Mini-project-notebook

July 5, 2020

1 MiniProject - Exploring World Indicators

[1]:
```
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
pd.set_option('display.max_rows', 40)
```

[2]:
```
global country_df, indicators_df

country_df = pd.read_csv('world-development-indicators/Country.csv')
indicators_df = pd.read_csv('world-development-indicators/Indicators.csv')
```

1.1 Basic exploration

1.1.1 Information in Countries.csv (stored in countries_df)

[3]:
```
## Basic exploration
countries = country_df['ShortName'].unique()
num_countries = len(countries)
print('Found %d countries:\n%s' % (num_countries, countries))
```

```
Found 247 countries:
['Afghanistan' 'Albania' 'Algeria' 'American Samoa' 'Andorra' 'Angola'
 'Antigua and Barbuda' 'Arab World' 'Argentina' 'Armenia' 'Aruba'
 'Australia' 'Austria' 'Azerbaijan' 'Bahrain' 'Bangladesh' 'Barbados'
 'Belarus' 'Belgium' 'Belize' 'Benin' 'Bermuda' 'Bhutan' 'Bolivia'
 'Bosnia and Herzegovina' 'Botswana' 'Brazil' 'Brunei' 'Bulgaria'
 'Burkina Faso' 'Burundi' 'Cabo Verde' 'Cambodia' 'Cameroon' 'Canada'
 'Caribbean small states' 'Cayman Islands' 'Central African Republic'
 'Central Europe and the Baltics' 'Chad' 'Channel Islands' 'Chile' 'China'
 'Colombia' 'Comoros' 'Congo' 'Costa Rica' "Côte d'Ivoire" 'Croatia'
 'Cuba' 'Curaçao' 'Cyprus' 'Czech Republic' "Dem. People's Rep. Korea"
 'Dem. Rep. Congo' 'Denmark' 'Djibouti' 'Dominica' 'Dominican Republic'
 'East Asia & Pacific (all income levels)'
 'East Asia & Pacific (developing only)' 'Ecuador' 'Egypt' 'El Salvador'
 'Equatorial Guinea' 'Eritrea' 'Estonia' 'Ethiopia' 'Euro area'
```

```
'Europe & Central Asia (all income levels)'
'Europe & Central Asia (developing only)' 'European Union'
'Faeroe Islands' 'Fiji' 'Finland'
'Fragile and conflict affected situations' 'France' 'French Polynesia'
'Gabon' 'Georgia' 'Germany' 'Ghana' 'Greece' 'Greenland' 'Grenada' 'Guam'
'Guatemala' 'Guinea' 'Guinea-Bissau' 'Guyana' 'Haiti'
'Heavily indebted poor countries (HIPC)' 'High income'
'High income: nonOECD' 'High income: OECD' 'Honduras'
'Hong Kong SAR, China' 'Hungary' 'Iceland' 'India' 'Indonesia' 'Iran'
'Iraq' 'Ireland' 'Isle of Man' 'Israel' 'Italy' 'Jamaica' 'Japan'
'Jordan' 'Kazakhstan' 'Kenya' 'Kiribati' 'Korea' 'Kosovo' 'Kuwait'
'Kyrgyz Republic' 'Lao PDR'
'Latin America & Caribbean (all income levels)'
'Latin America & Caribbean (developing only)' 'Latvia'
'Least developed countries: UN classification' 'Lebanon' 'Lesotho'
'Liberia' 'Libya' 'Liechtenstein' 'Lithuania' 'Low & middle income'
'Low income' 'Lower middle income' 'Luxembourg' 'Macao SAR, China'
'Macedonia' 'Madagascar' 'Malawi' 'Malaysia' 'Maldives' 'Mali' 'Malta'
'Marshall Islands' 'Mauritania' 'Mauritius' 'Mexico' 'Micronesia'
'Middle East & North Africa (all income levels)'
'Middle East & North Africa (developing only)' 'Middle income' 'Moldova'
'Monaco' 'Mongolia' 'Montenegro' 'Morocco' 'Mozambique' 'Myanmar'
'Namibia' 'Nepal' 'Netherlands' 'New Caledonia' 'New Zealand' 'Nicaragua'
'Niger' 'Nigeria' 'North America' 'Northern Mariana Islands' 'Norway'
'OECD members' 'Oman' 'Other small states' 'Pacific island small states'
'Pakistan' 'Palau' 'Panama' 'Papua New Guinea' 'Paraguay' 'Peru'
'Philippines' 'Poland' 'Portugal' 'Puerto Rico' 'Qatar' 'Romania'
'Russia' 'Rwanda' 'Samoa' 'San Marino' 'São Tomé and Principe'
'Saudi Arabia' 'Senegal' 'Serbia' 'Seychelles' 'Sierra Leone' 'Singapore'
'Sint Maarten (Dutch part)' 'Slovak Republic' 'Slovenia' 'Small states'
'Solomon Islands' 'Somalia' 'South Africa' 'South Asia' 'South Sudan'
'Spain' 'Sri Lanka' 'St. Kitts and Nevis' 'St. Lucia'
'St. Martin (French part)' 'St. Vincent and the Grenadines'
'Sub-Saharan Africa (all income levels)'
'Sub-Saharan Africa (developing only)' 'Sudan' 'Suriname' 'Swaziland'
'Sweden' 'Switzerland' 'Syrian Arab Republic' 'Tajikistan' 'Tanzania'
'Thailand' 'The Bahamas' 'The Gambia' 'Timor-Leste' 'Togo' 'Tonga'
'Trinidad and Tobago' 'Tunisia' 'Turkey' 'Turkmenistan'
'Turks and Caicos Islands' 'Tuvalu' 'Uganda' 'Ukraine'
'United Arab Emirates' 'United Kingdom' 'United States'
'Upper middle income' 'Uruguay' 'Uzbekistan' 'Vanuatu' 'Venezuela'
'Vietnam' 'Virgin Islands' 'West Bank and Gaza' 'World' 'Yemen' 'Zambia'
'Zimbabwe']
```

[4]:
```
## Basic exploration
keys_c = country_df.columns
num_keys_c = len(keys_c)
```

```
print('Found %d keys in countries.csv:\n%s\n\n' % (num_keys_c, keys_c))
```

```
Found 31 keys in countries.csv:
Index(['CountryCode', 'ShortName', 'TableName', 'LongName', 'Alpha2Code',
       'CurrencyUnit', 'SpecialNotes', 'Region', 'IncomeGroup', 'Wb2Code',
       'NationalAccountsBaseYear', 'NationalAccountsReferenceYear',
       'SnaPriceValuation', 'LendingCategory', 'OtherGroups',
       'SystemOfNationalAccounts', 'AlternativeConversionFactor',
       'PppSurveyYear', 'BalanceOfPaymentsManualInUse',
       'ExternalDebtReportingStatus', 'SystemOfTrade',
       'GovernmentAccountingConcept', 'ImfDataDisseminationStandard',
       'LatestPopulationCensus', 'LatestHouseholdSurvey',
       'SourceOfMostRecentIncomeAndExpenditureData',
       'VitalRegistrationComplete', 'LatestAgriculturalCensus',
       'LatestIndustrialData', 'LatestTradeData', 'LatestWaterWithdrawalData'],
      dtype='object')
```

1.1.2 Information in Indicators.csv (stored in indicators_df)

[5]:
```
indicators = indicators_df['IndicatorName'].unique()
num_indicators = len(indicators)

years = indicators_df['Year'].unique()
num_years = len(years)

countries_ind = indicators_df['CountryName'].unique()
num_countries_ind = len(countries_ind)

print('Data for %d years:\n%s\n\n' %(num_years, years))
print('Data for %d countries:\n%s\n\n' %(num_countries_ind, countries_ind))

print('Found %d unique indicators across %d countries and %d years' %␣
 ↪(num_indicators, num_countries_ind, num_years))
display("Here are the indicators", indicators)
```

```
Data for 56 years:
[1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973
 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987
 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001
 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015]


Data for 247 countries:
['Arab World' 'Caribbean small states' 'Central Europe and the Baltics'
 'East Asia & Pacific (all income levels)'
```

'East Asia & Pacific (developing only)' 'Euro area'
'Europe & Central Asia (all income levels)'
'Europe & Central Asia (developing only)' 'European Union'
'Fragile and conflict affected situations'
'Heavily indebted poor countries (HIPC)' 'High income'
'High income: nonOECD' 'High income: OECD'
'Latin America & Caribbean (all income levels)'
'Latin America & Caribbean (developing only)'
'Least developed countries: UN classification' 'Low & middle income'
'Low income' 'Lower middle income'
'Middle East & North Africa (all income levels)'
'Middle East & North Africa (developing only)' 'Middle income'
'North America' 'OECD members' 'Other small states'
'Pacific island small states' 'Small states' 'South Asia'
'Sub-Saharan Africa (all income levels)'
'Sub-Saharan Africa (developing only)' 'Upper middle income' 'World'
'Afghanistan' 'Albania' 'Algeria' 'American Samoa' 'Andorra' 'Angola'
'Antigua and Barbuda' 'Argentina' 'Armenia' 'Aruba' 'Australia' 'Austria'
'Azerbaijan' 'Bahamas, The' 'Bahrain' 'Bangladesh' 'Barbados' 'Belarus'
'Belgium' 'Belize' 'Benin' 'Bermuda' 'Bhutan' 'Bolivia'
'Bosnia and Herzegovina' 'Botswana' 'Brazil' 'Brunei Darussalam'
'Bulgaria' 'Burkina Faso' 'Burundi' 'Cabo Verde' 'Cambodia' 'Cameroon'
'Canada' 'Cayman Islands' 'Central African Republic' 'Chad'
'Channel Islands' 'Chile' 'China' 'Colombia' 'Comoros' 'Congo, Dem. Rep.'
'Congo, Rep.' 'Costa Rica' "Cote d'Ivoire" 'Croatia' 'Cuba' 'Curacao'
'Cyprus' 'Czech Republic' 'Denmark' 'Djibouti' 'Dominica'
'Dominican Republic' 'Ecuador' 'Egypt, Arab Rep.' 'El Salvador'
'Equatorial Guinea' 'Eritrea' 'Estonia' 'Ethiopia' 'Faeroe Islands'
'Fiji' 'Finland' 'France' 'French Polynesia' 'Gabon' 'Gambia, The'
'Georgia' 'Germany' 'Ghana' 'Greece' 'Greenland' 'Grenada' 'Guam'
'Guatemala' 'Guinea' 'Guinea-Bissau' 'Guyana' 'Haiti' 'Honduras'
'Hong Kong SAR, China' 'Hungary' 'Iceland' 'India' 'Indonesia'
'Iran, Islamic Rep.' 'Iraq' 'Ireland' 'Isle of Man' 'Israel' 'Italy'
'Jamaica' 'Japan' 'Jordan' 'Kazakhstan' 'Kenya' 'Kiribati'
'Korea, Dem. Rep.' 'Korea, Rep.' 'Kosovo' 'Kuwait' 'Kyrgyz Republic'
'Lao PDR' 'Latvia' 'Lebanon' 'Lesotho' 'Liberia' 'Libya' 'Liechtenstein'
'Lithuania' 'Luxembourg' 'Macao SAR, China' 'Macedonia, FYR' 'Madagascar'
'Malawi' 'Malaysia' 'Maldives' 'Mali' 'Malta' 'Marshall Islands'
'Mauritania' 'Mauritius' 'Mexico' 'Micronesia, Fed. Sts.' 'Moldova'
'Monaco' 'Mongolia' 'Montenegro' 'Morocco' 'Mozambique' 'Myanmar'
'Namibia' 'Nepal' 'Netherlands' 'New Caledonia' 'New Zealand' 'Nicaragua'
'Niger' 'Nigeria' 'Northern Mariana Islands' 'Norway' 'Oman' 'Pakistan'
'Palau' 'Panama' 'Papua New Guinea' 'Paraguay' 'Peru' 'Philippines'
'Poland' 'Portugal' 'Puerto Rico' 'Qatar' 'Romania' 'Russian Federation'
'Rwanda' 'Samoa' 'San Marino' 'Sao Tome and Principe' 'Saudi Arabia'
'Senegal' 'Serbia' 'Seychelles' 'Sierra Leone' 'Singapore'
'Sint Maarten (Dutch part)' 'Slovak Republic' 'Slovenia'
'Solomon Islands' 'Somalia' 'South Africa' 'South Sudan' 'Spain'

```
 'Sri Lanka' 'St. Kitts and Nevis' 'St. Lucia' 'St. Martin (French part)'
 'St. Vincent and the Grenadines' 'Sudan' 'Suriname' 'Swaziland' 'Sweden'
 'Switzerland' 'Syrian Arab Republic' 'Tajikistan' 'Tanzania' 'Thailand'
 'Timor-Leste' 'Togo' 'Tonga' 'Trinidad and Tobago' 'Tunisia' 'Turkey'
 'Turkmenistan' 'Turks and Caicos Islands' 'Tuvalu' 'Uganda' 'Ukraine'
 'United Arab Emirates' 'United Kingdom' 'United States' 'Uruguay'
 'Uzbekistan' 'Vanuatu' 'Venezuela, RB' 'Vietnam' 'Virgin Islands (U.S.)'
 'West Bank and Gaza' 'Yemen, Rep.' 'Zambia' 'Zimbabwe']


Found 1344 unique indicators across 247 countries and 56 years

'Here are the indicators'


array(['Adolescent fertility rate (births per 1,000 women ages 15-19)',
       'Age dependency ratio (% of working-age population)',
       'Age dependency ratio, old (% of working-age population)', ...,
       'Fish species, threatened', 'Mammal species, threatened',
       'Plant species (higher), threatened'], dtype=object)
```

[6]:
```
#Function to get the country code of a country
def get_country_code(country_name):
    country_pattern = '.*' + country_name + '.*'
    b = country_df['LongName'].str.contains(country_pattern, case = False,␣
 ↪regex = True)

    if b.any():
        country_info = country_df[b]
        #return country_info['CountryCode'].values[0]
        return country_info['CountryCode'].values
    else:
        b = country_df['TableName'].str.contains(country_pattern, case = False,␣
 ↪regex = True)
        if b.any():
            country_info = country_df[b]
            #return country_info['CountryCode'].values[0]
            return country_info['CountryCode'].values
        else:
            return np.array([])

#Test function
get_country_code('mexico')
```

[6]:
```
array(['MEX'], dtype=object)
```

[7]:
```
#Function to get the list of indicators containing a certain pattern associated␣
↪with a given country
def get_indicators(indicator_pattern, country_code):

    all_indicators_of_given_country = \
        pd.Series(indicators_df[indicators_df['CountryCode'] ==␣
↪country_code]['IndicatorName'].unique())

    indicator_pattern = '.*' + indicator_pattern + '.+'
    b = all_indicators_of_given_country.str.contains(indicator_pattern, case =␣
↪False, regex = True)

    if b.any():
        return all_indicators_of_given_country[b]
    else:
        return np.array([])

#Test function
get_indicators('poverty', 'MEX')
```

[7]:
```
666              Poverty gap at $1.90 a day (2011 PPP) (%)
667              Poverty gap at $3.10 a day (2011 PPP) (%)
668     Poverty headcount ratio at $1.90 a day (2011 P…
669     Poverty headcount ratio at $3.10 a day (2011 P…
1255    Poverty headcount ratio at national poverty li…
1259    Rural poverty headcount ratio at national pove…
1260    Urban poverty headcount ratio at national pove…
dtype: object
```

1.1.3 Let the exploration begin

Enter a string pattern representing an indicator and a string pattern representing the country.

E.g. indicator string pattern = fertility , country string pattern = india

[8]:
```
count = 1

indicator = input('Enter a string representing the indicator you want to␣
↪explore:  ')
country_name = input('Which country\'s indicators (enter a string):  ')

#Can we find a country with the given string representing a country?
country_code = get_country_code(country_name)
if len(country_code) == 0:
    print('** Did not find any country with containing the string \"%s"' %␣
↪country_name)
else:
```

```
    if len(country_code) > 1:
        print('Multiple countries found:')
        for code in country_code:
            print(code)
        country_code = input('Please type the country code you want: ')
    else:
        print(country_code)
        country_code = country_code[0]

    indicators_list = get_indicators(indicator, country_code)

    if len(indicators_list) > 0:
        print('\n\nFound %d indicators containing the string \"%s\" for %s:' \
              % (len(indicators_list), indicator,country_name))

        for ind in indicators_list:
            print('\t%d\t%s' % (count,ind))
            count += 1

    else:
        print('\n\nNo indicators were found for the given country and␣
↪indicator')
```

```
Enter a string representing the indicator you want to explore:  birth
Which country's indicators (enter a string):  united states of america
['USA']


Found 17 indicators containing the string "birth" for united states of america:
        1       Adolescent fertility rate (births per 1,000 women ages 15-19)
        2       Birth rate, crude (per 1,000 people)
        3       Fertility rate, total (births per woman)
        4       Life expectancy at birth, female (years)
        5       Life expectancy at birth, male (years)
        6       Life expectancy at birth, total (years)
        7       Mortality rate, infant (per 1,000 live births)
        8       Births attended by skilled health staff (% of total)
        9       Maternal mortality ratio (modeled estimate, per 100,000 live
births)
        10      Mortality rate, infant, female (per 1,000 live births)
        11      Mortality rate, infant, male (per 1,000 live births)
        12      Mortality rate, neonatal (per 1,000 live births)
        13      Mortality rate, under-5, female (per 1,000 live births)
        14      Mortality rate, under-5, male (per 1,000 live births)
        15      Maternal mortality ratio (national estimate, per 100,000 live
births)
        16      Low-birthweight babies (% of births)
```

```
17      Completeness of birth registration (%)
```

1.2 Determine correlation between fertility and birth rates

Going to focus on Birth rate and fertility rate and see how they are correlated among the developed and developing countries

[9]:
```
#Going to focus on Birth rate and fertility rate and see how they are␣
 ↪correlated among the
#countries with the lowest and highest GDP per capita

valid_years = range(1990,2014)

developed_countries = ['United States', 'United Kingdom', 'France', 'Germany',␣
 ↪'Australia', 'Japan']
developing_countries = ['Brazil', 'India', 'China', 'Nigeria', 'South Africa',␣
 ↪'Malaysia']

indicators_birth = ['Birth rate, crude (per 1,000 people)', 'Fertility rate,␣
 ↪total (births per woman)']
indicators_health = ['Death rate, crude (per 1,000 people)', 'Health␣
 ↪expenditure, total (% of GDP)']

#Function to give the data for a given set of indicators, countries and years
def get_indicator_data(list_of_indicators, list_of_countries, list_of_years):
    return indicators_df[\
                         (indicators_df['IndicatorName'].
 ↪isin(list_of_indicators)) & \
                          (indicators_df['CountryName'].isin(list_of_countries))␣
 ↪& \
                         (indicators_df['Year'].
 ↪isin(list_of_years))][['CountryName', 'IndicatorName', 'Year', 'Value']]


#Get required indicator values about the countries for specific years
def get_stats(indicators_list, countries_list, years_list):
    countries_stats = {}
    df_all = get_indicator_data(indicators_list, countries_list, years_list)
    for country in countries_list:
        df1 = pd.DataFrame(index = valid_years, columns = indicators_list)
        for ind in indicators_list:
            df1[ind] = df_all[(df_all['IndicatorName'] == ind) & \
                              (df_all['CountryName'] == country) & \
                             (df_all['Year'].isin(valid_years))].
 ↪sort_values(by='Year')['Value'].values
        countries_stats[country] = df1
    return countries_stats
```

```
#Get correlation value between 'Birth rate, crude (per 1,000 people)'
#and 'Fertility rate, total (births per woman)'
birth_fertility_data = get_stats(indicators_birth,␣
↪developing_countries+developed_countries, valid_years)
#health_data = get_stats(indicators_health,␣
↪developing_countries+developed_countries, valid_years)

e = []
for c in developing_countries+developed_countries:
    corr1 = birth_fertility_data[c].corr()
    #corr1 = health_data[c].corr()
    e.append([c, corr1.loc[corr1.columns[1]].values[0]])

country_corr = pd.DataFrame(e, columns = ['Country', 'Corr'])
country_corr.sort_values(by='Corr')
```

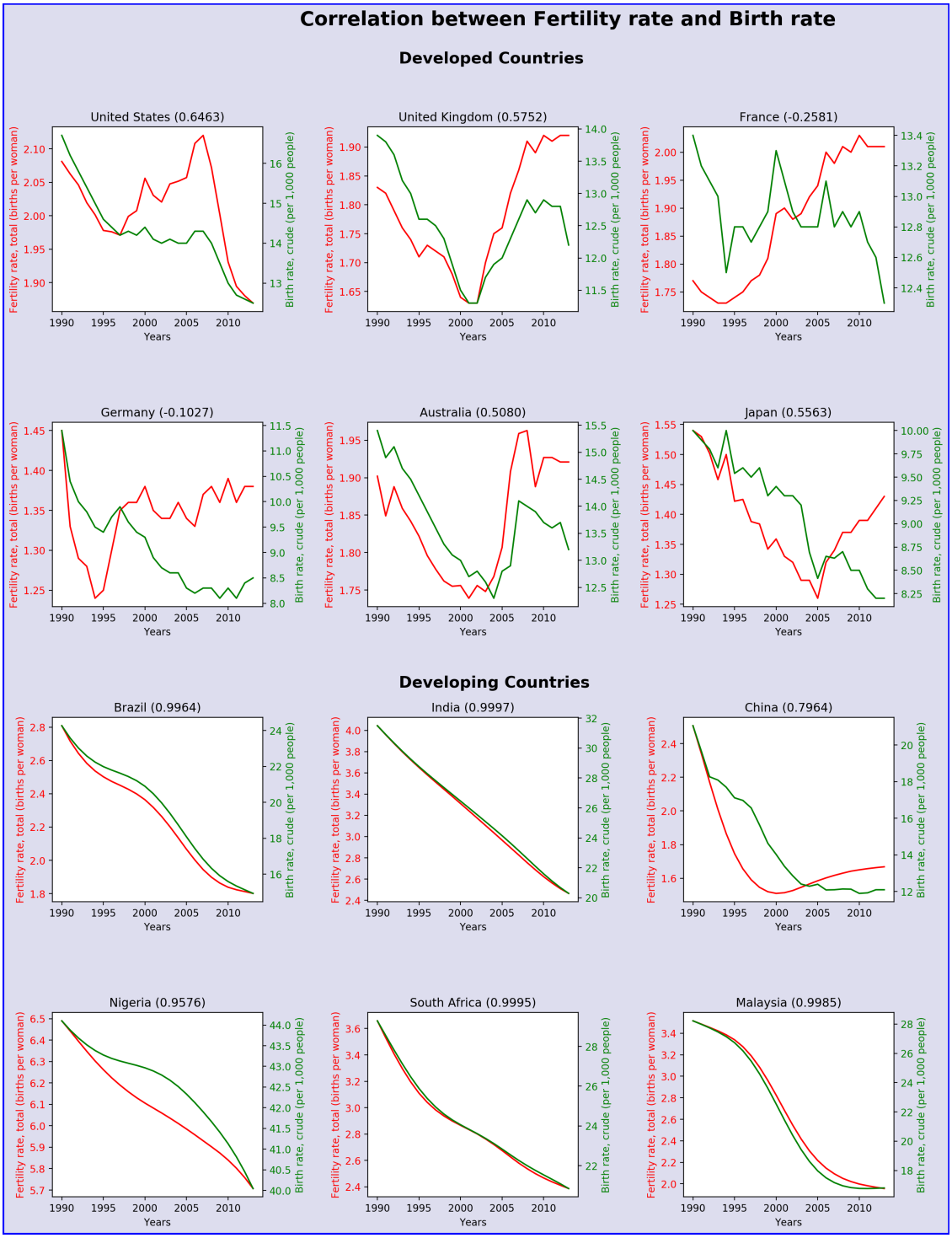
[9]:

	Country	Corr
8	France	-0.258092
9	Germany	-0.102722
10	Australia	0.507955
11	Japan	0.556251
7	United Kingdom	0.575226
6	United States	0.646321
2	China	0.796368
3	Nigeria	0.957638
0	Brazil	0.996429
5	Malaysia	0.998482
4	South Africa	0.999498
1	India	0.999669

1.3 Graph the fertility and birth rates, per country and indicate the correlation

[11]:
```
#Initial setup
fig = plt.figure(figsize=(14,18), dpi = 200, edgecolor = 'blue', linewidth = 3,␣
↪facecolor = '#ddddee')
fig.tight_layout()

plt.figtext(0.35,1, 'Correlation between Fertility rate and Birth rate',␣
↪fontsize = 20, fontweight = 'bold')
plt.figtext(0.45,0.97, "Developed Countries", fontsize = 16, fontweight =␣
↪'bold')
plt.figtext(0.45,0.48, "Developing Countries", fontsize = 16, fontweight =␣
↪'bold')

y1_color = 'red'
```

```
y2_color = 'green'

xvalues_common = birth_fertility_data['India'].index
axs_index = 1

for country in developed_countries+ developing_countries:

    #Draw the figure, drop in the correlation value
    corr = birth_fertility_data[country].corr()
    corr = corr.loc[corr.columns[1]].values[0]

    #Set the X-axis
    ax = fig.add_subplot(4, 3, axs_index)
    ax.set_title('%s (%-0.4f)' % (country, corr))
    ax.set_xlabel('Years')
    ax.set_xlim((1988,2015))
    ax.autoscale()

    #Set the first Y-axis
    y1_values = birth_fertility_data[country][[indicators_birth[1]]].values
    #ax.set_ylim(0,7)
    ax.set_ylabel(indicators_birth[1], color = y1_color)
    ax.tick_params(axis='y', labelcolor=y1_color)
    ax.plot(xvalues_common, y1_values,  color = y1_color)

    #Set the second Y-axis
    y2_values = birth_fertility_data[country][[indicators_birth[0]]].values
    ay = ax.twinx()
    #ay.set_ylim(5,45)
    ay.set_ylabel(indicators_birth[0], color = y2_color)
    ay.tick_params(axis='y', labelcolor=y2_color)
    ay.plot(xvalues_common, y2_values, color = y2_color)

    axs_index += 1

plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.6,␣
 ↪wspace=0.5)
plt.show()
```


Correlation between Fertility rate and Birth rate
Developed Countries
United States (0.6463)
United Kingdom (0.5752)
France (-0.2581)
Germany (-0.1027)
Australia (0.5080)
Japan (0.5563)
Developing Countries
Brazil (0.9964)
India (0.9997)
China (0.7964)
Nigeria (0.9576)
South Africa (0.9995)
Malaysia (0.9985)
Fertility rate, total (births per woman)
Birth rate, crude (per 1,000 people)
Years