Tools & Skills
Distribution

Dr. Yves J. Hilpisch

Overview of Tools & Skills

BASICS	DEVELOPMENT	DISTRIBUTION
Python Installation & Environments on Mac & Linux on Windows	**Code Editing with Vim**	**Python Packaging**
Docker Usage Jupyter Notebook on Mac & Linux on Windows	**Screen + Vim + q (editing, logging, debugging)**	**Documentation**
Cloud Usage on Mac & Linux on Windows	**Doctest & Unittest**	**Code Hosting (eg Github)**
Basic Linux Tools & Shell Basics (ongoing)	**Git Version Control**	**Case Study**

Tools & Skills 08

Packaging

+ Basic Linux Tools

Link to Github repository

http://github.com/yhilpisch/packaging

These are the basic steps to create a Python package:

1. transform your code into a Python package structure
2. create a `README.md` file (describing the project, providing a tutorial)
3. create a `setup.py` file (this contains relevant meta data)
4. [initialize a Git repository for version control, adjust `.gitignore`]
5. create an account under https://test.pypi.org/account/register/
6. build the files for distribution locally
7. upload the files to the PyPI (test) server
8. test the installation via

```
pip install --index-url https://test.pypi.org/simple/ $MY_PACKAGE
```

Tools & Skills 09

Documentation

+ Basic Linux Tools

Link to Github repository

http://github.com/yhilpisch/documentation

THE PYTHON QUANTS

Navigation

Quick search

DX Analytics

DX Analytics is a **Python-based financial analytics library** which allows the modeling of rather complex derivatives instruments and portfolios. Make sure to fully understand what you are using this Python package for and how to apply it. Please also read the license text and disclaimer.

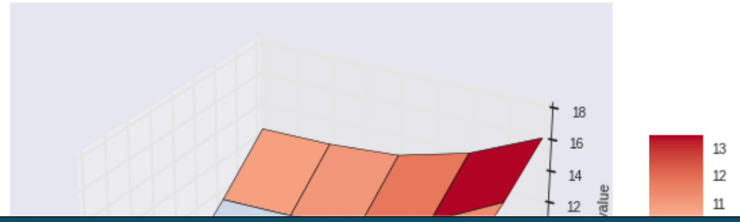
You find the **Github repository** under http://github.com/yhilpisch/dx.

Introduction

Basic Philosophy

DX Analytics is a Python-based financial analytics library that mainly implements what is sometimes called the **global valuation of (complex portfolios of) derivatives instruments** (see http://bit.ly/1SToEwb). The major characteristic of this approach is the **non-redundant modeling** of all components needed for the valuation (e.g. risk factors) and the **consistent simulation and valuation** of all relevant portfolio components (e.g. correlated risk factors, multi-risk derivatives and portfolios themselves).

With DX Analytics you can, for instance, model and risk manage multi-risk derivatives instruments (e.g. American maximum call option) and generate 3-dimensional **present value surfaces** like this one:



These are the basic steps to create a package documentation:

1. document your Python code (docstrings, examples, etc.)
2. create/update your `README.md` file
3. create/update your Git repository
4. install `Sphinx` and `nbsphinx`
5. create a Sphinx project via `sphinx-quickstart`
6. write your documentation in the form of Jupyter Notebooks
7. maybe recycle (parts of) the `README.md` file
8. put static content (e.g. images) in the `_static` folder (adjust paths/references)
9. include the Jupyter Notebook files and maybe other `.rst` files in the base file (e.g. `index.rst` or `project.rst`)
10. build the HTML documentation in the project folder via `make html`
11. check messages/errors and have a look at the build files in `_build`
12. share it with the world on a Web page … and receive feedback

Tools & Skills 10

Hosting & Case Study

Link to Github repository

http://github.com/yhilpisch/mvportfolio

The case study covers the following:

1. developing a Python class for Mean-Variance Portfolio (MVP) theory
2. including logging capabilities based on the `logging` package
3. writing `docstring`-based documentation of the class and methods
4. including `docstring`-based testing with `doctest`
5. writing a Jupyter Notebook-based tutorial
6. creating a `README.md` file from the Jupyter Notebook
7. documenting with `Sphinx` via `autodoc` & Jupyter Notebook
8. creating a `setup.py` script for installation, packaging and hosting on PyPI
9. initializing a `Git` repository, adding a `.gitignore` file
10. hosting the Git repository on Github
11. building and hosting the package on PyPI
12. hosting the HTML documentation on a Web server (using `nginx`)

The Python Quants GmbH

Dr. Yves J. Hilpisch
+49 3212 112 9194
http://tpq.io | team@tpq.io
@dyjh