Python 程序设计
课程实践报告

项目名称：Numpy Pandas 数据分析实践

指导教师：翟建芳

姓　　名：李肇宁

学　　号：SA18225209

报告时间：2018　年　11　月　23　日

1 项目内容简介

本次实验运用了 numpy、pandas 库和 matplotlib 工具对全球各国的近年 GDP 数据、能源供应和可再生电力生产指标以及对能源领域学术期刊的贡献排名三份数据进行数据清洗、分析和可视化操作。数据来源如下。

1960 至今各国 GDP:

http://data.worldbank.org/indicator/NY.GDP.MKTP.CD

各国年度能源供应和可再生电力生产指标：

http://unstats.un.org/unsd/environment/excel_file_tables/2017/Energy%20Indicators.xls

各国对 *scimagojr* 期刊的贡献排名

http://www.scimagor.com/countryrank.php? category=2102

参考用书：

《利用 PYTHON 进行数据分析》书。

2 实践环境

Jupyter python 编辑器

Numpy 库

Pandas 库

Matpltlib 工具

3 实践代码与结果展示

3.1 数据预处理和合并

本部分代码主要负责对三份数据进行清洗和合并。

读取三份数据文件时，需要对其中的内容进行去页眉页脚操作，只取出有用的行列，同时，为了方便三份数据的合并，读取数据时将对列名进行重命名，统一能源单位和并规范国家名称。简要代码如下所示。

```
energy = pd.read_excel('Energy Indicators.xls',skiprows=[x for x in range(17)],skip_footer=38,usecols=[2,3,4,5],names=['Country', 'Energy S
energy.replace('...', np.nan,inplace=True)
energy['Energy Supply'] = energy['Energy Supply']*1000000
energy['Country'] = energy['Country'].str.replace(r"(\s+\(.*\))|(\d*)","")
energy.replace("Republic of Korea","South Korea",inplace=True)
energy.replace("United States of America","United States",inplace=True)
energy.replace("United Kingdom of Great Britain and Northern Ireland","United Kingdom",inplace=True)
energy.replace("China, Hong Kong Special Administrative Region","Hong Kong",inplace=True)

GDP = pd.read_csv('world_bank.csv',skiprows=[0,1,2,3])
GDP.replace("Korea, Rep.","South Korea",inplace=True)
GDP.replace("Iran, Islamic Rep.","Iran",inplace=True)
GDP.replace("Hong Kong SAR, China","Hong Kong",inplace=True)

ScimEn = pd.read_excel('scimagojr.xlsx')
```

三份表格进行数据合并时以国家名为连接键，只保留对能源领域期刊贡献排名前 15 的国家和各国 08 年后的 GDP 数据，合并后的表格以 Country 为索引。

```
mergeScimEnAndEnergy = pd.merge(ScimEn,energy,how='inner',left_on='Country',right_on='Country')
merge2 = pd.merge(mergeScimEnAndEnergy,GDP,how='inner',left_on='Country',right_on='Country Name')
merge2 = merge2[merge2['Rank']<=15]
merge2.drop(labels=[str(x) for x in range(1960,2007)],axis=1,inplace=True)
merge2.drop(labels=['Country Code','Indicator Name','Indicator Code'],axis=1,inplace=True)
merge2.set_index('Country',inplace=True)
merge2 = merge2[['Rank', 'Documents', 'Citable documents', 'Citations', 'Self-citations', 'Citations per document', 'H index', 'Energy Supp
return merge2
```

进行数据清洗和合并得到的表格如图 2.1，2.2 所示。

Out[18]:

	Rank	Documents	Citable documents	Citations	Self-citations	Citations per document	H index	Energy Supply	Energy Supply per Capita	% Renewable	2008	2009
Country												
China	1	167992	167369	1057626	722578	6.30	176	1.271910e+11	93.0	19.754910	4.598206e+12	5.109954e+12
United States	2	126158	123332	1296212	444998	10.27	278	9.083800e+10	286.0	11.570980	1.471858e+13	1.441874e+13
Japan	3	37948	37612	316956	85620	8.35	155	1.898400e+10	149.0	10.232820	5.037908e+12	5.231383e+12
United Kingdom	4	28998	28176	335914	64609	11.58	170	7.920000e+09	124.0	10.600470	2.890564e+12	2.382826e+12
India	5	24872	24300	215787	68150	8.68	141	3.319500e+10	26.0	14.969080	1.186953e+12	1.323940e+12
Germany	6	24407	23963	231800	48820	9.50	151	1.326100e+10	165.0	17.901530	3.752366e+12	3.418005e+12
Russian Federation	7	23361	23196	55495	22485	2.38	69	3.070900e+10	214.0	17.288680	1.660844e+12	1.222644e+12
Canada	8	22908	22465	332093	62436	14.50	177	1.043100e+10	296.0	61.945430	1.549131e+12	1.371153e+12
France	9	17569	17230	203486	43210	11.58	139	1.059700e+10	166.0	17.020280	2.918383e+12	2.690222e+12
South Korea	10	16004	15848	176965	34607	11.06	124	1.100700e+10	221.0	2.279353	1.002219e+12	9.019350e+11
Italy	11	15880	15411	184184	45471	11.60	127	6.530000e+09	109.0	33.667230	2.390729e+12	2.185160e+12
Iran	12	13131	12983	119550	39184	9.10	94	9.172000e+09	119.0	5.707721	4.060709e+11	4.140591e+11
Brazil	13	12705	12559	99136	24398	7.80	103	1.214900e+10	59.0	69.648030	1.695825e+12	1.667020e+12
Australia	14	12635	12397	157471	27620	12.46	131	5.386000e+09	231.0	11.810810	1.052585e+12	9.264482e+11
Spain	15	12478	12270	195663	36353	15.68	138	4.923000e+09	106.0	37.968590	1.635015e+12	1.499100e+12

Out[18]:

2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
4.598206e+12	5.109954e+12	6.100620e+12	7.572554e+12	8.560547e+12	9.607224e+12	1.048237e+13	1.106467e+13	1.119099e+13	1.223770e+13
1.471858e+13	1.441874e+13	1.496437e+13	1.551793e+13	1.615526e+13	1.669152e+13	1.742761e+13	1.812071e+13	1.862448e+13	1.939060e+13
5.037908e+12	5.231383e+12	5.700098e+12	6.157460e+12	6.203213e+12	5.155717e+12	4.850414e+12	4.394978e+12	4.949273e+12	4.872137e+12
2.890564e+12	2.382826e+12	2.441173e+12	2.619700e+12	2.662085e+12	2.739819e+12	3.022828e+12	2.885570e+12	2.650850e+12	2.622434e+12
1.186953e+12	1.323940e+12	1.656617e+12	1.823050e+12	1.827638e+12	1.856722e+12	2.039127e+12	2.102391e+12	2.274230e+12	2.597491e+12
3.752366e+12	3.418005e+12	3.417095e+12	3.757698e+12	3.543984e+12	3.752514e+12	3.890607e+12	3.375611e+12	3.477796e+12	3.677439e+12
1.660844e+12	1.222644e+12	1.524916e+12	2.051662e+12	2.210257e+12	2.297128e+12	2.063663e+12	1.368401e+12	1.284728e+12	1.577524e+12
1.549131e+12	1.371153e+12	1.613464e+12	1.788648e+12	1.824289e+12	1.842628e+12	1.799269e+12	1.559623e+12	1.535768e+12	1.653043e+12
2.918383e+12	2.690222e+12	2.642610e+12	2.861408e+12	2.683825e+12	2.811078e+12	2.852166e+12	2.438208e+12	2.465134e+12	2.582501e+12
1.002219e+12	9.019350e+11	1.094499e+12	1.202464e+12	1.222807e+12	1.305605e+12	1.411334e+12	1.382764e+12	1.414804e+12	1.530751e+12
2.390729e+12	2.185160e+12	2.125058e+12	2.276292e+12	2.072823e+12	2.130491e+12	2.151733e+12	1.832868e+12	1.859384e+12	1.934798e+12
4.060709e+11	4.140591e+11	4.870696e+11	5.835004e+11	5.988534e+11	4.674149e+11	4.344746e+11	3.858745e+11	4.189767e+11	4.395135e+11
1.695825e+12	1.667020e+12	2.208872e+12	2.616202e+12	2.465189e+12	2.472807e+12	2.455994e+12	1.802214e+12	1.793989e+12	2.055506e+12
1.052585e+12	9.264482e+11	1.144261e+12	1.394281e+12	1.543411e+12	1.573697e+12	1.464955e+12	1.349034e+12	1.208039e+12	1.323421e+12
1.635015e+12	1.499100e+12	1.431617e+12	1.488067e+12	1.336019e+12	1.361854e+12	1.376911e+12	1.197790e+12	1.237255e+12	1.311320e+12

3.2 对比 pandas 库的 merge 方法的内、外连接方式

merge 方法的内连接只保留所合并表格相同的列，外连接保留所有列。

```
mergeScimEnAndEnergy = pd.merge(ScimEn,energy,how='outer',left_on='Country',right_on='Country')
merge2a = pd.merge(mergeScimEnAndEnergy,GDP,how='outer',left_on='Country',right_on='Country Name')

mergeScimEnAndEnergy = pd.merge(ScimEn,energy,how='inner',left_on='Country',right_on='Country')
merge2b = pd.merge(mergeScimEnAndEnergy,GDP,how='inner',left_on='Country',right_on='Country Name')
return len(merge2a)-len(merge2b)
```

输出两种方式生成表格的列数差。

```
Out[13]: 157
```

3.3 计算 2008-2017GDP 均值排名

使用第一步生成的表格，将 2008-2017 年列标签循环封装进 rows，利用 Numpy average 方法求平均值，按照国家索引生成一个 series，并进行降序排名输出。

```
Top15 = answer_one()
rows = [str(x) for x in range(2008,2017)]
avg = np.average(Top15[rows],axis=1)
return pd.Series(avg,index=Top15.index,name='avgGDP').sort_values(ascending=False)
```

```
Country
United States         1.629324e+13
China                 8.254126e+12
Japan                 5.297827e+12
Germany               3.598408e+12
France                2.707004e+12
United Kingdom        2.699491e+12
Brazil                2.130901e+12
Italy                 2.113838e+12
India                 1.787852e+12
Russian Federation    1.742694e+12
Canada                1.653775e+12
Spain                 1.395959e+12
Australia             1.295190e+12
South Korea           1.215381e+12
Iran                  4.662549e+11
Name: avgGDP, dtype: float64
```

3.4 为表格添加新列并重新排序

将第三步获得的平均 gdp 的 Series 作为 avg 新列添加到表格中，并将表格按照平均 GDP 重新排序，利用 iloc 方法取出中国的行数据（排名第二），并计算 2017 年和 2008 年的 GDP 差值。

```
Top15 = answer_one()
Top15['avg'] = answer_three()
Top15.sort_values(by='avg', inplace=True, ascending=False)
row = Top15.iloc[1]
g2008 = row['2008']
g2017 = row['2017']
return abs(g2017-g2008)
```

```
Out[8]: 7639494387991.0
```

3.5 计算 15 个国家的人均能源供应项平均值

```
Top15 = answer_one()
temp = Top15['Energy Supply per Capita'].values.tolist()
return sum(temp)/len(temp)
```

```
Out[7]: 157.6
```

3.6 求可再生能源百分比最高的国家

先通过 max 方法找出可再生能源供应占比最多的国家的行索引，取出这一行，返回国家名和比例的 tuple。

```
Top15 = answer_one()
sub = Top15[Top15['% Renewable']==max(Top15['% Renewable'])]
country = sub.index.values[0]
percentage = sub['% Renewable'].values[0]
return (country,percentage)
```

```
Out[8]: ('Brazil', 69.648030000000006)
```

3.7 根据能源供应和人均能源供应估计人口

使用能源供应和人均能源供应估计人口值生成一个 series,输出第三人口大国的名称。

```
Top15 = answer_one()
population = Top15['Energy Supply']/Top15['Energy Supply per Capita']
temp = pd.Series(population,index=Top15.index,name='population').sort_values()
return temp.index.values[-3]
```

```
Out[10]: 'United States'
```

3.8 corr()方法计算列与列的相关系数

使用上一步方法计算出的人口数值和表格中的可引用文档数量求出人均可引用文档数量，并使用 pandas 库的 corr()方法计算本列和人均能源供应量列的相关系数。

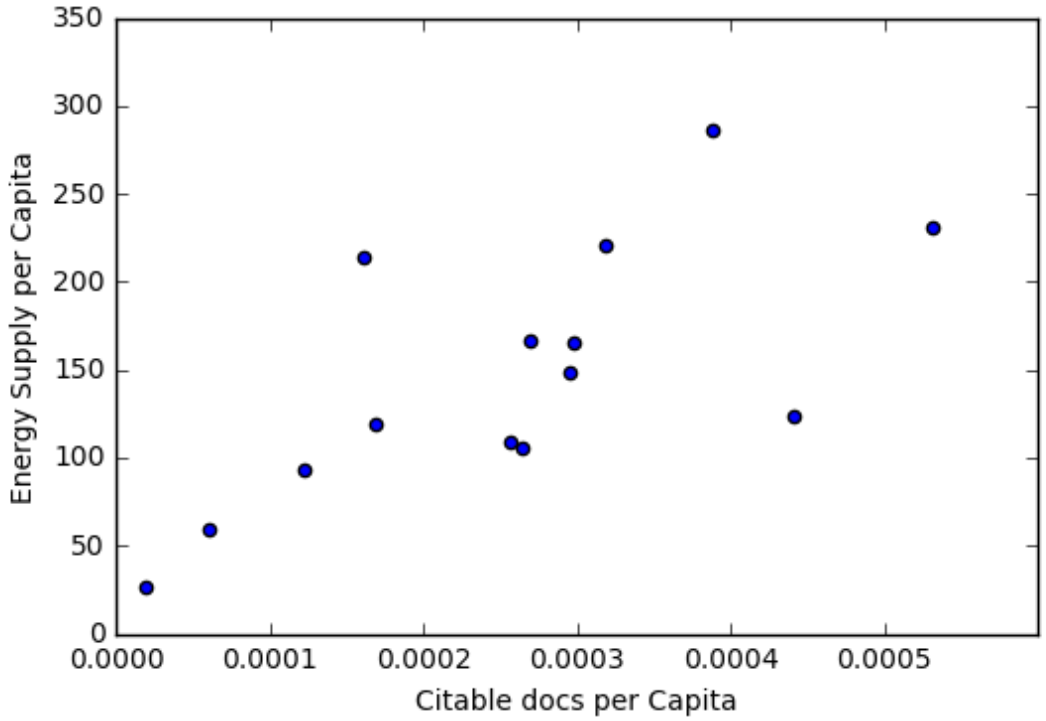
```
Top15 = answer_one()
newColumn = Top15['Citable documents']/(Top15['Energy Supply'] / Top15['Energy Supply per Capita'])
return Top15['Energy Supply per Capita'].corr(newColumn)
```

```
Out[11]: 0.77404323399911701
```

3.9 matplotlib 绘制散点图

使用 matpltlib 的 plot 方法可视化人均能源供应与人均可再生能源文档之间的关系。参数范围设置为 0-0.0006.

```
import matplotlib as plt
Top15 = answer_one()
Top15['Pop'] = Top15['Energy Supply'] / Top15['Energy Supply per Capita']
Top15['Citable docs per Capita'] = Top15['Citable documents'] / Top15['Pop']
Top15.plot(x='Citable docs per Capita', y='Energy Supply per Capita', kind='scatter', xlim=[0, 0.0006])
```



3.10 可再生能源使用率中值

使用 numpy 库的 median 方法计算出各国可再生能源使用率的中值，并根据是否大于中值对每个国家进行标记，大于中值标记 1，小于标记为 0，生成一个名为 HighRenew 的 series 并输出。

```
Top15 = answer_one().sort_values(by='Rank')
theColumn = Top15['% Renewable'].values
medianValue = np.median(theColumn)
mask1 = theColumn>=medianValue
mask2 = theColumn<medianValue
theColumn[mask1] = 1
theColumn[mask2] = 0
return pd.Series(theColumn,index=Top15.index,name='HighRenew')
```

```
Country
China                  1.0
United States          0.0
Japan                  0.0
United Kingdom         0.0
India                  0.0
Germany                1.0
Russian Federation     1.0
Canada                 1.0
France                 1.0
South Korea            0.0
Italy                  1.0
Iran                   0.0
Brazil                 1.0
Australia              0.0
Spain                  1.0
Name: HighRenew, dtype: float64
```

3.11 对各个国家按照洲分组统计分析人口数据

建立 ContinentDict 字典存放各个国家与洲的对应关系，使用 pandas 的 groupby 方法将 TOP15 表格中的国家依照 ContinentDict 字典分组，并以洲名为索引，每个洲的国家数，人口平均数，标准差，人口总数为列生成一个 DataFrame。

```
Top15 = answer_one()
Top15['pop'] = Top15['Energy Supply'] / Top15['Energy Supply per Capita']
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'}
index = []
item = []
for group,frame in Top15.groupby(ContinentDict):
    index.append(group)
    item.append({'size':len(frame),'sum':np.sum(frame['pop']),'mean':np.average(frame['pop']),'std':np.std(frame['pop'])})
return pd.DataFrame(item,index=index)
```

Out[15]:

	mean	size	std	sum
Asia	5.797333e+08	5	6.074036e+08	2.898666e+09
Australia	2.331602e+07	1	0.000000e+00	2.331602e+07
Europe	7.632161e+07	6	3.162885e+07	4.579297e+08
North America	1.764276e+08	2	1.411878e+08	3.528552e+08
South America	2.059153e+08	1	0.000000e+00	2.059153e+08

3.12 对各国可再生能源百分比进行区间划分

参照上一步，将每个国家对应的洲信息列加入表格，并使用 pandas 的分段函数 cut 按照各国的可再生能源百分比数值划分为 5 段，并输出每隔洲对应每个数据段中的国家数。

```
Top15 = answer_one()
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'}
Top15 = Top15.reset_index()
Top15['Continent'] = [ContinentDict[country] for country in Top15['Country']]
Top15['bins'] = pd.cut(Top15['% Renewable'],5)
return Top15.groupby(['Continent','bins']).size()
```

```
Continent      bins
Asia           (2.212, 15.753]     4
               (15.753, 29.227]    1
Australia      (2.212, 15.753]     1
Europe         (2.212, 15.753]     1
               (15.753, 29.227]    3
               (29.227, 42.701]    2
North America  (2.212, 15.753]     1
               (56.174, 69.648]    1
South America  (56.174, 69.648]    1
```

3.14 利用能源供应量估算人口数并转换为字符串

使用能源供应量列和人均能源供应量列估算每个国家的人口数（不省略小数），并将人口数转换为字符串，生成索引是国家名称，值是人口估计字符串的 series 输出。

```
Top15 = answer_one()
pop = Top15['Energy Supply'] / Top15['Energy Supply per Capita']
index = pop.index.values
value = []
for x in pop.values.tolist():
    value.append("{:,}".format(x))
return pd.Series(value,index=index,name='PopEst')
```

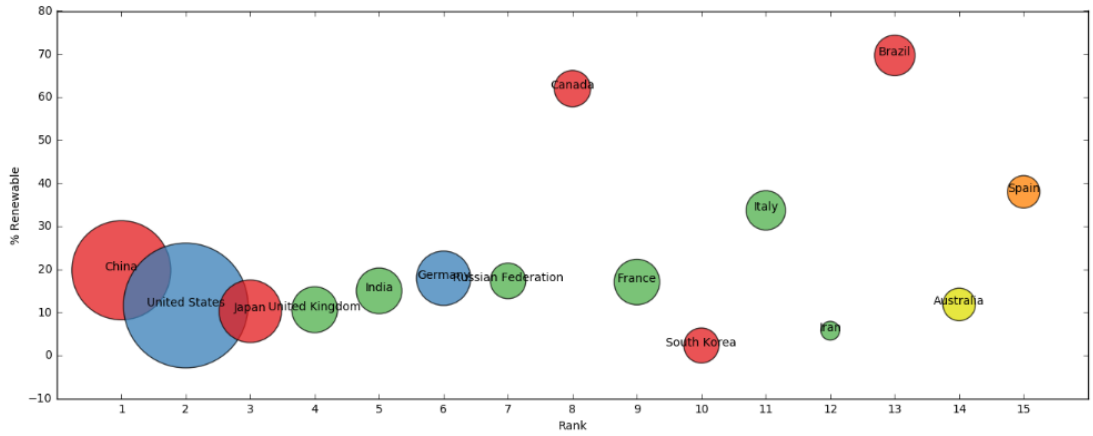
```
China                1,367,645,161.2903225
United States          317,615,384.61538464
Japan                  127,409,395.97315437
United Kingdom          63,870,967.741935484
India                1,276,730,769.2307692
Germany                 80,369,696.96969697
Russian Federation             143,500,000.0
Canada                  35,239,864.86486486
France                  63,837,349.39759036
South Korea             49,805,429.864253394
Italy                   59,908,256.880733944
Iran                    77,075,630.25210084
Brazil                 205,915,254.23728815
Australia              23,316,017.316017315
Spain                    46,443,396.2264151
```

3.15 气泡图绘制

使用 matplotlib.plot 方法以可再生能源百分比为纵轴，能源杂志贡献排名为横轴，2017 年 GDP 为气泡面积绘制气泡图。并使用 annotate 方法将国家名标记到气泡中央。

```
import matplotlib as plt
get_ipython().magic('matplotlib inline')
Top15 = answer_one()
ax = Top15.plot(x='Rank', y='% Renewable', kind='scatter',
                c=['#e41a1c','#377eb8','#e41a1c','#4daf4a','#4daf4a','#377eb8','#4daf4a','#e41a1c',
                   '#4daf4a','#e41a1c','#4daf4a','#4daf4a','#e41a1c','#dede00','#ff7f00'],
                xticks=range(1,16), s=6*Top15['2017']/10**10, alpha=.75, figsize=[16,6]);

for i, txt in enumerate(Top15.index):
    ax.annotate(txt, [Top15['Rank'][i], Top15['% Renewable'][i]], ha='center')
```



4 总结

本次实验实践参考《利用 python 进行数据分析》，使用了 numpy 的 average、median 等数据运算方法，pandans 库的 Dataframe、Series 数据结构，read_csv()、read_excel()、replace()、drop()、merge()、set_index()、iloc()、cor()、sort_value()、groupby()、cut()等方法对数据进行读取、索引、清洗、合并、切片、关系求取、分组、分段、排序等操作。并借助 Matpltlib 工具对数据的关系进行可视化。从中学到 python 数据分析的基本方法，为以后学习工作中进行系统的数据处理、分析打下了基础。