



[Scipy.org \(https://scipy.org/\)](https://scipy.org/) [Docs \(https://docs.scipy.org/\)](https://docs.scipy.org/)

[SciPy v1.1.0 Reference Guide \(../index.html\)](#)

[Statistical functions \(**scipy.stats** \) \(../stats.html\)](#)

[index \(../genindex.html\)](#) [modules \(../py-modindex.html\)](#)

[next \(scipy.stats.binned_statistic_dd.html\)](#) [previous \(scipy.stats.binned_statistic.html\)](#)

scipy.stats.binned_statistic_2d

[Previous topic](#)

`scipy.stats.binned_statistic_2d(x, y, values, statistic='mean', bins=10, range=None, expand_binnumbers=False)` (https://github.com/scipy/scipy/blob/v1.1.0/scipy/stats/_binned_statistic.py#L189-L346) [\[source\]](#)

[scipy.stats.binned_statistic_1d \(scipy.stats.binned_statistic_1d.html\)](#)

[Next topic](#)

Compute a bidimensional binned statistic for one or more sets of data. [scipy.stats.binned_statistic_1d \(scipy.stats.binned_statistic_1d.html\)](#)

This is a generalization of a `histogram2d` function. A histogram divides the space into bins, and returns the count of the number of points in each bin. This function allows the computation of the sum, mean, median, or other statistic of the values (or set of values) within each bin.

A sequence of values to be binned along the first dimension.

y : (N,) array_like

A sequence of values to be binned along the second dimension.

values : (N,) array_like or list of (N,) array_like

The data on which the statistic will be computed. This must be the same shape as *x*, or a list of sequences - each with the same shape as *x*. If *values* is such a list, the statistic will be computed on each independently.

statistic : string or callable, optional

The statistic to compute (default is 'mean').

The following statistics are available:

- 'mean' : compute the mean of values for points within each bin. Empty bins will be represented by NaN.
- 'median' : compute the median of values for points within each bin. Empty bins will be represented by NaN.
- 'count' : compute the count of points within each bin. This is identical to an unweighted histogram. *values* array is not referenced.
- 'sum' : compute the sum of values for points within each bin. This is identical to a weighted histogram.
- 'min' : compute the minimum of values for points within each bin. Empty bins will be represented by NaN.
- 'max' : compute the maximum of values for point within each bin. Empty bins will be represented by NaN.
- function : a user-defined function which takes a 1D array of values, and outputs a single numerical statistic. This function will be called on the values in each bin. Empty bins will be represented by function([]), or NaN if this returns an error.

bins : int or [int, int] or array_like or [array, array], optional

The bin specification:

- the number of bins in each dimension (nx = ny = bins),
- the number of bins in each dimension (nx, ny = bins),
- the bin edges for the two dimensions (x_edge = y_edge = bins),
- the bin edges in each dimension (x_edge, y_edge = bins).

If the bin edges are specified, the number of bins will be, (nx = len(x_edge)-1, ny = len(y_edge)-1).

range : (2,2) array_like, optional

The leftmost and rightmost edges of the bins along each dimension (if not specified explicitly in the *bins* parameters): [[xmin, xmax], [ymin, ymax]]. All values outside of this range will be considered outliers and not tallied in the histogram.

expand_binnumbers : bool, optional

'False' (default): the returned *binnumber* is a shape (N,) array of linearized bin indices.
'True': the returned *binnumber* is 'unraveled' into a shape (2,N) ndarray, where each row gives the bin numbers in the corresponding dimension. See the *binnumber* returned value, and the *Examples* section.

New in version 0.17.0.

Returns:

statistic : (nx, ny) ndarray

The values of the selected statistic in each two-dimensional bin.

x_edge : (nx + 1) ndarray

The bin edges along the first dimension.

y_edge : (ny + 1) ndarray

The bin edges along the second dimension.

binnumber : (N,) array of ints or (2,N) ndarray of ints

This assigns to each element of *sample* an integer that represents the bin in which this observation falls. The representation depends on the *expand_binnumbers* argument. See *Notes* for details.

See also:

[numpy.digitize](https://docs.scipy.org/doc/numpy/reference/generated/numpy.digitize.html#numpy.digitize) (<https://docs.scipy.org/doc/numpy/reference/generated/numpy.digitize.html#numpy.digitize>),

[numpy.histogram2d](https://docs.scipy.org/doc/numpy/reference/generated/numpy.histogram2d.html#numpy.histogram2d) (<https://docs.scipy.org/doc/numpy/reference/generated/numpy.histogram2d.html#numpy.histogram2d>),

[binned_statistic](https://docs.scipy.org/doc/scipy/stats/generated/scipy.stats.binned_statistic.html#scipy.stats.binned_statistic)

([scipy.stats.binned_statistic.html#scipy.stats.binned_statistic](https://docs.scipy.org/doc/scipy/stats/generated/scipy.stats.binned_statistic.html#scipy.stats.binned_statistic)), [binned_statistic_dd](https://docs.scipy.org/doc/scipy/stats/generated/scipy.stats.binned_statistic_dd.html#scipy.stats.binned_statistic_dd)

Notes

Binedges: All but the last (righthand-most) bin is half-open. In other words, if *bins* is [1, 2, 3, 4], then the first bin is [1, 2) (including 1, but excluding 2) and the second [2, 3). The last bin, however, is [3, 4], which *includes* 4.

***binnumber*:** This returned argument assigns to each element of *sample* an integer that represents the bin in which it belongs. The representation depends on the *expand_binnumbers* argument. If 'False' (default): The returned *binnumber* is a shape (N,) array of linearized indices mapping each element of *sample* to its corresponding bin (using row-major ordering). If 'True': The returned *binnumber* is a shape (2,N) ndarray where each row indicates bin placements for each dimension respectively. In each dimension, a binnumber of *i* means the corresponding value is between (D_edge[i-1], D_edge[i]), where 'D' is either 'x' or 'y'.

New in version 0.11.0.

Examples

```
>>> from scipy import stats
```

```
>>>
```

Calculate the counts with explicit bin-edges:

```
>>> x = [0.1, 0.1, 0.1, 0.6]
>>> y = [2.1, 2.6, 2.1, 2.1]
>>> binx = [0.0, 0.5, 1.0]
>>> biny = [2.0, 2.5, 3.0]
>>> ret = stats.binned_statistic_2d(x, y, None, 'count',
>>> , bins=[binx,biny])
>>> ret.statistic
array([[ 2.,  1.],
       [ 1.,  0.]])
```

The bin in which each sample is placed is given by the *binnumber* returned parameter. By default, these are the linearized bin indices:

```
>>> ret.binnumber
array([5, 6, 5, 9])
```

```
>>>
```

The bin indices can also be expanded into separate entries for each dimension using the *expand_binnumbers* parameter:

```
>>> ret = stats.binned_statistic_2d(x, y, None, 'count',
>>> , bins=[binx,biny],
>>> ... expand_binnumbers=True)
>>> ret.binnumber
array([[1, 1, 1, 2],
       [1, 2, 1, 1]])
```

scipy.stats.binned_statistic_2d → See y v1.1.0 re: <https://docs.scipy.org/doc/scipy/reference/gener...>
Which shows that the first three elements belong to xbin 1 and
the fourth into xbin 2; and so on for y.