Assignment 4

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!

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!

As this assignment utilizes global variables in the skeleton code, to avoid having errors in your code you can either:

1. You can place all of your code within the function definitions for all of the questions (other than import statements).
2. You can create copies of all the global variables with the copy() method and proceed as usual.

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.

In [2]:
```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re

nhl_df=pd.read_csv("assets/nhl.csv")
cities=pd.read_html("assets/wikipedia_data.html")[1]
cities=cities.iloc[:-1,[0,3,5,6,7,8]]

def nhl_correlation():
    # YOUR CODE HERE
    # extract data from year 2018
    df13=nhl_df[nhl_df['year']>=2018]
    df13_index = df13.set_index('team')
    #remove rows without numeric values
    df13_index = df13_index.drop(['Atlantic Division','Metropolitan Division','Central Division','Pacific Division'])
    df13_index=df13_index.reset_index()
    NHL_cities = {'Tampa Bay Lightning*':'Tampa Bay Area',
             'Boston Bruins*':'Boston',
             'Toronto Maple Leafs*':'Toronto',
             'Florida Panthers':'Miami–Fort Lauderdale',
             'Detroit Red Wings':'Detroit',
             'Montreal Canadiens':'Montreal',
             'Ottawa Senators':'Ottawa',
             'Buffalo Sabres':'Buffalo',
             'Washington Capitals*':'Washington, D.C.',
             'Pittsburgh Penguins*':'Pittsburgh',
             'Philadelphia Flyers*':'Philadelphia',
             'Columbus Blue Jackets*':'Columbus',
             'New Jersey Devils*':'New York City',
             'Carolina Hurricanes':'Raleigh',
             'New York Islanders':'New York City',
             'New York Rangers':'New York City',
             'Nashville Predators*':'Nashville',
             'Winnipeg Jets*':'Winnipeg',
             'Minnesota Wild*':'Minneapolis–Saint Paul',
             'Colorado Avalanche*':'Denver',
             'St. Louis Blues':'St. Louis',
             'Dallas Stars':'Dallas–Fort Worth',
             'Chicago Blackhawks':'Chicago',
```

```
            'Vegas Golden Knights*':'Las Vegas',
            'Anaheim Ducks*':'Los Angeles',
            'San Jose Sharks*':'San Francisco Bay Area',
            'Los Angeles Kings*':'Los Angeles',
            'Calgary Flames':'Calgary',
            'Edmonton Oilers':'Edmonton',
            'Vancouver Canucks':'Vancouver',
            'Arizona Coyotes':'Phoenix'
            }
    df13_index['Metropolitan area']=df13_index['team'].map(NHL_cities)
    NHL_2018=df13_index[['Metropolitan area','W','L']]
    # convert to object to numeric values
    NHL_2018["W"] = pd.to_numeric(NHL_2018["W"])
    NHL_2018["L"] = pd.to_numeric(NHL_2018["L"])
    # group NHL teams by cities
    combined_teams=NHL_2018.groupby(['Metropolitan area'])['W','L'].sum()
    merge1= pd.merge(combined_teams,cities,on='Metropolitan area',how='inner')
    merge2=merge1[['Metropolitan area','W','L','Population (2016 est.)[8]']]
    merge2=merge2.rename(columns={"Population (2016 est.)[8]": "population"})
    merge2['win_loss_ratio']=merge2['W']/(merge2['W']+merge2['L'])
    merge2["population"] = pd.to_numeric(merge2["population"])
    population_by_region=merge2['population'].values.tolist()
    win_loss_by_region=merge2['win_loss_ratio'].values.tolist()
    #raise NotImplementedError()

    #population_by_region = []  pass in metropolitan area population from cities
    #win_loss_by_region = []  pass in win/loss ratio from nhl_df in the same order as cities["Metropolitan area"]

    #raise NotImplementedError()

    #population_by_region = [] # pass in metropolitan area population from cities
    #win_loss_by_region = [] # pass in win/loss ratio from nhl_df in the same order as cities["Metropolitan area"]

    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]
nhl_correlation()
```

Out[2]: 0.01230899645574425

In [ ]:

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.

In [4]:
```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re

nba_df=pd.read_csv("assets/nba.csv")
cities=pd.read_html("assets/wikipedia_data.html")[1]
cities=cities.iloc[:-1,[0,3,5,6,7,8]]
cities['population']=cities['Population (2016 est.)[8]']
cities['NBA']=cities['NBA'].str.replace(r'\[.*\]','')
nba_cities = cities[['Metropolitan area', 'NBA']].set_index('NBA')
nba_cities = nba_cities.drop(['—', ''], axis=0)

def get_area(team):
    for each in list(nba_cities.index.values):
        if team in each:
            return nba_cities.at[each, 'Metropolitan area']

def nba_correlation():
    # YOUR CODE HERE
    df_NBA2018=nba_df[nba_df['year']==2018]
    df_NBA2018["W"] = pd.to_numeric(df_NBA2018["W"])
    df_NBA2018["L"] = pd.to_numeric(df_NBA2018["L"])
    # remove the cities in the team names
    df_NBA2018['team']=df_NBA2018['team'].str.replace(r"\*", '')
    df_NBA2018['team']=df_NBA2018['team'].str.split().str[-2]
    # extract metro cities
    df_NBA2018['Metropolitan area'] = df_NBA2018['team']
    df_NBA2018['Metropolitan area'] = df_NBA2018['Metropolitan area'].apply(lambda x: get_area(x))
    combined_teams=df_NBA2018.groupby(['Metropolitan area'])['W','L'].sum()
    merge1= pd.merge(combined_teams,cities,on='Metropolitan area',how='inner')
    merge2=merge1[['Metropolitan area','W','L','population']]
    merge2['win_loss_ratio']=merge2['W']/(merge2['W']+merge2['L'])
    merge2["population"] = pd.to_numeric(merge2["population"])
    population_by_region=merge2['population'].values.tolist()
    win_loss_by_region=merge2['win_loss_ratio'].values.tolist()
    #raise NotImplementedError()

    #population_by_region = [] # pass in metropolitan area population from cities
    #win_loss_by_region = [] # pass in win/loss ratio from nba_df in the same order as cities["Metropolitan area"]

    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]
nba_correlation()
```

Out[4]: -0.17657160252844614

In [ ]:

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.

In [2]:
```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re

mlb_df=pd.read_csv("assets/mlb.csv")
cities=pd.read_html("assets/wikipedia_data.html")[1]
cities=cities.iloc[:-1,[0,3,5,6,7,8]]
cities['population']=cities['Population (2016 est.)[8]']
cities['MLB']=cities['MLB'].str.replace(r'\[.*\]','')
MLB_cities = cities[['Metropolitan area', 'MLB']].set_index('MLB')
MLB_cities = MLB_cities.drop(['—', ''], axis=0)
def get_area(team):
    for each in list(MLB_cities.index.values):
        if team in each:
            return MLB_cities.at[each, 'Metropolitan area']

def mlb_correlation():
    # YOUR CODE HERE
    df_mlb2018=mlb_df[mlb_df['year']==2018]
    df_mlb2018["W"] = pd.to_numeric(df_mlb2018["W"])
    df_mlb2018["L"] = pd.to_numeric(df_mlb2018["L"])
    # remove the cities in the team names
    df_mlb2018['team']=df_mlb2018['team'].str.split().str[-1]
    # extract metro cities
    df_mlb2018['Metropolitan area'] = df_mlb2018['team']
    df_mlb2018['Metropolitan area'] = df_mlb2018['Metropolitan area'].apply(lambda x: get_area(x))
    # two sox teams: red sox in Boston and white sox in Chicago
    df_mlb2018.at[0,'Metropolitan area']='Boston'
    combined_teams=df_mlb2018.groupby(['Metropolitan area'])['W','L'].sum()
    merge1= pd.merge(combined_teams,cities,on='Metropolitan area',how='inner')
    merge2=merge1[['Metropolitan area','W','L','population']]
```

```
    merge2['win_loss_ratio']=merge2['W']/(merge2['W']+merge2['L'])
    merge2["population"] = pd.to_numeric(merge2["population"])
    population_by_region=merge2['population'].values.tolist()
    win_loss_by_region=merge2['win_loss_ratio'].values.tolist()
    #population_by_region = [] # pass in metropolitan area population from cities
    #win_loss_by_region = [] # pass in win/loss ratio from mlb_df in the same order as cities["Metropolitan area"]

    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]
mlb_correlation()
```

Out[2]:
```
0.1505230448710485
```

In [ ]:
```

```

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.

In [4]:
```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re

nfl_df=pd.read_csv("assets/nfl.csv")
cities=pd.read_html("assets/wikipedia_data.html")[1]
cities=cities.iloc[:-1,[0,3,5,6,7,8]]
cities['population']=cities['Population (2016 est.)[8]']
cities['NFL']=cities['NFL'].str.replace(r'\[.*\]','')
NFL_cities = cities[['Metropolitan area', 'NFL']].set_index('NFL')
NFL_cities = NFL_cities.drop(['—', ''], axis=0)
def get_area(team):
    for each in list(NFL_cities.index.values):
        if team in each:
            return NFL_cities.at[each, 'Metropolitan area']

def nfl_correlation():
    # YOUR CODE HERE
    df_nfl2018=nfl_df[nfl_df['year']==2018]
    # remove rows that don't contain teams w/l
    df_nfl2018=df_nfl2018.query('W.str.isnumeric()')
    df_nfl2018["W"] = pd.to_numeric(df_nfl2018["W"])
```

```
    df_nfl2018["L"] = pd.to_numeric(df_nfl2018["L"])
    # remove the cities in the team names
    df_nfl2018['team']=df_nfl2018['team'].str.split().str[-1]
    #remove special characters in strings
    df_nfl2018['team']=df_nfl2018['team'].str.replace('\W', '', regex=True)
    # extract metro cities
    df_nfl2018['Metropolitan area'] = df_nfl2018['team']
    df_nfl2018['Metropolitan area'] = df_nfl2018['Metropolitan area'].apply(lambda x: get_area(x))
    combined_teams=df_nfl2018.groupby(['Metropolitan area'])['W','L'].sum()
    merge1= pd.merge(combined_teams,cities,on='Metropolitan area',how='inner')
    merge2=merge1[['Metropolitan area','W','L','population']]
    merge2['win_loss_ratio']=merge2['W']/(merge2['W']+merge2['L'])
    merge2["population"] = pd.to_numeric(merge2["population"])
    population_by_region=merge2['population'].values.tolist()
    win_loss_by_region=merge2['win_loss_ratio'].values.tolist()

    #population_by_region = [] # pass in metropolitan area population from cities
    #win_loss_by_region = [] # pass in win/loss ratio from nfl_df in the same order as cities["Metropolitan area"]

    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]
nfl_correlation()
```

Out[4]: 0.004922112149349409

In [ ]:

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.

In [ ]:
```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re
```

```
mlb_df=pd.read_csv("assets/mlb.csv")
nhl_df=pd.read_csv("assets/nhl.csv")
nba_df=pd.read_csv("assets/nba.csv")
nfl_df=pd.read_csv("assets/nfl.csv")
cities=pd.read_html("assets/wikipedia_data.html")[1]
cities=cities.iloc[:-1,[0,3,5,6,7,8]]

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

    # Note: p_values is a full dataframe, so df.loc["NFL","NBA"] should be the same as df.loc["NBA","NFL"] and
    # df.loc["NFL","NFL"] should return np.nan
    sports = ['NFL', 'NBA', 'NHL', 'MLB']
    p_values = pd.DataFrame({k:np.nan for k in sports}, index=sports)

    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
```

In [ ]: