

Задание 1.

Исходные данные:

Загрузите модуль pyplot библиотеки matplotlib с псевдонимом plt, а также библиотеку numpy с псевдонимом np.

Примените магическую функцию %matplotlib inline для отображения графиков в Jupyter Notebook и настройки конфигурации ноутбука со значением 'svg' для более четкого отображения графиков.

Создайте список под названием x с числами 1, 2, 3, 4, 5, 6, 7 и список y с числами 3.5, 3.8, 4.2, 4.5, 5, 5.5, 7.

С помощью функции plot постройте график, соединяющий линиями точки с горизонтальными координатами из списка x и вертикальными - из списка y.

Затем в следующей ячейке постройте диаграмму рассеяния (другие названия - диаграмма разброса, scatter plot).

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x = [1, 2, 3, 4, 5, 6, 7]
>>> y = [3.5, 3.8, 4.2, 4.5, 5.0, 5.5, 7.0]
>>> plt.plot(x, y)
[<matplotlib.lines.Line2D object at 0x7f0e12be5790>]
>>> plt.show()
>>>
```

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x = [1, 2, 3, 4, 5, 6, 7]
>>> y = [3.5, 3.8, 4.2, 4.5, 5.0, 5.5, 7.0]
>>> plt.scatter(x, y)
<matplotlib.collections.PathCollection object at 0x7f1094c08a60>
>>> plt.show()
>>>
```

Задание 2.

Исходные данные:

С помощью функции linspace из библиотеки Numpy создайте массив t из 51 числа от 0 до 10 включительно.

Создайте массив Numpy под названием f, содержащий косинусы элементов массива t.

Постройте линейную диаграмму, используя массив t для координат по горизонтали, а массив f - для координат по вертикали. Линия графика должна быть зеленого цвета.

Выведите название диаграммы - 'График f(t)'. Также добавьте названия для горизонтальной оси - 'Значения t' и для вертикальной - 'Значения f'.

Ограничьте график по оси x значениями 0.5 и 9.5, а по оси y - значениями -2.5 и 2.5.

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```

>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> t = np.linspace(0, 10, 51)
>>> f = np.cos(t)
>>> plt.plot(t, f, color='g')
[<matplotlib.lines.Line2D object at 0x7fb8f207f400>]
>>> plt.xticks([0.5, 9.5])
([<matplotlib.axis.XTick object at 0x7fb8f2059dc0>, <matplotlib.axis.XTick object at
0x7fb8f2059d90>], <a list of 2 Text xticklabel objects>)
>>> plt.yticks([-2.5, 2.5])
([<matplotlib.axis.YTick object at 0x7fb8f205f730>, <matplotlib.axis.YTick object at
0x7fb8f205f310>], <a list of 2 Text yticklabel objects>)
>>> plt.title('График f(t)')
Text(0.5, 1.0, 'График f(t)')
>>> plt.show()
>>>

```

Задание 3.

Исходные данные:

С помощью функции `linspace` библиотеки `Numpy` создайте массив `x` из 51 числа от -3 до 3 включительно.

Создайте массивы `y1`, `y2`, `y3`, `y4` по следующим формулам:

$$y1 = x^{**2}$$

$$y2 = 2 * x + 0.5$$

$$y3 = -3 * x - 1.5$$

$$y4 = \sin(x)$$

Используя функцию `subplots` модуля `matplotlib.pyplot`, создайте объект `matplotlib.figure.Figure` с названием `fig` и массив объектов `Axes` под названием `ax`, причем так, чтобы у вас было 4 отдельных графика в сетке, состоящей из двух строк и двух столбцов. В каждом графике массив `x` используется для координат по горизонтали. В левом верхнем графике для координат по вертикали используйте `y1`, в правом верхнем - `y2`, в левом нижнем - `y3`, в правом нижнем - `y4`. Дайте название графикам: 'График y1', 'График y2' и т.д.

Для графика в левом верхнем углу установите границы по оси `x` от -5 до 5.

Установите размеры фигуры 8 дюймов по горизонтали и 6 дюймов по вертикали.

Вертикальные и горизонтальные зазоры между графиками должны составлять 0.3.

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```

>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x = np.linspace(-3, 3, 51)
>>> y1 = x ** 2
>>> y2 = 2 * x + 0.5
>>> y3 = -3 * x - 1.5
>>> y4 = np.sin(x)
>>> fig, ax = plt.subplots()
>>> plt.figure(figsize=(8, 6))
<Figure size 800x600 with 0 Axes>
>>> plt.plot(x, y1)
[<matplotlib.lines.Line2D object at 0x7f1d83ad34f0>]
>>> plt.plot(x, y2)

```

```
[<matplotlib.lines.Line2D object at 0x7f1d83ad39a0>]
>>> plt.plot(x, y3)
[<matplotlib.lines.Line2D object at 0x7f1d83ad3d30>]
>>> plt.plot(x, y4)
[<matplotlib.lines.Line2D object at 0x7f1d83ad3ca0>]
>>> plt.xticks([-5, 5])
([<matplotlib.axis.XTick object at 0x7f1d83abbf40>, <matplotlib.axis.XTick object at
0x7f1d83abbf10>], <a list of 2 Text xticklabel objects>)
>>> plt.show()
>>>
```

Задания на повторения.

Задание 1.

Исходные данные:

Создать одномерный массив Numpy под названием a из 12 последовательных целых чисел чисел от 12 до 24 не включительно.

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> a = np.arange(12, 24)
>>> print(a)
[12 13 14 15 16 17 18 19 20 21 22 23]
>>>
```

Задание 2.

Исходные данные:

Создать 5 двумерных массивов разной формы из массива a. Не использовать в аргументах метода reshape число -1.

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> a = np.array([ [1, 2, 3, 4], [25, 45, 78, 88], [123, 257, 789, 898], [1234, 1456, 5642, 9999] ])
>>> a.ravel()
array([ 1,  2,  3,  4, 25, 45, 78, 88, 123, 257, 789,
        898, 1234, 1456, 5642, 9999])
>>> a.transpose()
array([[ 1, 25, 123, 1234],
       [ 2, 45, 257, 1456],
       [ 3, 78, 789, 5642],
       [ 4, 88, 898, 9999]])
>>> a.reshape((4, 4))
array([[ 1,  2,  3,  4],
       [25, 45, 78, 88],
       [123, 257, 789, 898],
       [1234, 1456, 5642, 9999]])
>>> a.reshape((2, 8))
array([[ 1,  2,  3,  4, 25, 45, 78, 88],
       [123, 257, 789, 898, 1234, 1456, 5642, 9999]])
```

```
>>> a.resize()
>>> a
array([[ 1,  2,  3,  4],
       [25, 45, 78, 88],
       [123, 257, 789, 898],
       [1234, 1456, 5642, 9999]])
>>>
```

Задание 3.

Исходные данные:

Создать 5 двумерных массивов разной формы из массива а. Использовать в аргументах метода reshape число -1 (в трех примерах - для обозначения числа столбцов, в двух - для строк).

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> a = np.array([ [1, 2, 5, 8], [125, 584, 789, 888], [1254, 5894, 7894, 9999] ])
>>>
>>> a.reshape((3, -1))
array([[ 1,  2,  5,  8],
       [125, 584, 789, 888],
       [1254, 5894, 7894, 9999]])
>>> a.reshape((4, -1))
array([[ 1,  2,  5],
       [ 8, 125, 584],
       [789, 888, 1254],
       [5894, 7894, 9999]])
>>> a.reshape((1, -1))
array([[ 1,  2,  5,  8, 125, 584, 789, 888, 1254, 5894, 7894,
        9999]])
>>> a.reshape((-1, 3))
array([[ 1,  2,  5],
       [ 8, 125, 584],
       [789, 888, 1254],
       [5894, 7894, 9999]])
>>> a.reshape((-1, 4))
array([[ 1,  2,  5,  8],
       [125, 584, 789, 888],
       [1254, 5894, 7894, 9999]])
>>>
```

Задание 4.

Исходные данные:

Можно ли массив Numpy, состоящий из одного столбца и 12 строк, назвать одномерным?

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> a = np.arange(1, 121).reshape(12, -1)
```

```
>>> s = np.linalg.matrix_rank(a)
>>> s
2
>>>
# согласно данного варианта, нельзя назвать данный массив одномерным.
```

```
Python 3.8.10 (default, Jun 2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.arange(1, 121).ravel()
>>> s = np.linalg.matrix_rank(a)
>>> s
1
>>>
# согласно данного варианта, данный массив является одномерным.
```

Задание 5.

Исходные данные:

Создать массив из 3 строк и 4 столбцов, состоящий из случайных чисел с плавающей запятой из нормального распределения со средним, равным 0 и среднеквадратичным отклонением, равным 1.0. Получить из этого массива одномерный массив с таким же атрибутом size, как и исходный массив.

Решение:

```
Python 3.8.10 (default, Jun 2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.arange(0.5, 4, 0.3)
>>> a
array([0.5, 0.8, 1.1, 1.4, 1.7, 2. , 2.3, 2.6, 2.9, 3.2, 3.5, 3.8])
>>> a.size
12
>>> b = a.ravel()
>>> b
array([0.5, 0.8, 1.1, 1.4, 1.7, 2. , 2.3, 2.6, 2.9, 3.2, 3.5, 3.8])
>>>
```

Задание 6.

Исходные данные:

Создать массив a, состоящий из целых чисел, убывающих от 20 до 0 неключительно с интервалом 2.

Решение:

```
Python 3.8.10 (default, Jun 2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.arange(1, 20, 2)
>>> a[::-1].sort()
>>> print(a)
[19 17 15 13 11 9 7 5 3 1]
>>>
```

Задание 7.

Исходные данные:

Создать массив *b*, состоящий из 1 строки и 10 столбцов: целых чисел, убывающих от 20 до 1 не включительно с интервалом 2. В чем разница между массивами *a* и *b*?

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> a = np.arange(2, 22, 2)
>>> a[::-1].sort()
>>> a.reshape(1, 10)
array([[20, 18, 16, 14, 12, 10, 8, 6, 4, 2]])
>>>
```

Разница между массивом из предыдущего задания и из текущего в том что во-первых разные числа которые необходимо не включать в интервал.

Задание 8.

Исходные данные:

Вертикально соединить массивы *a* и *b*. *a* - двумерный массив из нулей, число строк которого больше 1 и на 1 меньше, чем число строк двумерного массива *b*, состоящего из единиц.

Итоговый массив *v* должен иметь атрибут *size*, равный 10.

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> a = np.zeros((3, 2))
>>> b = np.ones((2, 2))
>>> c = np.vstack((a, b))
>>> print(c)
[[0. 0.]
 [0. 0.]
 [0. 0.]
 [1. 1.]
 [1. 1.]]
>>> c.size
10
>>>
```

Задание 9.

Исходные данные:

Создать одномерный массив *a*, состоящий из последовательности целых чисел от 0 до 12.

Поменять форму этого массива, чтобы получилась матрица *A* (двумерный массив Numpy),

состоящая из 4 строк и 3 столбцов. Получить матрицу *A_t* путем транспонирования матрицы

A. Получить матрицу *B*, умножив матрицу *A* на матрицу *A_t* с помощью матричного

умножения. Какой размер имеет матрица *B*? Получится ли вычислить обратную матрицу для матрицы *B* и почему?

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> a = np.arange(0, 12).reshape(4, 3)
>>> b = np.transpose(a)
>>> c = np.dot(a, b)
>>> c
array([[ 5, 14, 23, 32],
       [14, 50, 86, 122],
       [23, 86, 149, 212],
       [32, 122, 212, 302]])
>>> np.linalg.det(c)
0.0
>>>
```

Размер матрицы c 4x4.

Так как определитель данной матрицы равен нулю, то мы не можем рассчитать обратную матрицу.

Задание 10.

Исходные данные:

Инициализируйте генератор случайных чисел с помощью объекта seed, равного 42.

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> np.random.seed(42)
>>> print(np.random.rand(42))
[0.37454012 0.95071431 0.73199394 0.59865848 0.15601864 0.15599452
 0.05808361 0.86617615 0.60111501 0.70807258 0.02058449 0.96990985
 0.83244264 0.21233911 0.18182497 0.18340451 0.30424224 0.52475643
 0.43194502 0.29122914 0.61185289 0.13949386 0.29214465 0.36636184
 0.45606998 0.78517596 0.19967378 0.51423444 0.59241457 0.04645041
 0.60754485 0.17052412 0.06505159 0.94888554 0.96563203 0.80839735
 0.30461377 0.09767211 0.68423303 0.44015249 0.12203823 0.49517691]
>>>
```

Задание 11.

Исходные данные:

Создайте одномерный массив c, составленный из последовательности 16-ти случайных равномерно распределенных целых чисел от 0 до 16 не включительно.

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> c = np.arange(0, 16)
>>> print(c)
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
>>>
```

Задание 12.

Исходные данные:

Поменяйте его форму так, чтобы получилась квадратная матрица С. Получите матрицу D, поэлементно прибавив матрицу В из предыдущего вопроса к матрице С, умноженной на 10. Вычислите определитель, ранг и обратную матрицу D_inv для D.

Решение:

```
vik@vik-Z580:~$ command python3
Python 3.8.10 (default, Jun 2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.arange(0, 12).reshape(4, 3)
>>> b = np.transpose(a)
>>> c = np.dot(a, b)
>>> d = np.arange(0, 16).reshape(4, 4)
>>> e = d + (c * 10)
>>> e
array([[ 50, 141, 232, 323],
       [ 144, 505, 866, 1227],
       [ 238, 869, 1500, 2131],
       [ 332, 1233, 2134, 3035]])
>>> np.linalg.matrix_rank(e)
2
>>> np.linalg.det(e)
0.0
>>>
```

Так как данная матрица имеет нулевой определитель, то не может быть вычислена обратная матрица.