Белорусско-Российский университет
Кафедра «Программное обеспечение информационных технологий»

Информатика. Программирование на Python

Тема: Python. Основы. Визуализация данных

КУТУЗОВ Виктор Владимирович

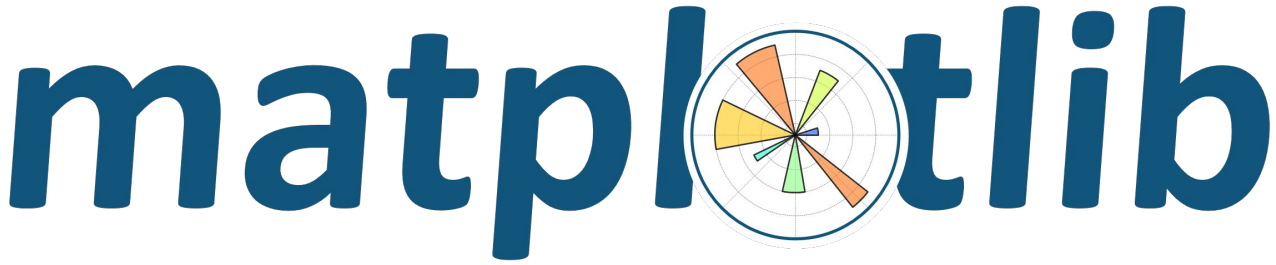
Могилев, 2021

Matplotlib

- **Matplotlib** — библиотека на языке программирования Python для визуализации данных двумерной 2D и трехмерной графики 3D.

- https://matplotlib.org/ - Официальный сайт библиотеки Matplotlib
- https://matplotlib.org/stable/contents.html - Руководство пользователя Matplotlib
- https://matplotlib.org/stable/gallery/index.html - Примеры графиков Matplotlib
- https://github.com/matplotlib/cheatsheets#cheatsheets – Шпаргалки по Matplotlib

Matplotlib



- Пакет поддерживает многие виды графиков и диаграмм:
  - Графики (line plot)
  - Диаграммы разброса (scatter plot)
  - Столбчатые диаграммы (bar chart) и гистограммы (histogram)
  - Круговые диаграммы (pie chart)
  - Ствол-лист диаграммы (stem plot)
  - Контурные графики (contour plot)
  - Поля градиентов (quiver)
  - Спектральные диаграммы (spectrogram)

Визуализация данных. Библиотека Matplotlib

- **Библиотека `Matplotlib` является одним из самых популярных средств визуализации данных на Python.**
- Она отлично подходит как для создания статичных изображений, так и анимированных, и интерактивных решений.
- Matplotlib является частью Scientific Python — набора библиотек для научных вычислений и визуализации данных, куда также входят NumPy, SciPy, Pandas, SymPy и ещё ряд других инструментов.

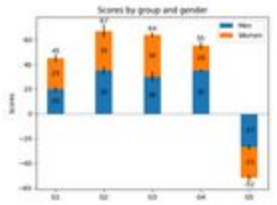
Matplotlib - модуль Pyplot

- Для построения графиков из библиотеки Matplotlib нужно импортировать модуль **Pyplot**.
- **Pyplot** это набор команд, созданных для построения графиков функций и уравнений.
- Для удобного построения графиков так же можно использовать библиотеку **NumPy**.

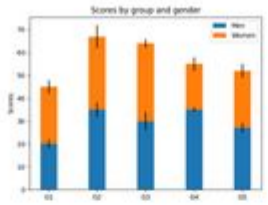
Примеры применения

Matplotlib – Gallery

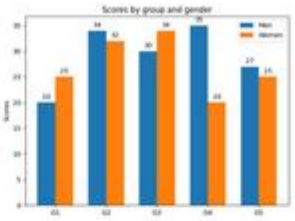
Matplotlib – Gallery - Lines, bars and markers



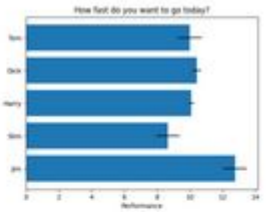
Bar Label Demo



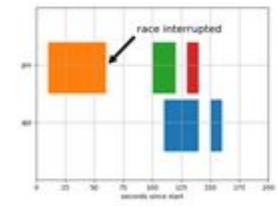
Stacked bar chart



Grouped bar chart with labels

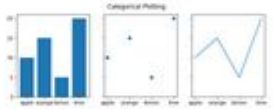


Horizontal bar chart

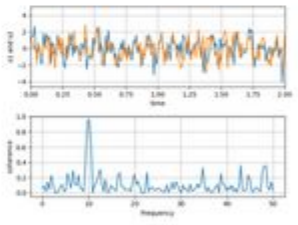


Broken Barh

CapStyle



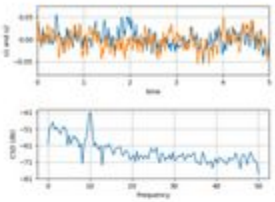
Plotting categorical variables



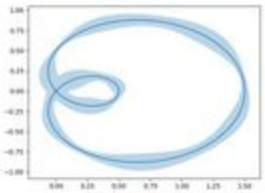
Plotting the coherence of two signals

https://matplotlib.org/stable/gallery/index.html

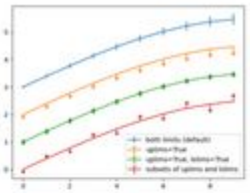
Matplotlib – Gallery - Lines, bars and markers



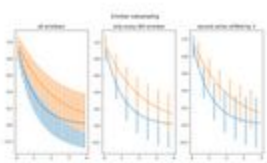
CSD Demo



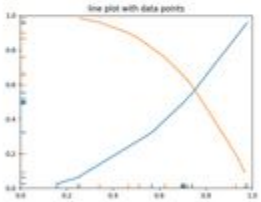
Curve with error band



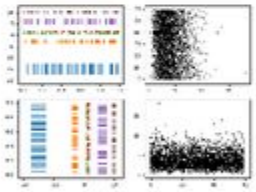
Errorbar limit selection



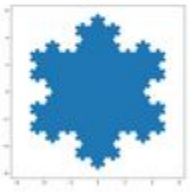
Errorbar subsampling



EventCollection Demo



Eventplot Demo

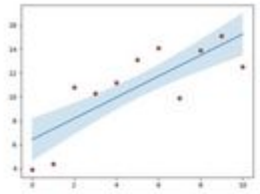


Filled polygon

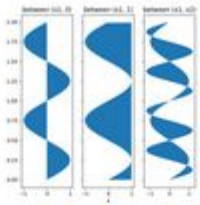
Fill Between and Alpha

https://matplotlib.org/stable/gallery/index.html

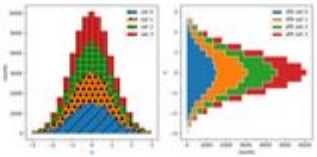
Matplotlib – Gallery - Lines, bars and markers



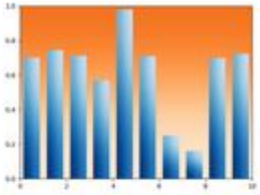
Filling the area between lines



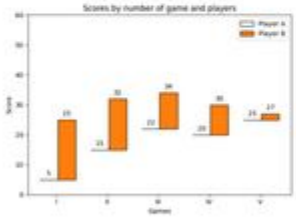
Fill Betweenx Demo



Hatch-filled histograms

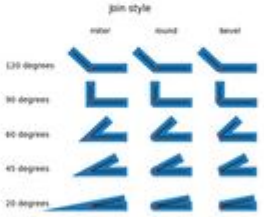


Bar chart with gradients

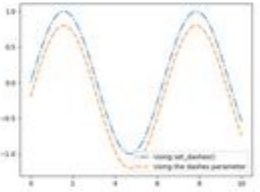


Hat graph

Discrete distribution as horizontal bar chart

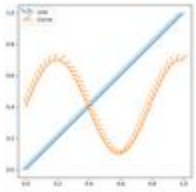


JoinStyle



Customizing dashed line styles

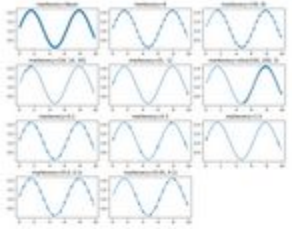
Matplotlib – Gallery - Lines, bars and markers



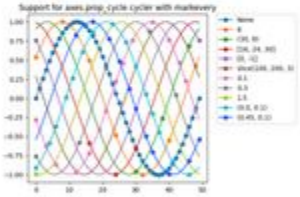
Lines with a ticked patheffect

Linestyles

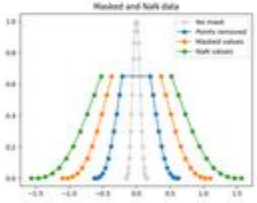
Marker reference



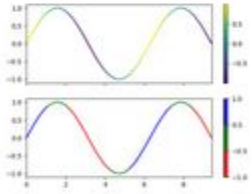
Markevery Demo



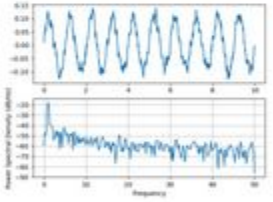
prop_cycle property markevery in rcParams



Plotting masked and NaN values



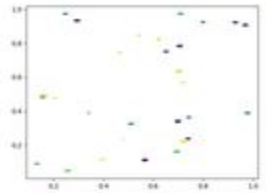
Multicolored lines



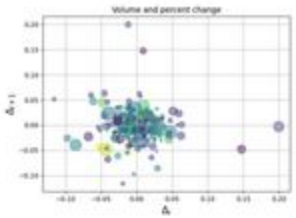
Psd Demo

https://matplotlib.org/stable/gallery/index.html

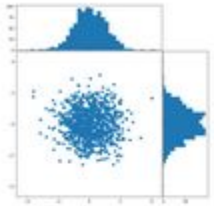
Matplotlib – Gallery - Lines, bars and markers



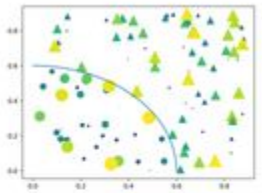
Scatter Custom Symbol



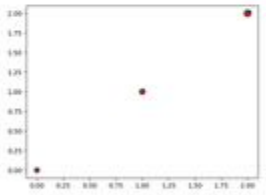
Scatter Demo2



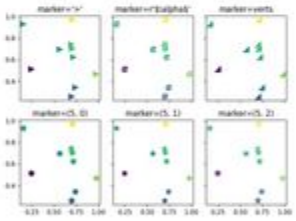
Scatter plot with histograms



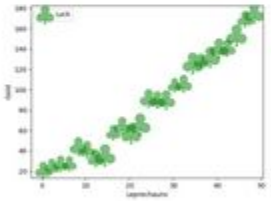
Scatter Masked



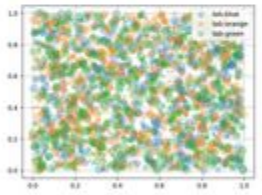
Scatter plot with pie chart markers



Marker examples



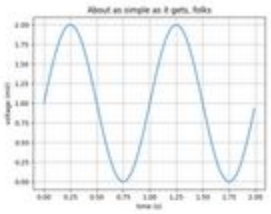
Scatter Symbol



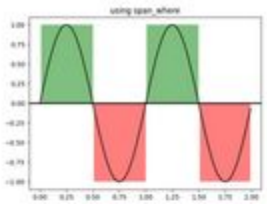
Scatter plots with a legend

https://matplotlib.org/stable/gallery/index.html

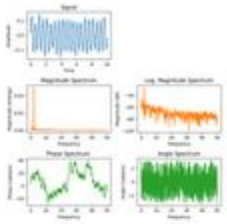
Matplotlib – Gallery - Lines, bars and markers



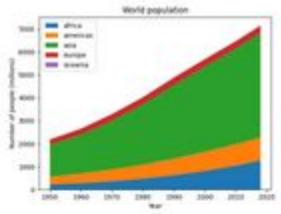
Simple Plot



Using span_where

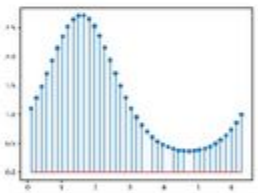


Spectrum Representations

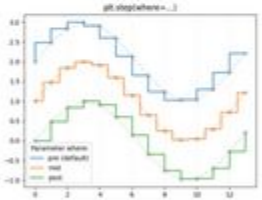


Stackplots and streamgraphs

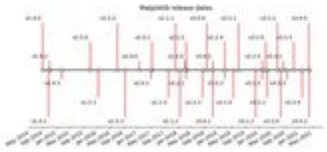
Stairs Demo



Stem Plot



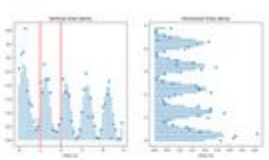
Step Demo



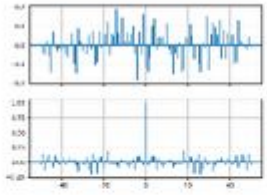
Creating a timeline with lines, dates, and text

https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Lines, bars and markers



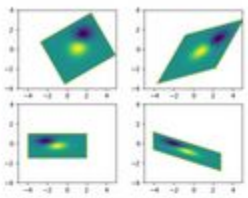
hlines and vlines



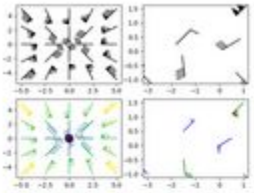
Cross- and Auto-Correlation Demo

https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Images, contours and fields

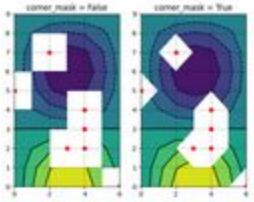


Affine transform of an image

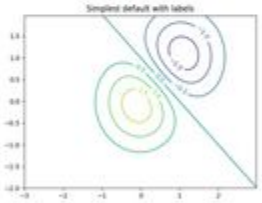


Wind Barbs

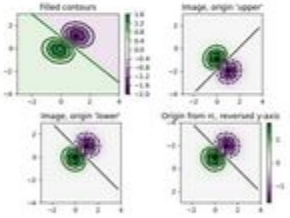
Barcode



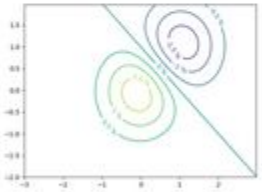
Contour Corner Mask



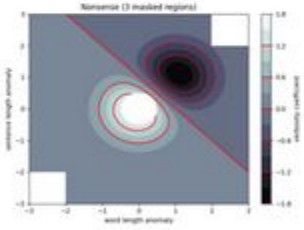
Contour Demo



Contour Image



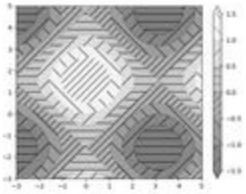
Contour Label Demo



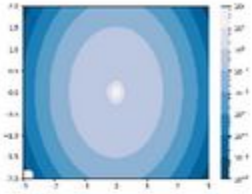
Contourf Demo

https://matplotlib.org/stable/gallery/index.html

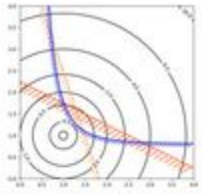
Matplotlib – Gallery - Images, contours and fields



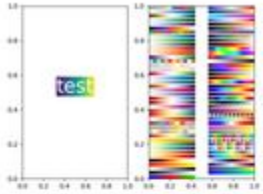
Contourf Hatching



Contourf and log color scale

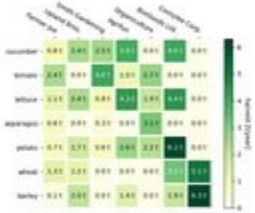


Contouring the solution space of optimizations



BboxImage Demo

Figimage Demo



Creating annotated heatmaps

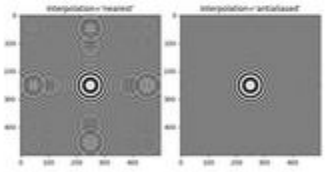


Image antialiasing

Clipping images with patches

https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Images, contours and fields

Image Demo

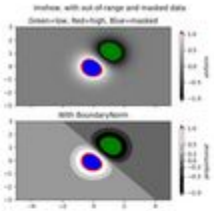


Image Masked

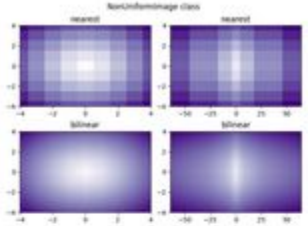
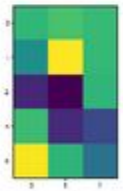
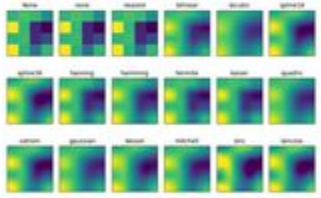


Image Nonuniform

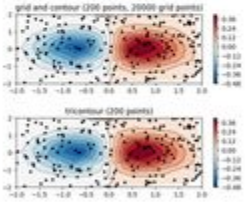
Blend transparency with color in 2D images



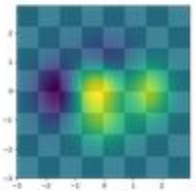
Modifying the coordinate formatter



Interpolations for imshow



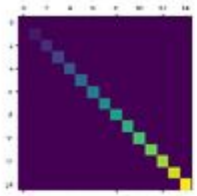
Contour plot of irregularly spaced data



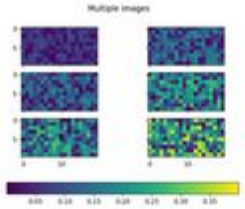
Layer Images

https://matplotlib.org/stable/gallery/index.html

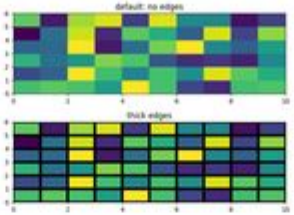
Matplotlib – Gallery - Images, contours and fields



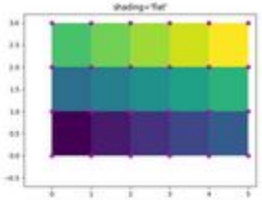
Matshow



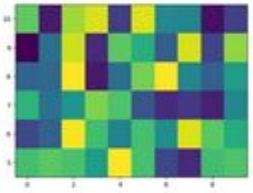
Multi Image



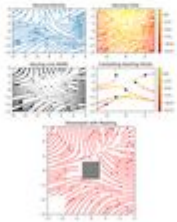
Pcolor Demo



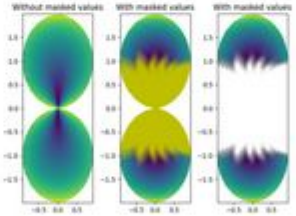
pcolormesh grids and shading



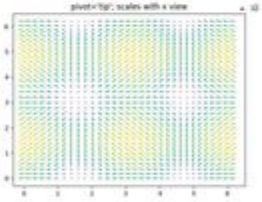
pcolormesh



Streamplot



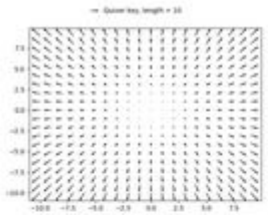
QuadMesh Demo



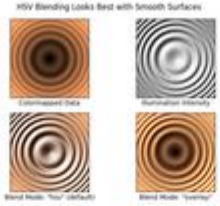
Advanced quiver and quiverkey functions

https://matplotlib.org/stable/gallery/index.html

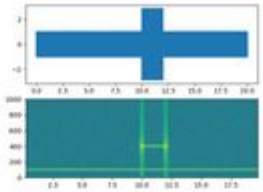
Matplotlib – Gallery - Images, contours and fields



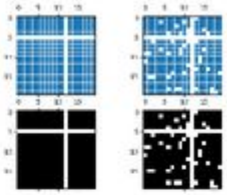
Quiver Simple Demo



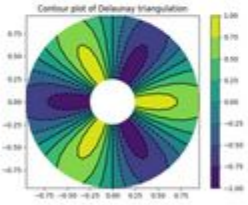
Shading example



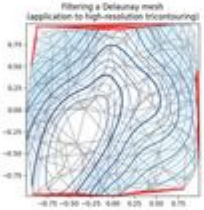
Spectrogram Demo



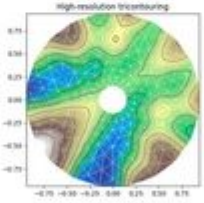
Spy Demos



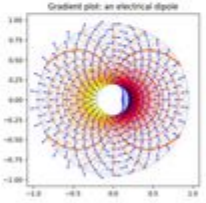
Tricontour Demo



Tricontour Smooth Delaunay



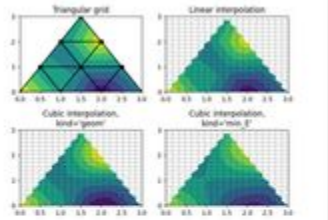
Tricontour Smooth User



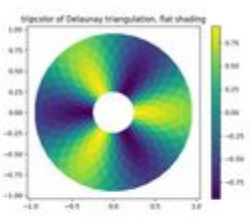
Trigradient Demo

https://matplotlib.org/stable/gallery/index.html

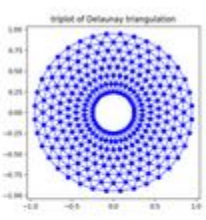
Matplotlib – Gallery - Images, contours and fields



Triinterp Demo



Tripcolor Demo

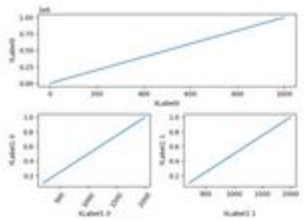


Triplot Demo

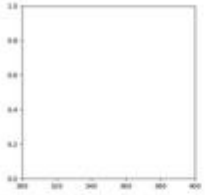
Watermark image

https://matplotlib.org/stable/gallery/index.html

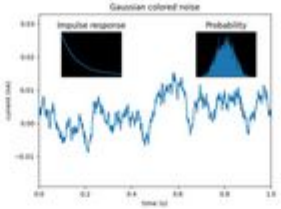
Matplotlib – Gallery - Subplots, axes and figures



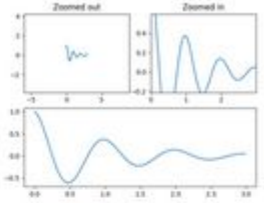
Aligning Labels



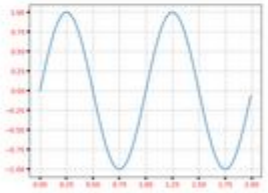
Axes box aspect



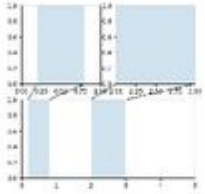
Axes Demo



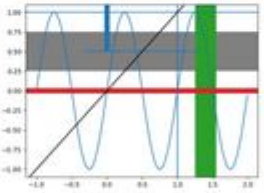
Controlling view limits using margins and sticky_edges



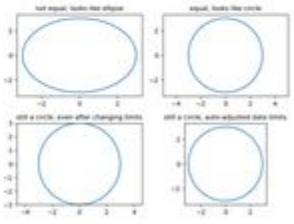
Axes Props



Axes Zoom Effect



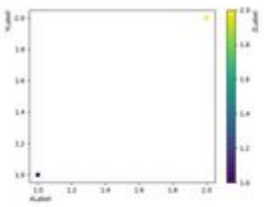
axhspan Demo



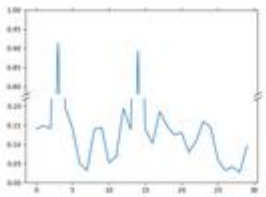
Equal axis aspect ratio

https://matplotlib.org/stable/gallery/index.html

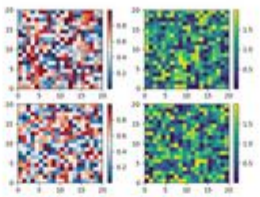
Matplotlib – Gallery - Subplots, axes and figures



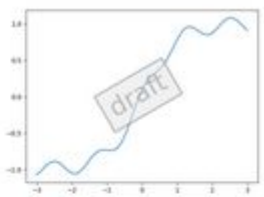
Axis Label Position



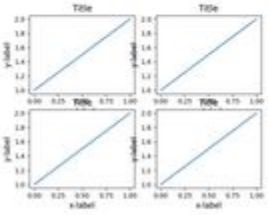
Broken Axis



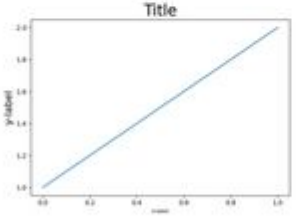
Placing Colorbars



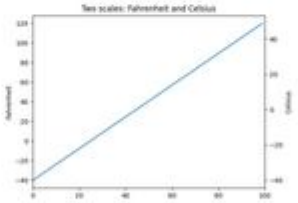
Custom Figure subclasses



Resizing axes with constrained layout



Resizing axes with tight layout



Different scales on the same axes

Figure size in different units

https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Subplots, axes and figures

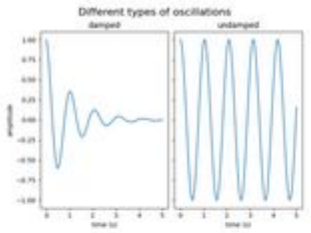
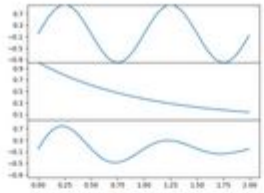
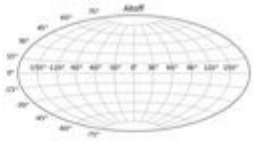


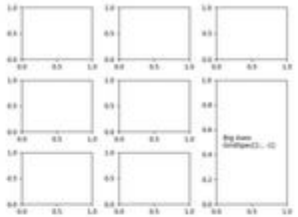
Figure labels: suptitle, supxlabel, supylabel



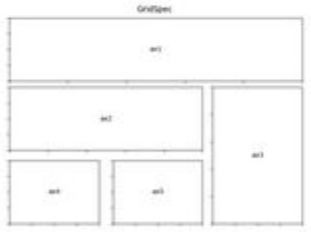
Creating adjacent subplots



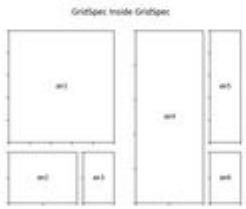
Geographic Projections



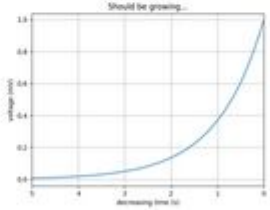
Combining two subplots using subplots and GridSpec



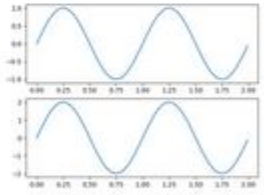
Using Gridspec to make multi-column/row subplot layouts



Nested Gridspecs



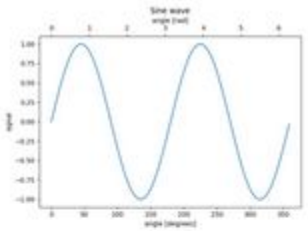
Invert Axes



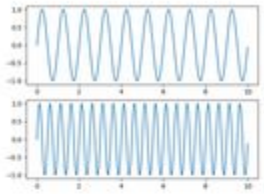
Managing multiple figures in pyplot

https://matplotlib.org/stable/gallery/index.html

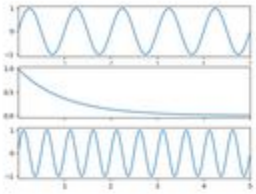
Matplotlib – Gallery - Subplots, axes and figures



Secondary Axis



Sharing axis limits and views



Shared Axis

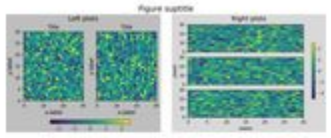
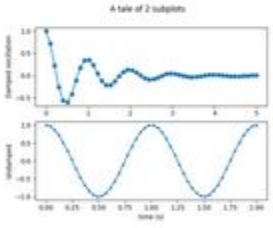
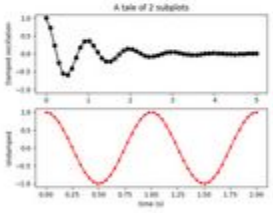


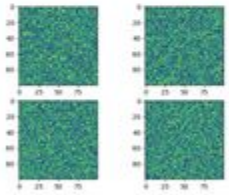
Figure subfigures



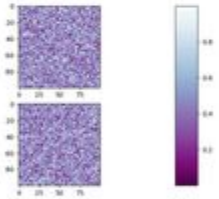
Multiple subplots



Basic Subplot Demo



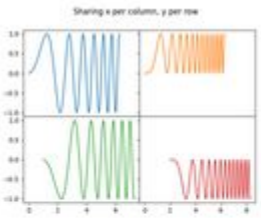
Subplot Toolbar



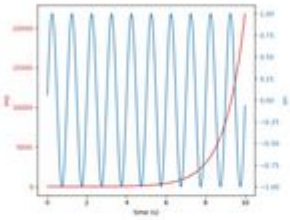
Subplots Adjust

https://matplotlib.org/stable/gallery/index.html

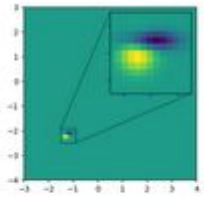
Matplotlib – Gallery - Subplots, axes and figures



Creating multiple subplots using plt.subplots



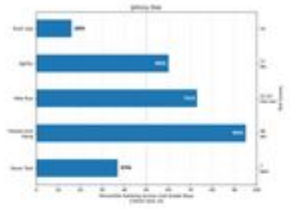
Plots with different scales



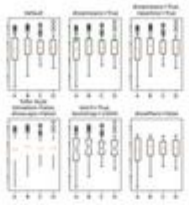
Zoom region inset axes

https://matplotlib.org/stable/gallery/index.html

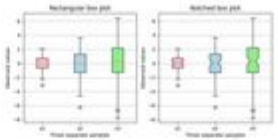
Matplotlib – Gallery - Statistics



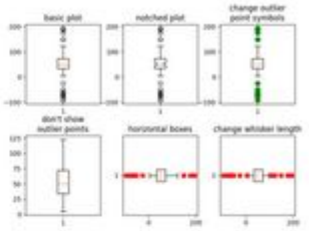
Percentiles as horizontal bar chart



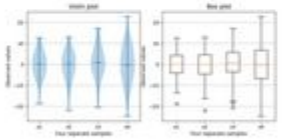
Artist customization in box plots



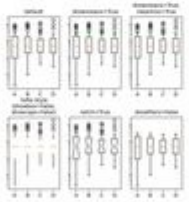
Box plots with custom fill colors



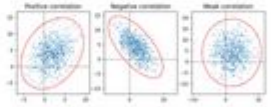
Boxplots



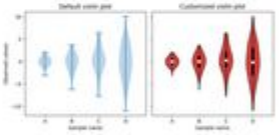
Box plot vs. violin plot comparison



Boxplot drawer function



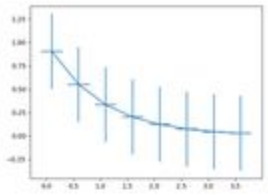
Plot a confidence ellipse of a two-dimensional dataset



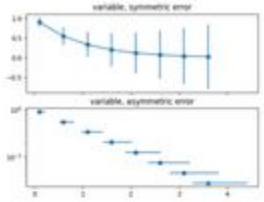
Violin plot customization

https://matplotlib.org/stable/gallery/index.html

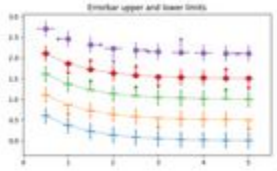
Matplotlib – Gallery - Statistics



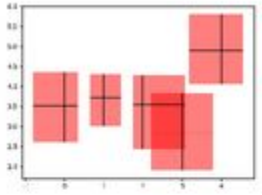
Errorbar function



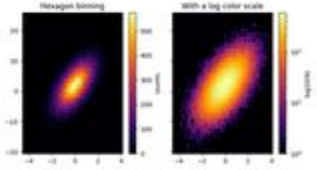
Different ways of specifying error bars



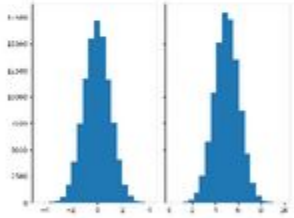
Including upper and lower limits in error bars



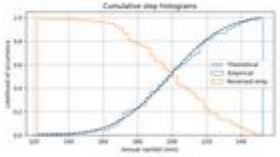
Creating boxes from error bars using PatchCollection



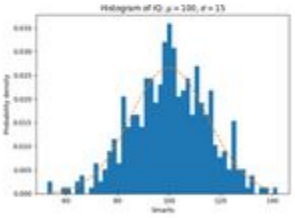
Hexbin Demo



Histograms



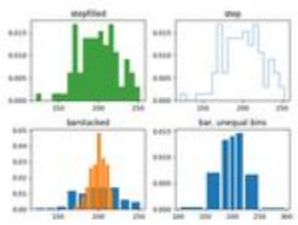
Using histograms to plot a cumulative distribution



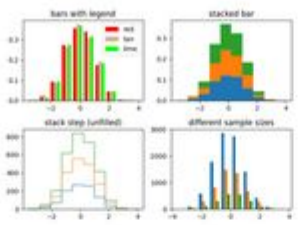
Some features of the histogram (hist) function

https://matplotlib.org/stable/gallery/index.html

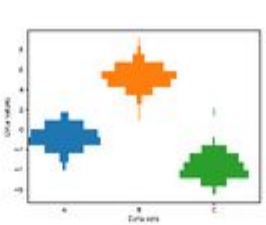
Matplotlib – Gallery - Statistics



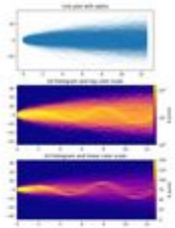
Demo of the histogram function's different histtype settings



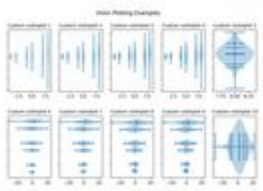
The histogram (hist) function with multiple data sets



Producing multiple histograms side by side



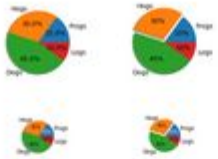
Time Series Histogram



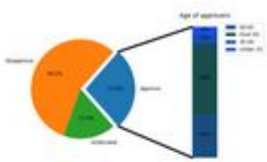
Violin plot basics

Matplotlib – Gallery - Pie and polar charts

Basic pie chart

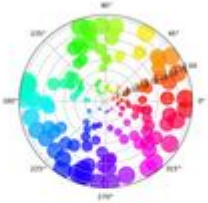


Pie Demo2

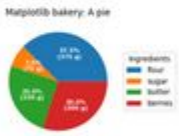


Bar of pie

Nested pie charts

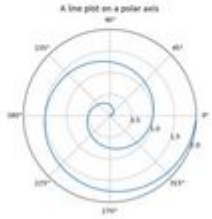


Scatter plot on polar axis

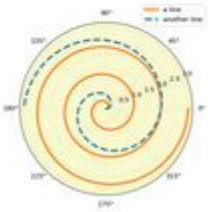


Labeling a pie and a donut

Bar chart on polar axis

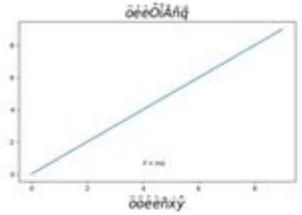


Polar plot

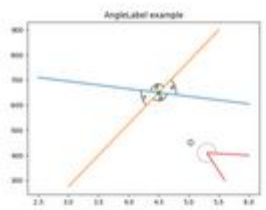


Polar Legend

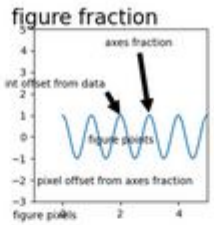
Matplotlib – Gallery - Text, labels and annotations



Using accented text in matplotlib

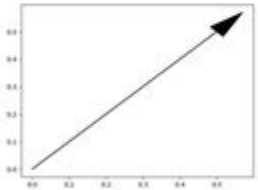


Scale invariant angle label

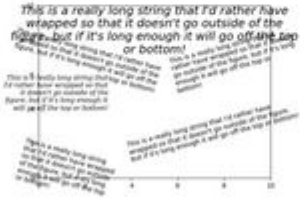


Annotating Plots

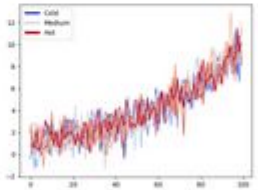
Arrow Demo



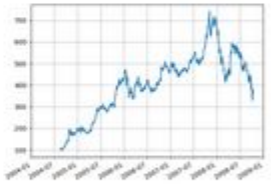
Arrow Simple Demo



Auto-wrapping text



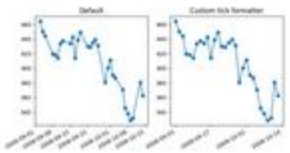
Composing Custom Legends



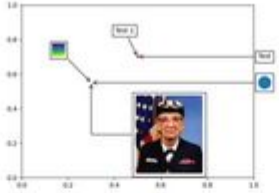
Date tick labels

https://matplotlib.org/stable/gallery/index.html

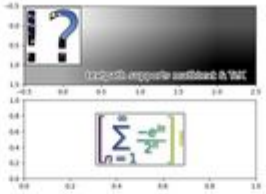
Matplotlib – Gallery - Text, labels and annotations



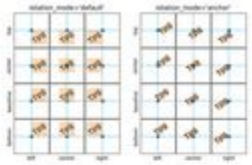
Custom tick formatter for time series



Demo Annotation Box

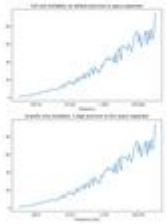


Using a text as a Path

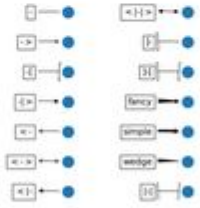


Text Rotation Mode

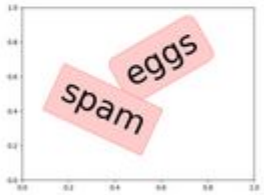
The difference between \dfrac and \frac



Labeling ticks using engineering notation



Annotation arrow style reference



Styling text boxes

https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Text, labels and annotations

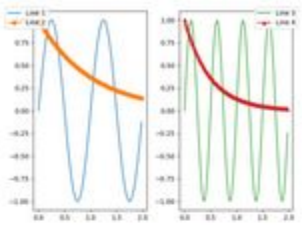
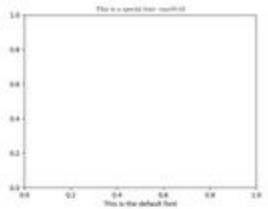
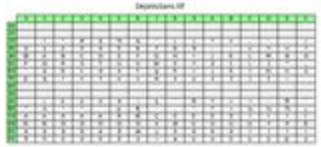


Figure legend demo

Configuring the font family



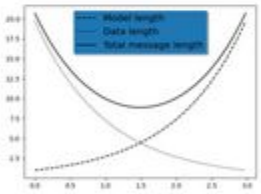
Using a ttf font file in Matplotlib



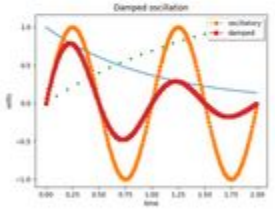
Font table

Fonts demo (object-oriented style)

Fonts demo (kwargs)



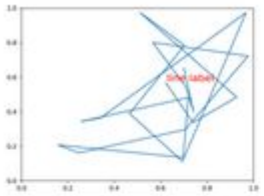
Legend using pre-defined labels



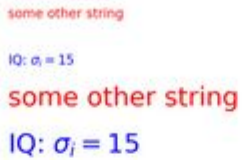
Legend Demo

https://matplotlib.org/stable/gallery/index.html

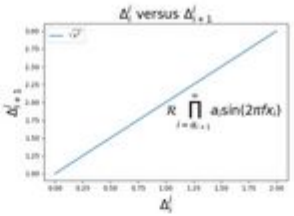
Matplotlib – Gallery - Text, labels and annotations



Artist within an artist

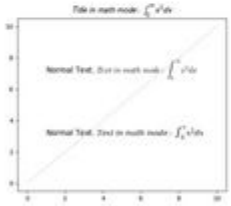


Convert texts to images

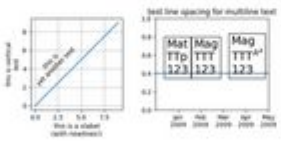


Mathtext

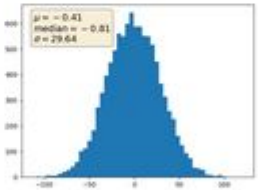
Mathtext Examples



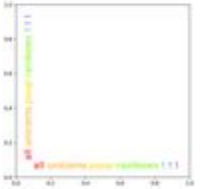
Math fontfamily



Multiline



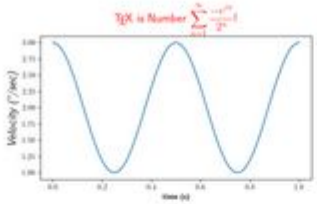
Placing text boxes



Rainbow text

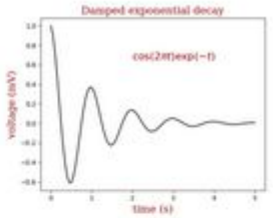
Matplotlib – Gallery - Text, labels and annotations

STIX Fonts

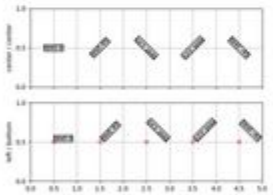


Rendering math equations using TeX

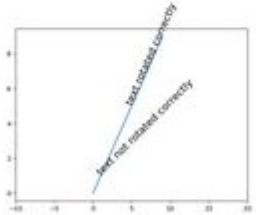
Precise text layout



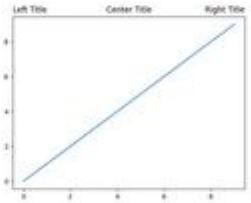
Controlling style of text and labels using a dictionary



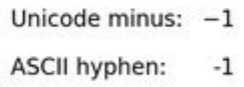
Default text rotation demonstration



Text Rotation Relative To Line



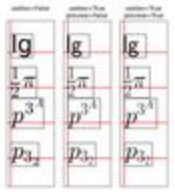
Title positioning



Unicode minus

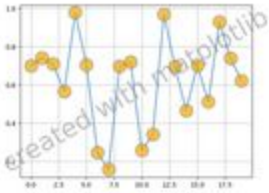
https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Text, labels and annotations



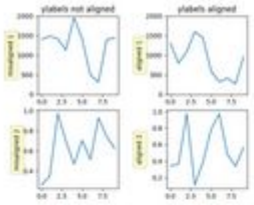
Usetex Baseline Test

Usetex Fonteffects

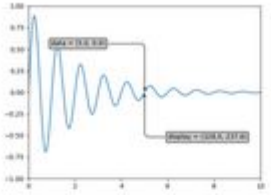


Text watermark

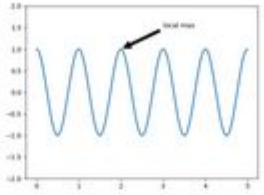
Matplotlib – Gallery - Pyplot



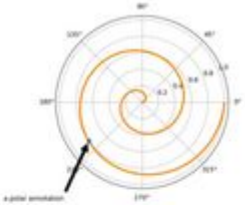
Align y-labels



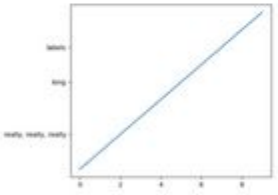
Annotate Transform



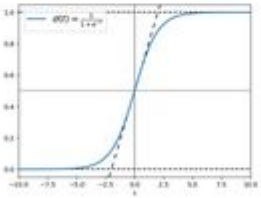
Annotating a plot



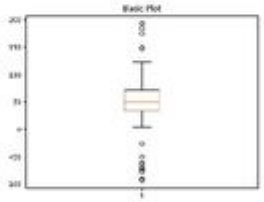
Annotation Polar



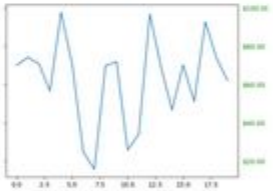
Auto Subplots Adjust



Infinite lines



Boxplot Demo



Dollar Ticks

https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Pyplot

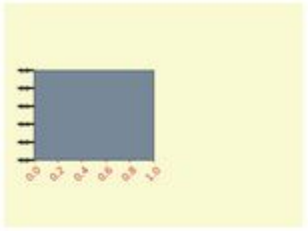
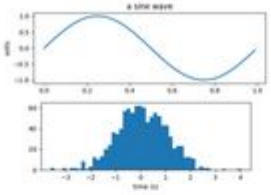
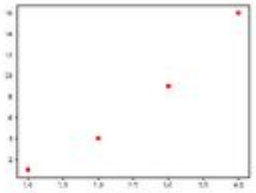


Fig Axes Customize Simple

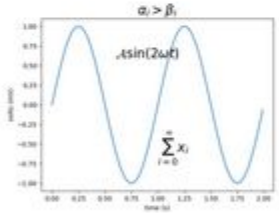


Simple axes labels

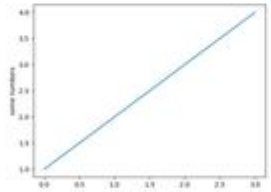
Adding lines to figures



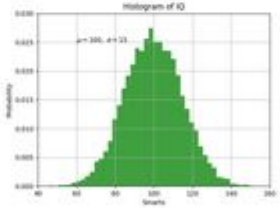
plot() format string



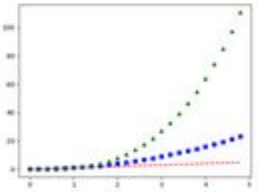
Pyplot Mathtext



Pyplot Simple



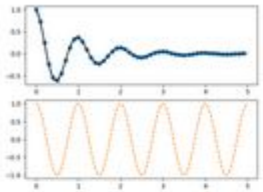
Pyplot Text



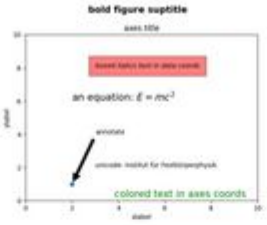
Pyplot Three

https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Pyplot

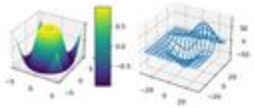


Pyplot Two Subplots

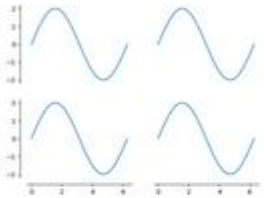


Text Commands

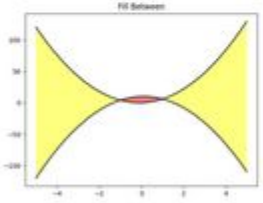
Text Layout



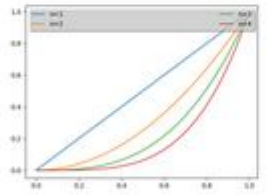
What's New 1 Subplot3d



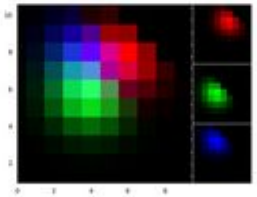
What's New 0.99 Spines



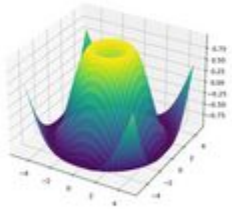
Fill Between



What's New 0.98.4 Legend



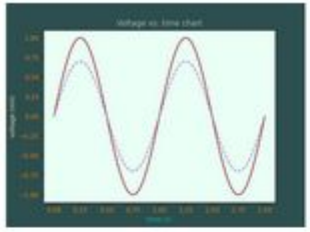
What's New 0.99 Axes Grid



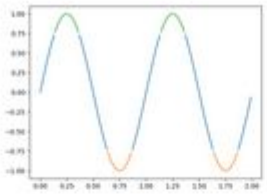
What's New 0.99 Mplot3d

https://matplotlib.org/stable/gallery/index.html

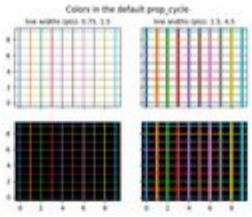
Matplotlib – Gallery - Color



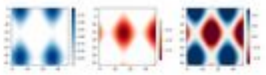
Color Demo



Color by y-value

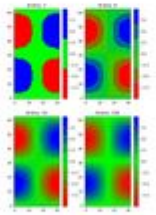


Colors in the default property cycle



Colorbar

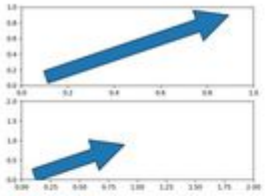
Colormap reference



Creating a colormap from a list of colors

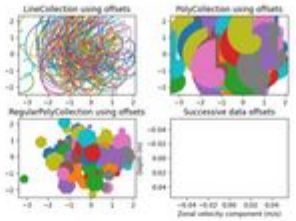
List of named colors

Matplotlib – Gallery - Shapes and collections

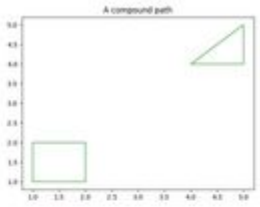


Arrow guide

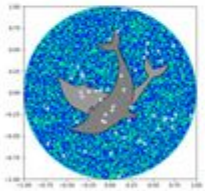
Reference for Matplotlib artists



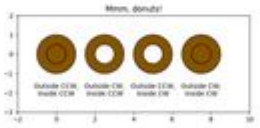
Line, Poly and RegularPoly Collection with autoscaling



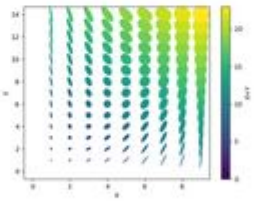
Compound path



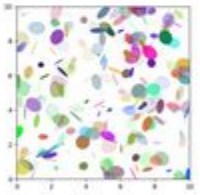
Dolphins



Mmh Donuts!!!



Ellipse Collection

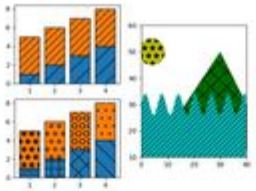


Ellipse Demo

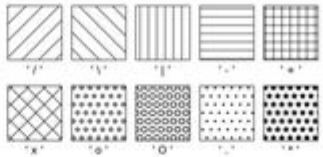
https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Shapes and collections

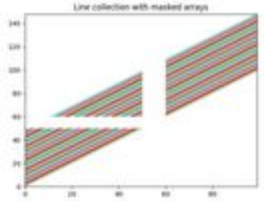
Drawing fancy boxes



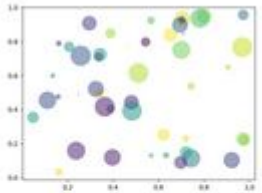
Hatch demo



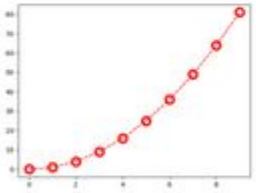
Hatch style reference



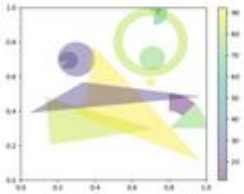
Line Collection



Scatter plot

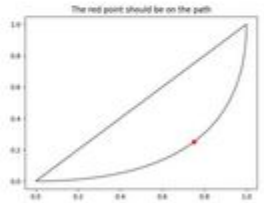


Marker Path



Circles, Wedges and Polygons

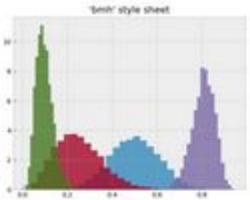
PathPatch object



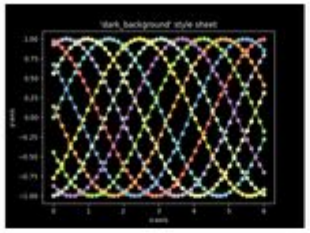
Bezier Curve

https://matplotlib.org/stable/gallery/index.html

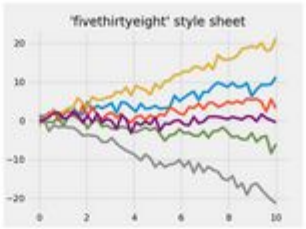
Matplotlib – Gallery - Style sheets



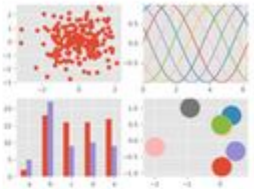
Bayesian Methods for Hackers style sheet



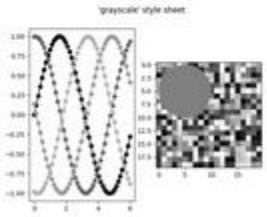
Dark background style sheet



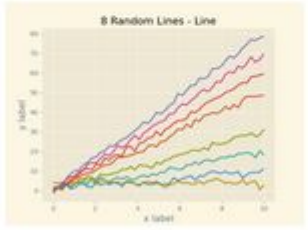
FiveThirtyEight style sheet



ggplot style sheet



Grayscale style sheet

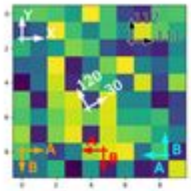


Solarized Light stylesheet

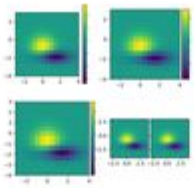
Style sheets reference

https://matplotlib.org/stable/gallery/index.html

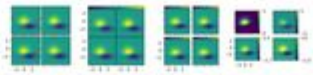
Matplotlib – Gallery - Axes Grid



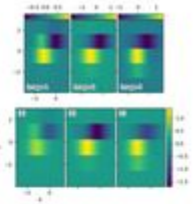
Anchored Direction Arrow



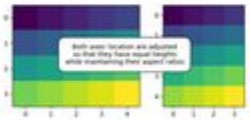
Axes Divider



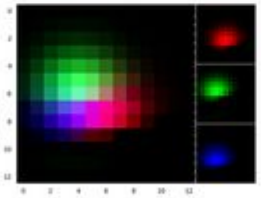
Demo Axes Grid



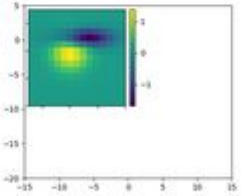
Axes Grid2



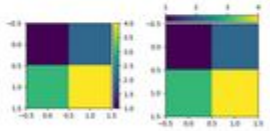
HBoxDivider demo



Showing RGB channels using RGBAxes



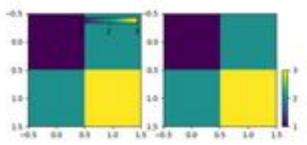
Demo Colorbar of Inset Axes



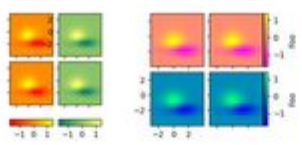
Colorbar with AxesDivider

https://matplotlib.org/stable/gallery/index.html

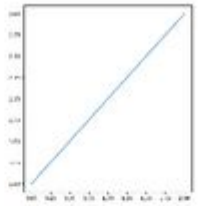
Matplotlib – Gallery - Axes Grid



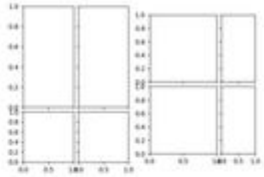
Controlling the position and size of colorbars with Inset Axes



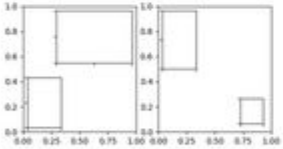
Demo Edge Colorbar



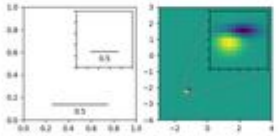
Demo Fixed Size Axes



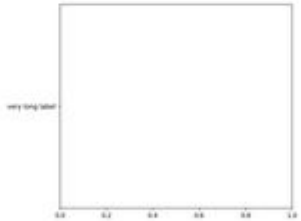
Demo Imagegrid Aspect



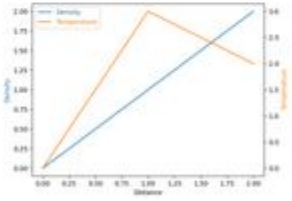
Inset Locator Demo



Inset Locator Demo2



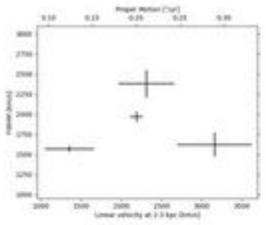
Make Room For Ylabel Using Axesgrid



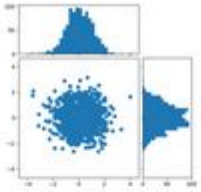
Parasite Simple

https://matplotlib.org/stable/gallery/index.html

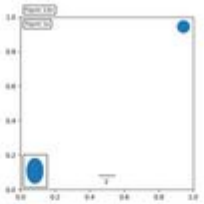
Matplotlib – Gallery - Axes Grid



Parasite Simple2

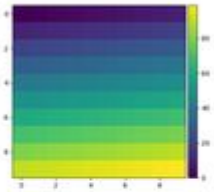


Scatter Histogram (Locatable Axes)

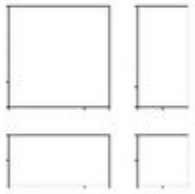


Simple Anchored Artists

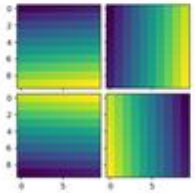
Simple Axes Divider 1



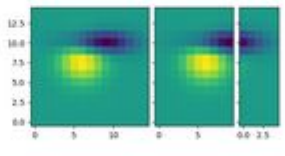
Simple Colorbar



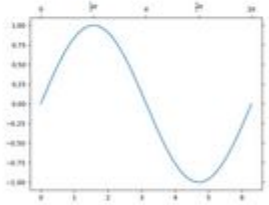
Simple Axes Divider 3



Simple ImageGrid



Simple ImageGrid 2

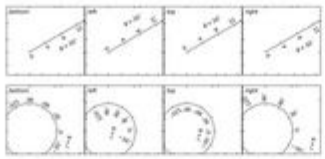


Simple Axisline4

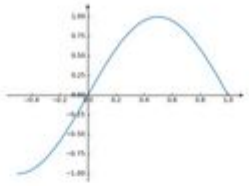
https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Axis Artist

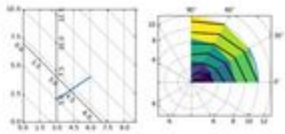
Axis Direction



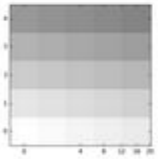
Demo Axis Direction



Axis line styles

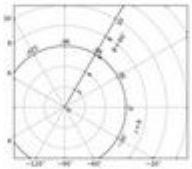


Curvilinear grid demo

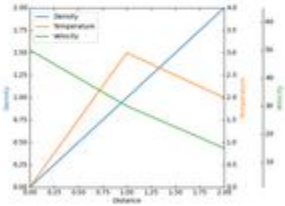


Demo CurveLinear Grid2

mpl_toolkits.axisartist.flo features



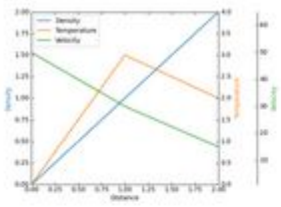
Demo Floating Axis



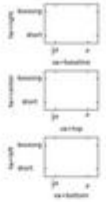
Parasite Axes demo

https://matplotlib.org/stable/gallery/index.html

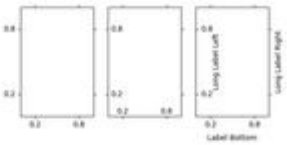
Matplotlib – Gallery - Axis Artist



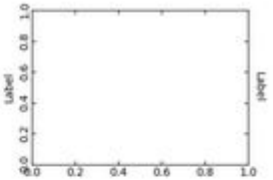
Parasite axis demo



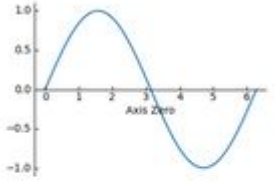
Ticklabel alignment



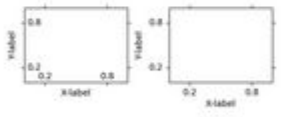
Ticklabel direction



Simple Axis Direction01

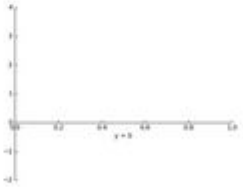


Simple Axisline2

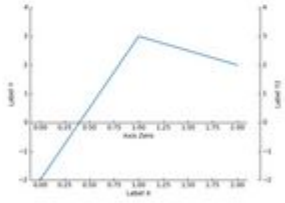


Simple Axis Direction03

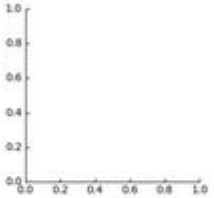
Simple Axis Pad



Simple Axisartist1

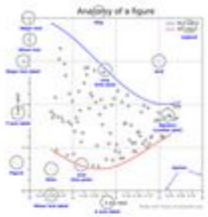


Simple Axisline



Simple Axisline3

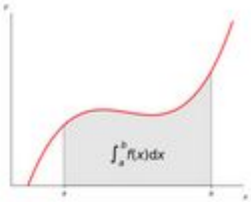
Matplotlib – Gallery - Showcase



Anatomy of a figure

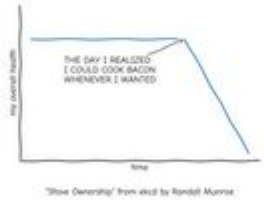
Bachelor's degrees by gender

Firefox



Integral as the area under a curve

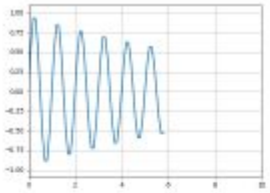
Shaded & power normalized rendering



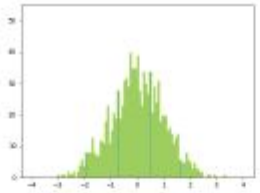
XKCD

https://matplotlib.org/stable/gallery/index.html

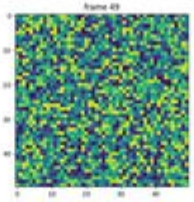
Matplotlib – Gallery - Animation



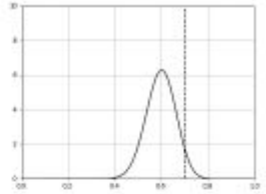
Decay



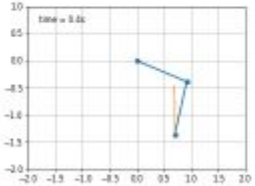
Animated histogram



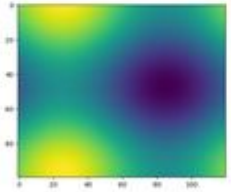
pyplot animation



The Bayes update

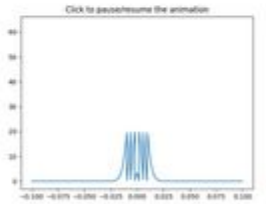


The double pendulum problem



Animated image using a precomputed list of images

Frame grabbing

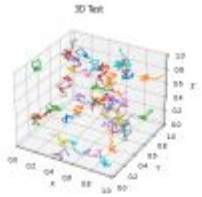


Pausing and Resuming an Animation

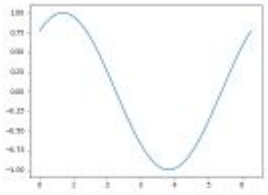
https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - Animation

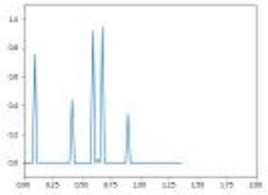
Rain simulation



Animated 3D random walk



Animated line plot

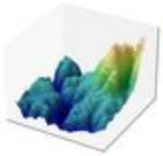


Oscilloscope

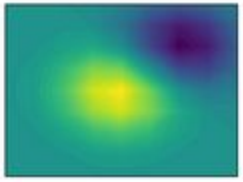
MATPLOTLIB UNCHAINED

https://matplotlib.org/stable/gallery/index.html

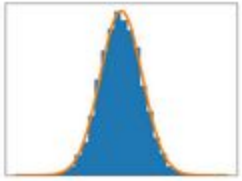
Matplotlib – Gallery - Front Page



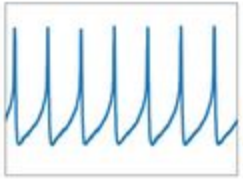
Frontpage 3D example



Frontpage contour example

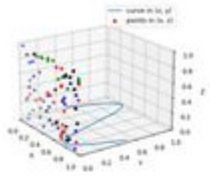


Frontpage histogram example

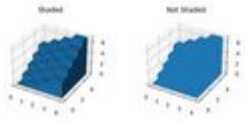


Frontpage plot example

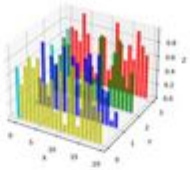
Matplotlib – Gallery - 3D plotting



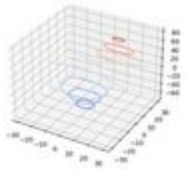
Plot 2D data on 3D plot



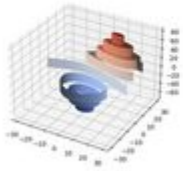
Demo of 3D bar charts



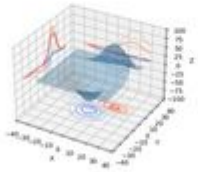
Create 2D bar graphs in different planes



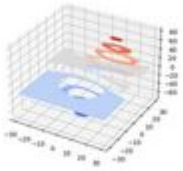
Demonstrates plotting contour (level) curves in 3D



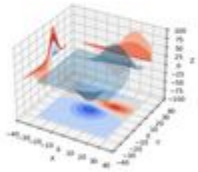
Demonstrates plotting contour (level) curves in 3D using the extend3d



Projecting contour profiles onto a graph



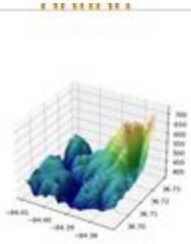
Filled contours



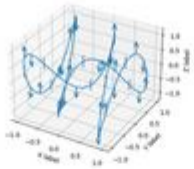
Projecting filled contour onto a graph

https://matplotlib.org/stable/gallery/index.html

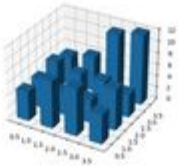
Matplotlib – Gallery - 3D plotting



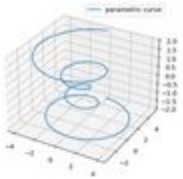
Custom hillshading in a 3D surface plot



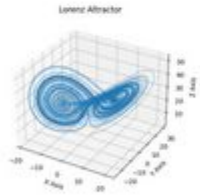
3D errorbars



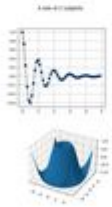
Create 3D histogram of 2D data



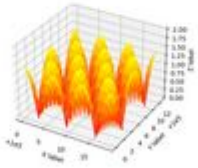
Parametric Curve



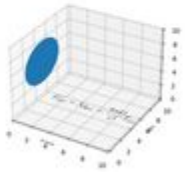
Lorenz Attractor



2D and 3D Axes in same Figure



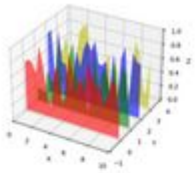
Automatic Text Offsetting



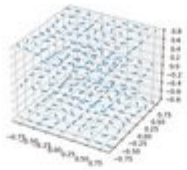
Draw flat objects in 3D plot

https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery - 3D plotting

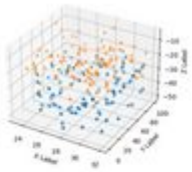


Generate polygons to fill under 3D line graph

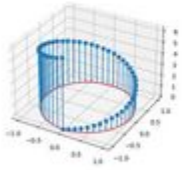


3D quiver plot

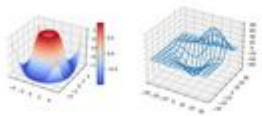
Rotating a 3D plot



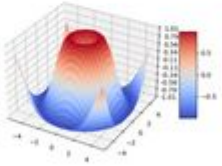
3D scatterplot



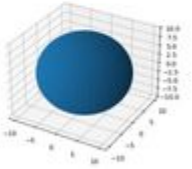
3D stem



3D plots as subplots

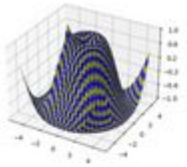


3D surface (colormap)

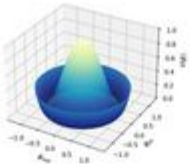


3D surface (solid color)

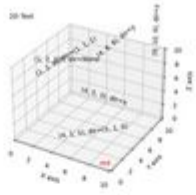
Matplotlib – Gallery - 3D plotting



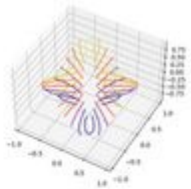
3D surface (checkerboard)



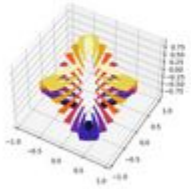
3D surface with polar coordinates



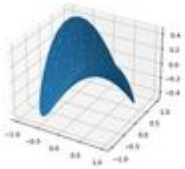
Text annotations in 3D



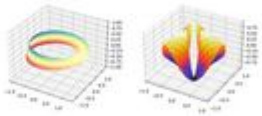
Triangular 3D contour plot



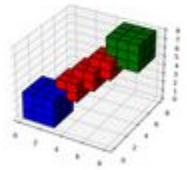
Triangular 3D filled contour plot



Triangular 3D surfaces



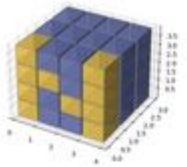
More triangular 3D surfaces



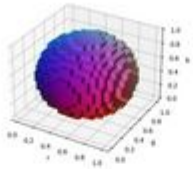
3D voxel / volumetric plot

https://matplotlib.org/stable/gallery/index.html

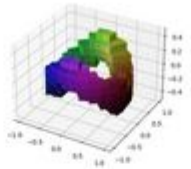
Matplotlib – Gallery - 3D plotting



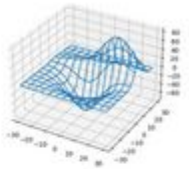
3D voxel plot of the numpy logo



3D voxel / volumetric plot with rgb colors

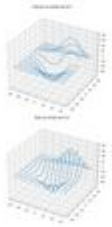


3D voxel / volumetric plot with cylindrical coordinates



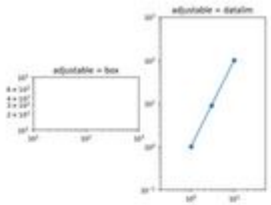
3D wireframe plot

Rotating 3D wireframe plot

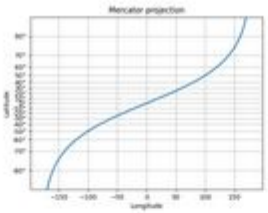


3D wireframe plots in one direction

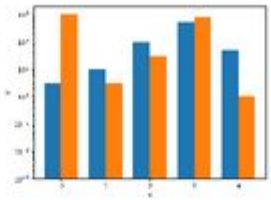
Matplotlib – Gallery - Scales



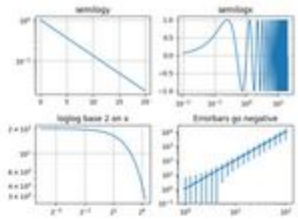
Loglog Aspect



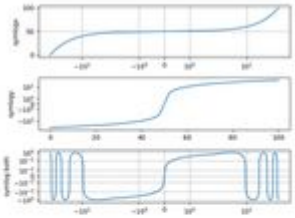
Custom scale



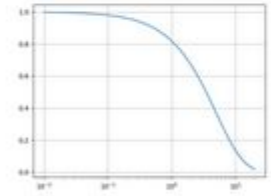
Log Bar



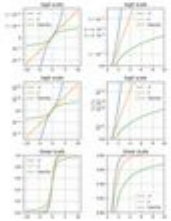
Log Demo



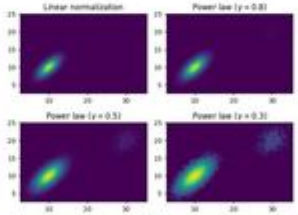
Symlog Demo



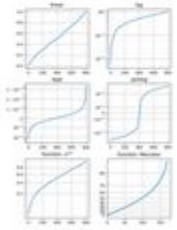
Log Axis



Logit Demo

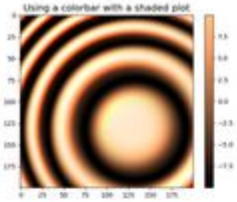


Exploring normalizations

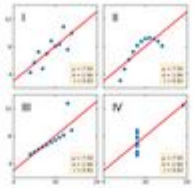


Scales

Matplotlib – Gallery - Specialty Plots

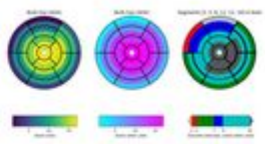


Hillshading

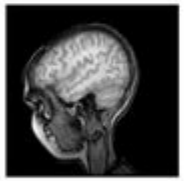


Anscombe's quartet

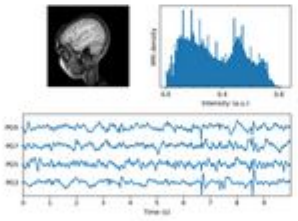
Hinton diagrams



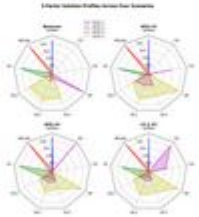
Left ventricle bullseye



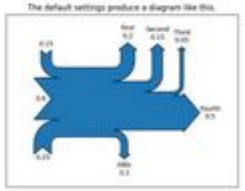
MRI



MRI With EEG



Radar chart (aka spider or star chart)

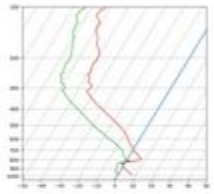


The Sankey class

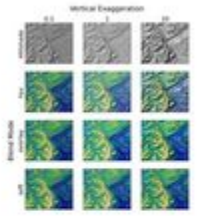
Matplotlib – Gallery - Specialty Plots

Long chain of connections using Sankey

Rankine power cycle



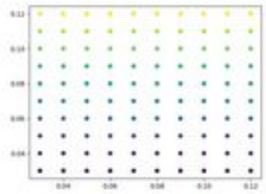
SkewT-logP diagram: using transforms and custom projections



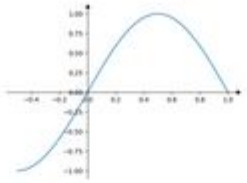
Topographic hillshading

https://matplotlib.org/stable/gallery/index.html

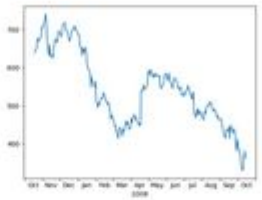
Matplotlib – Gallery - Ticks and spines



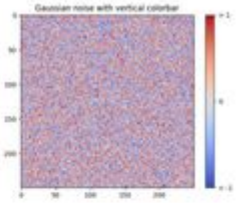
Automatically setting tick labels



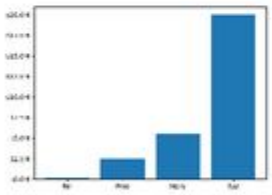
Centered spines with arrows



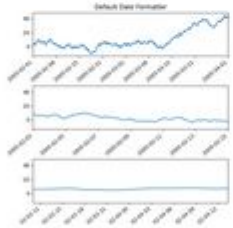
Centering labels between ticks



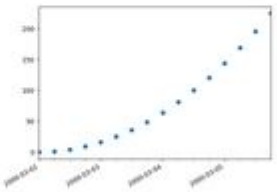
Colorbar Tick Labelling



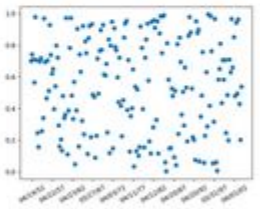
Custom Ticker1



Formatting date ticks using ConciseDateFormatter



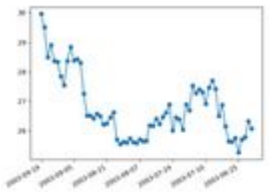
Date Demo Convert



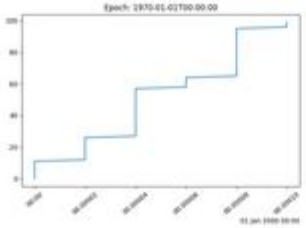
Placing date ticks using recurrence rules

https://matplotlib.org/stable/gallery/index.html

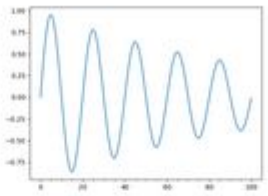
Matplotlib – Gallery - Ticks and spines



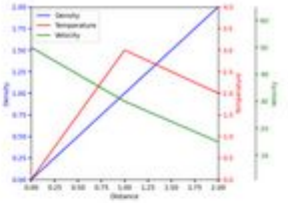
Date Index Formatter



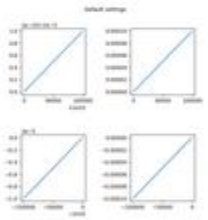
Date Precision and Epochs



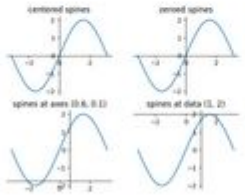
Major and minor ticks



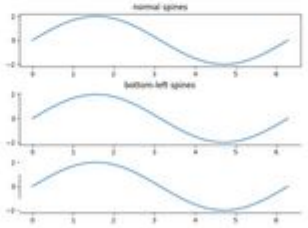
Multiple Yaxis With Spines



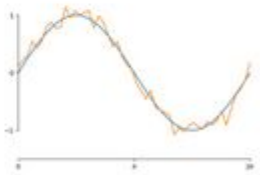
The default tick formatter



Spine Placement



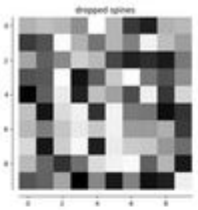
Spines



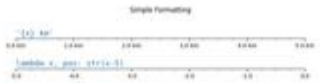
Custom spine bounds

https://matplotlib.org/stable/gallery/index.html

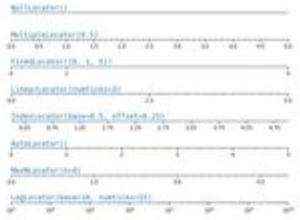
Matplotlib – Gallery - Ticks and spines



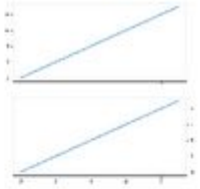
Dropped spines



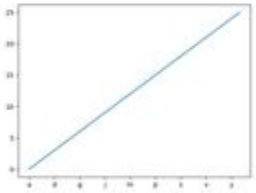
Tick formatters



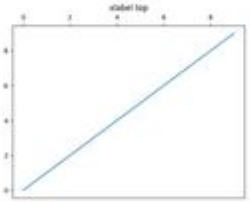
Tick locators



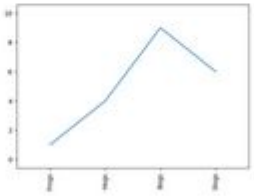
Set default y-axis tick labels on the right



Setting tick labels from a list of values



Set default x-axis tick labels on the top



Rotating custom tick labels

https://matplotlib.org/stable/gallery/index.html

Matplotlib – Gallery – Скачать все примеры

- Примеры – Галерея MatplotLib
  https://matplotlib.org/stable/gallery/index.html

- Скачать все примеры - Python исходники программ:
  `gallery_python.zip`
  https://matplotlib.org/stable/_downloads/63b34a63fc35d506739b9835d7e98958/gallery_python.zip

- Скачать все примеры - Jupyter notebooks:
  `gallery_jupyter.zip`
  https://matplotlib.org/stable/_downloads/a70483fff7b46b03f4d5c358b003188f/gallery_jupyter.zip

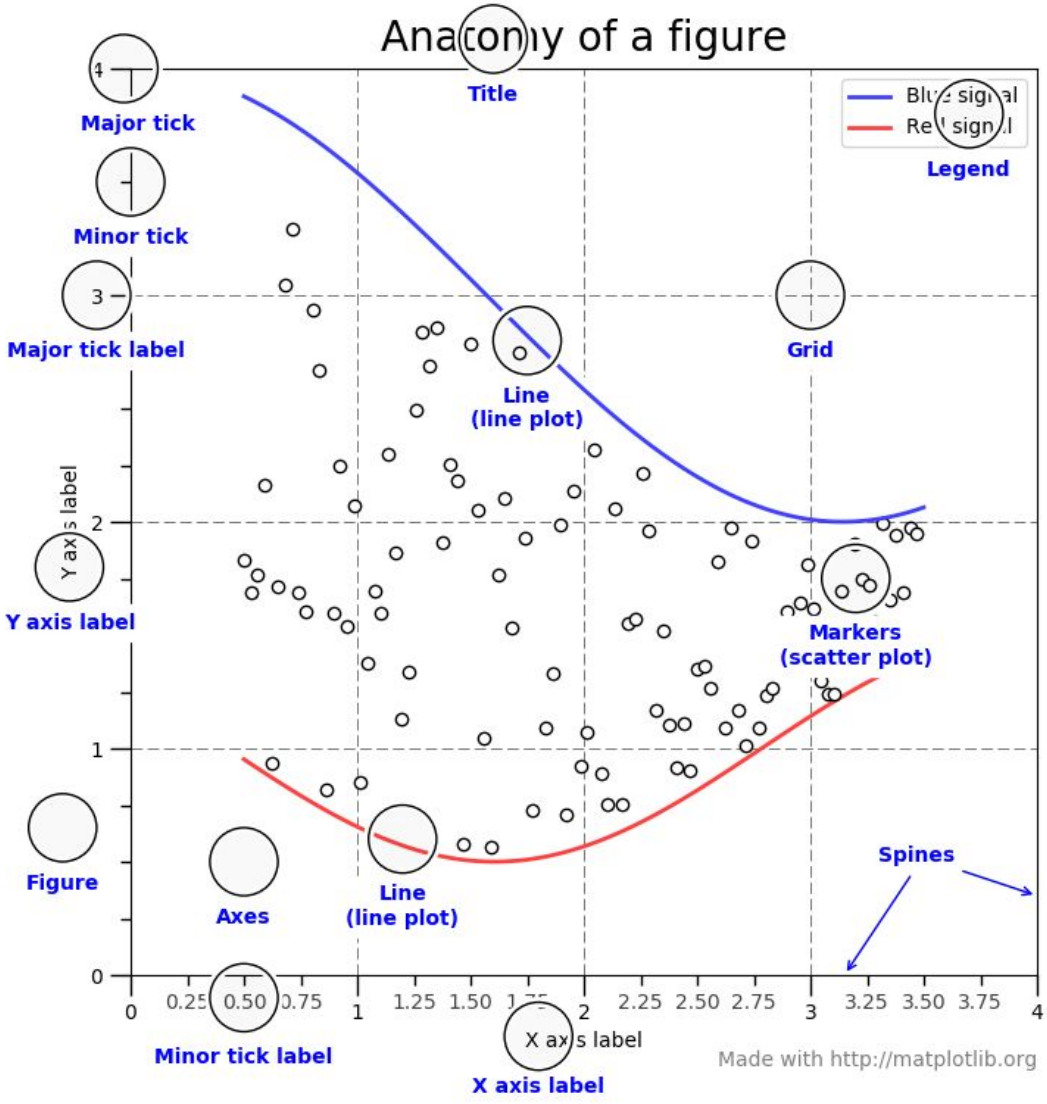
Matplotlib

Создание графиков

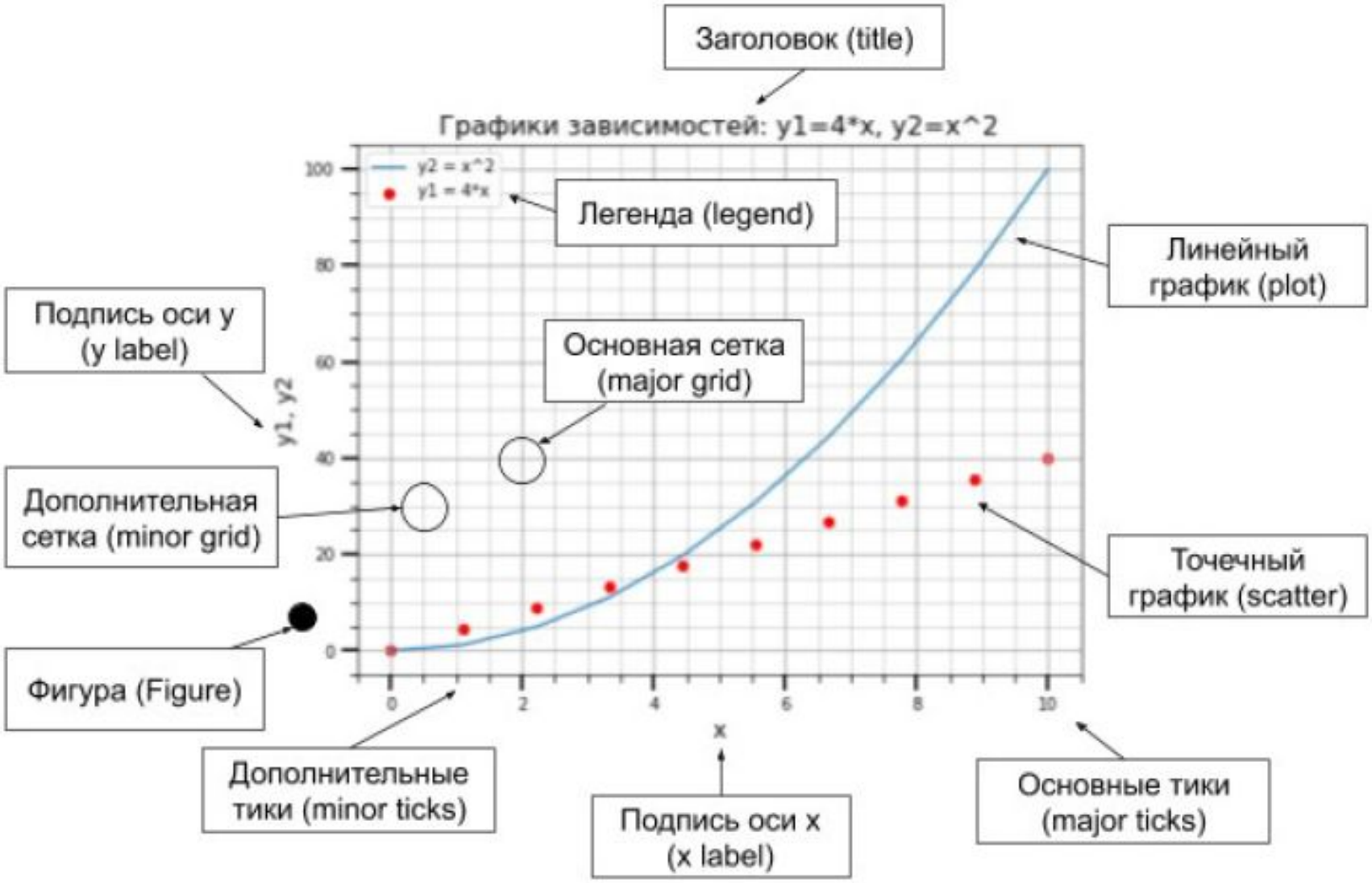
Matplotlib - Подключение

```
# подключение набор команд для работы с графиками из библиотеки matplotlib
import matplotlib.pyplot as plt

# подключение библиотеки numpy для выполнения математических расчетов и работы с матрицами (массивами, списками)
import numpy as np
```

Matplotlib - Основные элементы графика



Matplotlib - Основные элементы графика



Matplotlib Некоторые функции отрисовки

- **`plt.scatter(x, y, params)`** — нарисовать точки с координатами из x по горизонтальной оси и из y по вертикальной оси;
- **`plt.plot(x, y, params)`** — нарисовать график по точкам с координатами из x по горизонтальной оси и из y по вертикальной оси. Точки будут соединятся в том порядке, в котором они указаны в этих массивах;
- **`plt.fill_between(x, y1, y2, params)`** — закрасить пространство между y1 и y2 по координатам из x;
- **`plt.pcolormesh(x1, x1, y, params)`** — закрасить пространство в соответствии с интенсивностью y;
- **`plt.contour(x1, x1, y, lines)`** — нарисовать линии уровня. Затем нужно применить `plt.clabel`.

Matplotlib Вспомогательные функции

- **`plt.figure(figsize=(x, y))`** — создать график размера (x,y);
- **`plt.show()`** — показать график;
- **`plt.subplot(...)`** — добавить подграфик;
- **`plt.xlim(x_min, x_max)`** — установить пределы графика по горизонтальной оси;
- **`plt.ylim(y_min, y_max)`** — установить пределы графика по вертикальной оси;
- **`plt.title(name)`** — установить имя графика;
- **`plt.xlabel(name)`** — установить название горизонтальной оси;
- **`plt.ylabel(name)`** — установить название вертикальной оси;
- **`plt.legend(loc=...)`** — сделать легенду в позиции loc;
- **`plt.grid()`** — добавить сетку на график;
- **`plt.savefig(filename)`** — сохранить график в файл.

plt.plot

- **`plt.plot(x, y, params)`** — нарисовать график по точкам с координатами из **`x`** по горизонтальной оси и из **`y`** по вертикальной оси.
- Точки будут соединятся в том порядке, в котором они указаны в этих массивах;

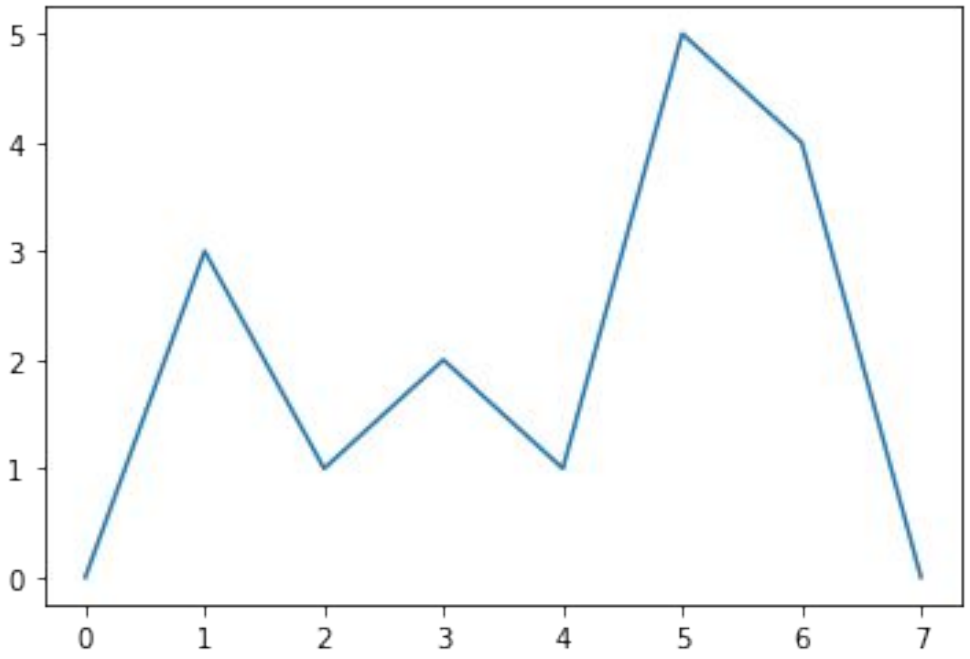
Matplotlib - График линии

- **Метод построения линии очень прост:**
  - есть массив абсцис (x);
  - есть массив ординат (y);
  - элементы с одинаковым индексом в этих массивах - это координаты точек на плоскости;
  - последовательные точки соединяются линией.

- Под массивами, подразумеваются списки, кортежи или массивы NumPy.

Matplotlib - График линии

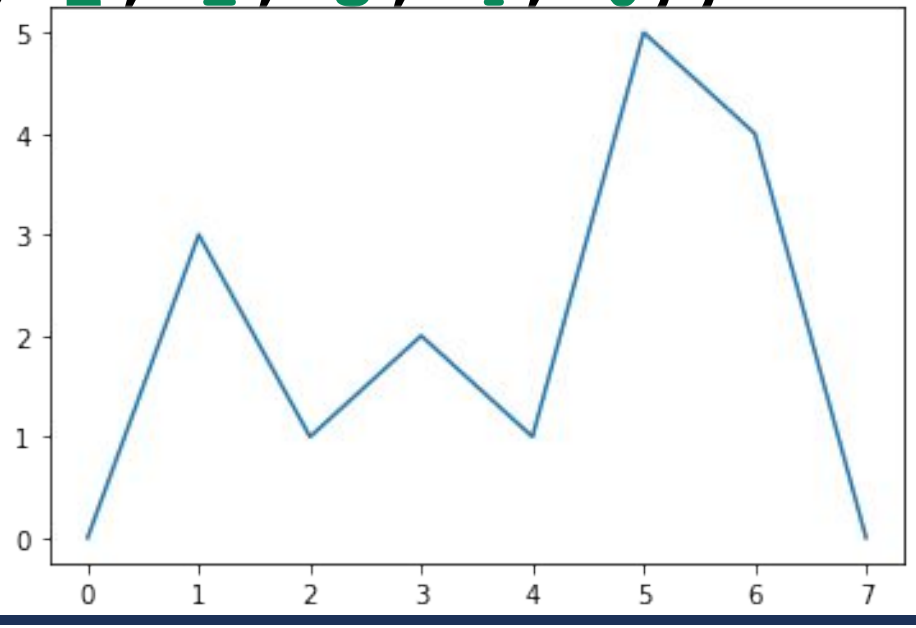
```
import matplotlib.pyplot as plt
plt.plot((0, 1, 2, 3, 4, 5, 6, 7),
         (0, 3, 1, 2, 1, 5, 4, 0))
plt.show()
```



Matplotlib - График линии

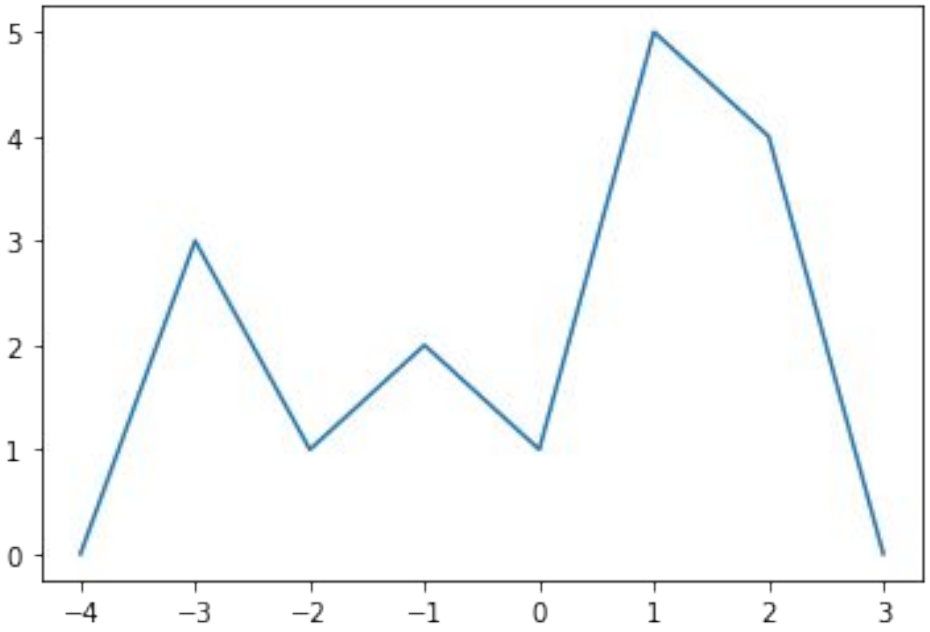
- Метод `plt.plot()`, в простейшем случае, принимает один аргумент - последовательность чисел, которая соответствует оси ординат (y), ось абсцис (x) строится автоматически от 0 до n, где n - это длинна массива ординат.

```
import matplotlib.pyplot as plt
plt.plot((0, 3, 1, 2, 1, 5, 4, 0))
plt.show()
```



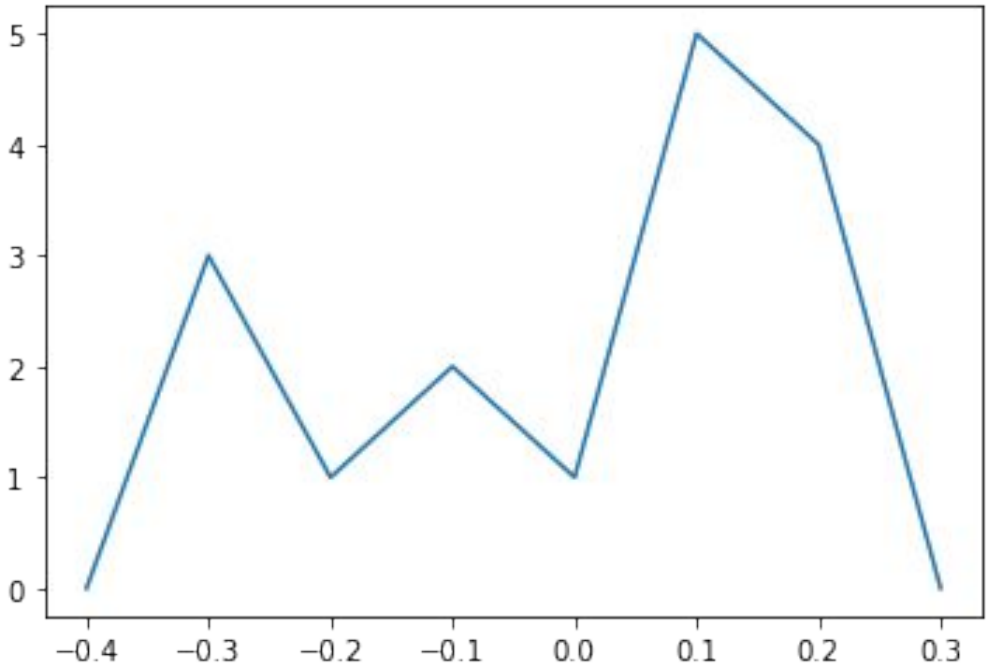
Matplotlib - График линии

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.plot((-4, -3, -2, -1, 0, 1, 2, 3),
         (0, 3, 1, 2, 1, 5, 4, 0))
plt.show()
```



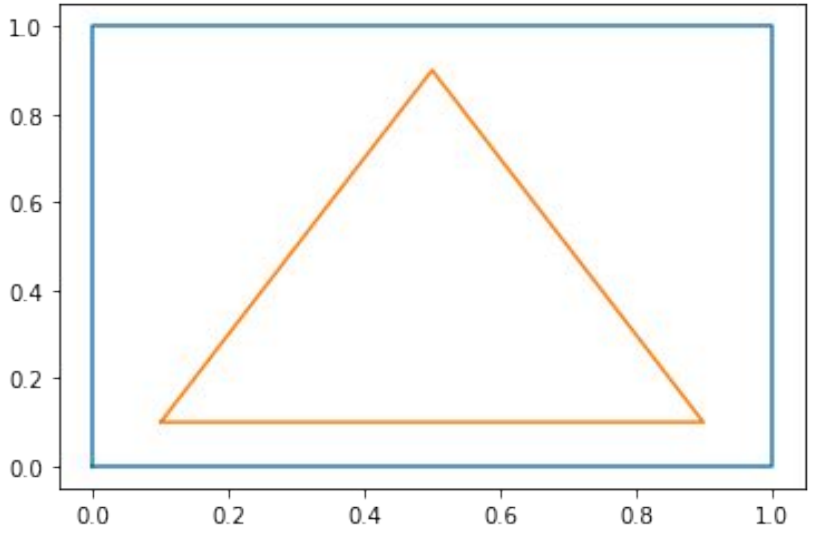
Matplotlib - График линии

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.plot((-0.4, -0.3, -0.2, -0.1, 0., 0.1, 0.2, 0.3),
         (0, 3, 1, 2, 1, 5, 4, 0))
plt.show()
```



Matplotlib – Рисуем фигуры линиями

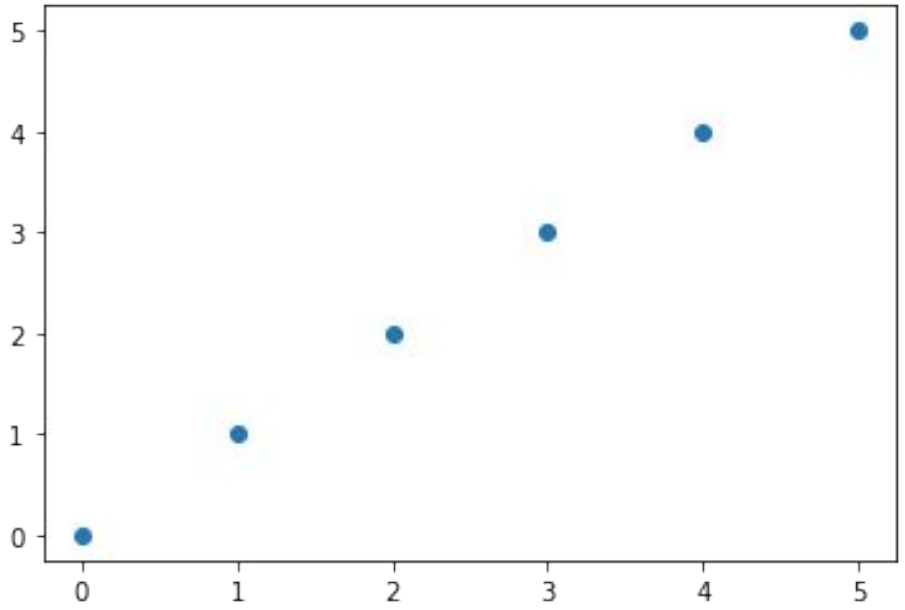
```
%matplotlib inline
import matplotlib.pyplot as plt
plt.plot((0, 0, 1, 1, 0),
         (0, 1, 1, 0, 0))
plt.plot((0.1, 0.5, 0.9, 0.1),
         (0.1, 0.9, 0.1, 0.1))
plt.show()
```



Matplotlib – График множества точек

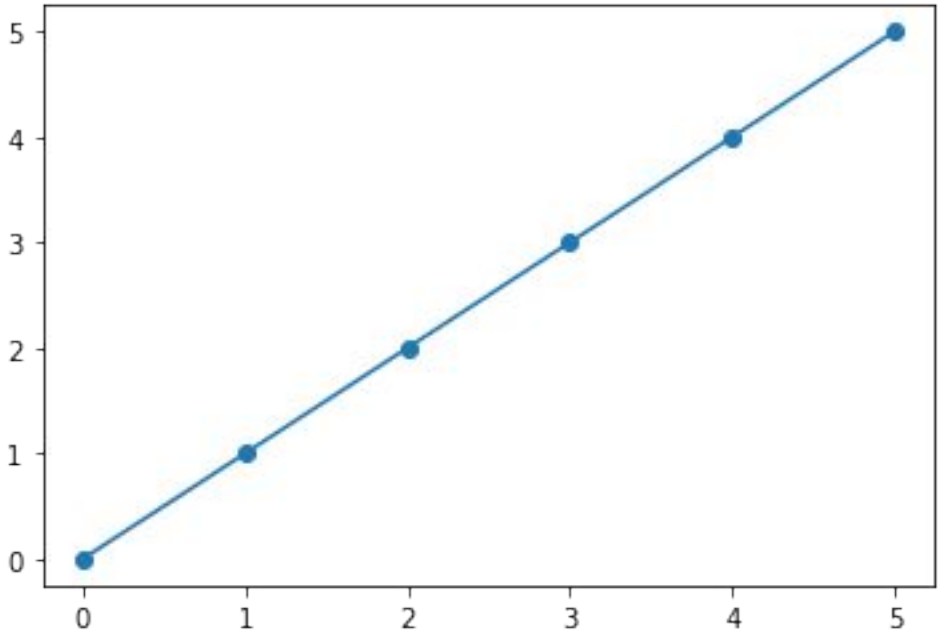
- Единственное отличие графика множества точек от графика линии - точки не соединяются линией.

```
import matplotlib.pyplot as plt
plt.scatter([0, 1, 2, 3, 4 , 5],
            [0, 1, 2, 3, 4 , 5])
plt.show()
```



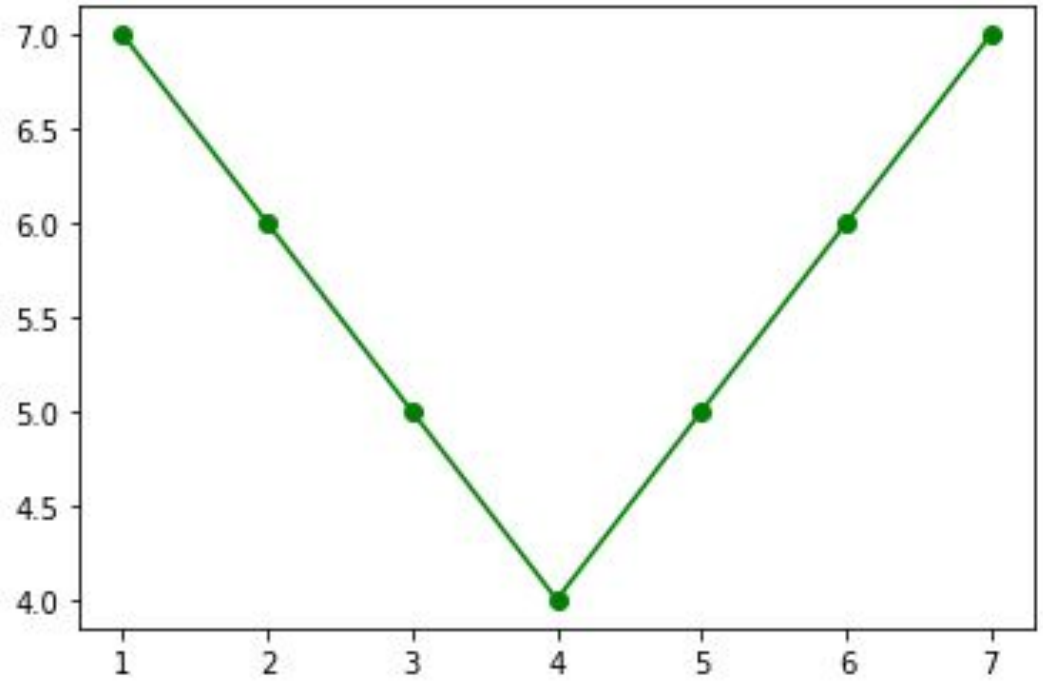
Matplotlib – точки и линии

```
import matplotlib.pyplot as plt
# график точек
plt.scatter([0, 1, 2, 3, 4 , 5],
            [0, 1, 2, 3, 4 , 5])
# график линий
plt.plot((0, 1, 2, 3, 4 , 5),
         (0, 1, 2, 3, 4 , 5))
# отображение графика
plt.show()
```



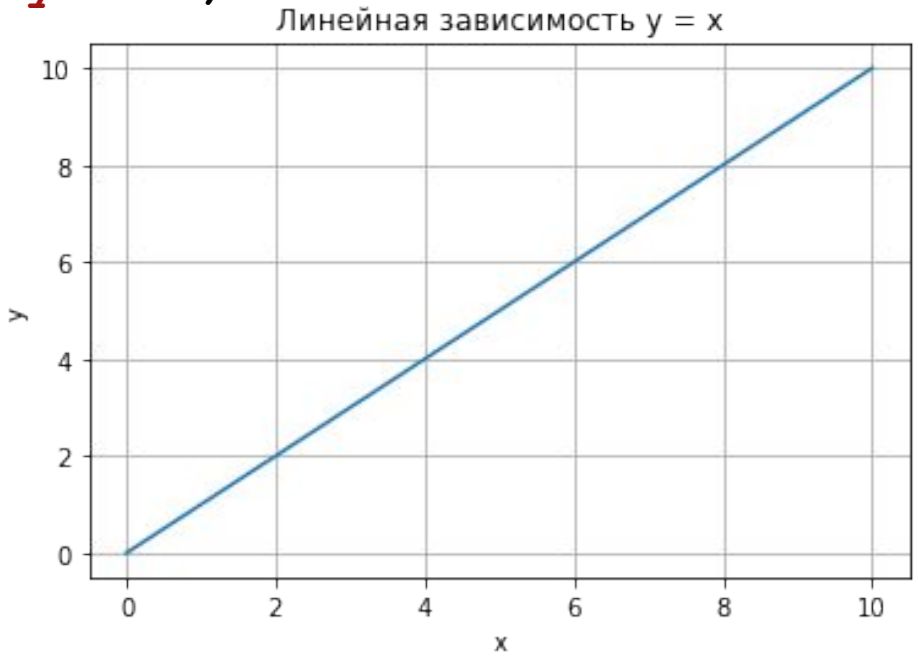
Matplotlib – График с маркировкой

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5, 6, 7]
y = [7, 6, 5, 4, 5, 6, 7]
plt.plot(x, y, marker='o', c='g')
```



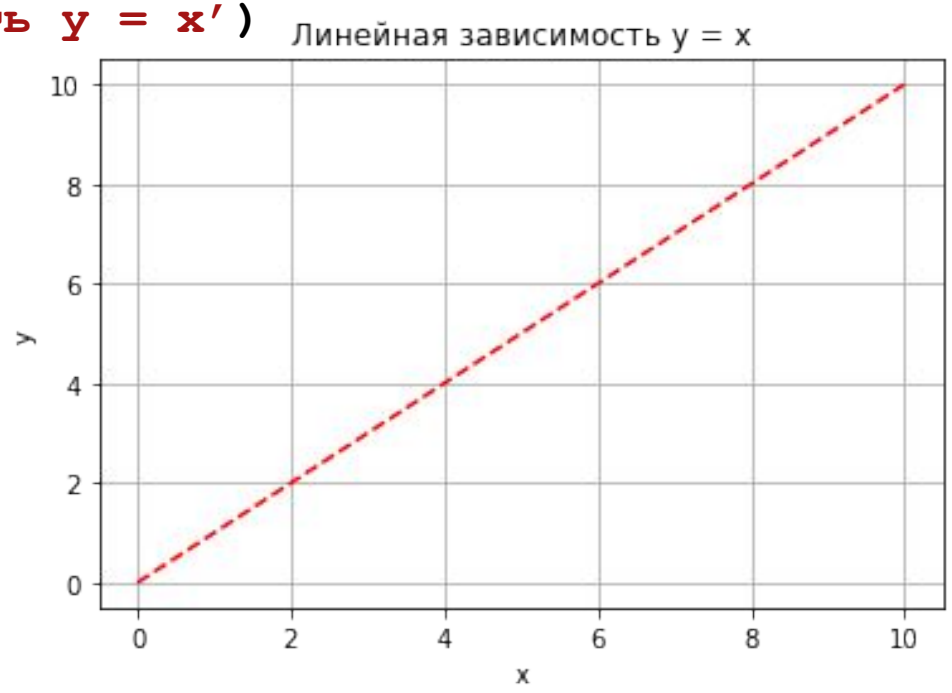
Matplotlib – Линейный график

```
import matplotlib.pyplot as plt
import numpy as np
# Независимая (x) и зависимая (y) переменные
x = np.linspace(0, 10, 50)
y = x
# Построение графика
# заголовок
plt.title('Линейная зависимость y = x')
# ось абсцисс
plt.xlabel('x')
# ось ординат
plt.ylabel('y')
# включение отображения сетки
plt.grid()
# построение графика
plt.plot(x, y)
```



Matplotlib – Линейный график

```
import matplotlib.pyplot as plt
import numpy as np
# Независимая (x) и зависимая (y) переменные
x = np.linspace(0, 10, 50)
y = x
# Построение графика
# заголовок
plt.title('Линейная зависимость y = x')
# ось абсцисс
plt.xlabel('x')
# ось ординат
plt.ylabel('y')
# включение отображения сетки
plt.grid()
# построение графика
plt.plot(x, y, 'r--')
```



Цвет и стиль графиков

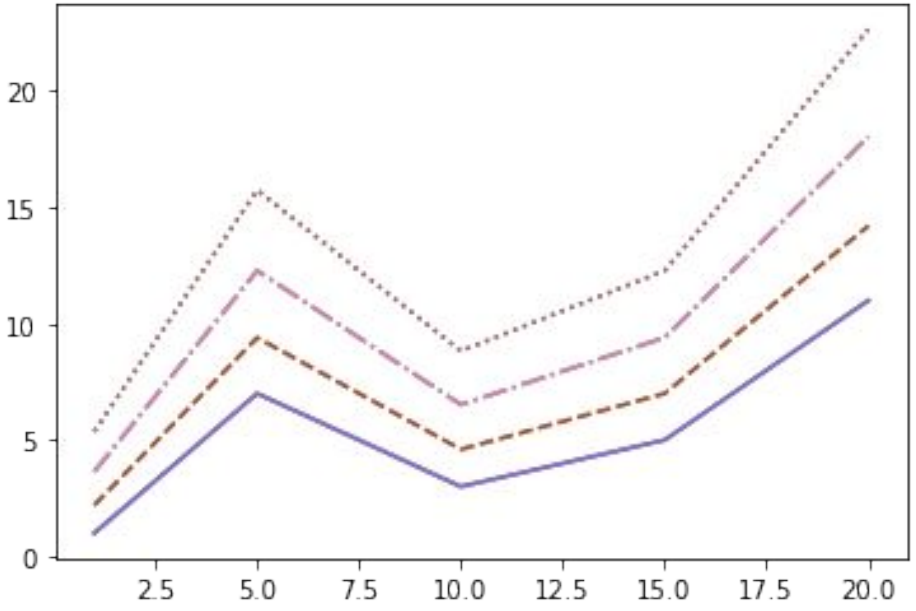
Стили линии линейного графика

Значение параметра	Описание
'-' или 'solid'	Непрерывная линия.
'--' или 'dashed'	Штриховая линия.
'-.' или 'dashdot'	Штрихпунктирная линия.
':' или 'dotted'	Пунктирная линия.
'None' или ' ' или ''	Не отображать линию.

- **Цвет линии графика** задаётся через параметр color (или c, если использовать сокращённый вариант). Значение может быть представлено в одном из следующих форматов:
  - RGB или RGBA: кортеж значений с плавающей точкой в диапазоне [0, 1] (пример: (0.1, 0.2, 0.3);
  - RGB или RGBA: значение в hex формате (пример: '#0a0a0a');
  - строковое представление числа с плавающей точкой в диапазоне [0, 1] (определяет цвет в шкале серого) (пример: '0.7');
  - символ из набора: {'b', 'g', 'r', 'c', 'm', 'y', 'k', 'w'};
  - имя цвета из палитры X11/CSS4;
  - цвет из палитры xkcd (https://xkcd.com/color/rgb/), должен начинаться с префикса 'xkcd:';
  - цвет из набора Tableau Color (палитра T10), должен начинаться с префикса 'tab:'.
- Если цвет задаётся с помощью символа из набора {'b', 'g', 'r', 'c', 'm', 'y', 'k', 'w'}, то он может быть совмещён со стилем линии в рамках параметра fmt функции plot(). Например: штриховая красная линия будет задаваться так: '--r', а штрихпунктирная зелёная так '-.g':

Matplotlib - Стили линии линейного графика

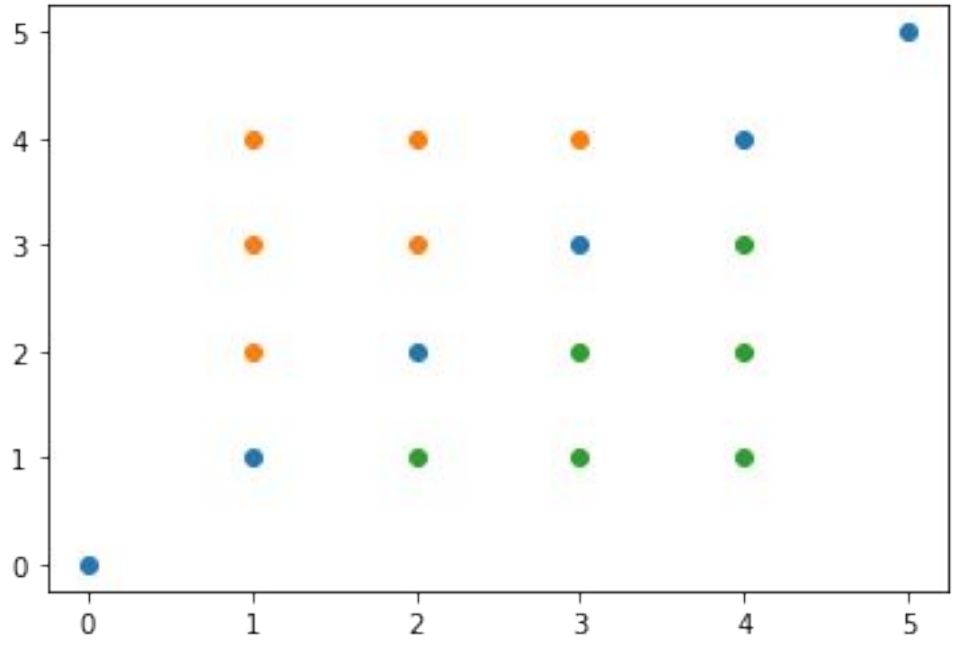
```
import matplotlib.pyplot as plt
import numpy as np
x = [1, 5, 10, 15, 20]
y1 = [1, 7, 3, 5, 11]
y2 = [i*1.2 + 1 for i in y1]
y3 = [i*1.2 + 1 for i in y2]
y4 = [i*1.2 + 1 for i in y3]
plt.plot(x, y1, '-', x, y2, '--', x, y3, '-.', x, y4, ':')
plt.plot(x, y1, '-')
plt.plot(x, y2, '--')
plt.plot(x, y3, '-.')
plt.plot(x, y4, ':')
```



Matplotlib – График множества точек

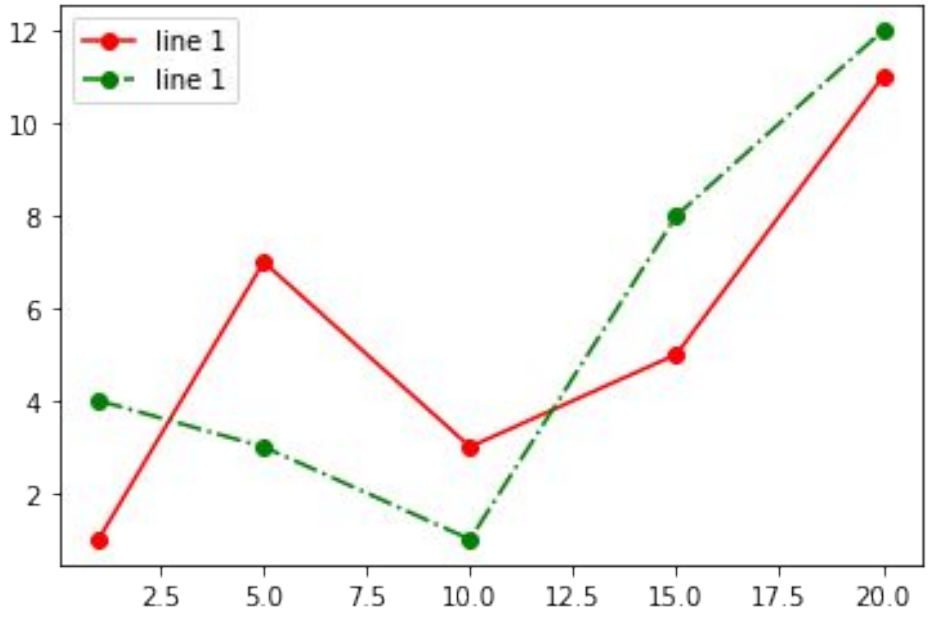
```
%matplotlib inline
import matplotlib.pyplot as plt
plt.scatter([0, 1, 2, 3, 4 , 5], [0, 1, 2, 3, 4 , 5])
plt.scatter([1, 2, 3, 1, 2 , 1], [2, 3, 4, 3, 4 , 4])
plt.scatter([2, 3, 4, 3, 4 , 4], [1, 2, 3, 1, 2 , 1])
plt.show()
```

Если у вас несколько множеств, то все их так же можно построить на одном графике:



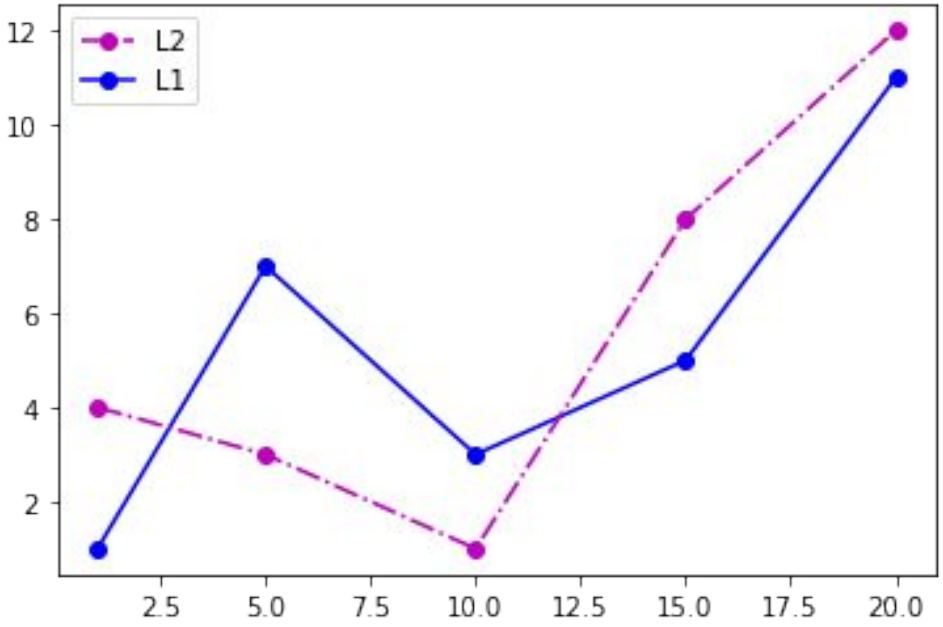
Matplotlib – Легенда на графике

```
import matplotlib.pyplot as plt
x = [1, 5, 10, 15, 20]
y1 = [1, 7, 3, 5, 11]
y2 = [4, 3, 1, 8, 12]
plt.plot(x, y1, 'o-r', label='line 1')
plt.plot(x, y2, 'o-.g', label='line 1')
plt.legend()
```



Matplotlib – Легенда на графике

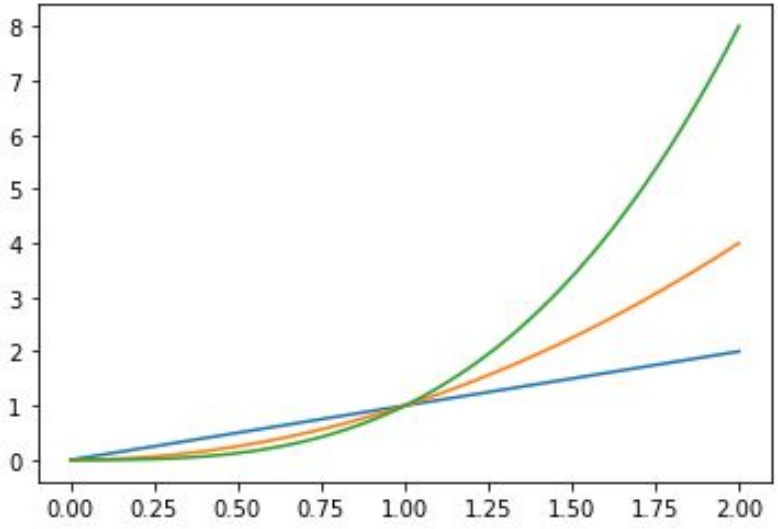
```
import matplotlib.pyplot as plt
x = [1, 5, 10, 15, 20]
y1 = [1, 7, 3, 5, 11]
y2 = [4, 3, 1, 8, 12]
line1, = plt.plot(x, y1, 'o-b')
line2, = plt.plot(x, y2, 'o-.m')
plt.legend((line2, line1), ['L2', 'L1'])
```



Matplotlib - Несколько графиков на одном поле

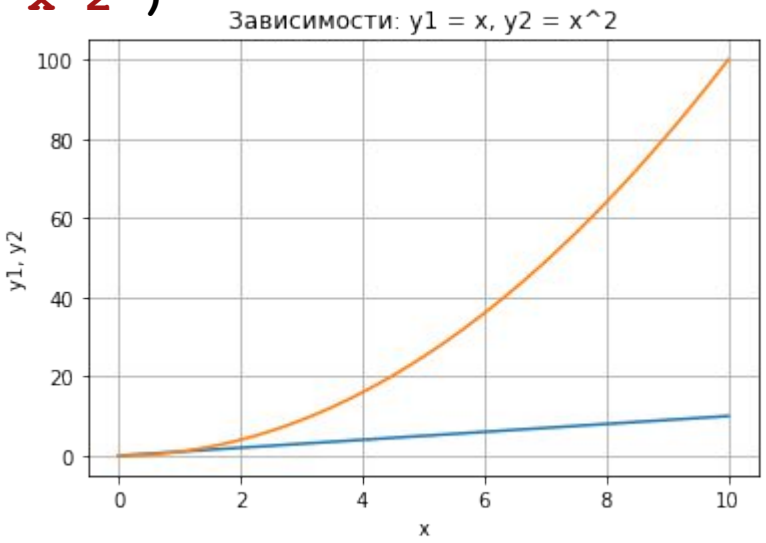
```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 2, 100)

plt.figure()
plt.plot(x, x, x, x**2, x, x**3)
plt.show()
```

Несколько кривых на одном графике. Каждая задаётся парой массивов — x и y координаты. По умолчанию, им присваиваются цвета из некоторой последовательности цветов; разумеется, их можно изменить.



Matplotlib - Несколько графиков на одном поле

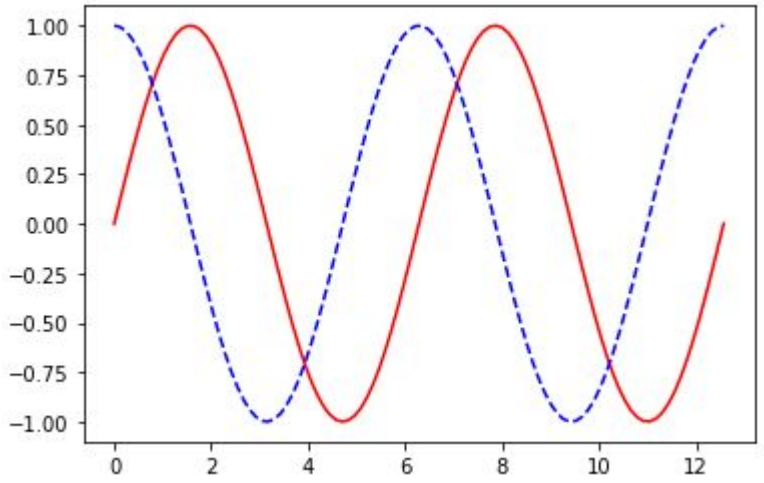
```
import matplotlib.pyplot as plt
import numpy as np
# Линейная зависимость
x = np.linspace(0, 10, 50)
y1 = x
# Квадратичная зависимость
y2 = [i**2 for i in x]
# Построение графика
# заголовок
plt.title('Зависимости: y1 = x, y2 = x^2')
# ось абсцисс
plt.xlabel('x')
# ось ординат
plt.ylabel('y1, y2')
# включение отображения сетки
plt.grid()
# построение графика
plt.plot(x, y1, x, y2)
```



Matplotlib - sin(x), cos(x)

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

x = np.linspace(0, 4 * np.pi, 100)
plt.figure()
plt.plot(x, np.sin(x), 'r-')
plt.plot(x, np.cos(x), 'b--')
plt.show()
```

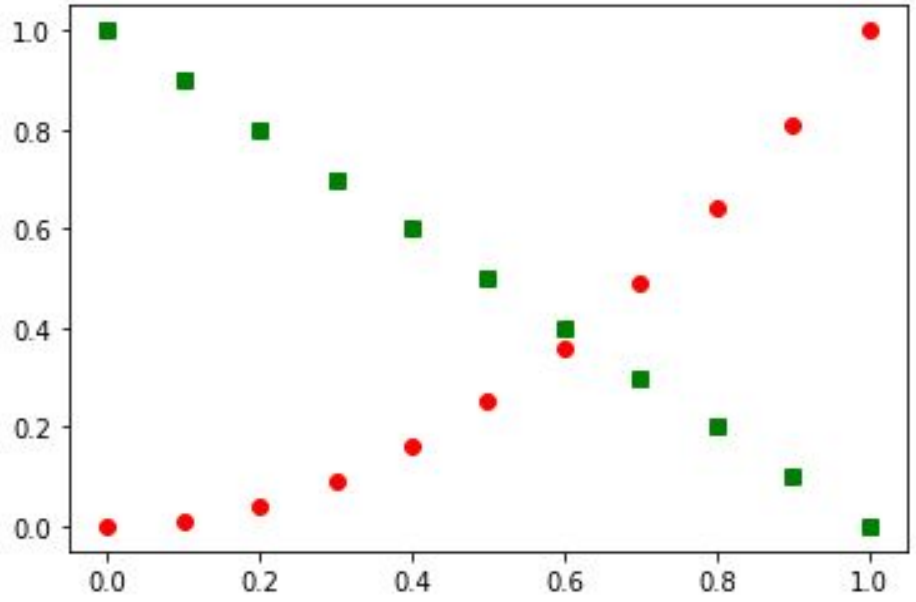
Для простой регулировки цветов и типов линий после пары x и y координат вставляется форматная строка. Первая буква определяет цвет ('r' — красный, 'b' — синий и т.д.), дальше задаётся тип линии ('-' — сплошная, '--' — пунктирная, '-.' — штрих-пунктирная и т.д.).



Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 1, 11)

plt.figure()
plt.plot(x, x ** 2, 'ro')
plt.plot(x, 1 - x, 'gs')
plt.show()
```

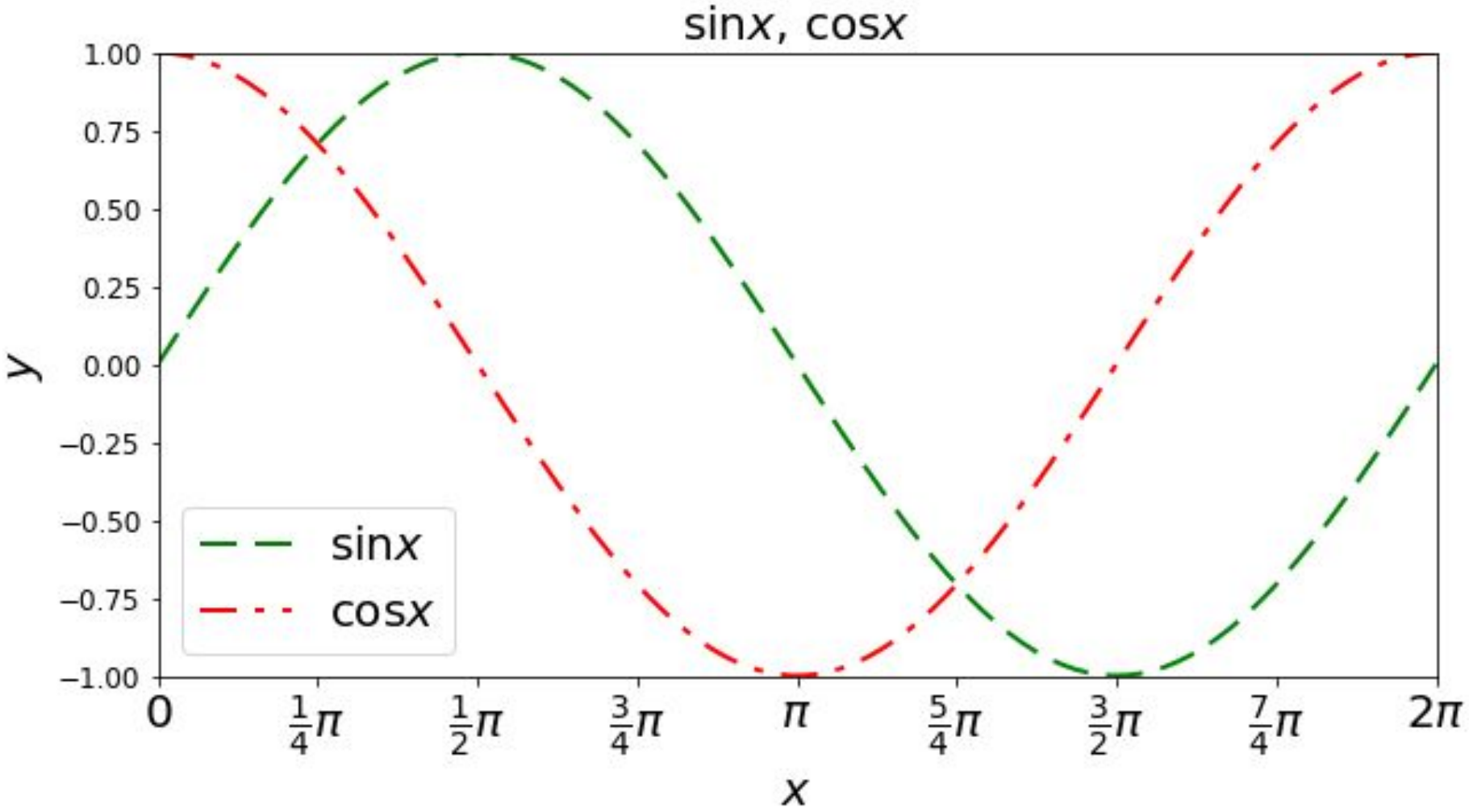
Если в качестве "типа линии" указано **'o'**, то это означает рисовать точки кружочками и не соединять их линиями; аналогично, **'s'** означает квадратики. Конечно, такие графики имеют смысл только тогда, когда точек не очень много.



Matplotlib - sin(x), cos(x)

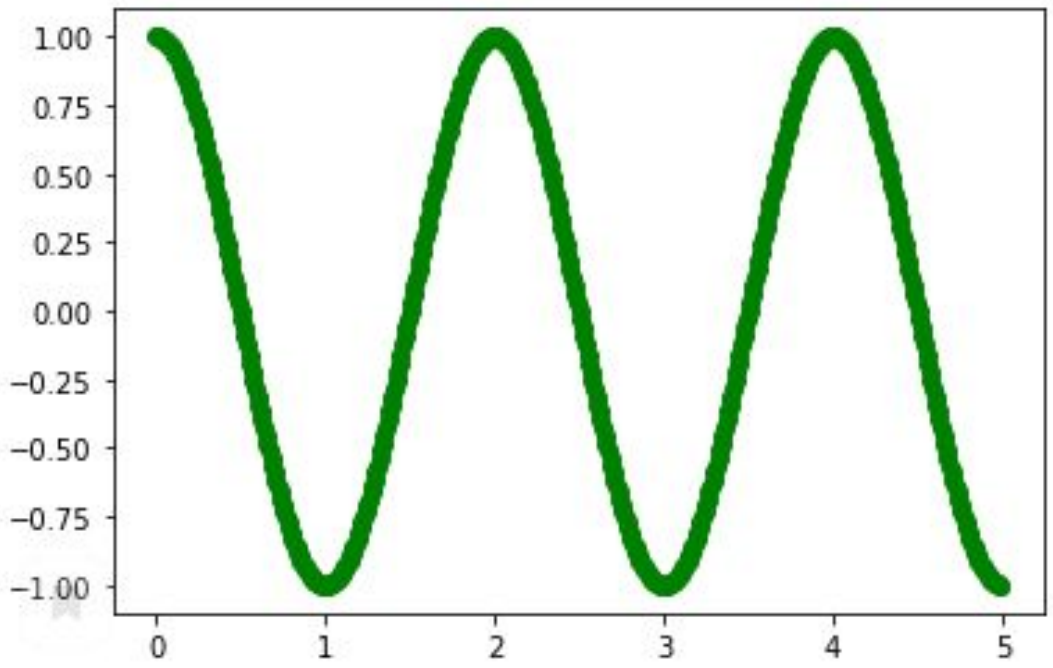
```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 2 * np.pi, 100)

plt.figure(figsize=(10, 5))
plt.plot(x, np.sin(x), linewidth=2, color='g', dashes=[8, 4], label=r'$\sin x$')
plt.plot(x, np.cos(x), linewidth=2, color='r', dashes=[8, 4, 2, 4], label=r'$\cos x
$')
plt.axis([0, 2 * np.pi, -1, 1])
plt.xticks(np.linspace(0, 2 * np.pi, 9),  # Где сделать отметки
           ('0',r'$\frac{1}{4}\pi$',r'$\frac{1}{2}\pi$',  # Как подписать
            r'$\frac{3}{4}\pi$',r'$\pi$',r'$\frac{5}{4}\pi$',
            r'$\frac{3}{2}\pi$',r'$\frac{7}{4}\pi$',r'$2\pi$'),
           fontsize=20)
plt.yticks(fontsize=12)
plt.xlabel(r'$x$', fontsize=20)
plt.ylabel(r'$y$', fontsize=20)
plt.title(r'$\sin x$, $\cos x$', fontsize=20)
plt.legend(fontsize=20, loc=0)
plt.show()
```

Matplotlib - sin(x), cos(x)



Matplotlib - График с большим количеством маркеров

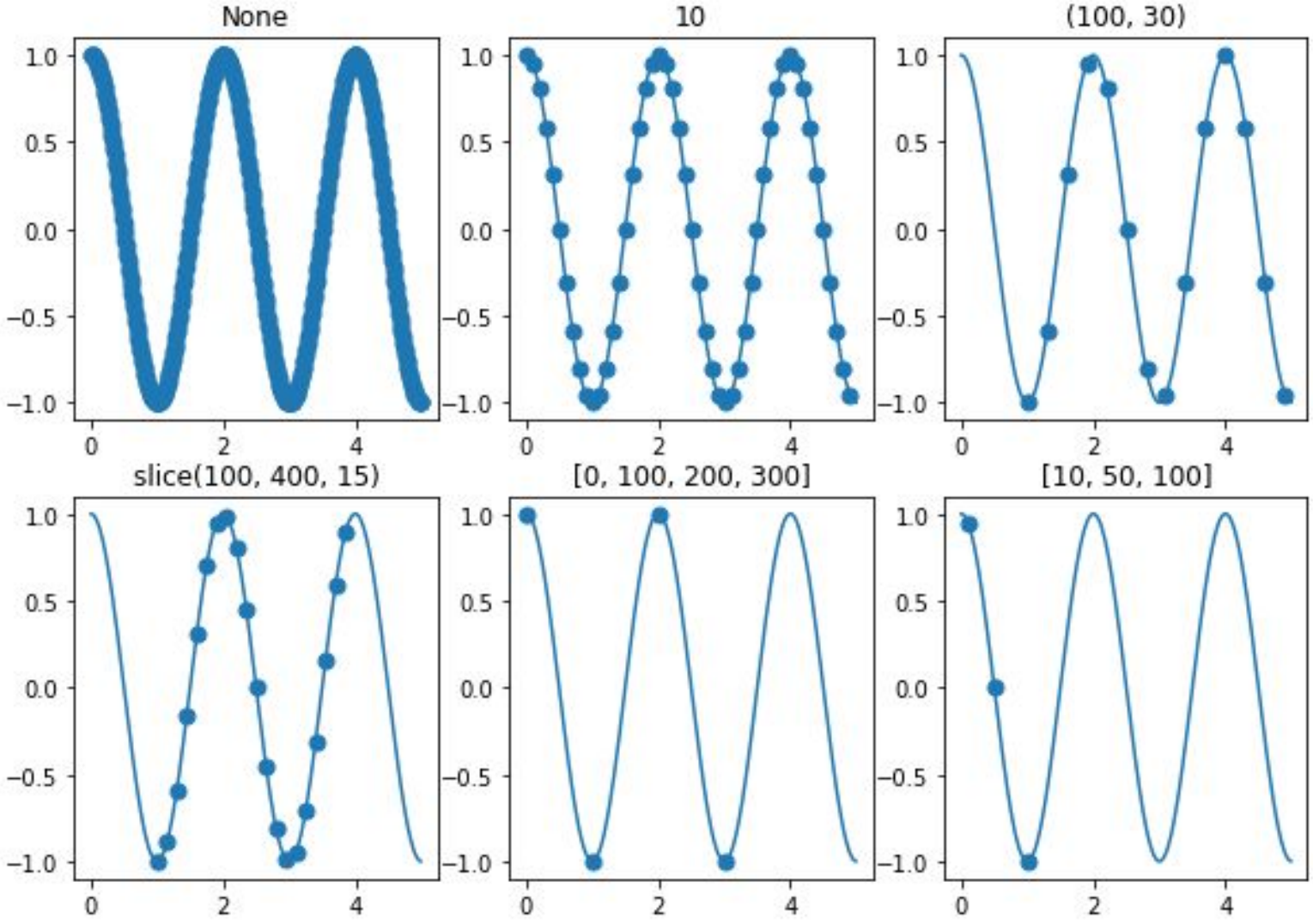
```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0.0, 5, 0.01)
y = np.cos(x*np.pi)
plt.plot(x, y, marker='o', c='g')
```



Различные варианты маркировки

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

x = np.arange(0.0, 5, 0.01)
y = np.cos(x*np.pi)

m_ev_case = [None, 10, (100, 30), slice(100,400,15),
[0, 100, 200, 300],[10, 50, 100]]

fig, ax = plt.subplots(2, 3, figsize=(10, 7))

axs = [ax[i, j] for i in range(2) for j in range(3)]

for i, case in enumerate(m_ev_case):
  axs[i].set_title(str(case))
  axs[i].plot(x, y, 'o', ls='-', ms=7, markevery=case)
```

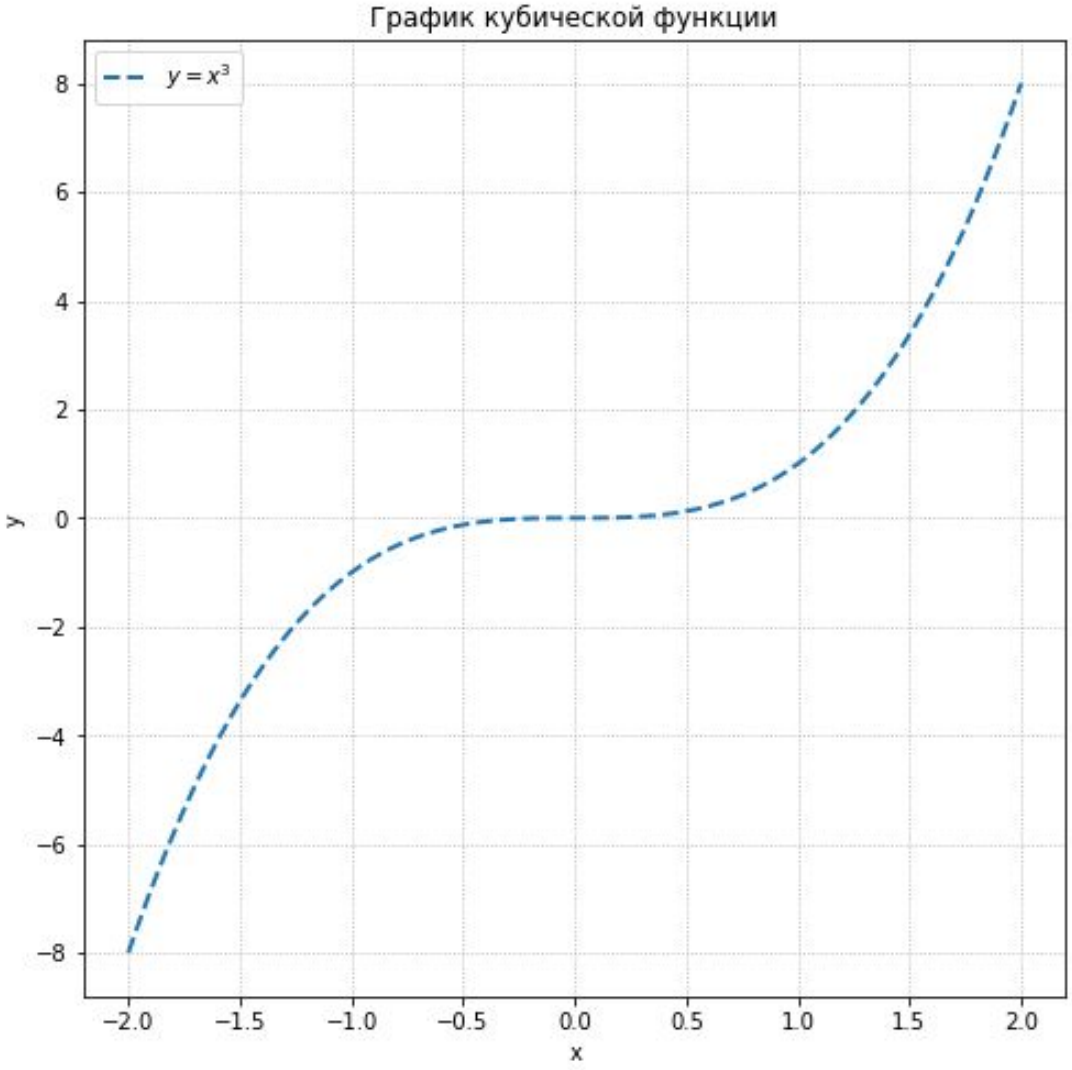
Различные варианты маркировки



Пунктирный график функции $y=x^3$

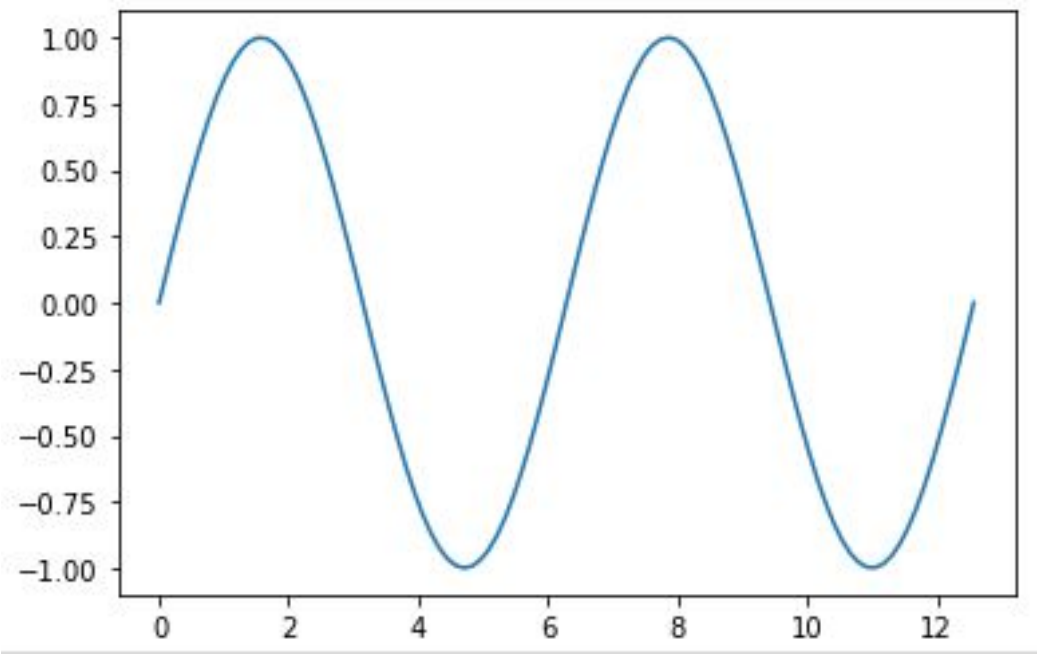
```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(-2, 2, 100)

plt.figure(figsize=(8, 8))
plt.plot(x, x**3, linestyle='--', lw=2,
label='$y=x^3$')
plt.xlabel('x'), plt.ylabel('y')
plt.legend()
plt.title('График кубической функции')
plt.grid(ls=':')
plt.show()
```

Пунктирный график функции $y=x^3$



sin(x)

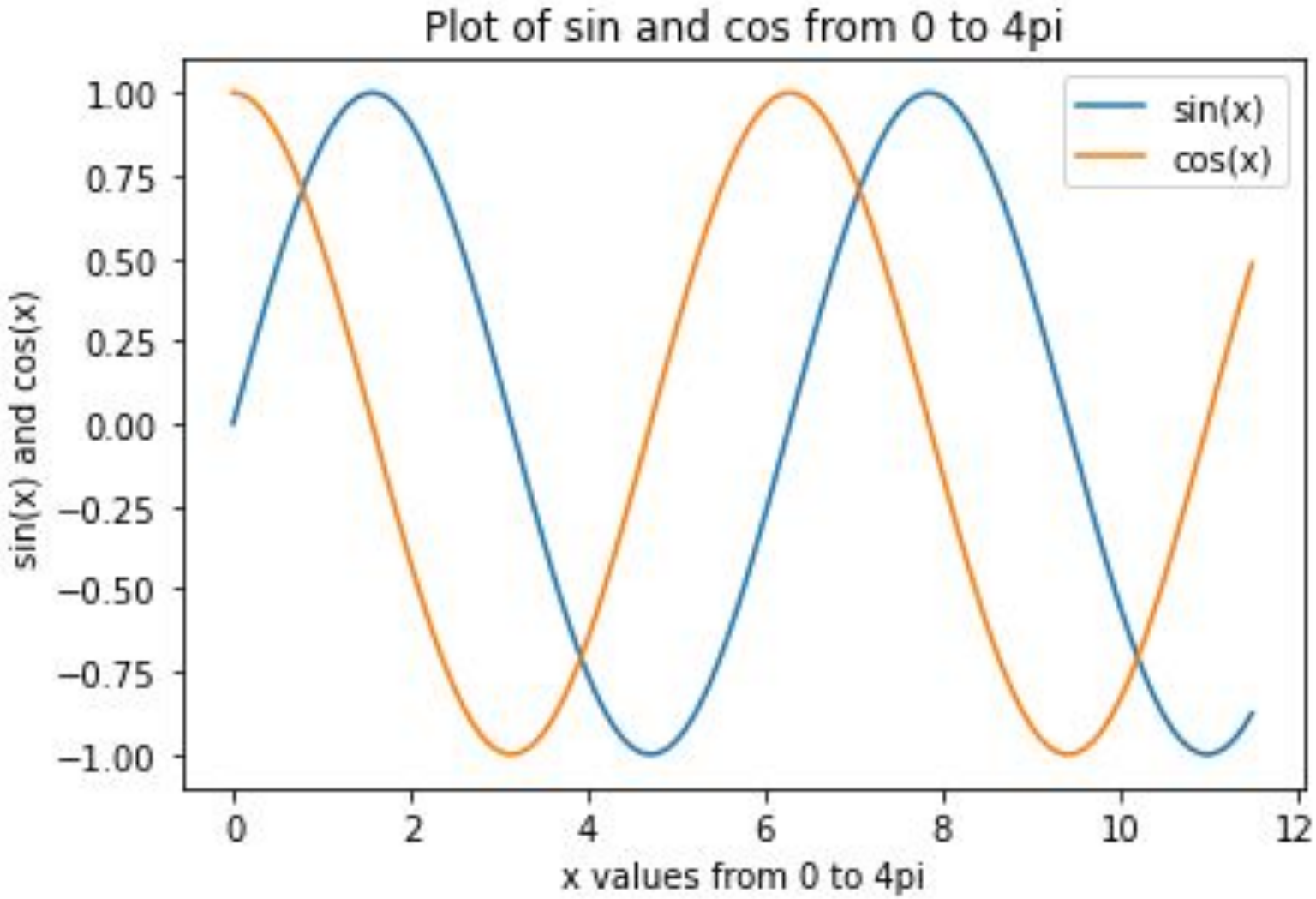
```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0,4*np.pi,100)
plt.plot(x,np.sin(x))
```



sin(x), cos(x)

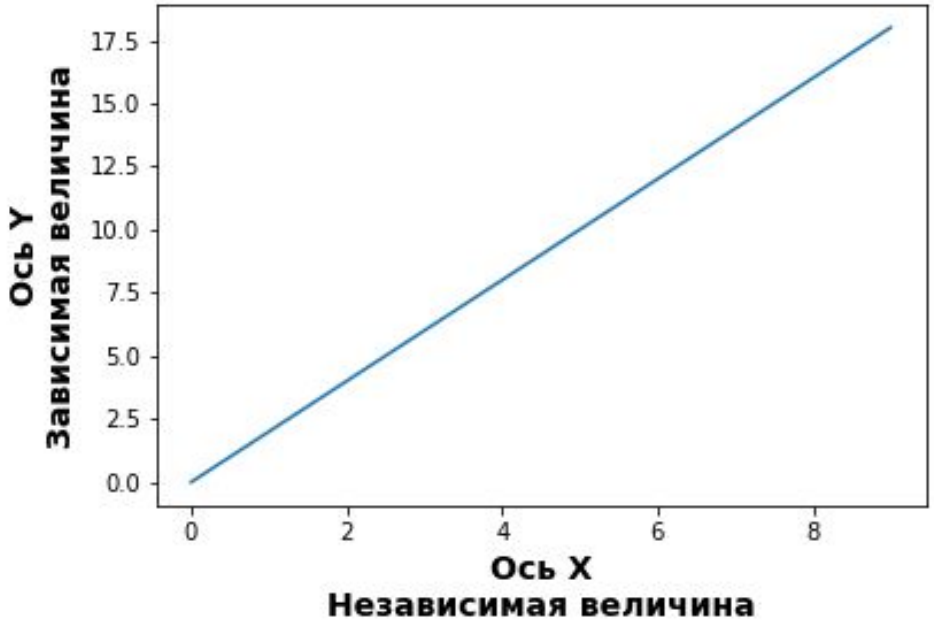
```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0,4*np.pi-1,0.1)# start,stop,step
y = np.sin(x)
z = np.cos(x)
plt.plot(x,y,x,z)
# string must be enclosed with quotes '  '
plt.xlabel('x values from 0 to 4pi')
plt.ylabel('sin(x) and cos(x)')
plt.title('Plot of sin and cos from 0 to 4pi')
plt.legend(['sin(x)', 'cos(x)'])# legend
entries as seperate strings in a list
plt.show()
```

sin(x), cos(x)


Plot of sin and cos from 0 to 4pi
sin(x)
cos(x)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
sin(x) and cos(x)
0
2
4
6
8
10
12
x values from 0 to 4pi

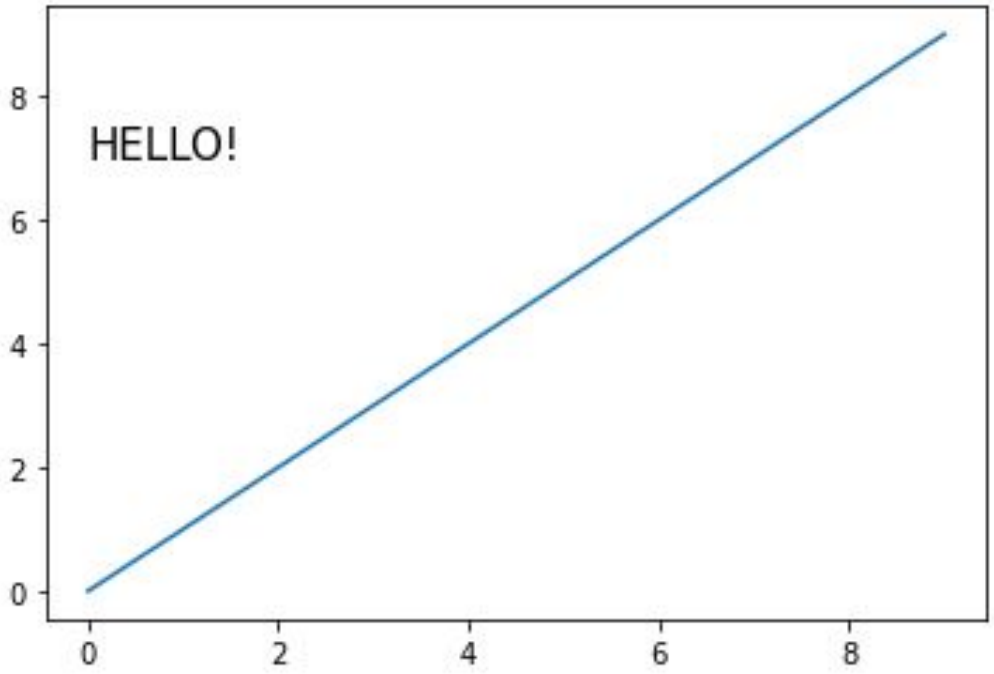
Matplotlib – Подписи осей графика

```
import matplotlib.pyplot as plt
x = [i for i in range(10)]
y = [i*2 for i in range(10)]

plt.plot(x, y)
plt.xlabel('Ось X\nНезависимая величина', fontsize=14,
fontweight='bold')
plt.ylabel('Ось Y\nЗависимая величина', fontsize=14,
fontweight='bold')
```



Matplotlib – Текстовый блок

```
import matplotlib.pyplot as plt
x = [i for i in range(10)]
y = [i*2 for i in range(10)]

plt.text(0, 7, 'HELLO!', fontsize=15)
plt.plot(range(0,10), range(0,10))
```



Столбчатые и круговые диаграммы

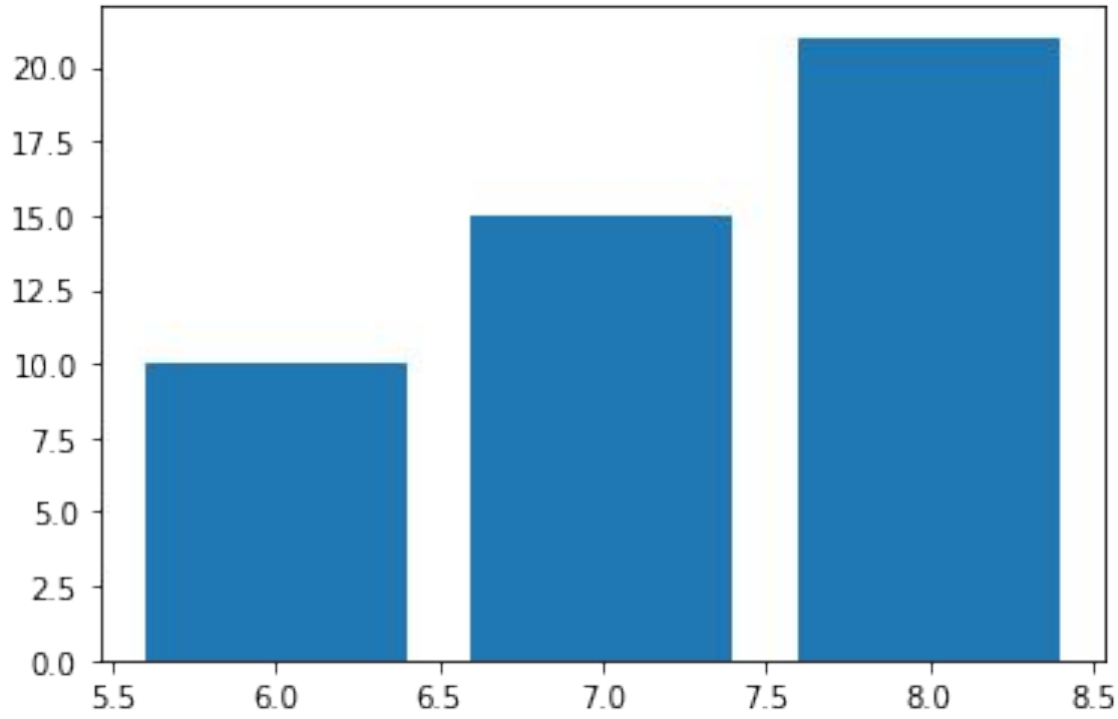
Столбчатые диаграммы

- Для визуализации категориальных данных хорошо подходят **столбчатые диаграммы**.

- Для их построения используются функции:
  - **`bar()`** — вертикальная столбчатая диаграмма;
  - **`barh()`** — горизонтальная столбчатая диаграмма.

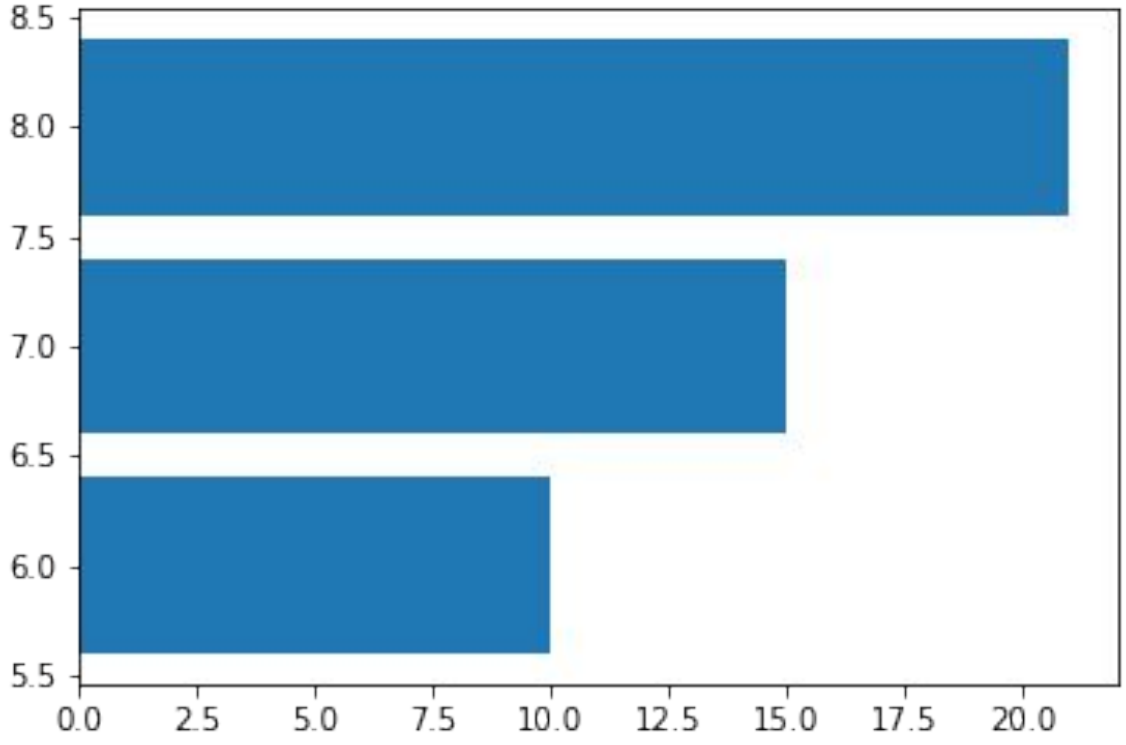
Гистограммы

```
import matplotlib.pyplot as plt
plt.bar([6, 7, 8],
        [10, 15, 21])
plt.show()
```



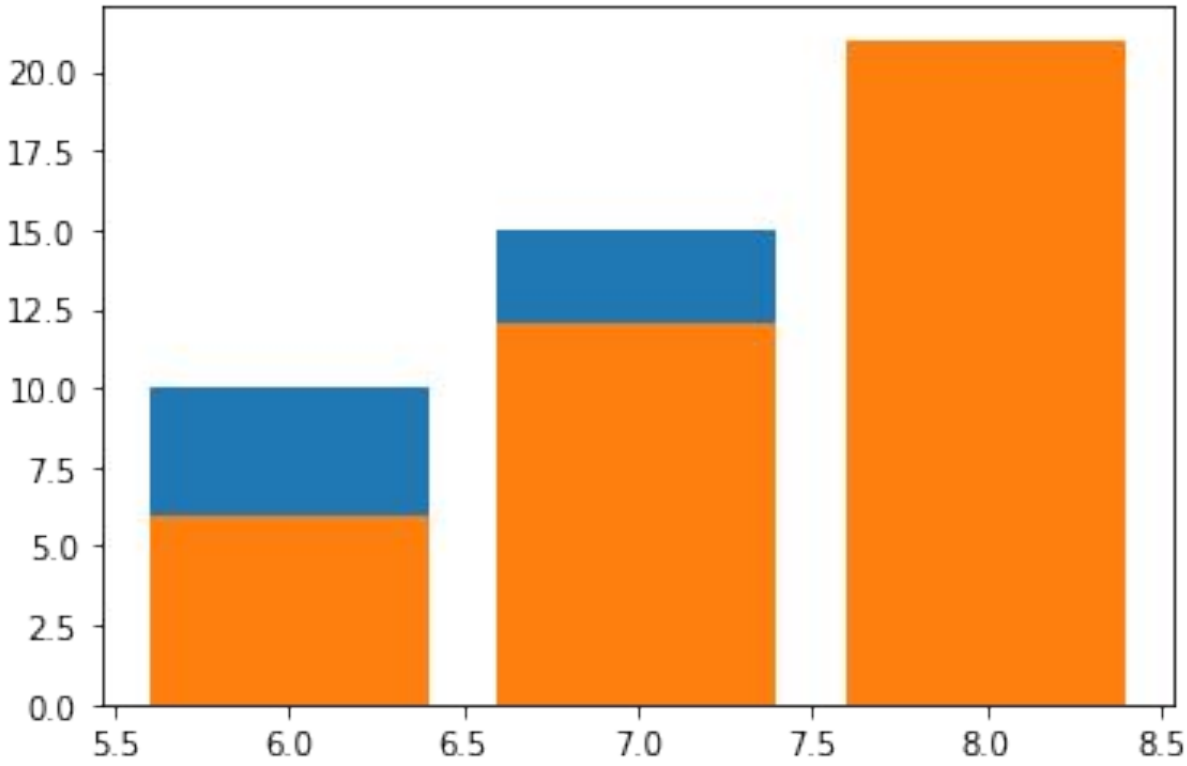
Гистограммы

```
import matplotlib.pyplot as plt
plt.barh([6, 7, 8],
         [10, 15, 21])
plt.show()
```



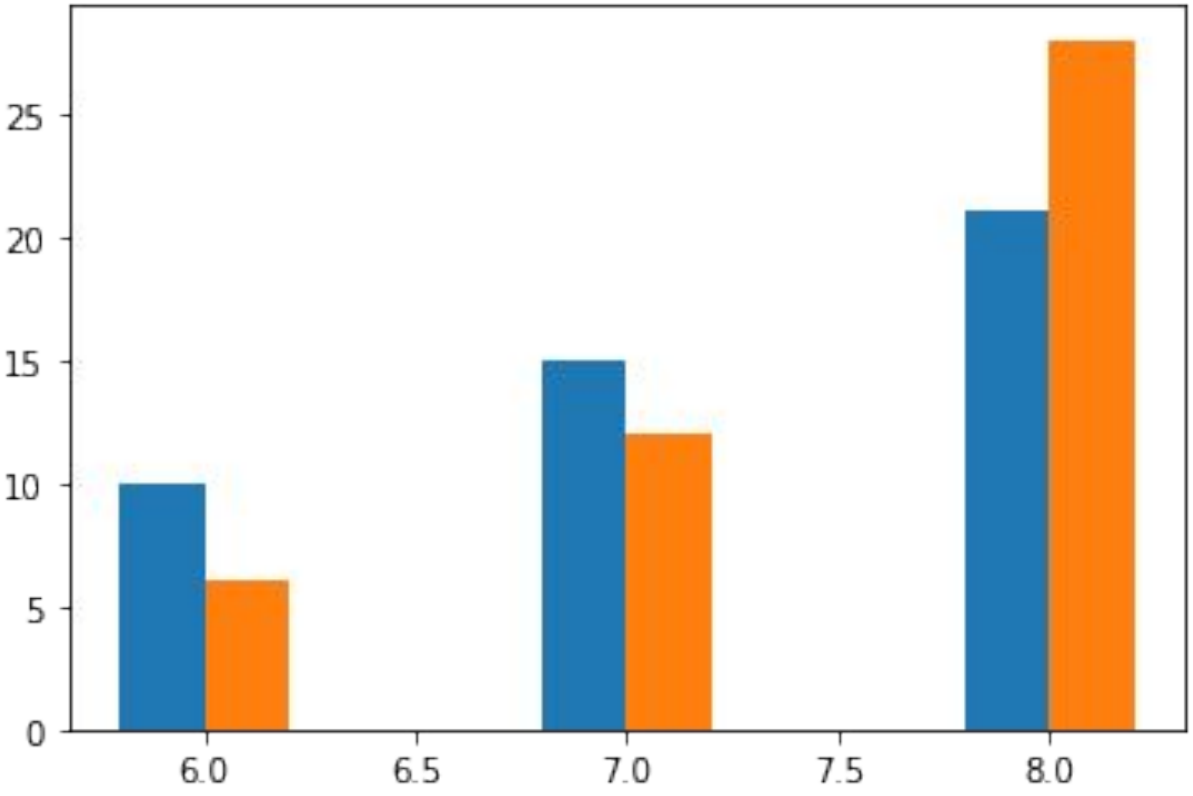
Гистограммы с несколькими наборами данных

```
import matplotlib.pyplot as plt
plt.bar([6, 7, 8], [10, 15, 21])
plt.bar([6, 7, 8], [6, 12, 21])
plt.show()
```



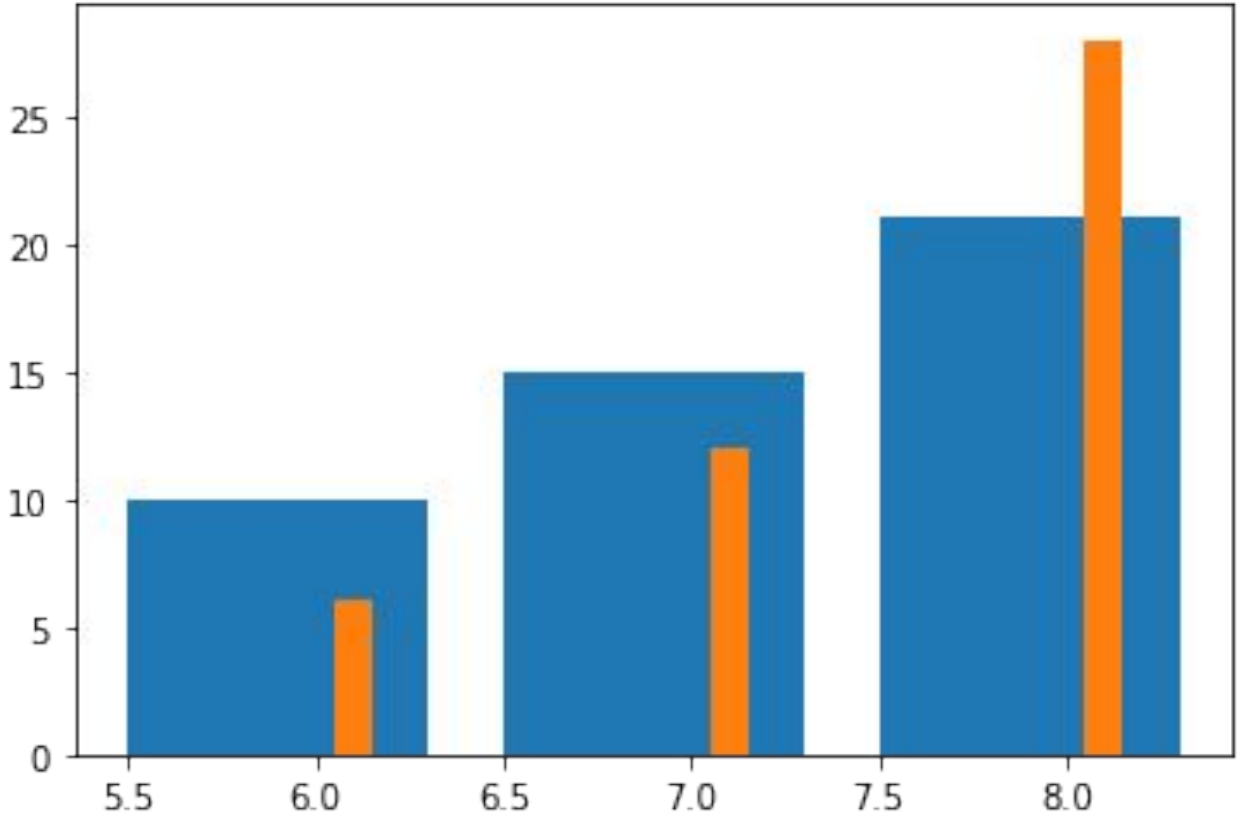
Гистограммы с несколькими наборами данных

```
import matplotlib.pyplot as plt
plt.bar([5.9, 6.9, 7.9], [10, 15, 21], width = 0.2)
plt.bar([6.1, 7.1, 8.1], [6, 12, 28], width = 0.2)
plt.show()
```



Гистограммы с несколькими наборами данных

```
import matplotlib.pyplot as plt
plt.bar([5.9, 6.9, 7.9], [10, 15, 21], width = 0.8)
plt.bar([6.1, 7.1, 8.1], [6, 12, 28], width = 0.1)
plt.show()
```

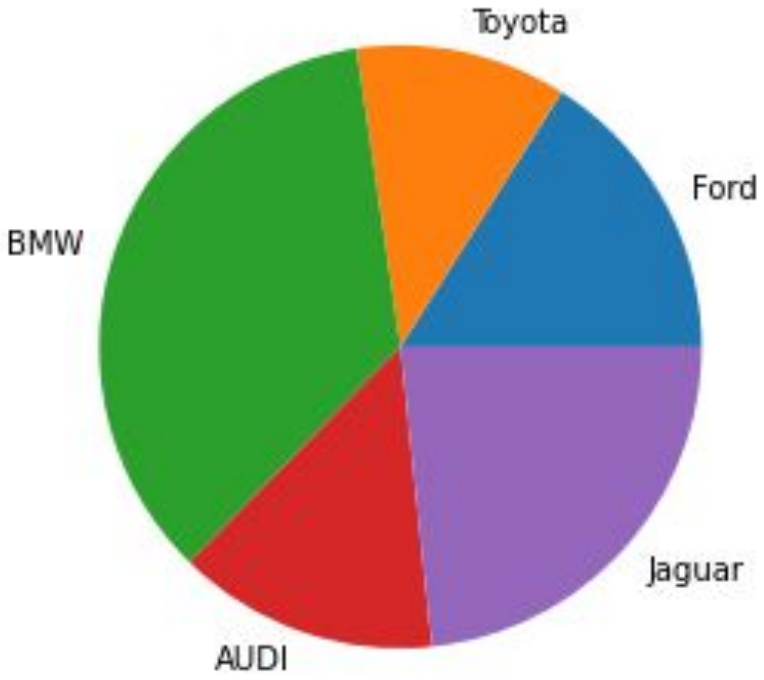


Круговые диаграммы

- **Круговые диаграммы** — это наглядный способ показать доли компонентов в наборе.
- Они идеально подходят для отчётов, презентаций и т.п.
- Для построения круговых диаграмм в Matplotlib используется функция `pie()`.

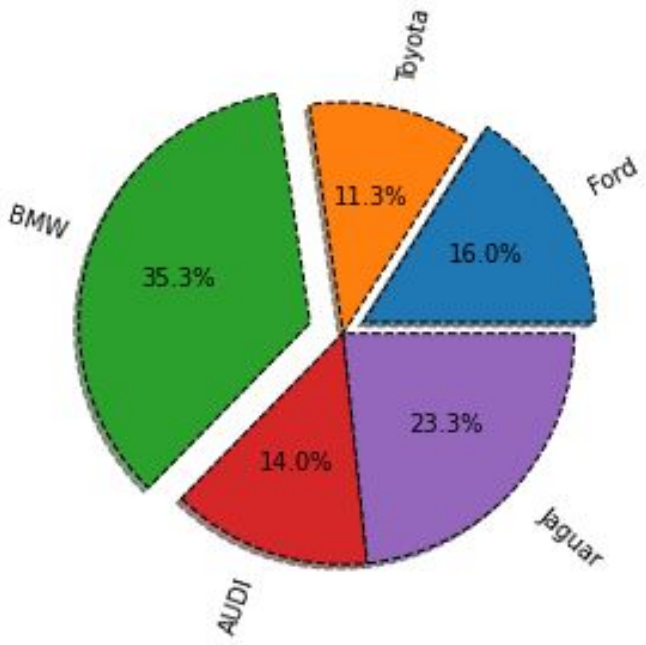
Круговая диаграмма

```
import matplotlib.pyplot as plt
vals = [24, 17, 53, 21, 35]
labels = ['Ford', 'Toyota', 'BMW', 'AUDI', 'Jaguar']
fig, ax = plt.subplots()
ax.pie(vals, labels=labels)
ax.axis('equal')
```



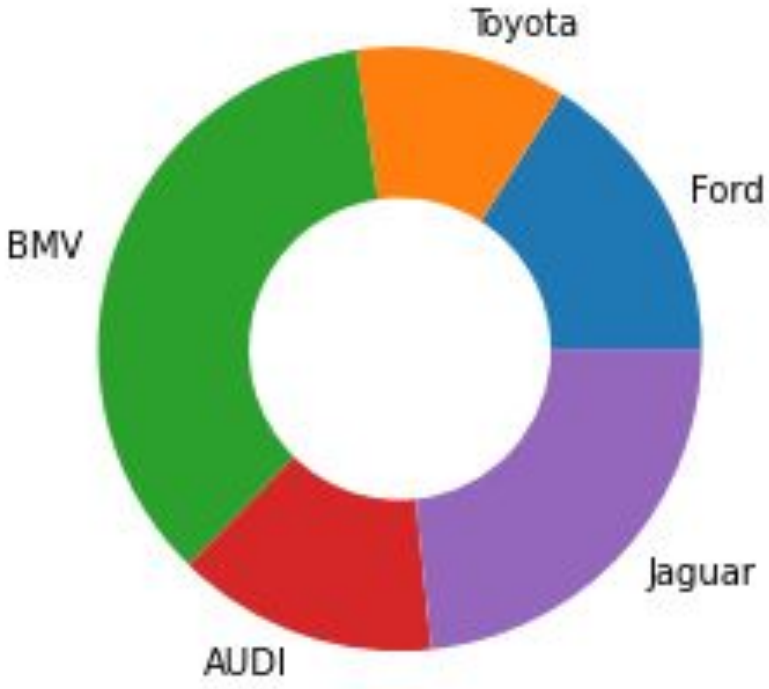
Модифицированная круговая диаграмма

```
import matplotlib.pyplot as plt
vals = [24, 17, 53, 21, 35]
labels = ['Ford', 'Toyota', 'BMW', 'AUDI', 'Jaguar']
explode = (0.1, 0, 0.15, 0, 0)
fig, ax = plt.subplots()
ax.pie(vals, labels=labels, autopct='%1.1f%%', shadow=True,
explode=explode, wedgeprops={'lw':1, 'ls':'--','edgecolor':'k
'},
rotatelabels=True)
ax.axis('equal')
```



Круговая диаграмма с отверстием

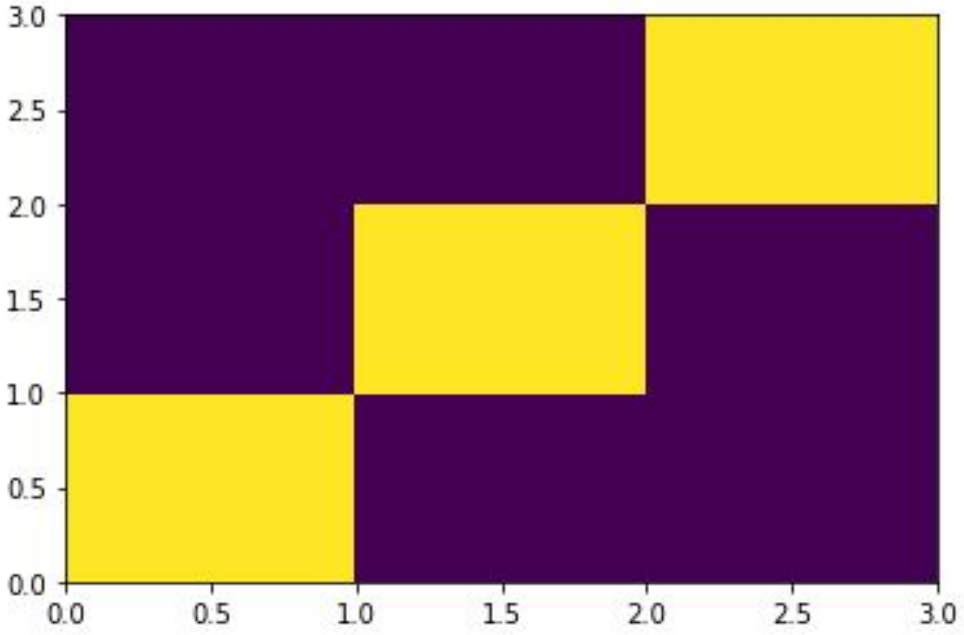
```
import matplotlib.pyplot as plt
vals = [24, 17, 53, 21, 35]
labels = ['Ford', 'Toyota', 'BMV', 'AUDI', 'Jaguar']
fig, ax = plt.subplots()
ax.pie(vals, labels=labels, wedgeprops=dict(width=0.5))
```



Визуализация двумерных массивов

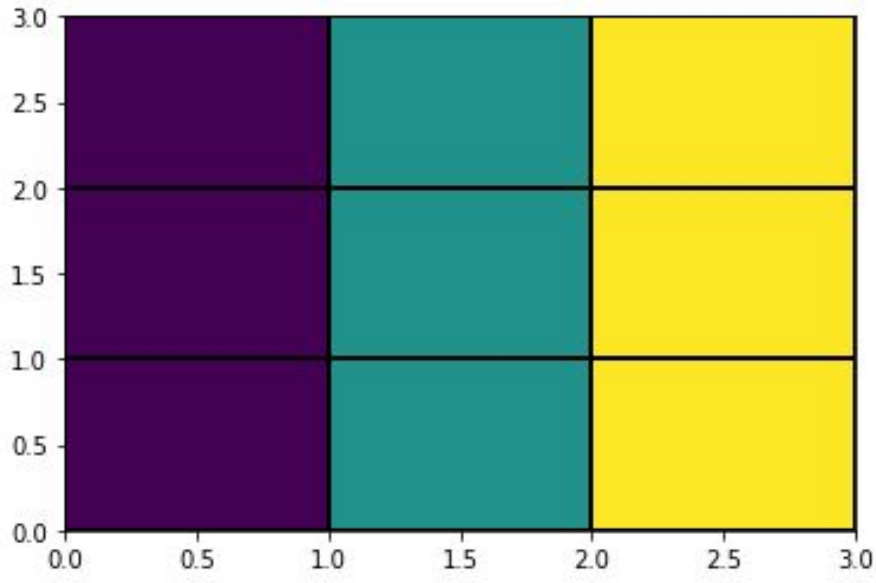
Визуализация двумерных массивов

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

a = [[1, 0, 0],
     [0, 1, 0],
     [0, 0, 1]]

plt.pcolor(a)
```



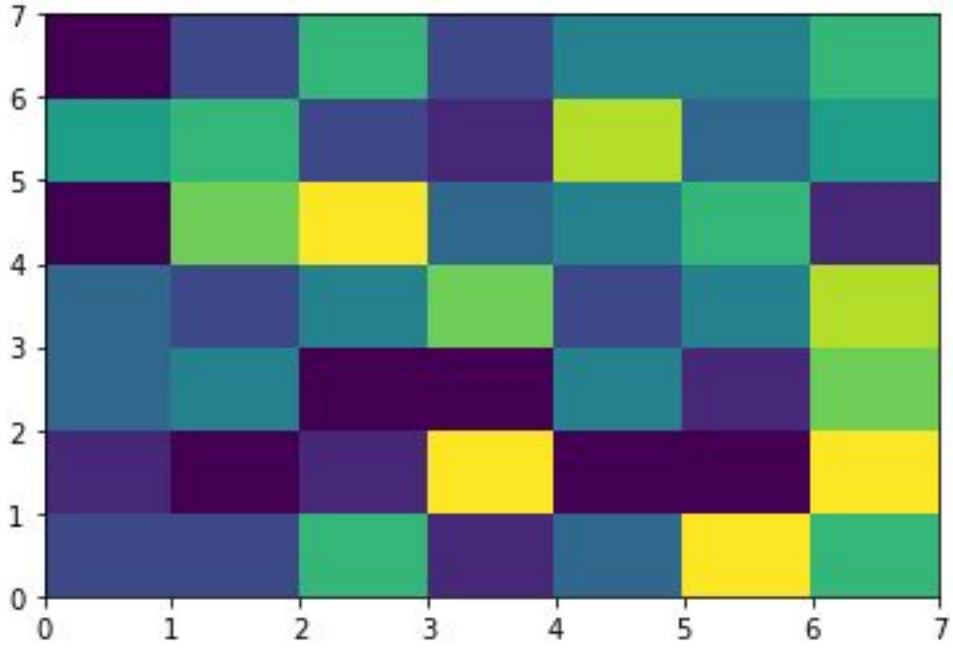
Визуализация двумерных массивов

```
import matplotlib.pyplot as plt
import numpy as np
a = [[0, 1, 2],
     [0, 1, 2],
     [0, 1, 2]]

plt.pcolormesh(a, edgecolors='black')
```



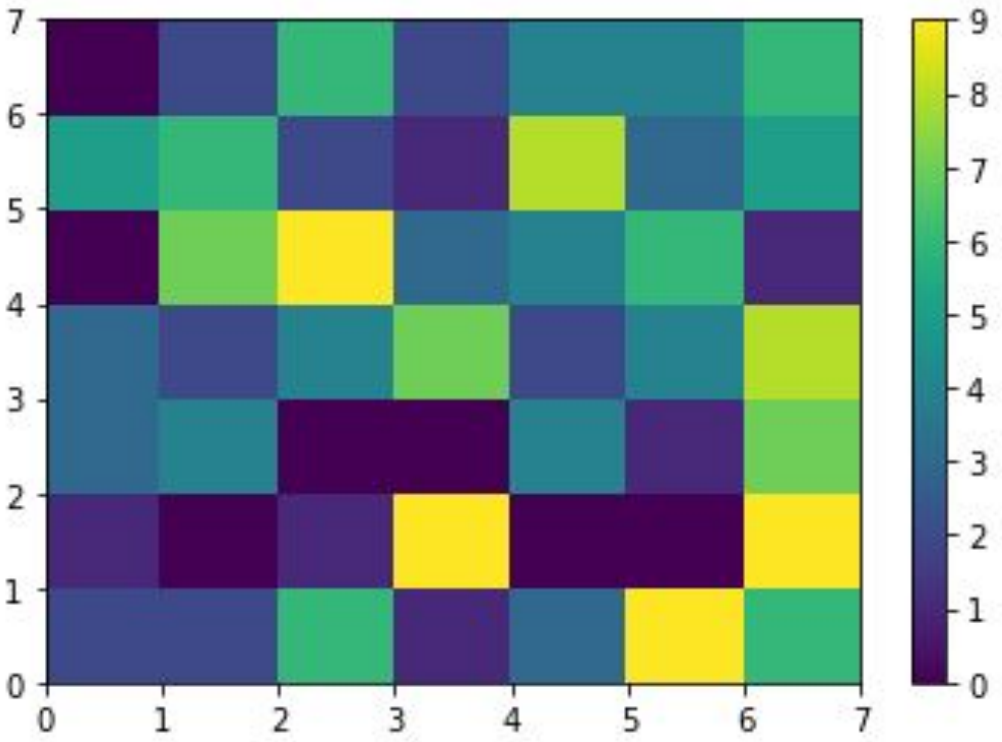
Цветовое распределение

```
import numpy as np
np.random.seed(123)
vals = np.random.randint(10, size=(7, 7))
plt.pcolor(vals)
```



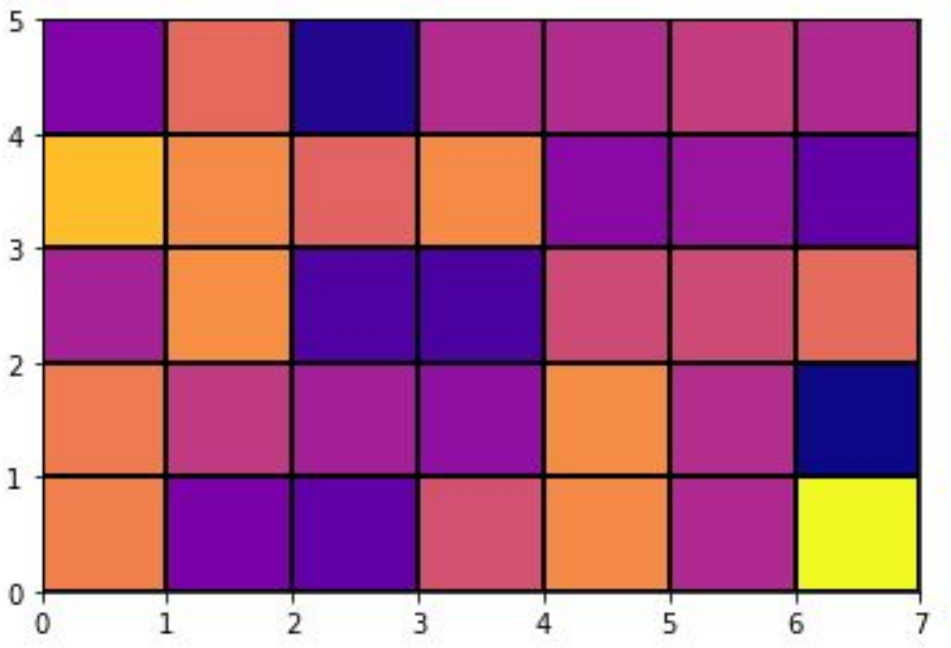
Цветовая полоса для заданного цветового распределения

```
import numpy as np
np.random.seed(123)
vals = np.random.randint(10, size=(7, 7))
plt.pcolor(vals)
plt.colorbar()
```



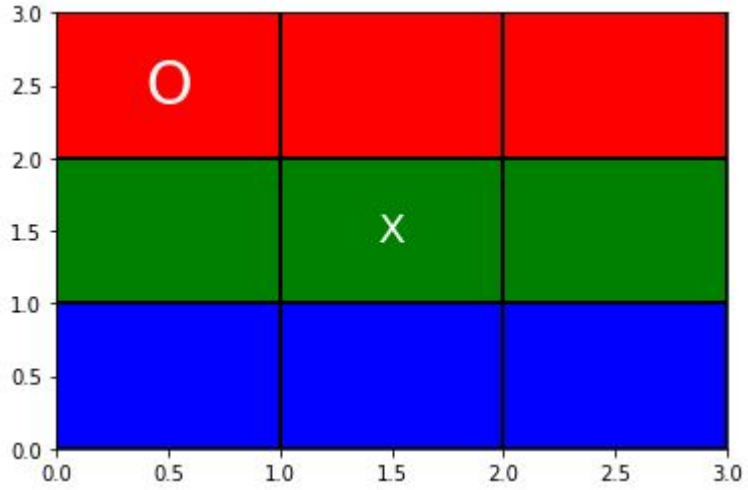
Визуализация двумерного набора данных с использованием pcolormesh()

```
import matplotlib.pyplot as plt
import numpy as np
np.random.seed(123)
data = np.random.rand(5, 7)
plt.pcolormesh(data, cmap='plasma', edgecolors='k',
shading='flat')
```



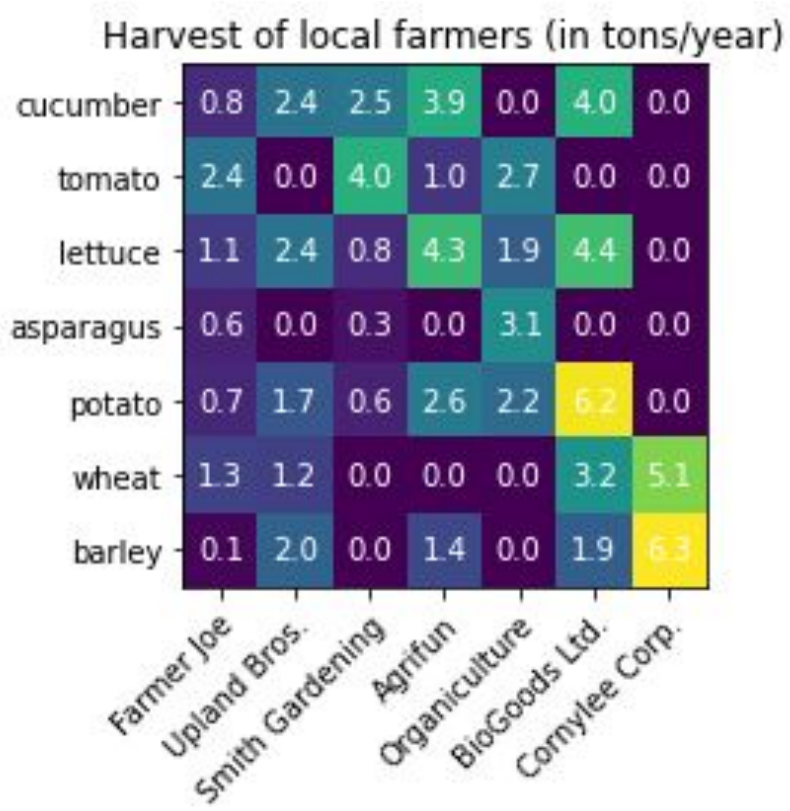
Добавление текста

```
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
import numpy as np
a = [[0, 1, 2],
     [0, 1, 2],
     [0, 1, 2]]

b = [[0, 0, 0],
     [1, 1, 1],
     [2, 2, 2]]

colours = (["blue", "green", "red"])
cmap = ListedColormap(colours)
plt.pcolormesh(a, edgecolors='black', cmap=cmap)

plt.pcolormesh(b, edgecolors='black', cmap=cmap)
plt.text(1.5, 1.5, 'X', color='white', fontsize='20', ha='center', va='center')
plt.text(0.5, 2.5, 'O', color='white', fontsize='30', ha='center', va='center')
```



Тепловые карты

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
vegetables = ["cucumber", "tomato", "lettuce", "asparagus",
              "potato", "wheat", "barley"]
farmers = ["Farmer Joe", "Upland Bros.", "Smith Gardening",
           "Agrifun", "Organiculture", "BioGoods Ltd.", "Cornylee Corp."]
harvest = np.array([[0.8, 2.4, 2.5, 3.9, 0.0, 4.0, 0.0],
                    [2.4, 0.0, 4.0, 1.0, 2.7, 0.0, 0.0],
                    [1.1, 2.4, 0.8, 4.3, 1.9, 4.4, 0.0],
                    [0.6, 0.0, 0.3, 0.0, 3.1, 0.0, 0.0],
                    [0.7, 1.7, 0.6, 2.6, 2.2, 6.2, 0.0],
                    [1.3, 1.2, 0.0, 0.0, 0.0, 3.2, 5.1],
                    [0.1, 2.0, 0.0, 1.4, 0.0, 1.9, 6.3]])
fig, ax = plt.subplots()
im = ax.imshow(harvest)

# We want to show all ticks...
ax.set_xticks(np.arange(len(farmers)))
ax.set_yticks(np.arange(len(vegetables)))
# ... and label them with the respective list entries
ax.set_xticklabels(farmers)
ax.set_yticklabels(vegetables)

# Rotate the tick labels and set their alignment.
plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
         rotation_mode="anchor")

# Loop over data dimensions and create text annotations.
for i in range(len(vegetables)):
    for j in range(len(farmers)):
        text = ax.text(j, i, harvest[i, j],
                       ha="center", va="center", color="w")

ax.set_title("Harvest of local farmers (in tons/year)")
fig.tight_layout()
plt.show()
```



https://matplotlib.org/stable/gallery/images_contours_and_fields/image_annotated_heatmap.html#sphx-glr-gallery-images-contours-and-fields-image-annotated-heatmap-py

Компоновка нескольких графиков вместе

Компоновка нескольких графиков вместе

Вариант подключения

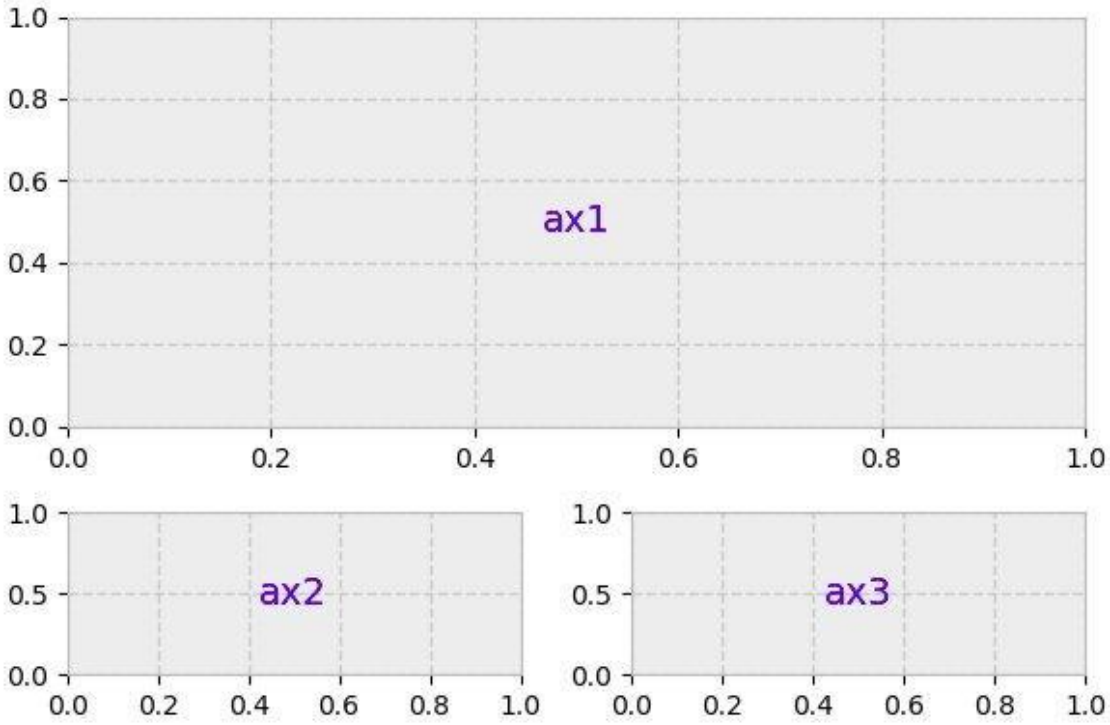
```
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
```

Примеры

https://matplotlib.org/stable/tutorials/intermediate/gridspec.html

matplotlib - gridspec

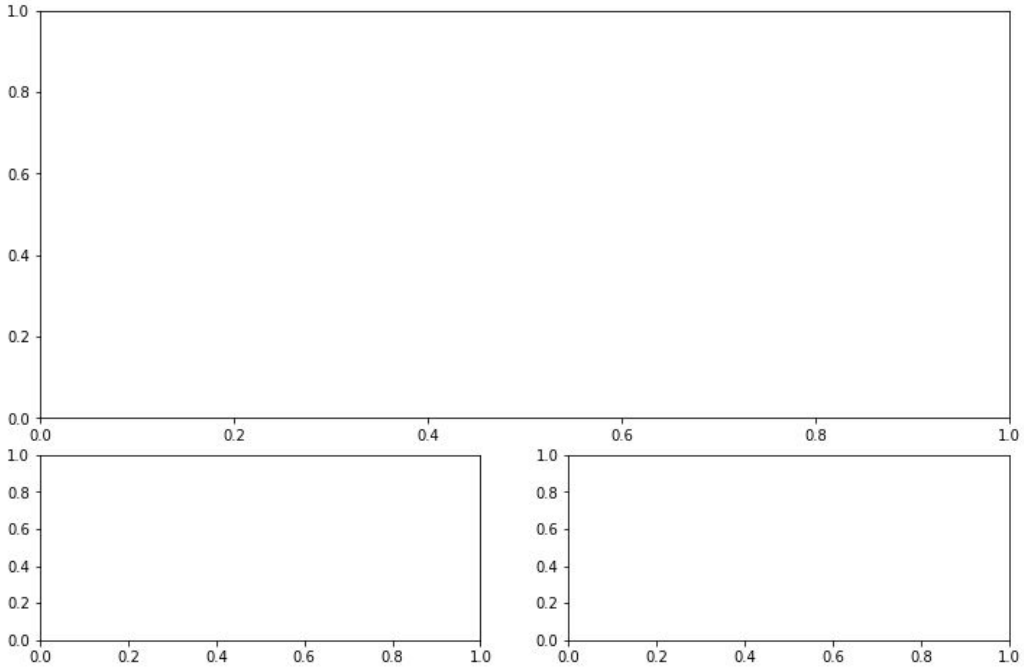
- Модуль gridspec библиотеки matplotlib открывает расширенные возможности для настройки объектов под графиком. subplot2grid() отлично взаимодействует с этим модулем и позволяет создавать например такие вар



matplotlib - gridspec

```
import matplotlib.pyplot as plt
gridsize = (3, 2)
fig = plt.figure(figsize=(12, 8))
ax1 = plt.subplot2grid(gridsize, (0, 0), colspan=2, rowspan=2)
ax2 = plt.subplot2grid(gridsize, (2, 0))
ax3 = plt.subplot2grid(gridsize, (2, 1))

plt.show()
```

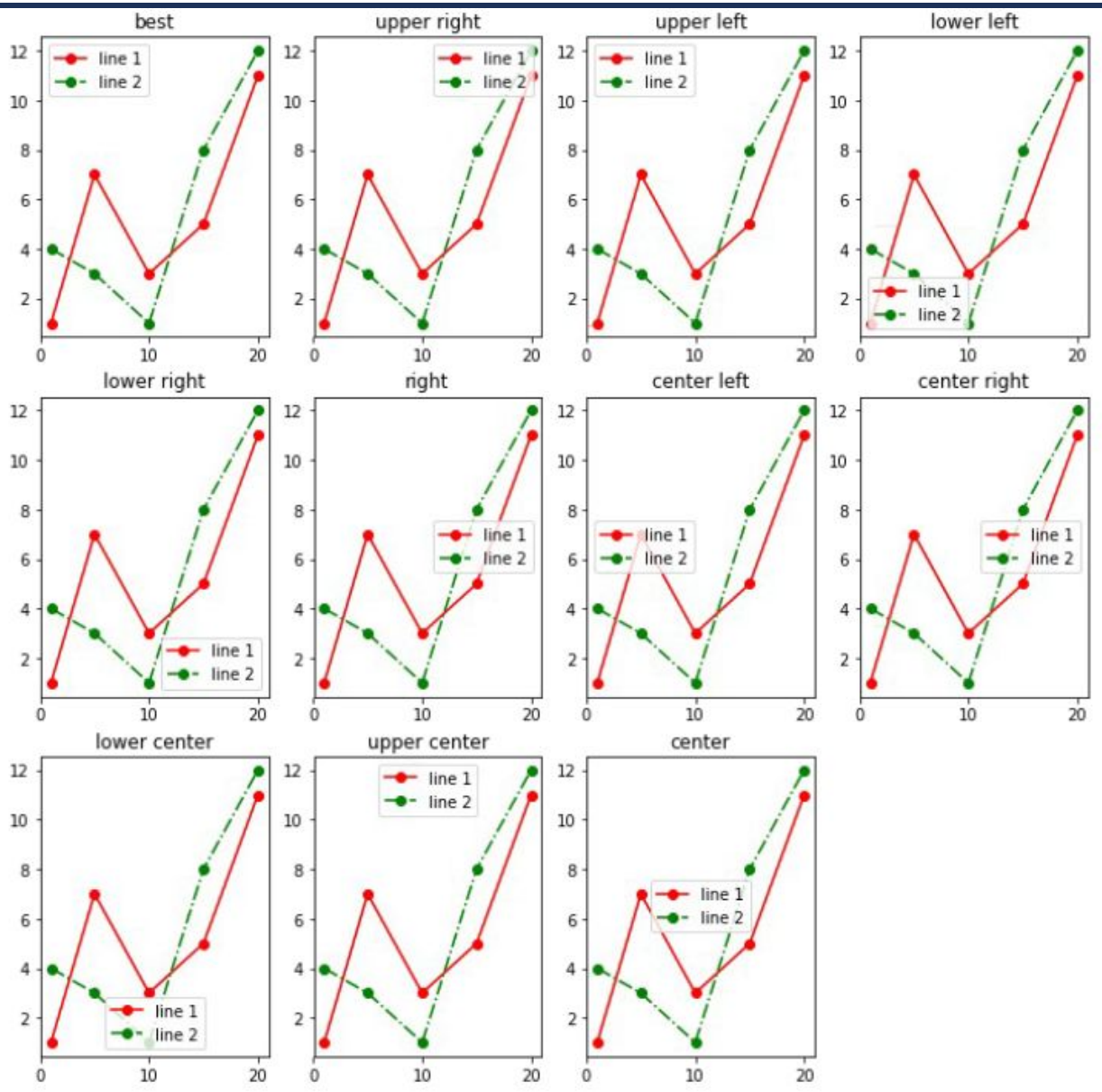
subplot2grid() – это (**ряд, строка**) локация объекта Axes со следующей сеткой:



Matplotlib – Различные варианты расположения легенды на графике

```
import matplotlib.pyplot as plt
locs = ['best', 'upper right', 'upper left', 'lower left',
'lower right', 'right', 'center left', 'center right',
'lower center', 'upper center', 'center']

plt.figure(figsize=(12, 12))

for i in range(3):
  for j in range(4):
    if i*4+j < 11:
      plt.subplot(3, 4, i*4+j+1)
      plt.title(locs[i*4+j])
      plt.plot(x, y1, 'o-r', label='line 1')
      plt.plot(x, y2, 'o-.g', label='line 2')
      plt.legend(loc=locs[i*4+j])
    else:
      break
```

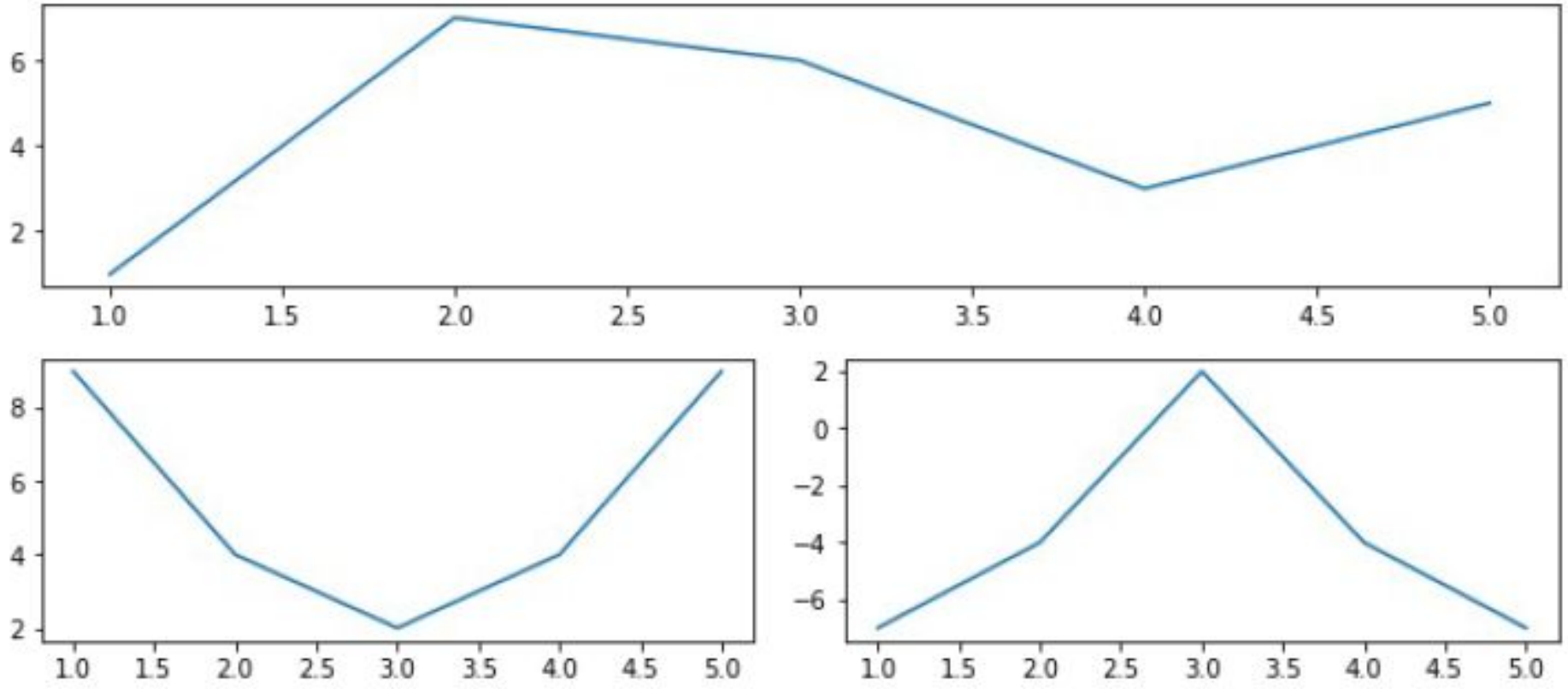
Matplotlib – Различные варианты расположения легенды на графике



Matplotlib – Свободная компоновка

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y1 = [9, 4, 2, 4, 9]
y2 = [1, 7, 6, 3, 5]
y3 = [-7, -4, 2, -4, -7]

#Построим графики в новой компоновке:
fg = plt.figure(figsize=(9, 4), constrained_layout=True)
gs = fg.add_gridspec(2, 2)

fig_ax_1 = fg.add_subplot(gs[0, :])
plt.plot(x, y2)
fig_ax_2 = fg.add_subplot(gs[1, 0])
plt.plot(x, y1)
fig_ax_3 = fg.add_subplot(gs[1, 1])
plt.plot(x, y3)
```

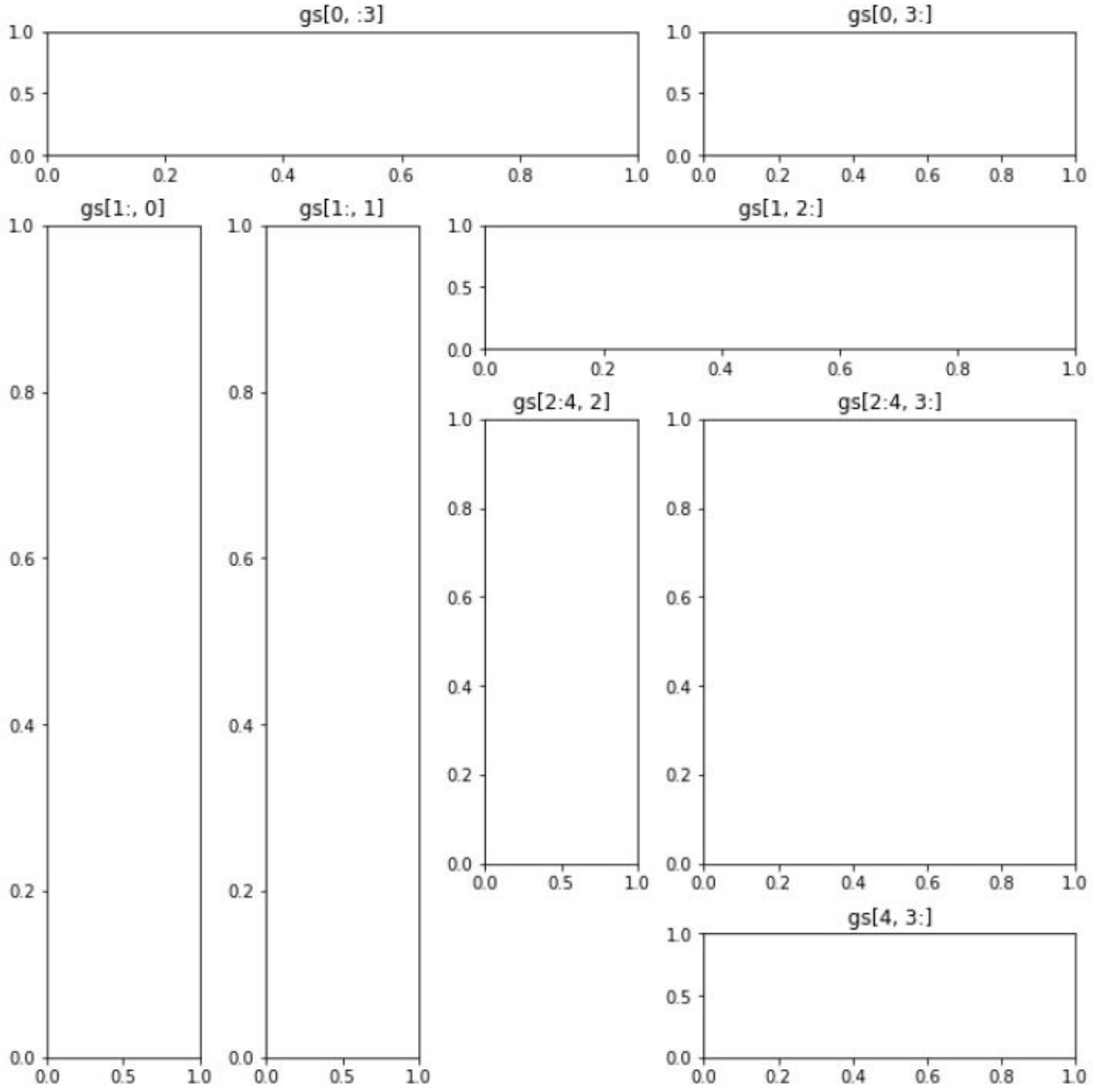
Matplotlib – Свободная компоновка



Matplotlib – Свободная компоновка

```
import matplotlib.pyplot as plt
fg = plt.figure(figsize=(9, 9), constrained_layout=True)
gs = fg.add_gridspec(5, 5)
fig_ax_1 = fg.add_subplot(gs[0, :3])
fig_ax_1.set_title('gs[0, :3]')
fig_ax_2 = fg.add_subplot(gs[0, 3:])
fig_ax_2.set_title('gs[0, 3:]')
fig_ax_3 = fg.add_subplot(gs[1:, 0])
fig_ax_3.set_title('gs[1:, 0]')
fig_ax_4 = fg.add_subplot(gs[1:, 1])
fig_ax_4.set_title('gs[1:, 1]')
fig_ax_5 = fg.add_subplot(gs[1, 2:])
fig_ax_5.set_title('gs[1, 2:]')
fig_ax_6 = fg.add_subplot(gs[2:4, 2])
fig_ax_6.set_title('gs[2:4, 2]')
fig_ax_7 = fg.add_subplot(gs[2:4, 3:])
fig_ax_7.set_title('gs[2:4, 3:]')
fig_ax_8 = fg.add_subplot(gs[4, 3:])
fig_ax_8.set_title('gs[4, 3:]')
```

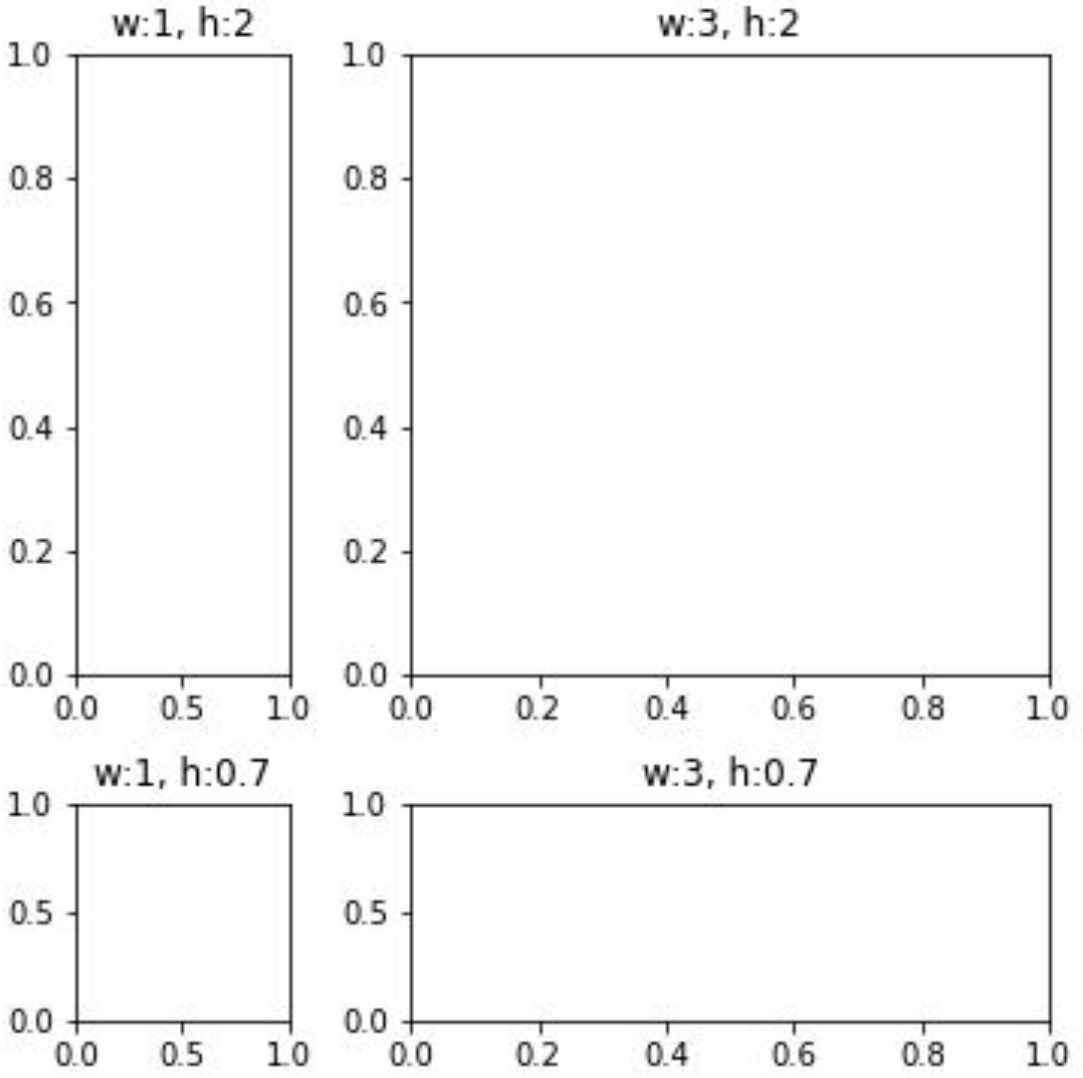
Matplotlib – Свободная компоновка



Matplotlib – Свободная компоновка

```
import matplotlib.pyplot as plt
fg = plt.figure(figsize=(5, 5),constrained_layout=True)
widths = [1, 3]
heights = [2, 0.7]
gs = fg.add_gridspec(ncols=2, nrows=2, width_ratios=widths,
height_ratios=heights)

fig_ax_1 = fg.add_subplot(gs[0, 0])
fig_ax_1.set_title('w:1, h:2’)

fig_ax_2 = fg.add_subplot(gs[0, 1])
fig_ax_2.set_title('w:3, h:2’)

fig_ax_3 = fg.add_subplot(gs[1, 0])
fig_ax_3.set_title('w:1, h:0.7’)

fig_ax_4 = fg.add_subplot(gs[1, 1])
fig_ax_4.set_title('w:3, h:0.7')
```

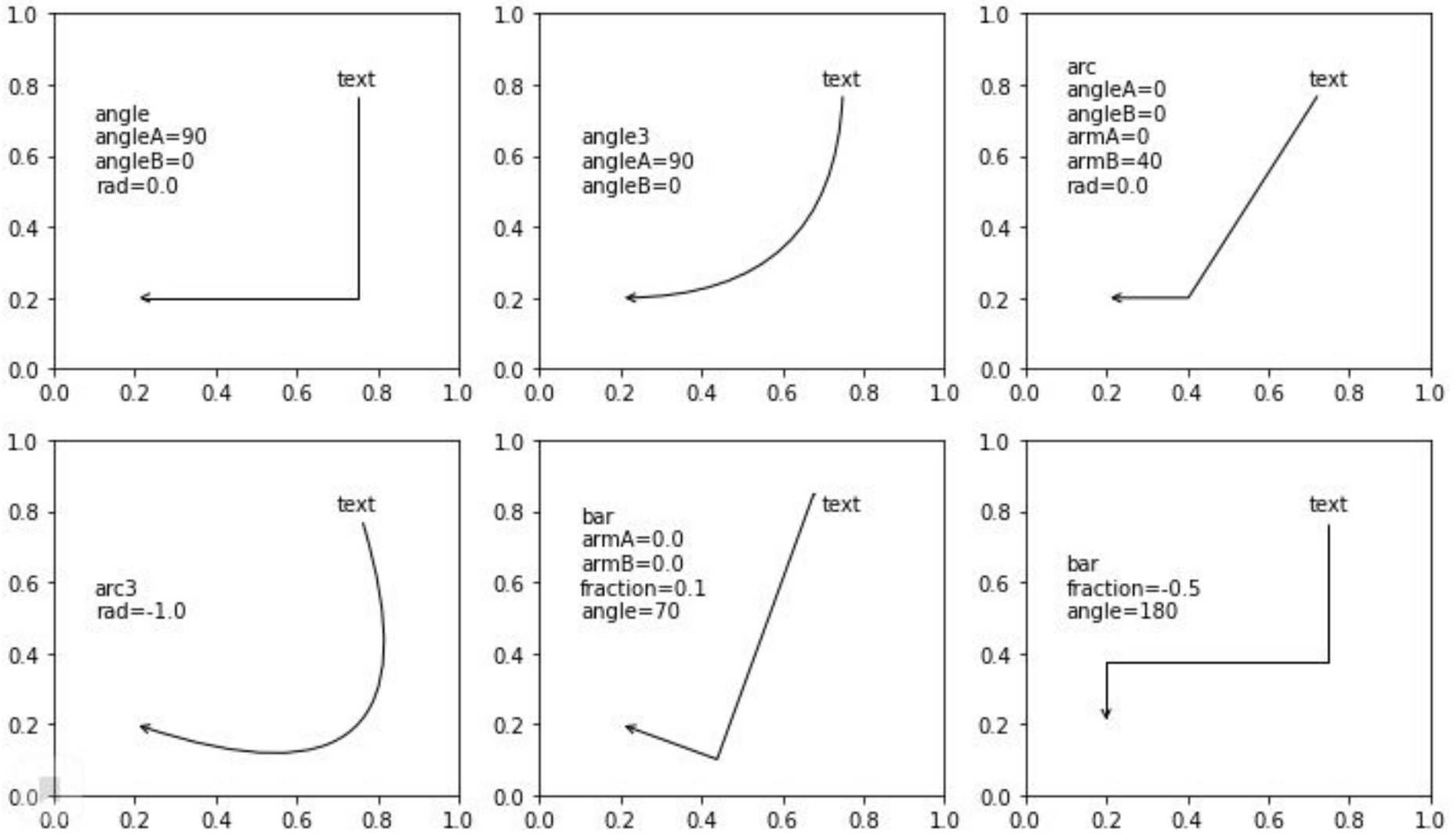
Matplotlib – Свободная компоновка



Matplotlib – Стили соединительной линии аннотации

```
import matplotlib.pyplot as plt
import math

fig, axs = plt.subplots(2, 3, figsize=(12, 7))
conn_style=[
'angle,angleA=90,angleB=0,rad=0.0',
'angle3,angleA=90,angleB=0',
'arc,angleA=0,angleB=0,armA=0,armB=40,rad=0.0',
'arc3,rad=-1.0',
'bar,armA=0.0,armB=0.0,fraction=0.1,angle=70',
'bar,fraction=-0.5,angle=180',
]

for i in range(2):
  for j in range(3):
    axs[i, j].text(0.1, 0.5, '\n'.join(conn_style[i*3+j].split(',')))
    axs[i, j].annotate('text', xy=(0.2, 0.2), xycoords='data',
xytext=(0.7, 0.8), textcoords='data', arrowprops=dict(arrowstyle='->',
connectionstyle=conn_style[i*3+j]))
```

Matplotlib – Стили соединительной линии аннотации



Белорусско-Российский университет
Кафедра «Программное обеспечение информационных технологий»

Информатика. Программирование на Python
Тема: Python. Основы. Визуализация данных.

Благодарю за внимание

КУТУЗОВ Виктор Владимирович

Список использованных источников

1. Python
   https://www.python.org/
2. Google Colaboratory
   https://colab.research.google.com/
3. Matplotlib: Visualization with Python
   https://matplotlib.org/
4. Matplotlib User's Guide
   https://matplotlib.org/stable/Matplotlib.pdf
5. Библиотека matplotlib
   https://mipt-stats.gitlab.io/courses/python/06_matplotlib.html
6. Matplotlib Я новичок. Можно попроще? | NumPy
   https://pyprog.pro/mpl/mpl_types_of_graphs.html
7. Matplotlib Gallery
   https://matplotlib.org/stable/gallery/index.html
8. Python в научных вычислениях
   https://inp.nsk.su/~grozin/python/
9. matplotlib: пакет для построения графиков
   https://inp.nsk.su/~grozin/python/b22_matplotlib.html

Список использованных источников

10. Абдрахманов М.И. Python. Визуализация данных. Matplotlib. - Devpractice Team, 2020 – 413 с. https://by1lib.org/book/7229033/21176f?id=7229033&secret=21176f
11. Plotting sine and cosine with Matplotlib and Python https://pythonforundergradengineers.com/plotting-sin-cos-with-matplotlib.html
12. matplotlib / cheatsheets https://github.com/matplotlib/cheatsheets#cheatsheets
13. Руководство пользователя Matplotlib https://matplotlib.org/stable/contents.html
14. Примеры графиков Matplotlib https://matplotlib.org/stable/gallery/index.html
15. Шпаргалки по Matplotlib https://github.com/matplotlib/cheatsheets#cheatsheets
16. Скачать все примеры - Python исходники программ: gallery_python.zip https://matplotlib.org/stable/_downloads/63b34a63fc35d506739b9835d7e98958/gallery_python.zip
17. Скачать все примеры - Jupyter notebooks: gallery_jupyter.zip https://matplotlib.org/stable/_downloads/a70483fff7b46b03f4d5c358b003188f/gallery_jupyter.zip
18. Построение графиков в Python при помощи Matplotlib https://python-scripts.com/matplotlib
19. 50 оттенков matplotlib — The Master Plots (с полным кодом на Python) https://habr.com/ru/post/468295/

Список использованных источников

20. Многомерные графики в Python — от трёхмерных и до шестимерных
https://habr.com/ru/post/456282/

21. Шпаргалка по визуализации данных в Python с помощью Plotly
https://habr.com/ru/post/502958/

22. he 954 most common RGB monitor colors, as defined by several hundred thousand participants in the xkcd color name survey.
https://xkcd.com/color/rgb/