
frame2package Documentation

Release 0.0.1

Robin Linderborg

Oct 24, 2019

Contents:

1	Quickstart	3
2	Usage	5
2.1	Data format	5
2.2	Concepts	5
2.3	Disaggregation levels	6
2.4	Entities	6
3	API Documentation	7
	Index	9

Frame2Package is a helper library for converting Pandas dataframes to [DDF packages](#).

```
f2p = Frame2Package()  
f2p.add_data(data=df, concepts=concepts)  
f2p.to_package('my-ddf-folder')
```


CHAPTER 1

Quickstart

Installation

To install frame2package, simply run this command (using pipenv or pip):

```
$ pipenv install frame2package
```

Basic usage

```
import pandas as pd
import io
from frame2package import Frame2Package

# Load some sample data

data = """area,year,age,education,distribution
Bahamas,2000,15+,Total,1.0
Fiji,1970,80+,Total,1.0
Gabon,2025,20--64,Under 15,0.0
Brunei Darussalam,2045,All,Total,1.0
Thailand,1985,15+,Upper Secondary,0.07
"""

df = pd.read_csv(io.StringIO(data))

# Specify all the concepts in the dataset
# as per the DDF data format specification.

concepts = [
    {
        'concept': 'area',
        'concept_type': 'entity_domain'
    },
    {
        'concept': 'year',
        'concept_type': 'time'
    }
]
```

(continues on next page)

(continued from previous page)

```
    },
    {
        'concept': 'age',
        'concept_type': 'string'
    },
    {
        'concept': 'education',
        'concept_type': 'entity_domain'
    },
    {
        'concept': 'distribution',
        'concept_type': 'measure'
    },
]

# Initialize a Frame2Package object
f2p = Frame2Package()

# Add the data
f2p.add_data(data=df, concepts=concepts)

# Save the package
f2p.to_package('sample-dataset')
```


2.1 Data format

In order for *frame2package* to create a datapackage, your data needs to adhere to the [tidy format](#):

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

2.2 Concepts

In addition to the data itself, you also need to specify what types of variables the dataset consists of. The variables are referred to as *concepts*. Each concept has a name and a type.

```
concepts = [  
  {  
    'concept': 'country',  
    'concept_type': 'entity_domain'  
  },  
  {  
    'concept': 'gdp',  
    'concept_type': 'measure'  
  }  
]
```

Feel free to add additional information about your concepts:

```
{  
  'concept': 'country',  
  'concept_type': 'entity_domain',  
  ...  
}
```

(continues on next page)

(continued from previous page)

```
'description': 'ISO 3 code of countries'
}
```

2.3 Disaggregation levels

Some of your variables may have “total” entries. For instance, a *sex* column may have the unique values *male*, *female*, and *both*. In such cases, the DDF package will create separate datasets with and without the sex dimension. To enable this feature, specify what variables have “total” entries and what they are called in the data.

```
f2p = Frame2Package()
f2p.add_data(df, concepts, totals={'sex': 'both'})
```

2.4 Entities

For each concept of type *entity_domain*, frame2package will generate an entity file *ddf-entities-<entity-name>.csv* listing all the unique values for that entity.

If you want to add additional information about the values, such as a description column, create a dataframe with all the unique values for that entity and any additional columns and use *.update_entity* to add the information to the package.

```
# Assuming entity with name "sex" and values "M", "F", "B"

# First copy the entity data
sex = f2p.entities['sex'].copy()

# Add a description column
sex['description'] = ['Male', 'Female', 'Both']

# Update the entity in the package
f2p.update_entity(name='sex', data=sex)
```

Documentation of Frame2Package's public API.

class `frame2package.Frame2Package`
Main class of frame2package.

to_package (*dirname*, *attrs*={})

Save data to a DDF package.

dirname [str] Name of the DDF directory to be created.

attrs [dict] Attributes to add/update in datapackage.json.

F

Frame2Package (*class in frame2package*), [7](#)

T

to_package() (*frame2package.Frame2Package*
method), [7](#)