

Template for contribution to Computo

Example dedicated to Julia users

Jane Doe ^{©1} Statistics, Name of Affiliation one John Doe [©] Computer Science, Name of Afficiliation two

Date published: 2025-05-04 Last modified: 2025-05-04

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

2	1	Intr	troduction 2			
3		1.1	About this document	2		
4		1.2	Setup a github repository for preparing your submission	2		
5		1.3	Quarto	2		
6		1.4	Requirements			
7		1.5	Link with your usual tools	2		
8	2	Forn	matting 2			
9		2.1	Basic markdown formatting	3		
10		2.2	Mathematics	3		
11			2.2.1 Mathematical formulae	3		
12			2.2.2 Theorems and other amsthem-like environments	3		
13		2.3	Julia Code	4		
14		2.4	Figures	4		
15		2.5	Tables	5		
16		2.6	Handling references	5		
17			2.6.1 Bibliographic references	5		
18			2.6.2 Other cross-references	5		
19		2.7	Advanced formatting	5		
20	3	Fina	lize your submission	6		
21		3.1	Handle Julia dependencies with Pkg	6		
22		3.2	Continuous integration	6		
23		3.3	Data and large files	7		
24	R.	forer	nces	7		

 $^{^1}$ Corresponding author: janedoe@nowhere.moon

1 Introduction

26 1.1 About this document

- 27 This document, accompanied with the hopefully finely tuned git repos, provides a template for
- writing contributions to Computo (Computo Team 2020). We show how Julia code can be included
- 29 and how the repository can be set up for triggering github actions for rendering the document, with
- 30 dependencies handled by the built-in Pkg manager.

31 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 to this document. Of course, you can set your repository private during the preparation
- of your manuscript.

35 1.3 Quarto

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

39 1.4 Requirements

- 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
- $_{42}$ (it creates a _extension directory which is ignored by git thanks to .gitignore by default):

quarto add computorg/computo-quarto-extension

Julia and Jupyter must be installed on your computer.

44 1.5 Link with your usual tools

- Quarto is expecting a .qmd markdown file, but will also works with a standard Jupyter notebook file
- if you are used to it (it will just require to add the proper YAML metadata²).
- Note: More advanced Jupyter-related functionality like Myst/Jupyter book are not supported in this
- 48 Quarto setup. The markdown syntax inside the Jupyter notebook should follow the Quarto syntax (c.f.
- below). If you are more comfortable with using Myst/Jupyter book, we provide a specific template but it
- 50 will requires more formatting work for Computo editorial team, thus highly encourage authors to use
- 51 the Quarto templates.

2 Formatting

- This section covers basic formatting guidelines for quarto documents.
- To render a document, run quarto render. By default, both PDF and HTML documents are generated:

quarto render template-computo-julia.qmd # will render both to html and PDF

²the same metadata as in the template-computo-julia.qmd file in the first cell, type "Raw", of the notebook

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

Basic markdown formatting Bold text or italic

· This is a list

59

63

- With more elements
- It isn't numbered. 60
- 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

Mathematics 2.2

2.2.1 Mathematical formulae

LaTeX code is natively supported³, which makes it possible to use mathematical formulae:

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

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

$$D_{x_N} = \frac{1}{2} \begin{bmatrix} x_L^\top & x_N^\top \end{bmatrix} \begin{bmatrix} L_L & B \\ B^\top & L_N \end{bmatrix} \begin{bmatrix} x_L \\ x_N \end{bmatrix}$$

$$= \frac{1}{2} (x_L^\top L_L x_L + 2x_N^\top B^\top x_L + x_N^\top L_N x_N),$$
(1)

Theorems and other amsthem-like environments 2.2.2

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:

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

See Theorem 2.1.

³We use lualatex for this purpose.

2.3 Julia 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, (Julia and Jupyter must be installed on your computer)

title: "My Document"
author "Jane Doe"
jupyter: julia-1.10

⁷⁸ julia code chunks may be embedded as follows:

```
10×1 Matrix{ComplexF32}:
     0.61330706f0 - 0.63762915f0im
80
    -0.34964928f0 - 0.6384568f0im
     0.61122406f0 + 1.5640336f0im
      0.3767559f0 - 0.19214594f0im
      0.3552041f0 - 0.36556262f0im
84
     -0.3963343f0 - 0.013641349f0im
85
     0.09055523f0 + 1.3101153f0im
86
    -0.58531713f0 + 0.077849716f0im
87
    -0.17760806f0 + 0.26142728f0im
     0.05099396f0 - 1.0630851f0im
```

90 2.4 Figures

Plots can be generated as follows:



Figure 1: Parametric Plots

- 12 It is also possible to create figures from static images:
- Note: Until Quarto version 1.3+ is released, including a remote image (from a web URL) in a document



Figure 2: Computo logo (label)

(like the image above) will work in the rendered HTML document but will generate an error when building the PDF document (c.f. related bug report).

96 **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

98 2.6 Handling references

99 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:

103 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.

107 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, including:
 - Tutorial
 - User guide
 - Options reference
- 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 Julia dependencies with Pkg

To make your work reproducible, you need to fix the packages and environment used to run your analysis. In Julia, the built-in Pkg package manager is a method of choice. https://towardsdata-science.com/how-to-setup-project-environments-in-julia-ec8ae73afe9c.

In this tutorial, we simply need to add the Plots library and its dependencies. It is important that you also add IJulia to your project, since it is a required dependency for quarto to correctly render your document via Jupyter.

Once added via the Pkg manager of Julia (add Plots; add IJulia), the files Project.toml and Manifest.toml in the current project directory contains all the information required.

3.2 Continuous integration

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

- 1. Check out repository on the ubuntu-latest machine
- 2. Install quarto and dependencies, including the Computo extension
- 3. Install Julia (1.10) and dependencies based on the files Project.toml and Manifest.toml
 - 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.

131

125

126

128

111

112

113

2 3.3 Data and large files

If your submission materials contain files larger than 50MB, especially data files, they won't fit on
 a git repository as is. For this reason, we encourage you to put your data or any materials you deem
 necessary on an external "open data" centered repository hub such a Zenodo or OSF.

136 References

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