

# Лабораторная работа 1. Знакомство с библиотеками для анализа данных

Результат лабораторной работы – отчет. Мы предпочитаем принимать отчеты в формате ноутбуков Jupyter (ipynb-файл). Постарайтесь сделать ваш отчет интересным рассказом, последовательно отвечающим на вопросы из заданий. Помимо ответов на вопросы, в отчете так же должен быть код, однако чем меньше кода, тем лучше всем: нам – меньше проверять, вам – проще найти ошибку или дополнить эксперимент. При проверке оценивается четкость ответов на вопросы, аккуратность отчета и кода.

## О задании

Лабораторная работа №1 направлена на работу с пакетами NumPy, Pandas и Matplotlib путём решения набора задач. В данном задании разрешено пользоваться только стандартной библиотекой языка Python, а также пакетами NumPy, Pandas и Matplotlib. Целью задания является редактирование данного jupyter notebook-a (в части блоков с кодом на python) таким образом, чтобы были реализованы указанные в задании функции.

## Оценивание и штрафы

- Каждая из задач имеет определенную «стоимость» (указана в скобках около задачи)
- Максимально допустимая оценка за работу — 15 баллов
- Сдавать задание после указанного срока сдачи нельзя
- «Похожие» решения считаются плагиатом и все задействованные студенты (в том числе те, у кого списали) не могут получить за него больше 0 баллов и понижают карму (подробнее о плагиате см. на странице курса)
- Если вы нашли решение какого-то из заданий в открытом источнике, необходимо прислать ссылку на этот источник (скорее всего вы будете не единственным, кто это нашел, поэтому чтобы исключить подозрение в плагиате, необходима ссылка на источник)
- Не оцениваются задания с удалёнными формулировкам
- Не оценивается лабораторная работа целиком, если она была выложена в открытый источник

## Правила сдачи

Работу необходимо сдавать в систему Anytask (более подробную информацию можно найти на странице курса).

## Знакомство с Numpy

Во всех заданиях данного раздела запрещено использовать циклы (ключевые слова *for* и *while*), [list comprehension](#), [map](#) и [т.п.](#)

Для каждой задачи приведите примеры использования реализованных функций. Без примеров баллы за задание будут снижены в 2 раза, примеры должны отличаться от тех что приведены в самом задании

Под матрицей в заданиях понимается двумерный [numpy.array](#)

In [41]:

```
import numpy as np
```

**(0.25 балла) Задание 1.** Реализуйте функцию, принимающую на вход матрицу `X` и два массива индексов `indices1` и `indices2` одинаковой длины и возвращающую `np.array`, состоящий из последовательности элементов

```
[X[indices1[0], indices2[0]], ... , X[indices1[N-1], indices2[N-1]]]
```

In [42]:

```
def construct_array(X, indices1, indices2):
    return np.array(np.array(X)[np.array(indices1), np.array(indices2)])

def print_example(ex_number, X, indices1, indices2):
    print("Ex #", ex_number)
    print("X =", X)
    print("Indices1 =", indices1)
    print("Indices2 =", indices2)
    print("Constructed array:", construct_array(X, indices1, indices2), "\n")
```

```

print_example(1,
              [[1, 2, 3], [4, 5, 6], [7, 8, 9]],
              [0, 1, 2],
              [0, 1, 1])
print_example(2,
              [[1, 2, 3], [4, 5, 6], [7, 8, 9]],
              [0, 1],
              [2, 1])

```

```

Ex # 1
X = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Indices1 = [0, 1, 2]
Indices2 = [0, 1, 1]
Constructed array: [1 5 8]

```

```

Ex # 2
X = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Indices1 = [0, 1]
Indices2 = [2, 1]
Constructed array: [3 5]

```

**(0.25 балла) Задача 2.** Реализуйте функцию, которая на вход принимает два массива `X` и `Y` (массивы могут быть n-мерные, но их размерности должны совпадать), а возвращает **True** если они равны и **False** - иначе.

In [43]:

```

def detect_identical(X, Y):
    return np.array_equal(np.array(X), np.array(Y))

def print_example(ex_number, a, b):
    print("Ex #", ex_number)
    print("X =", a)
    print("Y =", b)
    print("X == Y:", detect_identical(a, b), "\n")

print_example(1,
              [[[1], [0]], [[2], [0]], [[3], [0]]],
              [[[1], [0]], [[2], [0]], [[3], [0]]])
print_example(2,
              [0, 1, 2],
              [0, 1, 1])

print_example(3,
              [[], []],
              [[], []])

```

```

Ex # 1
X = [[[1], [0]], [[2], [0]], [[3], [0]]]
Y = [[[1], [0]], [[2], [0]], [[3], [0]]]
X == Y: True

```

```

Ex # 2
X = [0, 1, 2]
Y = [0, 1, 1]
X == Y: False

```

```

Ex # 3
X = [[], []]
Y = [[], []]
X == Y: True

```

**(0.25 балла) Задание 3.** Реализуйте функцию, которая на вход принимает цветное изображение `X` (трехмерный вектор размера (n, m, 3)) и возвращает среднее значение по трём каналам (вектор длины 3).

In [44]:

```

def mean_channel(X):
    return np.average(np.array(X), axis=(0, 1))

```

```
def print_example(ex_number, X):
    print("Ex #", ex_number)
    print("X:")
    print(X)
    print("Mean channel:", mean_channel(X), "\n")

X = np.ones((3, 4, 3))
X[0][0][0] = 0
X[0][0][1] = 2
print_example(1, X)

print_example(2, np.random.randint(0, 6, size=(2, 2, 3)))
```

```
Ex # 1
X:
[[[0. 2. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]

 [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]]
Mean channel: [0.91666667 1.08333333 1.         ]

Ex # 2
X:
[[[0 0 1]
  [1 0 2]]

 [[1 3 5]
  [1 1 2]]]
Mean channel: [0.75 1.         2.5 ]
```

**(0.25 балла) Задание 4.** Реализуйте функцию, принимающую на вход матрицу `X` и некоторое число `a` и возвращающую ближайший к числу элемент матрицы.

Например, для `X = np.arange(0,10).reshape((2, 5))` и `a = 3.6` ответом будет 4.

In [45]:

```
def nearest_value(X, a):
    index = (np.abs(np.array(X).flatten() - a)).argmin()
    return np.array(X).flatten()[index]

def print_example(ex_number, X, a):
    print("Ex #", ex_number)
    print("X:")
    print(X)
    print("a =", a)
    print("Nearest value =", nearest_value(X, a), "\n")

print_example(1, np.arange(1,11).reshape((2, 5)), 2.6)
print_example(2, [[0, 1], [1, 0], [0, 0]], 0.5)
print_example(3, [[0, 1], [-1, 0]], -2)
```

```
Ex # 1
X:
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
a = 2.6
Nearest value = 3
```

```
Ex # 2
X:
[[0, 1], [1, 0], [0, 0]]
a = 0.5
```

```
Nearest value = 0
```

```
Ex # 3
X:
[[0, 1], [-1, 0]]
a = -2
Nearest value = -1
```

**(0.5 балла) Задача 5.** Реализуйте функцию, принимающую на вход два одномерных массива `x` и `y` и возвращающую матрицу, в которой первый массив соответствует первому столбцу матрицы, второй - второму.

В этом задании **запрещается** пользоваться операцией транспонирования и рекомендуется воспользоваться методом [reshape](#).

In [46]:

```
def construct_matrix(x, y):
    return np.column_stack((np.array(x), np.array(y)))
def print_example(ex_number, x, y):
    print("Ex #", ex_number)
    print("X =", x)
    print("Y =", y)
    print("Matrix:")
    print(construct_matrix(x, y), "\n")

print_example(1, [0, 1, 2], [3, 4, 5])
print_example(2, [0, 1], [-1, 0])
```

```
Ex # 1
X = [0, 1, 2]
Y = [3, 4, 5]
Matrix:
[[0 3]
 [1 4]
 [2 5]]
```

```
Ex # 2
X = [0, 1]
Y = [-1, 0]
Matrix:
[[ 0 -1]
 [ 1  0]]
```

**(0.5 балла) Задание 6.** Реализуйте функцию, которая на вход принимает вектор `x`, содержащий целые числа, кроме 0, а возвращает вектор со вставленными нулями между числами.

Например, из `[1, -5, 3]` получить `[1, 0, -5, 0, 3]`

In [47]:

```
def add_zeros(x):
    zeros = np.zeros(len(np.array(x)) * 2 - 1, dtype=int)
    print(zeros)
    zeros[::2] = x
    return zeros
def print_example(ex_number, x):
    print("Ex #", ex_number)
    print("X =", x)
    print("X with zeros = ", add_zeros(x), "\n")

print_example(1, [1, -4, 3])
print_example(2, [2, 1, 2, 1])
```

```
Ex # 1
X = [1, -4, 3]
[0 0 0 0 0]
X with zeros = [ 1  0 -4  0  3]
```

```
Ex # 2
X = [2, 1, 2, 1]
```

```
[0 0 0 0 0 0 0]
X with zeros = [2 0 1 0 2 0 1]
```

**(0.75 балла) Задание 7.** Реализуйте функцию для подсчёта произведения ненулевых элементов на диагонали прямоугольной матрицы

Например, для `X = np.array([[1, 0, 1], [2, 0, 2], [3, 0, 3], [4, 4, 4]])` ответом является 3. Если ненулевых элементов нет, функция должна возвращать `None`.

In [48]:

```
def nonzero_product(X):
    diag = np.diagonal(np.array(X))
    if (len(diag[np.nonzero(diag)]) != 0):
        return np.product(diag[np.nonzero(diag)])
    else:
        return None

def print_example(ex_number, X):
    print("Ex #", ex_number)
    print("X:")
    print(X)
    print("Nonzero product = ", nonzero_product(X), "\n")

print_example(1, np.array([[0, 0, 1], [2, -2, 2], [3, 0, 6], [4, 4, 4]]))
print_example(2, np.array([[0, 1, 1, 1], [2, 0, 2, 2], [3, 3, 0, 3], [4, 4, 4, 0]]))
```

```
Ex # 1
X:
[[ 0  0  1]
 [ 2 -2  2]
 [ 3  0  6]
 [ 4  4  4]]
Nonzero product = -12
```

```
Ex # 2
X:
[[0 1 1 1]
 [2 0 2 2]
 [3 3 0 3]
 [4 4 4 0]]
Nonzero product = None
```

**(0.75 балла) Задание 8.** Реализуйте функцию, возвращающую максимальный элемент в массиве `X` среди элементов, перед которыми стоит нулевой.

Например, для `X = np.array([6, 2, 0, 3, 0, 0, 5, 7, 0])` ответом является 5. Если подходящих элементов нет, функция должна возвращать `None`.

In [49]:

```
def max_element(X):
    zero_indexes = np.where(X == 0)[0]
    if len(zero_indexes) == 0:
        return None
    if zero_indexes[len(zero_indexes) - 1] == len(X) - 1:
        zero_indexes = zero_indexes[:-1]
    if len(zero_indexes) == 0:
        return None
    zero_indexes += 1
    return np.amax(X[zero_indexes])

def print_example(ex_number, X):
    print("Ex #", ex_number)
    print("X =", X)
    print("Max element = ", max_element(X), "\n")

print_example(1, np.array([6, 2, 0, -1, 0, 0, 6, 7, 0]))
print_example(2, np.array([0, 1, 0, 2, 0, 3, 0, -4, 0, -5, 0, -6]))
print_example(3, np.array([1, 2, 3]))
```

```

Ex # 1
X = [ 6  2  0 -1  0  0  6  7  0]
Max element = 6

Ex # 2
X = [ 0  1  0  2  0  3  0 -4  0 -5  0 -6]
Max element = 3

Ex # 3
X = [1 2 3]
Max element = None

```

**(0.75 балла) Задание 9.** Реализуйте функцию, принимающую на вход матрицу `X` и возвращающую все её уникальные строки в виде матрицы.

In [50]:

```

def get_unique_rows(X):
    return np.array(np.unique(np.array(X), axis=0))
def print_example(ex_number, X):
    print("Ex #", ex_number)
    print("X:")
    print(X)
    print("Unique rows:")
    print(get_unique_rows(X), "\n")

print_example(1, [[0, 0, 1], [0, 1, 0], [0, 0, 1], [1, 0, 0], [1, 0, 0]])
print_example(2, [[1, 2, 3, -1], [1, 3, 3, -1], [1, 3, 3, -1]])

```

```

Ex # 1
X:
[[0, 0, 1], [0, 1, 0], [0, 0, 1], [1, 0, 0], [1, 0, 0]]
Unique rows:
[[0 0 1]
 [0 1 0]
 [1 0 0]]

Ex # 2
X:
[[1, 2, 3, -1], [1, 3, 3, -1], [1, 3, 3, -1]]
Unique rows:
[[ 1  2  3 -1]
 [ 1  3  3 -1]]

```

**(0.75 балла) Задача 10.** Реализуйте функцию, которая во входной вещественной матрице `X` находит все значения `nan` и заменяет их на среднее арифметическое всех остальных элементов. Если все элементы матрицы `nan`, то верните нулевую матрицу той же размерности.

In [51]:

```

def replace_nans(X):
    X_new = np.array(X)
    X_return = np.array(X)
    num_of_nan = np.sum(np.isnan(X_new))
    if num_of_nan == np.shape(X_new)[0] * np.shape(X_new)[1]:
        return np.zeros(np.shape(X_new))
    X_new[np.isnan(X_new)] = 0
    sum_of_elements = np.sum(X_new)
    X_return[np.isnan(X_return)] = sum_of_elements / (np.shape(X_new)[0] * np.shape(X_new)[1] -
num_of_nan)
    return X_return

def print_example(ex_number, X):
    print("Ex #", ex_number)
    print("X:")
    print(X)
    print("Matrix without nan:")
    print(replace_nans(X), "\n")

```

```
print_example(1, [[float('nan'), 2, 0], [0, -1, float('nan')]])
print_example(2, [[float('nan'), float('nan')], [float('nan'), float('nan')]])
print_example(3, [[float('nan'), 1], [-1, float('nan')]])
```

```
Ex # 1
X:
[[nan, 2, 0], [0, -1, nan]]
Matrix without nan:
[[ 0.25  2.    0. ]
 [ 0.   -1.   0.25]]
```

```
Ex # 2
X:
[[nan, nan], [nan, nan]]
Matrix without nan:
[[0. 0.]
 [0. 0.]]
```

```
Ex # 3
X:
[[nan, 1], [-1, nan]]
Matrix without nan:
[[ 0.  1.]
 [-1.  0.]]
```

**(1 балл) Задача 11.** Напишите функцию, генерирующую [матрицу Вандермонда](#), принимающую на вход вектор  $(x_1, \dots, x_n)$ .

В этом задании **запрещается** пользоваться готовыми реализациями (например, [numpy.vander](#)), а также [np.repeat](#) и [np.transpose](#).

При решении задействуйте [np.reshape](#) и/или [np.newaxis](#).

In [52]:

```
def vander(x):
    powers = np.arange(len(x))
    x1 = (np.array(x))[:, np.newaxis]
    return np.power(x1, powers)

def print_example(ex_number, X):
    print("Ex #", ex_number)
    print("X =", X)
    print("Vandermonde matrix:")
    print(vander(X), "\n")
print_example(1, [1, 2, 3])
print_example(2, [0, 0, 0])
print_example(3, [1, -1, 0, 2])
```

```
Ex # 1
X = [1, 2, 3]
Vandermonde matrix:
[[1 1 1]
 [1 2 4]
 [1 3 9]]
```

```
Ex # 2
X = [0, 0, 0]
Vandermonde matrix:
[[1 0 0]
 [1 0 0]
 [1 0 0]]
```

```
Ex # 3
X = [1, -1, 0, 2]
Vandermonde matrix:
[[ 1  1  1  1]
 [ 1 -1  1 -1]
 [ 1  0  0  0]
 [ 1  2  4  8]]
```

(1 балл) **Задача 12.** Даны две вещественные матрицы  $X$  и  $Y$  с одинаковым числом столбцов и, в общем случае, различным числом строк. Необходимо реализовать функцию, вычисляющую матрицу попарных [косинусных коэффициентов](#) между всеми  $X_i$  и  $Y_j$ , где  $X_i$  -  $i$ -ая строка матрицы  $X$ , а  $Y_j$  -  $j$ -ая строка матрицы  $Y$ .

В этом задании **запрещается** пользоваться готовыми реализациями, а также [np.repeat](#) и [np.transpose](#).

При решении задействуйте [np.reshape](#) и/или [np.newaxis](#), [np.sqrt](#), [np.sum](#) и [np.power](#).

In [53]:

```
def count_cosine_similarity(X, Y):
    powers_for_X = np.full(np.array(np.shape(X))[1], 2)
    powers_for_Y = np.full(np.array(np.shape(Y))[1], 2)
    big_Y = np.tile(np.matrix(Y), (np.shape(X)[0], 1))
    new_shape_X = (np.shape(X)[0] * np.shape(Y)[0], np.shape(X)[1])
    big_X = np.tile(np.matrix(X), (1, np.shape(Y)[0]))
    big_X = np.reshape(big_X, new_shape_X)
    norms_x = np.sqrt(np.sum(np.power(np.array(big_X), powers_for_X), axis=1))
    norms_x = np.reshape(norms_x, (np.shape(X)[0], np.shape(Y)[0]))
    norms_y = np.sqrt(np.sum(np.power(np.array(big_Y), powers_for_Y), axis=1))
    norms_y = np.reshape(norms_y, (np.shape(X)[0], np.shape(Y)[0]))
    inner_product = np.reshape(np.sum(np.multiply(big_X, big_Y), axis=1),
                               (np.shape(X)[0], np.shape(Y)[0]))
    return (np.divide(inner_product, np.multiply(norms_x, norms_y)))

def print_example(ex_number, X, Y):
    print("Ex #", ex_number)
    print("X:")
    print(X)
    print("Y:")
    print(Y)
    print("Cosine similarity matrix:")
    print(count_cosine_similarity(X, Y), "\n")
```

```
X = [[1, 1], [1, 1], [1, 1]]
Y = [[1, 2], [1, 2], [1, 2], [1, 2]]
print_example(1, X, Y)
```

```
Y = [[1, 0], [3, 4]]
X = [[0, 1], [1, 0], [0, 2]]
print_example(2, X, Y)
```

```
Ex # 1
X:
[[1, 1], [1, 1], [1, 1]]
Y:
[[1, 2], [1, 2], [1, 2], [1, 2]]
Cosine similarity matrix:
[[0.9486833 0.9486833 0.9486833 0.9486833]
 [0.9486833 0.9486833 0.9486833 0.9486833]
 [0.9486833 0.9486833 0.9486833 0.9486833]]
```

```
Ex # 2
X:
[[0, 1], [1, 0], [0, 2]]
Y:
[[1, 0], [3, 4]]
Cosine similarity matrix:
[[0.  0.8]
 [1.  0.6]
 [0.  0.8]]
```

(1 балл) **Задача 13.** Написать функцию, которая получает на вход матрицу и масштабирует каждый её столбец, а именно вычитает из столбца его среднее значение и делит столбец на стандартное отклонение.

Для тестирования можно сгенерировать с помощью метода [numpy.random.randint](#) случайную матрицу и проверить на ней работу метода.

Убедитесь, что в функции не будет происходить деления на ноль, если происходит деление на ноль, то верните **None**.

In [54]:

```
def scale(X):
```

```

mean = np.matrix(X).mean(axis=0)
mean_matrix = np.tile(mean, (np.array(np.shape(X))[0], 1))
res = X - mean_matrix
stds = np.matrix(X).std(axis=0)
#if 0 in stds:
#    return None
#return np.divide(res, stds)
return np.divide(res, stds, out=np.full(np.shape(X), None), where=stds!=0)

def print_example(ex_number, X):
    print("Ex #", ex_number)
    print("X:")
    print(X)
    print("Scaled matrix:")
    print(scale(X), "\n")

X = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print_example(1, X)

X = [[1, 2], [1, 1], [1, 0]]
print_example(2, X)

X = [[1, 0], [0, 1], [1, 0]]
print_example(3, X)

```

```

Ex # 1
X:
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Scaled matrix:
[[-1.2247448713915892 -1.2247448713915892 -1.2247448713915892]
 [0.0 0.0 0.0]
 [1.2247448713915892 1.2247448713915892 1.2247448713915892]]

Ex # 2
X:
[[1, 2], [1, 1], [1, 0]]
Scaled matrix:
[[None 1.224744871391589]
 [None 0.0]
 [None -1.224744871391589]]

Ex # 3
X:
[[1, 0], [0, 1], [1, 0]]
Scaled matrix:
[[0.7071067811865476 -0.7071067811865475]
 [-1.414213562373095 1.4142135623730951]
 [0.7071067811865476 -0.7071067811865475]]

```

**(1 балл) Задача 14.** Пусть  $N = 1000$ . Повторите  $N$  раз следующий эксперимент: сгенерируйте две матрицы размера  $N \times N$  из стандартного нормального распределения, перемножьте их (как матрицы) и найдите максимальный элемент. Какое среднее значение по экспериментам у максимальных элементов? 95-процентная квантиль?

При решении задачи для повторения экспериментов воспользуйтесь [list comprehension](#), а также [tqdm notebook](#) - для отслеживания прогресса.

In [55]:

```

from tqdm import tqdm

N = 1000
max_elements = []
for _ in tqdm(range(N)):
    matrix1 = np.random.normal(0, 1, (N, N))
    matrix2 = np.random.normal(0, 1, (N, N))
    mult = np.dot(matrix1, matrix2)
    max_elements.append(np.amax(mult))
print(np.mean(max_elements), np.quantile(max_elements, .95))

```

100% |██████████| 1000/1000 [02:00<00:00, 8.56it/s]

## Аналитика данных с Pandas

Загрузите таблицу с данными из `articles.csv`. Удалите записи, в которых присутствуют пропуски.

**(0.5 балла) Задача 15.** Прodelайте следующие базовые операции с датафреймами:

1. определите количество различных издательств в таблице;
2. найдите количество опубликованных статей в отрезке [2016-06-01, 2016-12-31];
3. посчитайте распределение статей автора *Tom Ciccotta* по годам;
4. найдите месяц, в котором было наибольшее число статей;
5. выпишите 3 первые статьи автора *John Hayward* в 2016 году.

In [56]:

```
import pandas as pd

# task 1
articles = pd.read_csv('articles.csv')
for name in list(articles.columns.values):
    articles[name].replace('', np.nan, inplace=True)
    articles.dropna(subset=[name], inplace=True)
num_publishers = articles.groupby('publication').nunique().shape[0]
print('Number of publishers:', num_publishers, '\n')

# task 2
num_articles = len((articles['date'] >= '2016-06-01') & (articles['date'] <= '2016-12-31')).nonzero()[0]
print('Number of articles between 2016-06-01 and 2016-12-31:', num_articles, '\n')

# task 3
Tom_Ciccotta_articles = articles.loc[articles['author'] == 'Tom Ciccotta'].groupby(
    'year').count().reset_index().drop(
    columns=['title', 'publication', 'author', 'date', 'month', 'content']).rename(
    columns={'id': 'number of publications'})
print('Tom Ciccotta articles:')
print(Tom_Ciccotta_articles, '\n')

# task 4
articles['date'] = pd.to_datetime(articles['date'])
best_month = articles.groupby(articles['date'].dt.strftime('%B')).count()['id'].idxmax()
print('Month with biggest number of articles:', best_month, '\n')

# task 5
John_Hayward_articles = ((articles['author'] == 'John Hayward') &
    (articles['date'] >= '2016-01-01') & (articles['date'] <= '2016-12-31'))
John_Hayward_three_articles = articles.loc[John_Hayward_articles].sort_values(by=['date']).head(3)
print("First three John Hayward's articles in 2016:")
John_Hayward_three_articles
```

Number of publishers: 5

Number of articles between 2016-06-01 and 2016-12-31: 17159

Tom Ciccotta articles:

	year	number of publications
0	2016	124
1	2017	132

Month with biggest number of articles: January

First three John Hayward's articles in 2016:

Out[56]:

id	title	publication	author	date	year	month	content
----	-------	-------------	--------	------	------	-------	---------

	id	title	publication	author	date	year	month	content
16861	35600	Indonesian Couple Beaten with Canes for Violat...	Breitbart	John Hayward	2016-01-01	2016	1	Islamic sharia law was enforced on couples on...
22041	40791	Islamic State Claims Credit For Gun Attack On ...	Breitbart	John Hayward	2016-01-01	2016	1	The Islamic State has claimed responsibility f...
27561	46325	Protests Across India Against Saudi Execution ...	Breitbart	John Hayward	2016-01-04	2016	1	Protests over Saudi Arabia's execution of Shii...

### Работа со строками в датафрейме.

Для датафреймов существуют методы работы со строковыми данными. Чтобы применить их, необходимо воспользоваться атрибутом `str`, после чего вызвать нужные методы работы со строками. Например, вызов:

```
df['content'].str.len()
```

подсчитает для каждой строчки в датафрейме количество символов в колонке `content`. Более подробную информацию про работу с текстовыми данными в Pandas можно найти [здесь](#).

### (0.5 балла) Задача 16.

Найдите в датафрейме всех авторов, имя которых содержит Faith. Выведите Series, состоящий из всех таких уникальных имен.

In [57]:

```
pd.Series(articles.loc[articles['author'].str.contains('Faith')]['author'].unique())
```

Out[57]:

```
0    Faith Haleh Robinson
1           Faith Karimi
2    Faith Haleh Robinson
3           Faith Karimi,
4           Faith Karimi
5           Faith Karimi,
6           Faith Karimi
7           Faith Karimi
dtype: object
```

### (1 балл) Задача 17.

Как можно заметить, в таблице существует множество различных написаний имени Faith Karimi. В основном эти написания различаются пунктуацией - лишние пробелы и запятые. Для правильного подсчета статистик для текстовых данных зачастую возникает необходимость в их предобработке.

Проведите следующие преобразования для колонок `author` и `content`:

1. приведение текста к нижнему регистру;
2. удаление всей пунктуации из текста;
3. удаление пробелов в начале и конце строки;
4. замена подряд идущих пробелов одним пробелом.

Например, строка " It's 6 a.m. and I'm still doing this homework :(( "

преобразуется в строку `its 6 am and im still doing this homework`

Подсчитайте статистику для имени Faith из прошлого задания. Проверьте, что теперь различные способы написания «схлопываются» в один.

In [58]:

```
articles['author'] = pd.Series(articles['author'].str.lower().str.replace(
    '[^\w\s]', '').str.strip().str.split(' ').str.join(' ')
articles['content'] = pd.Series(articles['content'].str.lower().str.replace(
    '[^\w\s]', '').str.strip().str.split(' ').str.join(' ')
pd.Series(articles.loc[articles['author'].str.contains('faith')]['author'].unique())
```

Out[58]:

```
0    faith haleh robinson
1           faith karimi
```

```
dtype: object
```

## Группировка данных

С помощью метода `groupby` удобно группировать данные по значениям одной или нескольких колонок. Далее можно вычислять различные статистики для каждой группы по отдельности.

### (1 балл) Задача 18.

Выведите для каждого автора максимальное количество публикаций за календарный год (колонка `year`), а также сам год, на котором достигается этот максимум. Выведите топ-20 строк в порядке убывания количества публикаций.

Пример:

Василий Пупкин написал 3 статьи в 2016, и 4 в 2017, а его брат Иван Пупкин только 1 статью в 2016, а в 2017 он отправился в армию и статей не писал.

Необходимо вывести

```
Василий Пупкин - 2017 - 4
Иван Пупкин    - 2016 - 1
```

Обратите внимание, что несколько вызовов методов над датафреймами можно объединить в один `pipeline`. Другими словами, можно писать

```
df.func1().func2().func3()
```

По возможности реализуйте требуемую функцию с помощью **одного** такого пайплайна. Делайте переносы для лучшей читаемости кода.

In [59]:

```
articles.groupby(
    ['author', 'year']).agg('count').reset_index().groupby(
    ['author']).max(level='id').reset_index().sort_values(
    by=['id'], ascending=False)[['author', 'year', 'id']].rename(columns={'id': 'number of'}).head(
    20)
```

Out[59]:

	author	year	number of
472	breitbart news	2017	1317
2695	pam key	2017	820
610	charlie spiering	2017	660
112	alex swoyer	2017	584
778	daniel nussbaum	2017	532
353	awr hawkins	2017	525
1598	john hayward	2017	521
1511	jerome hudson	2017	482
1292	ian hanchett	2017	429
1575	joel b pollak	2017	417
1451	jeff poor	2017	323
3496	warner todd huston	2017	316
3448	trent baker	2017	280
2726	patrick howley	2016	259
471	breitbart london	2017	258
1123	frances martel	2017	257
600	charlie nash	2017	254
1827	katherine rodriguez	2017	231
392	ben kew	2017	221

(1 балл) **Задача 19.** Для каждой статьи  $i$  исходного датафрейма посчитайте количество статей, опубликованных тем же издательством (publication), к моменту публикации  $i$  (включая статьи того же дня).

Обратите внимание, что для всех статей, выпущенных одним издательством в один день, должен получиться одинаковый ответ.

*Hint.* Возможный вариант решения:

1. сгруппировав данные по полям *publication*, *date*, посчитать размер каждой группы;
2. приджойнить размер группы к основному датафрейму с помощью функции `pd.merge`.

In [60]:

```
tmp = articles.groupby([
    'publication', 'date']).agg('count').groupby(
    'publication').cumsum()
tmp['cumsum'] = tmp['id']
tmp = tmp.drop(columns=['id', 'title', 'author', 'year', 'month', 'content'])
articles.join(tmp, on=['publication', 'date'], how='left')
```

Out[60]:

	id	title	publication	author	date	year	month	content	cumsum
0	17283	House Republicans Fret About Winning Their Hea...	New York Times	carl hulse	2016-12-31	2016	12	washington congressional republicans have ...	3608
1	17284	Rift Between Officers and Residents as Killing...	New York Times	benjamin mueller and al baker	2017-06-19	2017	6	after the bullet shells get counted the blood ...	7767
2	17285	Tyrus Wong, 'Bambi' Artist Thwarted by Racial ...	New York Times	margalit fox	2017-01-06	2017	1	when walt disneys bambi opened in 1942 critics...	3739
3	17286	Among Deaths in 2016, a Heavy Toll in Pop Musi...	New York Times	william mcdonald	2017-04-10	2017	4	death may be the great equalizer but it isnt n...	7108
4	17287	Kim Jong-un Says North Korea Is Preparing to T...	New York Times	choe sanghun	2017-01-02	2017	1	seoul south korea north koreas leader kim ...	3628
5	17288	Sick With a Cold, Queen Elizabeth Misses New Y...	New York Times	sewell chan	2017-01-02	2017	1	london queen elizabeth ii who has been bat...	3628
6	17289	Taiwan's President Accuses China of Renewed In...	New York Times	javier c hernández	2017-01-02	2017	1	beijing president tsai of taiwan sharply...	3628
7	17290	After 'The Biggest Loser,' Their Bodies Fought...	New York Times	gina kolata	2017-02-08	2017	2	danny cahill stood slightly dazed in a blizzar...	4941
8	17291	First, a Mixtape. Then a Romance. - The New Yo...	New York Times	katherine rosman	2016-12-31	2016	12	just how is hillary kerr the founder of a...	3608
9	17292	Calling on Angels While Enduring the Trials of...	New York Times	andy newman	2016-12-31	2016	12	angels are everywhere in the muñiz familys apa...	3608
10	17293	Weak Federal Powers Could Limit Trump's Climat...	New York Times	justin gillis	2017-01-03	2017	1	with donald j trump about to take control of t...	3649
11	17294	Can Carbon Capture Technology Prosper Under Tr...	New York Times	john schwartz	2017-01-05	2017	1	thompsons tex can one of the most promisin...	3718
12	17295	Mar-a-Lago, the Future Winter White House and ...	New York Times	maggie haberman	2017-01-02	2017	1	west palm beach fla when donald j trump ...	3628
13	17296	How to form healthy habits in your 20s - The N...	New York Times	charles duhigg	2017-01-02	2017	1	this article is part of a series aimed at help...	3628
14	17297	Turning Your Vacation Photos Into Works of Art...	New York Times	stephanie rosenbloom	2017-04-14	2017	4	its the season for family travel and photos ...	7452
15	17298	As Second Avenue Subway Opens, a Train Delay E...	New York Times	emma g fitzsimmons	2017-01-02	2017	1	finally the second avenue subway opened in new...	3628
16	17300	Dylann Roof Himself Rejects Best Defense Again...	New York Times	kevin sack and alan blinder	2017-01-02	2017	1	pages into the journal found in dylann s roo...	3628
17	17301	Modi's Cash Ban Brings Pain, but Corruption-We...	New York Times	geeta anand	2017-01-02	2017	1	mumbai india it was a bold and risky gambl...	3628
18	17302	Suicide Bombing in Baghdad Kills at Least 36 -...	New York Times	the associated press	2017-01-03	2017	1	baghdad a suicide bomber detonated a picku...	3649
19	17303	Fecal Pollution Taints Water at Melbourne's Be...	New York Times	brett cole	2017-01-03	2017	1	sydney australia the annual beach pilgrima...	3649

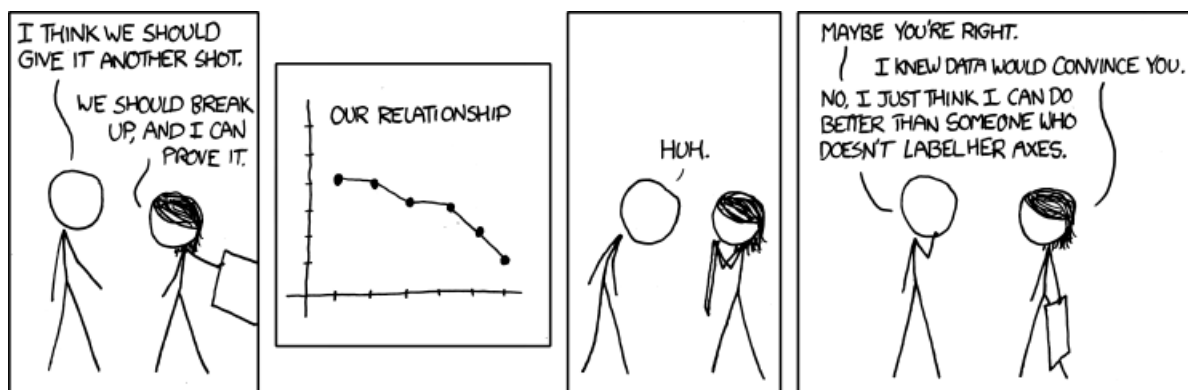
20	id	N.F.L. Playoffs: Schedule Matchups and Odds - ...	title	publication	author	date	year	month	when the green bay packers lost to the washington...	cumsum
21	17306	Mariah Carey's Manager Blames Producers for Ne...	New York Times	patrick healy	2017-01-02	2017	1	mariah carey suffered through a performance tr...	3628	
22	17307	Damaged by War, Syria's Cultural Sites Rise An...	New York Times	marlise simons	2017-01-01	2017	1	paris when the islamic state was about to ...	3613	
23	17308	George Michael's Freedom Video: An Oral Histor...	New York Times	guy trebay and jacob bernstein	2017-01-01	2017	1	pop music and fashion never met cuter than in ...	3613	
24	17309	With New Congress Poised to Convene, Obama's P...	New York Times	jennifer steinhauer	2017-01-02	2017	1	washington the most powerful and ambitious...	3628	
25	17311	Republicans Stonewalled Obama. Now the Ball Is...	New York Times	carl hulse	2017-01-03	2017	1	washington its or time for republicans...	3649	
26	17312	Istanbul, Donald Trump, Benjamin Netanyahu: Yo...	New York Times	charles mcdermid	2017-01-03	2017	1	good morning heres what you need to know the...	3649	
27	17313	Inside Trump Defense Secretary Pick's Efforts ...	New York Times	sheri fink and helene cooper	2017-01-05	2017	1	the body of the iraqi prisoner was found naked...	3718	
28	17314	ISIS Claims Responsibility for Istanbul Nightc...	New York Times	tim arango	2017-01-03	2017	1	istanbul the islamic state on monday issue...	3649	
29	17317	The Afghan War and the Evolution of Obama - Th...	New York Times	mark landler	2017-01-17	2017	1	washington president obamas advisers wrest...	4095	
...	...	...	...	...	...	...	...	...	...	
49970	73440	Barack Obama's Enduring Faith in America	Atlantic	david a graham	2017-01-10	2017	1	in his final speech to the nation as the 44th ...	154	
49971	73441	Michael Cohen: 'It Is Fake News Meant to Malig...	Atlantic	rosie gray	2017-01-10	2017	1	donald trump and his lawyer on tuesday night d...	154	
49972	73442	The Atlantic Daily: Change and Confirmation	Atlantic	rosa inocencio smith	2017-01-10	2017	1	this article is part of a feature we also send...	154	
49973	73443	What CNN's Report on Trump and Russia Does and...	Atlantic	david a graham	2017-01-10	2017	1	updated on january 10 at 636 p m despite all ...	154	
49974	73444	The Atlantic Politics & Policy Daily: Obama Out	Atlantic	candice norwood	2017-01-10	2017	1	this article is part of a feature we also send...	154	
49975	73445	What the World Might Look Like in 5 Years, Acc...	Atlantic	uri friedman	2017-01-10	2017	1	every four years a group of u s intelligence a...	154	
49976	73446	The U.S. Supreme Court Puts North Carolina's 2...	Atlantic	david a graham	2017-01-10	2017	1	durham n c the supreme court has a message f...	154	
49977	73447	Trump Meets With Vaccine Skeptic, Discusses 'C...	Atlantic	julie beck	2017-01-10	2017	1	updated on january 10 855 p m on tuesday donal...	154	
49978	73448	Can the Flaws in Credit Scoring Be Fixed?	Atlantic	gillian b white	2017-01-10	2017	1	that credit scoring and reporting is an opaque...	154	
49979	73449	Trump's Cyber-Appeasement Policy Might Encoura...	Atlantic	kaveh waddell	2017-01-10	2017	1	since well before he was elected president don...	154	
49980	73450	Taboo: A Grim, Gruesome Costume Drama Starring...	Atlantic	sophie gilbert	2017-01-10	2017	1	nobody excels at playing ferocious psychopaths...	154	
49981	73451	Clare Hollingworth: The Reporter Who Broke the...	Atlantic	david a graham	2017-01-10	2017	1	any big journalistic scoop requires a combinat...	154	
49982	73452	The Gaps in New York's Free-College Plan	Atlantic	james s murphy	2017-01-10	2017	1	new york governor andrew cuomo recently announ...	154	
49983	73453	'We Have a Problem': John Kerry on Making Poli...	Atlantic	uri friedman	2017-01-10	2017	1	in one of his last public appearances as u s s...	154	
49984	73454	The Enduring Mystery of Pain Measurement	Atlantic	john walsh	2017-01-10	2017	1	one night in may my wife sat up in bed and sai...	154	
49985	73455	What Conan O'Brien Means to Late Night's Future	Atlantic	david sims	2017-01-10	2017	1	conan obrien was once the upstart of the com...	154	
49986	73456	The Absurdity of Attacking Celebrities to Defe...	Atlantic	conor friedersdorf	2017-01-10	2017	1	fifty years ago california republicans elected...	154	
49987	73457	Drive-Through Redwoods Are Monuments to Vioen...	Atlantic	sarah zhang	2017-01-10	2017	1	this weekend amidst a torrent of rain one of c...	154	
49988	73458	How Superstar Economics Is Killing the NFL's R...	Atlantic	derek thompson	2017-01-10	2017	1	for years the national football league has bee...	154	
49989	73459	The Atlantic Daily: Passing the Presidential Mic	Atlantic	rosa inocencio smith	2017-01-11	2017	1	this article is part of a feature we also send...	171	
49990	73460	How Blackmail Works in Russia	Atlantic	julia ioffe	2017-01-11	2017	1	in january 1999 prosecutor general yury skurat...	171	
49991	73461	The Atlantic Politics & Policy Daily: Back-to...	Atlantic	candice norwood	2017-01-11	2017	1	this article is part of a feature we also send...	171	

	id	title	publication	author	date	year	month	content	cumsum
49992	73462	Obama Built an 'Infrastructure Plan' for Civil-Libe...	Atlantic	emma green	2017-01-11	2017	1	president obamas farewell speech was an exerci...	171
49993	73463	Why Trump's Conflict-of-Interest Plan Won't Pr...	Atlantic	clare foran	2017-01-11	2017	1	updated on january 11 at 556 p m et donald t...	171
49994	73464	The Irrationally Divided Critics of Donald Trump	Atlantic	conor friedersdorf	2017-01-11	2017	1	a large cohort of americans have reservations ...	171
49995	73465	Rex Tillerson Says Climate Change Is Real, but ...	Atlantic	robinson meyer	2017-01-11	2017	1	as chairman and ceo of exxonmobil rex tillerso...	171
49996	73466	The Biggest Intelligence Questions Raised by t...	Atlantic	amy zegart	2017-01-11	2017	1	ive spent nearly 20 years looking at intellige...	171
49997	73467	Trump Announces Plan That Does Little to Resol...	Atlantic	jeremy venook	2017-01-11	2017	1	donald trump will not be taking necessary step...	171
49998	73468	Dozens of For-Profit Colleges Could Soon Close	Atlantic	emily deruy	2017-01-11	2017	1	dozens of colleges could be forced to close ...	171
49999	73469	The Milky Way's Stolen Stars	Atlantic	marina koren	2017-01-11	2017	1	the force of gravity can be described using a ...	171

43694 rows × 9 columns

## Визуализация

Обратите внимание, что у графиков должны быть подписаны оси, заголовок графика и при необходимости обязательно наличие легенды. За отсутствие названий графиков и подписей к осям могут снижаться баллы. Все картинки должны быть самодостаточны и визуально удобны для восприятия, так чтобы не нужно было смотреть ваш код или знать задание, чтобы понять что на них изображено.



**(0.5 балла) Задача 20.** Используя функцию `gen_uncertain_data` для генерации выборки, отобразите на графике синим цветом функцию  $y(x)$ , а также ее доверительный интервал в виде закрашенной зеленым цветом области от  $y[i] - \text{error}[i]$  до  $y[i] + \text{error}[i]$ . Полезной может оказаться функция `fill_between`.

```
def gen_uncertain_data():
    x = np.linspace(0, 30, 100)
    y = np.sin(x/6*np.pi) + np.random.normal(0, 0.02, size=x.shape)
    error = np.random.normal(0.1, 0.02, size=y.shape)
    return x, y, error
```

In [61]:

```
%matplotlib inline
import matplotlib.pyplot as plt

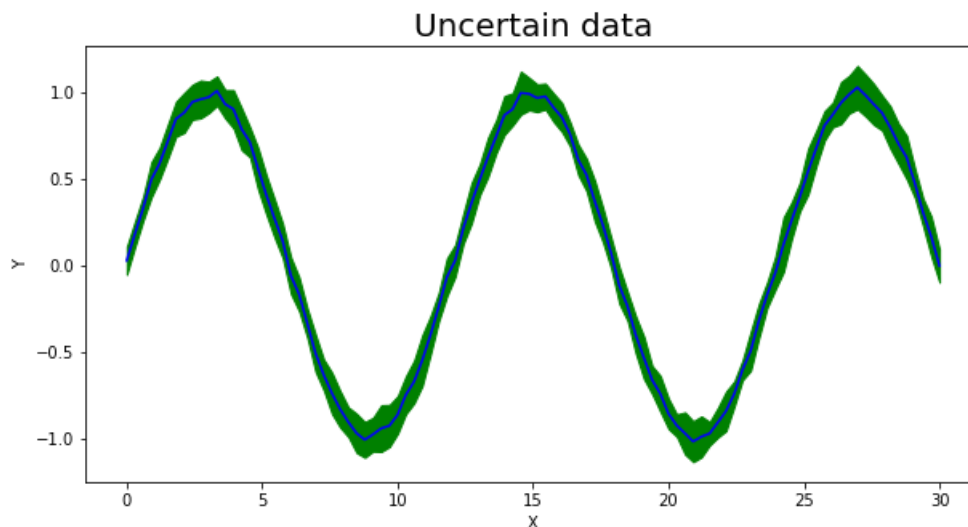
def gen_uncertain_data():
    x = np.linspace(0, 30, 100)
    y = np.sin(x/6*np.pi) + np.random.normal(0, 0.02, size=x.shape)
    error = np.random.normal(0.1, 0.02, size=y.shape)
    return x, y, error

X, Y, Error = gen_uncertain_data()
Y_plus_error = Y + Error
Y_minus_error = Y - Error

plt.figure(figsize=(10, 5))
plt.title('Uncertain data').set_size(20)
```

```
plt.plot(X, Y, color='blue')
plt.fill_between(X, Y_minus_error, Y_plus_error, color='green')

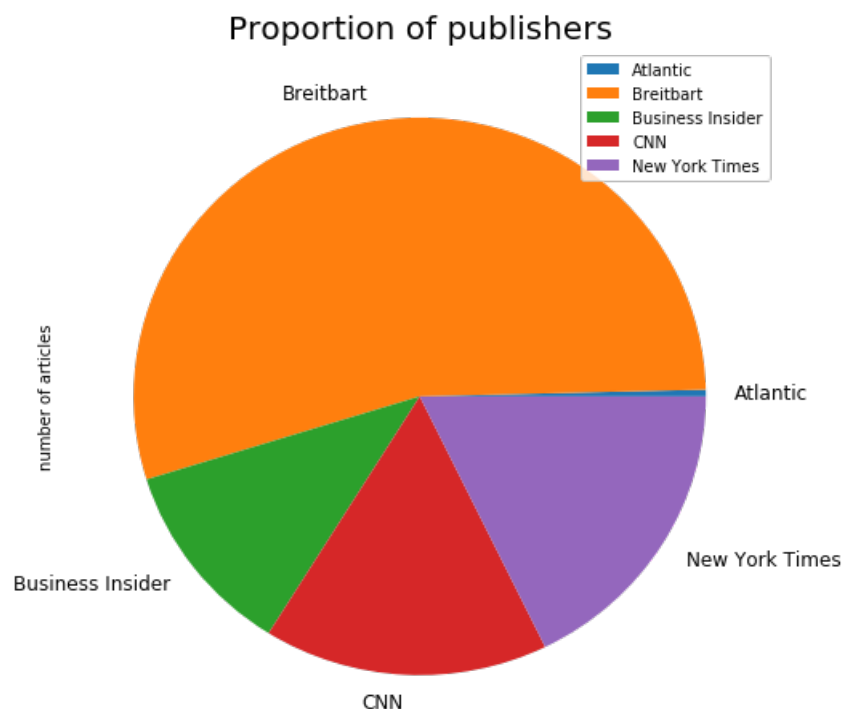
plt.xlabel('X').set_size(10)
plt.ylabel('Y').set_size(10)
plt.show()
```



(0.5 балла) **Задача 21.** Визуализируйте соотношение различных издательств, используя [pie plot](#).

In [62]:

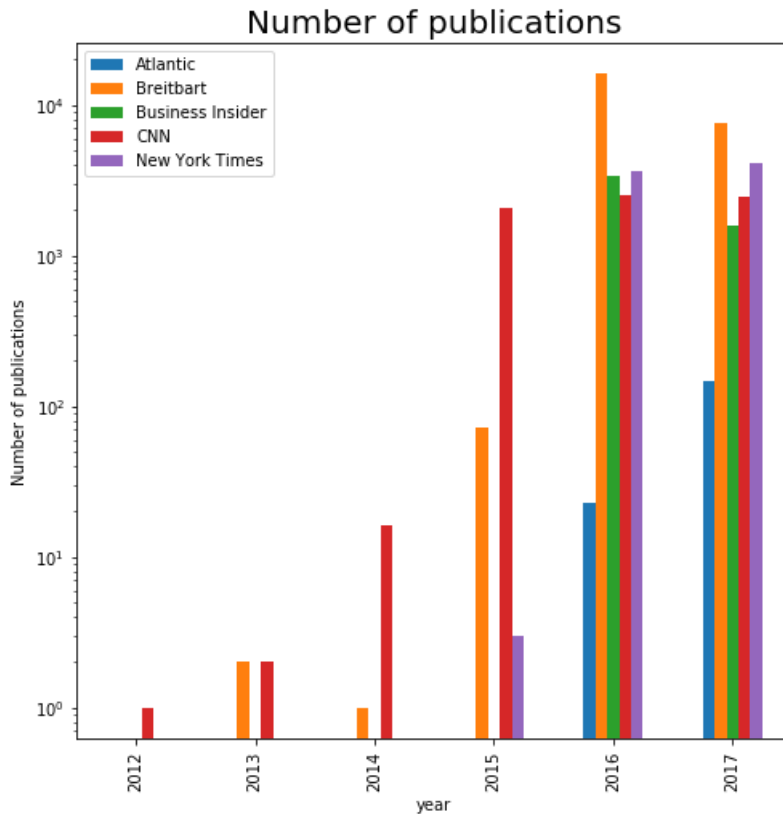
```
publishers_table = articles.groupby('publication').count().reset_index()
names = np.array(publishers_table['publication'])
publishers = pd.DataFrame({'number of articles': np.array(publishers_table['id'])},
                           index=publishers_table['publication'])
publishers.plot.pie(y='number of articles',
                    figsize=(8, 8), title="Proportion of publishers", fontsize=12).title.set_size(20)
```



(0.5 балла) **Задача 22.** Постройте распределение количества публикаций каждого издания по каждому году, используя [bar plot](#). В какой год было больше всего публикаций?

In [66]:

```
publishers_table = articles.drop(
    columns=['title', 'author', 'date', 'month', 'content']).rename(
    columns={'id': 'number of publications'}).groupby(
    ['year', 'publication']).count().unstack()
my_bar = publishers_table.plot.bar(figsize=(8, 8), title="Number of publications", fontsize=10)
my_bar.set_yscale('log')
my_bar.legend(names)
my_bar.set_ylabel('Number of publications')
my_bar.title.set_size(20)
```



**(0.5 балла) Задача 23.** Изобразите распределение длин заголовков (*title*) статей для каждого издательства. Для этого можно воспользоваться, например, функцией [sns.violinplot](#).

In [64]:

```
import seaborn as sns
publishers_table = articles.drop(
    columns=['id', 'author', 'date', 'month', 'content', 'year']).rename(columns={'title': 'title length'})
publishers_table['title length'] = publishers_table['title length'].str.len()
sns.violinplot(x="publication", y="title length", data=publishers_table).set_title(
    "Title length distribution", size=20)
```

```
/anaconda3/lib/python3.6/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple
sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`.
In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result
either in an error or a different result.
    return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[64]:

Text(0.5,1,'Title length distribution')

