one city into a single value!

1.3 Question 1

For this question, calculate the win/loss ratio's correlation with the population of the city it is in for the **NHL** using **2018** data.

[1]:

```
# Import labaries
import pandas as pd
import numpy as np
import scipy.stats as stats
import re
```

[2]:

```
# Loading the data
cities=pd.read_html("assets/wikipedia_data.html")[1]
cities=cities.iloc[:-1,[0,3,5,6,7,8]]
cities.rename(columns={'Population (2016 est.)[8]': 'Population'},inplace=True)

# Cleanup - removing notes on brackets
cities['NFL'] = cities['NFL'].str.replace(r"\[.*\]", "")
cities['MLB'] = cities['MLB'].str.replace(r"\[.*\]", "")
cities['NBA'] = cities['NBA'].str.replace(r"\[.*\]", "")
cities['NHL'] = cities['NHL'].str.replace(r"\[.*\]", "")
```

[3]:

```
'''
@author:  Steven Ponce
Date:     25 April 2021                                                    ␣
 ↪                                                                         ␣
 ↪

For each sport I would like you to answer the question: what is the win/loss␣
 ↪ratio's correlation with the population
of the city it is in?

Calculate the win/loss ratio's corr with the population of the city it is in␣
 ↪for the NHL using 2018 data.
'''

# Set Big 4 to NHL
B4='NHL'

def nhl_correlation():
    # YOUR CODE HERE
    # raise NotImplementedError()

    team = cities[B4].str.extract('([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)')
    team['Metropolitan area'] = cities['Metropolitan area']
    # pd.melt This function is useful to massage a DataFrame into a format where one or more columns are identifier
    # variables (id_vars), while all other columns, considered measured variables (value_vars), are unpivoted to
    # the row axis, leaving just two non-identifier columns, variable and value.
    team = pd.melt(team, id_vars=['Metropolitan area']).drop(columns=['variable']).replace("", np.nan).replace("",np.nan).dropna().reset_index().rename(columns={"value": "team"})
    team = pd.merge(team, cities, how='left', on='Metropolitan area').iloc[:, 1:4]

    team = team.astype({'Metropolitan area': str, 'team': str, 'Population': int})
    team['team'] = team['team'].str.replace('[\w.]*\ ', '')

    # loading NHL.csv
    temp_df = pd.read_csv("assets/" + str.lower(B4) + ".csv")
    temp_df = temp_df[temp_df['year'] == 2018]
    temp_df['team'] = temp_df['team'].str.replace(r'\*', "")
    temp_df = temp_df [['team', 'W', 'L']]

    dropList = []
    for j in range(temp_df.shape[0]):
        row = temp_df.iloc[j]
        if row['team'] == row['W'] and row['L'] == row['W']:
            dropList.append(j)
    temp_df = temp_df.drop(dropList)

    temp_df['team'] = temp_df['team'].str.replace('[\w.]* ', '')
    temp_df = temp_df.astype({'team': str, 'W': int, 'L': int})
    temp_df['W/L%'] = temp_df['W'] / (temp_df['W'] + temp_df['L'])

    merge = pd.merge(team, temp_df, 'outer', on='team')
    merge = merge.groupby('Metropolitan area').agg({'W/L%': np.nanmean, 'Population': np.nanmean})

    population_by_region = merge['Population']
    win_loss_by_region = merge['W/L%']

    assert len(population_by_region) == len(win_loss_by_region), "Q1: Your lists must be the same length"
    assert len(population_by_region) == 28, "Q1: There should be 28 teams being analysed for NHL"

    return stats.pearsonr(population_by_region, win_loss_by_region)[0]
```

[4]:
```
nhl_correlation()
```

[4]:
```
0.012486162921209907
```

[ ]:

1.4 Question 2

For this question, calculate the win/loss ratio's correlation with the population of the city it is in for the **NBA** using **2018** data.

[5]:
```
'''
@author:  Steven Ponce
Date:     25 April 2021                                                       ␣
 ↪                                                                            ␣
 ↪

For each sport I would like you to answer the question: what is the win/loss␣
 ↪ratio's correlation with the population
of the city it is in?

Calculate the win/loss ratio's corr with the population of the city it is in␣
 ↪for the NBA using 2018 data.
'''

# Set Big 4 to NBA
B4='NBA'

def nba_correlation():
    # YOUR CODE HERE
    # raise NotImplementedError()

    team = cities[B4].str.extract('([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)')
    team['Metropolitan area'] = cities['Metropolitan area']

    team = pd.melt(team, id_vars=['Metropolitan area']).
 ↪drop(columns=['variable']).replace("", np.nan).replace("",np.nan).dropna().
 ↪reset_index().rename(columns={"value": "team"})
    team = pd.merge(team, cities, how='left', on='Metropolitan area').iloc[:, 1:
 ↪4]

    team = team.astype({'Metropolitan area': str, 'team': str, 'Population':␣
 ↪int})
    team['team'] = team['team'].str.replace('[\w.]*\ ', '')

    # loading NBA.csv
    temp_df = pd.read_csv("assets/" + str.lower(B4) + ".csv")
    temp_df = temp_df[temp_df['year'] == 2018]

    # Cleanup
    temp_df['team'] = temp_df['team'].str.replace(r'[\*]', "")
    temp_df['team'] = temp_df['team'].str.replace(r'\(\d*\)', "")
    temp_df['team'] = temp_df['team'].str.replace(r'[\xa0]', "")
    temp_df['team'] = temp_df['team'].str.replace('[\w.]* ', "")

    temp_df = temp_df [['team', 'W/L%']]
    temp_df = temp_df.astype({'team': str, 'W/L%': float})

    merge = pd.merge(team, temp_df, 'outer', on='team')
    merge = merge.groupby('Metropolitan area').agg({'W/L%': np.nanmean,␣
 ↪'Population': np.nanmean})

    population_by_region = merge['Population']
    win_loss_by_region = merge['W/L%']

    assert len(population_by_region) == len(win_loss_by_region), "Q2: Your␣
 ↪lists must be the same length"
    assert len(population_by_region) == 28, "Q2: There should be 28 teams being␣
 ↪analysed for NBA"

    return stats.pearsonr(population_by_region, win_loss_by_region)[0]
```

[6]:

```
nba_correlation()
```

[6]:

```
-0.17636350642182938
```

[ ]:

```

```

1.5 Question 3

For this question, calculate the win/loss ratio's correlation with the population of the city it is in for the **MLB** using **2018** data.

[7]:

```
'''
@author:  Steven Ponce
Date:     25 April 2021                                                       ␣
 ↪                                                                            ␣
 ↪

For each sport I would like you to answer the question: what is the win/loss␣
 ↪ratio's correlation with the population
of the city it is in?
```

```
Calculate the win/loss ratio's corr with the population of the city it is in␣
 ↪for the MLB using 2018 data.
'''

# Set Big 4 to MLB
B4='MLB'

def mlb_correlation():
    # YOUR CODE HERE
    # raise NotImplementedError()

    team = cities[B4].str.extract('([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)')
    team['Metropolitan area'] = cities['Metropolitan area']

    team = pd.melt(team, id_vars=['Metropolitan area']).
 ↪drop(columns=['variable']).replace("", np.nan).replace("",np.nan).dropna().
 ↪reset_index().rename(columns={"value": "team"})
    team = pd.merge(team, cities, how='left', on='Metropolitan area').iloc[:, 1:
 ↪4]

    team = team.astype({'Metropolitan area': str, 'team': str, 'Population':␣
 ↪int})
    team['team'] = team['team'].str.replace('\ Sox', 'Sox')
    team['team'] = team['team'].str.replace('[\w.]*\ ', '')

    # loading MLB.csv
    temp_df = pd.read_csv("assets/" + str.lower(B4) + ".csv")
    temp_df = temp_df[temp_df['year'] == 2018]

    # Cleanup
    temp_df['team'] = temp_df['team'].str.replace(r'[\*]', "")
    temp_df['team'] = temp_df['team'].str.replace(r'\(\d*\)', "")
    temp_df['team'] = temp_df['team'].str.replace(r'[\xa0]', "")
    temp_df = temp_df[['team', 'W-L%']]

    temp_df.rename(columns={"W-L%": "W/L%"}, inplace=True)
    temp_df['team'] = temp_df['team'].str.replace('\ Sox', 'Sox')
    temp_df['team'] = temp_df['team'].str.replace('[\w.]* ', '')
    temp_df = temp_df.astype({'team': str, 'W/L%': float})

    merge = pd.merge(team, temp_df, 'outer', on='team')
    merge = merge.groupby('Metropolitan area').agg({'W/L%': np.nanmean,␣
 ↪'Population': np.nanmean})
```

```
    population_by_region = merge['Population']
    win_loss_by_region = merge['W/L%']

    assert len(population_by_region) == len(win_loss_by_region), "Q3: Your␣
 ↪lists must be the same length"
    assert len(population_by_region) == 26, "Q3: There should be 26 teams being␣
 ↪analysed for MLB"

    return stats.pearsonr(population_by_region, win_loss_by_region)[0]
```

[8]:
```
mlb_correlation()
```

[8]:
```
0.15003737475409495
```

[ ]:
```

```

1.6 Question 4

For this question, calculate the win/loss ratio's correlation with the population of the city it is in for the **NFL** using **2018** data.

[9]:
```
'''
@author:  Steven Ponce
Date:     25 April 2021                                                      ␣
 ↪                                                                          ␣
 ↪

For each sport I would like you to answer the question: what is the win/loss␣
 ↪ratio's correlation with the population
of the city it is in?

Calculate the win/loss ratio's corr with the population of the city it is in␣
 ↪for the MLB using 2018 data.
'''

# Set Big 4 to NFL
B4='NFL'

def nfl_correlation():
    # YOUR CODE HERE
    # raise NotImplementedError()

    team = cities[B4].str.extract('([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)')
    team['Metropolitan area'] = cities['Metropolitan area']
    team = pd.melt(team, id_vars=['Metropolitan area']).
↪drop(columns=['variable']).replace("", np.nan).replace("",np.nan).dropna().
↪reset_index().rename(columns={"value": "team"})
    team = pd.merge(team, cities, how='left', on='Metropolitan area').iloc[:, 1:
↪4]

    team = team.astype({'Metropolitan area': str, 'team': str, 'Population':␣
↪int})
    team['team'] = team['team'].str.replace('[\w.]*\ ', '')

    # loading NFL.csv
    temp_df = pd.read_csv("assets/" + str.lower(B4) + ".csv")
    temp_df = temp_df[temp_df['year'] == 2018]

    # Cleanup
    temp_df['team'] = temp_df['team'].str.replace(r'[\*]', "")
    temp_df['team'] = temp_df['team'].str.replace(r'\(\d*\)', "")
    temp_df['team'] = temp_df['team'].str.replace(r'[\xa0]', "")

    temp_df = temp_df [['team', 'W-L%']]
    temp_df.rename(columns={"W-L%": "W/L%"}, inplace=True)

    dropList = []
    for j in range(temp_df.shape[0]):
        row = temp_df.iloc[j]
        if row['team'] == row['W/L%'] :
            dropList.append(j)
    temp_df = temp_df.drop(dropList)

    temp_df['team'] = temp_df['team'].str.replace('[\w.]* ', '')
    temp_df['team'] = temp_df['team'].str.replace('+', '')
    temp_df = temp_df.astype({'team': str, 'W/L%': float})

    merge = pd.merge(team, temp_df, 'outer', on='team')
    merge = merge.groupby('Metropolitan area').agg({'W/L%': np.nanmean,␣
↪'Population': np.nanmean})

    population_by_region = merge['Population']
    win_loss_by_region = merge['W/L%']

    assert len(population_by_region) == len(win_loss_by_region), "Q4: Your␣
↪lists must be the same length"
    assert len(population_by_region) == 29, "Q4: There should be 29 teams being␣
↪analysed for NFL"
```

```
    return stats.pearsonr(population_by_region, win_loss_by_region)[0]
```

[10]:
```
nfl_correlation()
```

[10]:
```
0.004282141436393017
```

[ ]:
```

```

1.7 Question 5

In this question I would like you to explore the hypothesis that **given that an area has two sports teams in different sports, those teams will perform the same within their respective sports**. How I would like to see this explored is with a series of paired t-tests (so use `ttest_rel`) between all pairs of sports. Are there any sports where we can reject the null hypothesis? Again, average values where a sport has multiple teams in one region. Remember, you will only be including, for each sport, cities which have teams engaged in that sport, drop others as appropriate. This question is worth 20% of the grade for this assignment.

[11]:
```
'''
@author:  Steven Ponce
Date:     26 April 2021                                                        ␣
 ↪                                                                             ␣
 ↪
'''


def nhl_win_loss():

    # Set Big 4 to NHL
    B4='NHL'
    team = cities[B4].str.extract('([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)')
    team['Metropolitan area'] = cities['Metropolitan area']

    team = pd.melt(team, id_vars=['Metropolitan area']).
 ↪drop(columns=['variable']).replace("", np.nan).replace("",np.nan).dropna().
 ↪reset_index().rename(columns={"value": "team"})
    team = pd.merge(team, cities, how='left', on='Metropolitan area').iloc[:, 1:
 ↪4]

    team = team.astype({'Metropolitan area': str, 'team': str, 'Population':␣
 ↪int})
    team['team'] = team['team'].str.replace('[\w.]*\ ', '')

    # loading NHL.csv
    temp_df = pd.read_csv("assets/" + str.lower(B4) + ".csv")
    temp_df = temp_df[temp_df['year'] == 2018]
    temp_df['team'] = temp_df['team'].str.replace(r'\*', "")
    temp_df = temp_df [['team', 'W', 'L']]

    dropList = []
    for j in range(temp_df.shape[0]):
        row = temp_df.iloc[j]
        if row['team'] == row['W'] and row['L'] == row['W']:
            dropList.append(j)
    temp_df = temp_df.drop(dropList)

    temp_df['team'] = temp_df['team'].str.replace('[\w.]* ', '')
    temp_df = temp_df.astype({'team': str, 'W': int, 'L': int})
    temp_df['W/L%'] = temp_df['W'] / (temp_df['W'] + temp_df['L'])

    merge = pd.merge(team, temp_df, 'inner', on='team')
    merge = merge.groupby('Metropolitan area').agg({'W/L%': np.nanmean,␣
 ↪'Population': np.nanmean})

    return merge[['W/L%']]

def nba_win_loss():

    # Set Big 4 to NBA
    B4='NBA'

    team = cities[B4].str.extract('([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)')
    team['Metropolitan area'] = cities['Metropolitan area']

    team = pd.melt(team, id_vars=['Metropolitan area']).
 ↪drop(columns=['variable']).replace("", np.nan).replace("",np.nan).dropna().
 ↪reset_index().rename(columns={"value": "team"})
    team = pd.merge(team, cities, how='left', on='Metropolitan area').iloc[:, 1:
 ↪4]

    team = team.astype({'Metropolitan area': str, 'team': str, 'Population':␣
 ↪int})
    team['team'] = team['team'].str.replace('[\w.]*\ ', '')

    # loading NBA.csv
    temp_df = pd.read_csv("assets/" + str.lower(B4) + ".csv")
    temp_df = temp_df[temp_df['year'] == 2018]

    # Cleanup
    temp_df['team'] = temp_df['team'].str.replace(r'[\*]', "")
    temp_df['team'] = temp_df['team'].str.replace(r'\(\d*\)', "")
    temp_df['team'] = temp_df['team'].str.replace(r'[\xa0]', "")
    temp_df['team'] = temp_df['team'].str.replace('[\w.]* ', "")

    temp_df = temp_df [['team', 'W/L%']]
    temp_df = temp_df.astype({'team': str, 'W/L%': float})

    merge = pd.merge(team, temp_df, 'outer', on='team')
    merge = merge.groupby('Metropolitan area').agg({'W/L%': np.nanmean,␣
 ↪'Population': np.nanmean})

    return merge[['W/L%']]


def mlb_win_loss():

    # Set Big 4 to MLB
    B4='MLB'

    team = cities[B4].str.extract('([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)')
    team['Metropolitan area'] = cities['Metropolitan area']

    team = pd.melt(team, id_vars=['Metropolitan area']).
 ↪drop(columns=['variable']).replace("", np.nan).replace("",np.nan).dropna().
 ↪reset_index().rename(columns={"value": "team"})
    team = pd.merge(team, cities, how='left', on='Metropolitan area').iloc[:, 1:
 ↪4]

    team = team.astype({'Metropolitan area': str, 'team': str, 'Population':␣
 ↪int})
    team['team']=team['team'].str.replace('\ Sox','Sox')
    team['team'] = team['team'].str.replace('[\w.]*\ ', '')

    # loading MLB.csv
    temp_df = pd.read_csv("assets/" + str.lower(B4) + ".csv")
    temp_df = temp_df[temp_df['year'] == 2018]

    # Cleanup
    temp_df['team'] = temp_df['team'].str.replace(r'[\*]', "")
    temp_df['team'] = temp_df['team'].str.replace(r'\(\d*\)', "")
    temp_df['team'] = temp_df['team'].str.replace(r'[\xa0]', "")

    temp_df = temp_df [['team', 'W-L%']]
    temp_df.rename(columns={"W-L%": "W/L%"}, inplace=True)
    temp_df['team']= temp_df['team'].str.replace('\ Sox','Sox')
    temp_df['team'] = temp_df['team'].str.replace('[\w.]* ','')
    temp_df = temp_df.astype({'team': str, 'W/L%': float})

    merge = pd.merge(team, temp_df, 'outer', on='team')
    merge = merge.groupby('Metropolitan area').agg({'W/L%': np.nanmean,␣
 ↪'Population': np.nanmean})

    return merge[['W/L%']]

def nfl_win_loss():

    # Set Big 4 to NFL
    B4='NFL'

    team = cities[B4].str.extract('([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)([A-Z]{0,2}[a-z0-9]*\␣
 ↪[A-Z]{0,2}[a-z0-9]*|[A-Z]{0,2}[a-z0-9]*)')
    team['Metropolitan area'] = cities['Metropolitan area']

    team = pd.melt(team, id_vars=['Metropolitan area']).
 ↪drop(columns=['variable']).replace("", np.nan).replace("",np.nan).dropna().
 ↪reset_index().rename(columns={"value": "team"})
    team = pd.merge(team, cities, how='left', on='Metropolitan area').iloc[:, 1:
 ↪4]

    team = team.astype({'Metropolitan area': str, 'team': str, 'Population':␣
 ↪int})
    team['team'] = team['team'].str.replace('[\w.]*\ ', '')

    # loading NFL.csv
    temp_df = pd.read_csv("assets/" + str.lower(B4) + ".csv")
    temp_df = temp_df[temp_df['year'] == 2018]

    # Cleanup
    temp_df['team'] = temp_df['team'].str.replace(r'[\*]', "")
    temp_df['team'] = temp_df['team'].str.replace(r'\(\d*\)', "")
    temp_df['team'] = temp_df['team'].str.replace(r'[\xa0]', "")

    temp_df = temp_df [['team', 'W-L%']]
    temp_df.rename(columns={"W-L%": "W/L%"}, inplace=True)

    dropList = []
    for j in range(temp_df.shape[0]):
        row = temp_df.iloc[j]
        if row['team'] == row['W/L%'] :
            dropList.append(j)
    temp_df = temp_df.drop(dropList)

    temp_df['team'] = temp_df['team'].str.replace('[\w.]* ', '')
    temp_df['team'] = temp_df['team'].str.replace('+', '')
    temp_df = temp_df.astype({'team': str, 'W/L%': float})

    merge = pd.merge(team, temp_df, 'outer', on='team')
    merge = merge.groupby('Metropolitan area').agg({'W/L%': np.nanmean,␣
 ↪'Population': np.nanmean})

    return merge[['W/L%']]

def dataframe(n):
    if n == 'NFL':
        return nfl_win_loss()
    elif n == 'NBA':
        return nba_win_loss()
    elif n == 'NHL':
        return nhl_win_loss()
    elif n == 'MLB':
        return mlb_win_loss()
    else:
        print("Something is wrong here")

def sports_team_performance():
    sports = ['NFL', 'NBA', 'NHL', 'MLB']
    p_values = pd.DataFrame({k: np.nan for k in sports}, index=sports)

    for a in sports:
        for b in sports:
            if a != b:
                merge = pd.merge(dataframe(a), dataframe(b), 'inner',␣
 ↪on=['Metropolitan area'])   #####
                p_values.loc[a, b] = stats.ttest_rel(merge['W/L%_x'], merge['W/
 ↪L%_y'])[1]

    assert abs(p_values.loc["NBA", "NHL"] - 0.02) <= 1e-2, "The NBA-NHL p-value␣
 ↪should be around 0.02"
    assert abs(p_values.loc["MLB", "NFL"] - 0.80) <= 1e-2, "The MLB-NFL p-value␣
 ↪should be around 0.80"
    return p_values
```

[12]:

```
sports_team_performance()
```

[12]:

```
          NFL       NBA       NHL       MLB
NFL       NaN  0.937509  0.030318  0.803459
NBA  0.937509       NaN  0.022386  0.949566
NHL  0.030318  0.022386       NaN  0.000703
```

```
MLB  0.803459  0.949566  0.000703       NaN
```

[ ]:

```

```