



Australian
National
University

COMP1730/COMP6730

Programming for Scientists

Data Visualisation



Overview

- * Basic data visualisation
- * Formatting charts
- * Chart types
- * Animation and videos



Data visualisation

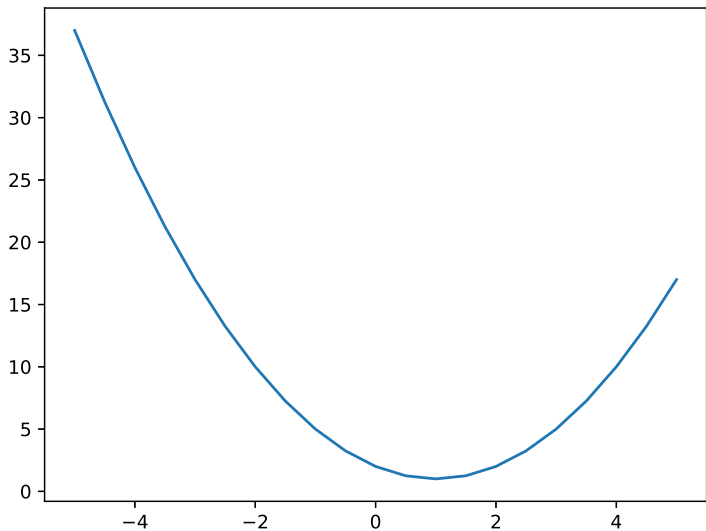
- * Data visualisation is about communication - not pretty pictures!

Matplotlib

- * There are many different visualisation libraries in Python. We will be making use of `matplotlib`.

```
import numpy as np
import matplotlib.pyplot as plt
```

```
x = np.linspace(-5, 5, 21)
y = x ** 2 - 2 * x + 2
plt.plot(x, y)
plt.show()
```

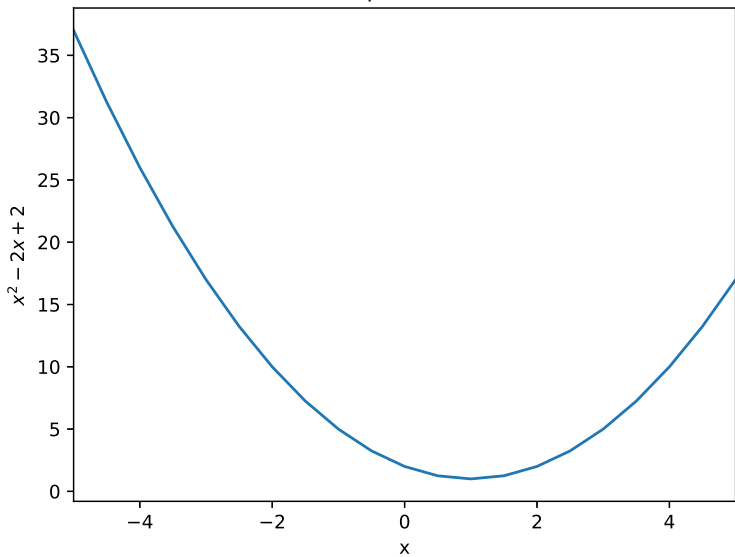


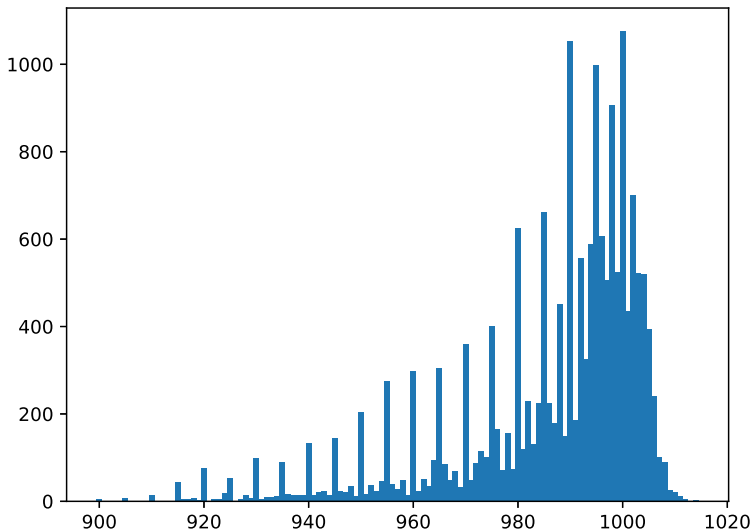
Formatting

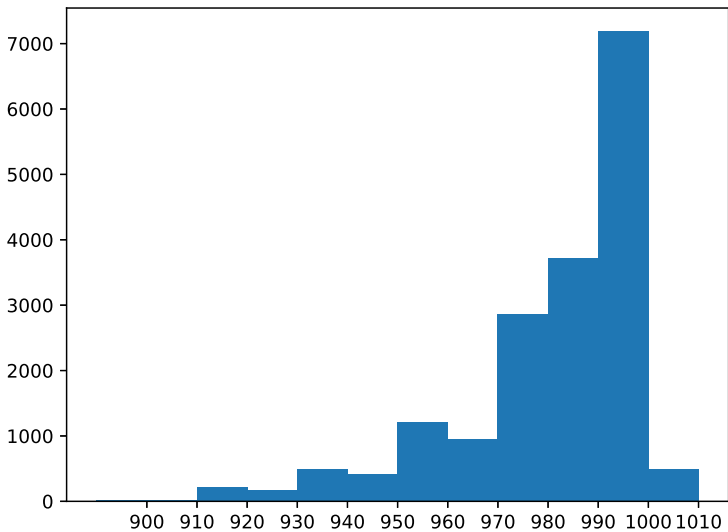
- * There are many different options for formatting charts and tables. These can all be done manually, but it is far better to do this programmatically.
- * Visualisation is about *communication*, so make sure your visualisations convey the information you want them to convey.
- * Include labels, titles, legends (if appropriate), rescale axis, etc.
- * Matplotlib can make use of Latex fonts in labels and titles if required.
- * Use `plt.savefig('Figure_name.png')`



Example Parabola



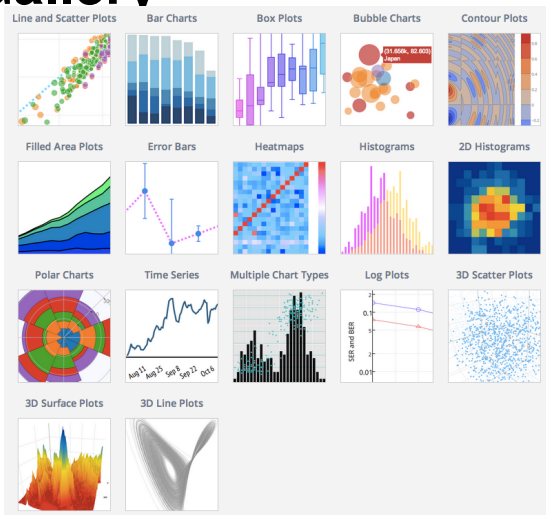




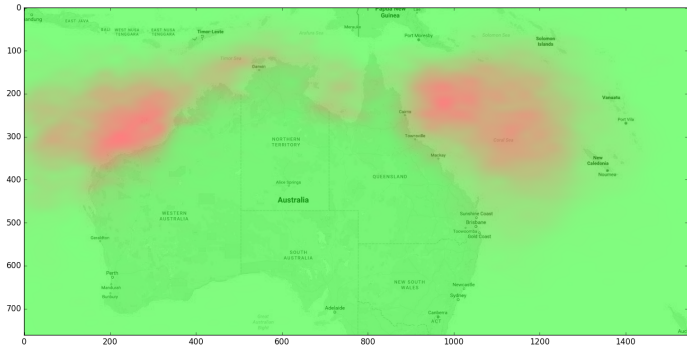
Plot Types

- * Line plot: relationship between two variables, e.g. a time series.
- * Scatter plot: paired numerical data.
- * Bar chart: compare quantities across a discrete set of categories.
- * Histogram: display a probability distribution, broken into discrete categories.
- * Many others: stacked bar chart, spider graph, box plot, heat map, pie chart, polar chart, contour plot, etc.

Gallery



Source: <https://plot.ly/javascript/basic-charts/>



Visualisation Tips

- * Use a chart that is appropriate for your data.
- * Format your chart appropriately, labels, title, axis, scale, etc., from within the code.
- * Make sure your colour scheme works well for printed reports (including black and white).
- * Be consistent with your colours and styles across figures in the same report.

Animation

- * You can produce animations in `matplotlib` by using `matplotlib.animation.FuncAnimation`.
- * Think of animation as drawing several individual graphics, one after another in (relatively) quick succession.
- * You pass in a function, and a sequence data structure that contains the arguments (example, a tuple of tuples).
- * `matplotlib` calls the function for each frame of the animation, using the next element in the sequence to provide the data.

User interfaces and videos

- * You can also use `matplotlib` to create interactible graphical user interfaces, with buttons and other controls.
- * If you have proper codecs installed, you can turn your animation into videos.
- * It may be easier to save your frames as `.png` images or similar and turn them into videos using different software.
- * There are good tutorials available if you are interested in exploring this further (we don't go over it in this course).