assignment4

September 23, 2023

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!

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.

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.

[261]:
```
import pandas as pd
import numpy as np
import scipy.stats as stats
import re

def nhl_correlation():

    # YOUR CODE HERE

    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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

    cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
    cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
    cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
    cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

    cities_nhl = cities[['City', 'Population', 'NHL']]

    cities_nhl_split = cities_nhl.iloc[0:2]

    cities_nhl_split['NHL'] = cities_nhl_split['NHL'].str.split(' ')

    cities_nhl_split = cities_nhl_split.explode('NHL')

    cities_nhl = pd.concat([cities_nhl_split, cities_nhl.iloc[2:]])

    nhl_df = nhl_df[nhl_df["year"] == 2018]

    substring = 'Division'
    filter = nhl_df['team'].str.contains(substring)
    nhl_df_filtered = nhl_df[~filter]

    nhl_df_filtered.W = nhl_df_filtered.W.astype(float)

    nhl_df_filtered.L = nhl_df_filtered.L.astype(float)
```

```
  nhl_df_filtered["win_loss_ratio"] = nhl_df_filtered.W / (nhl_df_filtered.W␣
↪+ nhl_df_filtered.L)

  nhl_df_filtered["team"] = nhl_df_filtered["team"].str.replace(r'\*', '',␣
↪regex = True)

  mascot = ['Lightning', 'Bruins', 'Maple Leafs', 'Panthers', 'Red Wings',␣
↪'Canadiens', 'Senators', 'Sabres', 'Capitals', 'Penguins',
          'Flyers', 'Blue Jackets', 'Devils', 'Hurricanes', 'Islanders',␣
↪'Rangers', 'Predators', 'Jets', 'Wild', 'Avalanche', 'Blues',
          'Stars', 'Blackhawks', 'Golden Knights', 'Ducks', 'Sharks',␣
↪'Kings', 'Flames', 'Oilers', 'Canucks', 'Coyotes']

  nhl_df_filtered['mascot'] = mascot

  combined = pd.merge(nhl_df_filtered, cities_nhl, how = 'left', left_on =␣
↪'mascot', right_on = 'NHL')

  combined = combined[['City', 'mascot', 'Population', 'win_loss_ratio']]

  combined_final = combined.groupby(by = 'City').mean('win_loss_ratio')

  combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
↪'City', right_on = 'City')

  combined_final = combined_final[['City', 'win_loss_ratio_x', 'Population']]

  combined_final.columns = ['City', 'win_loss_ratio', 'Population']

  combined_final.Population = combined_final.Population.astype(float)

  combined_final = combined_final.drop_duplicates(subset = ['City'])

  population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
↪metropolitan area population from cities
  win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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]
```

```
    raise NotImplementedError()
```

[ ]:

[262]:
```
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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

cities_nhl = cities[['City', 'Population', 'NHL']]

cities_nhl_split = cities_nhl.iloc[0:2]

cities_nhl_split['NHL'] = cities_nhl_split['NHL'].str.split(' ')

cities_nhl_split = cities_nhl_split.explode('NHL')

cities_nhl = pd.concat([cities_nhl_split, cities_nhl.iloc[2:]])

nhl_df = nhl_df[nhl_df["year"] == 2018]

substring = 'Division'
filter = nhl_df['team'].str.contains(substring)
nhl_df_filtered = nhl_df[~filter]

nhl_df_filtered.W = nhl_df_filtered.W.astype(float)

nhl_df_filtered.L = nhl_df_filtered.L.astype(float)

nhl_df_filtered["win_loss_ratio"] = nhl_df_filtered.W / (nhl_df_filtered.W +␣
 ↪nhl_df_filtered.L)

nhl_df_filtered["team"] = nhl_df_filtered["team"].str.replace(r'\*', '', regex␣
 ↪= True)

nhl_df_filtered

mascot = ['Lightning', 'Bruins', 'Maple Leafs', 'Panthers', 'Red Wings',␣
 ↪'Canadiens', 'Senators', 'Sabres', 'Capitals', 'Penguins',
```

```
       'Flyers', 'Blue Jackets', 'Devils', 'Hurricanes', 'Islanders',␣
 ↪'Rangers', 'Predators', 'Jets', 'Wild', 'Avalanche', 'Blues',
       'Stars', 'Blackhawks', 'Golden Knights', 'Ducks', 'Sharks', 'Kings',␣
 ↪'Flames', 'Oilers', 'Canucks', 'Coyotes']

nhl_df_filtered['mascot'] = mascot

nhl_df_filtered

combined = pd.merge(nhl_df_filtered, cities_nhl, how = 'left', left_on =␣
 ↪'mascot', right_on = 'NHL')

combined = combined[['City', 'mascot', 'Population', 'win_loss_ratio']]

combined_final = combined.groupby(by = 'City').mean('win_loss_ratio')

combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
 ↪'City', right_on = 'City')

combined_final = combined_final[['City', 'win_loss_ratio_x', 'Population']]

combined_final.columns = ['City', 'win_loss_ratio', 'Population']

combined_final.Population = combined_final.Population.astype(float)

combined_final = combined_final.drop_duplicates(subset = ['City'])

combined_final

population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
 ↪metropolitan area population from cities
win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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"

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

[262]:
```
0.012486162921209909
```

[263]:
```
nhl_correlation()
```

[263]:
```
0.012486162921209909
```

[264]:
```
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]]

cities.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

cities_nhl = cities[['City', 'Population', 'NHL']]

cities_nhl_split = cities_nhl.iloc[0:2]

cities_nhl_split['NHL'] = cities_nhl_split['NHL'].str.split(' ')

cities_nhl_split = cities_nhl_split.explode('NHL')

cities_nhl = pd.concat([cities_nhl_split, cities_nhl.iloc[2:]])

nhl_df = nhl_df[nhl_df["year"] == 2018]

substring = 'Division'
filter = nhl_df['team'].str.contains(substring)
nhl_df_filtered = nhl_df[~filter]

nhl_df_filtered.W = nhl_df_filtered.W.astype(float)

nhl_df_filtered.L = nhl_df_filtered.L.astype(float)

nhl_df_filtered["win_loss_ratio"] = nhl_df_filtered.W / nhl_df_filtered.L

nhl_df_filtered["team"] = nhl_df_filtered["team"].str.replace(r'\*', '', regex␣
 ↪= True)

mascot = ['Lightning', 'Bruins', 'Maple Leafs', 'Panthers', 'Red Wings',␣
 ↪'Canadiens', 'Senators', 'Sabres', 'Capitals', 'Penguins',
       'Flyers', 'Blue Jackets', 'Devils', 'Hurricanes', 'Islanders',␣
 ↪'Rangers', 'Predators', 'Jets', 'Wild', 'Avalanche', 'Blues',
       'Stars', 'Blackhawks', 'Golden Knights', 'Ducks', 'Sharks', 'Kings',␣
 ↪'Flames', 'Oilers', 'Canucks', 'Coyotes']
```

```
nhl_df_filtered['mascot'] = mascot

combined = pd.merge(nhl_df_filtered, cities_nhl, how = 'left', left_on =␣
 ↪'mascot', right_on = 'NHL')

combined

# combined = combined[['City', 'mascot', 'Population', 'win_loss_ratio']]

# combined_final = combined.groupby(by = 'City').mean('win_loss_ratio')

# combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
 ↪'City', right_on = 'City')

# combined_final = combined_final[['City', 'win_loss_ratio_x', 'Population']]

# combined_final.columns = ['City', 'win_loss_ratio', 'Population']

# combined_final.Population = combined_final.Population.astype(float)

# combined_final = combined_final.drop_duplicates(subset = ['City'])

# population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
 ↪metropolitan area population from cities
# win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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"

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

[264]:

	team	GP	W	L	OL	PTS	PTS%	GF	GA	SRS	\
0	Tampa Bay Lightning	82	54.0	23.0	5	113	.689	296	236	0.66	
1	Boston Bruins	82	50.0	20.0	12	112	.683	270	214	0.62	
2	Toronto Maple Leafs	82	49.0	26.0	7	105	.640	277	232	0.49	
3	Florida Panthers	82	44.0	30.0	8	96	.585	248	246	-0.01	
4	Detroit Red Wings	82	30.0	39.0	13	73	.445	217	255	-0.48	
5	Montreal Canadiens	82	29.0	40.0	13	71	.433	209	264	-0.68	
6	Ottawa Senators	82	28.0	43.0	11	67	.409	221	291	-0.85	
7	Buffalo Sabres	82	25.0	45.0	12	62	.378	199	280	-0.98	
8	Washington Capitals	82	49.0	26.0	7	105	.640	259	239	0.21	
9	Pittsburgh Penguins	82	47.0	29.0	6	100	.610	272	250	0.23	

10	Philadelphia Flyers	82	42.0	26.0	14	98	.598	251	243	0.07
11	Columbus Blue Jackets	82	45.0	30.0	7	97	.591	242	230	0.11
12	New Jersey Devils	82	44.0	29.0	9	97	.591	248	244	0.02
13	Carolina Hurricanes	82	36.0	35.0	11	83	.506	228	256	-0.35
14	New York Islanders	82	35.0	37.0	10	80	.488	264	296	-0.40
15	New York Rangers	82	34.0	39.0	9	77	.470	231	268	-0.46
16	Nashville Predators	82	53.0	18.0	11	117	.713	267	211	0.71
17	Winnipeg Jets	82	52.0	20.0	10	114	.695	277	218	0.74
18	Minnesota Wild	82	45.0	26.0	11	101	.616	253	232	0.29
19	Colorado Avalanche	82	43.0	30.0	9	95	.579	257	237	0.28
20	St. Louis Blues	82	44.0	32.0	6	94	.573	226	222	0.10
21	Dallas Stars	82	42.0	32.0	8	92	.561	235	225	0.17
22	Chicago Blackhawks	82	33.0	39.0	10	76	.463	229	256	-0.26
23	Vegas Golden Knights	82	51.0	24.0	7	109	.665	272	228	0.52
24	Anaheim Ducks	82	44.0	25.0	13	101	.616	235	216	0.24
25	San Jose Sharks	82	45.0	27.0	10	100	.610	252	229	0.28
26	Los Angeles Kings	82	45.0	29.0	8	98	.598	239	203	0.44
27	Calgary Flames	82	37.0	35.0	10	84	.512	218	248	-0.33
28	Edmonton Oilers	82	36.0	40.0	6	78	.476	234	263	-0.32
29	Vancouver Canucks	82	31.0	40.0	11	73	.445	218	264	-0.51
30	Arizona Coyotes	82	29.0	41.0	12	70	.427	208	256	-0.53

	SOS	RPt%	ROW	year	League	win_loss_ratio	mascot	\
0	-0.07	.634	48	2018	NHL	2.347826	Lightning	
1	-0.07	.610	47	2018	NHL	2.500000	Bruins	
2	-0.06	.567	42	2018	NHL	1.884615	Maple Leafs	
3	-0.04	.537	41	2018	NHL	1.466667	Panthers	
4	-0.01	.341	25	2018	NHL	0.769231	Red Wings	
5	0.00	.378	27	2018	NHL	0.725000	Canadiens	
6	0.00	.372	26	2018	NHL	0.651163	Senators	
7	0.01	.311	24	2018	NHL	0.555556	Sabres	
8	-0.04	.585	46	2018	NHL	1.884615	Capitals	
9	-0.04	.573	45	2018	NHL	1.620690	Penguins	
10	-0.03	.543	40	2018	NHL	1.615385	Flyers	
11	-0.04	.537	39	2018	NHL	1.500000	Blue Jackets	
12	-0.03	.530	39	2018	NHL	1.517241	Devils	
13	-0.01	.439	33	2018	NHL	1.028571	Hurricanes	
14	-0.01	.427	32	2018	NHL	0.945946	Islanders	
15	-0.01	.427	31	2018	NHL	0.871795	Rangers	
16	0.03	.652	47	2018	NHL	2.944444	Predators	
17	0.02	.622	48	2018	NHL	2.600000	Jets	
18	0.04	.549	42	2018	NHL	1.730769	Wild	
19	0.04	.518	41	2018	NHL	1.433333	Avalanche	
20	0.05	.518	41	2018	NHL	1.375000	Blues	
21	0.04	.506	38	2018	NHL	1.312500	Stars	
22	0.07	.409	32	2018	NHL	0.846154	Blackhawks	
23	-0.01	.616	47	2018	NHL	2.125000	Golden Knights	

```
24   0.01  .555  40  2018    NHL        1.760000          Ducks
25   0.00  .537  40  2018    NHL        1.666667         Sharks
26   0.00  .543  43  2018    NHL        1.551724          Kings
27   0.03  .470  35  2018    NHL        1.057143         Flames
28   0.03  .415  31  2018    NHL        0.900000         Oilers
29   0.05  .409  31  2018    NHL        0.775000        Canucks
30   0.05  .372  27  2018    NHL        0.707317        Coyotes
```

	City	Population	NHL
0	Tampa Bay Area	3032171	Lightning
1	Boston	4794447	Bruins
2	Toronto	5928040	Maple Leafs
3	Miami-Fort Lauderdale	6066387	Panthers
4	Detroit	4297617	Red Wings
5	Montreal	4098927	Canadiens
6	Ottawa	1323783	Senators
7	Buffalo	1132804	Sabres
8	Washington, D.C.	6131977	Capitals
9	Pittsburgh	2342299	Penguins
10	Philadelphia	6070500	Flyers
11	Columbus	2041520	Blue Jackets
12	New York City	20153634	Devils
13	Raleigh	1302946	Hurricanes
14	New York City	20153634	Islanders
15	New York City	20153634	Rangers
16	Nashville	1865298	Predators
17	Winnipeg	778489	Jets
18	Minneapolis-Saint Paul	3551036	Wild
19	Denver	2853077	Avalanche
20	St. Louis	2807002	Blues
21	Dallas-Fort Worth	7233323	Stars
22	Chicago	9512999	Blackhawks
23	Las Vegas	2155664	Golden Knights
24	Los Angeles	13310447	Ducks
25	San Francisco Bay Area	6657982	Sharks
26	Los Angeles	13310447	Kings
27	Calgary	1392609	Flames
28	Edmonton	1321426	Oilers
29	Vancouver	2463431	Canucks
30	Phoenix	4661537	Coyotes

[265]:
```
cities=pd.read_html("assets/wikipedia_data.html")[1]
cities=cities.iloc[:-1,[0,3,5,6,7,8]]

# cities
```

[266]:
```
cities.iloc[[10]]
```

[266]:
```
         Metropolitan area Population (2016 est.)[8]       NFL      MLB   NBA  \
10  Miami-Fort Lauderdale                    6066387  Dolphins  Marlins  Heat

          NHL
10  Panthers
```

[267]:
```
cities.iloc[[30]]
```

[267]:
```
   Metropolitan area Population (2016 est.)[8] NFL        MLB NBA        NHL
30          Montreal                    4098927   -  [note 59]   -  Canadiens
```

[268]:
```
cities.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']
```

[269]:
```
# cities
```

[270]:
```
cities.columns
```

[270]:
```
Index(['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL'], dtype='object')
```

[271]:
```
test = cities.iloc[[0]]

test
```

[271]:
```
            City Population                   NFL                    MLB  \
0  New York City   20153634  Giants Jets[note 1]  Yankees Mets[note 2]

           NBA                                NHL
0  Knicks Nets  Rangers Islanders Devils[note 3]
```

[272]:
```
test[test.columns[2]].iloc[0]
```

[272]:
```
'Giants Jets[note 1]'
```

[273]:
```
import re

re.sub('\[.*\]', '',test[test.columns[2]].iloc[0])

# this regex expression works
```

[273]:
```
'Giants Jets'
```

[274]:
```
copy = cities.copy()

copy.head()
```

[274]:
```
                   City Population                   NFL  \
0           New York City   20153634    Giants Jets[note 1]
```

```
1             Los Angeles   13310447  Rams Chargers[note 4]
2  San Francisco Bay Area    6657982  49ers Raiders[note 6]
3                 Chicago    9512999          Bears[note 8]
4       Dallas-Fort Worth    7233323                Cowboys

                     MLB              NBA                                NHL
0  Yankees Mets[note 2]      Knicks Nets  Rangers Islanders Devils[note 3]
1        Dodgers Angels  Lakers Clippers                     Kings Ducks
2      Giants Athletics         Warriors                  Sharks[note 7]
3        Cubs White Sox    Bulls[note 9]                      Blackhawks
4               Rangers        Mavericks                           Stars
```

[275]:
```
# df['P'] = df['P'].str.replace(r'\D+', '', regex=True).astype('int')

copy['NFL'] = copy['NFL'].str.replace(r'\[.*\]','', regex = True)

# this worked
```

[276]:
```
cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)
```

[277]:
```
# cities
```

[278]:
```
cities_nhl = cities[['City', 'Population', 'NHL']]

# cities_nhl
```

[279]:
```
# cities_nhl['NHL'] = cities_nhl['NHL'].str.split(' ')

# cities_nhl = cities_nhl.explode('NHL')
```

[280]:
```
cities_nhl_split = cities_nhl.iloc[0:2]

cities_nhl_split
```

[280]:
```
           City Population                      NHL
0  New York City   20153634  Rangers Islanders Devils
1    Los Angeles   13310447              Kings Ducks
```

[281]:
```
cities_nhl_split['NHL'] = cities_nhl_split['NHL'].str.split(' ')

cities_nhl_split = cities_nhl_split.explode('NHL')

cities_nhl_split
```

[281]:
```
            City Population        NHL
0  New York City   20153634    Rangers
0  New York City   20153634  Islanders
0  New York City   20153634     Devils
1    Los Angeles   13310447      Kings
1    Los Angeles   13310447      Ducks
```

[282]:
```
cities_nhl = pd.concat([cities_nhl_split, cities_nhl.iloc[2:]])
```

[283]:
```
# cities_nhl

# I was able to create separate rows for NYC and LA that have multiple NHL␣
↪teams
```

[284]:
```
nhl_df=pd.read_csv("assets/nhl.csv")

nhl_df = nhl_df[nhl_df["year"] == 2018]
```

[285]:
```
# nhl_df
```

[286]:
```
substring = 'Division'
filter = nhl_df['team'].str.contains(substring)
nhl_df_filtered = nhl_df[~filter]
```

[287]:
```
# nhl_df_filtered
```

[288]:
```
nhl_df_filtered.W = nhl_df_filtered.W.astype(float)
```

[289]:
```
nhl_df_filtered.L = nhl_df_filtered.L.astype(float)
```

[290]:
```
nhl_df_filtered["win_loss_ratio"] = nhl_df_filtered.W / nhl_df_filtered.L
```

[291]:
```
# nhl_df_filtered
```

[292]:
```
# cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)

nhl_df_filtered["team"] = nhl_df_filtered["team"].str.replace(r'\*', '', regex␣
↪= True)
```

[293]:
```
# nhl_df_filtered
```

[294]:
```
mascot = ['Lightning', 'Bruins', 'Maple Leafs', 'Panthers', 'Red Wings',␣
↪'Canadiens', 'Senators', 'Sabres', 'Capitals', 'Penguins',
                          'Flyers', 'Blue Jackets', 'Devils', 'Hurricanes',␣
↪'Islanders', 'Rangers', 'Predators', 'Jets', 'Wild', 'Avalanche', 'Blues',
                          'Stars', 'Blackhawks', 'Golden Knights', 'Ducks',␣
↪'Sharks', 'Kings', 'Flames', 'Oilers', 'Canucks', 'Coyotes']
```

[295]:
```
nhl_df_filtered['mascot'] = mascot
```

[296]:
```
nhl_df_filtered.head()
```

[296]:

	team	GP	W	L	OL	PTS	PTS%	GF	GA	SRS	SOS	RPt%	ROW	year	League	win_loss_ratio	mascot
1	Tampa Bay Lightning	82	54.0	23.0	5	113	.689	296	236	0.66	-0.07	.634	48	2018	NHL	2.347826	Lightning
2	Boston Bruins	82	50.0	20.0	12	112	.683	270	214	0.62	-0.07	.610	47	2018	NHL	2.500000	Bruins
3	Toronto Maple Leafs	82	49.0	26.0	7	105	.640	277	232	0.49	-0.06	.567	42	2018	NHL	1.884615	Maple Leafs
4	Florida Panthers	82	44.0	30.0	8	96	.585	248	246	-0.01	-0.04	.537	41	2018	NHL	1.466667	Panthers
5	Detroit Red Wings	82	30.0	39.0	13	73	.445	217	255	-0.48	-0.01	.341	25	2018	NHL	0.769231	Red Wings

[297]:
```
combined = pd.merge(nhl_df_filtered, cities_nhl, how = 'left', left_on =␣
↪'mascot', right_on = 'NHL')
```

[298]:
```
# combined
```

[299]:
```
combined = combined[['City', 'mascot', 'Population', 'win_loss_ratio']]

# combined
```

[300]:
```
combined_final = combined.groupby(by = 'City').mean('win_loss_ratio')
```

[301]:
```
combined = pd.merge(nhl_df_filtered, cities_nhl, how = 'left', left_on =␣
↪'mascot', right_on = 'NHL')

combined = combined[['City', 'mascot', 'Population', 'win_loss_ratio']]

combined_final = combined.groupby(by = 'City').mean('win_loss_ratio')

combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
↪'City', right_on = 'City')

combined_final = combined_final[['City', 'win_loss_ratio_x', 'Population']]

combined_final.columns = ['City', 'win_loss_ratio', 'Population']

combined_final.Population = combined_final.Population.astype(float)

combined_final.columns = ['City', 'win_loss_ratio', 'Population']
```

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.

[302]:
```
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]]

def nba_correlation():
    # YOUR CODE HERE

    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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

    cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
    cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
    cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
    cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

    cities_nba = cities[['City', 'Population', 'NBA']]

    cities_nba_split = cities_nba.iloc[0:2]

    cities_nba_split['NBA'] = cities_nba_split['NBA'].str.split(' ')

    cities_nba_split = cities_nba_split.explode('NBA')

    cities_nba = pd.concat([cities_nba_split, cities_nba.iloc[2:]])

    nba_df_filtered = nba_df[nba_df["year"] == 2018]

    nba_df_filtered["team"] = nba_df_filtered["team"].str.replace(r'\*', '',␣
↪regex = True)

    nba_df_filtered["team"] = nba_df_filtered["team"].str.replace(r'\(.*\)',␣
↪'', regex = True)

    # nba_df_filtered
```

```
    # mascot = ['Lightning', 'Bruins', 'Maple Leafs', 'Panthers', 'Red Wings',␣
↪'Canadiens', 'Senators', 'Sabres', 'Capitals', 'Penguins',
    #           'Flyers', 'Blue Jackets', 'Devils', 'Hurricanes', 'Islanders',␣
↪'Rangers', 'Predators', 'Jets', 'Wild', 'Avalanche', 'Blues',
    #           'Stars', 'Blackhawks', 'Golden Knights', 'Ducks', 'Sharks',␣
↪'Kings', 'Flames', 'Oilers', 'Canucks', 'Coyotes']

    # nhl_df_filtered['mascot'] = mascot

    mascot = ['Raptors', 'Celtics', '76ers', 'Cavaliers', 'Pacers', 'Heat',␣
↪'Bucks', 'Wizards', 'Pistons', 'Hornets', 'Knicks',
              'Nets', 'Bulls', 'Magic', 'Hawks', 'Rockets', 'Warriors', 'Trail␣
↪Blazers', 'Thunder', 'Jazz', 'Pelicans', 'Spurs',
              'Timberwolves', 'Nuggets', 'Clippers', 'Lakers', 'Kings',␣
↪'Mavericks', 'Grizzlies', 'Suns']

    nba_df_filtered['mascot'] = mascot

    combined = pd.merge(nba_df_filtered, cities_nba, how = 'left', left_on =␣
↪'mascot', right_on = 'NBA')

    combined = combined[['City', 'mascot', 'Population', 'W/L%']]

    combined[['W/L%']] = combined[['W/L%']].astype(float)

    combined_final = combined.groupby(by = 'City').mean('W/L%')

    combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
↪'City', right_on = 'City')

    combined_final = combined_final[['City', 'W/L%_x', 'Population']]

    combined_final.columns = ['City', 'win_loss_ratio', 'Population']

    combined_final.Population = combined_final.Population.astype(float)

    combined_final = combined_final.drop_duplicates(subset = ['City'])

    population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
↪metropolitan area population from cities
    win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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 NBA"

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

    raise NotImplementedError()
```

[ ]:

[303]:
```
nba_correlation()
```

[303]:
```
-0.17636350642182935
```

[304]:
```
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]]

# YOUR CODE HERE

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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

cities_nba = cities[['City', 'Population', 'NBA']]

cities_nba

cities_nba_split = cities_nba.iloc[0:2]

cities_nba_split['NBA'] = cities_nba_split['NBA'].str.split(' ')

cities_nba_split = cities_nba_split.explode('NBA')

cities_nba = pd.concat([cities_nba_split, cities_nba.iloc[2:]])
```

```
nba_df_filtered = nba_df[nba_df["year"] == 2018]

nba_df_filtered["team"] = nba_df_filtered["team"].str.replace(r'\*', '', regex␣
 ↪= True)

nba_df_filtered["team"] = nba_df_filtered["team"].str.replace(r'\(.*\)', '',␣
 ↪regex = True)

mascot = ['Raptors', 'Celtics', '76ers', 'Cavaliers', 'Pacers', 'Heat',␣
 ↪'Bucks', 'Wizards', 'Pistons', 'Hornets', 'Knicks',
        'Nets', 'Bulls', 'Magic', 'Hawks', 'Rockets', 'Warriors', 'Trail␣
 ↪Blazers', 'Thunder', 'Jazz', 'Pelicans', 'Spurs',
        'Timberwolves', 'Nuggets', 'Clippers', 'Lakers', 'Kings', 'Mavericks',␣
 ↪'Grizzlies', 'Suns']

nba_df_filtered['mascot'] = mascot

combined = pd.merge(nba_df_filtered, cities_nba, how = 'left', left_on =␣
 ↪'mascot', right_on = 'NBA')

combined = combined[['City', 'mascot', 'Population', 'W/L%']]

combined[['W/L%']] = combined[['W/L%']].astype(float)

combined_final = combined.groupby(by = 'City').mean('W/L%')

combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
 ↪'City', right_on = 'City')

combined_final = combined_final[['City', 'W/L%_x', 'Population']]

combined_final.columns = ['City', 'win_loss_ratio', 'Population']

combined_final.Population = combined_final.Population.astype(float)

combined_final = combined_final.drop_duplicates(subset = ['City'])

population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
 ↪metropolitan area population from cities
win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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 NBA"
```

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

[304]: -0.17636350642182935

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.

[305]:
```
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]]

def mlb_correlation():
    # YOUR CODE HERE

    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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

    cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
    cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
    cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
    cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

    cities_mlb = cities[['City', 'Population', 'MLB']]

    cities_mlb_split = cities_mlb.iloc[0:4]

    cities_mlb_split['MLB'] = cities_mlb_split['MLB'].str.split(' ')

    cities_mlb_split = cities_mlb_split.explode('MLB')

    cities_mlb = pd.concat([cities_mlb_split, cities_mlb.iloc[4:]])

    cities_mlb_split_white_sox = cities_mlb.iloc[7:9, ]

    cities_mlb_split_white_sox = cities_mlb_split_white_sox.groupby(['City',␣
↪'Population'])['MLB'].apply(' '.join).reset_index()
```

```
  cities_mlb = pd.concat([cities_mlb.iloc[0:7], cities_mlb_split_white_sox,␣
↪cities_mlb.iloc[9:]])

  cities_mlb

  mlb_df_filtered =  mlb_df[mlb_df["year"] == 2018]

  mlb_df_filtered

  mascot = ['Red Sox', 'Yankees', 'Rays', 'Blue Jays', 'Orioles', 'Indians',␣
↪'Twins', 'Tigers', 'White Sox', 'Royals', 'Astros',
           'Athletics', 'Mariners', 'Angels', 'Rangers', 'Braves',␣
↪'Nationals', 'Phillies', 'Mets', 'Marlins', 'Brewers',
           'Cubs', 'Cardinals', 'Pirates', 'Reds', 'Dodgers', 'Rockies',␣
↪'Diamondbacks', 'Giants', 'Padres']

  mlb_df_filtered['mascot'] = mascot

  combined = pd.merge(mlb_df_filtered, cities_mlb, how = 'left', left_on =␣
↪'mascot', right_on = 'MLB')

  combined = combined[['City', 'mascot', 'Population', 'W-L%']]

  combined[['W-L%']] = combined[['W-L%']].astype(float)

  combined_final = combined.groupby(by = 'City').mean('W-L%')

  combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
↪'City', right_on = 'City')

  combined_final = combined_final[['City', 'W-L%_x', 'Population']]

  combined_final.columns = ['City', 'win_loss_ratio', 'Population']

  combined_final.Population = combined_final.Population.astype(float)

  combined_final = combined_final.drop_duplicates(subset = ['City'])

  population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
↪metropolitan area population from cities
  win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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), "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]

    raise NotImplementedError()
```

[ ]:

[306]:
```
mlb_correlation()
```

[306]:
```
0.15003737475409498
```

[307]:
```
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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

cities_mlb = cities[['City', 'Population', 'MLB']]

cities_mlb_split = cities_mlb.iloc[0:4]

cities_mlb_split['MLB'] = cities_mlb_split['MLB'].str.split(' ')

cities_mlb_split = cities_mlb_split.explode('MLB')

cities_mlb = pd.concat([cities_mlb_split, cities_mlb.iloc[4:]])

cities_mlb_split_white_sox = cities_mlb.iloc[7:9, ]

cities_mlb_split_white_sox = cities_mlb_split_white_sox.groupby(['City',␣
↪'Population'])['MLB'].apply(' '.join).reset_index()

cities_mlb = pd.concat([cities_mlb.iloc[0:7], cities_mlb_split_white_sox,␣
↪cities_mlb.iloc[9:]])

cities_mlb

mlb_df_filtered =  mlb_df[mlb_df["year"] == 2018]
```

```
mlb_df_filtered

mascot = ['Red Sox', 'Yankees', 'Rays', 'Blue Jays', 'Orioles', 'Indians',␣
 ↪'Twins', 'Tigers', 'White Sox', 'Royals', 'Astros',
        'Athletics', 'Mariners', 'Angels', 'Rangers', 'Braves', 'Nationals',␣
 ↪'Phillies', 'Mets', 'Marlins', 'Brewers',
        'Cubs', 'Cardinals', 'Pirates', 'Reds', 'Dodgers', 'Rockies',␣
 ↪'Diamondbacks', 'Giants', 'Padres']

mlb_df_filtered['mascot'] = mascot

combined = pd.merge(mlb_df_filtered, cities_mlb, how = 'left', left_on =␣
 ↪'mascot', right_on = 'MLB')

combined = combined[['City', 'mascot', 'Population', 'W-L%']]

combined[['W-L%']] = combined[['W-L%']].astype(float)

combined_final = combined.groupby(by = 'City').mean('W-L%')

combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
 ↪'City', right_on = 'City')

combined_final = combined_final[['City', 'W-L%_x', 'Population']]

combined_final.columns = ['City', 'win_loss_ratio', 'Population']

combined_final.Population = combined_final.Population.astype(float)

combined_final = combined_final.drop_duplicates(subset = ['City'])

population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
 ↪metropolitan area population from cities
win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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"

stats.pearsonr(population_by_region, win_loss_by_region)
```

[307]: (0.15003737475409498, 0.46442827201123427)

[308]:
```
# cities_mlb_split_white_sox = cities_mlb.iloc[7:9, ]

# cities_mlb_split_white_sox = cities_mlb_split_white_sox.groupby(['City',␣
 ↪'Population'])['MLB'].apply(' '.join).reset_index()

# cities_mlb_split_white_sox


# code from stack overflow below:
# # df.groupby(['name','month'])['text'].apply(','.join).reset_index()
```

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.

[309]:
```
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]]

def nfl_correlation():
    # YOUR CODE HERE
    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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

    cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
    cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
    cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
    cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

    cities_nfl = cities[['City', 'Population', 'NFL']]

    cities_nfl_split = cities_nfl.iloc[0:3]

    cities_nfl_split['NFL'] = cities_nfl_split['NFL'].str.split(' ')

    cities_nfl_split = cities_nfl_split.explode('NFL')

    cities_nfl = pd.concat([cities_nfl_split, cities_nfl.iloc[3:]])
```

```
  nfl_df =  nfl_df[nfl_df["year"] == 2018]

  substring_1 = 'AFC'
  substring_2 = 'NFC'
  nfl_df['team'] = nfl_df['team'].astype(str)
  filter_1 = nfl_df['team'].str.contains(substring_1)
  nfl_df_filtered = nfl_df[~filter_1]
  filter_2 = nfl_df_filtered['team'].str.contains(substring_2)
  nfl_df_filtered = nfl_df_filtered[~filter_2]

  nfl_df_filtered["team"] = nfl_df_filtered["team"].str.replace(r'\*', '',␣
↪regex = True)
  nfl_df_filtered["team"] = nfl_df_filtered["team"].str.replace(r'\+', '',␣
↪regex = True)

  mascot = ['Patriots', 'Dolphins', 'Bills', 'Jets', 'Ravens', 'Steelers',␣
↪'Browns', 'Bengals', 'Texans', 'Colts',
            'Titans', 'Jaguars', 'Chiefs', 'Chargers', 'Broncos', 'Raiders',␣
↪'Cowboys', 'Eagles', 'Redskins', 'Giants',
            'Bears', 'Vikings', 'Packers', 'Lions', 'Saints', 'Panthers',␣
↪'Falcons', 'Buccaneers', 'Rams', 'Seahawks',
            '49ers', 'Cardinals']

  nfl_df_filtered['mascot'] = mascot

  combined = pd.merge(nfl_df_filtered, cities_nfl, how = 'left', left_on =␣
↪'mascot', right_on = 'NFL')

  combined = combined[['City', 'mascot', 'Population', 'W-L%']]

  combined[['W-L%']] = combined[['W-L%']].astype(float)

  combined_final = combined.groupby(by = 'City').mean('W-L%')

  combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
↪'City', right_on = 'City')

  combined_final = combined_final[['City', 'W-L%_x', 'Population']]

  combined_final.columns = ['City', 'win_loss_ratio', 'Population']

  combined_final.Population = combined_final.Population.astype(float)

  combined_final = combined_final.drop_duplicates(subset = ['City'])
```

```
    population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
↪metropolitan area population from cities
    win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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), "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]

    raise NotImplementedError()
```

[ ]:

[310]:
```
nfl_correlation()
```

[310]:
```
0.004282141436393022
```

[311]:
```
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]]

# YOUR CODE HERE
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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

cities_nfl = cities[['City', 'Population', 'NFL']]

cities_nfl_split = cities_nfl.iloc[0:3]

cities_nfl_split['NFL'] = cities_nfl_split['NFL'].str.split(' ')
```

```
cities_nfl_split = cities_nfl_split.explode('NFL')

cities_nfl = pd.concat([cities_nfl_split, cities_nfl.iloc[3:]])

nfl_df =  nfl_df[nfl_df["year"] == 2018]

substring_1 = 'AFC'
substring_2 = 'NFC'
nfl_df['team'] = nfl_df['team'].astype(str)
filter_1 = nfl_df['team'].str.contains(substring_1)
nfl_df_filtered = nfl_df[~filter_1]
filter_2 = nfl_df_filtered['team'].str.contains(substring_2)
nfl_df_filtered = nfl_df_filtered[~filter_2]

nfl_df_filtered["team"] = nfl_df_filtered["team"].str.replace(r'\*', '', regex␣
 ↪= True)
nfl_df_filtered["team"] = nfl_df_filtered["team"].str.replace(r'\+', '', regex␣
 ↪= True)

mascot = ['Patriots', 'Dolphins', 'Bills', 'Jets', 'Ravens', 'Steelers',␣
 ↪'Browns', 'Bengals', 'Texans', 'Colts',
        'Titans', 'Jaguars', 'Chiefs', 'Chargers', 'Broncos', 'Raiders',␣
 ↪'Cowboys', 'Eagles', 'Redskins', 'Giants',
        'Bears', 'Vikings', 'Packers', 'Lions', 'Saints', 'Panthers',␣
 ↪'Falcons', 'Buccaneers', 'Rams', 'Seahawks',
        '49ers', 'Cardinals']

nfl_df_filtered['mascot'] = mascot

combined = pd.merge(nfl_df_filtered, cities_nfl, how = 'left', left_on =␣
 ↪'mascot', right_on = 'NFL')

combined = combined[['City', 'mascot', 'Population', 'W-L%']]

combined[['W-L%']] = combined[['W-L%']].astype(float)

combined_final = combined.groupby(by = 'City').mean('W-L%')

combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
 ↪'City', right_on = 'City')

combined_final = combined_final[['City', 'W-L%_x', 'Population']]

combined_final.columns = ['City', 'win_loss_ratio', 'Population']
```

```
combined_final.Population = combined_final.Population.astype(float)

combined_final = combined_final.drop_duplicates(subset = ['City'])

population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
 ↪metropolitan area population from cities
win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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), "Q4: Your lists␣
 ↪must be the same length"
# assert len(population_by_region) == 29, "Q4: There should be 29 teams being␣
 ↪analysed for NFL"

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

[311]: 0.004282141436393022

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.

[316]:
```
# 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
    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]]

    cities.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

    cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
    cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
    cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
    cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)


    ##########################################################################


    cities_nhl = cities[['City', 'Population', 'NHL']]

    cities_nhl_split = cities_nhl.iloc[0:2]

    cities_nhl_split['NHL'] = cities_nhl_split['NHL'].str.split(' ')

    cities_nhl_split = cities_nhl_split.explode('NHL')

    cities_nhl = pd.concat([cities_nhl_split, cities_nhl.iloc[2:]])

    nhl_df = nhl_df[nhl_df["year"] == 2018]

    substring = 'Division'
    filter = nhl_df['team'].str.contains(substring)
    nhl_df_filtered = nhl_df[~filter]

    nhl_df_filtered.W = nhl_df_filtered.W.astype(float)

    nhl_df_filtered.L = nhl_df_filtered.L.astype(float)

    nhl_df_filtered["win_loss_ratio"] = nhl_df_filtered.W / (nhl_df_filtered.W␣
↪+ nhl_df_filtered.L)

    nhl_df_filtered["team"] = nhl_df_filtered["team"].str.replace(r'\*', '',␣
↪regex = True)

    mascot = ['Lightning', 'Bruins', 'Maple Leafs', 'Panthers', 'Red Wings',␣
↪'Canadiens', 'Senators', 'Sabres', 'Capitals', 'Penguins',
```

```
          'Flyers', 'Blue Jackets', 'Devils', 'Hurricanes', 'Islanders',␣
↪'Rangers', 'Predators', 'Jets', 'Wild', 'Avalanche', 'Blues',
          'Stars', 'Blackhawks', 'Golden Knights', 'Ducks', 'Sharks',␣
↪'Kings', 'Flames', 'Oilers', 'Canucks', 'Coyotes']

   nhl_df_filtered['mascot'] = mascot

   combined = pd.merge(nhl_df_filtered, cities_nhl, how = 'left', left_on =␣
↪'mascot', right_on = 'NHL')

   combined = combined[['City', 'mascot', 'Population', 'win_loss_ratio']]

   combined_final = combined.groupby(by = 'City').mean('win_loss_ratio')

   combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
↪'City', right_on = 'City')

   combined_final = combined_final[['City', 'win_loss_ratio_x', 'Population']]

   combined_final.columns = ['City', 'win_loss_ratio', 'Population']

   combined_final.Population = combined_final.Population.astype(float)

   combined_final = combined_final.drop_duplicates(subset = ['City'])

   nhl_final = combined_final[['City', 'win_loss_ratio']]

   nhl_final.columns = ['City', 'nhl_win_loss_ratio']

   ######################################################################

   cities_nba = cities[['City', 'Population', 'NBA']]

   cities_nba_split = cities_nba.iloc[0:2]

   cities_nba_split['NBA'] = cities_nba_split['NBA'].str.split(' ')

   cities_nba_split = cities_nba_split.explode('NBA')

   cities_nba = pd.concat([cities_nba_split, cities_nba.iloc[2:]])

   nba_df_filtered = nba_df[nba_df["year"] == 2018]

   nba_df_filtered["team"] = nba_df_filtered["team"].str.replace(r'\*', '',␣
↪regex = True)
```

```
   nba_df_filtered["team"] = nba_df_filtered["team"].str.replace(r'\(.*\)',␣
↪'', regex = True)

   # nba_df_filtered

   # mascot = ['Lightning', 'Bruins', 'Maple Leafs', 'Panthers', 'Red Wings',␣
↪'Canadiens', 'Senators', 'Sabres', 'Capitals', 'Penguins',
   #          'Flyers', 'Blue Jackets', 'Devils', 'Hurricanes', 'Islanders',␣
↪'Rangers', 'Predators', 'Jets', 'Wild', 'Avalanche', 'Blues',
   #          'Stars', 'Blackhawks', 'Golden Knights', 'Ducks', 'Sharks',␣
↪'Kings', 'Flames', 'Oilers', 'Canucks', 'Coyotes']

   # nhl_df_filtered['mascot'] = mascot

   mascot = ['Raptors', 'Celtics', '76ers', 'Cavaliers', 'Pacers', 'Heat',␣
↪'Bucks', 'Wizards', 'Pistons', 'Hornets', 'Knicks',
             'Nets', 'Bulls', 'Magic', 'Hawks', 'Rockets', 'Warriors', 'Trail␣
↪Blazers', 'Thunder', 'Jazz', 'Pelicans', 'Spurs',
             'Timberwolves', 'Nuggets', 'Clippers', 'Lakers', 'Kings',␣
↪'Mavericks', 'Grizzlies', 'Suns']

   nba_df_filtered['mascot'] = mascot

   combined = pd.merge(nba_df_filtered, cities_nba, how = 'left', left_on =␣
↪'mascot', right_on = 'NBA')

   combined = combined[['City', 'mascot', 'Population', 'W/L%']]

   combined[['W/L%']] = combined[['W/L%']].astype(float)

   combined_final = combined.groupby(by = 'City').mean('W/L%')

   combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
↪'City', right_on = 'City')

   combined_final = combined_final[['City', 'W/L%_x', 'Population']]

   combined_final.columns = ['City', 'win_loss_ratio', 'Population']

   combined_final.Population = combined_final.Population.astype(float)

   combined_final = combined_final.drop_duplicates(subset = ['City'])

   nba_final = combined_final[['City', 'win_loss_ratio']]

   nba_final.columns = ['City', 'nba_win_loss_ratio']
```

```
   ######################################################################


   cities_mlb = cities[['City', 'Population', 'MLB']]

   cities_mlb_split = cities_mlb.iloc[0:4]

   cities_mlb_split['MLB'] = cities_mlb_split['MLB'].str.split(' ')

   cities_mlb_split = cities_mlb_split.explode('MLB')

   cities_mlb = pd.concat([cities_mlb_split, cities_mlb.iloc[4:]])

   cities_mlb_split_white_sox = cities_mlb.iloc[7:9, ]

   cities_mlb_split_white_sox = cities_mlb_split_white_sox.groupby(['City',␣
↪'Population'])['MLB'].apply(' '.join).reset_index()

   cities_mlb = pd.concat([cities_mlb.iloc[0:7], cities_mlb_split_white_sox,␣
↪cities_mlb.iloc[9:]])

   mlb_df_filtered =  mlb_df[mlb_df["year"] == 2018]

   mlb_df_filtered

   mascot = ['Red Sox', 'Yankees', 'Rays', 'Blue Jays', 'Orioles', 'Indians',␣
↪'Twins', 'Tigers', 'White Sox', 'Royals', 'Astros',
             'Athletics', 'Mariners', 'Angels', 'Rangers', 'Braves',␣
↪'Nationals', 'Phillies', 'Mets', 'Marlins', 'Brewers',
             'Cubs', 'Cardinals', 'Pirates', 'Reds', 'Dodgers', 'Rockies',␣
↪'Diamondbacks', 'Giants', 'Padres']

   mlb_df_filtered['mascot'] = mascot

   combined = pd.merge(mlb_df_filtered, cities_mlb, how = 'left', left_on =␣
↪'mascot', right_on = 'MLB')

   combined = combined[['City', 'mascot', 'Population', 'W-L%']]

   combined[['W-L%']] = combined[['W-L%']].astype(float)

   combined_final = combined.groupby(by = 'City').mean('W-L%')
```

```
   combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
↪'City', right_on = 'City')

   combined_final = combined_final[['City', 'W-L%_x', 'Population']]

   combined_final.columns = ['City', 'win_loss_ratio', 'Population']

   combined_final.Population = combined_final.Population.astype(float)

   combined_final = combined_final.drop_duplicates(subset = ['City'])

   mlb_final = combined_final[['City', 'win_loss_ratio']]

   mlb_final.columns = ['City', 'mlb_win_loss_ratio']


   ######################################################################

   cities_nfl = cities[['City', 'Population', 'NFL']]

   cities_nfl_split = cities_nfl.iloc[0:3]

   cities_nfl_split['NFL'] = cities_nfl_split['NFL'].str.split(' ')

   cities_nfl_split = cities_nfl_split.explode('NFL')

   cities_nfl = pd.concat([cities_nfl_split, cities_nfl.iloc[3:]])

   nfl_df =  nfl_df[nfl_df["year"] == 2018]

   substring_1 = 'AFC'
   substring_2 = 'NFC'
   nfl_df['team'] = nfl_df['team'].astype(str)
   filter_1 = nfl_df['team'].str.contains(substring_1)
   nfl_df_filtered = nfl_df[~filter_1]
   filter_2 = nfl_df_filtered['team'].str.contains(substring_2)
   nfl_df_filtered = nfl_df_filtered[~filter_2]

   nfl_df_filtered["team"] = nfl_df_filtered["team"].str.replace(r'\*', '',␣
↪regex = True)
   nfl_df_filtered["team"] = nfl_df_filtered["team"].str.replace(r'\+', '',␣
↪regex = True)

   mascot = ['Patriots', 'Dolphins', 'Bills', 'Jets', 'Ravens', 'Steelers',␣
↪'Browns', 'Bengals', 'Texans', 'Colts',
             'Titans', 'Jaguars', 'Chiefs', 'Chargers', 'Broncos', 'Raiders',␣
↪'Cowboys', 'Eagles', 'Redskins', 'Giants',
```

```
            'Bears', 'Vikings', 'Packers', 'Lions', 'Saints', 'Panthers',␣
↪'Falcons', 'Buccaneers', 'Rams', 'Seahawks',
            '49ers', 'Cardinals']

   nfl_df_filtered['mascot'] = mascot

   combined = pd.merge(nfl_df_filtered, cities_nfl, how = 'left', left_on =␣
↪'mascot', right_on = 'NFL')

   combined = combined[['City', 'mascot', 'Population', 'W-L%']]

   combined[['W-L%']] = combined[['W-L%']].astype(float)

   combined_final = combined.groupby(by = 'City').mean('W-L%')

   combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
↪'City', right_on = 'City')

   combined_final = combined_final[['City', 'W-L%_x', 'Population']]

   combined_final.columns = ['City', 'win_loss_ratio', 'Population']

   combined_final.Population = combined_final.Population.astype(float)

   combined_final = combined_final.drop_duplicates(subset = ['City'])

   nfl_final = combined_final[['City', 'win_loss_ratio']]

   nfl_final.columns = ['City', 'nfl_win_loss_ratio']


   ######################################################################


   nhl_mlb_test = pd.merge(nhl_final, mlb_final, how = 'inner', on = 'City')

   nhl_mlb_test_pvalue = stats.ttest_rel(nhl_mlb_test['nhl_win_loss_ratio'],␣
↪nhl_mlb_test['mlb_win_loss_ratio'])[1]


   nhl_nba_test = pd.merge(nhl_final, nba_final, how = 'inner', on = 'City')

   nhl_nba_test_pvalue = stats.ttest_rel(nhl_nba_test['nhl_win_loss_ratio'],␣
↪nhl_nba_test['nba_win_loss_ratio'])[1]

   nhl_nfl_test = pd.merge(nhl_final, nfl_final, how = 'inner', on = 'City')
```

```
    nhl_nfl_test_pvalue = stats.ttest_rel(nhl_nfl_test['nhl_win_loss_ratio'],␣
↪nhl_nfl_test['nfl_win_loss_ratio'])[1]

    mlb_nba_test = pd.merge(mlb_final, nba_final, how = 'inner', on = 'City')

    mlb_nba_test_pvalue = stats.ttest_rel(mlb_nba_test['mlb_win_loss_ratio'],␣
↪mlb_nba_test['nba_win_loss_ratio'])[1]

    mlb_nfl_test = pd.merge(mlb_final, nfl_final, how = 'inner', on = 'City')

    mlb_nfl_test_pvalue = stats.ttest_rel(mlb_nfl_test['mlb_win_loss_ratio'],␣
↪mlb_nfl_test['nfl_win_loss_ratio'])[1]

    nba_nfl_test = pd.merge(nba_final, nfl_final, how = 'inner', on = 'City')

    nba_nfl_test_pvalue = stats.ttest_rel(nba_nfl_test['nba_win_loss_ratio'],␣
↪nba_nfl_test['nfl_win_loss_ratio'])[1]


    ###########################################################################

    # 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)

    p_values.loc['NHL', 'MLB'] = nhl_mlb_test_pvalue
    p_values.loc['MLB', 'NHL'] = nhl_mlb_test_pvalue

    p_values.loc['NHL', 'NBA'] = nhl_nba_test_pvalue
    p_values.loc['NBA', 'NHL'] = nhl_nba_test_pvalue

    p_values.loc['NHL', 'NFL'] = nhl_nfl_test_pvalue
    p_values.loc['NFL', 'NHL'] = nhl_nfl_test_pvalue

    p_values.loc['MLB', 'NBA'] = mlb_nba_test_pvalue
    p_values.loc['NBA', 'MLB'] = mlb_nba_test_pvalue

    p_values.loc['MLB', 'NFL'] = mlb_nfl_test_pvalue
    p_values.loc['NFL', 'MLB'] = mlb_nfl_test_pvalue

    p_values.loc['NBA', 'NFL'] = nba_nfl_test_pvalue
    p_values.loc['NFL', 'NBA'] = nba_nfl_test_pvalue
```

```
    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

    raise NotImplementedError()
```

[ ]:

[317]:
```
sports_team_performance()
```

[317]:

	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

[ ]:
```
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]]

cities.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

cities_nhl = cities[['City', 'Population', 'NHL']]

cities_nhl_split = cities_nhl.iloc[0:2]

cities_nhl_split['NHL'] = cities_nhl_split['NHL'].str.split(' ')

cities_nhl_split = cities_nhl_split.explode('NHL')

cities_nhl = pd.concat([cities_nhl_split, cities_nhl.iloc[2:]])

nhl_df = nhl_df[nhl_df["year"] == 2018]
```

```
substring = 'Division'
filter = nhl_df['team'].str.contains(substring)
nhl_df_filtered = nhl_df[~filter]

nhl_df_filtered.W = nhl_df_filtered.W.astype(float)

nhl_df_filtered.L = nhl_df_filtered.L.astype(float)

nhl_df_filtered["win_loss_ratio"] = nhl_df_filtered.W / (nhl_df_filtered.W +␣
 ↪nhl_df_filtered.L)

nhl_df_filtered["team"] = nhl_df_filtered["team"].str.replace(r'\*', '', regex␣
 ↪= True)

mascot = ['Lightning', 'Bruins', 'Maple Leafs', 'Panthers', 'Red Wings',␣
 ↪'Canadiens', 'Senators', 'Sabres', 'Capitals', 'Penguins',
       'Flyers', 'Blue Jackets', 'Devils', 'Hurricanes', 'Islanders',␣
 ↪'Rangers', 'Predators', 'Jets', 'Wild', 'Avalanche', 'Blues',
       'Stars', 'Blackhawks', 'Golden Knights', 'Ducks', 'Sharks', 'Kings',␣
 ↪'Flames', 'Oilers', 'Canucks', 'Coyotes']

nhl_df_filtered['mascot'] = mascot

combined = pd.merge(nhl_df_filtered, cities_nhl, how = 'left', left_on =␣
 ↪'mascot', right_on = 'NHL')

combined = combined[['City', 'mascot', 'Population', 'win_loss_ratio']]

combined_final = combined.groupby(by = 'City').mean('win_loss_ratio')

combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
 ↪'City', right_on = 'City')

combined_final = combined_final[['City', 'win_loss_ratio_x', 'Population']]

combined_final.columns = ['City', 'win_loss_ratio', 'Population']

combined_final.Population = combined_final.Population.astype(float)

combined_final = combined_final.drop_duplicates(subset = ['City'])

nhl_final = combined_final[['City', 'win_loss_ratio']]

nhl_final.columns = ['City', 'nhl_win_loss_ratio']

nhl_final.head()
```

```
# population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
 ↪metropolitan area population from cities
# win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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"

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

[ ]:
```
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]]

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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

cities_nba = cities[['City', 'Population', 'NBA']]

cities_nba_split = cities_nba.iloc[0:2]

cities_nba_split['NBA'] = cities_nba_split['NBA'].str.split(' ')

cities_nba_split = cities_nba_split.explode('NBA')

cities_nba = pd.concat([cities_nba_split, cities_nba.iloc[2:]])

nba_df_filtered = nba_df[nba_df["year"] == 2018]

nba_df_filtered["team"] = nba_df_filtered["team"].str.replace(r'\*', '', regex␣
 ↪= True)
```

```
nba_df_filtered["team"] = nba_df_filtered["team"].str.replace(r'\(.*\)', '',␣
 ↪regex = True)

# nba_df_filtered

# mascot = ['Lightning', 'Bruins', 'Maple Leafs', 'Panthers', 'Red Wings',␣
 ↪'Canadiens', 'Senators', 'Sabres', 'Capitals', 'Penguins',
#         'Flyers', 'Blue Jackets', 'Devils', 'Hurricanes', 'Islanders',␣
 ↪'Rangers', 'Predators', 'Jets', 'Wild', 'Avalanche', 'Blues',
#         'Stars', 'Blackhawks', 'Golden Knights', 'Ducks', 'Sharks', 'Kings',␣
 ↪'Flames', 'Oilers', 'Canucks', 'Coyotes']

# nhl_df_filtered['mascot'] = mascot

mascot = ['Raptors', 'Celtics', '76ers', 'Cavaliers', 'Pacers', 'Heat',␣
 ↪'Bucks', 'Wizards', 'Pistons', 'Hornets', 'Knicks',
        'Nets', 'Bulls', 'Magic', 'Hawks', 'Rockets', 'Warriors', 'Trail␣
 ↪Blazers', 'Thunder', 'Jazz', 'Pelicans', 'Spurs',
        'Timberwolves', 'Nuggets', 'Clippers', 'Lakers', 'Kings', 'Mavericks',␣
 ↪'Grizzlies', 'Suns']

nba_df_filtered['mascot'] = mascot

combined = pd.merge(nba_df_filtered, cities_nba, how = 'left', left_on =␣
 ↪'mascot', right_on = 'NBA')

combined = combined[['City', 'mascot', 'Population', 'W/L%']]

combined[['W/L%']] = combined[['W/L%']].astype(float)

combined_final = combined.groupby(by = 'City').mean('W/L%')

combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
 ↪'City', right_on = 'City')

combined_final = combined_final[['City', 'W/L%_x', 'Population']]

combined_final.columns = ['City', 'win_loss_ratio', 'Population']

combined_final.Population = combined_final.Population.astype(float)

combined_final = combined_final.drop_duplicates(subset = ['City'])

nba_final = combined_final[['City', 'win_loss_ratio']]
```

```
nba_final.columns = ['City', 'nba_win_loss_ratio']

nba_final.head()

# population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
 ↪metropolitan area population from cities
# win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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 NBA"

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

[ ]:
```
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]]

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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

cities_mlb = cities[['City', 'Population', 'MLB']]

cities_mlb_split = cities_mlb.iloc[0:4]

cities_mlb_split['MLB'] = cities_mlb_split['MLB'].str.split(' ')

cities_mlb_split = cities_mlb_split.explode('MLB')

cities_mlb = pd.concat([cities_mlb_split, cities_mlb.iloc[4:]])

cities_mlb_split_white_sox = cities_mlb.iloc[7:9, ]
```

```
cities_mlb_split_white_sox = cities_mlb_split_white_sox.groupby(['City',␣
 ↪'Population'])['MLB'].apply(' '.join).reset_index()

cities_mlb = pd.concat([cities_mlb.iloc[0:7], cities_mlb_split_white_sox,␣
 ↪cities_mlb.iloc[9:]])

mlb_df_filtered =  mlb_df[mlb_df["year"] == 2018]

mlb_df_filtered

mascot = ['Red Sox', 'Yankees', 'Rays', 'Blue Jays', 'Orioles', 'Indians',␣
 ↪'Twins', 'Tigers', 'White Sox', 'Royals', 'Astros',
        'Athletics', 'Mariners', 'Angels', 'Rangers', 'Braves', 'Nationals',␣
 ↪'Phillies', 'Mets', 'Marlins', 'Brewers',
        'Cubs', 'Cardinals', 'Pirates', 'Reds', 'Dodgers', 'Rockies',␣
 ↪'Diamondbacks', 'Giants', 'Padres']

mlb_df_filtered['mascot'] = mascot

combined = pd.merge(mlb_df_filtered, cities_mlb, how = 'left', left_on =␣
 ↪'mascot', right_on = 'MLB')

combined = combined[['City', 'mascot', 'Population', 'W-L%']]

combined[['W-L%']] = combined[['W-L%']].astype(float)

combined_final = combined.groupby(by = 'City').mean('W-L%')

combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
 ↪'City', right_on = 'City')

combined_final = combined_final[['City', 'W-L%_x', 'Population']]

combined_final.columns = ['City', 'win_loss_ratio', 'Population']

combined_final.Population = combined_final.Population.astype(float)

combined_final = combined_final.drop_duplicates(subset = ['City'])

mlb_final = combined_final[['City', 'win_loss_ratio']]

mlb_final.columns = ['City', 'mlb_win_loss_ratio']

mlb_final.head()
```

```
# population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
 ↪metropolitan area population from cities
# win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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), "Q3: Your lists␣
 ↪must be the same length"
# assert len(population_by_region) == 26, "Q3: There should be 26 teams being␣
 ↪analysed for MLB"

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

[ ]:
```
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]]

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.columns = ['City', 'Population', 'NFL', 'MLB', 'NBA', 'NHL']

cities['NFL'] = cities['NFL'].str.replace(r'\[.*\]','', regex = True)
cities['MLB'] = cities['MLB'].str.replace(r'\[.*\]','', regex = True)
cities['NBA'] = cities['NBA'].str.replace(r'\[.*\]','', regex = True)
cities['NHL'] = cities['NHL'].str.replace(r'\[.*\]','', regex = True)

cities_nfl = cities[['City', 'Population', 'NFL']]

cities_nfl_split = cities_nfl.iloc[0:3]

cities_nfl_split['NFL'] = cities_nfl_split['NFL'].str.split(' ')

cities_nfl_split = cities_nfl_split.explode('NFL')

cities_nfl = pd.concat([cities_nfl_split, cities_nfl.iloc[3:]])

nfl_df =  nfl_df[nfl_df["year"] == 2018]

substring_1 = 'AFC'
substring_2 = 'NFC'
nfl_df['team'] = nfl_df['team'].astype(str)
```

```
filter_1 = nfl_df['team'].str.contains(substring_1)
nfl_df_filtered = nfl_df[~filter_1]
filter_2 = nfl_df_filtered['team'].str.contains(substring_2)
nfl_df_filtered = nfl_df_filtered[~filter_2]

nfl_df_filtered["team"] = nfl_df_filtered["team"].str.replace(r'\*', '', regex␣
 ↪= True)
nfl_df_filtered["team"] = nfl_df_filtered["team"].str.replace(r'\+', '', regex␣
 ↪= True)

mascot = ['Patriots', 'Dolphins', 'Bills', 'Jets', 'Ravens', 'Steelers',␣
 ↪'Browns', 'Bengals', 'Texans', 'Colts',
        'Titans', 'Jaguars', 'Chiefs', 'Chargers', 'Broncos', 'Raiders',␣
 ↪'Cowboys', 'Eagles', 'Redskins', 'Giants',
        'Bears', 'Vikings', 'Packers', 'Lions', 'Saints', 'Panthers',␣
 ↪'Falcons', 'Buccaneers', 'Rams', 'Seahawks',
        '49ers', 'Cardinals']

nfl_df_filtered['mascot'] = mascot

combined = pd.merge(nfl_df_filtered, cities_nfl, how = 'left', left_on =␣
 ↪'mascot', right_on = 'NFL')

combined = combined[['City', 'mascot', 'Population', 'W-L%']]

combined[['W-L%']] = combined[['W-L%']].astype(float)

combined_final = combined.groupby(by = 'City').mean('W-L%')

combined_final = pd.merge(combined_final, combined, how = 'left', left_on =␣
 ↪'City', right_on = 'City')

combined_final = combined_final[['City', 'W-L%_x', 'Population']]

combined_final.columns = ['City', 'win_loss_ratio', 'Population']

combined_final.Population = combined_final.Population.astype(float)

combined_final = combined_final.drop_duplicates(subset = ['City'])

nfl_final = combined_final[['City', 'win_loss_ratio']]

nfl_final.columns = ['City', 'nfl_win_loss_ratio']

nfl_final.head()
```

```
# population_by_region = combined_final[['Population']].iloc[:, 0] # pass in␣
 ↪metropolitan area population from cities
# win_loss_by_region = combined_final[['win_loss_ratio']].iloc[:, 0] # 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), "Q4: Your lists␣
 ↪must be the same length"
# assert len(population_by_region) == 29, "Q4: There should be 29 teams being␣
 ↪analysed for NFL"

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

[ ]:
```
# win_loss_consolidated = pd.merge(nhl_final, mlb_final, how = 'outer', on =␣
 ↪'City')

# win_loss_consolidated = pd.merge(win_loss_consolidated, nba_final, how =␣
 ↪'outer', on = 'City')

# win_loss_consolidated = pd.merge(win_loss_consolidated, nfl_final, how =␣
 ↪'outer', on = 'City')

# win_loss_consolidated
```

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

nhl_mlb_test = pd.merge(nhl_final, mlb_final, how = 'inner', on = 'City')

nhl_mlb_test_pvalue = stats.ttest_rel(nhl_mlb_test['nhl_win_loss_ratio'],␣
 ↪nhl_mlb_test['mlb_win_loss_ratio'])[1]
```

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

nhl_nba_test = pd.merge(nhl_final, nba_final, how = 'inner', on = 'City')

nhl_nba_test_pvalue = stats.ttest_rel(nhl_nba_test['nhl_win_loss_ratio'],␣
 ↪nhl_nba_test['nba_win_loss_ratio'])[1]
```

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

```
import re

nhl_nfl_test = pd.merge(nhl_final, nfl_final, how = 'inner', on = 'City')

nhl_nfl_test_pvalue = stats.ttest_rel(nhl_nfl_test['nhl_win_loss_ratio'],␣
 ↪nhl_nfl_test['nfl_win_loss_ratio'])[1]
```

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

mlb_nba_test = pd.merge(mlb_final, nba_final, how = 'inner', on = 'City')

mlb_nba_test_pvalue = stats.ttest_rel(mlb_nba_test['mlb_win_loss_ratio'],␣
 ↪mlb_nba_test['nba_win_loss_ratio'])[1]
```

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

mlb_nfl_test = pd.merge(mlb_final, nfl_final, how = 'inner', on = 'City')

mlb_nfl_test_pvalue = stats.ttest_rel(mlb_nfl_test['mlb_win_loss_ratio'],␣
 ↪mlb_nfl_test['nfl_win_loss_ratio'])[1]
```

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

nba_nfl_test = pd.merge(nba_final, nfl_final, how = 'inner', on = 'City')

nba_nfl_test_pvalue = stats.ttest_rel(nba_nfl_test['nba_win_loss_ratio'],␣
 ↪nba_nfl_test['nfl_win_loss_ratio'])[1]
```

[ ]:
```
p_values
```

[ ]:
```
p_values.loc['NHL', 'MLB'] = nhl_mlb_test_pvalue
p_values.loc['MLB', 'NHL'] = nhl_mlb_test_pvalue
```

[ ]:
```
p_values.loc['NHL', 'NBA'] = nhl_nba_test_pvalue
p_values.loc['NBA', 'NHL'] = nhl_nba_test_pvalue
```

[ ]:
```
p_values.loc['NHL', 'NFL'] = nhl_nfl_test_pvalue
p_values.loc['NFL', 'NHL'] = nhl_nfl_test_pvalue
```

[ ]:
```
p_values.loc['MLB', 'NBA'] = mlb_nba_test_pvalue
p_values.loc['NBA', 'MLB'] = mlb_nba_test_pvalue
```

[ ]:
```
p_values.loc['MLB', 'NFL'] = mlb_nfl_test_pvalue
p_values.loc['NFL', 'MLB'] = mlb_nfl_test_pvalue
```

[ ]:
```
p_values.loc['NBA', 'NFL'] = nba_nfl_test_pvalue
p_values.loc['NFL', 'NBA'] = nba_nfl_test_pvalue
```

[ ]:
```
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"
```

[ ]:
```
p_values
```

[ ]:
```

```