def get\_energy():

1. energy = read "Energy Indicators.xls" #note that column C has

#the country names with footnote numbers while column B has no footnotes

2. if you read the C column in 1 then remove any digits in the country column

3. replace all entries with the value '...' with np.NaN in the energy dataframe

4. convert 'Energy Supply' column values to giga-jouls

dicts = {"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"}

5. replace each country in dicts's keys with the corresponding dicts value

6. remove the part ' (...)' form all countries that have paranthesis, make

sure the result has no trailing space

7. return energy

# read the energy indicator file

def get\_energy():

energy = pd.read\_excel(io='Energy Indicators.xls',sheetname="Energy",skiprows=17,skip\_footer=38,

parse\_cols="C:F",

names=['Country', 'Energy Supply', 'Energy Supply per Capita', '% Renewable'])

for index in range(len(energy.index)):

if energy.loc[index,'Energy Supply']=='...':

energy.loc[index,'Energy Supply']=np.NaN

if energy.loc[index,'Energy Supply per Capita']=='...':

energy.loc[index,'Energy Supply per Capita']=np.NaN

return energy

get\_energy()

def get\_GDP():

GDP=pd.read\_csv("world\_bank.csv",header=4) # header has 4 lines

dicts = {"Korea, Rep.": "South Korea",

"Iran, Islamic Rep.": "Iran",

"Hong Kong SAR, China": "Hong Kong"}

for key in dicts.keys():

for index in range(len(GDP.index)):

if GDP.loc[index,'Country Name']==key:

#print(index)

GDP.loc[index,'Country Name']=dicts[key]

return GDP

def get\_ScimEn():

ScimEn=pd.read\_excel(io="scimagojr-3.xlsx")

return ScimEn

energy=get\_energy()

GDP=get\_GDP()

ScimEn=get\_ScimEn()

GDP.rename(columns={'Country Name':'Country'},inplace=True)

cols=list(['Country','2006','2007','2008','2009','2010','2011','2012','2013','2014','2015'])

GDP\_new=GDP[cols]

data=pd.merge(ScimEn,energy,how='left')

data\_one=pd.merge(data,GDP\_new,how='left')

data\_res=data\_one[:15]

data\_res.set\_index('Country',inplace=True)

data\_res

def get\_dataframe():

energy=get\_energy()

GDP=get\_GDP()

ScimEn=get\_ScimEn()

return energy, GDP, ScimEn

def answer\_two():

Energy, GDP, ScimEn=get\_dataframe()

data\_all=pd.merge(ScimEn,Energy,how='left')

data\_all=pd.merge(data\_all,GDP,how='left')

data\_all.set\_index('Country',inplace=True)

data\_one=answer\_one()

row\_one=len(data\_one.index)

col\_one=len(data\_one.columns)

row\_two=len(data\_all.index)

col\_two=len(data\_all.columns)

miss=row\_two\*col\_two-row\_one\*col\_one

return miss

# return "ANSWER"

answer\_two()

def answer\_three():

Top15 = answer\_one()

data\_three=Top15.loc[:,'2006':'2015']

data\_three['avgGDP']=data\_three.mean(axis=1,skipna=True)

return data\_three['avgGDP']

# return "ANSWER"

answer\_three()

def answer\_eleven():

Top15 = answer\_one()

Top15['PopEst'] = Top15['Energy Supply'] / Top15['Energy Supply per Capita']

Top15.reset\_index(level=0,inplace=True)

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['Continent']

for key in ContinentDict.keys():

for index in range(len(Top15.index)):

if Top15.loc[index,'Country']==key:

Top15.loc[index,'Continent']=ContinentDict[key]

# Get the sum, mean and std deviation

RES=df11.set\_index('Continent').groupby(level=0)['PopEst'].agg({'sum':np.sum,'mean':np.mean,'std':np.std})

# Get the size of each continent

RES2=Top15.groupby('Continent').count()

RES['size']=RES2['Country'].copy()

cols=(['size','sum','mean','std'])

RES=RES[cols]

return RES

answer\_eleven()

def answer\_twelve():

Top15 = answer\_one()

# Get the continent

Top15.reset\_index(level=0,inplace=True)

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['Continent']

for key in ContinentDict.keys():

for index in range(len(Top15.index)):

if Top15.loc[index,'Country']==key:

Top15.loc[index,'Continent']=ContinentDict[key]

# Cut the % Renewable to 5 bins

Top15['New % Renewable']=pd.cut(Top15['% Renewable'],5)

df=Top15.copy()

RES=df.set\_index(['Continent','New % Renewable']).groupby(level=['Continent','New % Renewable']).count()

return RES['Country']

answer\_twelve()

Top15 = answer\_one()

Top15['PopEst'] = Top15['Energy Supply'] / Top15['Energy Supply per Capita']

Top15.PopEst=Top15.PopEst.astype(float)

Top15

**UODATE**

def answer\_two():

Energy, GDP, ScimEn=get\_dataframe()

data\_all=pd.merge(ScimEn,Energy,how='left')

data\_all=pd.merge(data\_all,GDP,how='left')

data\_all.set\_index('Country',inplace=True)

data\_one=answer\_one()

row\_one=len(data\_one.index)

col\_one=len(data\_one.columns)

row\_two=len(data\_all.index)

col\_two=len(data\_all.columns)

miss=row\_two-row\_one

#print(data\_all)

return miss

# return "ANSWER"

answer\_two()

def answer\_four():

Top15 = answer\_one()

data\_four=Top15.loc[:,'2006':'2015']

data\_four['avgGDP']=data\_four.mean(axis=1,skipna=True)

# sorting by descending order

data\_four.sort\_values(by='avgGDP',axis=0,ascending=False,inplace=True)

low=(data\_four.iloc[5]['2006']) # 6th largest average GDP

high=(data\_four.iloc[5]['2015'])

return high-low

# return "ANSWER"

answer\_four()

def answer\_five():

Top15 = answer\_one()

num=Top15['Energy Supply per Capita'].mean(axis=0,skipna=True)

num=float(num)

return num

answer\_five()

def answer\_six():

Top15 = answer\_one()

maxium=Top15['% Renewable'].max(axis=0,skipna=True)

total=Top15['% Renewable'].sum(axis=0,skipna=True)

percent=maxium/total

max\_id=Top15['% Renewable'].idxmax(axis=0,skipna=True)

percent=float(percent)

# print(type(max\_id))

# print(type(percent))

# return max\_id,"{0:.9f}%".format(percent\*100)

return max\_id,percent

answer\_six()

def answer\_nine():

Top15 = answer\_one()

Top15['PopEst'] = Top15['Energy Supply'] / Top15['Energy Supply per Capita']

Top15['Citable docs per Capita'] = Top15['Citable documents'] / Top15['PopEst']

# Convert the value of two columns to float64

Top15.PopEst=Top15.PopEst.astype(float)

Top15['Citable docs per Capita']=Top15['Citable docs per Capita'].astype(float)

corr=Top15['Energy Supply per Capita'].corr(Top15['Citable docs per Capita'],method='pearson')

return corr

answer\_nine()

def answer\_ten():

Top15 = answer\_one()

median=Top15['% Renewable'].median(axis=0,skipna=True)

# Initialize HighRenew

Top15['HighRenew']=0

Top15.reset\_index(level=0,inplace=True)

for index in range(len(Top15.index)):

if Top15.loc[index,'% Renewable']>= median:

Top15.loc[index,'HighRenew']=1

# Sorting by rank

Top15.sort\_values(by='Rank',axis=0,ascending=True,inplace=True)

Top15.set\_index('Country',inplace=True)

return Top15['HighRenew']

answer\_ten()

def answer\_eleven():

Top15 = answer\_one()

Top15['PopEst'] = Top15['Energy Supply'] / Top15['Energy Supply per Capita']

Top15.reset\_index(level=0,inplace=True)

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['Continent']

for key in ContinentDict.keys():

for index in range(len(Top15.index)):

if Top15.loc[index,'Country']==key:

Top15.loc[index,'Continent']=ContinentDict[key]

df11=Top15.copy()

df11.PopEst=df11.PopEst.astype(float)

# Get the sum, mean and std deviation

RES=df11.set\_index('Continent').groupby(level=0)['PopEst'].agg({'sum':np.sum,'mean':np.mean,'std':np.std})

# Get the size of each continent

RES2=Top15.groupby('Continent').count()

RES['size']=RES2['Country'].copy()

cols=(['size','sum','mean','std'])

RES=RES[cols]

return RES

answer\_eleven()

def answer\_twelve():

Top15 = answer\_one()

# Get the continent

Top15.reset\_index(level=0,inplace=True)

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['Continent']

for key in ContinentDict.keys():

for index in range(len(Top15.index)):

if Top15.loc[index,'Country']==key:

Top15.loc[index,'Continent']=ContinentDict[key]

# Cut the % Renewable to 5 bins

Top15['Bins']=pd.cut(Top15['% Renewable'],5)

Top15

df=Top15.copy()

RES=df.set\_index(['Continent','Bins']).groupby(level=['Continent','Bins']).count()

RES.reset\_index(level=['Continent','Bins'],inplace=True)

RES.loc[0,'Country']=RES.loc[0,'Country']+1

RES.set\_index(['Continent','Bins'],inplace=True)

return RES['Country']

answer\_twelve()

def answer\_thirteen():

import locale

Top15 = answer\_one()

Top15['PopEst'] = Top15['Energy Supply'] / Top15['Energy Supply per Capita']

Top15.PopEst=Top15.PopEst.astype(float)

locale.setlocale(locale.LC\_NUMERIC, '')

Top15.reset\_index(level=0,inplace=True)

for index in range(len(Top15.index)):

Top15.loc[index,'PopEst']=locale.format('%f',Top15.loc[index,'PopEst'],grouping=True)

Top15.set\_index('Country',inplace=True)

Top15.PopEst=Top15.PopEst.astype(str)

RES=Top15['PopEst']

return RES

answer\_thirteen()

**UPDATE**

**def answer\_five():**

**Top15 = answer\_one()**

**energyAll=Top15['Energy Supply per Capita'].sum(axis=0,skipna=True)**

**num=energyAll/15**

**num=float(num)**

**return num**

**answer\_five()**

**def answer\_six():**

**Top15 = answer\_one()**

**maxium=Top15['% Renewable'].max(axis=0,skipna=True)**

**max\_id=Top15['% Renewable'].idxmax(axis=0,skipna=True)**

**maxium=float(maxium)**

**# print(type(max\_id))**

**# print(type(percent))**

**# return max\_id,"{0:.9f}%".format(percent\*100)**

**return (max\_id,maxium)**

**answer\_six()**

**def answer\_nine():**

**Top15 = answer\_one()**

**Top15['PopEst'] = Top15['Energy Supply'] / Top15['Energy Supply per Capita']**

**Top15['Citable docs per Capita'] = Top15['Citable documents'] / Top15['PopEst']**

**# Convert the value of two columns to float64**

**Top15.PopEst=Top15.PopEst.astype(float)**

**Top15['Citable docs per Capita']=Top15['Citable docs per Capita'].astype(float)**

**Top15['Energy Supply per Capita']=Top15['Energy Supply per Capita'].astype(float)**

**corr=Top15['Energy Supply per Capita'].corr(Top15['Citable docs per Capita'],method='pearson')**

**return corr**

**answer\_nine()**

**# read the energy indicator file**

**def get\_energy():**

**energy = pd.read\_excel(io='Energy Indicators.xls',sheetname="Energy",skiprows=17,skip\_footer=38,**

**parse\_cols="C:F",**

**names=['Country', 'Energy Supply', 'Energy Supply per Capita', '% Renewable'])**

**# Rename part of the countries**

**dicts = {"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"}**

**for index in range(len(energy.index)):**

**if energy.loc[index,'Energy Supply']=='...':**

**energy.loc[index,'Energy Supply']=np.NaN**

**if energy.loc[index,'Energy Supply per Capita']=='...':**

**energy.loc[index,'Energy Supply per Capita']=np.NaN**

**energy.loc[index,'Energy Supply']\*=1e6 # Convert peta-joule to gig-joule**

**dicts = {"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"}**

**dict\_num={"0":"","1":"","2":"","3":"","4":"","5":"","6":"","7":"","8":"","9":""}**

**energy.Country=energy.Country.astype(str)**

**for index in range(len(energy.index)):**

**# Remove numbers**

**for key\_num in dict\_num.keys():**

**energy.loc[index,'Country']=energy.loc[index,'Country'].replace(key\_num,"")**

**# Remove text in (...)**

**energy.loc[index,'Country']=re.sub("\(.\*?\)", "",energy.loc[index,'Country'])**

**for key in dicts.keys():**

**if (energy.loc[index,'Country']==key):**

**energy.loc[index,'Country']=dicts[key]**

**if energy.loc[index,'Country']=='Republic of Korea':**

**energy.loc[index,'Country']='South Korea'**

**# Remove whitespace for Iran**

**energy.loc[98,'Country']=energy.loc[98,'Country'].replace(" ","")**

**energy['Energy Supply']=energy['Energy Supply'].astype(float)**

**energy['Energy Supply per Capita']=energy['Energy Supply per Capita'].astype(float)**

**return energy**