

---

# Dummydata Documentation

*Release 0.1*

**Bjoern Broetz, Alexander Loew**

**May 12, 2017**



---

## Contents

---

<b>1</b>	<b>Introduction to Dummydata</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	using github . . . . .	5
2.2	using pip . . . . .	5
2.3	using conda (not working yet) . . . . .	5
<b>3</b>	<b>How it works</b>	<b>7</b>
3.1	Characteristics and options . . . . .	7
3.2	Some further examples . . . . .	8
3.3	Current limitations . . . . .	8
<b>4</b>	<b>Indices and tables</b>	<b>9</b>



Contents:



# CHAPTER 1

---

## Introduction to Dummydata

---

Dummydata is a package that allows to generate geospatial data fields with predefined statistical properties and store these as netCDF files.



Currently the package is available from [github](#) and can be installed in addition via [pip](#) or [conda](#) .

### using github

To install the package from the git sources, just do the following:

```
# to get the development version
cd <SOME TEMPORARY DIRECTORY>
wget https://github.com/pygeo/dummydata/archive/master.zip
unzip master.zip
cd dummydata-master
python setup.py install
```

### using pip

To install the package using pip, just do the following:

```
pip install dummydata
```

### using conda (not working yet)

To install via conda do the following:

```
conda install [-n YOURENV] -c conda-forge dummydata
```



Dummydata allows to generate either two dimensional data fields with a time vector (e.g. sea surface temperature fields) or a 3D variable with an additional vertical coordinate.

Currently regular lat/lon grids are supported for coordinates.

A small example that generates a random dataset with dimensions (time, lat, lon) is provided as follows

```
from dummydata import Model2

# generate a 2D variable
M2 = Model2(start_year=2003, stop_year=2014)
```

This generates a monthly timeseries starting 1st of January 2003 and ending 31.12.2014. A netCDF file will be automatically generated and closed. To generate a field of vertical air temperature profiles a script would look as follows:

```
from dummydata import Model3

# generate a 3D variable
M3 = Model3(var='ta', oname='air_temperature', start_year=1998, stop_year=2002)
```

This will generate a file *air\_temperature.nc* from 1998 to 2002 with a variable named *ta*.

The dummy data which is generated includes common metadata for different variable types. The tool therefore contains already a set of predefined variables with predefined metadata. The current list of supported variables can be found in the file *meta.py*. In case a user wants to add additional variable options, the necessary metadata information has to be included in the dictionary specified in *meta.py*.

## Characteristics and options

The following options are currently available:

**var** [string][optional] specifies the name of the variable to be generated; note that the variable name needs to be part of the defined variables in *meta.py*

**oname** [string][optional] name of netCDF output file to be generated

**start\_year** [int][obligatory] start year for dataset to be generated

**stop\_year** [int][obligatory] stop year for dataset to be generated

**method** [string][obligatory] method to be used for data generation. At the moment the following options are supported:

- 'uniform' generates a white noise field
- 'constant': generates a field with constant values; the *constant* argument needs to be provided in that case as well.

**constant** [float][obligatory when method='constant'] specifies the constant value to be used

**append\_coordinates** [bool] specifies if fields with coordinates should be appended

**append\_cellsize** [bool] specifies if fields with the cellsize information should be appended to the output file

## Some further examples

```
from dummydata import Model2, Model3

# generate a 2D dataset with the value 5. everywhere
M2 = Model2(method='constant', constant=5., oname='myconst5', start_year=1998, stop_
↳ year=2002)
```

## Current limitations

- only monthly sampling frequencies supported at the moment
- no min/max can be specified to specify the range of the values
- specification of metadata is currently rather limited and done in *meta.py* which is not very user friendly. As an alternative user specific configuration files could be used.

## CHAPTER 4

---

### Indices and tables

---

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