
agate-stats Documentation

Release 0.4.1 (alpha)

Christopher Groskopf

December 19, 2016

1	Install	3
2	Usage	5
3	API	7
3.1	Authors	8
3.2	Changelog	8
3.3	License	9
3.4	Indices and tables	9

agate-stats adds statistical methods to agate.

Important links:

- agate <http://agate.rtfid.org>
- Documentation: <http://agate-stats.rtfid.org>
- Repository: <https://github.com/wireservice/agate-stats>
- Issues: <https://github.com/wireservice/agate-stats/issues>

Install

To install:

```
pip install agate-stats
```

For details on development or supported platforms see the [agate documentation](#).

Usage

agate-stats uses a monkey patching pattern to add additional statistical methods to all `agate.Table` instances.

```
import agate
import agatestats
```

Importing `agate-stats` adds methods to `agate.Table`. For example, to filter a table to only those rows whose `cost` value is an outlier by more than 3 standard deviations you would use `TableStats.stdev_outliers()`:

```
outliers = table.stdev_outliers('price')
```

In addition to `Table` methods `agatestats` also includes a variety of additional aggregations and computations. See the `API` section of the docs for a complete list of all the added features.

`agatestats.table.stdev_outliers` (*self*, *column_name*, *deviations=3*, *reject=False*)

A wrapper around `Table.where` that filters the dataset to rows where the value of the column are more than some number of standard deviations from the mean.

This method makes no attempt to validate that the distribution of your data is normal.

There are well-known cases in which this algorithm will fail to identify outliers. For a more robust measure see `TableStats.mad_outliers()`.

Parameters

- **column_name** – The name of the column to compute outliers on.
- **deviations** – The number of deviations from the mean a data point must be to qualify as an outlier.
- **reject** – If `True` then the new `Table` will contain everything *except* the outliers.

Returns A new `Table`.

`agatestats.table.mad_outliers` (*self*, *column_name*, *deviations=3*, *reject=False*)

A wrapper around `Table.where` that filters the dataset to rows where the value of the column are more than some number of [median absolute deviations](#) from the median.

This method makes no attempt to validate that the distribution of your data is normal.

Parameters

- **column_name** – The name of the column to compute outliers on.
- **deviations** – The number of deviations from the median a data point must be to qualify as an outlier.
- **reject** – If `True` then the new `Table` will contain everything *except* the outliers.

Returns A new `Table`.

`agatestats.tableset.stdev_outliers` (*self*, *column_name*, *deviations=3*, *reject=False*)

A wrapper around `Table.where` that filters the dataset to rows where the value of the column are more than some number of standard deviations from the mean.

This method makes no attempt to validate that the distribution of your data is normal.

There are well-known cases in which this algorithm will fail to identify outliers. For a more robust measure see `TableStats.mad_outliers()`.

Parameters

- **column_name** – The name of the column to compute outliers on.

- **deviations** – The number of deviations from the mean a data point must be to qualify as an outlier.
- **reject** – If `True` then the new `Table` will contain everything *except* the outliers.

Returns A new `Table`.

`agatestats.tableset.mad_outliers` (*self*, *column_name*, *deviations=3*, *reject=False*)

A wrapper around `Table.where` that filters the dataset to rows where the value of the column are more than some number of [median absolute deviations](#) from the median.

This method makes no attempt to validate that the distribution of your data is normal.

Parameters

- **column_name** – The name of the column to compute outliers on.
- **deviations** – The number of deviations from the median a data point must be to qualify as an outlier.
- **reject** – If `True` then the new `Table` will contain everything *except* the outliers.

Returns A new `Table`.

class `agatestats.aggregations.PearsonCorrelation` (*x_column_name*, *y_column_name*)

Calculates the [Pearson correlation coefficient](#) for *x_column_name* and *y_column_name*.

Returns a number between -1 and 1 with 0 implying no correlation. A correlation close to 1 implies a high positive correlation i.e. as *x* increases so does *y*. A correlation close to -1 implies a high negative correlation i.e. as *x* increases, *y* decreases.

Note: this implementation is borrowed from the MIT licensed [latimes-calculate](#). Thanks, LAT!

Parameters

- **x_column_name** – The name of a column.
- **y_column_name** – The name of a column.

run (*table*)

Returns `decimal.Decimal`.

class `agatestats.computations.ZScores` (*column_name*)

Computes the z-scores (standard scores) of a given column.

3.1 Authors

The following individuals have contributed code to agate-stats:

- Christopher Groskopf

3.2 Changelog

3.2.1 0.4.1

3.2.2 0.4.0 - December 19, 2016

- Update `ZScores` to use new `Computation` interface.

- Remove monkey patching.
- Upgrade agate dependency to 1.5.0.

3.2.3 0.3.1 - November 5, 2015

- Fix packaging issue.

3.2.4 0.3.0 - November 5, 2015

- Added usage documentation.
- Convert *PearsonCorrelation* to an aggregation.
- Update required version of agate to 1.1.0.
- Removed Python 2.6 support.

3.2.5 0.2.0 - October 22, 2015

- Update to support agate 1.0.0.

3.2.6 0.1.0 - October 6, 2015

- Initial version.

3.3 License

The MIT License

Copyright (c) 2015 Christopher Groskopf and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

3.4 Indices and tables

- genindex
- modindex

- search

M

mad_outliers() (in module agatestats.table), 7

mad_outliers() (in module agatestats.tableset), 8

P

PearsonCorrelation (class in agatestats.aggregations), 8

R

run() (agatestats.aggregations.PearsonCorrelation
method), 8

S

stdev_outliers() (in module agatestats.table), 7

stdev_outliers() (in module agatestats.tableset), 7

Z

ZScores (class in agatestats.computations), 8