Tomas Beuzen and Tiffany Timbers

Python Packages

To you, the Reader.

Never stop learning. You are capable of anything.

Contents

List of Tables

List of Figures

Preface

Python packages are the fundamental unit of shareable code in Python. Packages make it easy to reuse and maintain your code, and are how you share your code with your colleagues and the wider Python community. *Python Packages* is an open source textbook that describes modern and efficient workflows for creating Python packages. The focus of this text is overwhelmingly practical; we will demonstrate methods and tools you can use to develop and maintain packages quickly, reproducibly, and with as much automation as possible - so you can focus on writing and sharing code!

Why read this book?

Python packages are a core element of the Python programming language and are how you write reuseable and shareable code in Python. Despite their importance, packages can be difficult to understand and cumbersome to create for beginners and seasoned developers alike. This book aims to describe the packaging process at an accessible level for data scientists, developers and hobbyist programmers. Along the way, we'll walk through the development of a real Python package and we will explore all the key elements of Python packaging, including; package structure, when and why to write tests and documentation, and how to maintain and update your package with the help of automated CI/CD pipelines.

By reading this book, you will: - Understand what Python packages are, and when and why you should use them. - Be able to build your own Python package from scratch. - Learn how to document your Python code and packages, and how to render this documentation into a coherent, shareable document. - Write automated and formal tests for your code. - Learn how to release your package on the Python Package Index (PyPI) and discover best practices for updating and versioning your code. - Implement automation and CI/CD pipelines to build, test, and deploy your package and update its dependencies. - Get tips on Python coding style, best-practice packaging workflows, and other useful development tools.

Structure of the book

Chapter 1: introduction first gives a brief introduction to packages in Python and why you should know how to make them. **Chapter 2: [system-setup]** describes how to set up your development environment to develop packages and follow along with the remainder of the book. In **Chapter 3: [how-to-package-a-python]**, we will develop a small toy package from end-to-end to get a feel for the steps involved in the packaging process and to understand the final product we are aiming for. The remaining chapters then unpack this workflow and go into more details about each step in the packaging process, organised roughly in their order in the workflow:

- **Chapter 4: [package-structure-and-state]**
- **Chapter 5: [testing]**
- **Chapter 6: [documentation]**
- **Chapter 7: [releasing-and-versioning]**
- **Chapter 8: [continuous-integration-and-deployment]**
- **Appendix 1: [packages-with-a-command-line-interface]**
- **Appendix 2: [python-packaging-cheat-sheet]**

Conventions

Throughout this book we use `foo()` to refer to functions, and `bar` to refer to variables and function parameters.

Larger code snippets in the text appear as below and we will use type hints and the so-called "Numpy-style" for docstrings:

```
def palindrome(word: str) -> str:
   """Turns a string into a palindrome by concatenating it with a reversed version of itself.

    Parameters
    ----------
    word : str
        Word to turn into a palindrome.

    Returns
    -------
    str
        Palindrome of word.

    Examples
```

```
    --------
    >>> palindrome("Tomas")
    'TomassamoT'
    >>> palindrome("Python")
    'PythonnohtyP'
    """
    return word + word[::-1]
```

Code executed in the interactive Python interpreter to produce an output will appear as below:

```
a = 1
b = 2
a + b
```

If you have an electronic version of the book, e.g., https://py-pkgs.org, all code is rendered in such a way that you can easily copy and paste directly from your browser to your Python interpreter or editor.

Persistence

The Python software ecosystem is constantly evolving. While the packaging workflows and concepts discussed in this book are effectively tool-agnostic, the tools we do use in the book may have been updated by the time you read it. If the maintainers of these tools are doing the right thing by documenting, versioning, and properly deprecating their code (we'll explore these conepts ourselves in Chapter 7: [releasing-and-versioning]), then it should be straightforward to adapt any outdated code in the book.

Colophon

This book was written in JupyterLab[1], compiled using Jupyter Book[2], is hosted on GitHub[3], and the online version[4] is deployed with Netlify[5]. The

[1] https://jupyterlab.readthedocs.io/en/stable/index.html
[2] https://jupyterbook.org/intro.html
[3] https://github.com/
[4] https://py-pkgs.org/
[5] https://www.netlify.com/

complete source is available from GitHub[6], and is automatically updated after edits by GitHub Actions[7].

Acknowledgements

We'd like to thank everyone that has contributed to the development of *Python Packages*[8]. This is an open source textbook that began as supplementary material for the University of British Columbia's Master of Data Science program and was subsequently developed openly on GitHub where it has been read, revised, and supported by many students, educators, practioners and hobbyists. Without you all, this book wouldn't be nearly as good as it is, and we are deeply grateful. A special thanks to those who directly contributed to the text via GitHub (in alphabetical order): `@Carreau`, `@dcslagel`.

We would also like to acknowledge the software used to develop this book. This book was written in JupyterLab[9], compiled using Jupyter Book[10], is hosted on GitHub[11], and the online version[12] is deployed with Netlify[13].

[6] https://github.com/
[7] https://github.com/features/actions
[8] https://py-pkgs.org/
[9] https://jupyterlab.readthedocs.io/en/stable/index.html
[10] https://jupyterbook.org/intro.html
[11] https://github.com/
[12] https://py-pkgs.org/
[13] https://www.netlify.com/

About the Author

Frida Gomam is a famous lady. Police will always let her go.

1

Introduction

Now unplug your Internet cable, and start doing some serious work.

We have a nice figure in Figure 1.1, and also a table in Table 1.1.

```
par(mar = c(4, 4, 1, .1))
plot(cars, pch = 19)
```

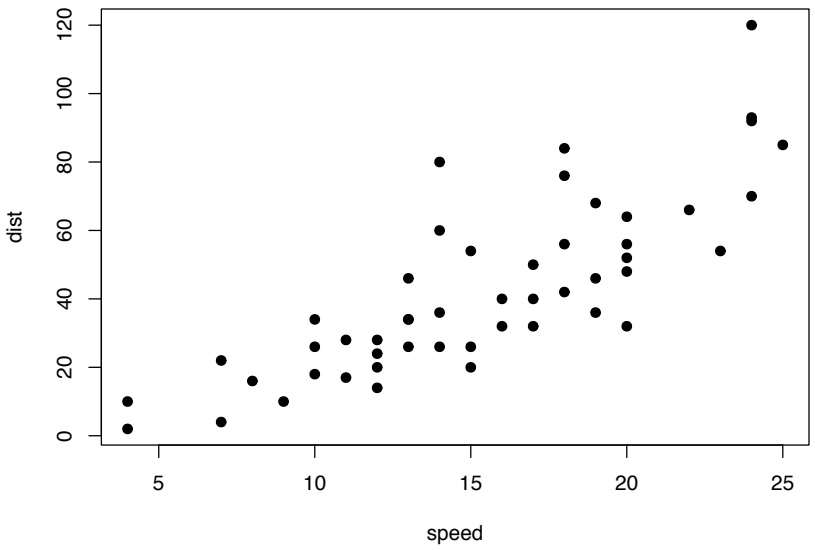

FIGURE 1.1: Hello World!

```
knitr::kable(
  head(iris), caption = 'The boring iris data.',
  booktabs = TRUE
)
```

More chapters to come in `02-foo.Rmd`, `03-bar`.Rmd, ...

TABLE 1.1: The boring iris data.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

2

The FOO Method

We talk about the *FOO* method in this chapter.

A

More to Say

Yeah! I have finished my book, but I have more to say about some topics. Let me explain them in this appendix.

To know more about **bookdown**, see https://bookdown.org.

Preface

Hi there, this is my great book. Here's a reference Structure of the book or this one Structure of the book

Why read this book

It is very important...

Structure of the book

Chapters 1 introduces a new topic, and ...

Software information and conventions

I used the **knitr** package (Xie, 2015) and the **bookdown** package (Xie, 2020) to compile my book. My R session information is shown below:

```
xfun::session_info()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Locale: en_CA.UTF-8 / en_CA.UTF-8 / en_CA.UTF-8 / C / en_CA.UTF-
8 / en_CA.UTF-8
##
## Package version:
##   base64enc_0.1.3 bookdown_0.21   compiler_4.0.2
```

```
##   digest_0.6.25   evaluate_0.14   glue_1.4.2
##   graphics_4.0.2  grDevices_4.0.2 highr_0.8
##   htmltools_0.5.0 jsonlite_1.7.1  knitr_1.30
##   magrittr_1.5    markdown_1.1    methods_4.0.2
##   mime_0.9        rlang_0.4.7     rmarkdown_2.6
##   rstudioapi_0.11 stats_4.0.2     stringi_1.5.3
##   stringr_1.4.0   tinytex_0.26    tools_4.0.2
##   utils_4.0.2     xfun_0.18       yaml_2.2.1
```

Package names are in bold text (e.g., **rmarkdown**), and inline code and filenames are formatted in a typewriter font (e.g., `knitr::knit('foo.Rmd')`). Function names are followed by parentheses (e.g., `bookdown::render_book()`).

Acknowledgments

A lot of people helped me when I was writing the book.

Frida Gomam
on the Mars

Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.21.

Index