drivendata/cookiecutter-data-science

github.com/drivendata/cookiecutter-data-science

drivendata

Cookiecutter Data Science

A logical, reasonably standardized, but flexible project structure for doing and sharing data science work.

Project homepage

Requirements to use the cookiecutter template:

- Python 2.7 or 3.5+
- Cookiecutter Python package >= 1.4.0: This can be installed with pip by or conda depending on how you manage your Python packages:

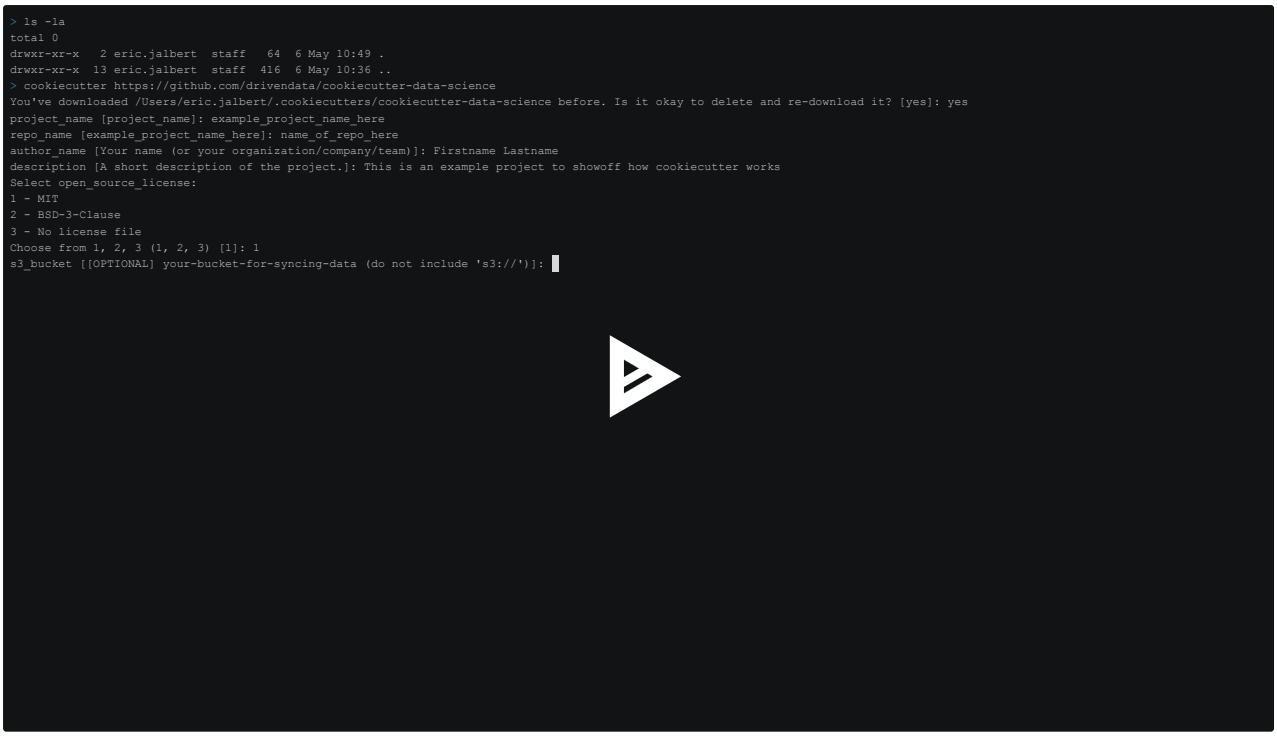
```
$ pip install cookiecutter
```

or

```
$ conda config --add channels conda-forge
$ conda install cookiecutter
```

To start a new project, run:

```
cookiecutter -c v1 https://github.com/drivendata/cookiecutter-data-science
```



New version of Cookiecutter Data Science

Cookiecutter data science is moving to v2 soon, which will entail using the command `ccds ...` rather than `cookiecutter ...`. The cookiecutter command will continue to work, and this version of the template will still be available. To use the legacy template, you will need to explicitly use `-c v1` to select it. Please update any scripts/automation you have to append the `-c v1` option (as above), which is available now.

The resulting directory structure

The directory structure of your new project looks like this:

```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Sphinx project; see sphinx-doc.org for details
│
├── models             <- Trained and serialized models, model predictions, or
model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for
ordering),
│                         the creator's initials, and a short `-` delimited
description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory
materials.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis
environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── setup.py           <- makes project pip installable (pip install -e .) so src
can be imported
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   ├── data           <- Scripts to download or generate data
│   │   └── make_dataset.py
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │   └── build_features.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to
make
│   │   │                 predictions
│   │   ├── predict_model.py
│   │   └── train_model.py
│   │
│   └── visualization  <- Scripts to create exploratory and results oriented
visualizations
│       └── visualize.py
│
└── tox.ini            <- tox file with settings for running tox; see
tox.readthedocs.io
```

Contributing

We welcome contributions! See the docs for guidelines.

Installing development requirements

```
pip install -r requirements.txt
```

Running the tests

```
py.test tests
```