In [1]:
```
import pandas as pd
from copy import deepcopy
import math
import json

from bokeh.io import output_notebook, show
from bokeh.plotting import figure
from bokeh.models import MultiChoice, CustomJS, RangeSlider, HoverTool, ColumnDataSource, Column, Slider
from bokeh.layouts import column
```

In [2]:
```
df = pd.read_csv('data/birth-rate-vs-death-rate.csv', usecols=[0,2,3,4,5])
```

In [3]:
```
df.drop(0, axis=0, inplace=True)
df.rename(columns={'Entity': 'Country','Birth rate - Sex: all - Age: all - Variant: estimates':'Birth rate/1000', 'Death rate - Sex: all - Age: all - Variant: estimates':'Death rate/1000', 'Population - Sex: all - Age: all - Varia
df['Population(millions)'] = df['Population(millions)'] / 1000000
```

In [4]:
```
df['Net population increase/1000'] = df['Birth rate/1000'] - df['Death rate/1000']
# Above is all data wrangling no need to worry
```

In [5]:
```
df
```

Out[5]:

	Country	Year	Birth rate/1000	Death rate/1000	Population(millions)	Net population increase/1000
1	Afghanistan	1950	48.866	37.945	7.480464	10.921
2	Afghanistan	1951	49.147	37.329	7.571542	11.818
3	Afghanistan	1952	49.331	36.618	7.667534	12.713
4	Afghanistan	1953	49.541	35.995	7.764549	13.546
5	Afghanistan	1954	49.616	35.645	7.864289	13.971
...	...	...	...	...	...	...
18404	Zimbabwe	2017	32.516	8.266	14.751101	24.250
18405	Zimbabwe	2018	32.074	7.972	15.052191	24.102
18406	Zimbabwe	2019	31.518	8.043	15.354606	23.475
18407	Zimbabwe	2020	31.009	8.132	15.669663	22.877
18408	Zimbabwe	2021	30.537	9.057	15.993525	21.480

18408 rows × 6 columns

In [6]:
```
d = df[(df['Country']=='France') & (df['Year'] >= 2000) & (df['Year'] >= 2000)]
```

In [7]:
```
p = figure(width=700, height=400, title="Line Plot")
p.line(d['Death rate/1000'], d['Birth rate/1000'], line_width=2)

# This is just an example
p.line(df[df['Country']=='United Kingdom']['Death rate/1000'], df[df['Country']=='United Kingdom']['Birth rate/1000'], line_width=2, line_color='red')

p.xaxis.axis_label = 'Death rate/1000'
p.yaxis.axis_label = 'Birth rate/1000'

output_notebook()
show(p)
```

Loading BokehJS ...

In [8]:
```
df1 = df.drop(df.index)
df1
```

Out[8]:

	Country	Year	Birth rate/1000	Death rate/1000	Population(millions)	Net population increase/1000

In [ ]:
```

```

In [9]:
```
# Below is code for the actual plot
data = df
data1 = df1 # new
data_source = ColumnDataSource(data)
source = ColumnDataSource(dict(name=[], deathrate=[], birthrate=[], net=[]))
data_source_1 = ColumnDataSource(data1) # new

# Defining the hover tool and figure
hover = HoverTool(tooltips = [('Country/region','@name'),('Death rate(in thousands)', '@deathrate'), ('Birth rate(in thousands)', '@birthrate'), ('Net population increase(in thousands)', '@net')])
p = figure(title='Birth rate vs Death rate of countries (in thousands)', width=700, height=400, tools=[hover, 'wheel_zoom', 'pan'])
p.xaxis.axis_label = 'Death rate/1000'
p.yaxis.axis_label = 'Birth rate/1000'
data1
```

Out[9]:

	Country	Year	Birth rate/1000	Death rate/1000	Population(millions)	Net population increase/1000

In [10]:
```
data
```

Out[10]:

	Country	Year	Birth rate/1000	Death rate/1000	Population(millions)	Net population increase/1000
1	Afghanistan	1950	48.866	37.945	7.480464	10.921
2	Afghanistan	1951	49.147	37.329	7.571542	11.818
3	Afghanistan	1952	49.331	36.618	7.667534	12.713
4	Afghanistan	1953	49.541	35.995	7.764549	13.546
5	Afghanistan	1954	49.616	35.645	7.864289	13.971
...	...	...	...	...	...	...
18404	Zimbabwe	2017	32.516	8.266	14.751101	24.250
18405	Zimbabwe	2018	32.074	7.972	15.052191	24.102
18406	Zimbabwe	2019	31.518	8.043	15.354606	23.475
18407	Zimbabwe	2020	31.009	8.132	15.669663	22.877
18408	Zimbabwe	2021	30.537	9.057	15.993525	21.480

18408 rows × 6 columns

In [11]:
```
# Attempted call back at range slider
callbackrs = CustomJS(args={'data_source': data_source, 'data_source_1': data_source_1}, code="""
    var year_range = cb_obj.value
    var data = data_source.data
    var data1 = data_source_1.data
    var update_data = {"Country":[], "Year":[], "Birth rate/1000":[], "Death rate/1000":[], "Population(millions)":[], "Net population increase/1000":[]}

    var target_column = data['Year']
    const year_length = data['Year'].length
    for (let i = 0; i < year_length; i++) {
        if (target_column[i] <= year_range[1] and target_column[i] >= year_range[1]) {
            for (const [key, value] of Object.entries(update_data)) {
                value.push(data[key][i])
                }
            }
        }
    data_source1.data = update_data
    data_source1.change.emit();
""")
# Define parameters for range slider
range_slider = RangeSlider(title='Year', start=1950, end=2021, step=1, value=(1950, 2021))
```

In [12]:
```
p.line('deathrate', 'birthrate', line_width=2, source=source)
# Line indicating where birth rate is equal to the death rate
p.line(x := [i for i  in range(0, 50)], x, alpha=0.2)

# JScallback for multi choice widget
# Passed in data_source_1 as source of plotting data instead of data_source
callbackmc = CustomJS(args={'source': source, 'data_source_1': data_source_1}, code="""
    var data = data_source_1.data;
    var s_data = source.data;
    var selected = cb_obj.value;

    var Country = data['Country'];
    var death_rate_data = data['Death rate/1000'];
    var birth_rate_data = data['Birth rate/1000'];
    var net_data = data['Net population increase/1000'];

    var name = s_data['name']
    name.length = 0;
    var deathrate = s_data['deathrate'];
    deathrate.length = 0;
    var birthrate = s_data['birthrate'];
    birthrate.length = 0;
    var net = s_data['net'];
    net.length = 0;

    for (var i = 0; i < death_rate_data.length; i++) {
        if (selected.indexOf(Country[i]) >= 0) {
            name.push(Country[i]);
            deathrate.push(death_rate_data[i]);
            birthrate.push(birth_rate_data[i]);
            net.push(net_data[i]);
        }
    }
    source.change.emit();
""")


multi_choice = MultiChoice(title='Select Countries', value=[], options=data['Country'].unique().tolist())
```

In [13]:
```
multi_choice.js_on_change('value', callbackmc)
range_slider.js_on_change('value', callbackrs)

layout = Column(multi_choice, range_slider, p)

output_notebook()

show(layout)
```

BokehJS 3.1.1 successfully loaded.

In [ ]:

In [ ]: