In [1]:
```
# To call pandas library and use its built-in modules, need to import the lib
import pandas as pd
```

In [2]:
```
# Below the databases are read turn them from files into DataFrames.
fert = pd.read_csv('gapminder_total_fertility.csv', index_col=0)
life = pd.read_excel('gapminder_lifeexpectancy.xlsx', index_col=0)
popu = pd.read_excel('gapminder_population.xlsx', index_col=0)
cont = pd.read_csv('continents.csv', sep=';', index_col=1)

# To explore information of dataframes
fert.info()
life.info()
popu.info()
cont.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 260 entries, Abkhazia to Åland
Columns: 216 entries, 1800 to 2015
dtypes: float64(216)
memory usage: 440.8+ KB
<class 'pandas.core.frame.DataFrame'>
Index: 260 entries, Abkhazia to South Sudan
Columns: 217 entries, 1800 to 2016
dtypes: float64(217)
memory usage: 442.8+ KB
<class 'pandas.core.frame.DataFrame'>
Index: 275 entries, Abkhazia to West Bank
Data columns (total 91 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   1800         229 non-null    float64
 1   1810         229 non-null    float64
 2   1820         229 non-null    float64
 3   1830         229 non-null    float64
 4   1840         229 non-null    float64
 5   1850         229 non-null    float64
 6   1860         229 non-null    float64
 7   1870         229 non-null    float64
 8   1880         229 non-null    float64
 9   1890         229 non-null    float64
 10  1900         229 non-null    float64
 11  1910         229 non-null    float64
 12  1920         229 non-null    float64
 13  1930         229 non-null    float64
 14  1940         229 non-null    float64
 15  1950         256 non-null    float64
 16  1951         256 non-null    float64
 17  1952         256 non-null    float64
 18  1953         256 non-null    float64
 19  1954         256 non-null    float64
 20  1955         256 non-null    float64
 21  1956         256 non-null    float64
 22  1957         256 non-null    float64
 23  1958         256 non-null    float64
 24  1959         256 non-null    float64
 25  1960         256 non-null    float64
 26  1961         256 non-null    float64
 27  1962         256 non-null    float64
 28  1963         256 non-null    float64
 29  1964         256 non-null    float64
 30  1965         256 non-null    float64
 31  1966         256 non-null    float64
 32  1967         256 non-null    float64
 33  1968         256 non-null    float64
 34  1969         256 non-null    float64
```

```
 35  1970          256 non-null    float64
 36  1971          256 non-null    float64
 37  1972          256 non-null    float64
 38  1973          256 non-null    float64
 39  1974          256 non-null    float64
 40  1975          256 non-null    float64
 41  1976          256 non-null    float64
 42  1977          256 non-null    float64
 43  1978          256 non-null    float64
 44  1979          256 non-null    float64
 45  1980          256 non-null    float64
 46  1981          256 non-null    float64
 47  1982          256 non-null    float64
 48  1983          256 non-null    float64
 49  1984          256 non-null    float64
 50  1985          256 non-null    float64
 51  1986          256 non-null    float64
 52  1987          256 non-null    float64
 53  1988          256 non-null    float64
 54  1989          256 non-null    float64
 55  1990          256 non-null    float64
 56  1991          256 non-null    float64
 57  1992          256 non-null    float64
 58  1993          256 non-null    float64
 59  1994          256 non-null    float64
 60  1995          256 non-null    float64
 61  1996          256 non-null    float64
 62  1997          256 non-null    float64
 63  1998          256 non-null    float64
 64  1999          256 non-null    float64
 65  2000          256 non-null    float64
 66  2001          256 non-null    float64
 67  2002          256 non-null    float64
 68  2003          256 non-null    float64
 69  2004          256 non-null    float64
 70  2005          256 non-null    float64
 71  2006          256 non-null    float64
 72  2007          256 non-null    float64
 73  2008          256 non-null    float64
 74  2009          238 non-null    float64
 75  2010          238 non-null    float64
 76  2011          233 non-null    float64
 77  2012          233 non-null    float64
 78  2013          233 non-null    float64
 79  2014          231 non-null    float64
 80  2015          231 non-null    float64
 81  Unnamed: 82   0 non-null      float64
 82  Unnamed: 83   0 non-null      float64
 83  Unnamed: 84   0 non-null      float64
 84  Unnamed: 85   0 non-null      float64
 85  Unnamed: 86   0 non-null      float64
 86  Unnamed: 87   0 non-null      float64
 87  Unnamed: 88   0 non-null      float64
 88  Unnamed: 89   0 non-null      float64
 89  Unnamed: 90   0 non-null      float64
 90  Unnamed: 91   0 non-null      float64
dtypes: float64(91)
memory usage: 197.7+ KB
<class 'pandas.core.frame.DataFrame'>
Index: 194 entries, Algeria to Venezuela
Data columns (total 1 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   continent  194 non-null    object
dtypes: object(1)
memory usage: 3.0+ KB
```

In [18]:

```
# To check the shapes
display(fert.shape)
```

```
display(life.shape)
display(popu.shape)
display(cont.shape)

# To illustrate first two rows
display(fert.head(2))
display(life.head(2))
display(popu.head(2))
display(cont.head(2))
```

```
(56160, 3)
(56420, 3)
(275, 91)
(194, 2)
```

	country	year	fertility_rate
0	Abkhazia	1800	NaN
1	Afghanistan	1800	7.0

	country	year	Life expectancy
0	Abkhazia	1800	NaN
1	Afghanistan	1800	28.21

country	1800	1810	1820	1830	1840	1850	1860
Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Afghanistan	3280000.0	3280000.0	3323519.0	3448982.0	3625022.0	3810047.0	3973968.0

2 rows × 91 columns

	country	continent
0	Algeria	Africa
1	Angola	Africa

In [19]:

```
# To check columns of Dataframes
display(fert.columns)
display(life.columns)
display(popu.columns)
display(cont.columns)
```

```
Index(['country', 'year', 'fertility_rate'], dtype='object')
Index(['country', 'year', 'Life expectancy'], dtype='object')
Index([         1800,          1810,          1820,          1830,
                1840,          1850,          1860,          1870,
                1880,          1890,          1900,          1910,
                1920,          1930,          1940,          1950,
                1951,          1952,          1953,          1954,
                1955,          1956,          1957,          1958,
                1959,          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, 'Unnamed: 82', 'Unnamed: 83', 'Unnamed: 84',
       'Unnamed: 85', 'Unnamed: 86', 'Unnamed: 87', 'Unnamed: 88',
       'Unnamed: 89', 'Unnamed: 90', 'Unnamed: 91'],
      dtype='object')
Index(['country', 'continent'], dtype='object')
```

In [5]:

```
# To delete columns starting with 'Unnamed' in Population dataframe
popu_clean = popu.drop(popu.iloc[:, -10:], axis=1)
```

In [6]:

```
# To change the type of fertility's column and Population's column from objec
fert.columns = fert.columns.astype(int)
popu_clean.columns = popu_clean.columns.astype(int)
```

In [7]:

```
# To have one column with identical label in all dataframes
display(fert.index)
fert.index.name = 'country'
display(fert.tail(3))

display(life.index)
life.index.name = 'country'
display(life.tail(3))

display(popu_clean.index)
popu_clean.index.name = 'country'
display(popu_clean.tail(3))

# 'continent' dataframe has already a column labeled with 'country'
display(cont.index)
display(cont.head(3))
```

```
Index(['Abkhazia', 'Afghanistan', 'Akrotiri and Dhekelia', 'Albania',
       'Algeria', 'American Samoa', 'Andorra', 'Angola', 'Anguilla',
       'Antigua and Barbuda',
       ...
       'Vietnam', 'Virgin Islands (U.S.)', 'North Yemen (former)',
       'South Yemen (former)', 'Yemen', 'Yugoslavia', 'Zambia', 'Zimbabwe',
       'Åland', 'Åland'],
      dtype='object', name='Total fertility rate', length=260)
```

country	1800	1801	1802	1803	1804	1805	1806	1807	1808	1809	...	2006	2007
Zimbabwe	6.75	6.75	6.75	6.75	6.75	6.75	6.75	6.75	6.75	6.75	...	3.94	3.9
Åland	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
Åland	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN

3 rows × 216 columns

```
Index(['Abkhazia', 'Afghanistan', 'Akrotiri and Dhekelia', 'Albania',
       'Algeria', 'American Samoa', 'Andorra', 'Angola', 'Anguilla',
       'Antigua and Barbuda',
       ...
       'Vietnam', 'Virgin Islands (U.S.)', 'North Yemen (former)',
       'South Yemen (former)', 'Yemen', 'Yugoslavia', 'Zambia', 'Zimbabwe',
```

```
       'Åland', 'South Sudan'],
      dtype='object', name='Life expectancy', length=260)
```

country	1800	1801	1802	1803	1804	1805	1806	1807	1808	1809	...	2007	20
Zimbabwe	33.70	33.70	33.70	33.70	33.70	33.70	33.70	33.70	33.70	33.70	...	46.4	4
Åland	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	N
South Sudan	26.67	26.67	26.67	26.67	26.67	26.67	26.67	26.67	26.67	26.67	...	55.5	5

3 rows × 217 columns

```
Index(['Abkhazia', 'Afghanistan', 'Akrotiri and Dhekelia', 'Albania',
       'Algeria', 'American Samoa', 'Andorra', 'Angola', 'Anguilla',
       'Antigua and Barbuda',
       ...
       'British Indian Ocean Territory', 'Clipperton',
       'French Southern and Antarctic Lands', 'Gaza Strip',
       'Heard and McDonald Islands', 'Northern Marianas',
       'South Georgia and the South Sandwich Islands',
       'US Minor Outlying Islands', 'Virgin Islands', 'West Bank'],
      dtype='object', name='Total population', length=275)
```

country	1800	1810	1820	1830	1840	1850	1860	1870	1880	1890	...	2006	2007	2
US Minor Outlying Islands	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
Virgin Islands	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
West Bank	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	

3 rows × 81 columns

```
Index(['Algeria', 'Angola', 'Benin', 'Botswana', 'Burkina', 'Burundi',
       'Cameroon', 'Cape Verde', 'Central African Republic', 'Chad',
       ...
       'Brazil', 'Chile', 'Colombia', 'Ecuador', 'Guyana', 'Paraguay', 'Peru',
       'Suriname', 'Uruguay', 'Venezuela'],
      dtype='object', name='country', length=194)
```

country	continent
Algeria	Africa
Angola	Africa
Benin	Africa

In [8]:

```
# To move the row index into a column
fert = fert.reset_index()
display(fert.head(3))

life = life.reset_index()
display(life.head(3))

popu_clean = popu_clean.reset_index()
display(popu_clean.head(3))
```

```
cont = cont.reset_index()
display(cont.head(3))
```

	country	1800	1801	1802	1803	1804	1805	1806	1807	1808	...	2006	2007	2
0	Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
1	Afghanistan	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	...	6.7	6.46	
2	Akrotiri and Dhekelia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	

3 rows × 217 columns

	country	1800	1801	1802	1803	1804	1805	1806	1807	1808	...	2007	2008	2
0	Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
1	Afghanistan	28.21	28.2	28.19	28.18	28.17	28.16	28.15	28.14	28.13	...	52.4	52.8	
2	Akrotiri and Dhekelia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	

3 rows × 218 columns

	country	1800	1810	1820	1830	1840	1850	1860
0	Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Afghanistan	3280000.0	3280000.0	3323519.0	3448982.0	3625022.0	3810047.0	3973968.0
2	Akrotiri and Dhekelia	NaN	NaN	NaN	NaN	NaN	NaN	NaN

3 rows × 82 columns

	country	continent
0	Algeria	Africa
1	Angola	Africa
2	Benin	Africa

In [9]:

```
# To convert wide format to long format (continent dataframe is already in lo
fert = fert.melt(id_vars='country', var_name='year', value_name='fertility_ra
life = life.melt(id_vars='country', var_name='year', value_name='Life expecta
popu_clean = popu_clean.melt(id_vars='country', var_name='year', value_name='

display(fert)
display(life)
display(popu_clean)
display(cont)
```

	country	year	fertility_rate
0	Abkhazia	1800	NaN
1	Afghanistan	1800	7.00
2	Akrotiri and Dhekelia	1800	NaN
3	Albania	1800	4.60

	country	year	fertility_rate
4	Algeria	1800	6.99
...	...	...	...
56155	Yugoslavia	2015	NaN
56156	Zambia	2015	5.59
56157	Zimbabwe	2015	3.35
56158	Åland	2015	NaN
56159	Åland	2015	NaN

56160 rows × 3 columns

	country	year	Life expectancy
0	Abkhazia	1800	NaN
1	Afghanistan	1800	28.21
2	Akrotiri and Dhekelia	1800	NaN
3	Albania	1800	35.40
4	Algeria	1800	28.82
...	...	...	...
56415	Yugoslavia	2016	NaN
56416	Zambia	2016	57.10
56417	Zimbabwe	2016	61.69
56418	Åland	2016	NaN
56419	South Sudan	2016	56.10

56420 rows × 3 columns

	country	year	Total population
0	Abkhazia	1800	NaN
1	Afghanistan	1800	3280000.0
2	Akrotiri and Dhekelia	1800	NaN
3	Albania	1800	410445.0
4	Algeria	1800	2503218.0
...	...	...	...
22270	Northern Marianas	2015	NaN
22271	South Georgia and the South Sandwich Islands	2015	NaN
22272	US Minor Outlying Islands	2015	NaN
22273	Virgin Islands	2015	NaN
22274	West Bank	2015	NaN

22275 rows × 3 columns

	country	continent

	country	continent
0	Algeria	Africa
1	Angola	Africa
2	Benin	Africa
3	Botswana	Africa
4	Burkina	Africa
...	...	...
189	Paraguay	South America
190	Peru	South America
191	Suriname	South America
192	Uruguay	South America
193	Venezuela	South America

194 rows × 2 columns

In [10]:
```
"""To merge all four dataframes with common column(s) with same lebels.
   Below first fertility and lifeexpectancy are merged then the resulting dat
   to population dataframe, afterall continent dataframe is merged to resulti
df1 = fert.merge(life,on=['country','year']).merge(popu_clean,on=['country','
df1
```

Out[10]:

	country	year	fertility_rate	Life expectancy	Total population	continent
0	Afghanistan	1800	7.00	28.21	3280000.0	Asia
1	Afghanistan	1810	7.00	28.11	3280000.0	Asia
2	Afghanistan	1820	7.00	28.01	3323519.0	Asia
3	Afghanistan	1830	7.00	27.90	3448982.0	Asia
4	Afghanistan	1840	7.00	27.80	3625022.0	Asia
...	...	...	...	...	...	...
14170	Zimbabwe	2011	3.64	51.60	14255592.0	Africa
14171	Zimbabwe	2012	3.56	54.20	14565482.0	Africa
14172	Zimbabwe	2013	3.49	55.70	14898092.0	Africa
14173	Zimbabwe	2014	3.41	57.00	15245855.0	Africa
14174	Zimbabwe	2015	3.35	59.30	15602751.0	Africa

14175 rows × 6 columns

In [20]:
```
# To define a datafrmae for years between 1960 & 2015 (including 2015)
df2 = df1[df1['year'].between(1960,2016)]
df2
```

Out[20]:

	country	year	fertility_rate	Life expectancy	Total population	continent
25	Afghanistan	1960	7.67	31.94	8994793.0	Asia
26	Afghanistan	1961	7.67	32.47	9164945.0	Asia

	country	year	fertility_rate	Life expectancy	Total population	continent
27	Afghanistan	1962	7.67	33.01	9343772.0	Asia
28	Afghanistan	1963	7.67	33.53	9531555.0	Asia
29	Afghanistan	1964	7.67	34.07	9728645.0	Asia
...	...	...	...	...	...	...
14170	Zimbabwe	2011	3.64	51.60	14255592.0	Africa
14171	Zimbabwe	2012	3.56	54.20	14565482.0	Africa
14172	Zimbabwe	2013	3.49	55.70	14898092.0	Africa
14173	Zimbabwe	2014	3.41	57.00	15245855.0	Africa
14174	Zimbabwe	2015	3.35	59.30	15602751.0	Africa

9800 rows × 6 columns

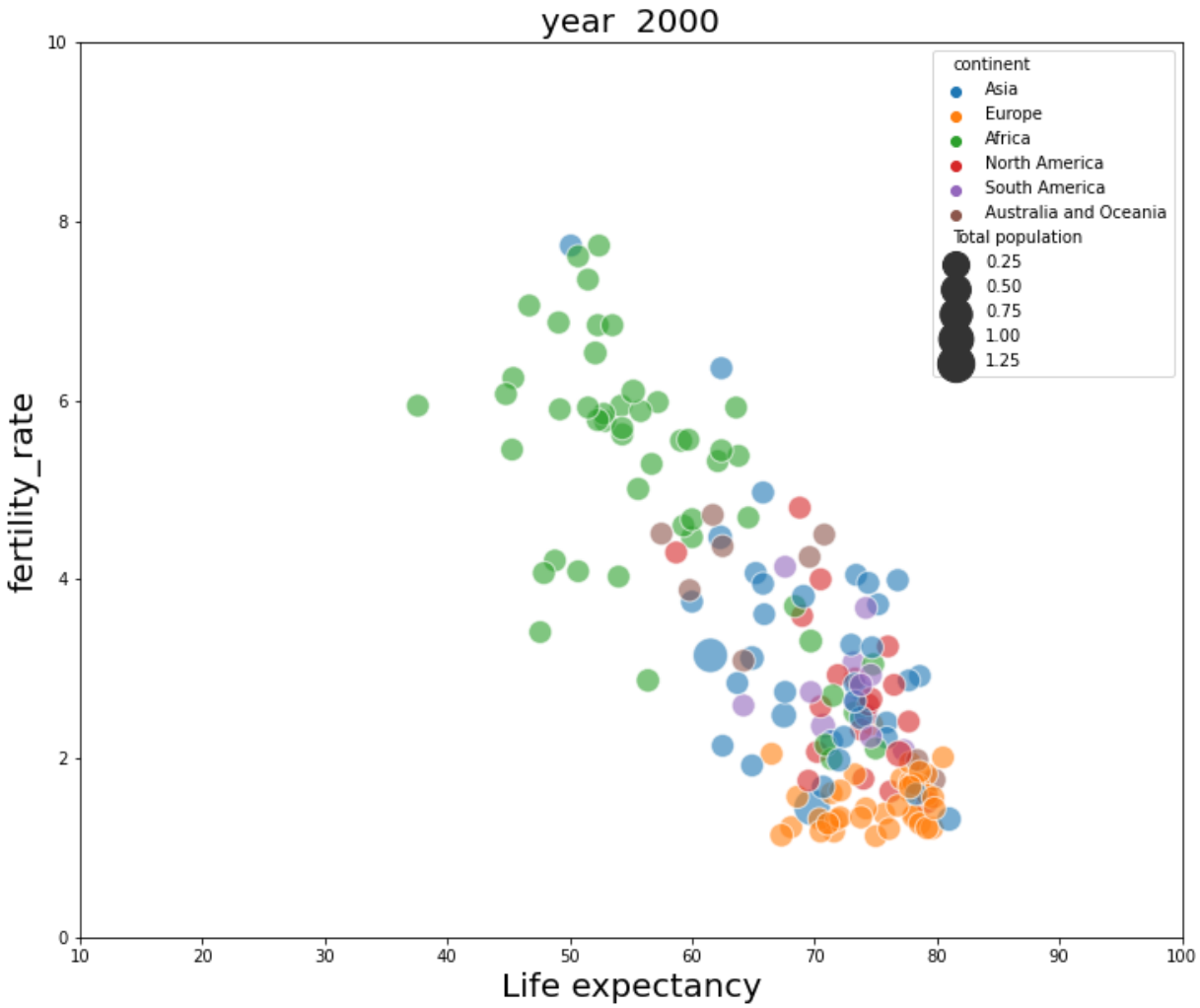
In [12]:

```
import matplotlib.pyplot as plt
import seaborn as sns

"""To plot relationship between ``Life expectancy`` and ``fertility_rate`` us
   of the dataframe using the ``continent``, ``Total population`` parameters
   Here the ``Total population`` is given as size of circle and ``continent``

df = df2.loc[df2['year'] == 2000]
plt.figure(figsize=(12,10))
sns.scatterplot(data=df, x='Life expectancy', y='fertility_rate', hue='contin
                size='Total population',
                sizes=(200,500), palette=None, alpha=0.6, legend=True)
plt.xlabel('Life expectancy', fontsize=20)
plt.ylabel('fertility_rate', fontsize=20)
plt.title('year'+str('  2000'), fontsize=20)

# To fix the coordinates of the plot
plt.axis((10, 100, 0, 10))
plt.savefig('2000.png', dpi=300, facecolor='white')
```



In [14]:
```
# To go through each year and produce a scatterplot related to a year
for i in range(1960,2016,1):
    df = df2.loc[df2['year'] == i]
    plt.figure(figsize=(12,10))
    sns.scatterplot(data=df, x='Life expectancy', y='fertility_rate', size='T
                    , sizes=(200,500), palette=None, alpha=0.6, legend=True)
    plt.xlabel('Life expectancy', fontsize=20)
    plt.ylabel('fertility_rate', fontsize=20)
    plt.title('year'+str(' ')+str(i), fontsize=20)

    plt.axis((10, 100, 0, 10))
    plt.savefig('plot_'+str(i)+'.png', dpi=300, format='png', facecolor='whit
    plt.close()
```

In [15]:
```
pip install imageio
```

```
Requirement already satisfied: imageio in /Users/Disalo/opt/anaconda3/lib/pyth
on3.8/site-packages (2.9.0)
Requirement already satisfied: pillow in /Users/Disalo/opt/anaconda3/lib/pytho
n3.8/site-packages (from imageio) (8.2.0)
Requirement already satisfied: numpy in /Users/Disalo/opt/anaconda3/lib/python
3.8/site-packages (from imageio) (1.20.1)
Note: you may need to restart the kernel to use updated packages.
```

In [17]:
```
# To generate a gif file from the generated scatterplots
import imageio

images = []
```

```
for i in range(1960, 2016,1):
    filename = 'plot_{}.png'.format(i)
    images.append(imageio.imread(filename))

imageio.mimsave('1.8. Long vs. Wide Format_Animated Scatterplot.gif', images,
```