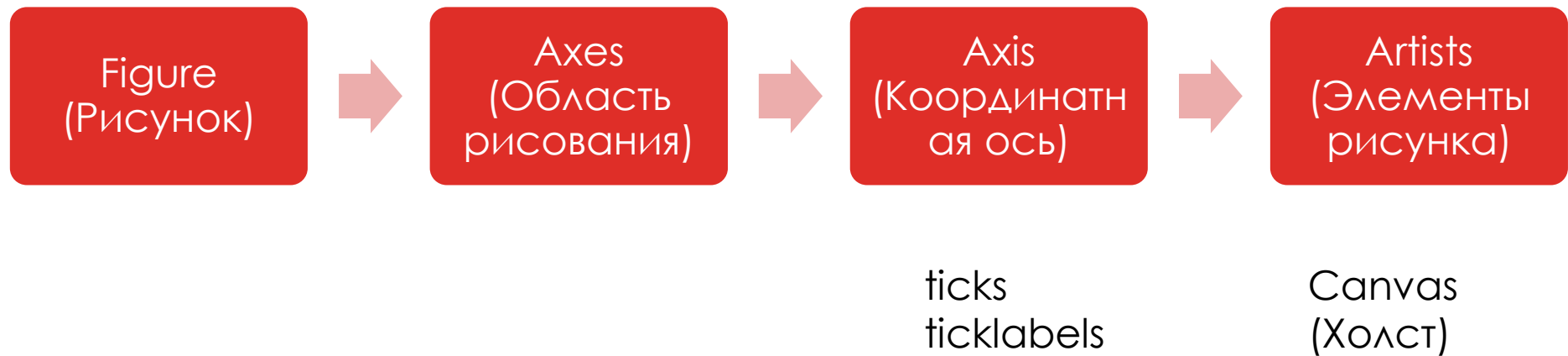


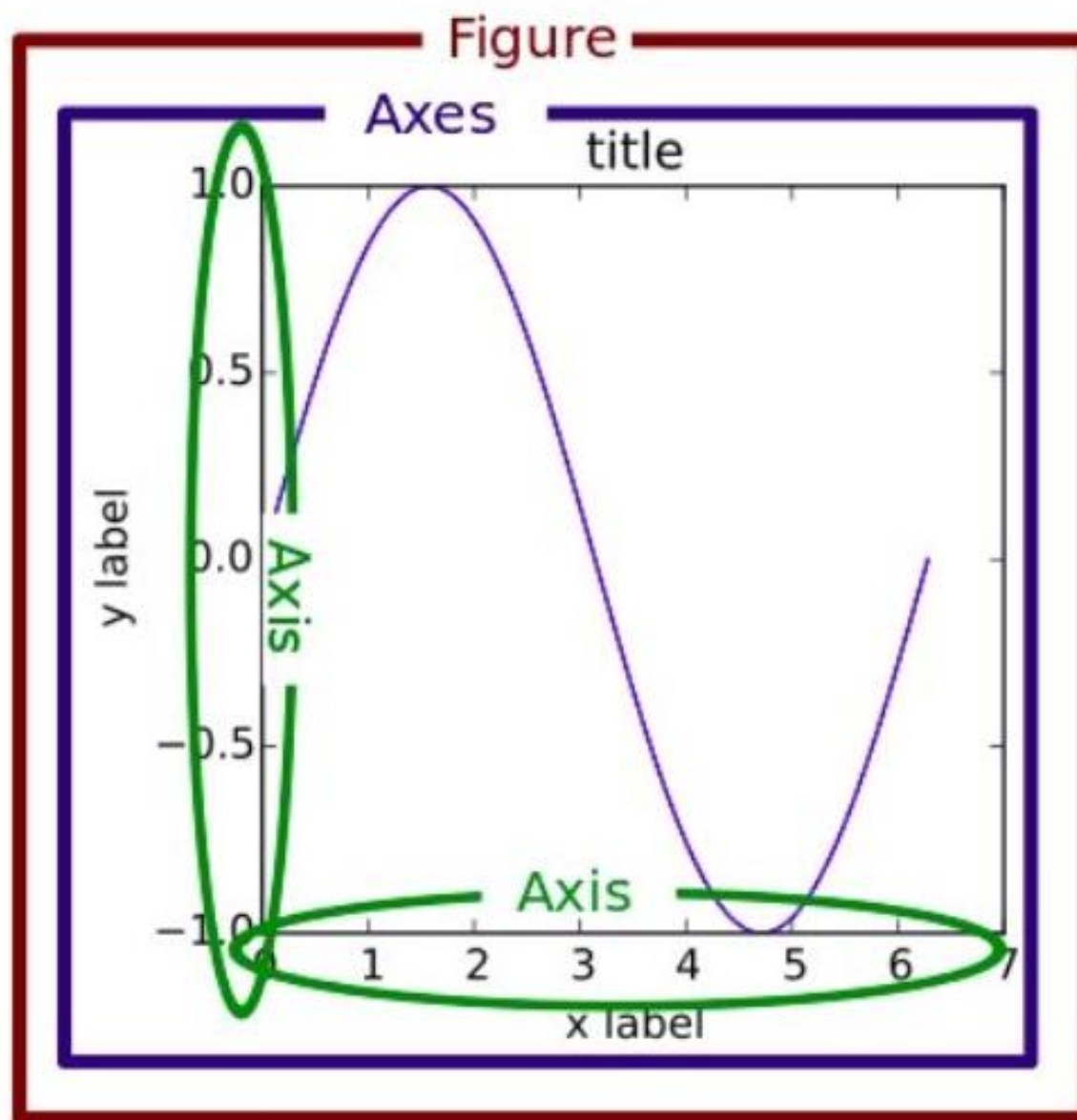


MATPLOTLIB

СТРУКТУРА РИСУНКА

- `import matplotlib as mpl`

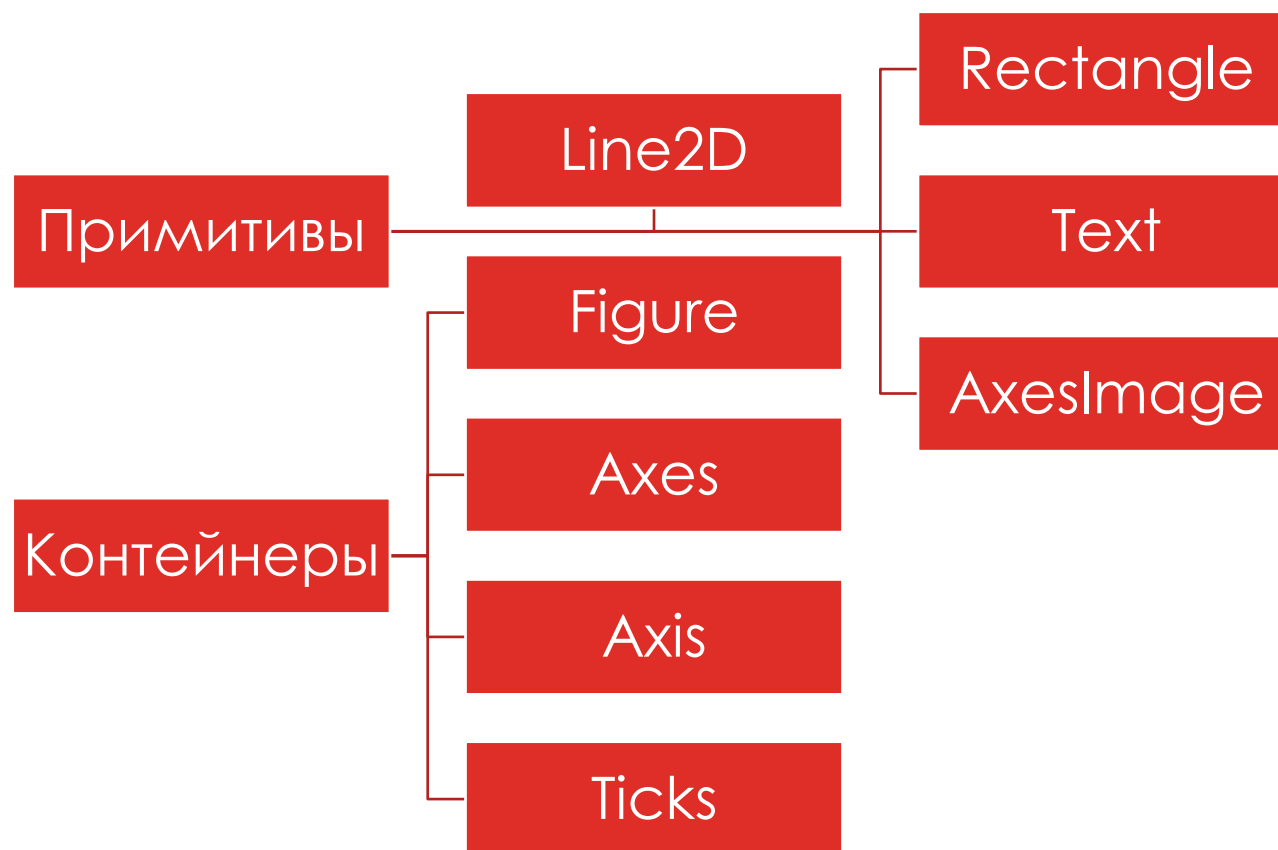




MATPLOTLIB API

1. **matplotlib.backend_bases.FigureCanvas** - абстрактный базовый класс, который позволяет рисовать и визуализировать результаты команд.
2. **matplotlib.backend_bases.Renderer** - объект (абстрактный класс), который знает как рисовать на FigureCanvas;
3. **matplotlib.artist.Artist** - объект, который знает, как использовать визуализатор (renderer), чтобы рисовать на холсте (canvas).

ARTISTS



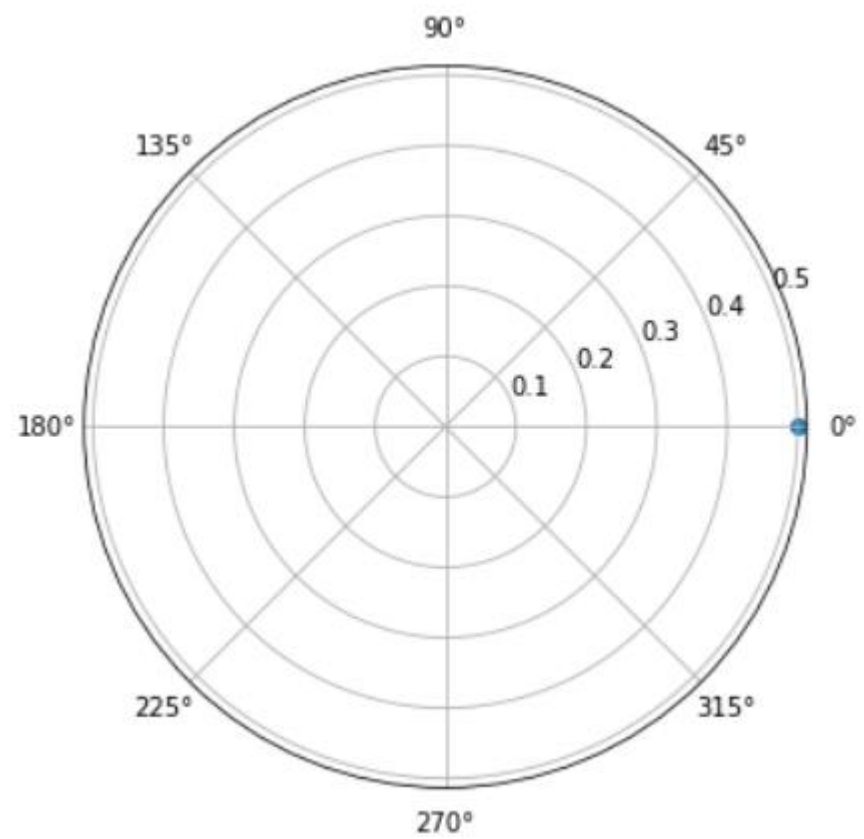
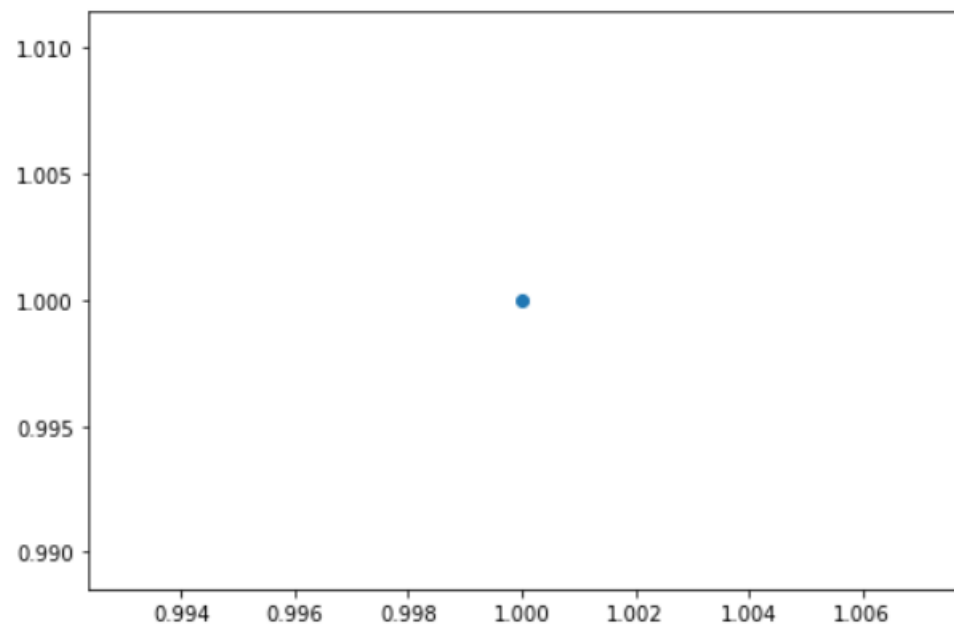


АЛГОРИТМ СОЗДАНИЯ РИСУНКА

- Экземпляр класса Figure
- Области рисования Axes (subplot)
- Возможно настройки координатной сетки, деления и подписей
- Создание графических примитивов

ИНТЕРФЕЙС PYPLOT

- `import matplotlib.pyplot as plt`
- `fig = plt.figure()`
- `ax=fig.add_axes([0,0,1,1])` или `ax = fig.add_axes([0, 0, 1, 1], polar=True)`
- `plt.scatter(1.0, 1.0)`
- `plt.savefig('pic.png', fmt='png')`
- `plt.show()`



1. Область рисования Axes

- Заголовок области рисования -> `plt.title()`;

2. Ось абсцисс Xaxis

- Подпись оси абсцисс OX -> `plt.xlabel()`;

3. Ось абсцисс Yaxis

- Подпись оси абсцисс OY -> `plt.ylabel()`;

4. Легенда -> `plt.legend()`

5. Цветовая шкала -> `plt.colorbar()`

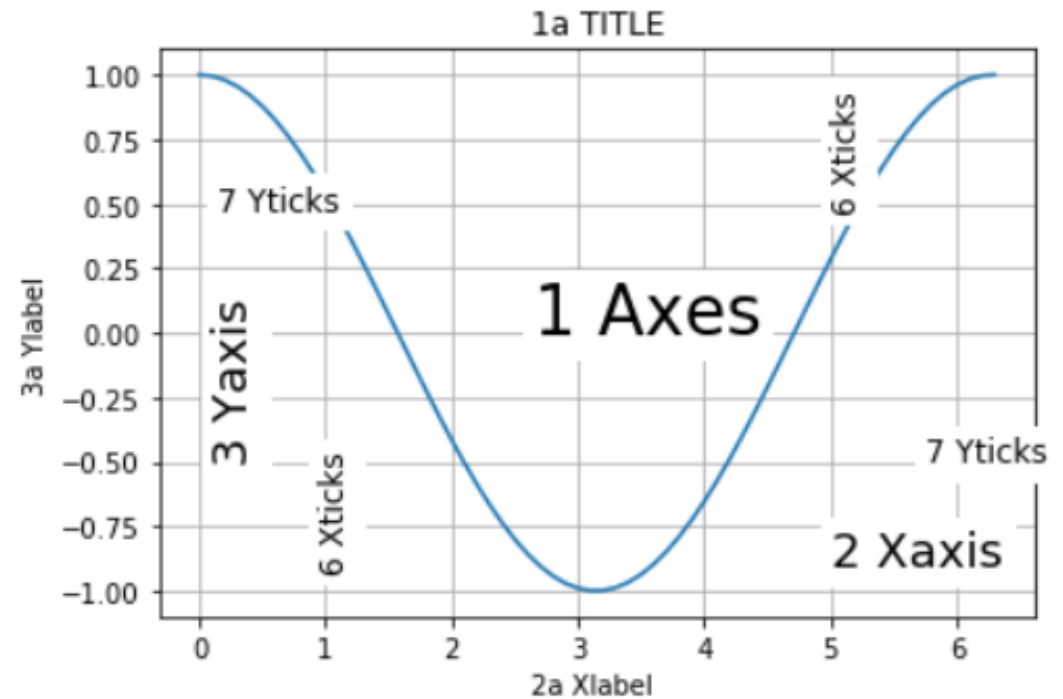
- Подпись горизонтальной оси абсцисс OY -> `cbar.ax.set_xlabel()`;
- Подпись вертикальной оси абсцисс OY -> `cbar.ax.set_ylabel()`;

6. Деления на оси абсцисс OX -> `plt.xticks()`

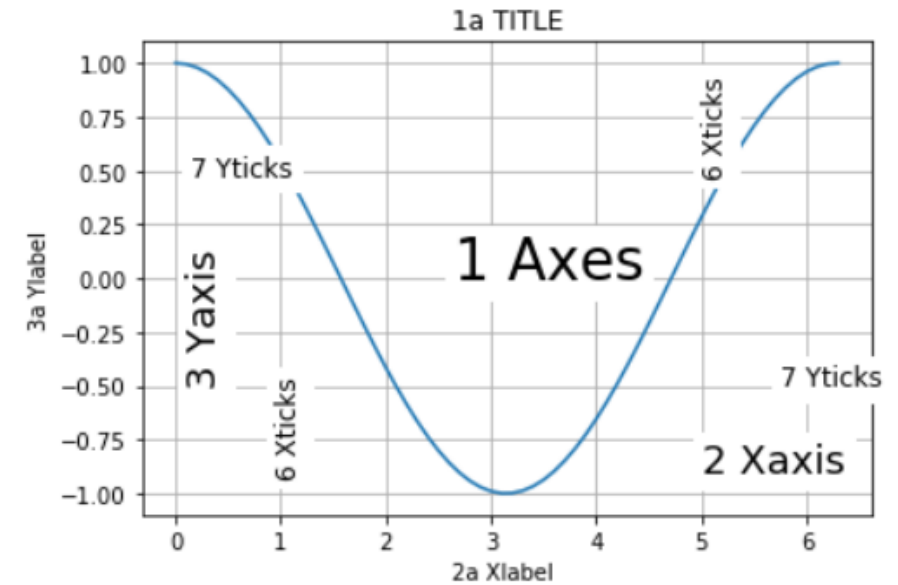
7. Деления на оси ординат OY -> `plt.yticks()`

```
import matplotlib.pyplot as plt
import numpy as np
lag = 0.1
x = np.arange(0.0, 2*np.pi+lag, lag)
y = np.cos(x)
fig = plt.figure()
plt.plot(x, y)
```

```
plt.text(np.pi-0.5, 0, '1 Axes', fontsize=26, bbox=dict(edgecolor='w', color='w'))
plt.text(0.1, 0, '3 Yaxis', fontsize=18, bbox=dict(edgecolor='w', color='w'), rotation=90)
plt.text(5, -0.9, '2 Xaxis', fontsize=18, bbox=dict(edgecolor='w', color='w'))
plt.title('1a TITLE')
```



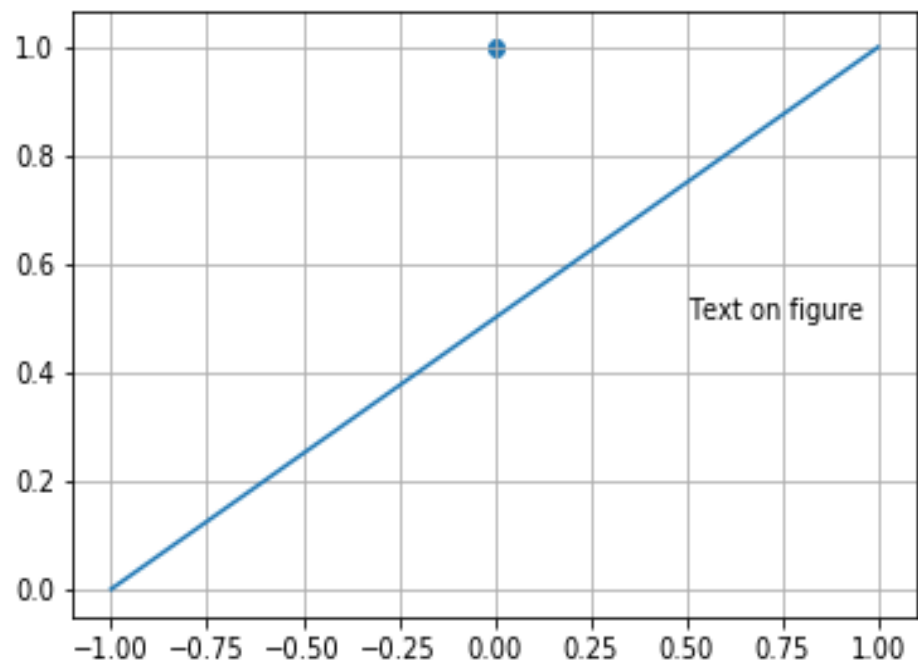
```
plt.ylabel('3a Ylabel')
plt.xlabel('2a Xlabel ')
plt.text(5, 0.85, '6 Xticks', fontsize=12, bbox=dict(edgecolor='w', color='w'), rotation=90)
plt.text(0.95, -0.55, '6 Xticks', fontsize=12, bbox=dict(edgecolor='w', color='w'), rotation=90)
plt.text(5.75, -0.5, '7 Yticks', fontsize=12, bbox=dict(edgecolor='w', color='w'))
plt.text(0.15, 0.475, '7 Yticks', fontsize=12, bbox=dict(edgecolor='w', color='w'))
plt.grid(True)
plt.savefig('example.png', fmt='png')
plt.show()
```



ГРАФИЧЕСКИЕ КОМАНДЫ

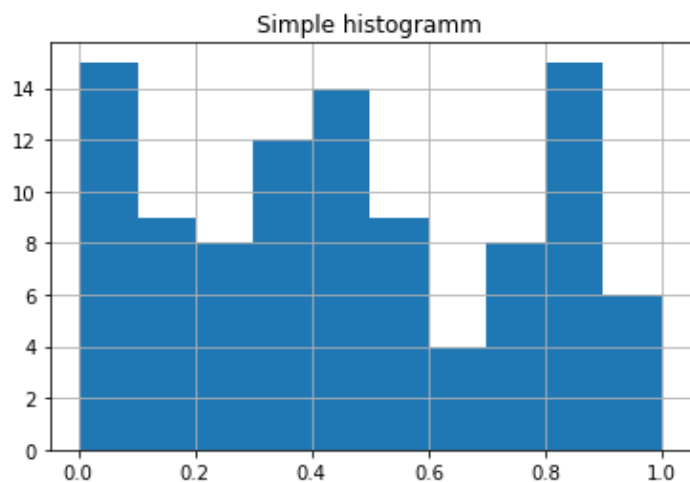
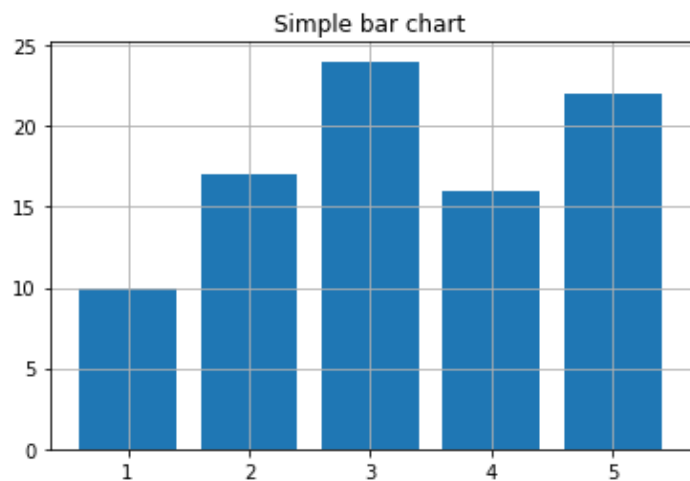
- `plt.scatter()` – маркер, точечное рисование
- `plt.plot()` – ломанная линия
- `plt.text()` – нанесение текста
- `plt.bar()` – столбчатая диаграмма
- `plt.hist()` – гистограмма
- `plt.pie()` – круговая диаграмма

ПРИМЕР



- `import matplotlib.pyplot as plt`
- `fig = plt.figure()`
- `scatter1 = plt.scatter(0.0, 1.0)`
- `graph1 = plt.plot([-1.0, 1.0], [0.0, 1.0])`
- `text1 = plt.text(0.5, 0.5, 'Text on figure')`
- `grid1 = plt.grid(True)`
- `plt.show()`

ПРИМЕР



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

```
s = ['one', 'two', 'three ', 'four' , 'five']
x = [1, 2, 3, 4, 5]
z = np.random.random(100)
z1 = [10, 17, 24, 16, 22]
z2 = [12, 14, 21, 13, 17]
```

```
# bar()
fig = plt.figure()
plt.bar(x, z1)
plt.title('Simple bar chart')
plt.grid(True)    # ЛИНИИ ВСПОМОГАТЕЛЬНОЙ СЕТКИ
```

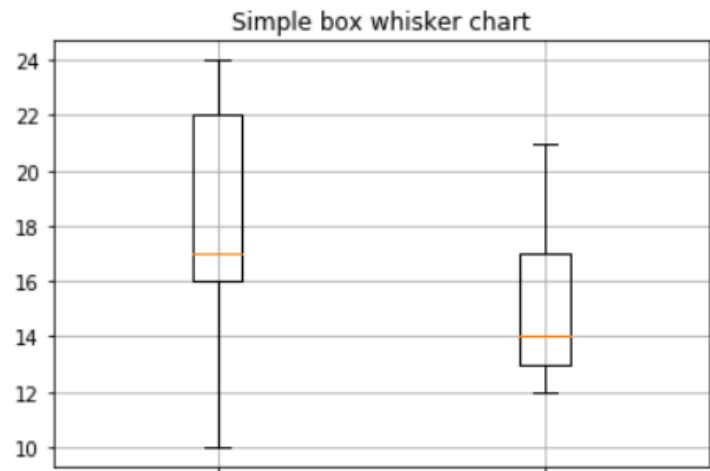
```
# hist()
fig = plt.figure()
plt.hist(z)
plt.title('Simple histogramm')
plt.grid(True)
```

```
..
```

ПРИМЕР



```
# pie()  
fig = plt.figure()  
plt.pie(x, labels=s)  
plt.title('Simple pie chart')
```

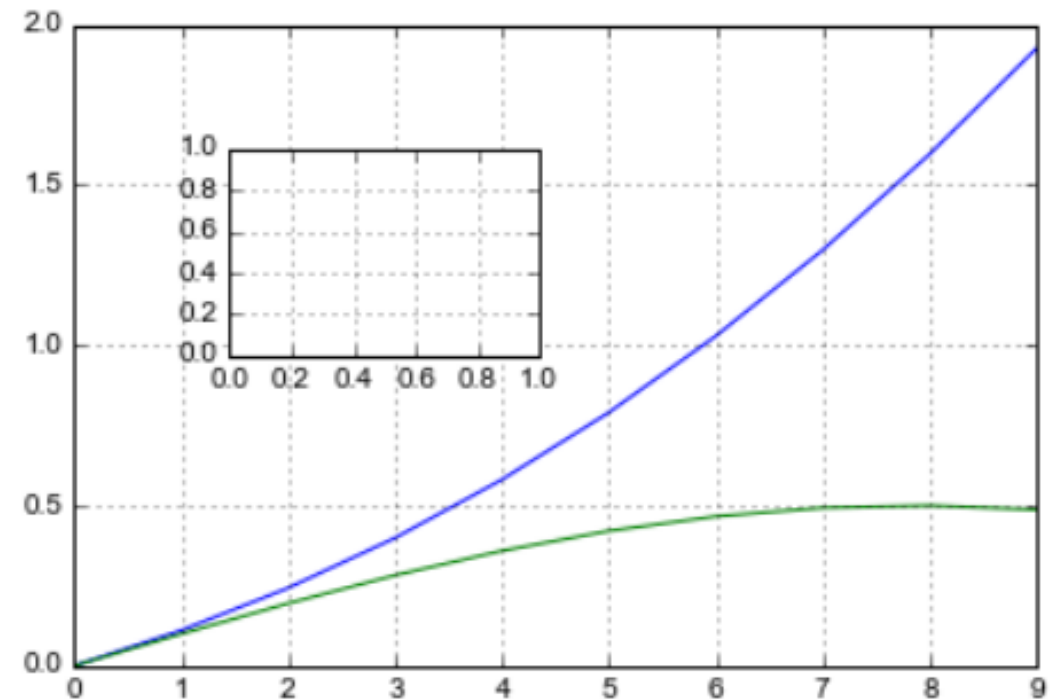


```
fig = plt.figure()  
plt.boxplot([z1, z2])  
plt.title('Simple box whisker chart')  
plt.grid(True)
```

```
plt.show()
```

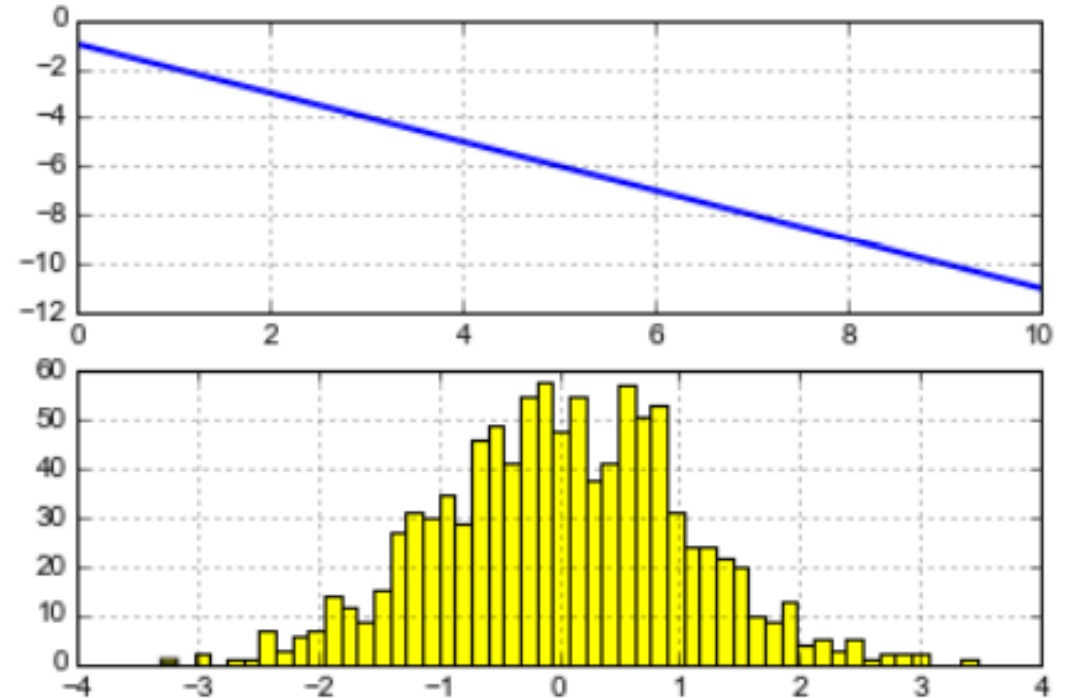

КОНТЕЙНЕР FIGURE

- `import matplotlib.pyplot as plt`
- `import numpy as np`
- `fig = plt.figure()`
- `ax = fig.add_subplot(111)`
- `box = [0.25, 0.5, 0.25, 0.25]`
- `ax2 = fig.add_axes(box)`
- `x = np.arange(0.0, 1.0, 0.1)`
- `y = np.sin(x)*np.exp(x)`
- `z = np.cos(x)*np.sin(x)`
- `ax.plot(y)`
- `ax.plot(z)`
- `for ax in fig.axes:`
 - `ax.grid(True)`
- `plt.show()`



КОНТЕЙНЕР AXES

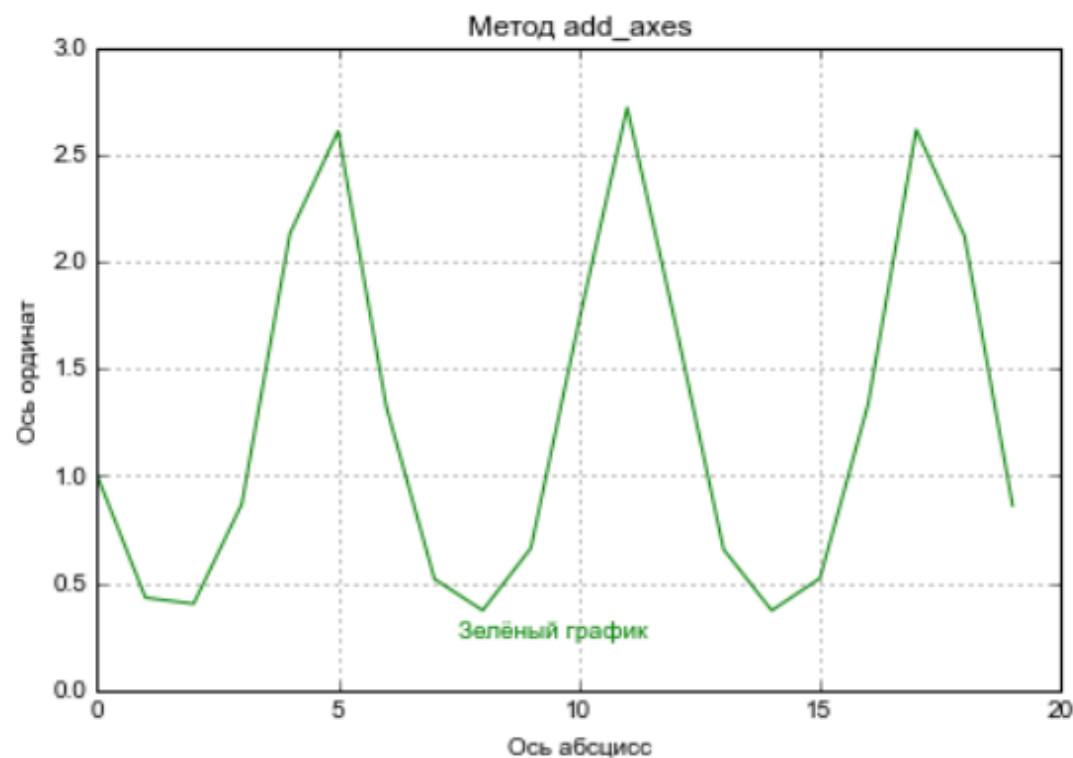
- `import matplotlib.pyplot as plt` `import numpy as np`
- `x = [0,1,2,3,4,5,6,7,8,9,10]`
- `y = [-1,-2,-3,-4,-5,-6,-7,-8,-9,-10,-11]`
- `fig = plt.figure()`
- `ax = fig.add_subplot(211)`
- `line = ax.plot(x, y, '-', color='blue', linewidth=2)`
- `ax2 = fig.add_subplot(212)`
- `n, bins, rectangles = ax2.hist(np.random.randn(1000), 50, facecolor='yellow')`
- `for ax in fig.axes:`
 - `ax.grid(True)`
- `plt.show()`



Триплет: число ячеек по вертикали,
число ячеек по горизонтали,
номер ячейки

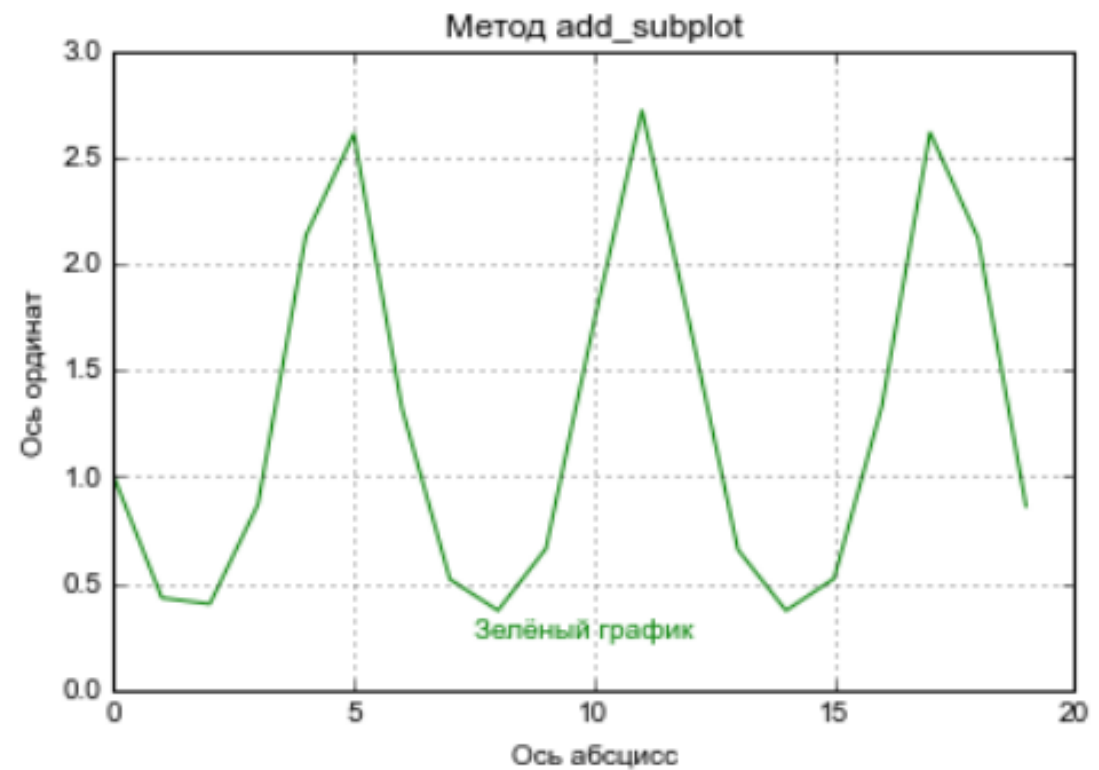
ОБЛАСТЬ РИСОВАНИЯ

- `import matplotlib.pyplot as plt`
- `fig = plt.figure()`
- `x = np.arange(20)`
- `y = np.exp(-np.sin(x))`
- `x0 = 0.05`
- `y0 = 0.05`
- `dx = 0.9`
- `dy = 0.9`
- `rect = [x0, y0, dx, dy]`
- `ax = fig.add_axes(rect)`
- `ax.plot(x, y, 'g')`
- `ax.text(7.5, 0.25, u'Зелёный график', color='g')`
- `ax.grid(True)`
- `ax.set_title(u'Метод add_axes')`
- `ax.set_xlabel(u'Ось абсцисс')`
- `ax.set_ylabel(u'Ось ординат')`
- `plt.show()`



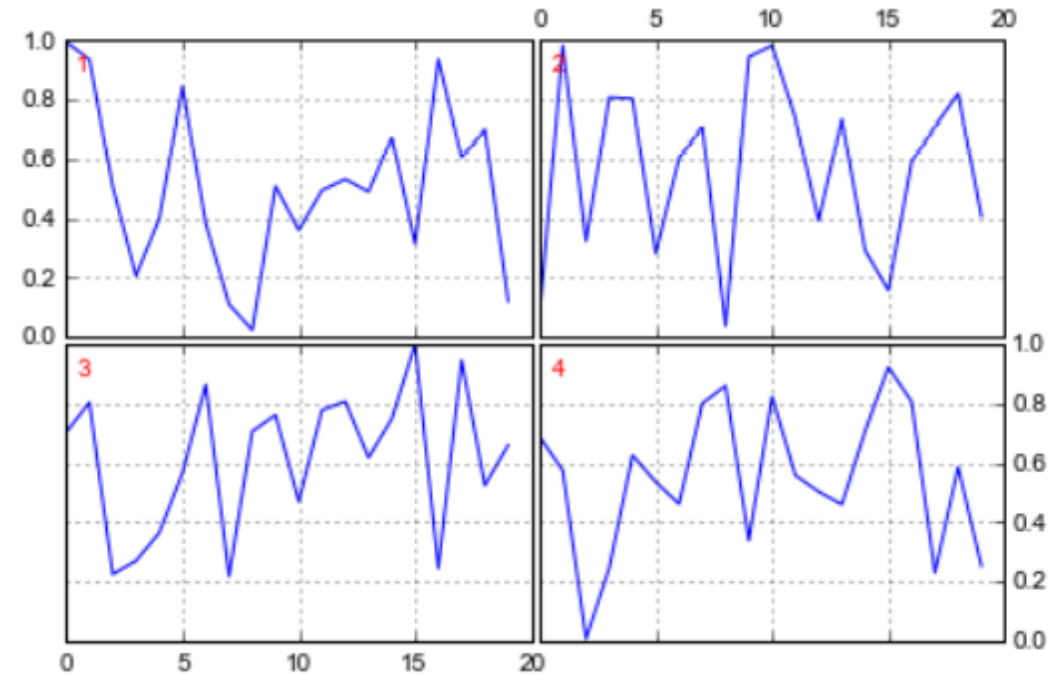
ОБЛАСТЬ ДЛЯ РИСОВАНИЯ

- `fig = plt.figure()`
- `x = np.arange(20)`
- `y = np.exp(-np.sin(x))`
- `ax = fig.add_subplot(111)`
- `ax.plot(x, y, 'g')`
- `ax.text(7.5, 0.25, u'Зелёный график', color='g')`
- `ax.grid(True)`
- `plt.show()`
- `ax.set_title(u'Метод add_subplot')`
- `ax.set_xlabel(u'Ось абсцисс')`
- `ax.set_ylabel(u'Ось ординат')`
- `plt.show()`



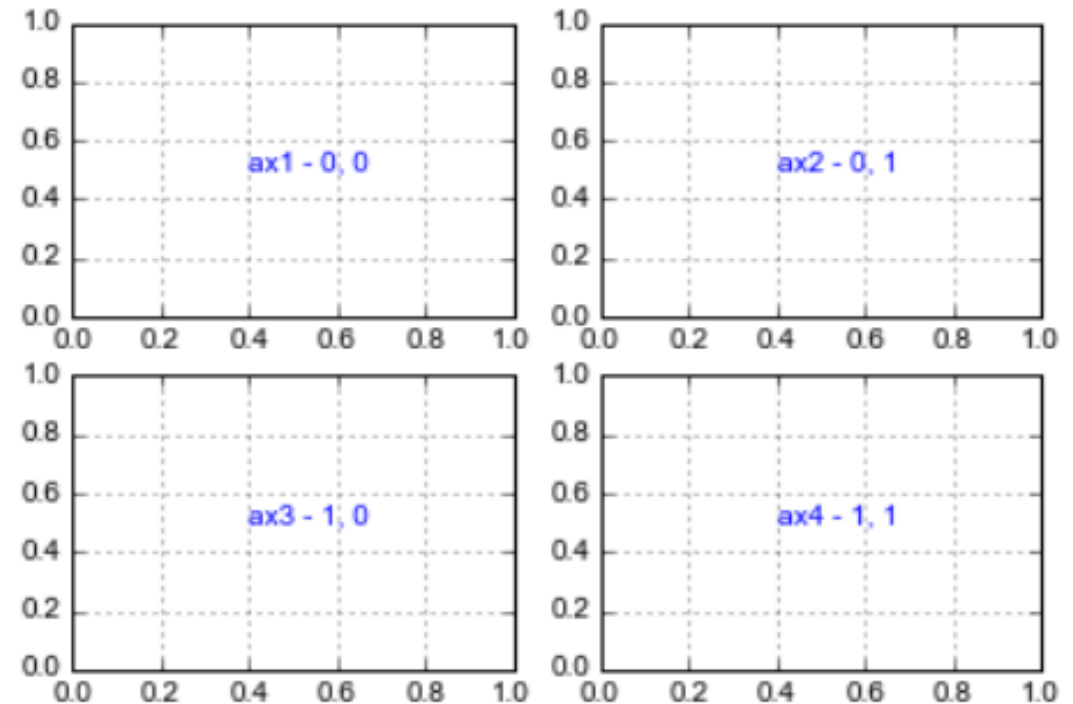
МУЛЬТИОКНА

- `fig, subplots = plt.subplots(nrows=2, ncols=2, sharex=True, sharey=True)`
- `x = np.arange(20)`
- `i = -1`
- `for ax in fig.axes:`
 - `i += 1`
 - `y = np.random.rand(np.size(x))`
 - `ax.grid(True)`
 - `ax.text(0.5, 0.9, str(i+1), color='red')`
 - `ax.plot(x, y)`
- `plt.show()`



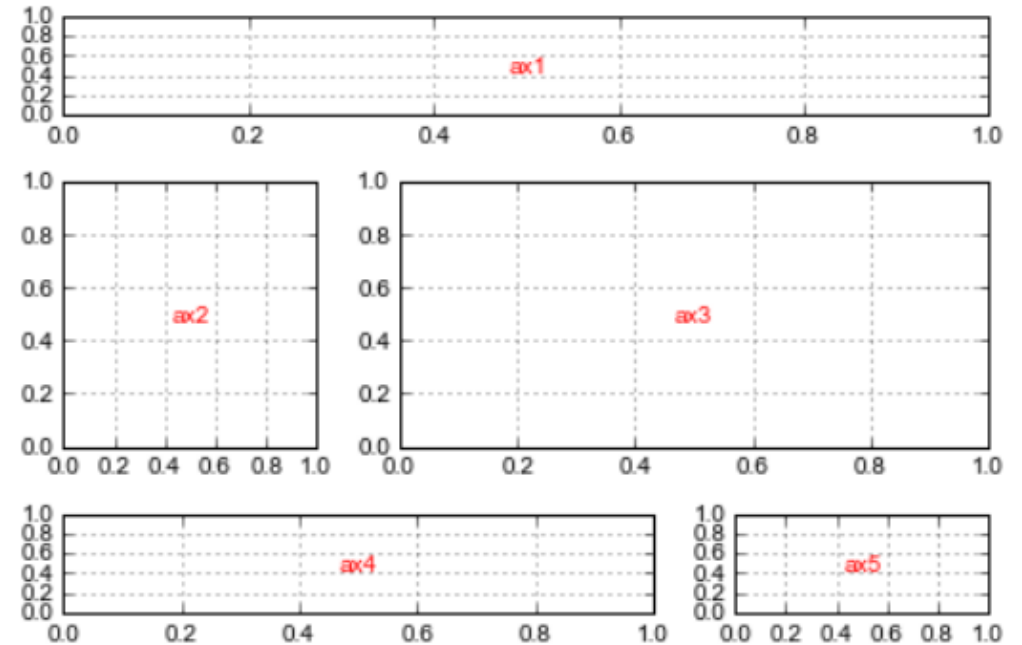
МУЛЬТИОКНА

- `import matplotlib.pyplot as plt`
- `fig = plt.figure()`
- `ax1 = plt.subplot2grid((2,2), (0, 0))`
- `ax2 = plt.subplot2grid((2,2), (0, 1))`
- `ax3 = plt.subplot2grid((2,2), (1, 0))`
- `ax4 = plt.subplot2grid((2,2), (1, 1))`
- `i = -1`
- `jj = [0, 0, 1, 1]`
- `kk = [0, 1, 0, 1]`
- `for ax in fig.axes:`
 - `i += 1`
 - `stext = 'ax%d - %d, %d' % (i+1, jj[i], kk[i])`
 - `ax.text(0.4, 0.5, stext, color='b')`
 - `ax.grid(True)`
- `plt.show()`



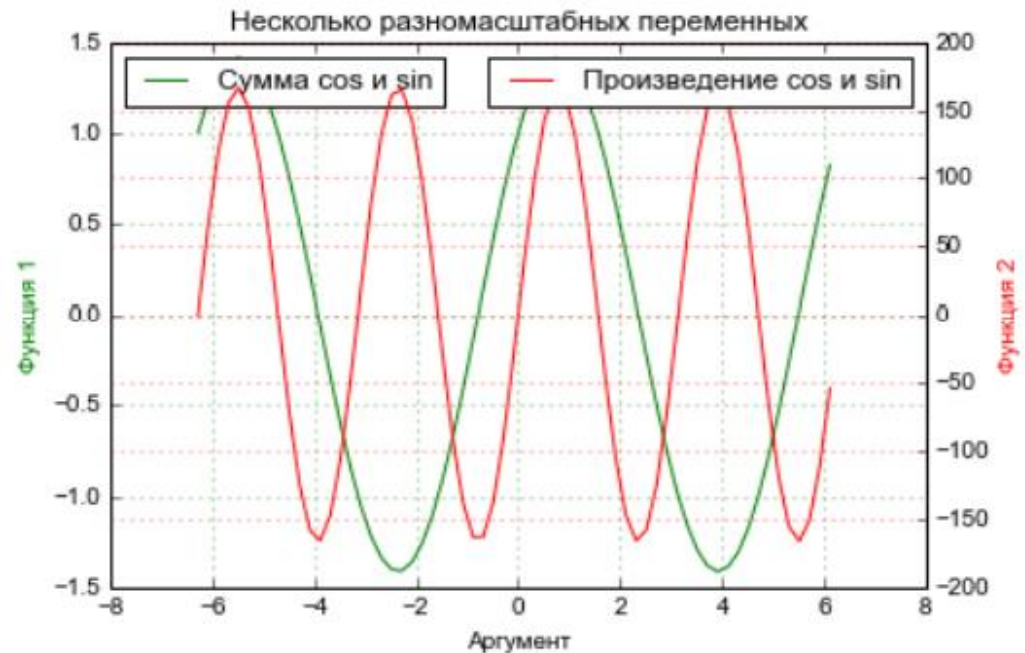
МУЛЬТИОКНА

- `import matplotlib.pyplot as plt`
- `fig = plt.figure()`
- `egrid = (4,3)`
- `ax1 = plt.subplot2grid(egrid, (0, 0), colspan=3)`
- `ax2 = plt.subplot2grid(egrid, (1, 0), rowspan=2)`
- `ax3 = plt.subplot2grid(egrid, (1, 1), rowspan=2, colspan=2)`
- `ax4 = plt.subplot2grid(egrid, (3, 0), colspan=2)`
- `ax5 = plt.subplot2grid(egrid, (3, 2))`
- `for i, ax in enumerate(fig.axes):`
 - `ax.text(0.5, 0.5, "ax%d" % (i+1), va="center", ha="center", color='red', transform=ax.transAxes)`
 - `ax.grid(True)`
- `plt.tight_layout()`
- `plt.show()`



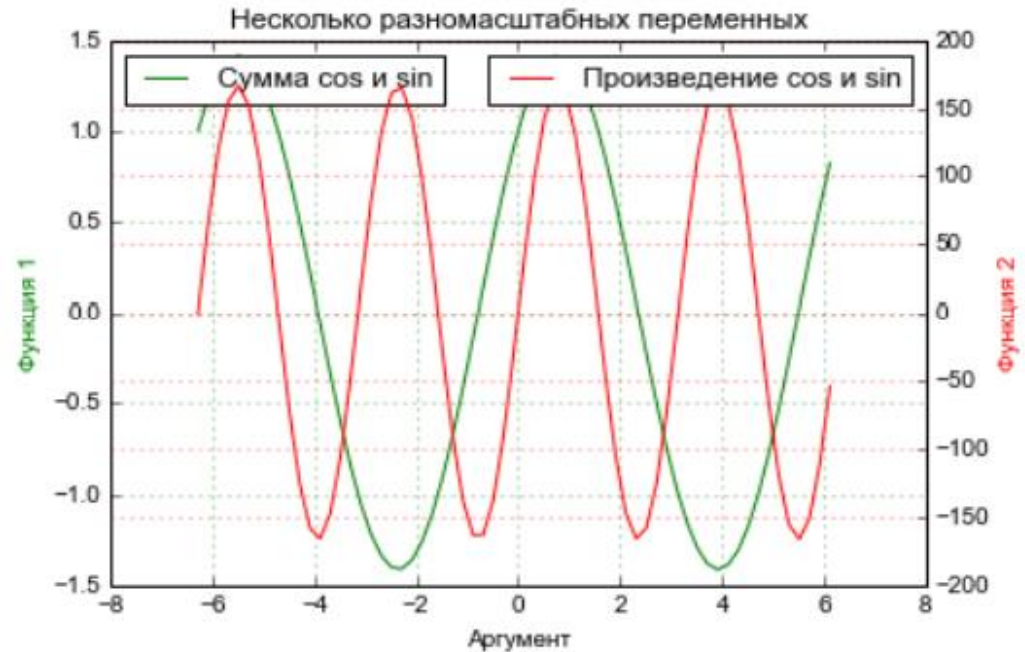
ДОПОЛНИТЕЛЬНАЯ КООРДИНАТНАЯ ОСЬ

- `x = np.arange(-2*np.pi, 2*np.pi, 0.2)`
- `y = np.sin(x) * np.cos(x)`
- `f = np.sin(x) + np.cos(x)`
- `fig = plt.figure()`
- `ax1 = fig.add_subplot(111)`
- `ax2 = ax1.twinx()`
- `line1 = ax1.plot(x, f, label = u'Сумма cos и sin', color='green')`
- `ax1.set_xlabel(u'Аргумент')`
- `ax1.set_ylabel(u'Функция 1', color='green')`
- `ax1.grid(True, color='green')`
- `ax1.tick_params(axis='y', which='major', labelcolor='green')`
- `ax1.legend(loc=2)`



ДОПОЛНИТЕЛЬНАЯ КООРДИНАТНАЯ ОСЬ

- `line2 = ax2.plot(x, y*333, label = u'Произведение cos и sin', color='red')`
- `ax2.set_ylabel(u'Функция 2', color='red')`
- `ax2.grid(True, color='red')`
- `ax2.tick_params(axis='y', which='major', labelcolor='red')`
- `ax2.legend(loc=1)`
- `ax1.set_title(u'Несколько разномасштабных переменных')`
- `plt.title(u'Несколько разномасштабных переменных')`
- `plt.show()`



ПРИМЕР

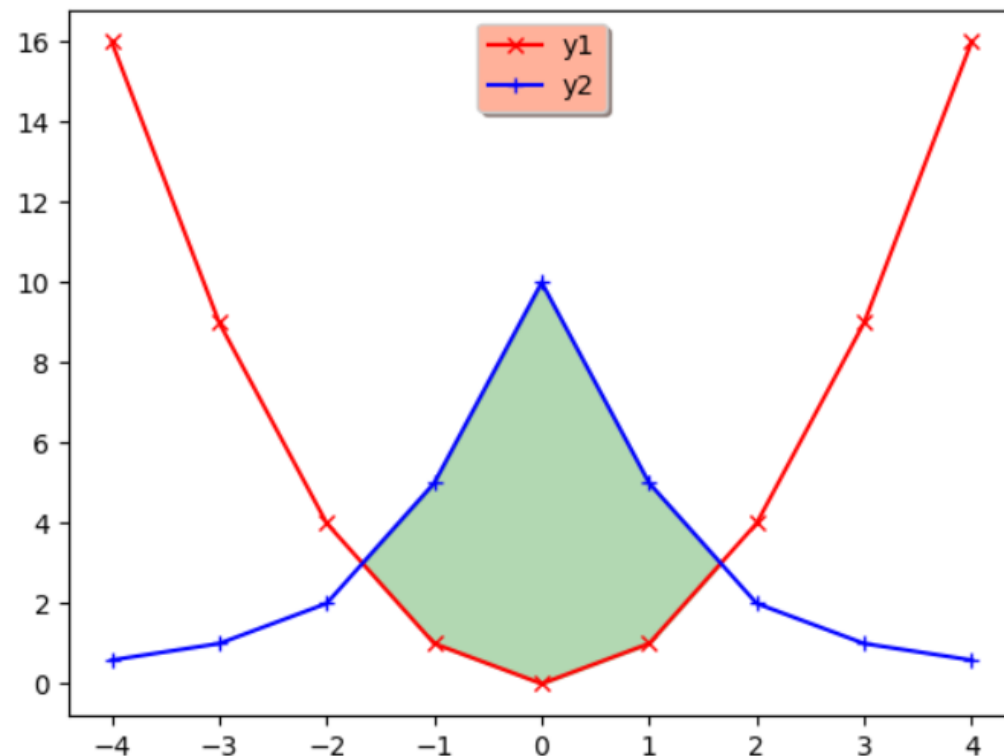
```
import numpy as np
import matplotlib.pyplot as plt

plt.ioff()
x = np.arange(-4, 5)
y1 = x ** 2
y2 = 10 / (x ** 2 + 1)
fig, ax = plt.subplots()

ax.plot(x, y1, 'rx', x, y2, 'b+', linestyle='solid')
ax.fill_between(x, y1, y2, where=y2>y1, interpolate=True,
               color='green', alpha=0.3)

lgnd = ax.legend(['y1', 'y2'], loc='upper center', shadow=True)
lgnd.get_frame().set_facecolor('#ffb19a')

plt.show()
```



ПРИМЕР

```
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()

ax1.set_title(
    'Home value as a function of home age & area population',
    fontsize=14
)

sctr = ax1.scatter(x=age, y=pop, c=y, cmap='RdYlGn')
plt.colorbar(sctr, ax=ax1, format='${:d}')
ax1.set_yscale('log')
ax2.hist(age, bins='auto')
ax3.hist(pop, bins='auto', log=True)

add_titlebox(ax2, 'Histogram: home age')
add_titlebox(ax3, 'Histogram: area population (log scl.)')
```

