МАТЕМАТИЧЕСКИЕ МЕТОДЫ ОПТИЧЕСКОЙ БИОСПЕКТРОСКОПИИ *IN VIVO*

Практикум №1 (15 февраля 2023)

Описательная статистика

Формируем массив случайных величин в Python numpy

Библиотека **numpy** имеет класс **Array**, в который мы можем записать многомерную матрицу.
Создавать матрицу можно с помощью конструктора класса, в который передаются данные.

```
>>> import numpy as np
>>> a = np.array([1, 2, 3])
>>> a
array([1, 2, 3])
```

Можно заполнить массив нулями или единицами:

```
>>> np.zeros((3, 5))
array([[ 0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.]])
>>> np.ones((2, 2, 2))
array([[[ 1.,  1.],
        [ 1.,  1.]],
       [[ 1.,  1.],
        [ 1.,  1.]]])
```

Функция empty() создает массив без его заполнения. Исходное содержимое случайно и зависит от состояния памяти на момент создания массива (то есть от того мусора, что в ней хранится):

```
>>> np.empty((3, 3))
```

https://numpy.org/doc/stable/reference/arrays.ndarray.html#constructing-arrays
https://pythonworld.ru/numpy/1.html

Для создания последовательностей чисел, в NumPy имеется функция arange(), аналогичная встроенной в Python range(), только вместо списков она возвращает массивы, и принимает не только целые значения:

```
>>> np.arange(10, 30, 5)
array([10, 15, 20, 25])
>>> np.arange(0, 1, 0.1)
array([ 0. ,  0.1,  0.2,  0.3,  0.4,  0.5,  0.6,  0.7,  0.8,  0.9])
```

Заполним массив величинами из нормального распределения:

```
>>>c = 15
>>>w = 5
>>>num = 20
>>>inp_norm = np.random.normal(c,w,num)
>>>inp_norm
array([ 9.1742703 , 13.03650447,  5.93089495, 13.04962049, 19.29911111,
       12.74087993, 10.40564479, 14.35761725, -2.36282785, 14.28766252,
       19.80453278, 10.20603218, 11.44792894, 19.61000287,  7.87949471,
        9.157177  , 23.93331462, 14.00932658,  9.38303872,  9.84051895])
```

https://numpy.org/doc/stable/reference/random/generated/numpy.random.normal.html

Основные понятия описательной статистики

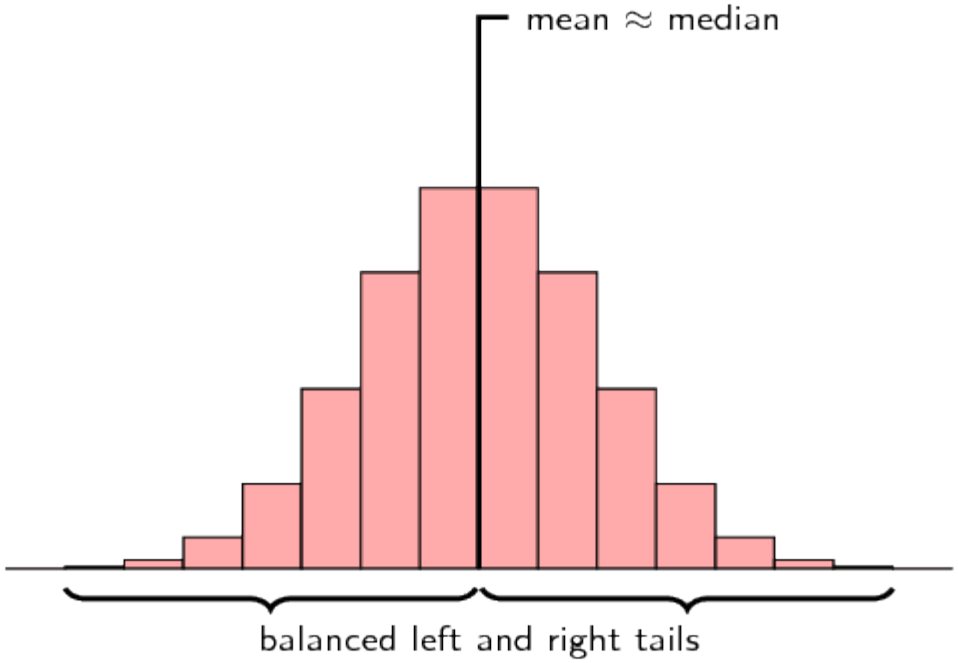
Для описания наших данных можно использовать меры центральной тенденции и меры разброса данных.
К мерам центральной тенденции относятся: **среднее** значение (среднее арифметическое выборочных данных), **медиана** (срединное значение в упорядоченном наборе данных (значение, отделяющее наибольшую половину значений от наименьшей половины значений)), **мода** (наиболее частое значение в наборе данных).

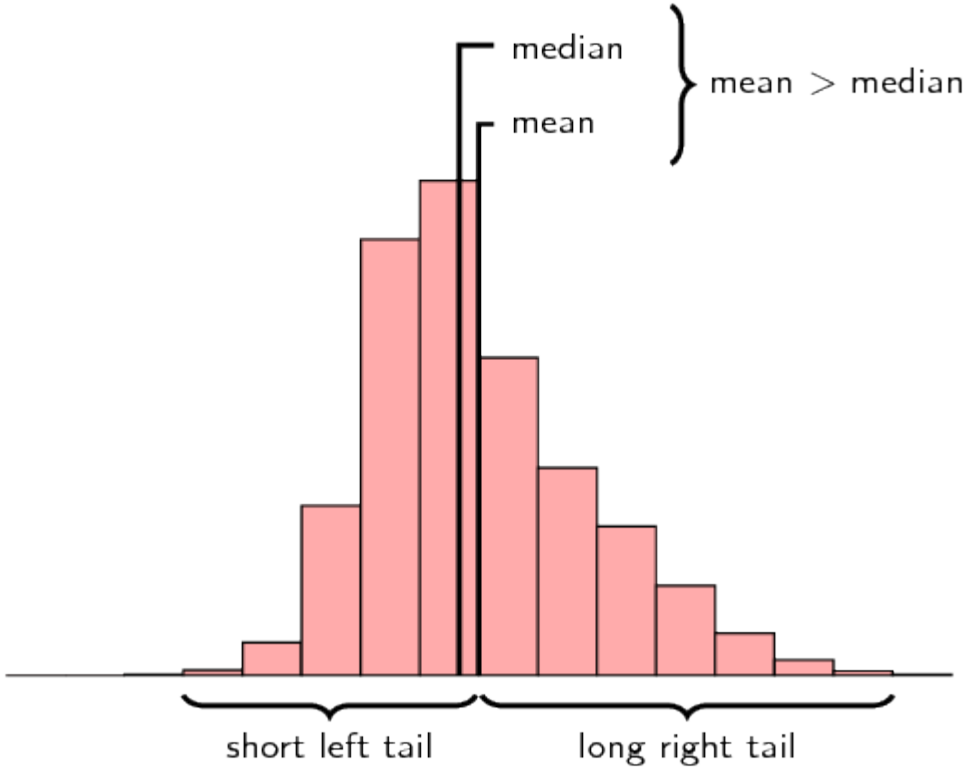
К мерам разброса данных относятся **размах** (разность наибольшего и наименьшего значений в выборке), **стандартное отклонение (корень из суммы квадратов** отклонений от среднего), **дисперсия выборки** (сумма квадратов отклонений от среднего), а также **перцентели**, отделяющие доли данных в вариационном ряду.

Асимметрия и эксцесс (мера остроты пика распределения случайной величины) характеризуют геометрическую форму распределения.

Среднее или медиана?

По форме гистограммы можно сделать уже достаточно валидные выводы о распределении данных, в частности, выбрать меры центральной тенденции и разброса. Если значения большинства измерений близки к своему среднему и с равной вероятностью отклоняются от него в большую или меньшую сторону, то само среднее значение и стандартное отклонение будут наилучшим образом описывать данные. Напротив, когда характеристические значения распределены асимметрично относительно среднего, совокупность лучше описывается с помощью медианы и перцентилей.





Описательная статистика в numpy

ptp(a[, axis, out, keepdims])	Диапазон значений (максимум-минимум) по оси.
percentile(a, q[, axis, out, ...])	Вычислить q-й процентиль данных по указанной оси.
nanpercentile(a, q[, axis, out, ...])	Вычислить q-й процентиль данных вдоль указанной оси, игнорируя значения nan.
quantile(a, q[, axis, out, overwrite_input, ...])	Вычислить q-й квантиль данных вдоль указанной оси.
nanquantile(a, q[, axis, out, ...])	Вычислите квантиль q данных вдоль заданной оси, игнорируя значения nan.

median(a[, axis, out, overwrite_input, keepdims])	Вычислить медиану вдоль указанной оси.
average(a[, axis, weights, returned, keepdims])	Вычислить средневзвешенное значение вдоль указанной оси.
mean(a[, axis, dtype, out, keepdims, where])	Вычислить среднее арифметическое вдоль указанной оси.
std(a[, axis, dtype, out, ddof, keepdims, where])	Вычислить стандартное отклонение вдоль указанной оси.
var(a[, axis, dtype, out, ddof, keepdims, where])	Вычислить дисперсию вдоль указанной оси.

https://numpy.org/doc/stable/reference/routines.statistics.html

Методы построения гистограммы

Такие методы можно разделить на две группы, в которых мы изначально задаем ширину столбцов, либо количество интервалов в области значений.
К первому методу относятся: самый простой подход, в котором ширина интервала определяется как квадратный корень из числа измерений, метод Стерджеса, определяющий ширину интервала через двоичный логарифм от числа измерений плюс единица, правило Райса, вычисляющее число интервалов через кубический корень числа измерений.

К второй группе методов можно отнести правило Скотта, которое позволяет рассчитать ширину интервала через отношение стандартного отклонения и кубического корня из числа измерений, и метод Фридмана-Диакониса, основанный на интерквартильном размахе. О квартилях мы с вами поговорим чуть позже.

	Method formula bins	Method formula width
Square-root	$\sqrt{n}$	$\frac{\max(values) - \min(values)}{\sqrt{n}}$
Sturges 1926	$\text{ceil}(\log_2 n) + 1$	$\frac{\max(values) - \min(values)}{\text{ceil}(\log_2 n) + 1}$
Rice 1944	$2 * \sqrt[3]{n}$	$\frac{\max(values) - \min(values)}{2 * \sqrt[3]{n}}$
Scott 1979	$\frac{\max(values) - \min(values)}{3.5 * \frac{stdev(values)}{\sqrt[3]{n}}}$	$3.5 * \frac{stdev(values)}{\sqrt[3]{n}}$
Freedman-Diaconis 1981	$\frac{\max(values) - \min(values)}{2 * \frac{IQR(values)}{\sqrt[3]{n}}}$	$2 * \frac{IQR(values)}{\sqrt[3]{n}}$

Построение гистограмм в numpy

histogram(a[, bins, range, density, weights])	Вычислить гистограмму набора данных.
histogram2d(x, y[, bins, range, density, ...])	Вычислить двумерную гистограмму двух выборок данных.
histogramdd(sample[, bins, range, density, ...])	Вычислите многомерную гистограмму некоторых данных.
bincount(x, /[, weights, minlength])	Подсчитайте количество вхождений каждого значения в массиве неотрицательных целых чисел.
histogram_bin_edges(a[, bins, range, weights])	Функция для вычисления только краев интервалов, используемых функцией гистограммы.
digitize(x, bins[, right])	Возвращает индексы интервалов, которым принадлежит каждое значение во входном массиве.
numpy.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None, axis=0)[source]	Возвращает равномерно распределенные числа за указанный интервал. Возвращает число равноотстоящих выборок, рассчитанных по интервалу [начало, стоп]. Конечная точка интервала может быть дополнительно исключена.

https://numpy.org/doc/stable/reference/routines.statistics.html

Параметры построения гистограмм в numpy

numpy.histogram(a, bins=10, range=None, density=None, weights=None)
Parameters:
a array_like Input data. The histogram is computed over the flattened array.
bins int or sequence of scalars or str, optional
If bins is an int, it defines the number of equal-width bins in the given range (10, by default). If bins is a sequence, it defines a monotonically increasing array of bin edges, including the rightmost edge, allowing for non-uniform bin widths.
If bins is a string, it defines the method used to calculate the optimal bin width, as defined by histogram_bin_edges.
range(float, float), optional
The lower and upper range of the bins. If not provided, range is simply (a.min(), a.max()). Values outside the range are ignored. The first element of the range must be less than or equal to the second. range affects the automatic bin computation as well. While bin width is computed to be optimal based on the actual data within range, the bin count will fill the entire range including portions containing no data.
weights array_like, optional
An array of weights, of the same shape as a. Each value in a only contributes its associated weight towards the bin count (instead of 1). If density is True, the weights are normalized, so that the integral of the density over the range remains 1.
density bool, optional
If False, the result will contain the number of samples in each bin. If True, the result is the value of the probability density function at the bin, normalized such that the integral over the range is 1. Note that the sum of the histogram values will not be equal to 1 unless bins of unity width are chosen; it is not a probability mass function.

https://numpy.org/doc/stable/reference/routines.statistics.html

numpy.histogram_bin_edges

numpy.histogram_bin_edges(a, bins=10, range=None, weights=None)

a - массив NumPy или подобный массиву объект.

bins - целое число, последовательность целых чисел или строка (необязательный параметр).
Если указано целое число, то оно определяет количество интервалов равной ширины всех ячеек (по умолчанию 10 ячеек). Если указана последовательность целых чисел, то границы интервалов определяют соседние числа в данной последовательности.

В NumPy начиная с версии 1.11.0 в качестве данного параметра можно указывать строку с названием метода расчета оптимальной ширины ячеек.

'auto' - максимальные значения 'sturges' и 'fd'. Обеспечивает наилучшую производительность;
'fd' - метод оценки Фридмана-Диакониса, который учитывает размер данных и их изменчивость. Устойчив к выбросам и считается наиболее надежным;
'doane' - улучшенная версия 'sturges', которая лучше всего подходит для данных с ненормальным распределением значений;
'scott' - метод, который учитывает изменчивость и размер данных, но является менее надежным;
'rice' - учитывает только размер данных и обычно переоценивает количество необходимых ячеек;
'sturges' - R-метод, который учитывает только размер данных. Оптимален только для данных с гаусовым распределением значений. Для больших негаусовых данных недооценивается необходимое количество ячеек;
'sqrt' - метод на основе квадратного корня от размера данных, самый быстрый и простой метод, но может подойти не для всех данных.

range - (float_1, float_2) (необязательный параметр).
Определяет минимальное и максимальное значение ширины ячеек, при этом значения выходящие за пределы диапазона игнорируются. Если range = None то границы определяются интервалом (a.min(), a.max()). Должно выполняться условие float_1 < float_2.

Возвращает:
результат - массив NumPy
Массив с границами ячеек для подсчета одномерной гистограммы исходных данных.

https://pyprog.pro/statistics_functions/histogram_bin_edges.html

Отрисовка гистограмм в matplotlib

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

matplotlib.pyplot.hist(x, bins=None, range=None, density=False, weights=None, cumulative=False, bottom=None, histtype='bar', align='mid', orientation='vertical', rwidth=None, log=False, color=None, label=None, stacked=False, *, data=None, **kwargs)[source]

Этот метод использует numpy.histogram для группировки данных по x и подсчета количества значений в каждой ячейке, а затем рисует распределение в виде BarContainer или Polygon. Параметры ячеек, диапазона, плотности и веса передаются в numpy.histogram.

СПАСИБО ЗА ВНИМАНИЕ!