
Truth Documentation

Release 0.1.1

Ben Schroeter

Mar 06, 2019

Contents:

1	Installing Truth	3
2	API Usage	5
2.1	Metric	5
2.2	Result	6
3	Command-line usage (EXPERIMENTAL!!!)	7
4	truth package	9
4.1	Submodules	9
4.2	Module contents	9
5	Indices and tables	11

Truth is a Python library and command-line utility for the verification of Earth system data.

CHAPTER 1

Installing Truth

Install the current development version of truth via pip:

```
$ pip install https://github.com/bschroeter/truth/archive/dev.zip
```

As releases become available I'll make this available on PyPi.

Truth has two main concepts:

- Metrics
- Results

2.1 Metric

Every calculation performed by truth is done so by an object called a *Metric*. The metric class is constructed as follows:

```
from truth.core.metric import Metric

diff = Metric(
    name="Difference",
    key="diff",
    func=lambda model, reference, *args, **kwargs: model - reference
)
```

The *func* argument expects a function which (at minimum) takes the arguments above (including **args* and ***kwargs*) and returns the calculation. We've used a lambda here for brevity, but it is just as easy to supply a fully-formed function as the argument, which may be preferable for more complex calculations.

```
def _diff(model, reference, *args, **kwargs):
    return model - reference

diff = Metric(
    name="Difference",
    key="diff",
    func=_diff
)
```

The result of the calculation is obtained by calling the *calculate* method on the object, which produces a *Result* object, the important attribute of which is the *data* containing the answer:

```
import numpy as np
model = np.arange(1,4)
reference = np.arange(4,7)
result = diff.calculate(model, reference)
print(result)
# <truth.core.result.Result object at 0x102fd9978>
print(result.data)
# array([-3, -3, -3])
```

Why do we use a *Result* object? Read on.

2.2 Result

The result object contains the result of a metric calculation. It has distinct advantages over just returning the answer by addition additional metadata to the answer which can be useful in writing out the result to a file. Under the hood, the *Result* object is constructed as follows:

```
from truth.core.result import Result

diff_result = Result(
    metric = diff, # A pointer to the metric that generated it,
    model = model, # A pointer to the model data that went into the calculation
    reference = reference, # A pointer to the reference data that went into the
    ↪ calculation
    data = _result # The actual data resulting from the calculation (typically a
    ↪ numpy array)
    **kwargs # Any additional arguments are added as attributes that may be accessed
    ↪ when writing results to a file
)
```

Command-line usage (EXPERIMENTAL!!!)

Truth can be invoked from the command line:

```
$ truth
usage: truth [-h] [-v] [-d [DIMENSION [DIMENSION ...]]] {verify} ...

positional arguments:
  {verify}              Sub-command help
  verify                Compute verification statistics

optional arguments:
  -h, --help            show this help message and exit
  --version             Display Truth version
  -d [DIMENSION [DIMENSION ...]], --dimension [DIMENSION [DIMENSION ...]]
                        Dimensional subsetting. -d dim,[min],[max],[stride]
```


CHAPTER 4

truth package

4.1 Submodules

4.2 Module contents

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`