Giorgio Mendoza

CS539-F23-F02

Dr. J. Sethi

Lab_2-0: Pandas Introduction

Please answer the questions by filling in the code where indicated below.

Part 1

The following code loads the olympics dataset (olympics.csv), which was derrived from the Wikipedia entry on All Time Olympic Games Medals, and does some basic data cleaning.

The columns are organized as # of Summer games, Summer medals, # of Winter games, Winter medals, total # number of games, total # of medals. Use this dataset to answer the questions below.

```
import pandas as pd

from google.colab import drive
drive.mount('/content/drive')
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/olympics.csv', index_col=0, skiprows=1)

for col in df.columns:
    if col[:2]=='01':
        df.rename(columns={col:'Gold'+col[4:]}, inplace=True)
    if col[:2]=='02':
        df.rename(columns={col:'Silver'+col[4:]}, inplace=True)
    if col[:2]=='03':
        df.rename(columns={col:'Bronze'+col[4:]}, inplace=True)
    if col[:1]=='№':
        df.rename(columns={col:'#'+col[1:]}, inplace=True)

names_ids = df.index.str.split('\s\(') # split the index by '('

df.index = names_ids.str[0] # the [0] element is the country name (new index)
df['ID'] = names_ids.str[1].str[:3] # the [1] element is the abbreviation or ID (take first 3 characters from that)

df = df.drop('Totals')
df.head()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call driv
```

	# Summer	Gold	Silver	Bronze	Total	# Winter	Gold.1	Silver.1	Bronze.
Afghanistan	13	0	0	2	2	0	0	0	
Algeria	12	5	2	8	15	3	0	0	
Argentina	23	18	24	28	70	18	0	0	
Armenia	5	1	2	9	12	6	0	0	
Australasia	2	3	4	5	12	0	0	0	

Question 0 (Example)

What is the first country in df?

This function should return a Series.

```
def answer_zero():
    #return country at position 0
    return df.iloc[0]
answer_zero()
```

```
# Summer    13
Gold         0
```

```
Silver             0
Bronze             2
Total              2
# Winter           0
Gold.1             0
Silver.1           0
Bronze.1           0
Total.1            0
# Games            13
Gold.2             0
Silver.2           0
Bronze.2           2
Combined total     2
ID               AFG
Name: Afghanistan, dtype: object
```

Question 1

Which country has won the most gold medals in summer games?

This function should return a single string value.

```
def answer_one():
    max_gold_country = df['Gold'].idxmax()  # Find the index (country name) with the maximum gold medals
    return max_gold_country

result = answer_one()
print(result)
```

```
 United States
```

Question 2

Which country had the biggest difference between their summer and winter gold medal counts?

Answer:

United States

This function should return a single string value.

```
import pandas as pd

def answer_two():
    # get absolute difference between summer & winter gold medal counts
    medal_difference = abs(df['Gold'] - df['Gold.1'])

    # find country with maximum difference
    max_difference_country = medal_difference.idxmax()

    return max_difference_country

result = answer_two()
print(result)
```

```
 United States
```

Question 3

Which country has the biggest difference between their summer gold medal counts and winter gold medal counts relative to their total gold medal count?

Answer:

Bulgaria

$$\frac{Summer\ Gold - Winter\ Gold}{Total\ Gold}$$

Only include countries that have won at least 1 gold in both summer and winter.

This function should return a single string value.

```
import pandas as pd

def answer_three():
    new_df = df[(df['Gold'] > 0) & (df['Gold.1'] > 0)] # filter countries that have won 1 gold in both summer & winter

    # get relative difference between summer & winter gold medal counts
    relative_difference = abs(new_df['Gold'] - new_df['Gold.1']) / new_df['Gold.2']

    # get country with maximum relative difference
    max_difference_country = relative_difference.idxmax()

    return max_difference_country

result = answer_three()
print(result)
```

```
 Bulgaria
```

Question 4

Write a function that creates a Series called "Points" which is a weighted value where each gold medal (`Gold.2`) counts for 3 points, silver medals (`Silver.2`) for 2 points, and bronze medals (`Bronze.2`) for 1 point. The function should return only the column (a Series object) which you created, with the country names as indices.

This function should return a Series named `Points` of length 146

```
def answer_four():
  #multiply weights by medals
  Total_Gold = df['Gold.2']*3
  Total_Silver = df['Silver.2']*2
  Total_Bronze = df['Bronze.2']*1
  #add up total points
  pd.Points = Total_Gold + Total_Silver + Total_Bronze
  return pd.Points
result = answer_four()
print(result)
```

```
Afghanistan                          2
Algeria                             27
Argentina                          130
Armenia                             16
Australasia                         22
                                   ...
Yugoslavia                         171
Independent Olympic Participants     4
Zambia                               3
Zimbabwe                            18
Mixed team                          38
Length: 146, dtype: int64
```

Part 2

For the next set of questions, we will be using census data from the United States Census Bureau. Counties are political and geographic subdivisions of states in the United States. This dataset contains population data for counties and states in the US from 2010 to 2015. See this document for a description of the variable names.

The census dataset (census.csv) should be loaded as census_df. Answer questions using this as appropriate.

Question 5

Which state has the most counties in it? (hint: consider the sumlevel key carefully! You'll need this for future questions too...)

Answer:

Texas does

This function should return a single string value.

```
import pandas as pd
```

```
from google.colab import drive
drive.mount('/content/drive')
census_df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/census.csv', index_col=0, skiprows=0)

census_df.head()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mour
```

SUMLEV	REGION	DIVISION	STATE	COUNTY	STNAME	CTYNAME	CENSUS2010POP	ESTIMATESBASE2
40	3	6	1	0	Alabama	Alabama	4779736	478(
50	3	6	1	1	Alabama	Autauga County	54571	54
50	3	6	1	3	Alabama	Baldwin County	182265	182
50	3	6	1	5	Alabama	Barbour County	27457	27
						Bibb		

```
import pandas as pd

census_df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/census.csv')

def answer_five():
    # get rows with sumlevel=50 (county-level data)
    county_df = census_df[census_df['SUMLEV'] == 50]

    # group by state and count unique county names within each state
    state_counts = county_df['STNAME'].value_counts()

    # get state with maximum county count
    max_county_state = state_counts.idxmax()

    return max_county_state

result = answer_five()
print(result)
```

```
Texas
```

Question 6

Only looking at the three most populous counties for each state, what are the three most populous states (in order of highest population to lowest population)? Use `CENSUS2010POP`.

Answer:

['California', 'Texas', 'Illinois']

This function should return a list of string values.

```
def answer_six():
    # group by state and sort counties by population
    sorted_counties = census_df[census_df['SUMLEV'] == 50].groupby('STNAME')['CENSUS2010POP'].nlargest(3).reset_index()

    # sum populations for top three counties in each state
    summed_populations = sorted_counties.groupby('STNAME')['CENSUS2010POP'].sum().reset_index()

    # sort states based on summed population
    sorted_states = summed_populations.sort_values(by='CENSUS2010POP', ascending=False)

    # extract names of top three states
    top_states = sorted_states['STNAME'].head(3).tolist()

    return top_states

result = answer_six()
print(result)
```

```
['California', 'Texas', 'Illinois']
```

Question 7

Which county has had the largest absolute change in population within the period 2010-2015? (Hint: population values are stored in columns POPESTIMATE2010 through POPESTIMATE2015, you need to consider all six columns.)

Answer:

Texas county

e.g. If County Population in the 5 year period is 100, 120, 80, 105, 100, 130, then its largest change in the period would be |130-80| = 50.

This function should return a single string value.

```
def answer_seven():
    # get relevant columns
    population_columns = ['POPESTIMATE2010', 'POPESTIMATE2011', 'POPESTIMATE2012', 'POPESTIMATE2013', 'POPESTIMATE2014', 'POPESTIMATE2015']
    county_population = census_df[['CTYNAME'] + population_columns].copy()  # Make a copy to avoid the SettingWithCopyWarning

    # get absolute change in population
    county_population['POP_CHANGE'] = county_population[population_columns].max(axis=1) - county_population[population_columns].min(axis=1)

    # get county with largest absolute change in population
    largest_change_county = county_population.loc[county_population['POP_CHANGE'].idxmax(), 'CTYNAME']

    return largest_change_county

result = answer_seven()
print(result)
```

```
 Texas
```

Question 8

In this datafile, the United States is broken up into four regions using the "REGION" column.

Create a query that finds the counties that belong to regions 1 or 2, whose name starts with 'Washington', and whose POPESTIMATE2015 was greater than their POPESTIMATE 2014.

This function should return a 5x2 DataFrame with the columns = ['STNAME', 'CTYNAME'] and the same index ID as the census_df (sorted ascending by index).

```
def answer_eight():
    # get dataframe based on conditions given
    filtered_counties = census_df[(census_df['REGION'].isin([1, 2])) &
                                  (census_df['CTYNAME'].str.startswith('Washington')) &
                                  (census_df['POPESTIMATE2015'] > census_df['POPESTIMATE2014'])]

    # desired columns
    result = filtered_counties[['STNAME', 'CTYNAME']]

    return result

result = answer_eight()
print(result)
```

```
              STNAME            CTYNAME
896             Iowa  Washington County
1419       Minnesota  Washington County
2345    Pennsylvania  Washington County
2355    Rhode Island  Washington County
3163       Wisconsin  Washington County
```