Quarto Template Report

Hailey Knolton Ruby Peterman

2024-07-11

Table of contents

1 Quarto Report Template

This Quarto project is a template that you can use for developing your own final report. It suggest a potential structure along with some demonstrations of the advantages of using Quarto for technical documentation (e.g., in-line data processing and plot generation, references, clean formatting, figures, etc.)

You can find the source code for the whole project in the accompanying Github repo.

If you have questions or problems with getting it up and running on your own personal machine, the best way to get help is to open a new issue on the Github repo.

File Structure

The suggested structure for this repository is designed to maintain clarity and keep related files together.

```
.
  _book
  _environment.yml
  _quarto.yml
  _site
  ieee.csl
  index.qmd
  references.bib
  sections
      abstract
          abstract.qmd
      acknowledgements
          acknowledgements.qmd
      conclusion
          conclusion.qmd
      introduction
          images
          introduction.qmd
      references
      results
```

```
        data
            test_data.mat
        images
            test_data_plot.png
        results.qmd
styles.css
```

Tree diagram created using tree.nathanfriend.io.

2 Abstract

As a reminder, the abstract should be a short summary of the main results of the project.

It should:

- Explain to the reader what they will learn from the report
- State the main **quantitative** result of the report

It should not:

- Be longer than 200 words
- Contain references.

3 Introduction

3.1 1.1 In Vivo Imaging Systems

3.2 1.2 Flourescence Imaging

3.3 1.3 Laser Speckle Contrast Imaging

The introduction should contain background information to help a reader understand the context for the work that will be presented in the remaining sections of the report. Typically, an introduction contains a number of references to prior work to help the reader understand the relevant literature that the project is building upon.

3.4 References in Quarto

In Quarto, you can manage references using BibTeX. To do so, you should create a `reference.bib` file in the root directory of your project. This file will contain BibTeX entries for your citations.

For example, this project contains an example `references.bib` file containing a single reference.

```
@article{goodman1976some,
  title={Some fundamental properties of speckle},
  author={Goodman, Joseph W},
  journal={JOSA},
  volume={66},
  number={11},
  pages={1145--1150},
  year={1976},
  publisher={Optica Publishing Group}
}
```

To link the references file we need to include two new entries in the `_quarto.yml` file.

```
bibliography: references.bib
csl: ieee.csl
```

These lines tell Quarto where to find the `.bib` file and what citation style to use (controlled by the citation style with a `.csl` file). In this project we use an IEEE style.

To cite the reference in the main text, type the `@` symbol followed by the `citekey`. In this case, that would be `@goodman1976some`.

Here is an example.

> The probability distribution of a linearly polarized, fully-developed speckle pattern follows a negative exponential distribution [1].

When Quarto renders the document, it will place a citation in place of the `@citekey` pattern and automatically include a linked footnote and a list of references at the bottom.

4 Methods

4.1 2.1 Physical Construction

4.2 2.2 Electrical Components and Software

4.3 2.3 Experiments

4.3.1 2.3.1 Verification

4.3.2 2.3.2 FITC

4.3.3 2.3.3 LSCI

5 Results

5.1 3.1 System Verification and Setup

5.2 3.2 Flourescence Imaging

5.3 3.3 Laser Speckle Contrast Imaging

The advantage to using Quarto as compared to another method of documenting your work is that you can include executable Python code inline with your project. This means that you can provide your data and have your figures re-render without needing to execute separately and then include them into the project.

5.4 Demonstration

5.4.1 Create sample dataset

For example, let's consider a toy example. First, we'll create and save a data set. In your case, you'll likely skip this step since you'll likely have a dataset from elsewhere to use.

First, let's create a Python code block to create some random data.

```
import numpy as np

N = 100 # Set number of data points

x = np.linspace(0, 10, N) # Create x vector
y = 5*x + np.random.rand(N)      # Generate linear data with simulated noise
```

Next, let's save this data in our `/data/` subfolder. You can choose various formats, but one convenient option is to save the data in a `.mat` file. This way it can be opened and processed in Matlab as well as Python. It also allows you to nicely structure the data and will make the loading and plotting code we develop Matlab compatible.

```
import scipy.io as sio  # Load Scipy IO module which has the function to save and load .mat f

data_to_save = {'x': x, 'y': y} # Create dictionary to story the data
sio.savemat('data/test_data.mat', data_to_save)  # Save the data to a .mat file
```

5.4.2 Load and Plot Data

First let's load the data from the `.mat` file.

```
data_load = sio.loadmat('data/test_data.mat')
```

Then, let's plot it.

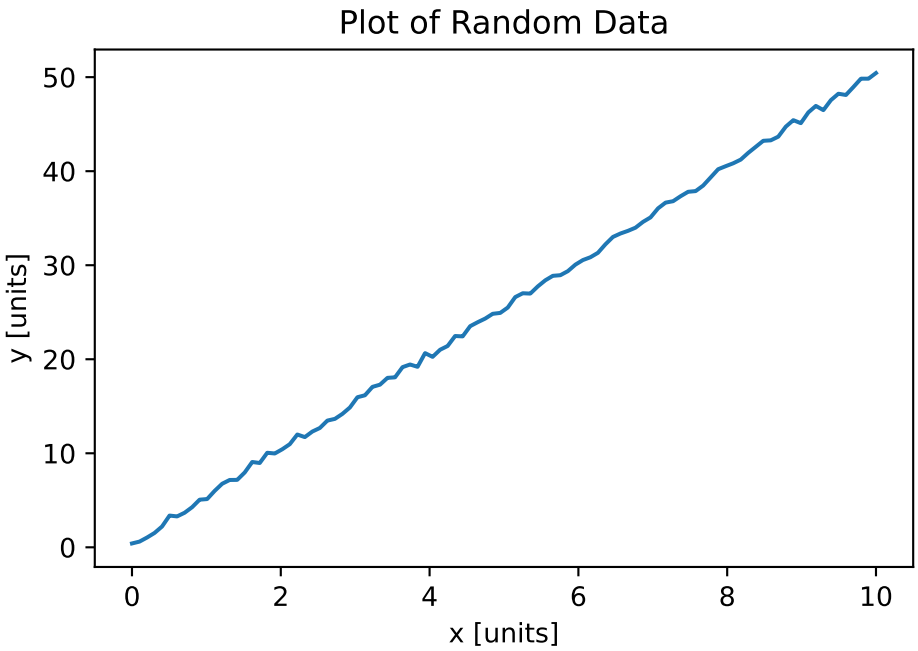
> **Warning**
>
> Note carefully that because of how `savemat()` works, we need to index into the **zeroth element** of the dictionary values in order to get to the data we want.

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

ax.plot(data_load['x'][0],data_load['y'][0]) # Note that we are using the loaded data here an
ax.set_xlabel('x [units]')
ax.set_ylabel('y [units]')
ax.set_title('Plot of Random Data')
```

```
Text(0.5, 1.0, 'Plot of Random Data')
```



5.4.3 Save and Plot Images of Figures

We can even go ahead and save the plots as images if we want to have the flexiblity to include them elsewhere.

```
fig.savefig('images/test_data_plot.png', dpi=300)
```

Then we can directly display as a Quarto figure as usual.

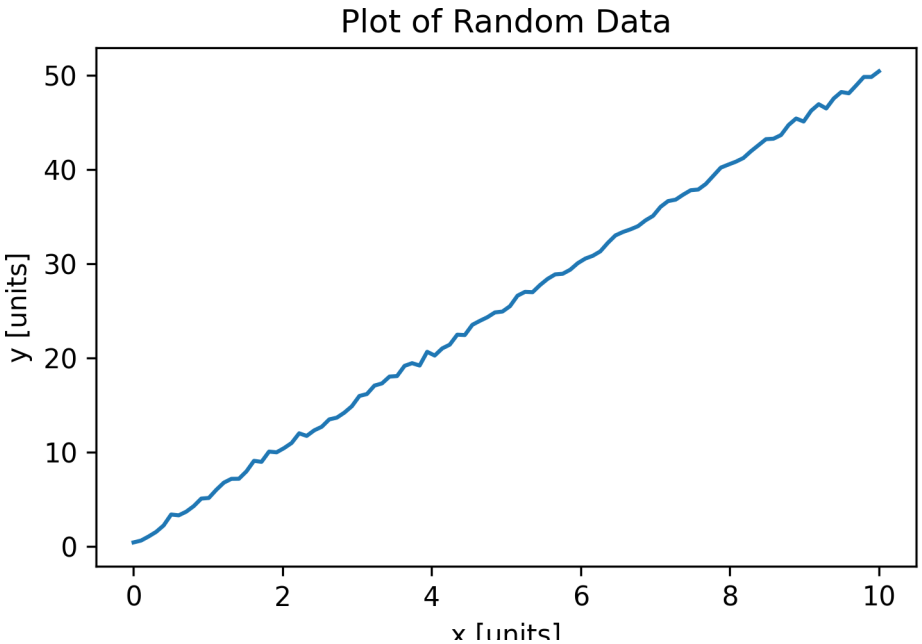


Figure 5.1: Displaying image of plotted random example data.

6 Conclusion

The conclusion summaries the main results of your paper. It generally mirrors the abstract, but in slightly more detail.

It is also common for the conclusion to include a discussion of the data, limitations of the work, and directions for future work.

7 Acknowledgments

The acknowledgements section should always recognize the funding sources which supported the work. This is particularly important for research funded by organizations like the National Science Foundation (NSF) or the National Institutes of Health (NIH). Acknowledging funding properly is very important, so make sure to check in with your PI if you have any questions and to confirm that you have properly recognized the funding sources.

7.1 Funding Acknowledgement

A sample acknowledgement of funding by an NSF grant might read something like the following.

> This research was supported by the National Science Foundation under Grant No. [Your Grant Number]. The authors would like to thank [any other contributors, institutions, or facilities] for their support and collaboration. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation."

[1] J. W. Goodman, "Some fundamental properties of speckle," *JOSA*, vol. 66, no. 11, pp. 1145–1150, 1976.