



2.7 Форматирование графиков

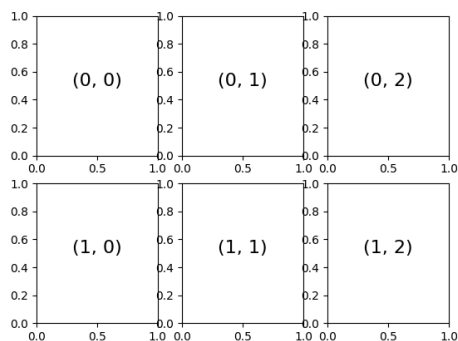
Простые сетки

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt

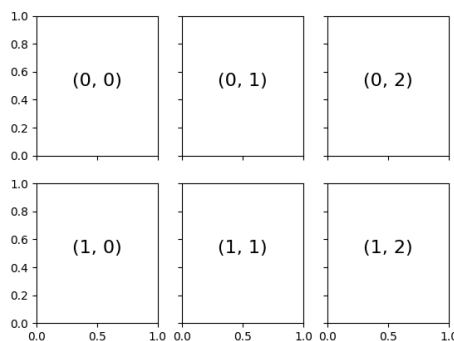
fig, ax = plt.subplots(2, 3, sharex="col", sharey="row")

for i in range(2):
    for j in range(3):
        ax[i, j].text(0.5, 0.5, str((i, j)), fontsize=16, ha="center")

plt.savefig("img.png")
plt.show()
```



fig, ax = plt.subplots(2, 3)



fig, ax = plt.subplots(2, 3, sharex="col", sharey="row")

параметр **sharex** в функции `plt.subplots()` используется для указания того, должны ли подграфики (axes) делить одну и ту же ось X или Y. Когда **sharex="col"** или **sharey="row"**, это означает, что все подграфики в одном столбце (строке) будут иметь общую ось X или Y.

Размещение областей графиков (GridSpec)

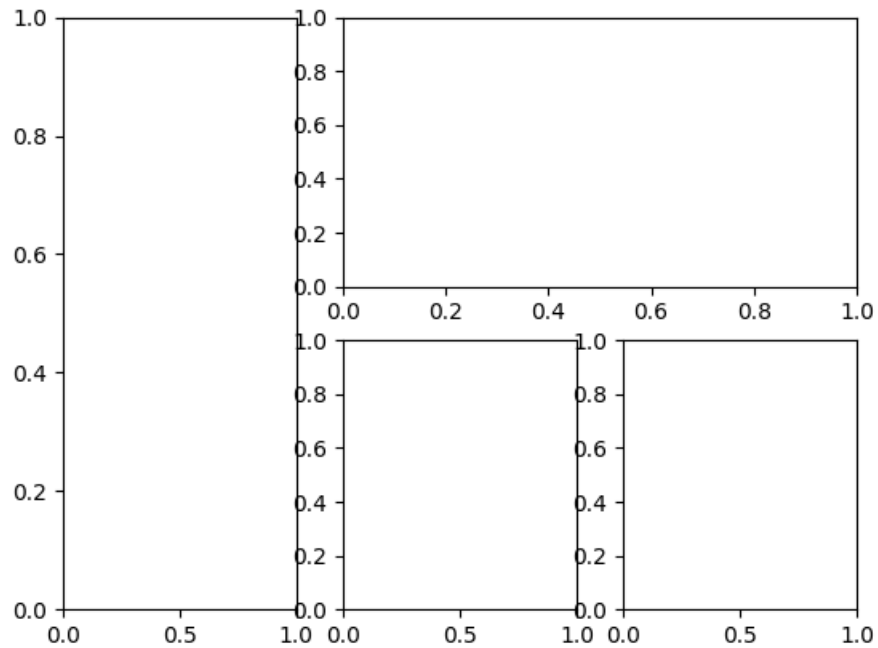
```
import numpy as np
import pandas as pd
```

```
import matplotlib as mpl
import matplotlib.pyplot as plt

grid = plt.GridSpec(2, 3)

plt.subplot(grid[:2, 0])
plt.subplot(grid[0, 1:])
plt.subplot(grid[1, 1])
plt.subplot(grid[1, 2])

plt.savefig("img.png")
plt.show()
```



Практический пример

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt

mean = [0, 0]
cov = [[1, 1], [1, 2]]
```

```

rng = np.random.default_rng(1)
x, y = rng.multivariate_normal(mean, cov, 3000).T

fig = plt.figure()
grid = plt.GridSpec(3, 4, hspace=0.5, wspace=0.5)

main_ax = fig.add_subplot(grid[:-1, 1:])

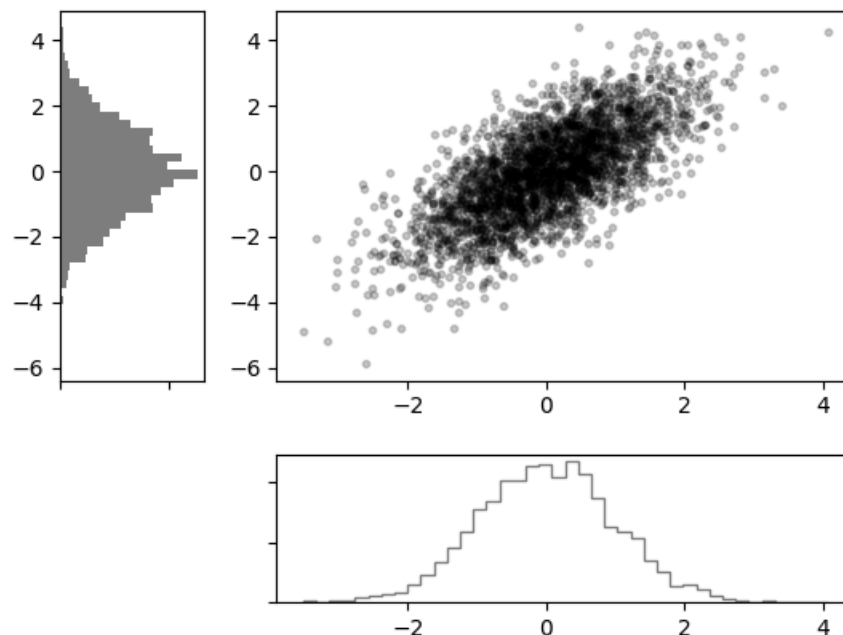
y_hist = fig.add_subplot(grid[:-1, 0], xticklabels=[], sharey=main_ax)
x_hist = fig.add_subplot(grid[-1, 1:], yticklabels=[], sharex=main_ax)

main_ax.plot(x, y, "ok", markersize=3, alpha=0.2)

y_hist.hist(y, 40, orientation="horizontal", color="gray", histtype="stepfilled")
x_hist.hist(x, 40, orientation="vertical", color="gray", histtype="step")

plt.savefig("img.png")
plt.show()

```



Позиционирование (поясняющие надписи)

```

import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from datetime import datetime

births = pd.read_csv("./data/births-1969.csv")

births.index = pd.to_datetime(10000 * births.year + 100 * births.month + births.day, \
    format="%Y%m%d")
print(births.head())

births_by_date = births.pivot_table("births", [births.index.month, births.index.day])

births_by_date.index = [
    datetime(1969, month, day) for (month, day) in births_by_date.index
]

fig, ax = plt.subplots()
births_by_date.plot(ax=ax)

style = dict(size=10, color="gray")

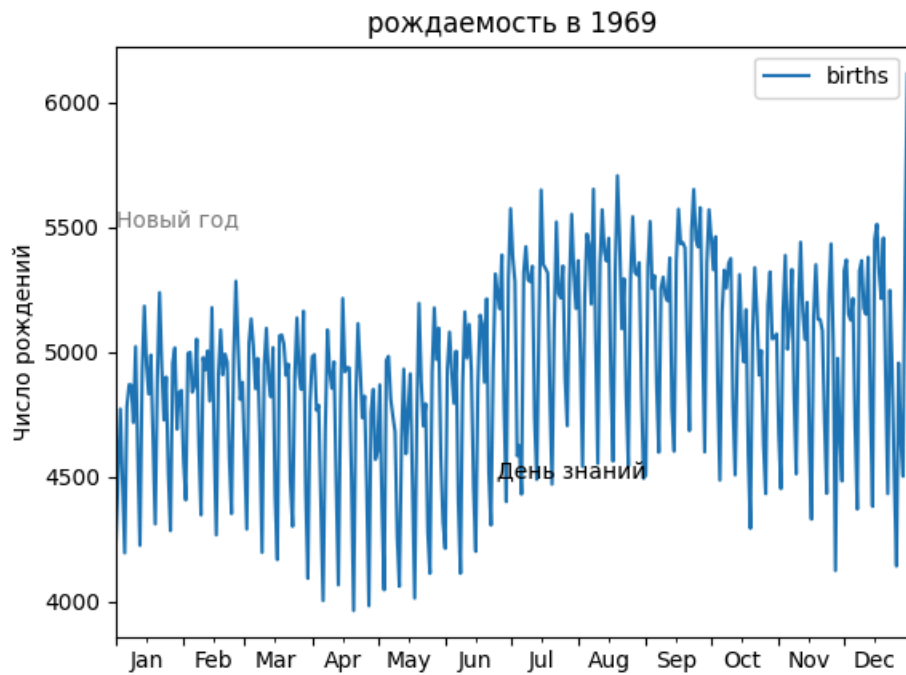
ax.text('1969-01-01', 5500, "Новый год", **style)
ax.text('1969-09-01', 4500, "День знаний", ha="right")

ax.set(title="рождаемость в 1969", ylabel="Число рождений")

ax.xaxis.set_major_formatter(plt.NullFormatter())
ax.xaxis.set_minor_formatter(mpl.dates.DateFormatter("%h"))
ax.xaxis.set_minor_locator(mpl.dates.MonthLocator(bymonthday=15))

plt.savefig("img.png")
plt.show()

```



```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from datetime import datetime

fig = plt.figure()

ax1 = plt.axes()
ax2 = plt.axes([0.4, 0.3, 0.1, 0.2])

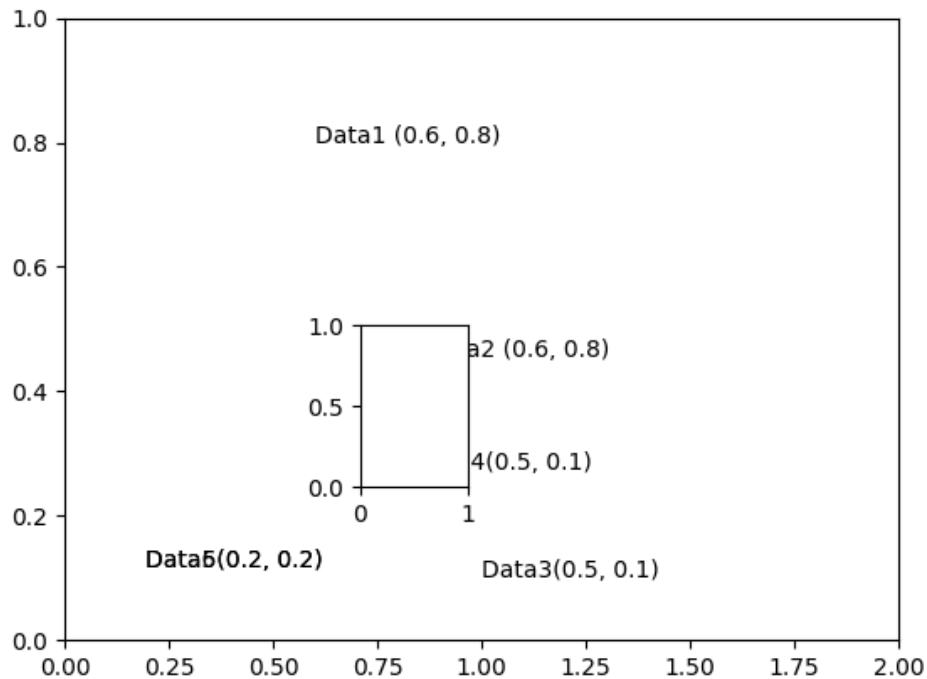
ax1.set_xlim(0, 2)

ax1.text(0.6, 0.8, "Data1 (0.6, 0.8)", transform=ax1.transData)
ax1.text(0.6, 0.8, "Data2 (0.6, 0.8)", transform=ax2.transData)

ax1.text(0.5, 0.1, "Data3(0.5, 0.1)", transform=ax1.transAxes)
ax1.text(0.5, 0.1, "Data4(0.5, 0.1)", transform=ax2.transAxes)

ax1.text(0.2, 0.2, "Data5(0.2, 0.2)", transform=fig.transFigure)
ax1.text(0.2, 0.2, "Data6(0.2, 0.2)", transform=fig.transFigure)
```

```
plt.savefig("img.png")
plt.show()
```



Указатель (стрелочка) на графике

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from datetime import datetime

fig, ax = plt.subplots()

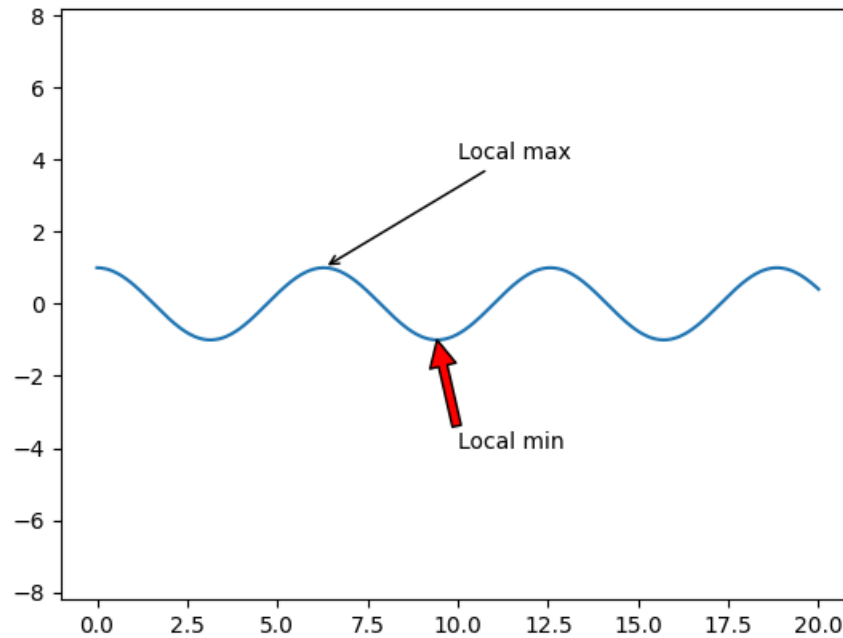
x = np.linspace(0, 20, 1000)

ax.plot(x, np.cos(x))
ax.axis("equal")

ax.annotate("Local max", xy=(np.pi * 2, 1), xytext=(10, 4), \
           arrowprops=dict(facecolor="red", arrowstyle="→"))
```

```
ax.annotate("Local min", xy=(np.pi * 3, -1), xytext=(10, -4), \
           arrowprops=dict(facecolor="red"))

plt.savefig("img.png")
plt.show()
```



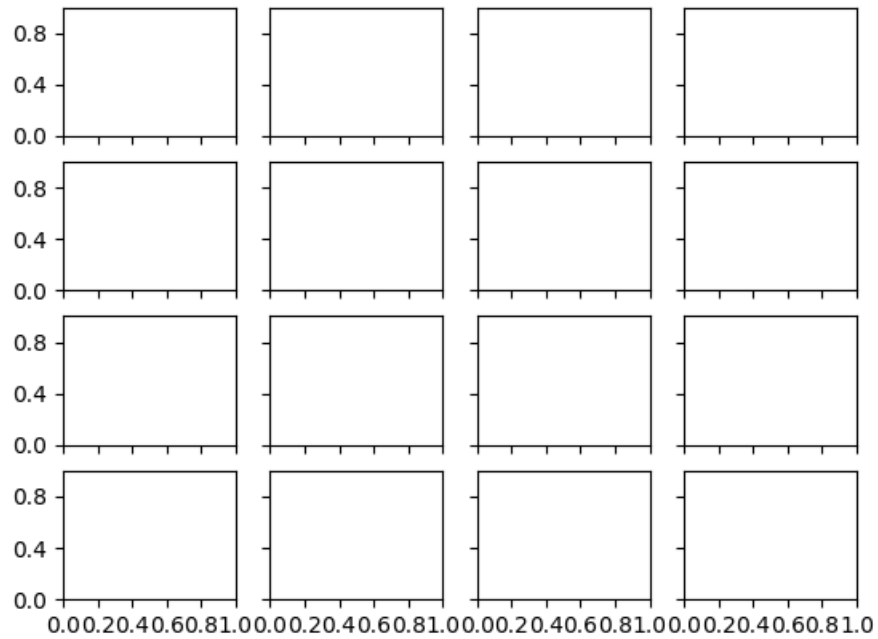
Количество делений на графике

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from datetime import datetime

fig, ax = plt.subplots(4, 4, sharex=True, sharey=True)

for axi in ax.flat:
    axi.xaxis.set_major_locator(plt.MaxNLocator(5))
    axi.yaxis.set_major_locator(plt.MaxNLocator(3))
```

```
plt.savefig("img.png")
plt.show()
```



Конфигурация графика

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from datetime import datetime

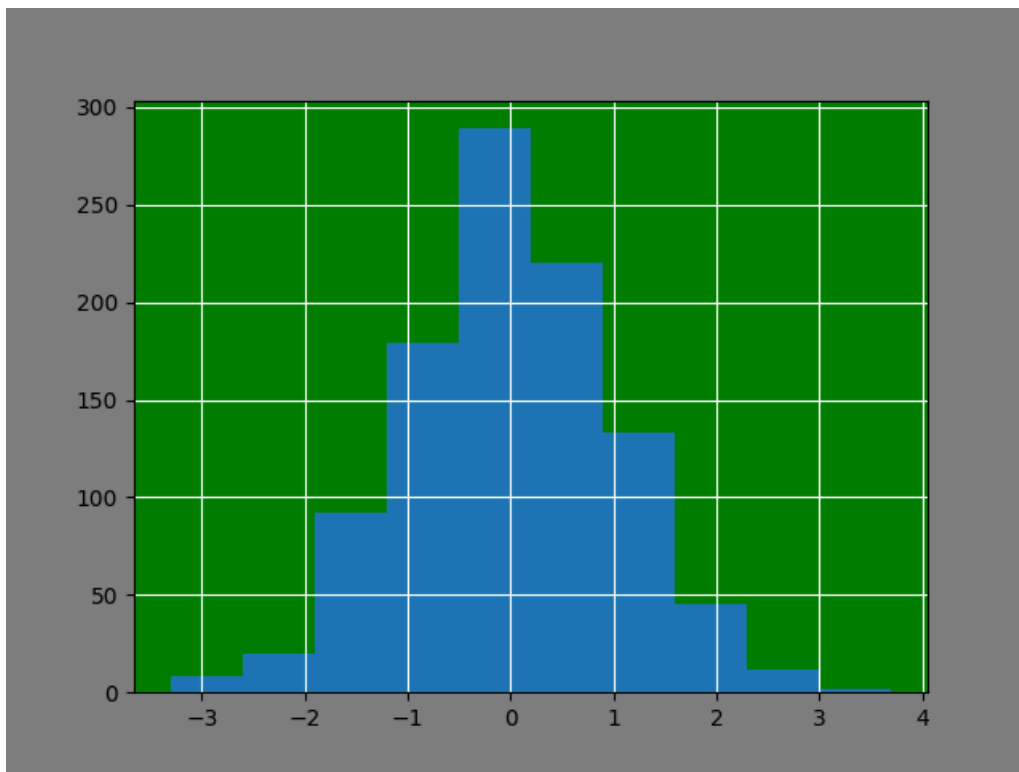
x = np.random.randn(1000)

fig = plt.figure(facecolor="gray")
ax = plt.axes(facecolor="green")
plt.grid(color="white", linestyle="solid")
ax.xaxis.tick_bottom()

plt.hist(x)
```



```
plt.savefig("img.png")  
plt.show()
```



.matplotlibrc

