COMPUTO

ISSN 2824-7795

Template for contribution to Computo

Example dedicated to `Python` users

Jane Doe Statistics, Name of Affiliation one
John Doe Computer Science, Name of Afficiliation two

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur posuere vestibulum facilisis. Aenean pretium orci augue, quis lobortis libero accumsan eu. Nam mollis lorem sit amet pellentesque ullamcorper. Curabitur lobortis libero eget malesuada vestibulum. Nam nec nibh massa. Pellentesque porttitor cursus tellus. Mauris urna erat, rhoncus sed faucibus sit amet, venenatis eu ipsum.

Keywords: key1, key2, key3

Contents

1 Introduction **2**
1.1 About this document . . . 2
1.2 Setup a GitHub repository for preparing your submission . . . 2
1.3 Quarto . . . 2

2 Formatting **2**
2.1 Basic markdown formatting . . . 3
2.2 Mathematics . . . 3
2.2.1 Mathematical formulae . . . 3
2.2.2 Theorems and other amsthem-like environments . . . 3
2.3 Python Code . . . 4
2.4 Figures . . . 4
2.5 Tables . . . 5
2.6 Handling references . . . 6
2.6.1 Bibliographic references . . . 6
2.6.2 Other cross-references . . . 6
2.7 Advanced formatting . . . 6

3 Finalize your submission **6**
3.1 Handle `Python` dependencies with `venv` . . . 6
3.1.1 What about `conda`? . . . 7
3.2 Continuous integration . . . 7
3.2.1 What about CI and `conda`? . . . 8

References **8**

1 Introduction

1.1 About this document

This document, accompanied by the customized GitHub repository, provides a template for writing contributions to **Computo** (Computo Team 2020). We show how `Python` code can be included and how the repository can be set up for triggering GitHub actions for rendering the document, with dependencies handled by `venv` and `pip`.

1.2 Setup a GitHub repository for preparing your submission

You can start by clicking the "use this template" button, on the top of the page of the github repository associated with this document. Of course, you can set your repository private during the preparation of your manuscript.

1.3 Quarto

You need quarto installed on your system and the Computo extension to prepare your document. For the latter, once quarto is installed, run the following to install the extension in the current directory (it creates an `_extension` directory which is ignored by git thanks to `.gitignore` by default):

```
quarto add computorg/computo-quarto-extension
```

2 Formatting

This section covers basic formatting guidelines for Quarto, a versatile formatting system for authoring documents integrating markdown, LaTeX and code blocks interpreted either via Jupyter or Knitr (thus supporting Python, R or Julia). It relies on the Pandoc document converter.

To render a document, run `quarto render`. By default, both PDF and HTML documents are generated:

```
quarto render template-computo-python.qmd # renders both HTML and PDF
```

> **Note**
>
> To check the syntax of the formatting below, you can use the `</> source` button at the top right of this document.

2.1 Basic markdown formatting

Bold text or *italic*

- This is a list
- With more elements
- It isn't numbered.

But we can also do a numbered list

1. This is my first item
2. This is my second item
3. This is my third item

2.2 Mathematics

2.2.1 Mathematical formulae

LaTeX code is natively supported[1], which makes it possible to use mathematical formulae:

$$f(x_1, \dots, x_n; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right)$$

It is also posible to cross-reference an equation, see Equation 1:

$$\begin{aligned} D_{x_N} &= \frac{1}{2}\begin{bmatrix} x_L^\top & x_N^\top \end{bmatrix}\begin{bmatrix} L_L & B \\ B^\top & L_N \end{bmatrix}\begin{bmatrix} x_L \\ x_N \end{bmatrix} \\ &= \frac{1}{2}(x_L^\top L_L x_L + 2x_N^\top B^\top x_L + x_N^\top L_N x_N), \end{aligned} \tag{1}$$

2.2.2 Theorems and other amsthem-like environments

Quarto includes a nice support for theorems, with predefined prefix labels for theorems, lemmas, proposition, etc. see this page. Here is a simple example:

Theorem 2.1 (Strong law of large numbers)**.** *The sample average converges almost surely to the expected value:*

$$\overline{X}_n \xrightarrow{a.s.} \mu \qquad \text{when } n \to \infty.$$

See Theorem 2.1.

[1]We use lualatex for this purpose.

2.3 Python Code

Quarto uses either Jupyter or knitr to render code chunks. This can be triggered in the yaml header. In this tutorial, we use `Jupyter` (`Python` and `Jupyter` must be installed on your computer).

```
---
title: "My Document"
author "Jane Doe"
jupyter: python3
---
```

`python` code chunks may be embedded as follows:

```
import numpy as np
x = np.random.normal(0, 1, 10)
x
```

```
array([-0.56065699, -0.05991313, -1.42675325, -0.19598422, -0.88738727,
       -1.27015397, -0.51325394,  0.52419507, -0.87075656, -
1.16642483])
```

2.4 Figures

Plots can be generated as follows:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0.1, 2 * np.pi, 41)
y = np.exp(np.sin(x))

plt.stem(x, y)
plt.show()
```

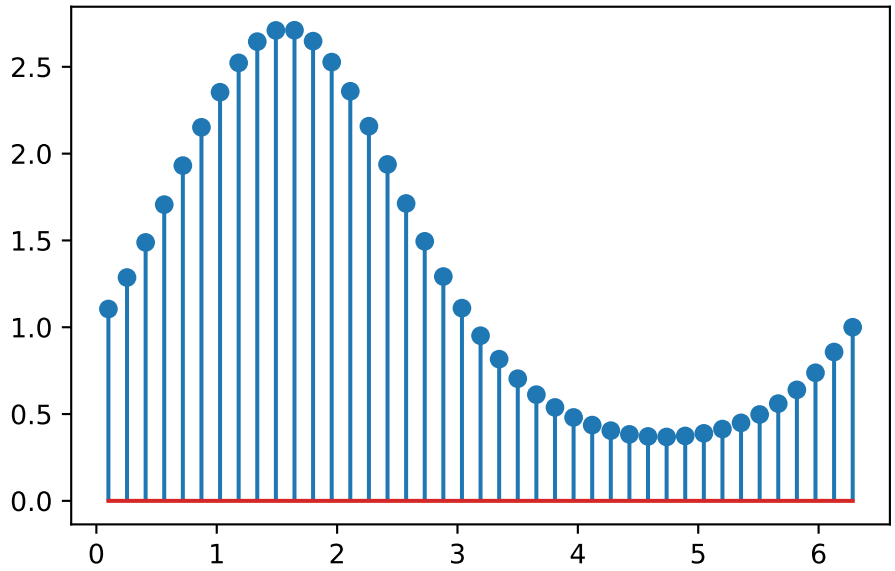

Figure 1: A basic Stem plot

It is also possible to create figures from static images:

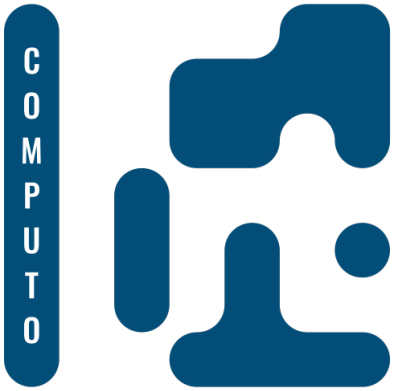

Figure 2: Computo logo (label)

2.5 Tables

Tables (with label: `@tbl-mylabel` renders Table 1) can be generated with markdown as follows

Table 1: my table caption

Tables	Are	Cool
col 1 is	left-aligned	$1600
col 2 is	centered	$12
col 3 is	right-aligned	$1

2.6 Handling references

2.6.1 Bibliographic references

References are displayed as footnotes using BibTeX, e.g. `[@computo]` will be displayed as (Computo Team 2020), where `computo` is the bibtex key for this specific entry. The bibliographic information is automatically retrieved from the `.bib` file specified in the header of this document (here: `references.bib`).

2.6.2 Other cross-references

As already (partially) seen, Quarto includes a mechanism similar to the bibliographic references for sections, equations, theorems, figures, lists, etc. Have a look at this page.

2.7 Advanced formatting

Advanced formatting features are possible and documented (including interactive plots, pseudo-code, (Tikz) diagrams, Lua filters, mixing R + Python in the same document), but are beyond the scope of this simple introduction. We point several entries in this direction.

> ⚠ More information
>
> - The Quarto web site for comprehensive documentation.
> - The template distributed with the Computo Quarto extension, which uses such advanced features.
> - Our mock version of the t-SNE paper, a full and advanced example using Python and the Jupyter kernel.
> - The previously published papers in Computo can be used as references.

3 Finalize your submission

3.1 Handle `Python` dependencies with `venv`

To make your work reproducible, you need to fix the packages and environment used to run your analysis. For `Python`, `venv` is one of the possible reliable method, supported by the community. You basically need a couple of commands to setup your environment on your local machine. First, to create a new virtual environment in the directory `my_env`

```
python3 -m venv my_env
```

and activate it

```
source my_env/bin/activate
```

Then installed the packages required to perform your analysis. Here,

```
python3 -m pip install jupyter matplotlib numpy
```

Once you are all set up, you need to save your working environment into a file so that anyone can reproduce your analysis on their side:

```
python3 -m pip freeze > requirements.txt
```

The corresponding `requirements.txt` file found in this repository is then

Listing 1 `requirements.txt`

```
jupyter
matplotlib
numpy
```

> **Important**
>
> `requirements.txt` is the only file that needs to be versioned by git.

More details for using `venv` and `pip` can be found on the quarto page dedicated to environments.

3.1.1 What about `conda`?

For `conda` users, it is also possible to follow the same path with your favorite version of `conda`. There is a quarto page dedicated to the conda environments.

3.2 Continuous integration

The repository associated with this template is pre-configured to trigger an action on push that performs the following:

1. Check out the repository on an `ubuntu-latest` machine
2. Install quarto and dependencies, including the Computo extension
3. Install Python (3.10) and dependencies with `venv`, using your `requirements.txt` file
4. Render your .qmd file and Publish the results on a gh-page (both HTML and PDF)

The file .github/workflows/build_n_publish.yml is largely inspired from this file.

Once this is successful, you are ready to submit your manuscript to the Computo submission platform.

> ⚠ Warning
>
> The first time, you possibly need to create the branch for the action to work. This can be done by running the following command from your computer, in your git repository:
>
> ```
> quarto publish gh-pages
> ```
>
> Then, set the branch `gh-page` as the source of your github page, and trigger the action to check that everything works fine.

3.2.1 What about CI and `conda`?

The build and deploy process of our Computo quarto extension shows how `miniconda` can be set used in place of `venv`. The main striking difference is the use of a `environment.yml` file in place of `requirements.txt`.

References

Computo Team. 2020. “Computo: Reproducible Computational/Algorithmic Contributions in Statistics and Machine Learning.”