:mod:`statistics` --- Mathematical statistics functions

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 4)

Unknown directive type "module".

```
.. module:: statistics
   :synopsis: Mathematical statistics functions
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 7)

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Steven D'Aprano <steve+python@pearwood.info>
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 8)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Steven D'Aprano <steve+python@pearwood.info>
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 10)

Unknown directive type "versionadded".

```
.. versionadded:: 3.4
```

Source code: :source:`Lib/statistics.py`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 12); *backlink*

Unknown interpreted text role "source".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 14)

Unknown directive type "testsetup".

```
.. testsetup:: *

   from statistics import *
   __name__ = '<doctest>'
```

This module provides functions for calculating mathematical statistics of numeric (:class:`~numbers.Real`-valued) data.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 21); *backlink*

Unknown interpreted text role "class".

The module is not intended to be a competitor to third-party libraries such as NumPy, SciPy, or proprietary full-featured statistics packages aimed at professional statisticians such as Minitab, SAS and Matlab. It is aimed at the level of graphing and scientific calculators.

Unless explicitly noted, these functions support :class:`int`, :class:`float`, :class:`~decimal.Decimal` and :class:`~fractions.Fraction`. Behaviour with other types (whether in the numeric tower or not) is currently unsupported. Collections with a mix of types are also undefined and implementation-dependent. If your input data consists of mixed types, you may be able to use :func:`map` to ensure a consistent result, for example: `map(float, input_data)`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 30); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 30); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 30); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 30); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 30); *backlink*

Unknown interpreted text role "func".

Averages and measures of central location

These functions calculate an average or typical value from a population or sample.

:func:`mean` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 46); *backlink*** Unknown interpreted text role "func".	Arithmetic mean ("average") of data.

:func:`fmean` **System Message: ERROR/3** (**D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst**, **line 47**); ***backlink*** Unknown interpreted text role "func".	Fast, floating point arithmetic mean, with optional weighting.
:func:`geometric_mean` **System Message: ERROR/3** (**D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst**, **line 48**); ***backlink*** Unknown interpreted text role "func".	Geometric mean of data.
:func:`harmonic_mean` **System Message: ERROR/3** (**D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst**, **line 49**); ***backlink*** Unknown interpreted text role "func".	Harmonic mean of data.
:func:`median` **System Message: ERROR/3** (**D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst**, **line 50**); ***backlink*** Unknown interpreted text role "func".	Median (middle value) of data.

:func:`median_low` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 51); *backlink* Unknown interpreted text role "func".	Low median of data.
:func:`median_high` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 52); *backlink* Unknown interpreted text role "func".	High median of data.
:func:`median_grouped` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 53); *backlink* Unknown interpreted text role "func".	Median, or 50th percentile, of grouped data.
:func:`mode` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 54); *backlink* Unknown interpreted text role "func".	Single mode (most common value) of discrete or nominal data.

:func:`multimode` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 55); *backlink* Unknown interpreted text role "func".	List of modes (most common values) of discrete or nominal data.
:func:`quantiles` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 56); *backlink* Unknown interpreted text role "func".	Divide data into intervals with equal probability.

Measures of spread

These functions calculate a measure of how much the population or sample tends to deviate from the typical or average values.

:func:`pstdev` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 66); *backlink* Unknown interpreted text role "func".	Population standard deviation of data.

:func:`pvariance` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 67); *backlink* Unknown interpreted text role "func".	Population variance of data.
:func:`stdev` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 68); *backlink* Unknown interpreted text role "func".	Sample standard deviation of data.
:func:`variance` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 69); *backlink* Unknown interpreted text role "func".	Sample variance of data.

Statistics for relations between two inputs

These functions calculate statistics regarding relations between two inputs.

:func:`covariance` **System Message: ERROR/3** (**D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst**, **line 78**); ***backlink*** Unknown interpreted text role "func".	Sample covariance for two variables.
:func:`correlation` **System Message: ERROR/3** (**D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst**, **line 79**); ***backlink*** Unknown interpreted text role "func".	Pearson's correlation coefficient for two variables.
:func:`linear_regression` **System Message: ERROR/3** (**D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst**, **line 80**); ***backlink*** Unknown interpreted text role "func".	Slope and intercept for simple linear regression.

Function details

Note: The functions do not require the data given to them to be sorted. However, for reading convenience, most of the examples show sorted sequences.

System Message: ERROR/3 (**D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst**, **line 89**)

Unknown directive type "function".

```
.. function:: mean(data)

   Return the sample arithmetic mean of *data* which can be a sequence or iterable.

   The arithmetic mean is the sum of the data divided by the number of data
   points.  It is commonly called "the average", although it is only one of many
   different mathematical averages.  It is a measure of the central location of
   the data.
   If *data* is empty, :exc:`StatisticsError` will be raised.

   Some examples of use:

   .. doctest::

      >>> mean([1, 2, 3, 4, 4])
      2.8
      >>> mean([-1.0, 2.5, 3.25, 5.75])
      2.625

      >>> from fractions import Fraction as F
      >>> mean([F(3, 7), F(1, 21), F(5, 3), F(1, 3)])
      Fraction(13, 21)

      >>> from decimal import Decimal as D
      >>> mean([D("0.5"), D("0.75"), D("0.625"), D("0.375")])
      Decimal('0.5625')

   .. note::

      The mean is strongly affected by `outliers
      <https://en.wikipedia.org/wiki/Outlier>`_ and is not necessarily a
      typical example of the data points. For a more robust, although less
      efficient, measure of `central tendency
      <https://en.wikipedia.org/wiki/Central_tendency>`_, see :func:`median`.

      The sample mean gives an unbiased estimate of the true population mean,
      so that when taken on average over all the possible samples,
      ``mean(sample)`` converges on the true mean of the entire population.  If
      *data* represents the entire population rather than a sample, then
      ``mean(data)`` is equivalent to calculating the true population mean Î¼.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 132)

Unknown directive type "function".

```
.. function:: fmean(data, weights=None)

   Convert *data* to floats and compute the arithmetic mean.

   This runs faster than the :func:`mean` function and it always returns a
   :class:`float`.  The *data* may be a sequence or iterable.  If the input
   dataset is empty, raises a :exc:`StatisticsError`.

   .. doctest::

      >>> fmean([3.5, 4.0, 5.25])
      4.25

   Optional weighting is supported.  For example, a professor assigns a
   grade for a course by weighting quizzes at 20%, homework at 20%, a
   midterm exam at 30%, and a final exam at 30%:

   .. doctest::

      >>> grades = [85, 92, 83, 91]
      >>> weights = [0.20, 0.20, 0.30, 0.30]
      >>> fmean(grades, weights)
      87.6

   If *weights* is supplied, it must be the same length as the *data* or
   a :exc:`ValueError` will be raised.

   .. versionadded:: 3.8

   .. versionchanged:: 3.11
      Added support for *weights*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 165)

Unknown directive type "function".

```
.. function:: geometric_mean(data)

   Convert *data* to floats and compute the geometric mean.

   The geometric mean indicates the central tendency or typical value of the
   *data* using the product of the values (as opposed to the arithmetic mean
   which uses their sum).

   Raises a :exc:`StatisticsError` if the input dataset is empty,
   if it contains a zero, or if it contains a negative value.
   The *data* may be a sequence or iterable.

   No special efforts are made to achieve exact results.
   (However, this may change in the future.)

   .. doctest::

      >>> round(geometric_mean([54, 24, 36]), 1)
      36.0

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]statistics.rst, line 188)

Unknown directive type "function".

```
.. function:: harmonic_mean(data, weights=None)

   Return the harmonic mean of *data*, a sequence or iterable of
   real-valued numbers.  If *weights* is omitted or *None*, then
   equal weighting is assumed.

   The harmonic mean is the reciprocal of the arithmetic :func:`mean` of the
   reciprocals of the data. For example, the harmonic mean of three values *a*,
   *b* and *c* will be equivalent to ``3/(1/a + 1/b + 1/c)``.  If one of the
   values is zero, the result will be zero.

   The harmonic mean is a type of average, a measure of the central
   location of the data.  It is often appropriate when averaging
   ratios or rates, for example speeds.

   Suppose a car travels 10 km at 40 km/hr, then another 10 km at 60 km/hr.
   What is the average speed?

   .. doctest::

      >>> harmonic_mean([40, 60])
      48.0

   Suppose a car travels 40 km/hr for 5 km, and when traffic clears,
   speeds-up to 60 km/hr for the remaining 30 km of the journey. What
   is the average speed?

   .. doctest::

      >>> harmonic_mean([40, 60], weights=[5, 30])
      56.0

   :exc:`StatisticsError` is raised if *data* is empty, any element
   is less than zero, or if the weighted sum isn't positive.

   The current algorithm has an early-out when it encounters a zero
   in the input.  This means that the subsequent inputs are not tested
   for validity.  (This behavior may change in the future.)

   .. versionadded:: 3.6

   .. versionchanged:: 3.10
      Added support for *weights*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]statistics.rst, line 232)

Unknown directive type "function".

```
.. function:: median(data)
```

```
   Return the median (middle value) of numeric data, using the common "mean of
   middle two" method.  If *data* is empty, :exc:`StatisticsError` is raised.
   *data* can be a sequence or iterable.

   The median is a robust measure of central location and is less affected by
   the presence of outliers.  When the number of data points is odd, the
   middle data point is returned:

   .. doctest::

      >>> median([1, 3, 5])
      3

   When the number of data points is even, the median is interpolated by taking
   the average of the two middle values:

   .. doctest::

      >>> median([1, 3, 5, 7])
      4.0

   This is suited for when your data is discrete, and you don't mind that the
   median may not be an actual data point.

   If the data is ordinal (supports order operations) but not numeric (doesn't
   support addition), consider using :func:`median_low` or :func:`median_high`
   instead.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 262)

Unknown directive type "function".

```
.. function:: median_low(data)

   Return the low median of numeric data.  If *data* is empty,
   :exc:`StatisticsError` is raised.  *data* can be a sequence or iterable.

   The low median is always a member of the data set.  When the number of data
   points is odd, the middle value is returned.  When it is even, the smaller of
   the two middle values is returned.

   .. doctest::

      >>> median_low([1, 3, 5])
      3
      >>> median_low([1, 3, 5, 7])
      3

   Use the low median when your data are discrete and you prefer the median to
   be an actual data point rather than interpolated.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 282)

Unknown directive type "function".

```
.. function:: median_high(data)

   Return the high median of data.  If *data* is empty, :exc:`StatisticsError`
   is raised.  *data* can be a sequence or iterable.

   The high median is always a member of the data set.  When the number of data
   points is odd, the middle value is returned.  When it is even, the larger of
   the two middle values is returned.

   .. doctest::

      >>> median_high([1, 3, 5])
      3
      >>> median_high([1, 3, 5, 7])
      5

   Use the high median when your data are discrete and you prefer the median to
   be an actual data point rather than interpolated.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 302)

Unknown directive type "function".

```
.. function:: median_grouped(data, interval=1)

   Return the median of grouped continuous data, calculated as the 50th
   percentile, using interpolation.  If *data* is empty, :exc:`StatisticsError`
   is raised.  *data* can be a sequence or iterable.

   .. doctest::

      >>> median_grouped([52, 52, 53, 54])
      52.5

   In the following example, the data are rounded, so that each value represents
   the midpoint of data classes, e.g. 1 is the midpoint of the class 0.5--1.5, 2
   is the midpoint of 1.5--2.5, 3 is the midpoint of 2.5--3.5, etc.  With the data
   given, the middle value falls somewhere in the class 3.5--4.5, and
   interpolation is used to estimate it:

   .. doctest::

      >>> median_grouped([1, 2, 2, 3, 4, 4, 4, 4, 4, 5])
      3.7

   Optional argument *interval* represents the class interval, and defaults
   to 1.  Changing the class interval naturally will change the interpolation:

   .. doctest::

      >>> median_grouped([1, 3, 3, 5, 7], interval=1)
      3.25
      >>> median_grouped([1, 3, 3, 5, 7], interval=2)
      3.5

   This function does not check whether the data points are at least
   *interval* apart.

   .. impl-detail::

      Under some circumstances, :func:`median_grouped` may coerce data points to
      floats.  This behaviour is likely to change in the future.

   .. seealso::

      * "Statistics for the Behavioral Sciences", Frederick J Gravetter and
        Larry B Wallnau (8th Edition).

      * The `SSMEDIAN
        <https://help.gnome.org/users/gnumeric/stable/gnumeric.html#gnumeric-function-SSMEDIAN>`_
        function in the Gnome Gnumeric spreadsheet, including `this discussion
        <https://mail.gnome.org/archives/gnumeric-list/2011-April/msg00018.html>`_.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 353)

Unknown directive type "function".

```
.. function:: mode(data)

   Return the single most common data point from discrete or nominal *data*.
   The mode (when it exists) is the most typical value and serves as a
   measure of central location.

   If there are multiple modes with the same frequency, returns the first one
   encountered in the *data*.  If the smallest or largest of those is
   desired instead, use ``min(multimode(data))`` or ``max(multimode(data))``.
   If the input *data* is empty, :exc:`StatisticsError` is raised.

   ``mode`` assumes discrete data and returns a single value. This is the
   standard treatment of the mode as commonly taught in schools:

   .. doctest::
      >>> mode([1, 1, 2, 3, 3, 3, 3, 4])
      3

   The mode is unique in that it is the only statistic in this package that
   also applies to nominal (non-numeric) data:

   .. doctest::

      >>> mode(["red", "blue", "blue", "red", "green", "red", "red"])
      'red'

   .. versionchanged:: 3.8
      Now handles multimodal datasets by returning the first mode encountered.
      Formerly, it raised :exc:`StatisticsError` when more than one mode was
      found.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 386)

Unknown directive type "function".

```
.. function:: multimode(data)

   Return a list of the most frequently occurring values in the order they
   were first encountered in the *data*.  Will return more than one result if
   there are multiple modes or an empty list if the *data* is empty:

   .. doctest::

        >>> multimode('aabbbbccddddeeffffgg')
        ['b', 'd', 'f']
        >>> multimode('')
        []

   .. versionadded:: 3.8
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 402)

Unknown directive type "function".

```
.. function:: pstdev(data, mu=None)

   Return the population standard deviation (the square root of the population
   variance).  See :func:`pvariance` for arguments and other details.

   .. doctest::

      >>> pstdev([1.5, 2.5, 2.5, 2.75, 3.25, 4.75])
      0.986893273527251
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 413)

Unknown directive type "function".

```
.. function:: pvariance(data, mu=None)

   Return the population variance of *data*, a non-empty sequence or iterable
   of real-valued numbers.  Variance, or second moment about the mean, is a
   measure of the variability (spread or dispersion) of data.  A large
   variance indicates that the data is spread out; a small variance indicates
   it is clustered closely around the mean.

   If the optional second argument *mu* is given, it is typically the mean of
   the *data*.  It can also be used to compute the second moment around a
   point that is not the mean.  If it is missing or ``None`` (the default),
   the arithmetic mean is automatically calculated.

   Use this function to calculate the variance from the entire population.  To
   estimate the variance from a sample, the :func:`variance` function is usually
   a better choice.
   Raises :exc:`StatisticsError` if *data* is empty.

   Examples:

   .. doctest::

      >>> data = [0.0, 0.25, 0.25, 1.25, 1.5, 1.75, 2.75, 3.25]
      >>> pvariance(data)
      1.25

   If you have already calculated the mean of your data, you can pass it as the
   optional second argument *mu* to avoid recalculation:

   .. doctest::

      >>> mu = mean(data)
      >>> pvariance(data, mu)
      1.25

   Decimals and Fractions are supported:

   .. doctest::

      >>> from decimal import Decimal as D
      >>> pvariance([D("27.5"), D("30.25"), D("30.25"), D("34.5"), D("41.75")])
      Decimal('24.815')

      >>> from fractions import Fraction as F
      >>> pvariance([F(1, 4), F(5, 4), F(1, 2)])
      Fraction(13, 72)

   .. note::

      When called with the entire population, this gives the population variance
      ÏƒÂ².  When called on a sample instead, this is the biased sample variance
      sÂ², also known as variance with N degrees of freedom.

      If you somehow know the true population mean Î¼, you may use this
      function to calculate the variance of a sample, giving the known
      population mean as the second argument.  Provided the data points are a
      random sample of the population, the result will be an unbiased estimate
      of the population variance.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 474)

Unknown directive type "function".

```
.. function:: stdev(data, xbar=None)

   Return the sample standard deviation (the square root of the sample
   variance).  See :func:`variance` for arguments and other details.

   .. doctest::

      >>> stdev([1.5, 2.5, 2.5, 2.75, 3.25, 4.75])
      1.0810874155219827
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 485)

Unknown directive type "function".

```
.. function:: variance(data, xbar=None)

   Return the sample variance of *data*, an iterable of at least two real-valued
   numbers.  Variance, or second moment about the mean, is a measure of the
   variability (spread or dispersion) of data.  A large variance indicates that
   the data is spread out; a small variance indicates it is clustered closely
   around the mean.

   If the optional second argument *xbar* is given, it should be the mean of
   *data*.  If it is missing or ``None`` (the default), the mean is
   automatically calculated.

   Use this function when your data is a sample from a population. To calculate
   the variance from the entire population, see :func:`pvariance`.

   Raises :exc:`StatisticsError` if *data* has fewer than two values.

   Examples:

   .. doctest::

      >>> data = [2.75, 1.75, 1.25, 0.25, 0.5, 1.25, 3.5]
      >>> variance(data)
      1.3720238095238095

   If you have already calculated the mean of your data, you can pass it as the
   optional second argument *xbar* to avoid recalculation:

   .. doctest::

      >>> m = mean(data)
      >>> variance(data, m)
      1.3720238095238095

   This function does not attempt to verify that you have passed the actual mean
   as *xbar*.  Using arbitrary values for *xbar* can lead to invalid or
   impossible results.

   Decimal and Fraction values are supported:

   .. doctest::

      >>> from decimal import Decimal as D
      >>> variance([D("27.5"), D("30.25"), D("30.25"), D("34.5"), D("41.75")])
      Decimal('31.01875')

      >>> from fractions import Fraction as F
      >>> variance([F(1, 6), F(1, 2), F(5, 3)])
      Fraction(67, 108)

   .. note::

      This is the sample variance sÂ² with Bessel's correction, also known as
      variance with N-1 degrees of freedom.  Provided that the data points are
      representative (e.g. independent and identically distributed), the result
      should be an unbiased estimate of the true population variance.

      If you somehow know the actual population mean Î¼ you should pass it to the
      :func:`pvariance` function as the *mu* parameter to get the variance of a
      sample.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 546)

Unknown directive type "function".

```
.. function:: quantiles(data, *, n=4, method='exclusive')

   Divide *data* into *n* continuous intervals with equal probability.
   Returns a list of ``n - 1`` cut points separating the intervals.

   Set *n* to 4 for quartiles (the default).  Set *n* to 10 for deciles.  Set
   *n* to 100 for percentiles which gives the 99 cuts points that separate
   *data* into 100 equal sized groups.  Raises :exc:`StatisticsError` if *n*
   is not least 1.

   The *data* can be any iterable containing sample data.  For meaningful
   results, the number of data points in *data* should be larger than *n*.
   Raises :exc:`StatisticsError` if there are not at least two data points.

   The cut points are linearly interpolated from the
   two nearest data points.  For example, if a cut point falls one-third
   of the distance between two sample values, ``100`` and ``112``, the
   cut-point will evaluate to ``104``.

   The *method* for computing quantiles can be varied depending on
   whether the *data* includes or excludes the lowest and
   highest possible values from the population.

   The default *method* is "exclusive" and is used for data sampled from
   a population that can have more extreme values than found in the
   samples.  The portion of the population falling below the *i-th* of
   *m* sorted data points is computed as ``i / (m + 1)``.  Given nine
   sample values, the method sorts them and assigns the following
   percentiles: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%.

   Setting the *method* to "inclusive" is used for describing population
   data or for samples that are known to include the most extreme values
   from the population.  The minimum value in *data* is treated as the 0th
   percentile and the maximum value is treated as the 100th percentile.
   The portion of the population falling below the *i-th* of *m* sorted
   data points is computed as ``(i - 1) / (m - 1)``.  Given 11 sample
   values, the method sorts them and assigns the following percentiles:
   0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100%.

   .. doctest::

        # Decile cut points for empirically sampled data
        >>> data = [105, 129, 87, 86, 111, 111, 89, 81, 108, 92, 110,
        ...         100, 75, 105, 103, 109, 76, 119, 99, 91, 103, 129,
        ...         106, 101, 84, 111, 74, 87, 86, 103, 103, 106, 86,
        ...         111, 75, 87, 102, 121, 111, 88, 89, 101, 106, 95,
        ...         103, 107, 101, 81, 109, 104]
        >>> [round(q, 1) for q in quantiles(data, n=10)]
        [81.0, 86.2, 89.0, 99.4, 102.5, 103.6, 106.0, 109.8, 111.0]

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 598)

Unknown directive type "function".

```
.. function:: covariance(x, y, /)

   Return the sample covariance of two inputs *x* and *y*. Covariance
   is a measure of the joint variability of two inputs.

   Both inputs must be of the same length (no less than two), otherwise
   :exc:`StatisticsError` is raised.

   Examples:

   .. doctest::

      >>> x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
      >>> y = [1, 2, 3, 1, 2, 3, 1, 2, 3]
      >>> covariance(x, y)
      0.75
      >>> z = [9, 8, 7, 6, 5, 4, 3, 2, 1]
      >>> covariance(x, z)
      -7.5
      >>> covariance(z, x)
      -7.5

   .. versionadded:: 3.10
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 622)

Unknown directive type "function".

```
.. function:: correlation(x, y, /)

   Return the `Pearson's correlation coefficient
   <https://en.wikipedia.org/wiki/Pearson_correlation_coefficient>`_
   for two inputs. Pearson's correlation coefficient *r* takes values
   between -1 and +1. It measures the strength and direction of the linear
   relationship, where +1 means very strong, positive linear relationship,
   -1 very strong, negative linear relationship, and 0 no linear relationship.

   Both inputs must be of the same length (no less than two), and need
   not to be constant, otherwise :exc:`StatisticsError` is raised.

   Examples:

   .. doctest::

      >>> x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
      >>> y = [9, 8, 7, 6, 5, 4, 3, 2, 1]
      >>> correlation(x, x)
      1.0
      >>> correlation(x, y)
      -1.0

   .. versionadded:: 3.10
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 647)

Unknown directive type "function".

```
.. function:: linear_regression(x, y, /, *, proportional=False)

   Return the slope and intercept of `simple linear regression
   <https://en.wikipedia.org/wiki/Simple_linear_regression>`_
   parameters estimated using ordinary least squares. Simple linear
   regression describes the relationship between an independent variable *x* and
   a dependent variable *y* in terms of this linear function:

      *y = slope \* x + intercept + noise*

   where ``slope`` and ``intercept`` are the regression parameters that are
   estimated, and ``noise`` represents the
   variability of the data that was not explained by the linear regression
   (it is equal to the difference between predicted and actual values
   of the dependent variable).

   Both inputs must be of the same length (no less than two), and
   the independent variable *x* cannot be constant;
   otherwise a :exc:`StatisticsError` is raised.

   For example, we can use the `release dates of the Monty
   Python films <https://en.wikipedia.org/wiki/Monty_Python#Films>`_
   to predict the cumulative number of Monty Python films
   that would have been produced by 2019
   assuming that they had kept the pace.

   .. doctest::

      >>> year = [1971, 1975, 1979, 1982, 1983]
      >>> films_total = [1, 2, 3, 4, 5]
      >>> slope, intercept = linear_regression(year, films_total)
      >>> round(slope * 2019 + intercept)
      16

   If *proportional* is true, the independent variable *x* and the
   dependent variable *y* are assumed to be directly proportional.
   The data is fit to a line passing through the origin.
   Since the *intercept* will always be 0.0, the underlying linear
   function simplifies to:

      *y = slope \* x + noise*

   .. versionadded:: 3.10

   .. versionchanged:: 3.11
      Added support for *proportional*.
```

Exceptions

A single exception is defined:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 699)

Unknown directive type "exception".

```
.. exception:: StatisticsError

   Subclass of :exc:`ValueError` for statistics-related exceptions.
```

:class:`NormalDist` objects

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 704); *backlink*

Unknown interpreted text role "class".

:class:`NormalDist` is a tool for creating and manipulating normal distributions of a random variable. It is a class that treats the mean and standard deviation of data measurements as a single entity.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 707); *backlink*

Unknown interpreted text role "class".

Normal distributions arise from the Central Limit Theorem and have a wide range of applications in statistics.

Returns a new *NormalDist* object where *mu* represents the arithmetic mean and *sigma* represents the standard deviation.

If *sigma* is negative, raises :exc:`StatisticsError`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 724); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 726)

Unknown directive type "attribute".

```
.. attribute:: mean

   A read-only property for the `arithmetic mean
   <https://en.wikipedia.org/wiki/Arithmetic_mean>`_ of a normal
   distribution.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 732)

Unknown directive type "attribute".

```
.. attribute:: median

   A read-only property for the `median
   <https://en.wikipedia.org/wiki/Median>`_ of a normal
   distribution.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 738)

Unknown directive type "attribute".

```
.. attribute:: mode

   A read-only property for the `mode
   <https://en.wikipedia.org/wiki/Mode_(statistics)>`_ of a normal
   distribution.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 744)

Unknown directive type "attribute".

```
.. attribute:: stdev

   A read-only property for the `standard deviation
   <https://en.wikipedia.org/wiki/Standard_deviation>`_ of a normal
   distribution.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 750)

Unknown directive type "attribute".

```
.. attribute:: variance

   A read-only property for the `variance
   <https://en.wikipedia.org/wiki/Variance>`_ of a normal
   distribution. Equal to the square of the standard deviation.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 756)

Unknown directive type "classmethod".

```
.. classmethod:: NormalDist.from_samples(data)

   Makes a normal distribution instance with *mu* and *sigma* parameters
   estimated from the *data* using :func:`fmean` and :func:`stdev`.

   The *data* can be any :term:`iterable` and should consist of values
   that can be converted to type :class:`float`.  If *data* does not
   contain at least two elements, raises :exc:`StatisticsError` because it
   takes at least one point to estimate a central value and at least two
   points to estimate dispersion.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 767)

Unknown directive type "method".

```
.. method:: NormalDist.samples(n, *, seed=None)

   Generates *n* random samples for a given mean and standard deviation.
   Returns a :class:`list` of :class:`float` values.

   If *seed* is given, creates a new instance of the underlying random
   number generator.  This is useful for creating reproducible results,
   even in a multi-threading context.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 776)

Unknown directive type "method".

```
.. method:: NormalDist.pdf(x)

   Using a `probability density function (pdf)
   <https://en.wikipedia.org/wiki/Probability_density_function>`_, compute
   the relative likelihood that a random variable *X* will be near the
   given value *x*.  Mathematically, it is the limit of the ratio ``P(x <=
   X < x+dx) / dx`` as *dx* approaches zero.

   The relative likelihood is computed as the probability of a sample
   occurring in a narrow range divided by the width of the range (hence
   the word "density").  Since the likelihood is relative to other points,
   its value can be greater than `1.0`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 789)

Unknown directive type "method".

```
.. method:: NormalDist.cdf(x)

   Using a `cumulative distribution function (cdf)
   <https://en.wikipedia.org/wiki/Cumulative_distribution_function>`_,
   compute the probability that a random variable *X* will be less than or
   equal to *x*.  Mathematically, it is written ``P(X <= x)``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 796)

Unknown directive type "method".

```
.. method:: NormalDist.inv_cdf(p)

   Compute the inverse cumulative distribution function, also known as the
   `quantile function <https://en.wikipedia.org/wiki/Quantile_function>`_
   or the `percent-point
   <https://www.statisticshowto.datasciencecentral.com/inverse-distribution-function/>`_
   function.  Mathematically, it is written ``x : P(X <= x) = p``.

   Finds the value *x* of the random variable *X* such that the
   probability of the variable being less than or equal to that value
   equals the given probability *p*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 808)

Unknown directive type "method".

```
.. method:: NormalDist.overlap(other)

   Measures the agreement between two normal probability distributions.
   Returns a value between 0.0 and 1.0 giving `the overlapping area for
   the two probability density functions
   <https://www.rasch.org/rmt/rmt101r.htm>`_.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 815)

Unknown directive type "method".

```
.. method:: NormalDist.quantiles(n=4)

    Divide the normal distribution into *n* continuous intervals with
    equal probability.  Returns a list of (n - 1) cut points separating
    the intervals.

    Set *n* to 4 for quartiles (the default).  Set *n* to 10 for deciles.
    Set *n* to 100 for percentiles which gives the 99 cuts points that
    separate the normal distribution into 100 equal sized groups.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 825)

Unknown directive type "method".

```
.. method:: NormalDist.zscore(x)

    Compute the
    `Standard Score <https://www.statisticshowto.com/probability-and-statistics/z-score/>`_
    describing *x* in terms of the number of standard deviations
    above or below the mean of the normal distribution:
    ``(x - mean) / stdev``.

    .. versionadded:: 3.9
```

Instances of :class:`NormalDist` support addition, subtraction, multiplication and division by a constant. These operations are used for translation and scaling. For example:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 835); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst, line 839)

Unknown directive type "doctest".

```
.. doctest::
    >>> temperature_february = NormalDist(5, 2.5)             # Celsius
    >>> temperature_february * (9/5) + 32                      # Fahrenheit
    NormalDist(mu=41.0, sigma=4.5)
```

Dividing a constant by an instance of :class:`NormalDist` is not supported because the result wouldn't be normally distributed.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 845); *backlink*

Unknown interpreted text role "class".

Since normal distributions arise from additive effects of independent variables, it is possible to add and subtract two independent normally distributed random variables represented as instances of :class:`NormalDist`. For example:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 848); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 854)

Unknown directive type "doctest".

```
.. doctest::

    >>> birth_weights = NormalDist.from_samples([2.5, 3.1, 2.1, 2.4, 2.7, 3.5])
    >>> drug_effects = NormalDist(0.4, 0.15)
    >>> combined = birth_weights + drug_effects
    >>> round(combined.mean, 1)
    3.1
    >>> round(combined.stdev, 1)
    0.5
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 864)

Unknown directive type "versionadded".

```
.. versionadded:: 3.8
```

:class:`NormalDist` Examples and Recipes

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 867); *backlink*

Unknown interpreted text role "class".

:class:`NormalDist` readily solves classic probability problems.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 870); *backlink*

Unknown interpreted text role "class".

For example, given historical data for SAT exams showing that scores are normally distributed with a mean of 1060 and a standard deviation of 195, determine the percentage of students with test scores between 1100 and 1200, after rounding to the nearest whole number:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 878)

Unknown directive type "doctest".

```
.. doctest::

    >>> sat = NormalDist(1060, 195)
    >>> fraction = sat.cdf(1200 + 0.5) - sat.cdf(1100 - 0.5)
    >>> round(fraction * 100.0, 1)
    18.4
```

Find the quartiles and deciles for the SAT scores:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 888)

Unknown directive type "doctest".

```
.. doctest::

    >>> list(map(round, sat.quantiles()))
    [928, 1060, 1192]
    >>> list(map(round, sat.quantiles(n=10)))
    [810, 896, 958, 1011, 1060, 1109, 1162, 1224, 1310]
```

To estimate the distribution for a model than isn't easy to solve analytically, :class:`NormalDist` can generate input samples for a Monte Carlo simulation:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 895); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 899)

Unknown directive type "doctest".

```
.. doctest::

    >>> def model(x, y, z):
    ...     return (3*x + 7*x*y - 5*y) / (11 * z)
    ...
    >>> n = 100_000
    >>> X = NormalDist(10, 2.5).samples(n, seed=3652260728)
    >>> Y = NormalDist(15, 1.75).samples(n, seed=4582495471)
    >>> Z = NormalDist(50, 1.25).samples(n, seed=6582483453)
    >>> quantiles(map(model, X, Y, Z))       # doctest: +SKIP
    [1.4591308524824727, 1.8035946855390597, 2.175091447274739]
```

Normal distributions can be used to approximate Binomial distributions when the sample size is large and when the probability of a successful trial is near 50%.

For example, an open source conference has 750 attendees and two rooms with a 500 person capacity. There is a talk about Python and another about Ruby. In previous conferences, 65% of the attendees preferred to listen to Python talks. Assuming the population preferences haven't changed, what is the probability that the Python room will stay within its capacity limits?

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 922)

Unknown directive type "doctest".

```
.. doctest::

    >>> n = 750             # Sample size
    >>> p = 0.65            # Preference for Python
    >>> q = 1.0 - p         # Preference for Ruby
    >>> k = 500             # Room capacity

    >>> # Approximation using the cumulative normal distribution
    >>> from math import sqrt
    >>> round(NormalDist(mu=n*p, sigma=sqrt(n*p*q)).cdf(k + 0.5), 4)
    0.8402

    >>> # Solution using the cumulative binomial distribution
    >>> from math import comb, fsum
    >>> round(fsum(comb(n, r) * p**r * q**(n-r) for r in range(k+1)), 4)
    0.8402

    >>> # Approximation using a simulation
    >>> from random import seed, choices
    >>> seed(8675309)
    >>> def trial():
    ...     return choices(('Python', 'Ruby'), (p, q), k=n).count('Python')
    >>> mean(trial() <= k for i in range(10_000))
    0.8398
```

Normal distributions commonly arise in machine learning problems.

Wikipedia has a nice example of a Naive Bayesian Classifier. The challenge is to predict a person's gender from measurements of normally distributed features including height, weight, and foot size.

We're given a training dataset with measurements for eight people. The measurements are assumed to be normally distributed, so we summarize the data with :class:`NormalDist`:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 954); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 958)

Unknown directive type "doctest".

```
.. doctest::

   >>> height_male = NormalDist.from_samples([6, 5.92, 5.58, 5.92])
   >>> height_female = NormalDist.from_samples([5, 5.5, 5.42, 5.75])
   >>> weight_male = NormalDist.from_samples([180, 190, 170, 165])
   >>> weight_female = NormalDist.from_samples([100, 150, 130, 150])
   >>> foot_size_male = NormalDist.from_samples([12, 11, 12, 10])
   >>> foot_size_female = NormalDist.from_samples([6, 8, 7, 9])
```

Next, we encounter a new person whose feature measurements are known but whose gender is unknown:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 970)

Unknown directive type "doctest".

```
.. doctest::

   >>> ht = 6.0        # height
   >>> wt = 130        # weight
   >>> fs = 8          # foot size
```

Starting with a 50% prior probability of being male or female, we compute the posterior as the prior times the product of likelihoods for the feature measurements given the gender:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 981)

Unknown directive type "doctest".

```
.. doctest::

  >>> prior_male = 0.5
  >>> prior_female = 0.5
  >>> posterior_male = (prior_male * height_male.pdf(ht) *
  ...                   weight_male.pdf(wt) * foot_size_male.pdf(fs))

  >>> posterior_female = (prior_female * height_female.pdf(ht) *
  ...                     weight_female.pdf(wt) * foot_size_female.pdf(fs))
```

The final prediction goes to the largest posterior. This is known as the maximum a posteriori or MAP:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]statistics.rst`, line 995)

Unknown directive type "doctest".

```
.. doctest::

  >>> 'male' if posterior_male > posterior_female else 'female'
  'female'
```