Documentation

Code documentation

Document preparation

- Word/excel briefly
- Latex
- Jupyter notebook
- markdown

2D-3D graphs with python

Code documentation

Documentation generator

- a tool that generates software documentation

Examples from Doxygen

- $ sudo apt-get install doxygen
- $ sudo apt install graphviz
  - Graph visualization library
  - Graphviz
- $ doxygen –version

https://en.wikipedia.org/wiki/Comparison_of_documentation_generators

Example: Doxygen

All commands in the documentation start with a backslash (`\`) or an at-sign (`@`)

```
/**
<A short one line description>

<Longer description>
<May span multiple lines or paragraphs
as needed>

@param  someParameter  Description
of method's or function's input or output
parameter
@param  ...
@return Description of the return value
*/
```

```
/**
 * <A short one line description>
 *
 * <Longer description>
 * <May span multiple lines or paragraphs
as needed>
 *
 * @param  someParameter  Description
 * @param  ...
 * @return Description of the return value
 */
```

```
/**
 * @file
 * @brief  The header file of the Time class.
 * @author  John Doe <jdoe@example.com>
 * @version  1.0
 * @copyright  CC BY-SA or GFDL.
 * @sa  <a href="https://en.wikipedia.org/wiki/Wikipedia:Copyrights">Wikipedia:Copyrights - Wikipedia</a>
 */

/**
 * The time class represents a moment of time.
 *
 * @author John Doe
 */
class Time {
public:

    /**
     * Constructor that sets the time to a given value.
     *
     * @param timeMillis A number of milliseconds passed since Jan 1, 1970.
     */
    explicit Time(long long timeMillis) {
        // the code
    }

    /**
     * Get the current time.
     *
     * @return A time object set to the current time.
     */
    static Time now() {
        // the code
    }

private:
    long long m_timeMillis; ///< Milliseconds passed since Jan 1, 1970.
};
```

```
# you can also use Sphynix for Python
"""@package docstring
Documentation for this module.

More details.
"""


def func():
    """Documentation for a function.

    More details.
    """
```

Example output of doxygen

Example of output generated by doxygen

Example: Sphinx for Python

Documentation » Installing Sphinx

- sudo apt-get install python3-sphinx
  - Or sudo pip install sphinx
- mkdir docs
- cd docs
- sphinx-quickstart
- sphynix

Example Basic Sphinx Example Project

- sphinx-autobuild source-dir output-dir
- make html

Documentation » Projects using Sphinx

Document preparation/editing with Word-excel

Alternatives

- MS office
- Libre office
- Google docs, sheets

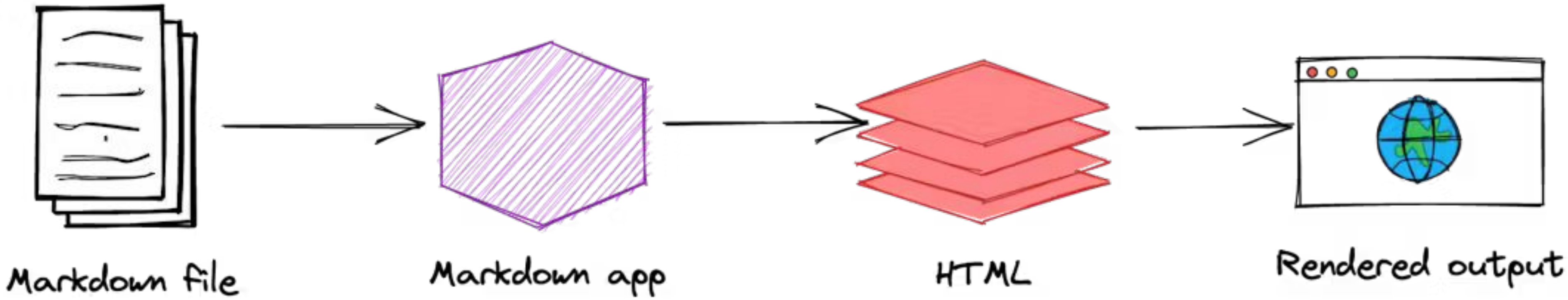
Markdown

Markdown is a lightweight markup language that you can use to add formatting elements to plaintext text documents

Different than ("What You See Is What You Get")WYSIWYG - Wikipedia

Markdown can be used for everything.

- to create websites, documents, notes, books, presentations, email messages, and technical documentation.



https://www.markdownguide.org/getting-started/

Markdown: basic syntax

Basic Syntax | Markdown Guide

Markdown	HTML	Rendered output
# Heading level 1	<h1>Heading level 1</h1>	Heading level 1
## Heading level 2	<h2>Heading level 2</h2>	Heading level 2
Heading level 1 ===============	<h1>Heading level 1</h1>	Heading level 1
Heading level 2 ---------------	<h2>Heading level 2</h2>	Heading level 2
I just love **bold text**.	I just love <strong>bold text</strong>.	I just love **bold text**.
I just love __bold text__.	I just love <strong>bold text</strong>.	I just love **bold text**.
Italicized text is the *cat's meow*.	Italicized text is the <em>cat's meow</em>.	Italicized text is the *cat’s meow*.

Markdown: basic syntax

Table

```
| Syntax | Description |
| ----------- | ----------- |
| Header | Title |
| Paragraph | Text |
```

Fenced Code Block

~~~~
```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```
~~~~

Emoji
(see also Copying and Pasting Emoji)

```
That is so funny! :joy:
```

```
For more see
```

Basic Syntax | Markdown Guide

Markdown Cheat Sheet

Latex: Mathematical document editing

The latex program processes your text and commands to produce a beautifully formatted document.

Latex: The rain in Spain falls **\emph{mainly}** on the plain.

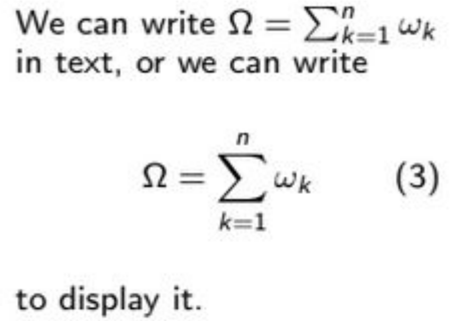
Pdf: The rain in Spain falls *mainly* on the plain.

https://www.overleaf.com/learn/latex/Free_online_introduction_to_LaTeX_(part_1)

Latex	pdf
\begin{itemize} \item Tea \item Milk \item Biscuits \end{itemize}	● Tea ● Milk ● Biscuits

\begin{figure}
\includegraphics{gerbil}
\end{figure}	

\begin{equation}
\alpha + \beta + 1
\end{equation}
	$\alpha + \beta + 1$ (1)

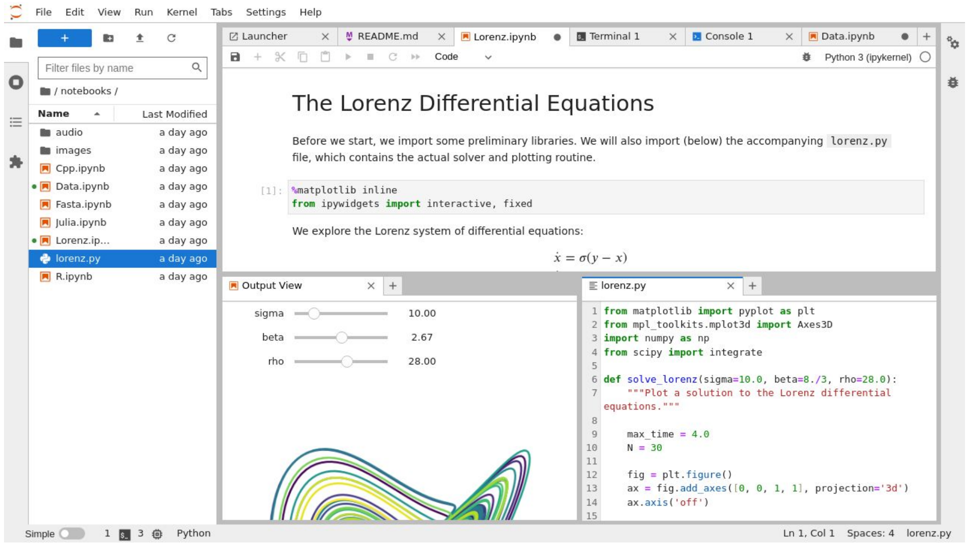
We can write $ \Omega = \sum_{k=1}^{n} \omega_k $ in text, or we can write \begin{equation} \Omega = \sum_{k=1}^{n} \omega_k \end{equation} to display it.	 We can write $\Omega = \sum_{k=1}^{n} \omega_k$ in text, or we can write $\Omega = \sum\limits_{k=1}^{n} \omega_k$ (3) to display it.

https://www.overleaf.com/learn/latex/Free_online_introduction_to_LaTeX_(part_1)

Jupyter notebook

A computational notebook is a shareable document that combines computer code, plain language descriptions, data, rich visualizations like 3D models, charts, graphs and figures, and interactive controls.

Jupyter notebooks are documents that combine live runnable code with narrative text (Markdown), equations (LaTeX), images, interactive visualizations and other rich output:



Name	Last Modified
audio	a day ago
images	a day ago
Cpp.ipynb	a day ago
Data.ipynb	a day ago
Fasta.ipynb	a day ago
Julia.ipynb	a day ago
Lorenz.ip...	a day ago
lorenz.py	a day ago
R.ipynb	a day ago

The Lorenz Differential Equations

Before we start, we import some preliminary libraries. We will also import (below) the accompanying `lorenz.py` file, which contains the actual solver and plotting routine.

[1]:
```
%matplotlib inline
from ipywidgets import interactive, fixed
```

We explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$

sigma 10.00
beta 2.67
rho 28.00

```
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
from scipy import integrate

def solve_lorenz(sigma=10.0, beta=8./3, rho=28.0):
    """Plot a solution to the Lorenz differential
equations."""

    max_time = 4.0
    N = 30

    fig = plt.figure()
    ax = fig.add_axes([0, 0, 1, 1], projection='3d')
    ax.axis('off')

```

Jupyter notebook editor programs

REPL(Read-Eval-Print-Loop)

The programmer writes snippets of code that are

- first read (R),
- then Evaluated (E)
  - or in other words executed,
- the results are printed (P)
  - to some kind of display or output,
- and that process happens repeatedly in a loop (L),
  - where the REPL waits until the programmer has another snippet of code to execute.

We use interpreted language such as Python

Kernels

A REPL in a language of choice is created for each open notebook files.

Python-jupyter notebook installation

You can download from python.org

- Python: Download Python
- Install pip Installation - pip documentation v24.3.1
- Jupyter: Project Jupyter | Installing Jupyter

Or you can install anaconda

- https://www.anaconda.com/download/success
  - This includes many scientific packages
  - $ `conda install anaconda::jupyter`

Online jupyter notebook

Project Jupyter | Try Jupyter

Other notebooks (cloud services)

Google Colab

- You can also run linux terminal
  - Try **!sh**

Microsoft & github notebooks Notebooks at Microsoft - Visual Studio

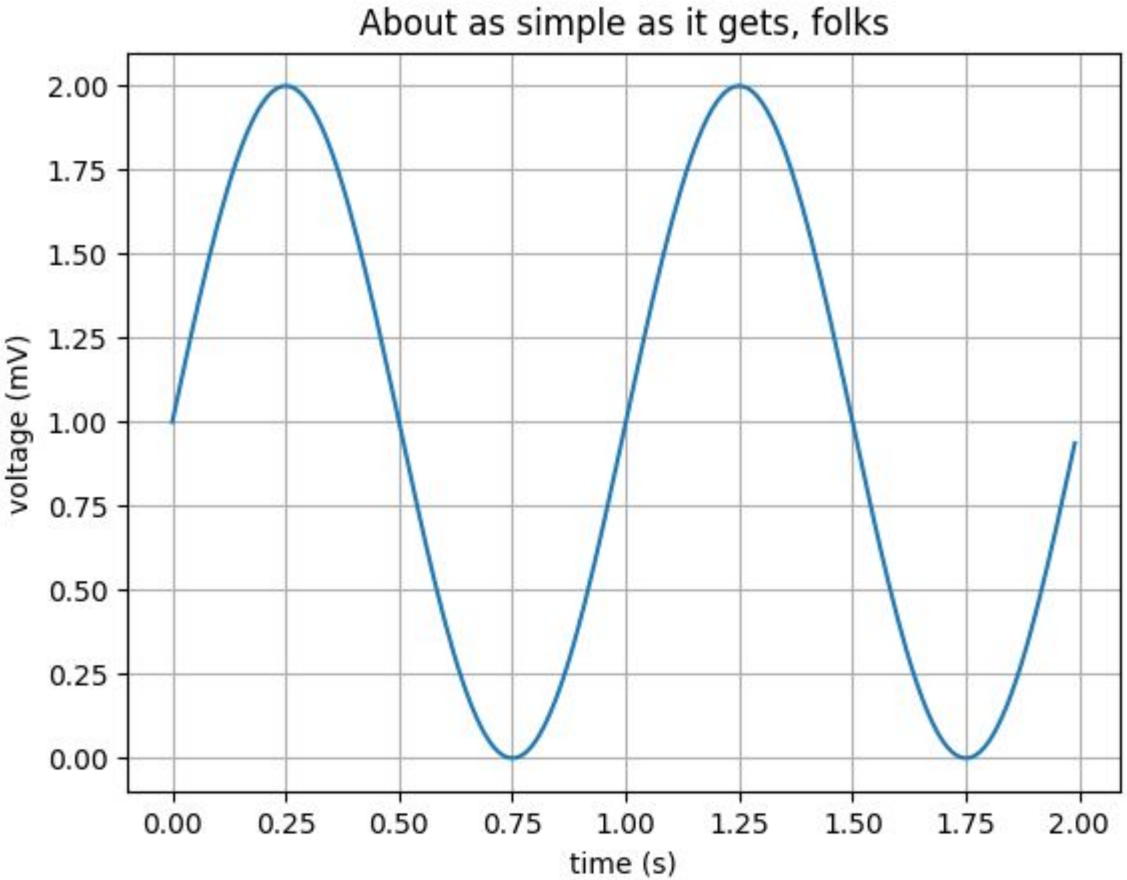
Kaggle Notebooks

Amazon SageMaker Studio Lab

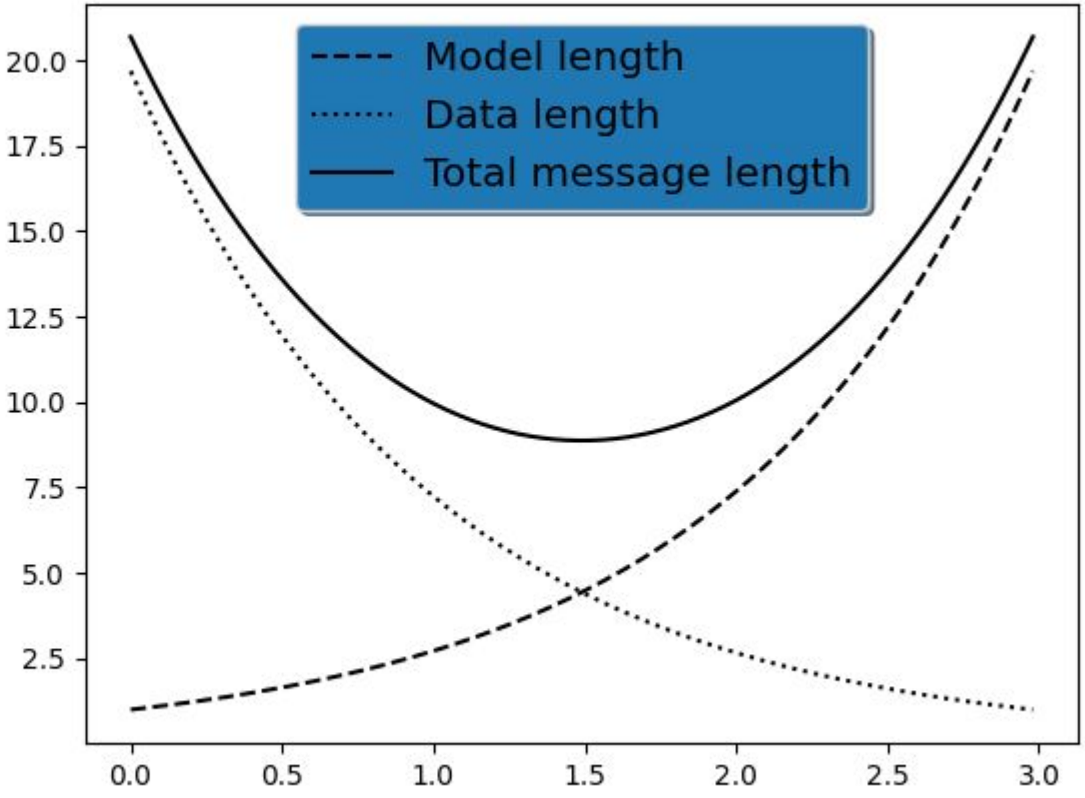
Visualization of data

2D line plots

plot()



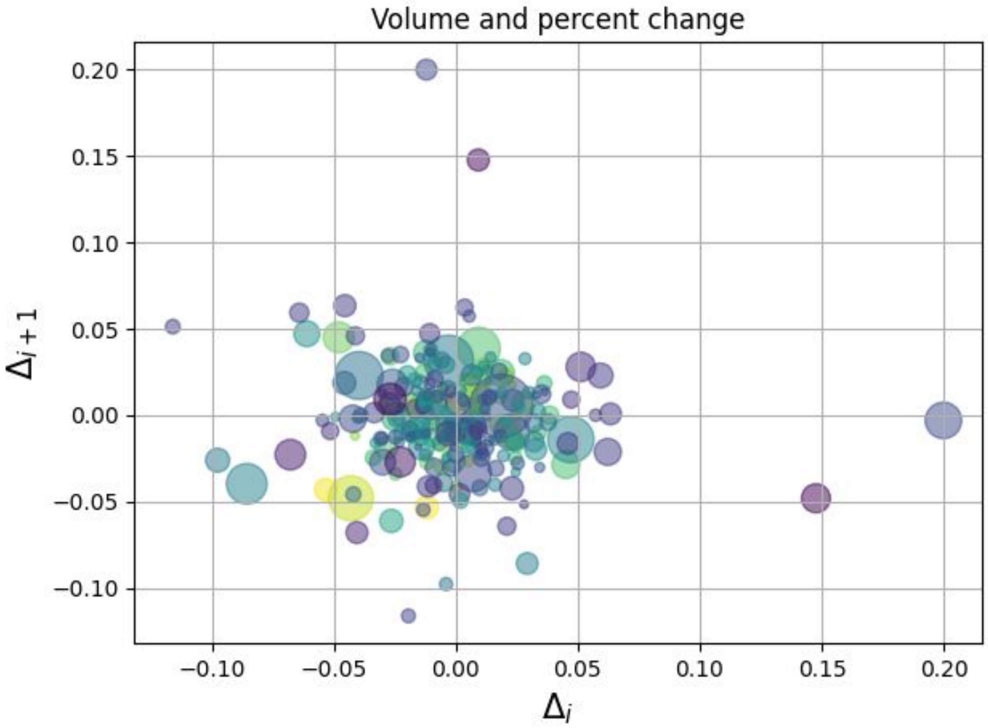
https://matplotlib.org/stable/gallery/lines_bars_and_markers/simple_plot.html legend()



https://matplotlib.org/stable/gallery/text_labels_and_annotations/legend.html

2D Scatter plots

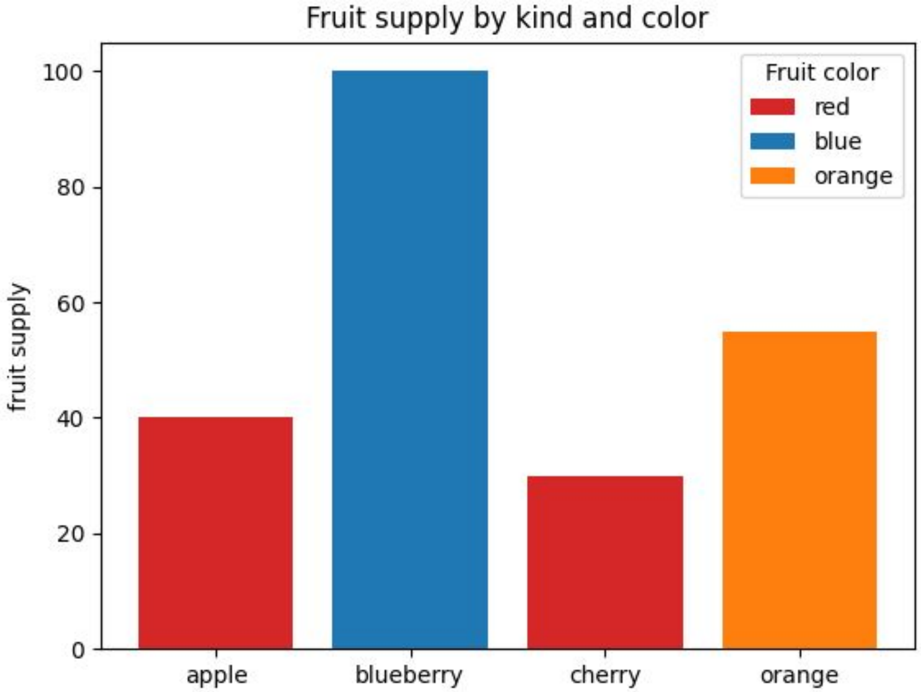
scatter()



https://matplotlib.org/stable/gallery/lines_bars_and_markers/scatter_demo2.html

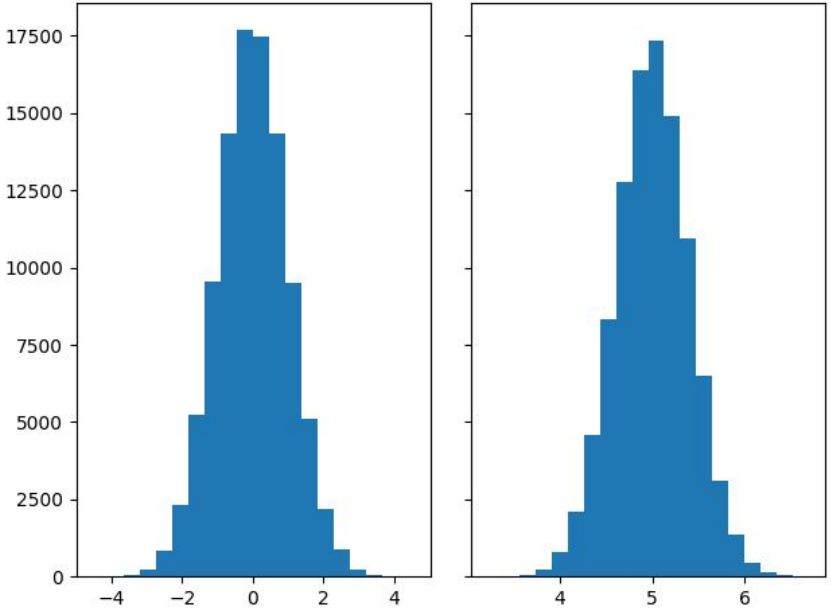
2D Bar charts

bar()



https://matplotlib.org/stable/gallery/lines_bars_and_markers/bar_colors.html

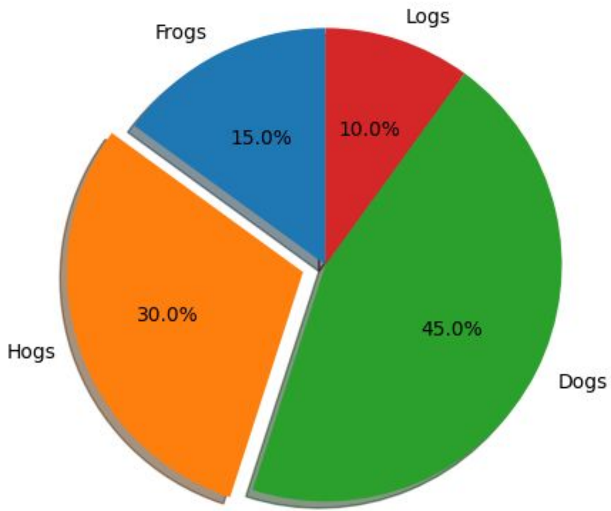
2D histogram



https://matplotlib.org/stable/gallery/statistics/hist.html

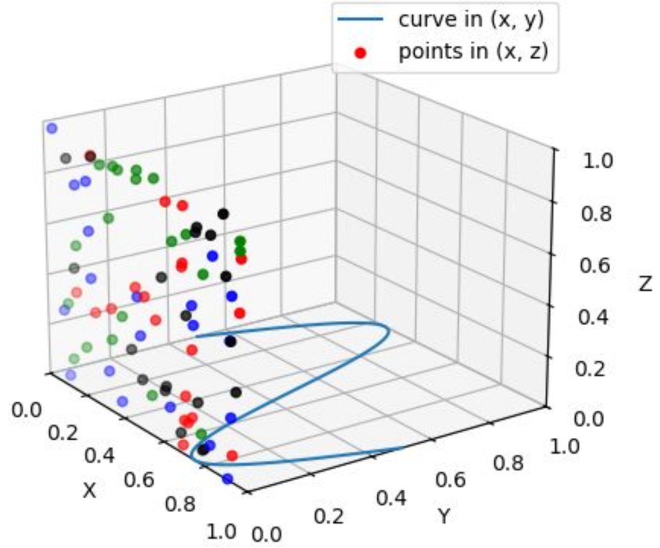
2D Pie charts

pie()



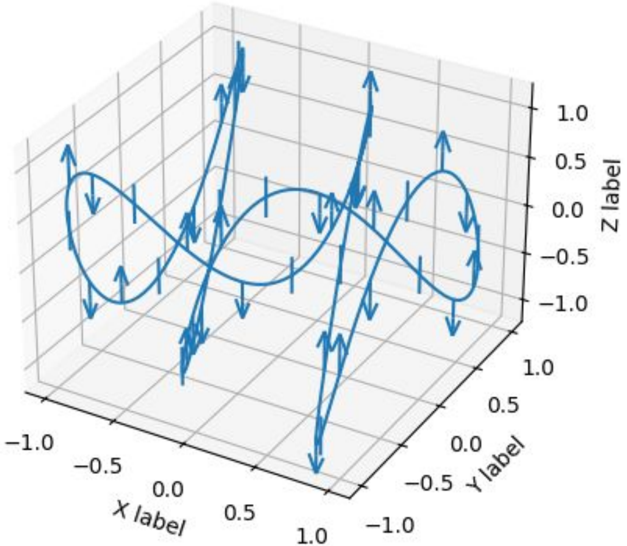
https://matplotlib.org/stable/gallery/pie_and_polar_charts/pie_features.html

3d plots



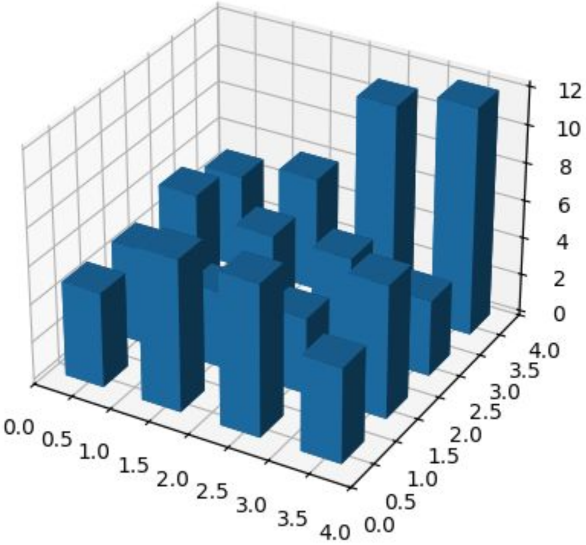
https://matplotlib.org/stable/gallery/mplot3d/2dcollections3d.html#sphx-glr-gallery-mplot3d-2dcollections3d-py

3d errorbar



https://matplotlib.org/stable/gallery/mplot3d/errorbar3d.html

3D histogram



https://matplotlib.org/stable/gallery/mplot3d/hist3d.html#sphx-glr-gallery-mplot3d-hist3d-py