

Мухаметшин А.Р. БМО-01-25

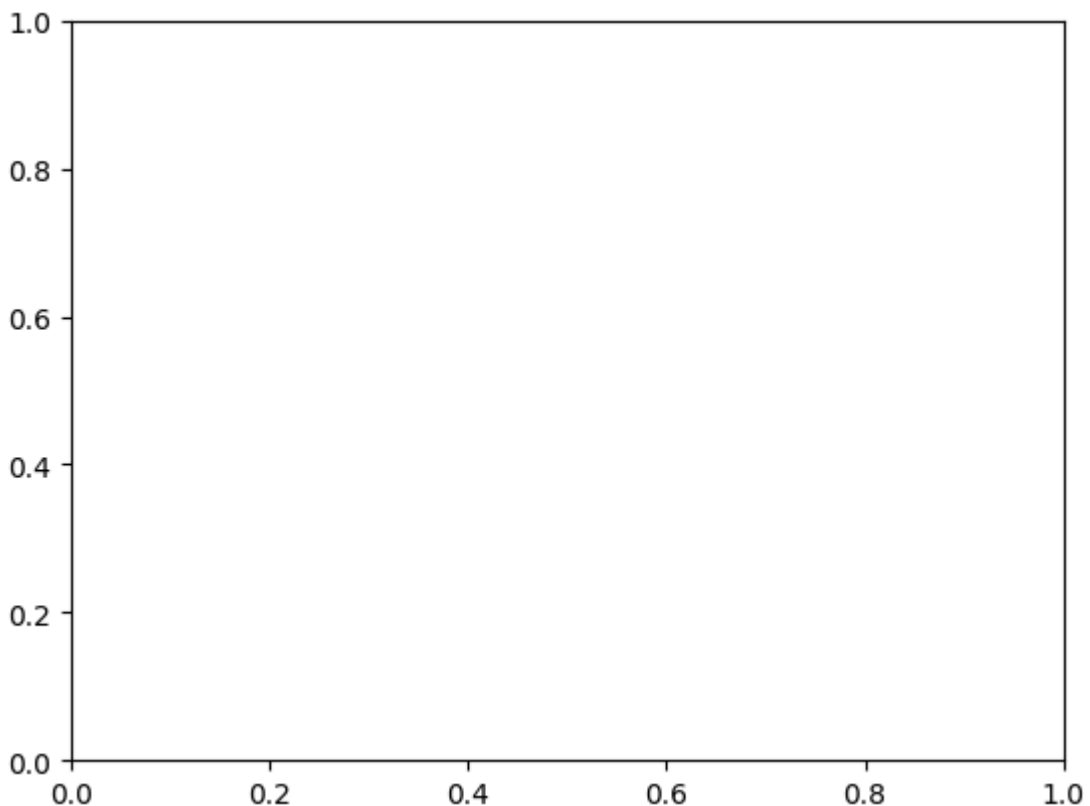
```
!git clone https://github.com/neuralcomputer/ML_School.git
```

```
Cloning into 'ML_School'...
remote: Enumerating objects: 94, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 94 (delta 5), reused 0 (delta 0), pack-reused 79 (from 1)
Receiving objects: 100% (94/94), 33.83 MiB | 24.66 MiB/s, done.
Resolving deltas: 100% (29/29), done.
```

```
import matplotlib.pyplot as plt # подключаем библиотеку
import numpy as np #
np.random.seed(333) # генератор случайных чисел установим на произвольное, но с
```

```
fig, _ = plt.subplots() # создадим полотно
print(type(fig)) # что это за тип?
print(dir(fig.axes[0])) # сколько у него разных свойств и методов....
```

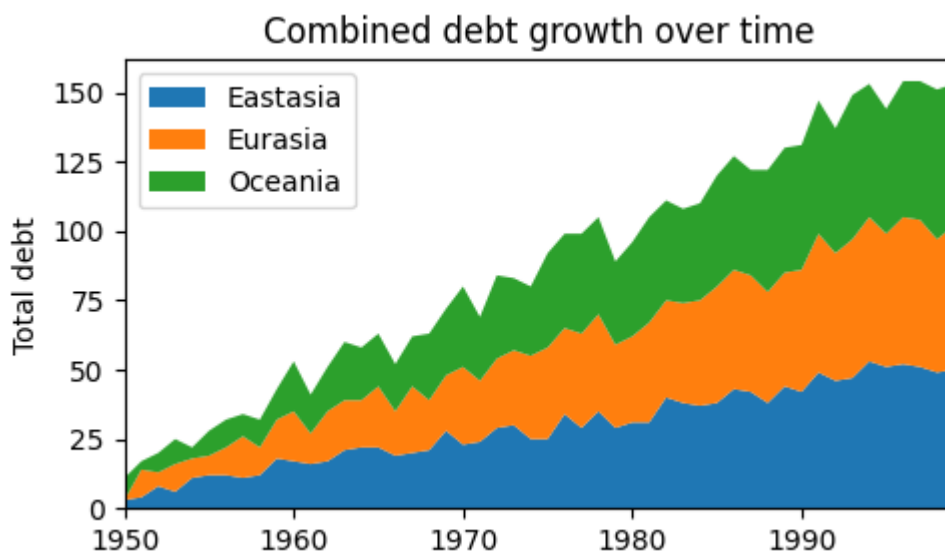
```
<class 'matplotlib.figure.Figure'>
['ArtistList', '_AxesBase__clear', '_PROPERTIES_EXCLUDED_FROM_SET', '__annotatio
```



```
one_tick = fig.axes[0].yaxis.get_major_ticks()[1].label1 # посмотрим на одно из
print(one_tick) #
```

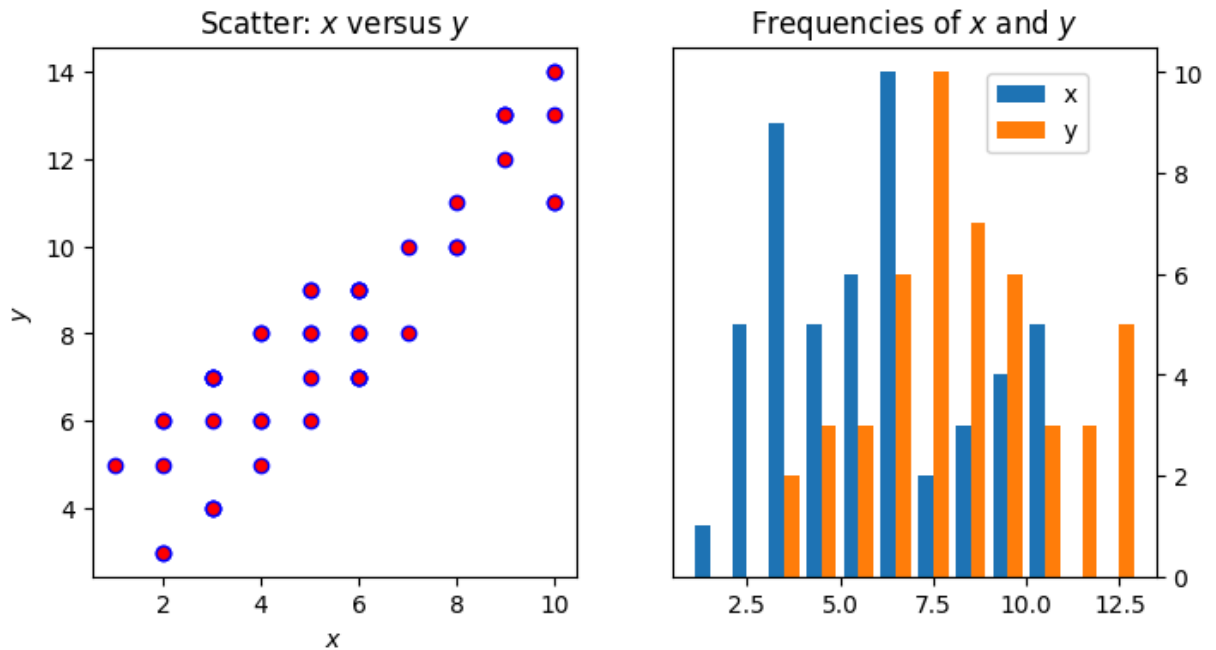
```
Text(0, 0.2, '0.2')
```

```
import numpy as np #
import matplotlib.pyplot as plt #
# задаем данные
rng = np.arange(50) #
rnd = np.random.randint(0, 10, size=(3, rng.size)) #
yrs = 1950 + rng #
fig, ax = plt.subplots(figsize=(5, 3)) # создаем полотно заданного размера
ax.stackplot(yrs, rng + rnd, labels=['Eastasia', 'Eurasia', 'Oceania']) # рисуем
ax.set_title('Combined debt growth over time') # название всего графика
ax.legend(loc='upper left') # отображаем подпись графиков, указываем его расположение
plt.show() # выводим на экран все полотна
ax.set_ylabel('Total debt') # название оси OY
ax.set_xlim(xmin=yrs[0], xmax=yrs[-1]) # диапазон отображения для оси OX
fig.tight_layout() # более компактное расположение элементов графиков
plt.show() # выводим на экран все полотна
```



```
import numpy as np #
import matplotlib.pyplot as plt # подключаем библиотеку
# задаем данные
x = np.random.randint(low=1, high=11, size=50) #
y = x + np.random.randint(1, 5, size=x.size) #
data = np.column_stack((x, y)) #
fig, (ax1, ax2) = plt.subplots( #делаем полотно с несколькими графиками
    nrows=1, ncols=2, # расположенными в один ряд два столбца
    figsize=(8, 4) # размер полотна
) #
ax1.scatter(x=x, y=y, marker='o', c='r', edgecolor='b') # строим первый график
ax1.set_title('Scatter: $x$ versus $y$') # название первого графика
ax1.set_xlabel('$x$') # название горизонтальной оси первого графика, используем
ax1.set_ylabel('$y$') # название вертикальной оси первого графика, используем к
ax2.hist( # строим второй график - гистограмму
    data, bins=np.arange(data.min(), data.max()), # задаем данные и диапазоны
    label=('x', 'y') # и подписи данных
)
ax2.legend(loc=(0.65, 0.8)) # отображаем легенду и задаем ее расположение
```

```
ax2.set_title('Frequencies of $x$ and $y$') # название второго графика
ax2.yaxis.tick_right() # устанавливаем, что вертикальная ось должна быть справа
plt.show() # отображаем полотно
```



```
import ssl # проблемы безопасности
# выполним, чтобы получить разрешение на загрузку
ssl._create_default_https_context = ssl._create_unverified_context
from io import BytesIO # для более удобной работы с данными как-будто они файл
import tarfile # для работы с архивами
from urllib.request import urlopen # для доступа к данным по адресу
url = 'http://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.tgz' # адрес, где
b = BytesIO(urlopen(url).read()) # читаем данные и делаем интерфейс для более у
fpath = 'CaliforniaHousing/cal_housing.data' # путь и название нужного файла в
with tarfile.open(mode='r', fileobj=b) as archive: # открываем загруженный архив
    housing = np.loadtxt(archive.extractfile(fpath), delimiter=',') # извлекаем н
```

```
y = housing[:, -1] #последний столбец
pop, age = housing[:, [4, 7]].T # 5 и 8 столбцы в отдельные переменные (мы же г
#pop = housing[:, 4]
#age = housing[:, 7]
```

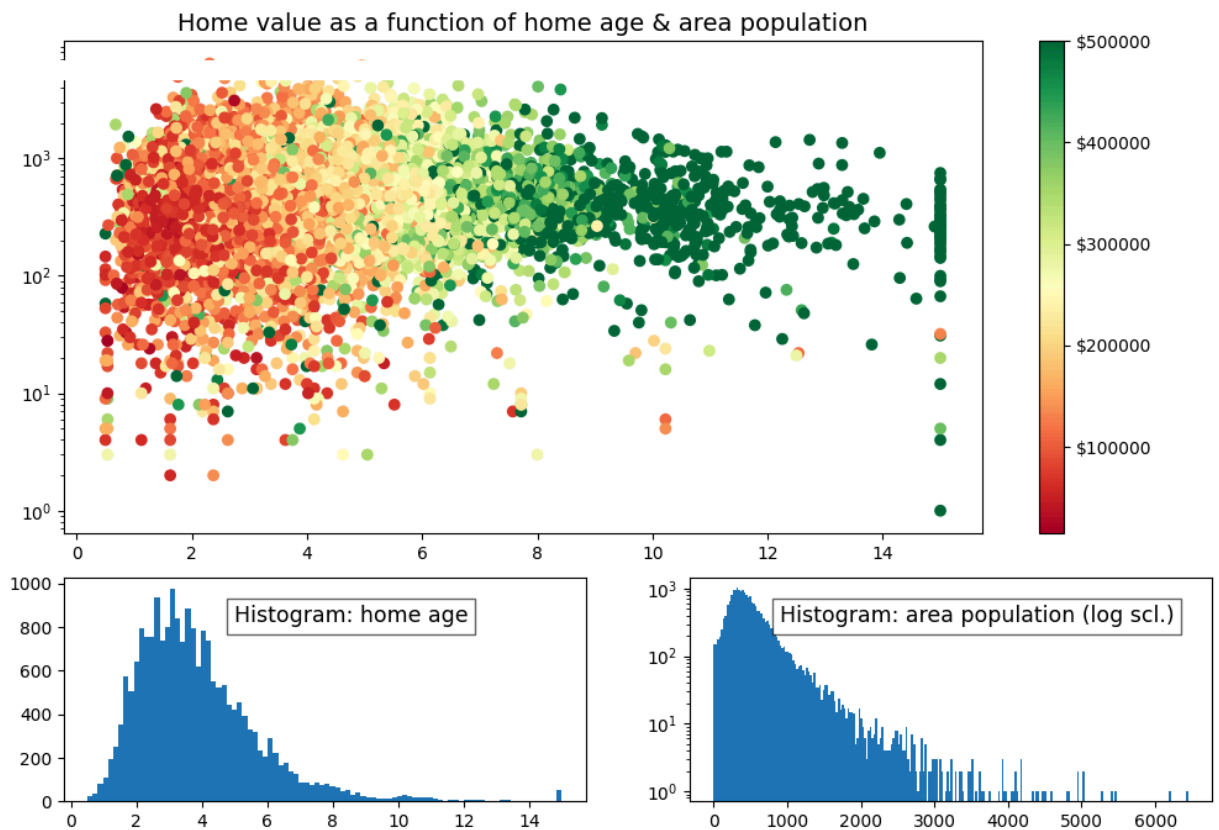
```
def add_titlebox(ax, text): # создаем свою вспомогательную функцию для заданног
    ax.text(.55, .8, text, # она располагает текстовое окно
            horizontalalignment='center', # с расположением текста по центру
            transform=ax.transAxes, # "оси" для текста
            bbox=dict(facecolor='white', alpha=0.6), # очерчивание текста прямоугольнико
            fontsize=12.5) # размер шрифта
    return ax # возвращаем измененный объект графика
```

```
import matplotlib.pyplot as plt #
gridsize = (3, 2) # размер сетки
```

```

fig = plt.figure(figsize=(12, 8)) # создаем полотно
ax1 = plt.subplot2grid(gridsize, (0, 0), colspan=2, rowspan=2) # первый график
ax2 = plt.subplot2grid(gridsize, (2, 0)) # а эти по одной
ax3 = plt.subplot2grid(gridsize, (2, 1)) #
ax1.set_title( # название первого графика
    'Home value as a function of home age & area population',
    fontsize=14 # размер шрифта
)
sctr = ax1.scatter(x=age, y=pop, c=y, cmap='RdYlGn') # строим дискретное распре
plt.colorbar(sctr, ax=ax1, format='$%d') # отображаем цветовую карту со шкалой
ax1.set_yscale('log') # логарифмический масштаб
ax2.hist(age, bins='auto') # второй график - гистограмма возрастов
ax3.hist(pop, bins='auto', log=True) # третий график - гистограмма населенности
add_titlebox(ax2, 'Histogram: home age') # название второго графика
add_titlebox(ax3, 'Histogram: area population (log scl.)') # название третьего
plt.show() #

```



```

import numpy as np #
x = np.diag(np.arange(2, 12))[:-1] #
x[np.diag_indices_from(x[:-1])] = np.arange(2, 12) #
x2 = np.arange(x.size).reshape(x.shape) #
print(x, '\n\n', x2) #

```

```

[[ 2  0  0  0  0  0  0  0  0 11]
 [ 0  3  0  0  0  0  0  0 10  0]
 [ 0  0  4  0  0  0  0  9  0  0]
 [ 0  0  0  5  0  0  8  0  0  0]
 [ 0  0  0  0  6  7  0  0  0  0]
 [ 0  0  0  0  6  7  0  0  0  0]
 [ 0  0  0  5  0  0  8  0  0  0]
 [ 0  0  4  0  0  0  0  9  0  0]
 [ 0  3  0  0  0  0  0  0 10  0]

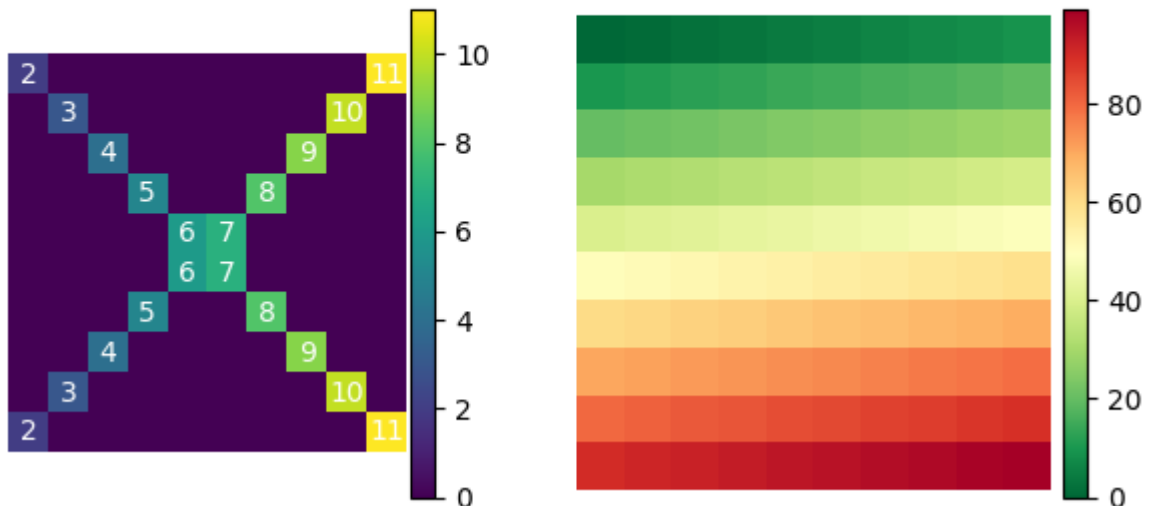
```

```
[ 2  0  0  0  0  0  0  0  0  0 11]]
```

```
[[ 0  1  2  3  4  5  6  7  8  9]
 [10 11 12 13 14 15 16 17 18 19]
 [20 21 22 23 24 25 26 27 28 29]
 [30 31 32 33 34 35 36 37 38 39]
 [40 41 42 43 44 45 46 47 48 49]
 [50 51 52 53 54 55 56 57 58 59]
 [60 61 62 63 64 65 66 67 68 69]
 [70 71 72 73 74 75 76 77 78 79]
 [80 81 82 83 84 85 86 87 88 89]
 [90 91 92 93 94 95 96 97 98 99]]
```

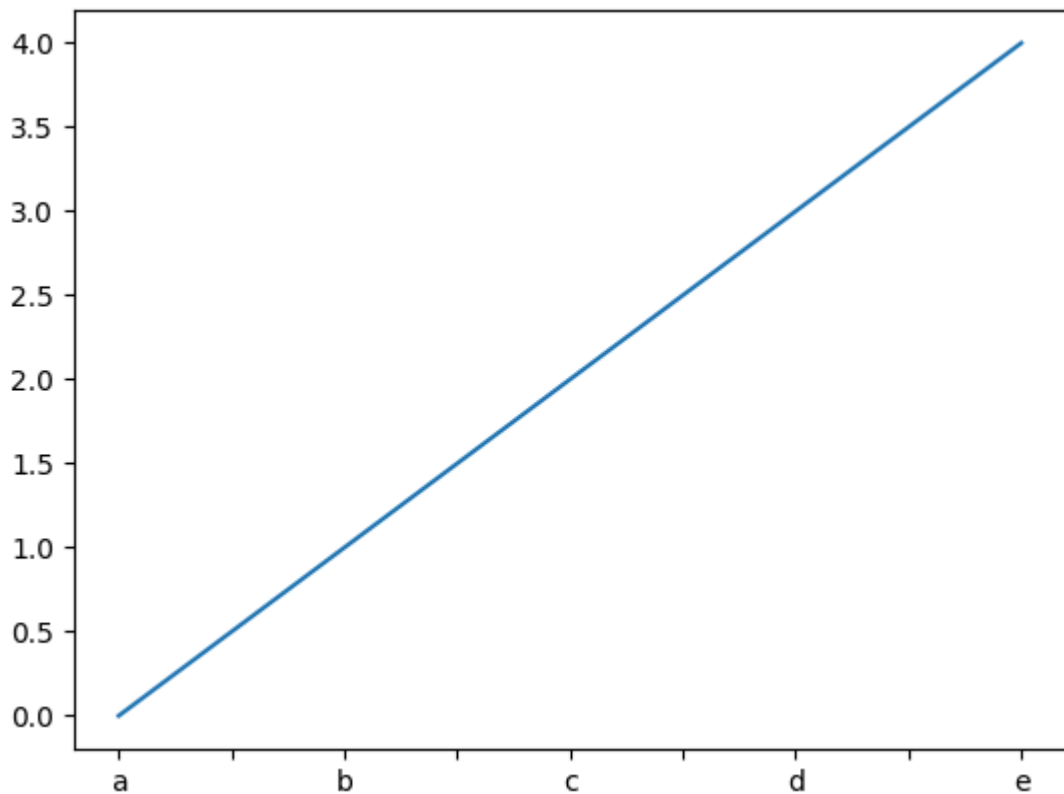
```
import numpy as np #
import matplotlib.pyplot as plt #
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8, 4)) #
img1=ax1.matshow(x) #
ax1.axis('off')
img2 = ax2.matshow(x2, cmap='RdYlGn_r') #
ax2.axis('off')
for i, j in zip(*x.nonzero()): #
    ax1.text(j, i, x[i, j], color='white', ha='center', va='center') #
plt.colorbar(img2, ax=[ax1,ax2], shrink=0.79, pad=0.01)
plt.colorbar(img1, ax=[ax1], shrink=0.79, pad=0.01)
fig.suptitle('Heatmaps with `Axes.matshow`', fontsize=16) #
plt.show() #
```

Heatmaps with `Axes.matshow`



```
import pandas as pd # подключим pandas
s = pd.Series(np.arange(5), index=list('abcde')) # сделаем какую-нибудь последс
ax = s.plot() # отобразим его. Это метод из pandas, а не из matplotlib
print(type(ax)) # Но тип его это <matplotlib.axes._subplots.AxesSubplot at 0x12
print(id(plt.gca()) == id(ax)) # Такой же как у графиков из matplotlib: True
```

```
<class 'matplotlib.axes._axes.Axes'>
True
```



```
import numpy as np # numpy пригодится
import matplotlib.pyplot as plt # графики строить
import pandas as pd # pandas для данных

url = 'https://fred.stlouisfed.org/graph/fredgraph.csv?id=VIXCLS' # адрес, откуда брать данные
vix = pd.read_csv(url, # адрес
                  index_col=0, # номер столбца с датами
                  parse_dates=True) # парсить даты

# Выводим первые несколько строк данных
print(vix.head())

# Строим график
plt.figure(figsize=(12, 6))
plt.plot(vix.index, vix['VIXCLS'], label='VIX', color='blue')
plt.title('Индекс волатильности VIX')
plt.xlabel('Дата')
plt.ylabel('VIX')
plt.legend()
plt.grid()
plt.show()
```

```
VIXCLS
observation_date
1990-01-02      17.24
1990-01-03      18.19
1990-01-04      19.22
1990-01-05      20.11
1990-01-08      20.26
```

