assignment4

April 29, 2022

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.

[5]:
```
def nhl_correlation():

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

    cities=pd.read_html("assets/wikipedia_data.html")[1]
    cities=cities.iloc[:-1,[0,3,5,6,7,8]]                   #cargo las columnas␣
 ↪que necesito
    cities.replace('\[\w.*\]','', regex=True, inplace=True) # reemplazo [] por␣
 ↪espacio ''
    cities.replace('',0, regex=True, inplace=True)         # reemplazo  por 0 =␣
 ↪NaN
    cities.replace('',0, regex=True, inplace=True)          # reemplazo '' por␣
 ↪0 = NaN
    cities['NHL']=cities['NHL'].str.strip()                 # retiro espacios
    cities['NFL']=cities['NFL'].str.strip()
    cities['MLB']=cities['MLB'].str.strip()
    cities['NBA']=cities['NBA'].str.strip()
    copy_NHL=cities[['Metropolitan area','NHL','Population (2016 est.)[8]']].
 ↪dropna()
    copy_NHL.sort_values(by=['NHL'], inplace=True)
    copy_NHL['Population (2016 est.)[8]']=copy_NHL['Population (2016 est.)[8]'].
 ↪astype('int64')

    population_by_region=copy_NHL['Population (2016 est.)[8]']



    nhl_df=pd.read_csv("assets/nhl.csv")
    nhl_df=nhl_df.iloc[:35,[0,2,3,13,14]] #cargo las columnas que necesito
    nhl_df=nhl_df.drop(nhl_df.index[[0, 9, 18, 26]], axis=0) #elimino filas que␣
 ↪no necesito
    nhl_df['team'].replace("\*",'',inplace=True,regex=True)  # reemplazo * por␣
 ↪espacio"
    nhl_df['team'].replace("[\D].*\s",'',inplace=True,regex=True)
    nhl_df['team'].replace({'Knights' : 'Golden Knights', 'Jackets': 'Blue␣
 ↪Jackets', 'Leafs': 'Maple Leafs', 'Wings': 'Red Wings' }, inplace=True)
    nhl_df['team']=nhl_df['team'].str.strip()               #retiro espacios
    nhl_df['W']=nhl_df['W'].astype('int64')                 #convierto columna␣
 ↪str a int para poder dividir
```

```
    nhl_df['L']=nhl_df['L'].astype('int64')                    #convierto columna␣
↪str a int para poder dividir
    nhl_df['W/L Ratio']=nhl_df['W']/(nhl_df['W']+nhl_df['L'])  #convierto␣
↪columna str a int
    nhl_df=nhl_df.drop(nhl_df.columns[[1, 2, 3, 4]], axis=1) #elimino columnas␣
↪que no necesito
    nhl_df['team'].replace({'Rangers' : 'RangersIslandersDevils'}, inplace=True)
    nhl_df.iloc[15,1]=(nhl_df.iloc[15,1]+nhl_df.iloc[14,1]+nhl_df.iloc[12,1])/3␣
↪    #RangersIslandersDevils promedio de los 3
    nhl_df=nhl_df.drop(nhl_df.index[[14, 12]], axis=0)                         ␣
↪    #elimino filas Islanders y Devils
    nhl_df['team'].replace({'Kings' : 'KingsDucks'}, inplace=True)
    nhl_df.iloc[24,1]=(nhl_df.iloc[24,1]+nhl_df.iloc[22,1])/2     #KingsDucks␣
↪promedio de los 2
    nhl_df=nhl_df.drop(nhl_df.index[[22]], axis=0)
    nhl_df.sort_values(by=['team'], inplace=True)

    win_loss_by_region = nhl_df['W/L Ratio']

    corr, pval= stats.pearsonr(population_by_region, win_loss_by_region)



    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 corr
nhl_correlation()
```

[5]:
```
0.012486162921209909
```

[ ]:

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.

[8]:
```
def nba_correlation():
    import pandas as pd
    import numpy as np
    import scipy.stats as stats
    import re

    cities=pd.read_html("assets/wikipedia_data.html")[1]
    cities=cities.iloc[:-1,[0,3,5,6,7,8]]
    cities.replace('\[\w.*\]','', regex=True, inplace=True) # reemplazo [] por espacio ''
    cities.replace('',0, regex=True, inplace=True)         # reemplazo  por 0 = NaN
    cities.replace('',0, regex=True, inplace=True)          # reemplazo '' por 0 = NaN
    cities['NBA']=cities['NBA'].str.strip()
    copy_NBA=cities[['Metropolitan area','NBA','Population (2016 est.)[8]']].dropna()
    copy_NBA.sort_values(by=['NBA'], inplace=True)

    copy_NBA['Population (2016 est.)[8]']=copy_NBA['Population (2016 est.)[8]'].astype('int64')


    population_by_region=copy_NBA['Population (2016 est.)[8]']

    nba_df=pd.read_csv("assets/nba.csv")

    nba_df=nba_df.iloc[:30,[0,3]] #cargo las columnas que necesito
    # nba_df['W/L%']=nba_df['W/L%'].astype('int64')
    nba_df.rename(columns={"W/L%": "W/L Ratio"}, inplace=True)
    nba_df['W/L Ratio']=nba_df['W/L Ratio'].astype('float64')
    nba_df.replace('\(\d.*\)','', regex=True, inplace=True) # reemplazo () por espacio ''
    nba_df.replace('\*','', regex=True, inplace=True) # reemplazo () por espacio ''
    nba_df['team']=nba_df['team'].str.strip()
    nba_df['team'].replace("[\D].*\s",'',inplace=True,regex=True)


    nba_df['team'].replace({'Blazers' : 'Trail Blazers', 'Clippers': 'LakersClippers', 'Knicks': 'KnicksNets'}, inplace=True)
    nba_df.iloc[24,1]=(nba_df.iloc[24,1]+nba_df.iloc[25,1])/2     #KingsDucks promedio de los 2
    nba_df.iloc[10,1]=(nba_df.iloc[10,1]+nba_df.iloc[11,1])/2     #KingsDucks promedio de los 2
    nba_df=nba_df.drop(nba_df.index[[11, 25]], axis=0)
    nba_df.sort_values(by=['team'], inplace=True)

    win_loss_by_region = nba_df['W/L Ratio']
    corr, pval= stats.pearsonr(population_by_region, win_loss_by_region)
    print(corr)
    return corr
nba_correlation()
```

```
-0.17636350642182938
```

[8]:
```
-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.

[1]:
```
def mlb_correlation():
    import pandas as pd
    import numpy as np
    import scipy.stats as stats
    import re

    cities=pd.read_html("assets/wikipedia_data.html")[1]
    cities=cities.iloc[:-1,[0,3,5,6,7,8]]
    cities.replace('\[\w.*\]','', regex=True, inplace=True) # reemplazo [] por␣
 ↪espacio ''
    cities.replace('',0, regex=True, inplace=True)         # reemplazo  por 0 =␣
 ↪NaN
    cities.replace('',0, regex=True, inplace=True)          # reemplazo '' por␣
 ↪0 = NaN
    cities['MLB']=cities['MLB'].str.strip()
    copy_MLB=cities[['Metropolitan area','MLB','Population (2016 est.)[8]']].
 ↪dropna()
    copy_MLB.sort_values(by=['MLB'], inplace=True)  #len=26
    copy_MLB['Population (2016 est.)[8]']=copy_MLB['Population (2016 est.)[8]'].
 ↪astype('int64')

    population_by_region=copy_MLB['Population (2016 est.)[8]']


    mlb_df=pd.read_csv("assets/mlb.csv")
    mlb_df=mlb_df.iloc[:30,[0,1,2]] #cargo las columnas que necesito

    mlb_df['W']=mlb_df['W'].astype('int64')                  #convierto columna␣
 ↪str a int para poder dividir
    mlb_df['L']=mlb_df['L'].astype('int64')                  #convierto columna␣
 ↪str a int para poder dividir
    mlb_df['W/L Ratio']=mlb_df['W']/(mlb_df['W']+mlb_df['L'])  #convierto␣
 ↪columna str a int


    mlb_df['team'].replace("[\D].*\s",'',inplace=True,regex=True)
    mlb_df['team']=mlb_df['team'].str.strip()
    mlb_df.iloc[0,0]='Red Sox'
    mlb_df.iloc[8,0]='White Sox'
    mlb_df['team'].replace({'Yankees' : 'YankeesMets', 'Dodgers':␣
↪'DodgersAngels', 'Giants': 'GiantsAthletics', 'Cubs': 'CubsWhite Sox',␣
↪'Jays': 'Blue Jays'}, inplace=True)
    mlb_df.iloc[1,3]=(mlb_df.iloc[1,3] + mlb_df.iloc[18,3])/2
    mlb_df.iloc[25,3]=(mlb_df.iloc[25,3] + mlb_df.iloc[13,3])/2
    mlb_df.iloc[28,3]=(mlb_df.iloc[28,3] + mlb_df.iloc[11,3])/2
    mlb_df.iloc[21,3]=(mlb_df.iloc[21,3] + mlb_df.iloc[8,3])/2

    mlb_df=mlb_df.drop(mlb_df.index[[18,13,11,8]], axis=0)
    mlb_df.sort_values(by=['team'], inplace=True)

    win_loss_by_region = mlb_df['W/L Ratio']

    corr, pval= stats.pearsonr(population_by_region, win_loss_by_region)
    print(corr)
    return corr

mlb_correlation()
```

```
0.1502769830266931
```

[1]:
```
0.1502769830266931
```

[ ]:
```
```

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.

[4]:
```
def nfl_correlation():
    import pandas as pd
    import numpy as np
    import scipy.stats as stats
    import re

    cities=pd.read_html("assets/wikipedia_data.html")[1]
    cities=cities.iloc[:-1,[0,3,5,6,7,8]]
    cities.replace('\[\w.*\]','', regex=True, inplace=True) # reemplazo [] por␣
↪espacio ''
    cities.replace('',0, regex=True, inplace=True)         # reemplazo  por 0 =␣
↪NaN
    cities.replace('',0, regex=True, inplace=True)          # reemplazo '' por␣
↪0 = NaN
    cities['NFL']=cities['NFL'].str.strip()
    copy_NFL=cities[['Metropolitan area','NFL','Population (2016 est.)[8]']].
↪dropna()
    copy_NFL.sort_values(by=['NFL'], inplace=True)  #len=26
    copy_NFL['Population (2016 est.)[8]']=copy_NFL['Population (2016 est.)[8]'].
↪astype('int64')


    population_by_region=copy_NFL['Population (2016 est.)[8]']


    nfl_df=pd.read_csv("assets/nfl.csv")
    nfl_df=nfl_df.iloc[:40,[1,2,11,13,14]] #cargo las columnas que necesito
    nfl_df=nfl_df.drop(nfl_df.index[[0,5,10,15,20,25,30,35]], axis=0)
    nfl_df['W']=nfl_df['W'].astype('int64')                    #convierto columna␣
↪str a int para poder dividir
    nfl_df['L']=nfl_df['L'].astype('int64')                    #convierto columna␣
↪str a int para poder dividir
    nfl_df['W/L Ratio']=nfl_df['W']/(nfl_df['W']+ nfl_df['L'])  #convierto␣
↪columna str a int
    nfl_df.replace('\*','', regex=True, inplace=True)
    nfl_df.replace('\+','', regex=True, inplace=True)
    nfl_df['team'].replace("[\D].*\s",'',inplace=True,regex=True)
    nfl_df['team']=nfl_df['team'].str.strip()
    nfl_df['team'].replace({'Giants' : 'GiantsJets', 'Rams': 'RamsChargers',␣
↪'49ers': '49ersRaiders'}, inplace=True)
    nfl_df.loc[24,('W/L Ratio')]=(nfl_df.loc[24,('W/L Ratio')] + nfl_df.
↪loc[4,('W/L Ratio')])/2
    nfl_df.loc[36,('W/L Ratio')]=(nfl_df.loc[36,('W/L Ratio')] + nfl_df.
↪loc[17,('W/L Ratio')])/2
    nfl_df.loc[38,('W/L Ratio')]=(nfl_df.loc[38,('W/L Ratio')] + nfl_df.
↪loc[19,('W/L Ratio')])/2
    nfl_df=nfl_df.drop([4,17,19],axis=0)
    nfl_df.sort_values(by=['team'], inplace=True)

    win_loss_by_region = nfl_df['W/L Ratio']


    corr, pval= stats.pearsonr(population_by_region, win_loss_by_region)
    print(corr)
    return corr

nfl_correlation()
```

```
0.004922121149349429
```

```
[4]: 0.004922121149349429
```

[ ]:

```
```

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.

[ ]:

```
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
```

[ ]:

```
```