Assignment4 (Score: 80.0 / 100.0)

1. Test cell (Score: 20.0 / 20.0)
2. Test cell (Score: 20.0 / 20.0)
3. Test cell (Score: 20.0 / 20.0)
4. Test cell (Score: 20.0 / 20.0)
5. Test cell (Score: 0.0 / 20.0)

Assignment 4

Description

In this assignment you must read in a file of metropolitan regions and associated sports teams from assets/wikipedia_data.html (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 (assets/nfl.csv)), MLB (baseball, in assets/mlb.csv (assets/mlb.csv)), NBA (basketball, in assets/nba.csv (assets/nba.csv) or NHL (hockey, in assets/nhl.csv (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` (https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html), 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!

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 [1]:

```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re
```

In [2]:

Student's answer (Top)

```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re

def clear_data(string1):
    if re.search(r'\[[a-z]* [0-9]+\]', string1) is None:
        return string1
    else:
        return string1.replace(re.search(r'\[[a-z]* [0-9]+\]', strin
g1).group(), '')


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


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]]
cities['NHL'] = cities['NHL'].apply(lambda x: clear_data(x))
nhl_cities = cities[['Metropolitan area', 'NHL']].set_index('NHL')
nhl_cities = nhl_cities.drop(['—', ''], axis=0)
nhl_df = nhl_df[nhl_df['year'] == 2018].drop([0, 9, 18, 26], axis=0)
# get only 2018 stats
population = cities[['Metropolitan area', 'Population (2016 est.)
[8]']]
population = population.set_index('Metropolitan area')
nhl_df['team'] = nhl_df['team'].apply(lambda x: x[:-1].strip() if x.
endswith("*") else x.strip())
nhl_df['area'] = nhl_df['team'].apply(lambda x: x.split(" ")[-1])
nhl_df['area'] = nhl_df['area'].apply(lambda x: get_area(x))
out = []
for group, frame in nhl_df.groupby('area'):
    total_wins = np.sum(pd.to_numeric(frame['W']))
    total_losses = np.sum(pd.to_numeric(frame['L']))
    total_matches = total_wins + total_losses
    ratio = total_wins / total_matches
    out_dict = {
        'Area': group,
        'Ratio': ratio
    }
    out.append(out_dict)
new_df = pd.DataFrame(out)
new_df = new_df.set_index('Area')
out_df = pd.merge(new_df, population, how="inner", left_index=True,
right_index=True)
out_df['Population (2016 est.)[8]'] = pd.to_numeric(out_df['Populati
on (2016 est.)[8]'])

def nhl_correlation():
    population_by_region = []  # pass in metropolitan area populatio
n from cities
```

```
    win_loss_by_region = []  # pass in win/loss ratio from nhl_df in the same order as cities["Metropolitan area"]
    population_by_region = out_df['Population (2016 est.)[8]'].to_list()
    win_loss_by_region = out_df['Ratio'].to_list()

    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 np.float64(stats.pearsonr(population_by_region, win_loss_by_region)[0])


def get_nhl_data():
    return out_df

nhl_correlation()
```

Out[2]:

```
0.012308996455744249
```

In [3]:

Grade cell: cell-ebe0b2dfe1067e63 Score: 20.0 / 20.0 (Top)

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]: Student's answer (Top)

```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re

def clear_data(string1):
    if re.search(r'\[[a-z]* [0-9]+\]', string1) is None:
        return string1
    else:
        return string1.replace(re.search(r'\[[a-z]* [0-9]+\]', strin
g1).group(), '')


def clear_nba_data(string1):
    if re.search(r"\*\Â\s\(\d*\)|\Â.*\(\d*\)|\s+\(\d*\)|\*\s+\(\d*
\)", string1) is None:
        return string1
    else:
        return string1.replace(re.search(r"\*\Â\s\(\d*\)|\Â.*\(\d*\)
|\s+\(\d*\)|\*\s+\(\d*\)", string1).group(), '')


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


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['NBA'] = cities['NBA'].apply(lambda x: clear_data(x))
nba_cities = cities[['Metropolitan area', 'NBA']].set_index('NBA')
nba_cities = nba_cities.drop(['—', ''], axis=0)
nba_df = nba_df[nba_df['year'] == 2018] # get only 2018 stats
population = cities[['Metropolitan area', 'Population (2016 est.)
[8]']]
population = population.set_index('Metropolitan area')
nba_df['team'] = nba_df['team'].apply(lambda x: clear_nba_data(x))
nba_df['area'] = nba_df['team'].apply(lambda x: x.split(" ")[-1])
nba_df['area'] = nba_df['area'].apply(lambda x: get_area(x))
out = []
for group, frame in nba_df.groupby('area'):
    total_wins = np.sum(pd.to_numeric(frame['W']))
    total_losses = np.sum(pd.to_numeric(frame['L']))
    total_matches = total_wins + total_losses
    ratio = total_wins / total_matches
    out_dict = {
        'Area': group,
        'Ratio': ratio
    }
    out.append(out_dict)
new_df = pd.DataFrame(out)
new_df = new_df.set_index('Area')
out_df = pd.merge(new_df, population, how="inner", left_index=True,
right_index=True)
out_df['Population (2016 est.)[8]'] = pd.to_numeric(out_df['Populati
on (2016 est.)[8]'])

print(out_dict)

def nba_correlation():
    population_by_region = []  # pass in metropolitan area populatio
n from cities
    win_loss_by_region = []  # pass in win/loss ratio from nhl_df in
the same order as cities["Metropolitan area"]
    population_by_region = out_df['Population (2016 est.)[8]'].to_li
st()
    win_loss_by_region = out_df['Ratio'].to_list()

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

    return np.float64(stats.pearsonr(population_by_region, win_loss_
by_region)[0])


def get_nba_data():
    return out_df
```

```
{'Area': 'Washington, D.C.', 'Ratio': 0.524390243902439}
```

In [5]: Grade cell: cell-e573b2b4a282b470 Score: 20.0 / 20.0 (Top)

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 [6]:

Student's answer (Top)

```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re


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

def clear_data(string1):
    if re.search(r'\[[a-z]* [0-9]+\]', string1) is None:
        return string1
    else:
        return string1.replace(re.search(r'\[[a-z]* [0-9]+\]', string1).group(), '')


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['MLB'] = cities['MLB'].apply(lambda x: clear_data(x))
mlb_cities = cities[['Metropolitan area', 'MLB']].set_index('MLB')
mlb_cities = mlb_cities.drop(['—', ''], axis=0)
mlb_df = mlb_df[mlb_df['year'] == 2018]  # get only 2018 stats no need of dropping rows
population = cities[['Metropolitan area', 'Population (2016 est.)[8]']]
population = population.set_index('Metropolitan area')
mlb_df['area'] = mlb_df['team'].apply(lambda x: x.split(" ")[-1])
mlb_df['area'] = mlb_df['area'].apply(lambda x: get_area(x))
mlb_df.at[0, 'area'] = 'Boston'
out = []
for group, frame in mlb_df.groupby('area'):
    total_wins = np.sum(pd.to_numeric(frame['W']))
    total_losses = np.sum(pd.to_numeric(frame['L']))
    total_matches = total_wins + total_losses
    ratio = total_wins / total_matches
    out_dict = {
        'Area': group,
        'Ratio': ratio
    }
    out.append(out_dict)
new_df = pd.DataFrame(out)
new_df = new_df.set_index('Area')
out_df = pd.merge(new_df, population, how="inner", left_index=True, right_index=True)
out_df['Population (2016 est.)[8]'] = pd.to_numeric(out_df['Population (2016 est.)[8]'])


def mlb_correlation():
    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"]
    population_by_region = out_df['Population (2016 est.)[8]'].to_list()
    win_loss_by_region = out_df['Ratio'].to_list()

    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 np.float64(stats.pearsonr(population_by_region, win_loss_by_region)[0])

def get_mlb_data():
    return out_df
```

In [7]: Grade cell: cell-764d4476f425c5a2 Score: 20.0 / 20.0 (Top)

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 [8]:

Student's answer (Top)

```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re


def clear_data(string1):
    if re.search(r'\[[a-z]* [0-9]+\]', string1) is None:
        return string1
    else:
        return string1.replace(re.search(r'\[[a-z]* [0-9]+\]', strin
g1).group(), '')


def clear_nba_data(string1):
    if re.search(r'\*|\+', string1) is None:
        return string1
    else:
        return string1.replace(re.search(r'\*|\+', string1).group(),
'')


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



cities=pd.read_html("assets/wikipedia_data.html")[1]
cities=cities.iloc[:-1,[0,3,5,6,7,8]]
cities['NFL'] = cities['NFL'].apply(lambda x: clear_data(x))
nfl_cities = cities[['Metropolitan area', 'NFL']].set_index('NFL')
nfl_cities = nfl_cities.drop(['—', ''], axis=0)
nfl_df = pd.read_csv("assets/nfl.csv")
nfl_df = nfl_df[nfl_df['year'] == 2018].drop([0, 5, 10, 15, 20, 25,
30, 35])  # get only 2018 stats
population = cities[['Metropolitan area', 'Population (2016 est.)
[8]']]
population = population.set_index('Metropolitan area')
nfl_df['team'] = nfl_df['team'].apply(lambda x: clear_nba_data(x))
nfl_df['area'] = nfl_df['team'].apply(lambda x: x.split(" ")[-1])
nfl_df['area'] = nfl_df['area'].apply(lambda x: get_area(x))
out = []
for group, frame in nfl_df.groupby('area'):
    total_wins = np.sum(pd.to_numeric(frame['W']))
    total_losses = np.sum(pd.to_numeric(frame['L']))
    total_matches = total_wins + total_losses
    ratio = total_wins / total_matches
    out_dict = {
        'Area': group,
        'Ratio': ratio
    }
    out.append(out_dict)
new_df = pd.DataFrame(out)
new_df = new_df.set_index('Area')
out_df = pd.merge(new_df, population, how="inner", left_index=True, right_index=True)
out_df['Population (2016 est.)[8]'] = pd.to_numeric(out_df['Population (2016 est.)[8]'])

def nfl_correlation():
    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"]
    population_by_region = out_df['Population (2016 est.)[8]'].to_list()
    win_loss_by_region = out_df['Ratio'].to_list()

    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 np.float64(stats.pearsonr(population_by_region, win_loss_by_region)[0])

def get_nfl_data():
    return out_df
```

In [9]: Grade cell: `cell-de7b148b9554dbda` Score: 20.0 / 20.0 (Top)

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` (https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_rel.html)) 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 [10]:

Student's answer (Top)

```
import pandas as pd
import scipy.stats as stats


MLB = get_mlb_data().drop('Population (2016 est.)[8]', axis=1)
NHL = get_nhl_data().drop('Population (2016 est.)[8]', axis=1)
NBA = get_nba_data().drop('Population (2016 est.)[8]', axis=1)
NFL = get_nfl_data().drop('Population (2016 est.)[8]', axis=1)
cities = pd.read_html("assets/wikipedia_data.html")[1]
cities = cities.iloc[:-1, [0, 3, 5, 6, 7, 8]]
data_set = {'NFL': NFL,
            'NBA': NBA,
            'NHL': NHL,
            'MLB': MLB}
sports = ['NFL', 'NBA', 'NHL', 'MLB']

def get_p_value(k):
    p_values = []
    for each in sports:
        df = pd.merge(data_set[k], data_set[each], how="inner", left
_index=True, right_index=True)
        corr = stats.ttest_rel(df['Ratio_x'], df['Ratio_y'])[1]
        nhl_corr = round(corr, 2)
        p_values.append(corr)
    return p_values


def sports_team_performance():
    sports = ['NFL', 'NBA', 'NHL', 'MLB']
    p_values = pd.DataFrame({k: get_p_value(k) for k in sports}, ind
ex=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 [11]:

Grade cell: `cell-fb4b9cb5ff4570a6` Score: 0.0 / 20.0 (Top)

```
You have failed this test due to an error. The traceback has been remo
ved because it may contain hidden tests. This is the exception that wa
s thrown:

AssertionError: Q5: Some or all of your values disagree with ours.
```

This assignment was graded by mooc_adswpy:dfcd934e45ab, v1.28.011023