Investigate_a_Dataset

April 30, 2019

> **Tip**: Welcome to the Investigate a Dataset project! You will find tips in quoted sections like this to help organize your approach to your investigation. Before submitting your project, it will be a good idea to go back through your report and remove these sections to make the presentation of your work as tidy as possible. First things first, you might want to double-click this Markdown cell and change the title so that it reflects your dataset and investigation.

1 Project: What could influence birth rate in Kuwait after Gulf War

1.1 Table of Contents

## Introduction

> **Tip**: Birth rate flutruates over the years. This study is looking at the factors that could influence the birth rate between 1991 and 2017 in Kuwait, after Gulf War. After Feb, 1991, Kuwait was liberated. Kuwait spent more than 5 billion to repair oil infrastructure damaged during the Gulf war. In this study, I am particularly interested in how the economy recovery resulting in GDP and female employment rate, thus in turn influenced birth rate.
>
> The dependent variable is the birth rate. The indepedant variables include female employment rate, GDP, urban population growth, and military expenditure during the same period in Kuwait.
>
> Four csv files are downloaded from https://www.gapminder.org/data/. children_per_woman_total_fertility.csv, females_aged_15_24_employment_rate_percent.csv, military_expenditure_percent_of_gdp.csv, urban_population_growth_annual_percent.csv and income_per_person_gdppercapita_ppp_inflation_adjusted.csv

```
In [1]: # Use this cell to set up import statements for all of the packages that you
        #   plan to use.
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline

        # Remember to include a 'magic word' so that your visualizations are plotted
        #   inline with the notebook. See this page for more:
        #   http://ipython.readthedocs.io/en/stable/interactive/magics.html (later)
```

## Data Wrangling

Tip: In this section of the report, you will load in the data, check for cleanliness, and then trim and clean your dataset for analysis. Make sure that you document your steps carefully and justify your cleaning decisions.

1.1.1 General Properties

```
In [2]: # Load four csv files as mentioned above.

        df_birth_rate = pd.read_csv('children_per_woman_total_fertility.csv')
        df_employment_rate = pd.read_csv('females_aged_15_24_employment_rate_percent.csv')
        df_military_expenditure = pd.read_csv('military_expenditure_percent_of_gdp.csv')
        df_urban_growth = pd.read_csv('urban_population_growth_annual_percent.csv')
        df_gdp = pd.read_csv('income_per_person_gdppercapita_ppp_inflation_adjusted.csv')
```

```
In [3]: # investigate each table
        # birth rate
        df_birth_rate.shape
```

```
Out[3]: (184, 220)
```

```
In [4]: # Female employment rate
        df_employment_rate.shape
```

```
Out[4]: (179, 33)
```

```
In [5]: # Urban growth rate
        df_urban_growth.shape
```

```
Out[5]: (194, 59)
```

```
In [6]: # military expenditure
        df_military_expenditure.shape
```

```
Out[6]: (165, 59)
```

```
In [7]: # gdp
        df_gdp.shape
```

```
Out[7]: (193, 220)
```

```
In [8]: df_birth_rate.head()
```

```
Out[8]:               country  1800  1801  1802  1803  1804  1805  1806  1807  1808  \
        0          Afghanistan  7.00  7.00  7.00  7.00  7.00  7.00  7.00  7.00  7.00
        1              Albania  4.60  4.60  4.60  4.60  4.60  4.60  4.60  4.60  4.60
        2              Algeria  6.99  6.99  6.99  6.99  6.99  6.99  6.99  6.99  6.99
        3               Angola  6.93  6.93  6.93  6.93  6.93  6.93  6.93  6.94  6.94
        4  Antigua and Barbuda  5.00  5.00  4.99  4.99  4.99  4.98  4.98  4.97  4.97

           ...  2009  2010  2011  2012  2013  2014  2015  2016  2017  2018
        0  ...  6.04  5.82  5.60  5.38  5.17  4.98  4.80  4.64  4.48  4.33
        1  ...  1.65  1.65  1.67  1.69  1.70  1.71  1.71  1.71  1.71  1.71
        2  ...  2.83  2.89  2.93  2.94  2.92  2.89  2.84  2.78  2.71  2.64
        3  ...  6.24  6.16  6.08  6.00  5.92  5.84  5.77  5.69  5.62  5.55
        4  ...  2.15  2.13  2.12  2.10  2.09  2.08  2.06  2.05  2.04  2.03

        [5 rows x 220 columns]
```

```
In [9]: # take data from 1991 to 2017
        # birth rate
        df_birth_rate = df_birth_rate.filter(items=['country', '1991', '1992', '1993', '1994', '
        df_birth_rate.head(1)
```

```
Out[9]:        country  1991  1992  1993  1994  1995  1996  1997  1998  1999  ...   \
        0  Afghanistan  7.48   7.5  7.54  7.57  7.61  7.63  7.64  7.62  7.57  ...

           2008  2009  2010  2011  2012  2013  2014  2015  2016  2017
        0  6.25  6.04  5.82   5.6  5.38  5.17  4.98   4.8  4.64  4.48

        [1 rows x 28 columns]
```

```
In [10]: # female employment rate
         df_employment_rate = df_employment_rate.filter(items=['country', '1991', '1992', '1993'
         df_employment_rate.head()
```

```
Out[10]:        country  1991  1992   1993  1994   1995   1996   1997   1998   1999  \
         0  Afghanistan  12.8  13.1  12.80  12.7  12.90  12.80  12.70  12.70  12.70
         1      Albania  42.5  42.3  38.00  37.5  36.40  37.00  40.30  37.20  35.20
         2      Algeria  10.7  10.2   9.59   9.1   7.97   8.53   8.68   7.89   7.07
         3       Angola  25.4  25.7  25.70  25.1  24.20  24.30  24.60  24.70  24.80
         4    Argentina  33.7  32.4  26.60  25.1  17.50  20.10  21.60  25.10  24.20

            ...     2008   2009   2010   2011  2012   2013   2014   2015   2016   2017
         0  ...    13.50  13.80  13.50  13.90  14.7  15.40  16.20  17.10  17.90  18.00
         1  ...    22.90  20.60  19.30  19.90  20.6  14.90  13.80  13.30  15.70  16.40
         2  ...     5.69   5.58   5.75   6.69   6.3   6.22   5.06   4.76   5.07   5.05
         3  ...    37.30  41.00  44.30  48.00  47.9  47.80  47.70  47.60  46.60  45.90
         4  ...    27.20  25.90  23.60  24.90  24.2  23.90  23.20  22.90  22.10  21.70

         [5 rows x 28 columns]
```

```
In [11]: # urban growth
         df_urban_growth = df_urban_growth.filter(items=['country', '1991', '1992', '1993', '199
         df_urban_growth.head()
```

```
Out[11]:        country   1991   1992   1993   1994   1995   1996   1997   1998  \
         0  Afghanistan  7.420  8.850  9.180  8.410  7.080  5.640  4.580  4.080
         1      Albania  0.141  0.878  0.856  0.844  0.824  0.812  0.788  0.775
         2      Algeria  3.950  3.810  3.650  3.470  3.290  3.120  2.950  2.820
         3      Andorra  3.700  3.620  3.250  2.520  1.610  0.542 -0.313 -0.560
         4       Angola  5.560  5.700  5.720  5.620  5.430  5.220  5.070  5.020

              1999  ...     2008   2009   2010  2011  2012  2013  2014  2015  2016  \
         0  4.3100  ...    4.020  4.090  4.350  4.64  4.83  4.89  4.76  4.53  4.28
         1  0.7560  ...    1.440  1.470  1.610  1.79  1.85  1.74  1.63  1.46  1.51
         2  2.7800  ...    2.760  2.810  2.870  2.93  2.97  2.96  2.89  2.77  2.64
         3  0.0738  ...    0.858  0.133 -0.622 -1.47 -2.22 -2.64 -2.58 -2.14 -1.54
         4  5.1200  ...    5.630  5.600  5.580  5.55  5.49  5.41  5.31  5.21  5.10

             2017
         0  4.090
         1  1.510
         2  2.520
         3 -0.985
         4  4.990

         [5 rows x 28 columns]
```

```
In [12]: # military expenditure
         df_military_expenditure = df_military_expenditure.filter(items=['country', '1991', '199
         df_military_expenditure.head()
```

```
Out[12]:        country  1991  1992   1993  1994  1995  1996  1997  1998   1999  ...    \
         0  Afghanistan   NaN   NaN    NaN   NaN   NaN   NaN   NaN   NaN    NaN  ...
         1      Albania  5.79  4.45   3.20  2.50  2.10  1.38  1.28  1.24   1.25  ...
         2      Algeria  1.24  2.19   2.56  3.14  2.96  3.09  3.64  3.97   3.76  ...
         3       Angola  8.12  5.25  16.10  5.22  4.28  2.45  5.97  2.62  17.30  ...
         4    Argentina  1.51  1.42   1.42  1.46  1.47  1.24  1.14  1.14   1.22  ...

             2008   2009   2010   2011   2012   2013   2014   2015   2016   2017
         0  2.330  2.060  1.900  1.780  1.140  1.050  1.300  0.993  0.955  0.916
         1  1.980  1.520  1.560  1.530  1.490  1.410  1.350  1.160  1.100  1.250
         2  3.020  3.850  3.520  4.330  4.460  4.840  5.550  6.270  6.420  5.910
         3  3.760  4.390  4.240  3.500  3.640  4.880  5.400  3.520  2.960  2.470
         4  0.763  0.887  0.815  0.764  0.785  0.838  0.878  0.850  0.813  0.891

         [5 rows x 28 columns]
```

```
In [13]: # gdp
         df_gdp = df_gdp.filter(items=['country', '1991', '1992', '1993', '1994', '1995', '1996'
         df_gdp.head()
```

```
Out[13]:        country   1991   1992   1993   1994   1995   1996   1997   1998   1999  \
         0  Afghanistan   1030    950    818    732    881    904    930    956    982
         1      Albania   3230   3010   3320   3620   4130   4530   4070   4460   5100
         2      Algeria   9870   9820   9400   9130   9300   9510   9460   9800   9970
         3      Andorra  28000  27200  26000  25900  26100  27200  29700  30800  31900
         4       Angola   4210   3790   2760   2770   2970   3210   3370   3500   3510

            ...     2008   2009   2010   2011   2012   2013   2014   2015   2016   2017
         0  ...     1300   1530   1610   1660   1840   1810   1780   1750   1740   1800
         1  ...     9150   9530   9930  10200  10400  10500  10700  11000  11400  11900
         2  ...    12700  12600  12900  13000  13200  13300  13500  13700  14000  13800
         3  ...    41400  41700  39000  42000  41900  43700  44900  46600  48200  49800
         4  ...     5980   5910   5900   5910   6000   6190   6260   6230   6030   5940

         [5 rows x 28 columns]

In [14]: # get data for Kuwait
         # birth rate
         df_birth_rate_kuwait = df_birth_rate[df_birth_rate['country'] == "Kuwait"]
         df_birth_rate_kuwait.head()

Out[14]:    country  1991  1992  1993  1994  1995  1996  1997  1998  1999  ...   2008  \
         86  Kuwait  2.79  2.68  2.64  2.66  2.72   2.8  2.86   2.9  2.89  ...   2.34

             2009  2010  2011  2012  2013  2014  2015  2016  2017
         86  2.28  2.22  2.15  2.09  2.05  2.01  1.99  1.97  1.96

         [1 rows x 28 columns]

In [15]: # female employment rate
         df_employment_rate_kuwait = df_employment_rate[df_employment_rate['country'] == "Kuwait
         df_employment_rate_kuwait.head()

Out[15]:    country  1991  1992  1993  1994  1995  1996  1997  1998  1999  ...   2008  \
         83  Kuwait  18.4  18.7  19.0  19.5  19.2  20.2  19.8  19.8  20.0  ...   22.1

             2009  2010  2011  2012  2013  2014  2015  2016  2017
         83  22.9  23.6  20.3  17.8  17.0  16.8  15.7  15.6  16.6

         [1 rows x 28 columns]

In [16]: # urban growth
         df_urban_growth_kuwait = df_urban_growth[df_urban_growth['country'] == "Kuwait"]
         df_urban_growth_kuwait.head()

Out[16]:    country  1991  1992  1993  1994  1995  1996  1997  1998  1999  ...   2008  \
         88  Kuwait -3.08   NaN   NaN   NaN   NaN  1.32  5.01  6.83  6.38  ...   5.79

             2009  2010  2011  2012  2013  2014  2015  2016  2017
         88  6.11  6.18  6.25  6.23  5.82   5.0  3.99  2.94  2.07

         [1 rows x 28 columns]
```

```
In [17]: # military expenditure
         df_military_expenditure_kuwait = df_military_expenditure[df_military_expenditure['count
         df_military_expenditure_kuwait.head()
```

```
Out[17]:    country   1991  1992  1993  1994  1995  1996  1997  1998  1999  ...   2008  \
         79  Kuwait  117.0  31.8  12.4  13.3  13.6  10.3  8.09   8.8  7.59  ...   3.01

             2009  2010  2011  2012  2013  2014  2015  2016  2017
         79  3.97  3.76   3.5  3.41  3.27  3.59  5.01  5.81  5.69

         [1 rows x 28 columns]
```

```
In [18]: # gdp
         df_gdp_kuwait = df_gdp[df_gdp['country'] == "Kuwait"]
         df_gdp_kuwait.head()
```

```
Out[18]:    country   1991   1992   1993   1994   1995   1996   1997   1998   1999  \
         88  Kuwait  18500  50200  68600  74400  81000  80500  78400  75900  70000

             ...     2008   2009   2010   2011   2012   2013   2014   2015   2016  \
         88  ...    93700  81900  75200  77500  77600  74100  70800  69300  67300

              2017
         88  67700

         [1 rows x 28 columns]
```

```
In [19]: # I want to create a table. Year is the index. Columns are birth rate, employment rate,
         # First, make a new table having two columns, year and birth rate.
         inp = [{'year':'1991', 'birth_rate':2.79}]
         df_temp = pd.DataFrame(inp)
         df_temp.head()

         bf_columns = df_birth_rate_kuwait.columns.values.tolist()
         for j in range(1, df_birth_rate_kuwait.shape[1]):
             inp = [{'year': bf_columns[j], 'birth_rate': df_birth_rate_kuwait.iloc[0][j]}]
             df_temp = df_temp.append(pd.DataFrame(inp).round(2))

         # check how the new table looks like
         df_temp.head()
```

```
Out[19]:    birth_rate  year
         0        2.79  1991
         0        2.79  1991
         0        2.68  1992
```

```
0        2.64  1993
0        2.66  1994
```

```
In [20]: # drop the first row, same as the second row
         df_temp.drop_duplicates(inplace=True)
         df_temp.head()
```

```
Out[20]:    birth_rate  year
         0        2.79  1991
         0        2.68  1992
         0        2.64  1993
         0        2.66  1994
         0        2.72  1995
```

```
In [21]: df_temp.shape
```

```
Out[21]: (27, 2)
```

```
In [22]: df_birth_rate_kuwait = df_temp
```

```
In [23]: # Do the same for female employment rate
         inp = [{'year':'1991', 'employment_rate':18.4}]
         df_temp = pd.DataFrame(inp)
         df_temp.head()


         bf_columns = df_employment_rate_kuwait.columns.values.tolist()
         for j in range(1, df_employment_rate_kuwait.shape[1]):
             inp = [{'year': bf_columns[j], 'employment_rate': df_employment_rate_kuwait.iloc[0]
             df_temp = df_temp.append(pd.DataFrame(inp).round(2))

         df_temp.drop_duplicates(inplace=True)
         df_employment_rate_kuwait = df_temp
         df_employment_rate_kuwait.head()
```

```
Out[23]:    employment_rate  year
         0             18.4  1991
         0             18.7  1992
         0             19.0  1993
         0             19.5  1994
         0             19.2  1995
```

```
In [24]: # Repeat for urban growth
         inp = [{'year':'1991', 'urban_growth':-3.08}]
         df_temp = pd.DataFrame(inp)

         bf_columns = df_urban_growth_kuwait.columns.values.tolist()
         for j in range(1, df_urban_growth_kuwait.shape[1]):
             inp = [{'year': bf_columns[j], 'urban_growth': df_urban_growth_kuwait.iloc[0][j]}]
```

```
            df_temp = df_temp.append(pd.DataFrame(inp).round(3))

        df_temp.drop_duplicates(inplace=True)
        df_urban_growth_kuwait = df_temp
        df_urban_growth_kuwait.head()
```

```
Out[24]:    urban_growth  year
        0         -3.08  1991
        0           NaN  1992
        0           NaN  1993
        0           NaN  1994
        0           NaN  1995
```

```
In [25]: # Repeat for military expenditure
        inp = [{'year':'1991', 'military_expenditure':117.0}]
        df_temp = pd.DataFrame(inp)

        bf_columns = df_military_expenditure_kuwait.columns.values.tolist()
        for j in range(1, df_military_expenditure_kuwait.shape[1]):
            inp = [{'year': bf_columns[j], 'military_expenditure': df_military_expenditure_kuwa
            df_temp = df_temp.append(pd.DataFrame(inp).round(2))

        df_temp.drop_duplicates(inplace=True)
        df_military_expenditure_kuwait = df_temp
        df_military_expenditure_kuwait.head()
```

```
Out[25]:    military_expenditure  year
        0                  117.0  1991
        0                   31.8  1992
        0                   12.4  1993
        0                   13.3  1994
        0                   13.6  1995
```

```
In [26]: # Repeat for gdp
        inp = [{'year':'1991', 'gdp':18500}]
        df_temp = pd.DataFrame(inp)

        bf_columns = df_gdp_kuwait.columns.values.tolist()
        for j in range(1, df_gdp_kuwait.shape[1]):
            inp = [{'year': bf_columns[j], 'gdp': df_gdp_kuwait.iloc[0][j]}]
            df_temp = df_temp.append(pd.DataFrame(inp).round(1))

        df_temp.drop_duplicates(inplace=True)
        df_gdp_kuwait = df_temp
        df_gdp_kuwait.head()
```

```
Out[26]:      gdp  year
        0  18500  1991
        0  50200  1992
```

```
0  68600  1993
0  74400  1994
0  81000  1995
```

Tip: You should *not* perform too many operations in each cell. Create cells freely to explore your data. One option that you can take with this project is to do a lot of explorations in an initial notebook. These don't have to be organized, but make sure you use enough comments to understand the purpose of each code cell. Then, after you're done with your analysis, create a duplicate notebook where you will trim the excess and organize your steps so that you have a flowing, cohesive report.

Tip: Make sure that you keep your reader informed on the steps that you are taking in your investigation. Follow every code cell, or every set of related code cells, with a markdown cell to describe to the reader what was found in the preceding cell(s). Try to make it so that the reader can then understand what they will be seeing in the following cell(s).

1.1.2 Data Cleaning (Replace this with more specific notes!)

```
In [27]: # Merge all the tables.
         df_merge = pd.merge(df_birth_rate_kuwait, df_employment_rate_kuwait, on = 'year')
         df_merge = pd.merge(df_merge, df_urban_growth_kuwait, on = 'year')
         df_merge = pd.merge(df_merge, df_military_expenditure_kuwait, on = 'year')
         df_merge = pd.merge(df_merge, df_gdp_kuwait, on = 'year')
         df_merge.head()
```

```
Out[27]:    birth_rate  year  employment_rate  urban_growth  military_expenditure  \
         0        2.79  1991             18.4         -3.08                 117.0
         1        2.68  1992             18.7           NaN                  31.8
         2        2.64  1993             19.0           NaN                  12.4
         3        2.66  1994             19.5           NaN                  13.3
         4        2.72  1995             19.2           NaN                  13.6

              gdp
         0  18500
         1  50200
         2  68600
         3  74400
         4  81000
```

```
In [28]: # investigate the table
         df_merge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 27 entries, 0 to 26
Data columns (total 6 columns):
birth_rate              27 non-null float64
year                    27 non-null object
employment_rate         27 non-null float64
```

```
urban_growth            23 non-null float64
military_expenditure    27 non-null float64
gdp                     27 non-null int64
dtypes: float64(4), int64(1), object(1)
memory usage: 1.5+ KB
```

```
In [29]: # urban growth is missing four values. I am filling it with the mean value of the avera
         df_merge.fillna(df_urban_growth_kuwait.mean().round(2), inplace = True)
         df_merge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 27 entries, 0 to 26
Data columns (total 6 columns):
birth_rate              27 non-null float64
year                    27 non-null object
employment_rate         27 non-null float64
urban_growth            27 non-null float64
military_expenditure    27 non-null float64
gdp                     27 non-null int64
dtypes: float64(4), int64(1), object(1)
memory usage: 1.5+ KB
```

```
In [30]: # all cells have value, df_merge table is clean
         df_merge.head()
```

```
Out[30]:    birth_rate  year  employment_rate  urban_growth  military_expenditure  \
         0        2.79  1991             18.4         -3.08                 117.0
         1        2.68  1992             18.7          3.98                  31.8
         2        2.64  1993             19.0          3.98                  12.4
         3        2.66  1994             19.5          3.98                  13.3
         4        2.72  1995             19.2          3.98                  13.6

              gdp
         0  18500
         1  50200
         2  68600
         3  74400
         4  81000
```

## Exploratory Data Analysis

> **Tip**: Now that you've trimmed and cleaned your data, you're ready to move on to exploration. Compute statistics and create visualizations with the goal of addressing the research questions that you posed in the Introduction section. It is recommended that you be systematic with your approach. Look at one variable at a time, and then follow it up by looking at relationships between variables.

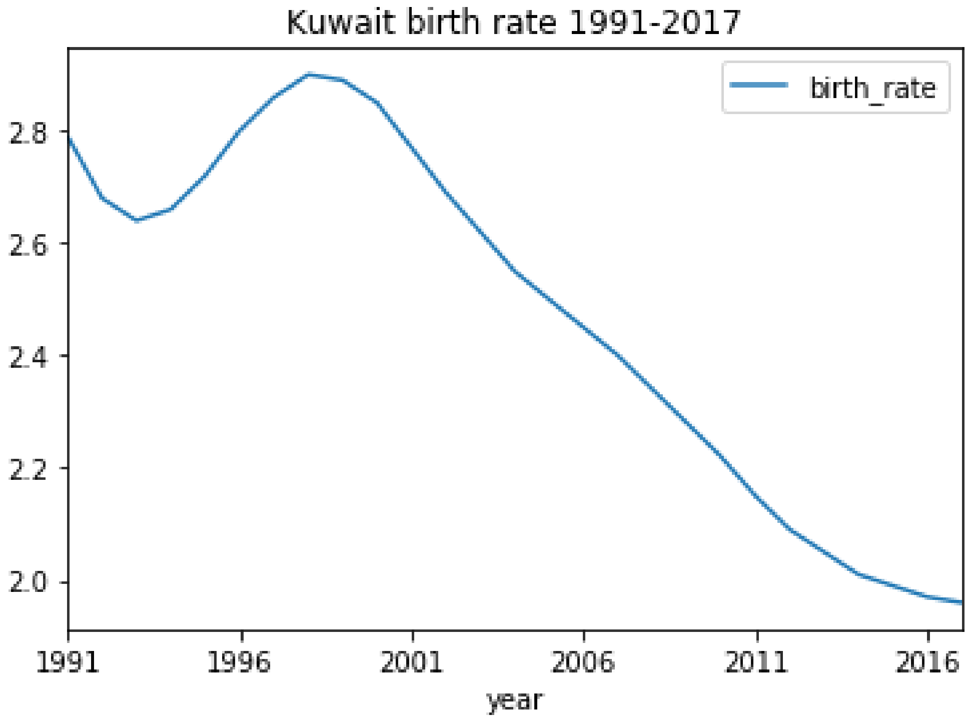
1.1.3 What is Kuwait birth rate associating?

```
In [32]: # Let's look at the overall stat first
         df_merge.describe()
```

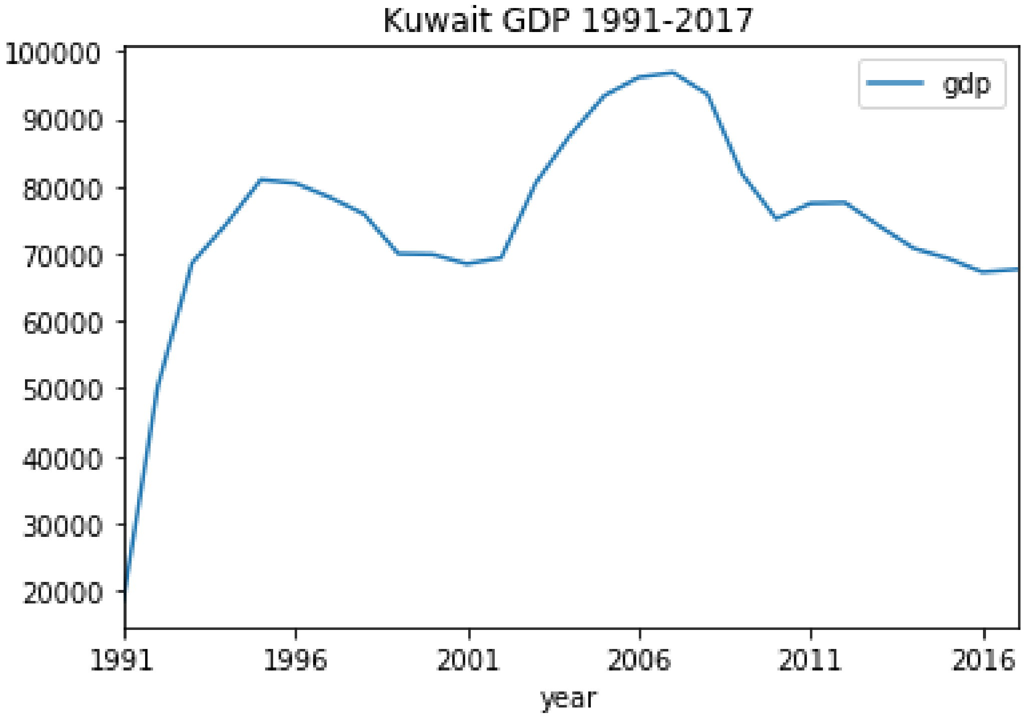
```
Out[32]:        birth_rate  employment_rate  urban_growth  military_expenditure  \
         count   27.000000        27.000000     27.000000             27.000000
         mean     2.475185        19.977778      3.981852             11.480000
         std      0.322507         2.386232      2.202907             21.862846
         min      1.960000        15.600000     -3.080000              3.010000
         25%      2.185000        18.550000      2.885000              3.675000
         50%      2.550000        20.000000      3.990000              5.810000
         75%      2.745000        22.000000      5.805000              8.445000
         max      2.900000        23.800000      6.830000            117.000000

                         gdp
         count     27.000000
         mean   74633.333333
         std    15295.046961
         min    18500.000000
         25%    69350.000000
         50%    75200.000000
         75%    80750.000000
         max    96900.000000
```

Observation: the maximum value of birth rate is 2.9 and minimum value is 1.96. In average, 2.55 children per women. Employment rate falls between 15% and 23.8%. The highest GDP is $96,900 per person. The lowest is $18,500 per person. Assume the lowest point is in 1991, when the war just finished.

```
In [33]: # Kuwait's birth rate from 1991 to 2017.

         df_birth_rate_kuwait.plot.line(x= 'year', y='birth_rate', title="Kuwait birth rate 1991
```

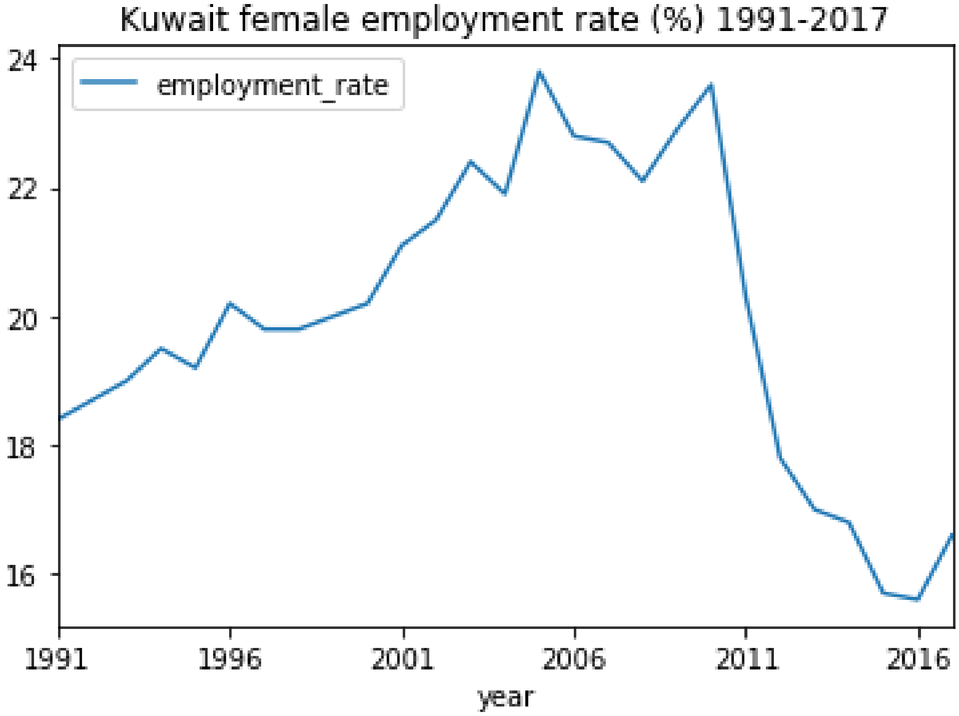


```
In [35]: # Kuwait's GDP from 1991 to 2017.

         df_gdp_kuwait.plot.line(x= 'year', y='gdp', title="Kuwait GDP 1991-2017");
```



Observation: the lowest GDP was in 1991, right after the war. It reached its highest value in 2006 then started to go down.

```
In [38]: # Kuwait's Female employment rate from 1991 to 2017.

         df_employment_rate_kuwait.plot.line(x= 'year', y='employment_rate', title="Kuwait femal
```



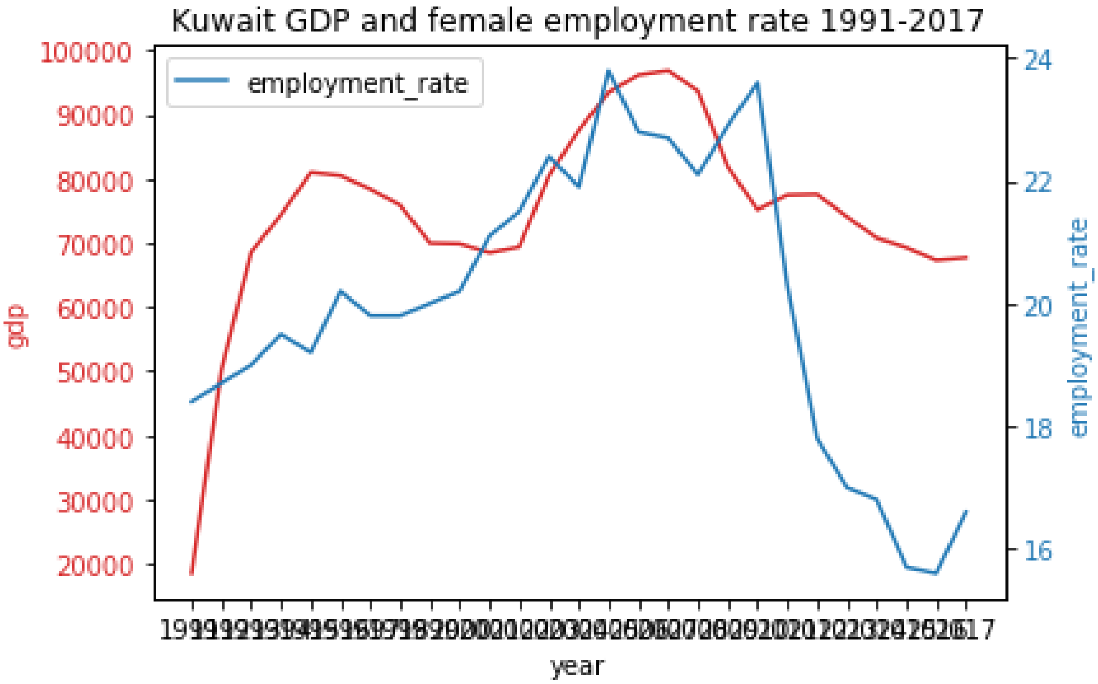
```
In [41]: df_employment_rate_kuwait

Out[41]:    employment_rate  year
         0             18.4  1991
         0             18.7  1992
         0             19.0  1993
         0             19.5  1994
         0             19.2  1995
         0             20.2  1996
         0             19.8  1997
         0             19.8  1998
         0             20.0  1999
         0             20.2  2000
         0             21.1  2001
```

```
0               21.5  2002
0               22.4  2003
0               21.9  2004
0               23.8  2005
0               22.8  2006
0               22.7  2007
0               22.1  2008
0               22.9  2009
0               23.6  2010
0               20.3  2011
0               17.8  2012
0               17.0  2013
0               16.8  2014
0               15.7  2015
0               15.6  2016
0               16.6  2017
```

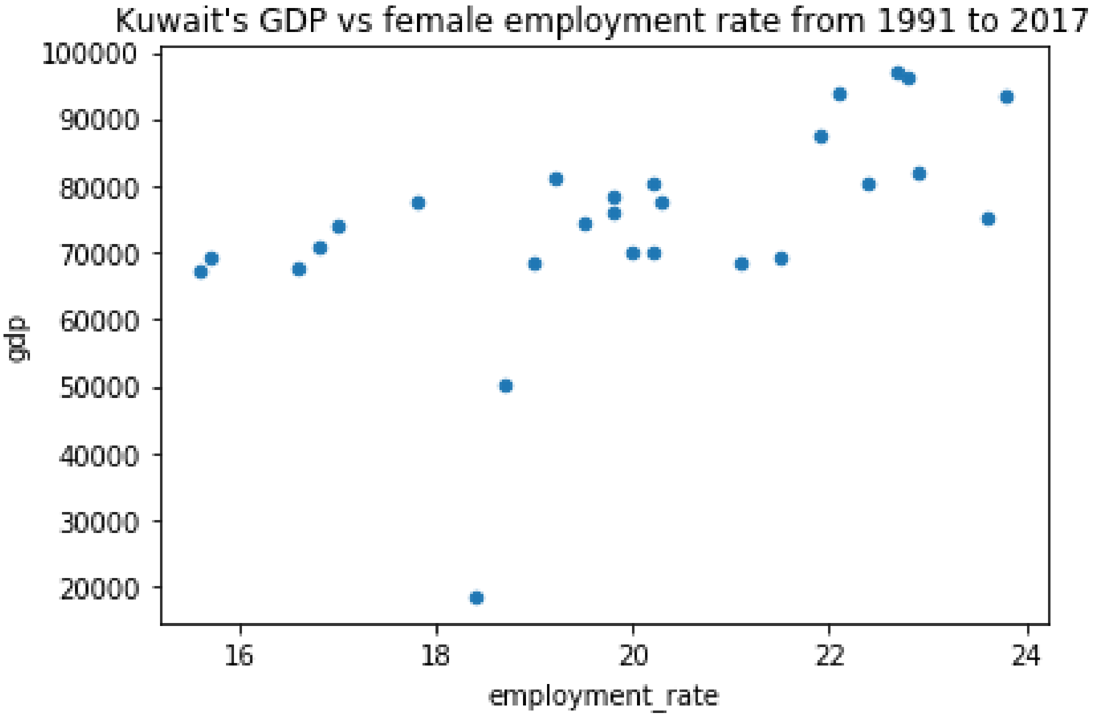
Observation: The female employment rate kept increasing until 2006. After 2006, it went down but started to climb up again in 2008 until 2011, after when the rate had a sharp decrease, employment rate dropped 33.8% within 6 years.

Observation: the overall birth rate in Kuwait is decreasing. However, from 1993 to 1998, there was a slight increase.

```
In [43]: # I want to see if the employment rate is associated with GDP
         df_merge_er_gdp = pd.merge(df_employment_rate_kuwait, df_gdp_kuwait, on = 'year')

         fig, ax1 = plt.subplots()

         color = 'tab:red'
         ax1.set_xlabel('year')
         ax1.set_ylabel('gdp', color=color)
         ax1.plot(df_merge_er_gdp['year'], df_merge_er_gdp['gdp'], color=color)
         ax1.tick_params(axis='y', labelcolor=color)
         ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

         color = 'tab:blue'
         ax2.set_ylabel('employment_rate', color=color)  # we already handled the x-label with a
         ax2.plot(df_merge_er_gdp['year'], df_merge_er_gdp['employment_rate'], color=color)
         ax2.tick_params(axis='y', labelcolor=color)

         plt.title("Kuwait GDP and female employment rate 1991-2017")   # add a title
         plt.legend()
         plt.show()
```

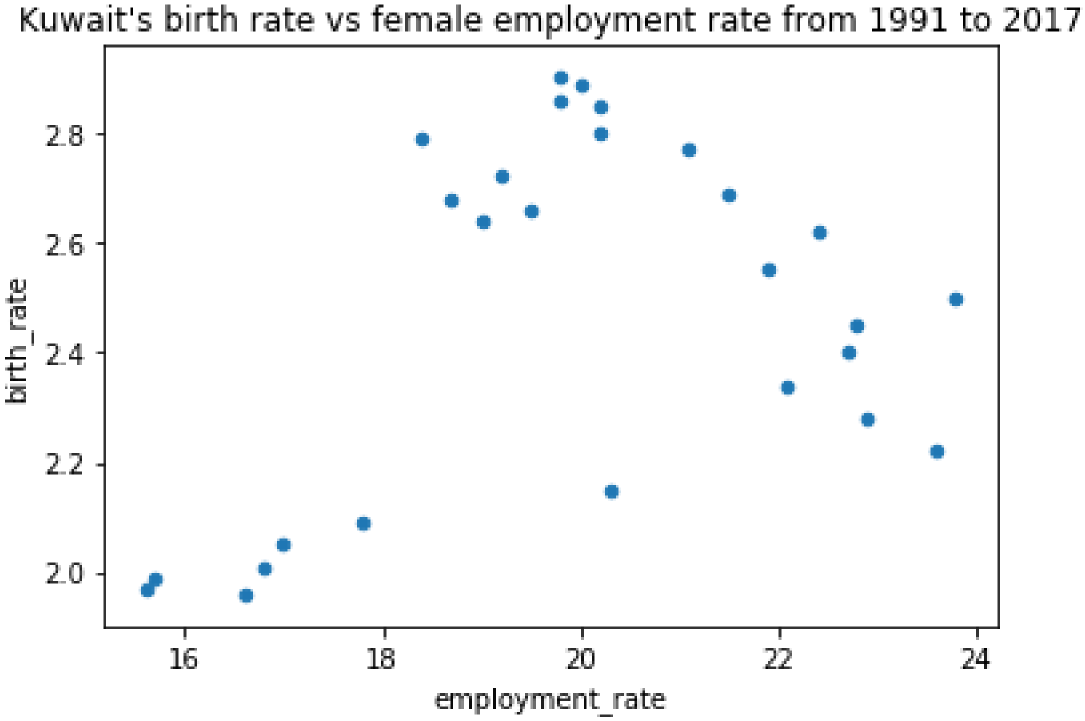


```
In [44]: # Plot a scatter gram
         df_merge.plot(kind = 'scatter', x = 'employment_rate', y='gdp', title="Kuwait's GDP vs
```



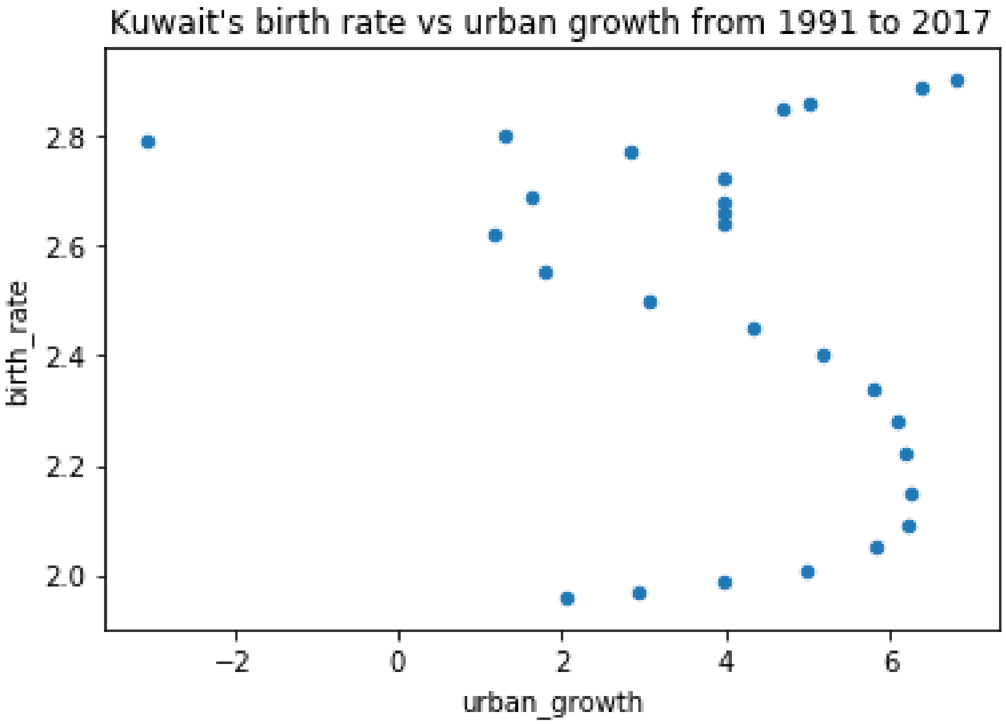
Observation: GDP and female employment rate have positive corelation. More women are employed when GDP is higher.

```
In [45]: # Now let's see if Kuwait's birth rate and female employment rate are related.
         #
         df_merge.plot(kind = 'scatter', x = 'employment_rate', y='birth_rate', title="Kuwait's
```



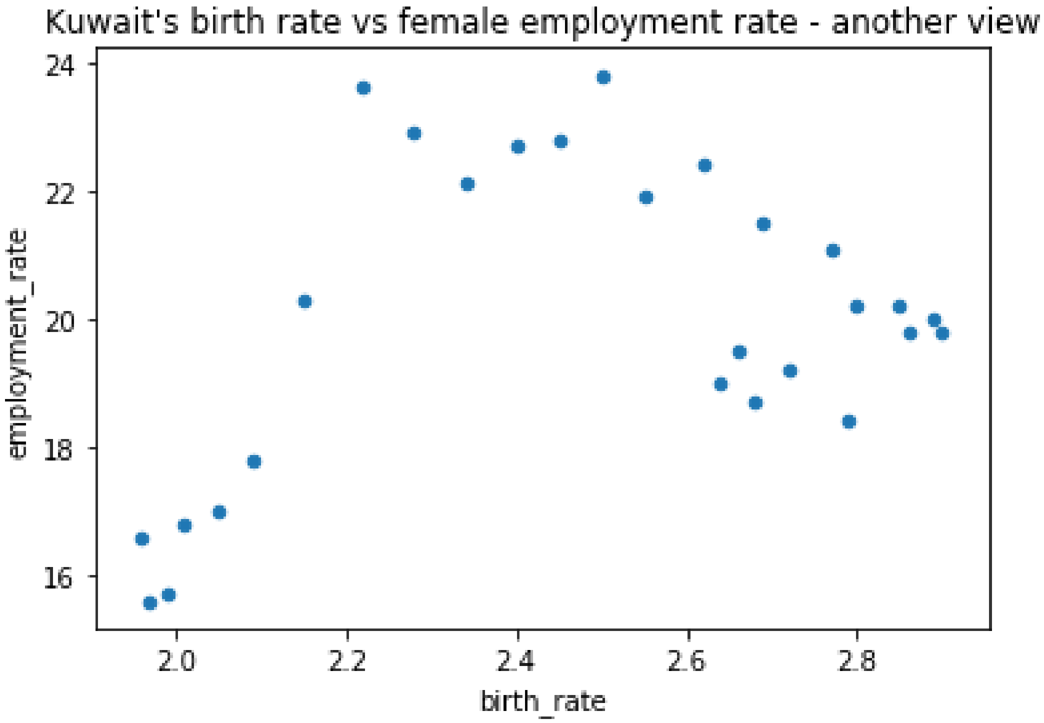
Observation: when the female employment rate is less than 20%, birth rate has positive corelation with employment rate. When the employment rate is greater than 20%, birth rate has negative corelation with employment rate.

```
In [163]: # Kuwait's birth rate vs urban growth from 1991-2017.
          #
          df_merge.plot(kind = 'scatter', x = 'urban_growth', y='birth_rate', title="Kuwait's bi
```



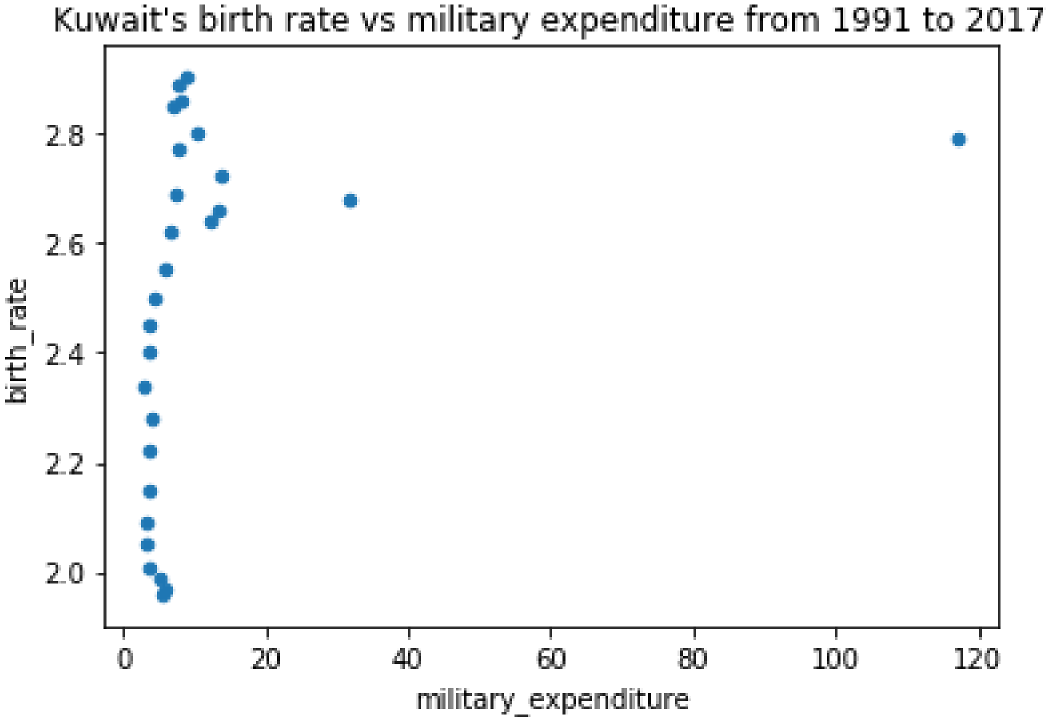
I cannot tell from the graph above. Let me flip x and y

```
In [33]: df_merge.plot(kind = 'scatter', x='birth_rate', y = 'employment_rate', title="Kuwait's
```



Observation: it is still not very clear if birth rate is relating to urban growth.

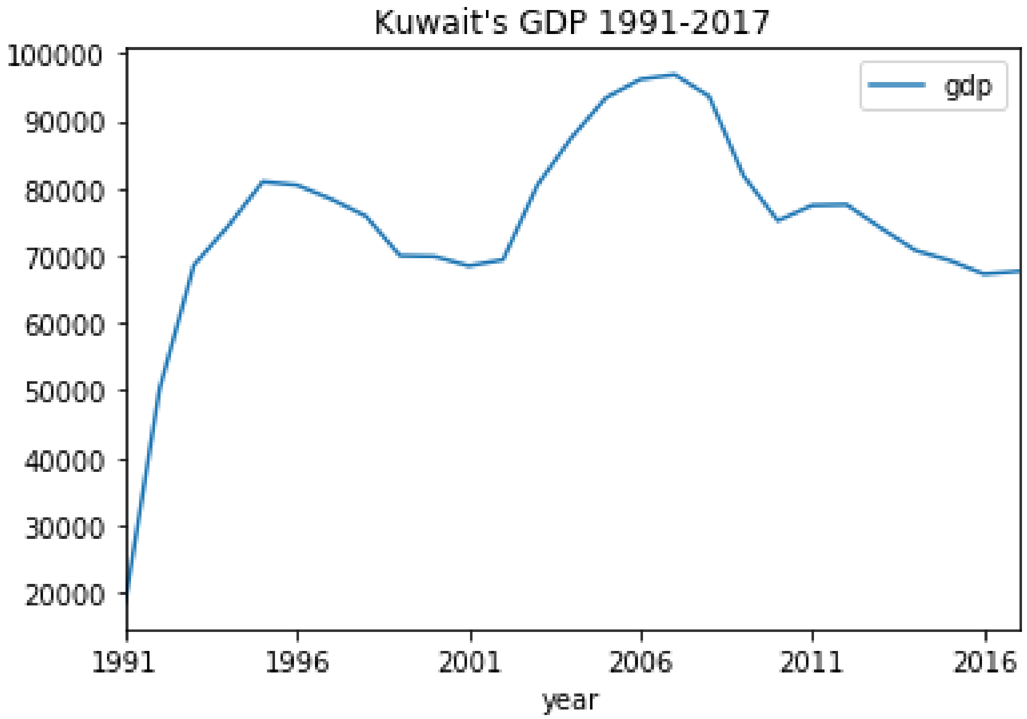
```
In [164]: # Kuwait's birth rate vs military expenditure from 1991 to 2017.
          #
          df_merge.plot(kind = 'scatter', x = 'military_expenditure', y='birth_rate', title="Kuw
```



Observation: military expenditure is stable most of the time. It has no influence on birth rate.

1.1.4 Is birth rate relating to GDP?

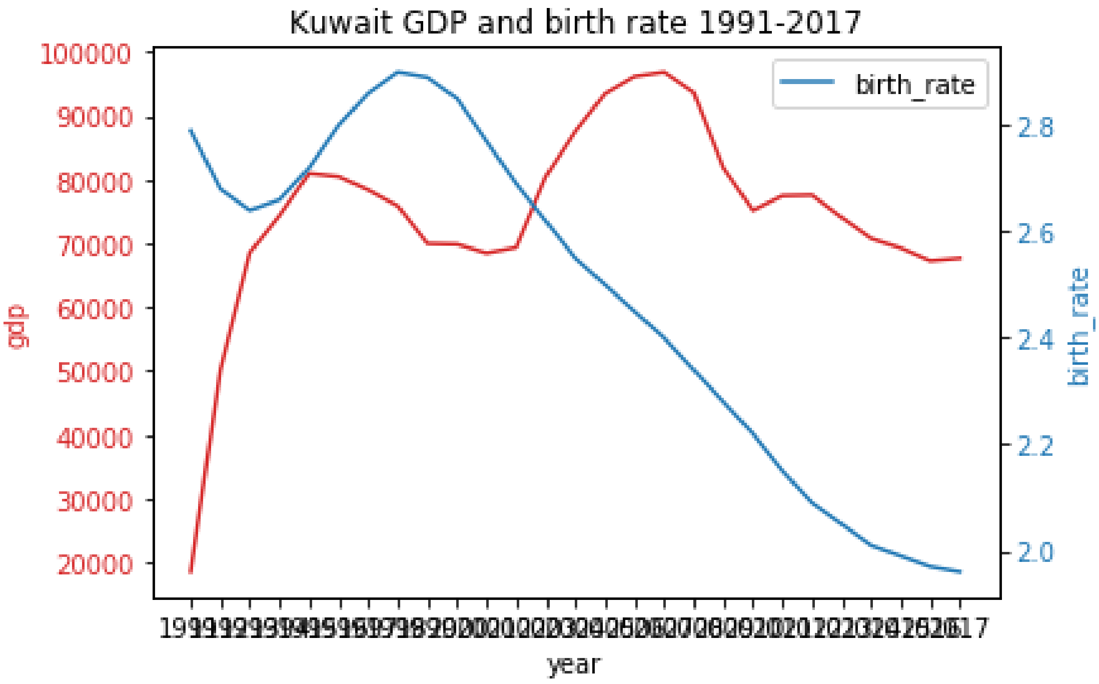
```
In [177]: # Plot Kuwait GDP
          df_gdp_kuwait.plot.line(x= 'year', y='gdp', title="Kuwait's GDP 1991-2017");# Continue
```



Observation: After Feb, 1991, Kuwait was liberated. Kuwait spent more than 5 billion to repair oil infrastructure damaged during the Gulf war. The economy recovered quickly.

```
In [39]: # next I would like to make a plot with both GDP and birth rate to observe if they are

         # Make a table just has birth rate and gdp
         df_merge_br_gdp = pd.merge(df_birth_rate_kuwait, df_gdp_kuwait, on = 'year')
         df_merge_br_gdp.head()

         fig, ax1 = plt.subplots()

         color = 'tab:red'
         ax1.set_xlabel('year')
         ax1.set_ylabel('gdp', color=color)
         ax1.plot(df_merge_br_gdp['year'], df_merge_br_gdp['gdp'], color=color)
         ax1.tick_params(axis='y', labelcolor=color)
         ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

         color = 'tab:blue'
         ax2.set_ylabel('birth_rate', color=color)  # we already handled the x-label with ax1
         ax2.plot(df_merge_br_gdp['year'], df_merge_br_gdp['birth_rate'], color=color)
         ax2.tick_params(axis='y', labelcolor=color)
```

```
plt.title("Kuwait GDP and birth rate 1991-2017")    # add a title
plt.legend()
plt.show()
```



## Conclusions

The change of birth rate results from multiple factors. From the many factors, I chose four factors to observe if they have any impact on birth rate. The result shows birth rate is associated with female employment rate, which is tied with GDP. There is no obvious corelation between birth rate and urban growth. Nor does it relate to military expenditure. GDP itself plays very little in determining birth rate. After 1998, birth rate has decreases drastically regardless the flutruation of GDP.

1.2 Limitations

I filled the null value in urban growth with the mean value. There should be a better value other than the mean value.

1.3 Submitting your Project

Before you submit your project, you need to create a .html or .pdf version of this notebook in the workspace here. To do that, run the code cell below. If it worked correctly, you should get a return code of 0, and you should see the generated .html file in the workspace directory (click on the orange Jupyter icon in the upper left).

Alternatively, you can download this report as .html via the **File** > **Download as** submenu, and then manually upload it into the workspace directory by clicking on the orange Jupyter icon in the upper left, then using the Upload button.

Once you've done this, you can submit your project by clicking on the "Submit Project" button in the lower right here. This will create and submit a zip file with this .ipynb doc and the .html or .pdf version you created. Congratulations!

```
In [1]: from subprocess import call
        call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])

Out[1]: 0

In [ ]:
```