uncertain_panda Documentation

Release 0.2.0

Nils Braun

Nov 12, 2018

Contents

`uncertain_panda` helps you with constructing uncertainties of quantities calculated on your `pandas` data frames, by applying the method of bootstrapping.

CHAPTER 1

Content

1.1 Examples

To be filled.

1.2 Bootstrapping

Suppose you want to calculate a quantity $f(X)$ on your data frame X. Bootstrapping samples multiple versions Y_i of X by drawing elements with replacement from the data frame with the same length the data frame itself. On all these Y_i, the function f is evaluated, creating a distribution of possible values for $f(X)$. The standard deviation of this distribution is the (symmetric) uncertainty returned by `uncertain_panda`. If you request the asymmetric uncertainty, the 1 sigma quantile in both directions around the median is returned. You can find some more information on bootstrapping in the net, e.g. on wikipedia.

1.3 API

To be filled.

CHAPTER 2

Why is the panda uncertain?

Have you ever calculated quantities on your pandas data frame/series and wanted to know their uncertainty? Did you ever wondered if the difference in the average of two methods is significant?

Then you want to have an uncertain panda!

`uncertain_panda` helps you calculate uncertainties on arbitrary quantities related to your pandas data frame e.g. `mean`, `median`, `quantile` or `min`/`max` and every other arbitrary function on pandas data frames!

You can use any measured data (e.g. from A/B testing, recorded data from an experiment or any type of tabular data) and calculate any quantity using `pandas` and `uncertain_panda` will give you the uncertainty on this quantity.

CHAPTER 3

How to use it?

First, install the package

```
pip install uncertain_panda
```

Now, just import pandas from the `uncertain_panda` package and prefix `unc` before every calculation to get the value with the uncertainty:

```
from uncertain_panda import pandas as pd

series = pd.Series([1, 2, 3, 4, 5, 6, 7])
series.unc.mean()
```

That's it! The return value is an instance of the uncertainty `Variable` from the superb uncertainties package. As this package already knows how to calculate with uncertainties, you can use the results as if they were normal numbers in your calculations.

```
series.unc.mean() + 2* series.unc.std()
```

Super easy!

You can find some more examples in *Examples*.

3.1 Comparison in A/B testing

Suppose you have done some A/B testing with a brand new feature you want to introduce. You have measured the quality of your service before (*A*) and after (*B*) the feature introduction. The averge quality is better, but is the change significant?

A first measure for this problem might be the uncertainty of the average, so lets calculate it:

```
data_frame.groupby("feature_introduced").quality.unc.mean()
```

which will not only give you the two average qualities but also their uncertainties.

CHAPTER 4

Features

The development has just started and there is a lot that can still be added. Here is a list of already implemented features

- Automatic calculation of uncertainties of every built in pandas function for
  - data frames
  - series
  - grouped data frames

  using the prefix `unc` before the function name, e.g.

```
df.unc.mean()
```

  In the background, it used the method of bootstrapping (see below) to calculate the uncertainties.
- Possibility to calculate asymmetric or symmetric uncertainties, with `unc` or `unc_asym`.
- Opional usage of `dask` for large data samples. Enable it with

```
df.unc.mean(pandas=False)
```

  to use `dask` instead of pandas.
- Plotting functionality for uncertainties with

```
df.unc.mean().plot_with_uncertainties(kind="bar")
```

  for a nice error-bar plot.
- Full configurable bootstrapping with either using pandas built-in methods or `dask` (optionally enabled). Just pass the options to your called method, e.g.

```
df.unc.mean(number_of_draws=300)
```

  to use 300 draws in the bootstrapping.

CHAPTER 5

How does it work?

Under the hood, `uncertain_panda` is using bootstrapping for calculating the uncertainties. Find more information on bootstrapping in *Bootstrapping*.

CHAPTER 6

Other packages

There are probably plenty of packages out there for this job - but the only known one I am aware of is the bootstrapped package. Compared to this package, `uncertain_panda` tries to automate the quantity calculation and works for arbitrary functions as well can use `dask` for the calculation. `bootstrapped` is very nice for sparse arrays, which is not (yet) implemented in `uncertain_panda`.