Laboratory Work 5

DataFrames Merging, Data Aggregation and Data Visualization

Goal: Learning Pandas methods for data merging and aggregation.

2. Tasks:

> Notes: In this lab you should use Pandas aggregation functions, `loc`, `iloc` attributes, slicing and DO NOT use list comprehensions in any tasks

1. Load the energy data from the file "En_In.xls", which is a list of indicators of energy supply and renewable electricity production, and put into a DataFrame.

Keep in mind that this is an Excel file, and not a comma separated values file. Also, make sure to exclude the footer and header information from the datafile. The first two columns are unneccessary, so you should get rid of them, and you should change the column labels so that the columns are:

`['Country', 'Energy Supply', 'Energy Supply per Capita', '% Renewable]`

2. Convert '`Energy Supply`' to gigajoules (*Note: there are 1,000,000 gigajoules in a petajoule*). For all countries which have missing data (e.g. data with "...") make sure this is reflected as `np.NaN` values.

3. Rename the following list of countries:
`"Republic of Korea": "South Korea",`
`"United States of America": "United States", "`
`United Kingdom of Great Britain and Northern Ireland": "United Kingdom",`
`"China, Hong Kong Special Administrative Region": "Hong Kong"`

4. There are also several countries with numbers and/or parenthesis in their name. Be sure to remove these, e.g. `'Bolivia (Plurinational State of)'` should be '`Bolivia`', '`Switzerland17`' should be `Switzerland`'.

Expected output for tasks 1-4.

```
In [38]: Energy.loc[Energy['Country'].isin(['American Samoa','South Korea', 'Bolivia' ])]
```

Out[38]:

	Country	Energy Supply	Energy Supply per Capita	% Renewable
4	American Samoa	nan	nan	0.641
25	Bolivia	336000000.000	32.000	31.477
165	South Korea	11007000000.000	221.000	2.279

5. Next, load the GDP data from the file “gpd.csv”, which is a csv containing countries’ GDP from 1960 to 2015 from World Bank.

Make sure to skip the header, and rename the following list of countries:

```
"Korea, Rep.": "South Korea",
"Iran, Islamic Rep.": "Iran",
"Hong Kong SAR, China": "Hong Kong"
```

Expected output for task 5 (only 11 columns are shown):

```
In [46]: GPD.head(1)
```

Out[46]:

	Country	Country Code	Indicator Name	Indicator Code	2006	2007	2008	2009	2010	2011	2012	2
0	Aruba	ABW	GDP at market prices (constant 2010 US$)	NY.GDP.MKTP.KD	nan	nan	nan	nan	2467703910.615	nan	nan	

6. Load the Sciamgo Journal and Country Rank data for Energy Engineering and Power Technology from the file “scimagojr.xlsx”, which ranks countries based on their journal contributions in the aforementioned area.

.

7. Join the three datasets from tasks 1-6 into a new dataset (using the intersection of country names).

- Use only the last 10 years (2006-2015) of GDP data and only the top 15 countries by Scimagojr 'Rank' (Rank 1 through 15).

- The index of this DataFrame should be the name of the country, and the columns should be ['Rank', 'Documents', 'Citable documents', 'Citations', 'Self-citations', 'Citations per document', 'H index', 'Energy Supply', 'Energy Supply per Capita', '% Renewable', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015'].

You should obtain a DataFrame with 15 rows and 20 columns.

Expected output for task 7 (only 10columns are shown):

```
In [55]: Result.head(3)
```

Out[55]:

Country	Rank	Documents	Citable documents	Citations	Self-citations	Citations per document	H index	Energy Supply	Energy Supply per Capita
China	1	127050	126767	597237	411683	4.700	138	127191000000.000	93.000
United States	2	96661	94747	792274	265436	8.200	230	90838000000.000	286.000
Japan	3	30504	30287	223024	61554	7.310	134	18984000000.000	149.000

```
In [56]: Result.shape
```

```
Out[56]: (15, 20)
```

Task 8 – 14 should be solved using dataset from task 7.

8. Create a function to define what are the top 15 countries for average GDP over the last 10 years?

This function should return a Series with 15 countries and their average GDP sorted in descending order.

Expected output for task 8

```
In [12]: task_eight()

Out[12]: Country
         United States         15364344302990.000
         China                  6348608932836.100
         Japan                  5542207638235.176
         Germany                3493025339072.848
         France                 2681724635761.589
         United Kingdom         2487906661418.417
         Brazil                 2189794143774.905
         Italy                  2120175089933.776
         India                  1769297396603.860
         Canada                 1660647466307.512
         Russian Federation     1565459478480.661
         Spain                  1418078278145.694
         Australia              1164042729991.427
         South Korea            1106714508244.852
         Iran                    444155754051.095
         Name: avgGDP, dtype: float64
```

9. Create a function to define by how much had the GDP changed over the past 10 year for the country with the 5th largest average GDP?

This function should return a tuple with the country's name and number

Expected output for task 9

```
In [30]: task_nine()

Out[30]: ('France', 153345695364.24023)
```

10. Create a function to define what country has the maximum % Renewable and what is the percentage?

This function should return a tuple with the name of the country and the percentage.

Expected output for task 10

```
In [41]: task_ten()
Out[41]: ('Brazil', 69.64803)
```

11. Create a column that estimates the population using Energy Supply and Energy Supply per capita. What is the sixth most populous country according to this estimate?

This function should return a tuple with the name of the country and the population

Expected output for task 11

```
In [76]: task_eleven()
Out[76]: ('Japan', 127409395.97315437)
```

12. Create a column that estimates the number of citable documents per person. What is the correlation between the number of citable documents per capita and the energy supply per capita? Use the .corr() method, (Pearson's correlation).

This function should return a single number.

Expected output for task 12

```
In [88]: task_twelve()
Out[88]: 0.7940010435442942
```

13. Create a new column with a 1 if the country's % Renewable value is at or above the median for all countries in the top 15, and a 0 if the country's % Renewable value is below the median.

This function should return a series whose index is the country name sorted in ascending order of rank.

Expected output for task 13

```
In [117]: task_thirteen()

Out[117]: Country
          China                1
          United States        0
          Japan                0
          United Kingdom       0
          Russian Federation   1
          Canada               1
          Germany              1
          India                0
          France               1
          South Korea          0
          Italy                1
          Spain                1
          Iran                 0
          Australia            0
          Brazil               1
          dtype: int32
```

14. Use the following dictionary to group the Countries by Continent, then create a DataFrame that displays the sample size (the number of countries in each continent bin), and the sum, mean, and std deviation for the estimated population of each country.

```
ContinentDict  = {'China':'Asia',
                  'United States':'North America',
                  'Japan':'Asia',
                  'United Kingdom':'Europe',
                  'Russian Federation':'Europe',
                  'Canada':'North America',
                  'Germany':'Europe',
                  'India':'Asia',
                  'France':'Europe',
                  'South Korea':'Asia',
                  'Italy':'Europe',
                  'Spain':'Europe',
                  'Iran':'Asia',
                  'Australia':'Australia',
                  'Brazil':'South America'}
```

This function should return a DataFrame with index named Continent ['Asia', 'Australia', 'Europe', 'North America', 'South America'] and columns ['size', 'sum', 'mean', 'std']

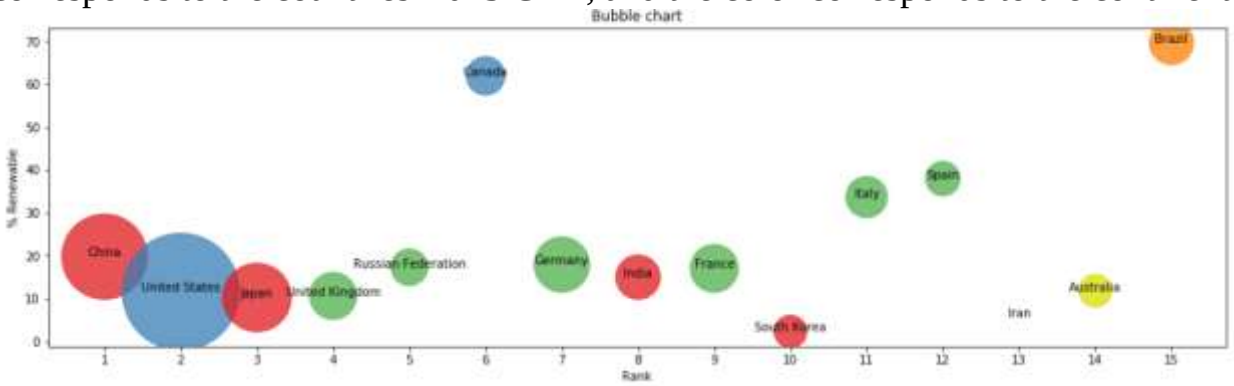
Expected output for task 14

```
In [120]: task_forteen()
```

Out[120]:

Continent	size	sum	mean	std
Asia	5	2898666386.611	579733277.322	679097888.366
Australia	1	23316017.316	23316017.316	nan
Europe	6	457929667.216	76321611.203	34647667.066
North America	2	352855249.480	176427624.740	199669644.857
South America	1	205915254.237	205915254.237	nan

15. Create a bubble chart showing % Renewable vs. Rank. The size of the bubble corresponds to the countries' 2015 GDP, and the color corresponds to the continent



3. The content of the report

1. Cover page of the report.
2. Topic and goal of the lab.
3. Progress of the work.
4. Link to the created Jupyter Notebook on GitHub, rendered by nbviewer.
5. Conclusions.