Introduction

Many chart types can be built with `matplotlib`: histograms, bars, polar, ... One can look at the **examples presented in the documentation and gallery.**.

Yet, a lot of code lines are needed to build only a simple graph. In some cases, the prefered way is this one:

1. Use higher-level packages **built on top of `matplotlib`**
2. Use `matplotlib` to customize the plot if needed

Case study definition

A dataset about a pengouins population is downloaded on seaborn_data and stored on disk as 'data.csv'.

In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('data.csv')
df
```

Out[1]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	F
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	F
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	F
...	...	...	...	...	...	...	
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	F
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	F
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	

344 rows × 7 columns

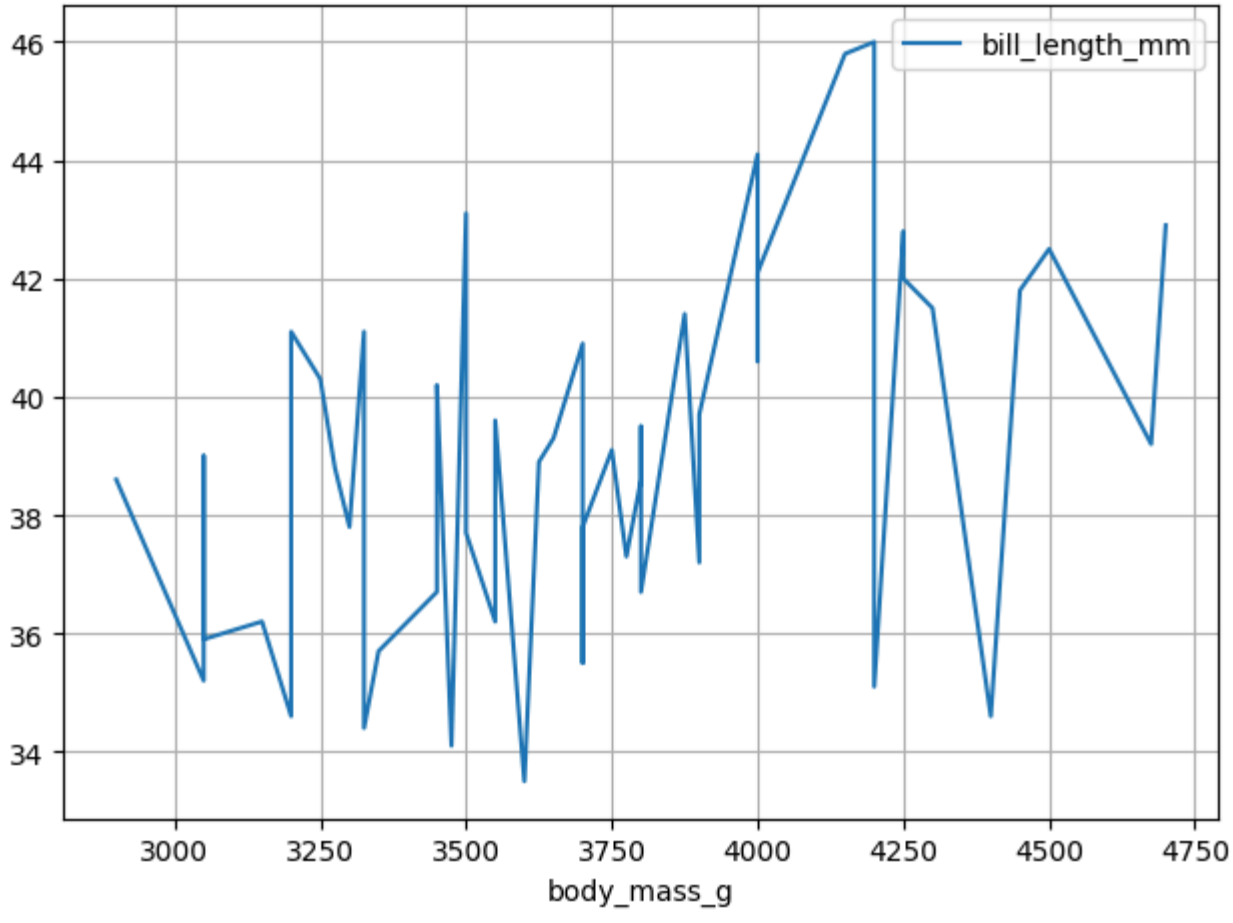
pandas

`pandas` can perform quick plot of data stored in either a `DataFrame` or a `Series`.

Example 1: *line plot*

Let's plot the length of the bill of pengouns from Torgersen island, as a function of their mass:

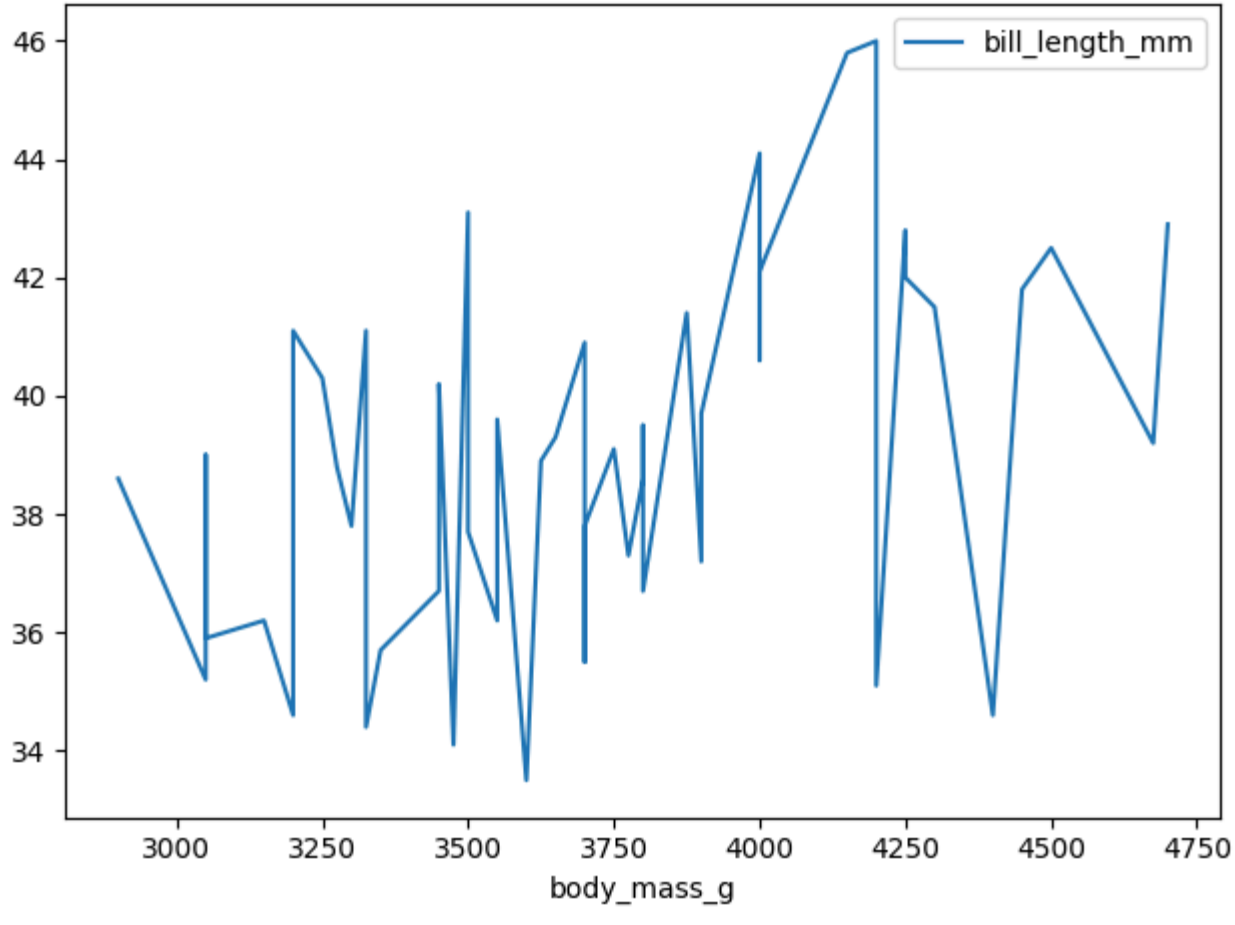
In [2]:
```
df1 = df[df['island']=='Torgersen']
df1 = df1.sort_values('body_mass_g')
ax = df1.plot(x='body_mass_g', y='bill_length_mm', kind='line')
```


bill_length_mm
46
44
42
40
38
36
34
3000
3250
3500
3750
4000
4250
4500
4750
body_mass_g

The call to the `plot` method of `pandas` returned a `matplotlib` `axes` object: **let's modify it**.

In [3]:
```
ax.grid()
ax.get_figure()   # needed to display once again
                  # the figure in Jupyter Notebook.
```

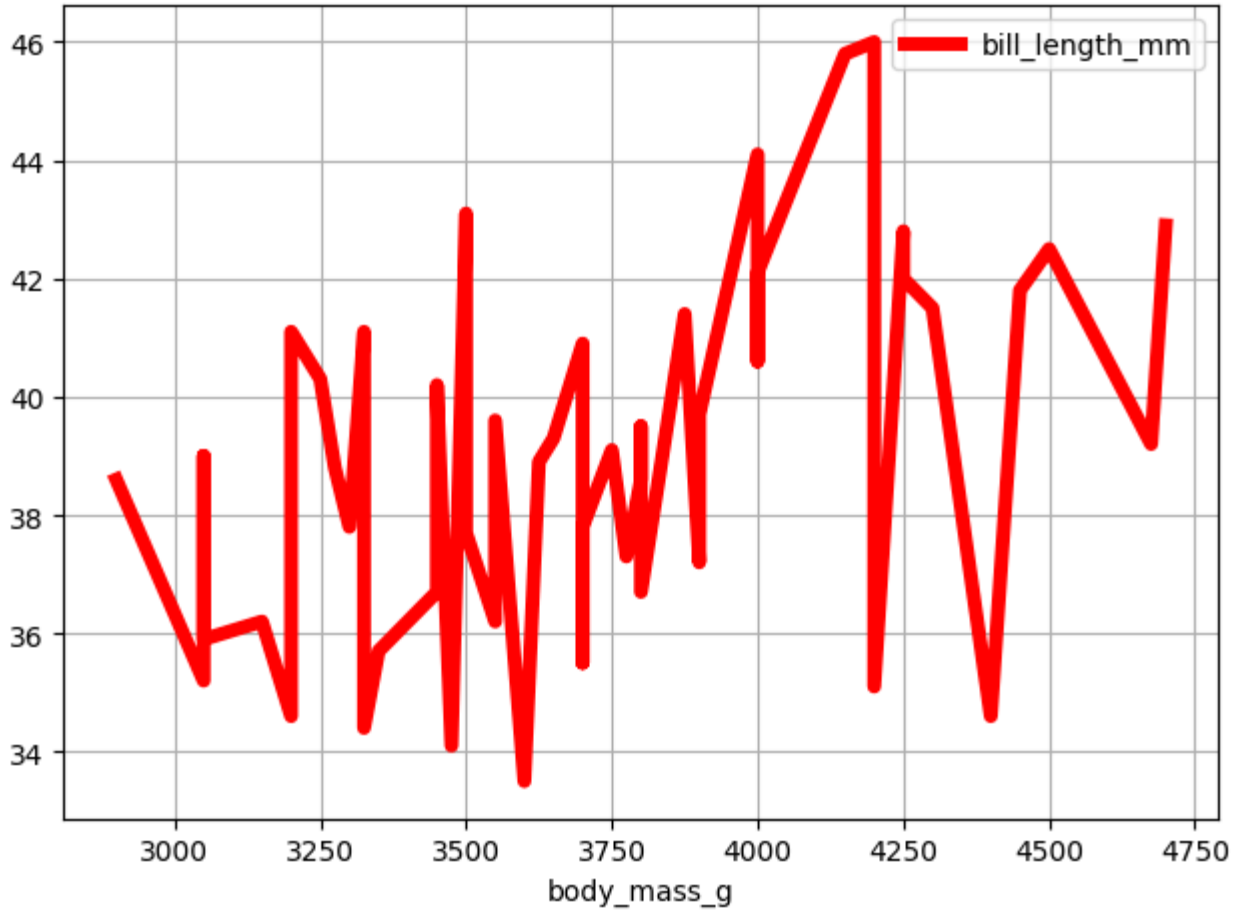
Out[3]:



Another way is to modify the plot when created. For this purpose, some *keywords arguments* can be passed to the `plot` method of the dataframe. These are the same than the plot function of `matplotlib`.

In [4]:
```
df1.plot(x='body_mass_g', y='bill_length_mm', kind='line',
        color='red', linewidth=5)    # these arguments are passed to matplotlib
```

Out[4]:
```
<Axes: xlabel='body_mass_g'>
```


bill_length_mm
46
44
42
40
38
36
34
3000
3250
3500
3750
4000
4250
4500
4750
body_mass_g

Example 2: *barplot*

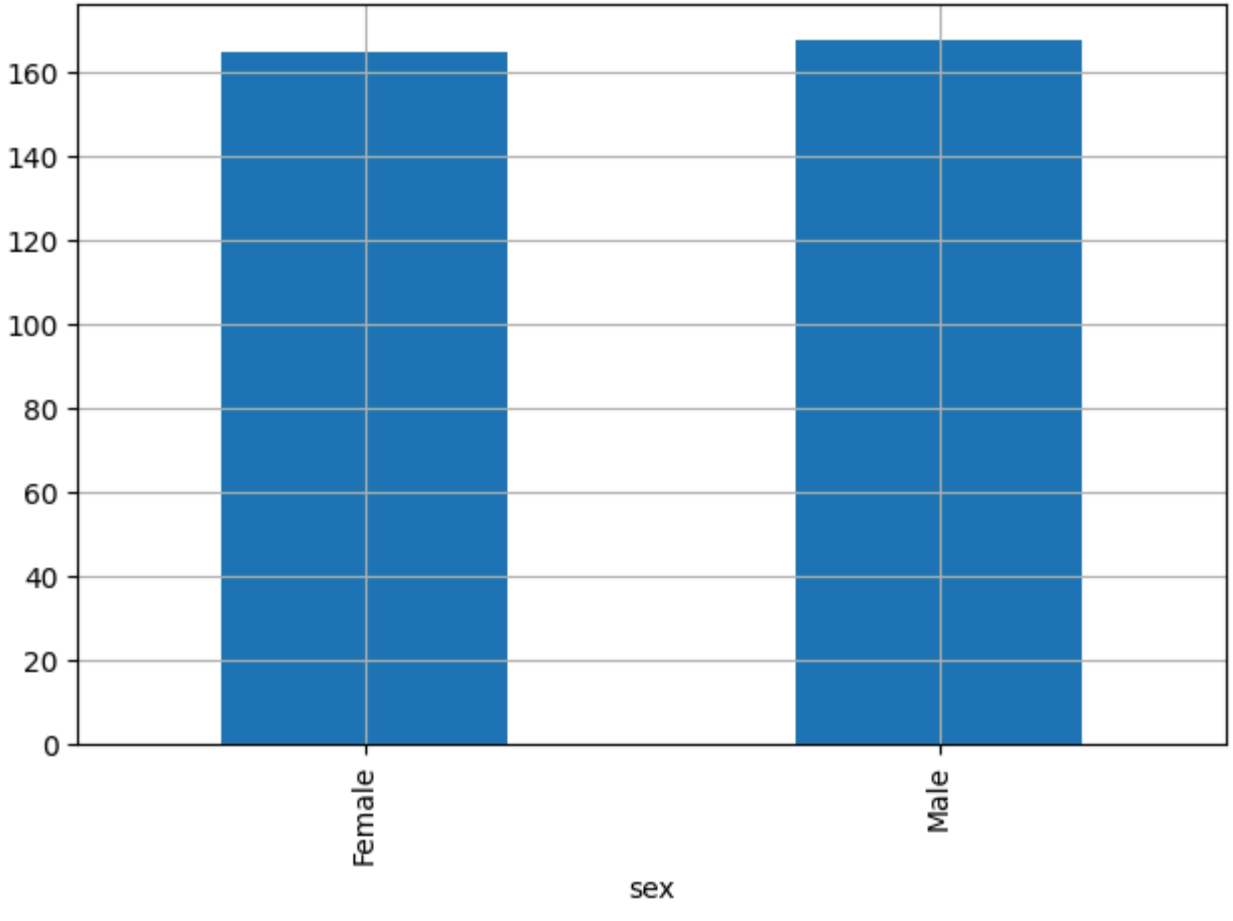
Let's plot the numbers of females and males on Torgersen island:

In [5]:
```
df2 = df.groupby('sex')['bill_length_mm'].count()  # a random column is needed
                                                   # for rows to be counted
df2
```

Out[5]:
```
sex
Female    165
Male      168
Name: bill_length_mm, dtype: int64
```

In [6]:
```
df2.plot(kind='bar')
```

Out[6]:
```
<Axes: xlabel='sex'>
```



Conclusion

The `plot` method is an easy way to quickly inspect the content of a DataFrame. Yet, it is unadapted to advanced statistical plots

In that case, the preferred tool is `seaborn`, which produces clear an pretty charts.

`seaborn`

`seaborn` is a plotting library that is built on top of `matplotlib`.

The strength of `seaborn` is that it can directly be used with `pandas` DataFrames and Series.

Learning `seaborn` mainly consists in understanding the arguments it takes:

- `x` : abscissa data
- `y` : ordinate data
- `hue` : data differenciated according to a **color** code
- `style` : data differenciated according to the line/marker **style**
- `size` : data differenciated according to the line/marker **size**

`seaborn` comes with 2 kinds of funtions:

1. Functions that create only one ax to plot the data

2. Functions that create several axes using arguments `row` and `col` (related to a `FacetGrid` ojects):

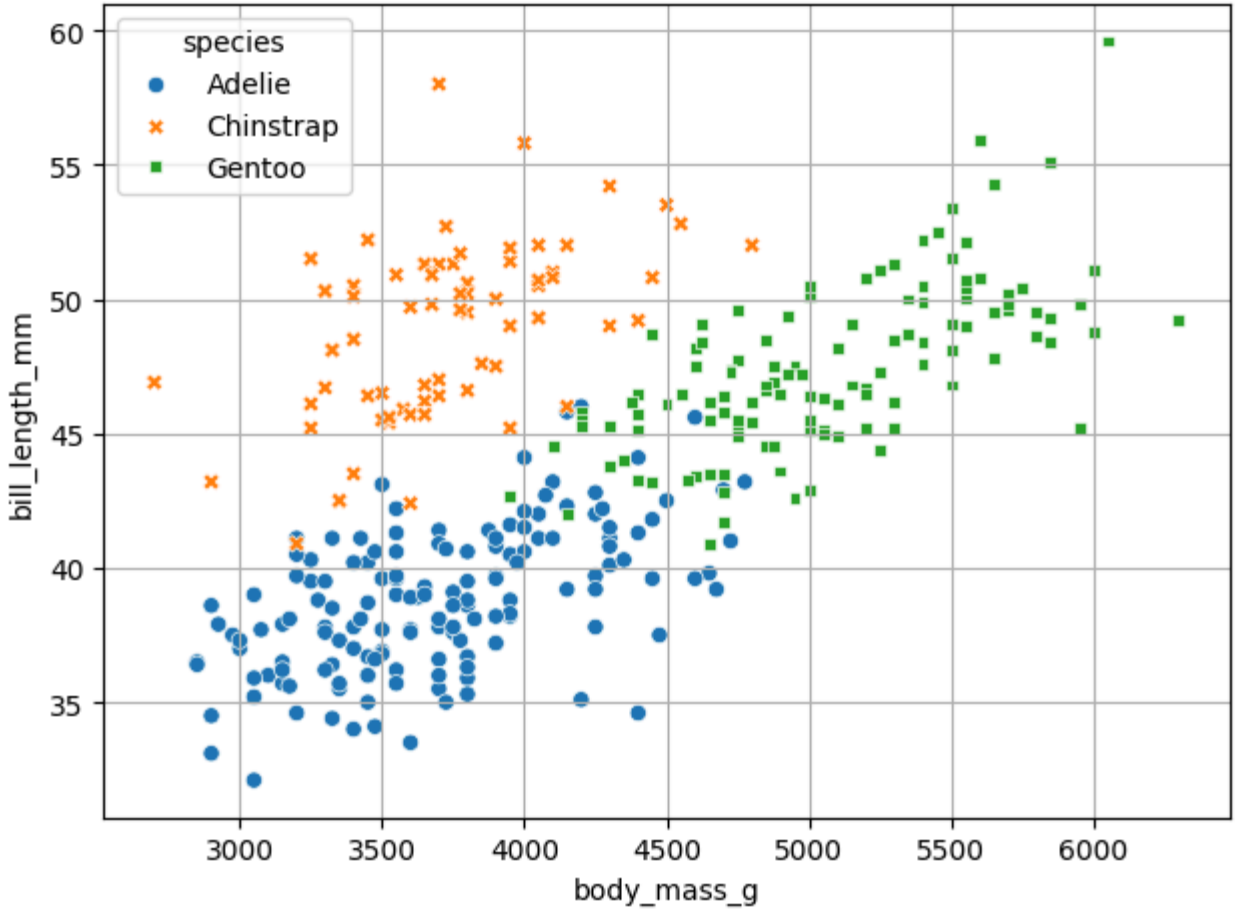
    - `row` : data differenciated according to the **row** of the ax object in the figure
    - `col` : data differenciated according to the **col** of the ax object in the figure

One `axes`

Let's go back to our pengouins. The bill length is plotted as a function of body mass, with a color code and style differenciation for species. The returned object is an ax object: it can be customized if needed.

In [7]:
```
import seaborn as sns
sns.scatterplot(df, x='body_mass_g', y='bill_length_mm', hue='species', style='specie
```

Out[7]:
```
<Axes: xlabel='body_mass_g', ylabel='bill_length_mm'>
```



Several `axes`

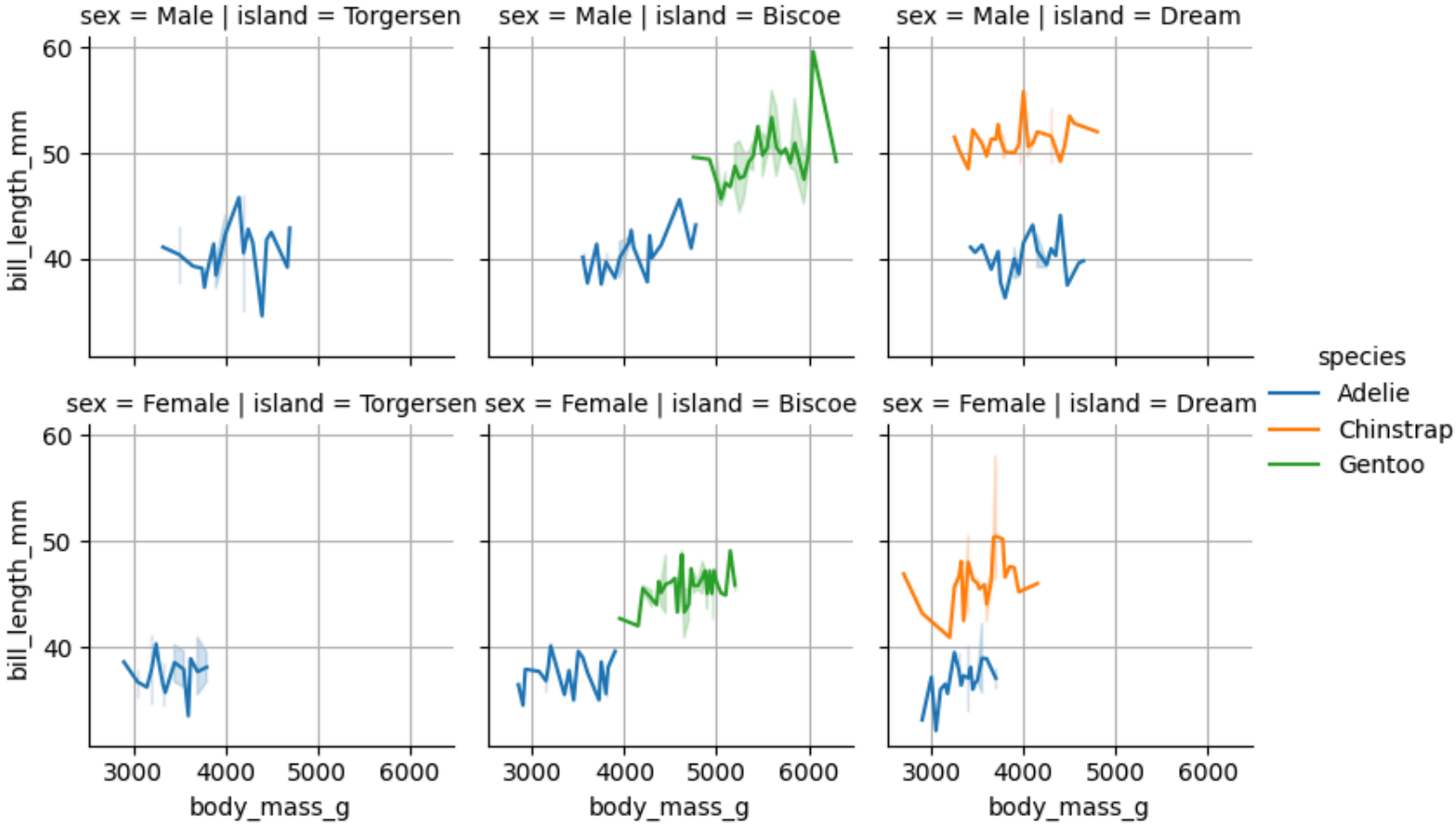
Complete example

Here after, the bill length is plotted:

- as a function of body mass
- with a color code for species
- with sex differenciation along rows
- with island differenciation along columns

In [8]:

```
import seaborn as sns
import matplotlib
fg = sns.relplot(df, kind='line', x='body_mass_g', y='bill_length_mm', hue='species',
                 height=2.5, aspect=1)
```



Notes:

- `relplot` needs `kind='line'` to behave as `lineplot`
- `height` is the height of each *subplot*, i.e. each ax.

  `aspect` is the width/height ratio.
- Light color zones represent uncertainties. Indeed, given a tuple of (species, sex, island, mass), there are several individuals.

Let's investigate the uncertainties zones:

In [9]:
```
df_ = df.groupby(['body_mass_g', 'species', 'sex', 'island']).count()
df_[(df_['bill_depth_mm']!=1)|(df_['flipper_length_mm']!=1)].head(3)
```

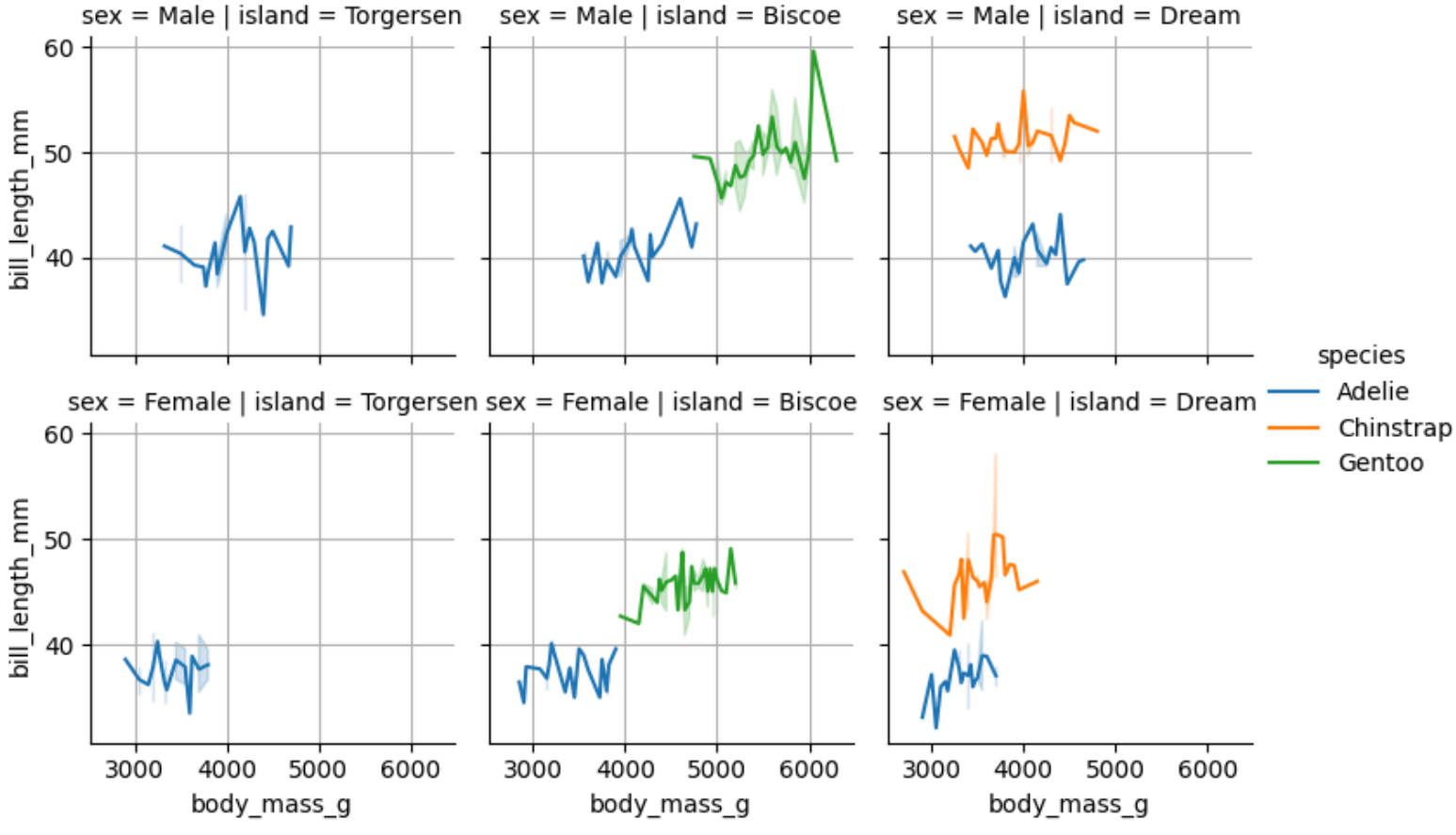
Out[9]:

				bill_length_mm	bill_depth_mm	flipper_length_mm
body_mass_g	**species**	**sex**	**island**			
2850.0	**Adelie**	**Female**	**Biscoe**	2	2	
3000.0	**Adelie**	**Female**	**Dream**	2	2	
3050.0	**Adelie**	**Female**	**Torgersen**	3	3	

Finally, note that `relplot` returned a `FacetGrid` instance. It has an `axes` attribute (numpy array) that can be used to customize plots.

In [10]:
```
axes = fg.axes
axes[1, 2].grid()
axes[1, 2].get_figure()  # needed for Jupyter
```

Out[10]:

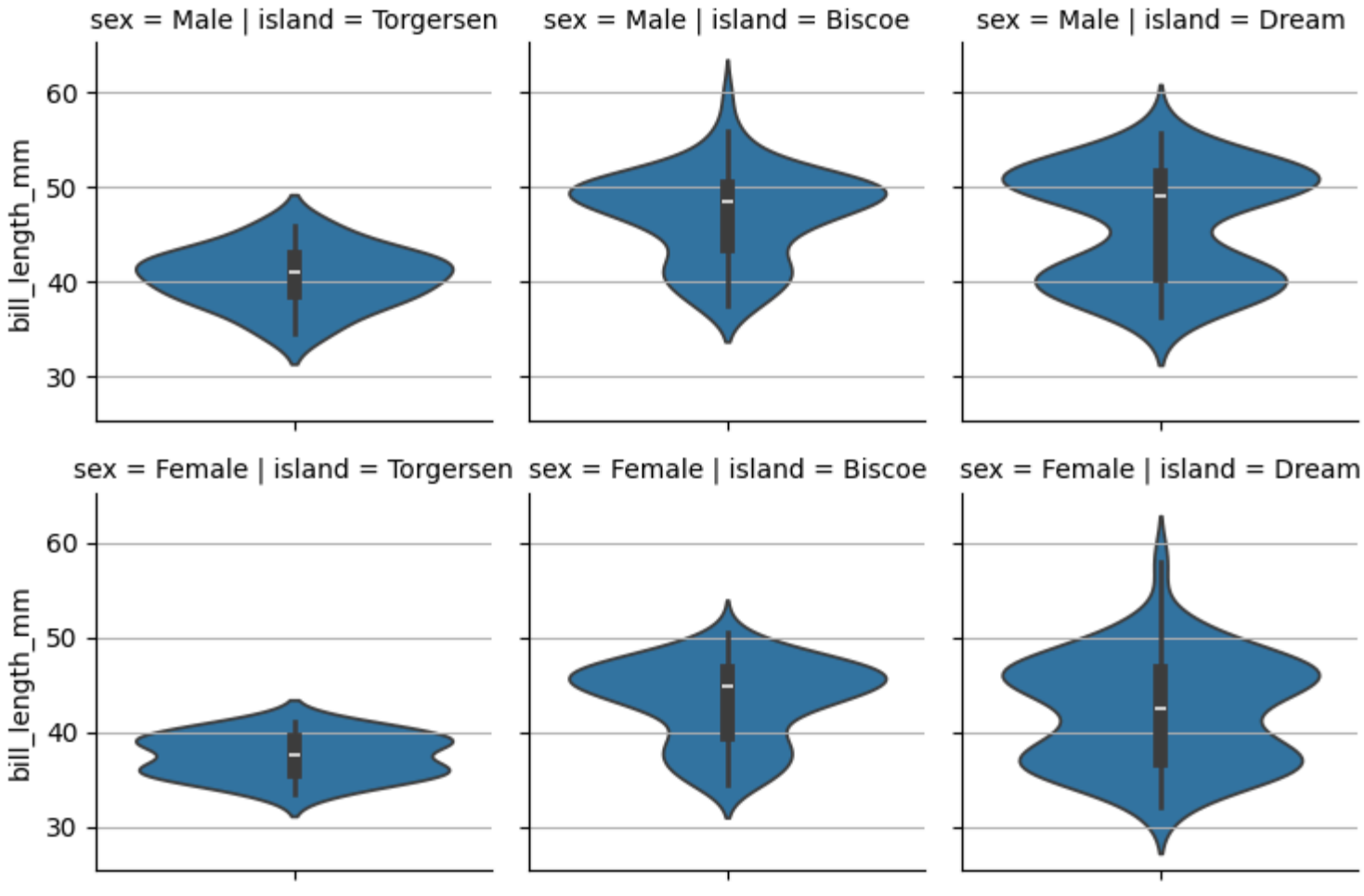


Other chart types

Many charts types can be created using `seaborn` (see the gallery).

For instance, let's create a `violinplot` to get a statistcial approach of data:

In [11]:
```
fg = sns.catplot(df, kind='violin', y='bill_length_mm',  row='sex', col='island',
                 height=2.5)
```


sex = Male | island = Torgersen
sex = Male | island = Biscoe
sex = Male | island = Dream
sex = Female | island = Torgersen
sex = Female | island = Biscoe
sex = Female | island = Dream
bill_length_mm
60
50
40
30