Received: 11 October 2024 | Revised: 7 May 2025 | Accepted: 10 May 2025

DOI: 10.1111/test.12409

ORIGINAL ARTICLE

Innovative and interactive statistics teaching using Quarto

Kristian Evans

Computational Foundry, Bay Campus, Swansea University—Mathematics, Swansea, UK

Correspondence
Kristian Evans, Computational Foundry, Bay Campus, Fabian Way Swansea SA1 8EN, Swansea University—Mathematics, Swansea, UK.
Email: k.evans@swansea.ac.uk

Abstract

Quarto is an open-source system that is useful for creating scientific teaching resources, particularly statistical resources, in various formats. Quarto is the next-generation version of R Markdown, and it can produce content using a variety of programming languages, including R, Python, Julia, and Observable JS. The main purpose of this article is to showcase the benefits of using Quarto, to provide a quick-start guide for the system, and to present and evaluate feedback from students using Quarto-produced resources. An overview of Quarto and its capabilities will be provided, followed by examples of resources created by the system. Particular focus will be made on its advantages over more traditional approaches, which include executing common coding languages and interactivity. This article is based on the author's experiences of using Quarto to create resources for teaching Statistics to post-graduate Data Science students and to second-year undergraduate Mathematics students.

KEYWORDS

interactive teaching, observable JS, python, quarto, R, teaching statistics

1 | INTRODUCTION

1.1 | Setting the scene

The author has tried various techniques to overcome students' challenges with learning a new programming language alongside statistical concepts. As stated in Reference 1, *In R-based courses, much of students' time is spent getting up to speed with R: setting up a development environment, learning the syntax, understanding basic computing concepts (file paths are a perpetual problem), and inevitably dealing with bugs in their code.*

The different techniques the author has used include PDF or Word-style, physical teaching resources, where students follow step-by-step guides, manually typing the relevant code into their editors in order to follow worked examples. In addition, the technique of providing script files where students can access and run code for worked examples has also been trialed. With the former technique, students benefit from practicing typing code into their editors; however, the drawback is that this can be time-consuming and result in not enough of the lab class material being covered by the students within the supervised scheduled slot.

On the other hand, and arguably to the other extreme, simply providing code to students for worked examples can result in students quickly working through the coding parts of examples since they only need to run the code on their own device. This approach does not encourage students to understand or learn the code that they are using. From the author's experience, this technique appears to result in students working quickly through the lab sheets, but the downside is that this is primarily surface-level learning because it is straightforward and too tempting for students to run large blocks of code quickly in order to obtain the required outputs. In some instances, it also results in confusion about which outputs relate to which line of code. Large blocks of code provided in script files can be broken up by including comments, but the author's experience is that students tend to be drawn to running the code immediately, as evidenced by students experiencing problems when trying new exercises that follow worked examples.

1.2 | Why Quarto?

These lab teaching approaches with their respective advantages and disadvantages led the author to seek an alternative approach that finds a compromise between the detailed, well-explained approach of requiring students to work through physical teaching resources and manually typing code and the accessibility of merely providing large blocks of code that only require compiling. After some investigation, the author decided to try using Quarto to provide such teaching resources. Quarto is a system that allows users to create a variety of different outputs from a single source file. In particular, it can compile a variety of programming languages, for example, Python, R, Observable JS, and Julia, which makes it particularly useful for producing computer lab teaching material.

As stated in Reference,[2] *Quarto is an open-source scientific and technical publishing system built on Pandoc. You can weave together narrative text and code to produce elegantly formatted output as documents, web pages, blog posts, books, and more*. In particular, the intertwining of text, LaTeX, programming code, and associated outputs allows for the production of resources which can combine theory and practice—this may help with some students' perception of statistics being a group of isolated skills.[3] Pandoc Markdown is an extended and revised version of John Gruber's Markdown syntax, which has excellent support for LaTeX equations and citations. Reference [4] states that *Markdown is a lightweight markup language that you can use to add formatting elements to plaintext text documents*. The HTML web page outputs produced by Quarto can also enhance accessibility, in particular, the outputs follow many of the guidelines set out in the *Increasing Accessibility* section of Reference,[5] for example, *Open-Access Materials* and *Screen Reader Accessibility*. Many of the guidelines in this paper are set out for R Markdown, but they naturally extend to Quarto-produced documents.

For any reader who is familiar with R Markdown, Quarto can be thought of as the next-generation version of this system, although it is important to mention that the developers state that R Markdown will continue to be actively supported. R Markdown has been used for many years, and there are many interesting articles on its use, including.[6,7] While R Markdown is designed for R, and hence its appeal is primarily from users with an R background, Quarto is designed to have a wider appeal to users from a variety of coding backgrounds. In particular, Quarto was developed to be multilingual, and it is planned to work with new languages that do not yet exist.[2] In addition to being multi-language, Quarto is also multi-engine, currently supporting knitr, Jupyter, and Observable. These are the engines that run code and produce outputs. Knitr is an engine for dynamic report generation in R. The Jupyter engine can render Python, Julia, and R, and it is common in data science. Observable JS is JavaScript for use with the Observable engine; however, there are differences between vanilla JavaScript and Observable JS. For readers new to JavaScript,[8] provides a good introduction to its use in Observable. Furthermore, information about Observable can be found in Reference 9. Quarto also potentially supports other engines in the future, demonstrating the developers' aim for Quarto to appeal to a wider range of users. It should also be noted that Quarto is highly compatible with existing content; in particular, most R Markdown documents and Jupyter notebooks can be rendered unmodified with Quarto.[2]

In addition to being multilingual, Quarto has further benefits over R Markdown. Both systems have the option of including code chunks, that is, blocks of code embedded into a document encapsulated by three single inverted commas (see Figure 4 for an example). However, as stated in Reference,[2] *Quarto chunk options are typically included in special comments at the top of code chunks rather than within the line that begins the chunk*. This makes it straightforward to edit chunk options within more structured editors, and it better accommodates longer options. However, it is still possible to include chunk options on the first line in Quarto. As already mentioned, Quarto can be used to create a range of output formats, and many more of these outputs are inbuilt into the system with more options, compared with R Markdown, for example, websites, books, and blogs.[2]

1.3 | Structure of the paper

This paper will highlight the benefits of using Quarto, together with a guide on getting started with the system. The author will also draw on experiences from delivering a presentation at UKCOTS 2024: *Teaching Statistics in Higher Education, across all disciplines* in June 2024; see

Reference 10. At this conference, the author delivered an introduction to Quarto, indicating its benefits in teaching statistics, together with a live demonstration on how to produce various outputs, see Reference 11. Furthermore, the author's experiences of using Quarto for producing teaching resources will be shared, together with feedback from students who have used Quarto-produced resources for their statistics learning.

A quick-start guide to Quarto will be provided in Section 2. This guide will include information about downloading the essential tools and how to produce basic outputs.

Section 3 covers the author's experiences of working with Quarto to produce statistics teaching resources using primarily R, but also Python. In addition, feedback from student users will be presented and evaluated in this section.

Finally, Section 4 highlights potential future developments using Quarto. These include enhancements based on the student feedback described in Section 3 and the use of Observable JS to produce interactive, mathematical diagrams.

2 | QUARTO—QUICK-START GUIDE

2.1 | Downloading essential tools

The Quarto website[2] (https://quarto.org/) contains step-by-step guides on setting up Quarto along with a wealth of information and useful resources for creating a variety of outputs. If you are new to Quarto, you may find it beneficial to return to this section for a second reading after installing and trialing Quarto.

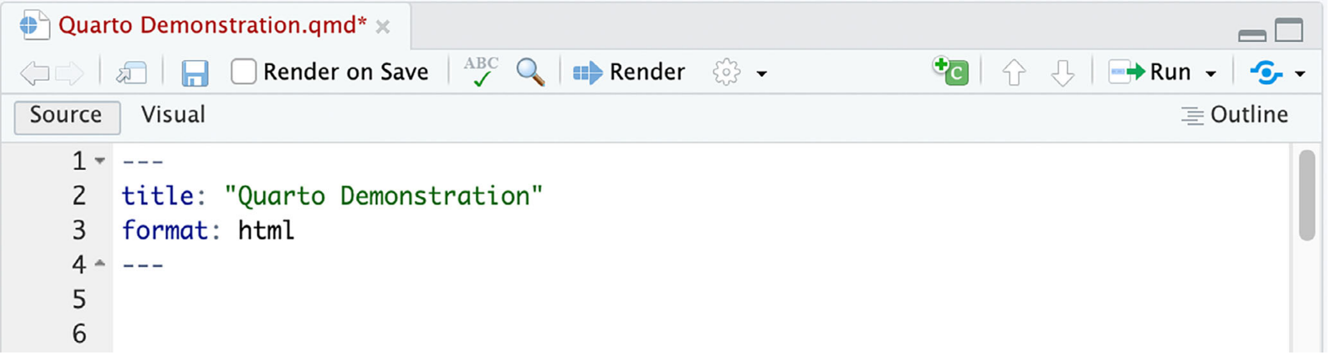
Quarto can be used with a variety of editors, for example, R Studio or VS Code; therefore, you may want to use your editor of choice when creating Quarto documents. However, the developers advise that if you use either the Jupyter or Knitr engine, then you will likely be better off using JupyterLab or VS Code (for.ipynb notebooks) or RStudio (for.qmd documents). It is also advisable to keep Quarto and your choice of editor as up-to-date as possible.

2.2 | Producing basic materials using Quarto

Once Quarto has been set up correctly with a suitable editor, there are a number of basic points to consider when starting to use the system. Firstly, after opening a Quarto document, YAML syntax is used to format the Quarto document. This code is contained within --- symbols—see Figure 1. YAML originally stood *for Yet Another Markup Language*, but it has now been updated to YAML *Ain't Markup Language*. This was to emphasize its data-oriented purpose rather than for markup/documents. Formally speaking, Reference [12] states that *YAML is a human-friendly data serialization language for all programming languages*. In this context, YAML syntax is used for formatting the Quarto document. Figure 1 provides an example of very simple YAML code for a Quarto document entitled *Quarto Demonstration* to produce an HTML output using RStudio as an editor.

Outputs are produced by rendering your Quarto file. It is also possible to *render on save*—this option automatically renders your document whenever you save. Both of these functions can be seen in Figure 1 above. Even though you may need to select an output type when creating a Quarto document, it is possible to change the output type by adapting the "*format*" in the YAML code—thus different output types are possible from the same source file by modifying the YAML code.

We can also see in Figure 1 that we have *Source* and *Visual* editors in RStudio. The Source editor is for users who are confident in coding their Quarto document, whereas the Visual editor has a more detailed user interface, providing a more user-friendly typesetting. This is akin to, but not as extreme as, using LaTeX versus Word.



```
---
title: "Quarto Demonstration"
format: html
---


```

FIGURE 1 Example YAML code in a Quarto document using RStudio. [Colour figure can be viewed at wileyonlinelibrary.com]

It is important to note that if a user starts with the Visual editor, they can change to the Source editor to see the corresponding code—this should allow for a seamless transition to the coding editor for any user who wishes to do so.

Text can just be typed into your Quarto document, and Quarto's support for LaTeX allows for mathematical statements to be included in a straightforward way. See Figure 2 for an example of text and LaTeX code entered into a Quarto document.

Rendering the code in Figure 2 in Quarto using the HTML format obtains the output in Figure 3.

2.3 | Adding code blocks

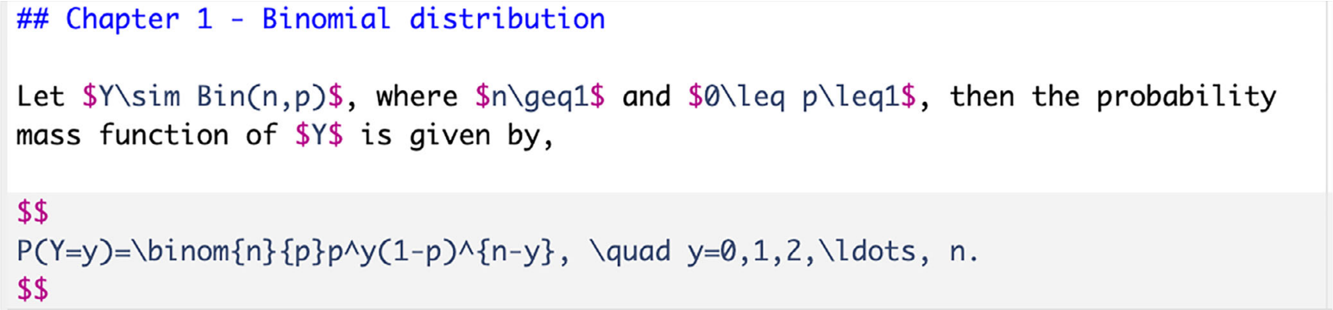
Code blocks are encapsulated within ``` symbols in Quarto, followed by the language name. If the language name is contained within braces, Quarto will run the code. Otherwise, the code will be displayed in the output but not run.

See Figure 4 for an example of some simple R code in Quarto for creating a graph of the Poisson distribution with parameter $\lambda = 3$.

Below is the output obtained by rendering the Quarto document containing the code block in Figure 4 using the HTML format (Figure 5).

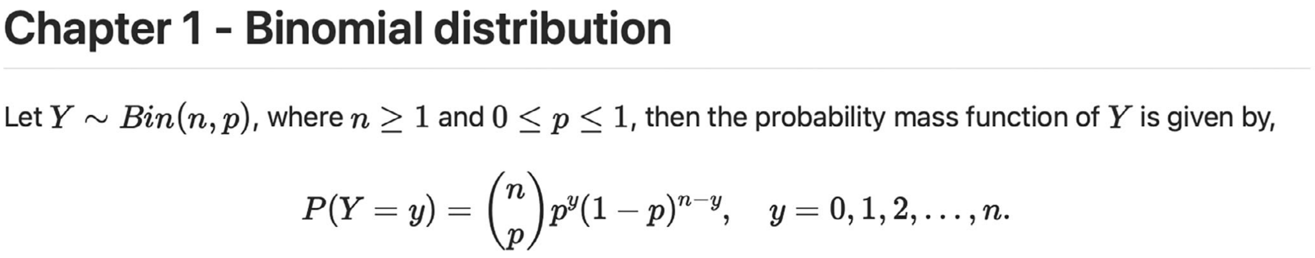
Finally, we input some text and Python code to perform some basic descriptive statistics using the *NumPy* library in Python. Figure 6 shows the Quarto document in RStudio, and Figure 7 illustrates the associated output in HTML format.

Below is the output obtained by rendering the Quarto document containing the code block in Figure 6 using the HTML format.



```
## Chapter 1 - Binomial distribution

Let $Y\sim Bin(n,p)$, where $n\geq1$ and $0\leq p\leq1$, then the probability
mass function of $Y$ is given by,

$$
P(Y=y)=\binom{n}{p}p^y(1-p)^{n-y}, \quad y=0,1,2,\ldots, n.
$$
```

FIGURE 2 Example of simple text and LaTeX code entered into Quarto. [Colour figure can be viewed at wileyonlinelibrary.com]

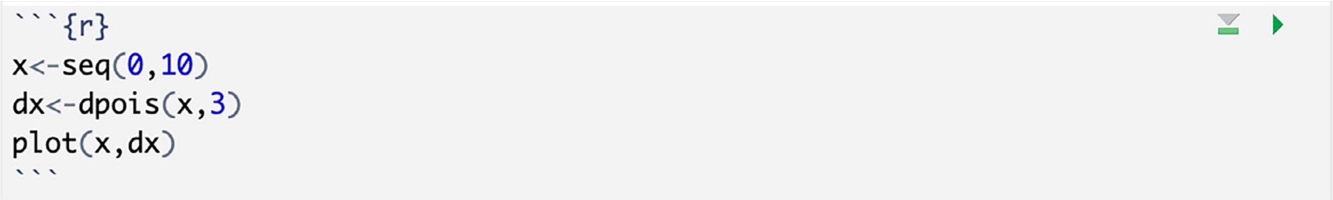


Chapter 1 - Binomial distribution

Let $Y \sim Bin(n,p)$, where $n \geq 1$ and $0 \leq p \leq 1$, then the probability mass function of Y is given by,

$$P(Y=y)=\binom{n}{p}p^y(1-p)^{n-y}, \quad y=0,1,2,\ldots, n.$$

FIGURE 3 Output of Figure 2 when rendered in Quarto in the HTML format.



~~~
```{r}
x<-seq(0,10)
dx<-dpois(x,3)
plot(x,dx)
```
~~~

FIGURE 4 Example of R code block in Quarto. [Colour figure can be viewed at wileyonlinelibrary.com]

Note that in Figure 7, we can see both the code block and its output. In addition, if the learner were to hover over the code block, a copy symbol would appears for copying into coding editors. The coding language inside the braces in Figure 6 can be changed to another compatible one if required.

It is worth noting that the #| *echo: false* option results in the code not being printed in your output, which could be useful in certain scenarios. In addition, the #| *code-fold: true* option may be useful in some scenarios, as it results in having to click on *Code* in order to view the code in the output.

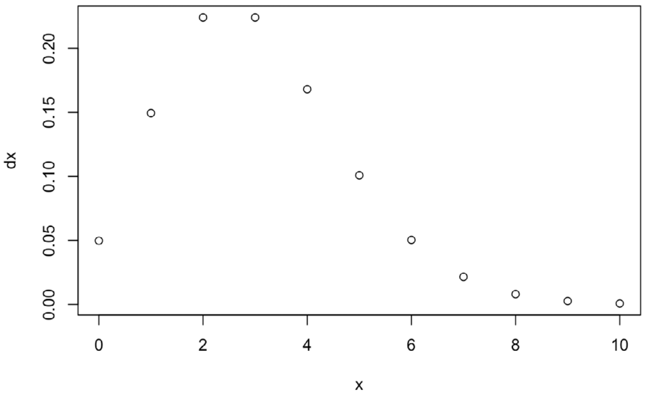


FIGURE 5 Output of the code in Figure 4 when rendered in Quarto.

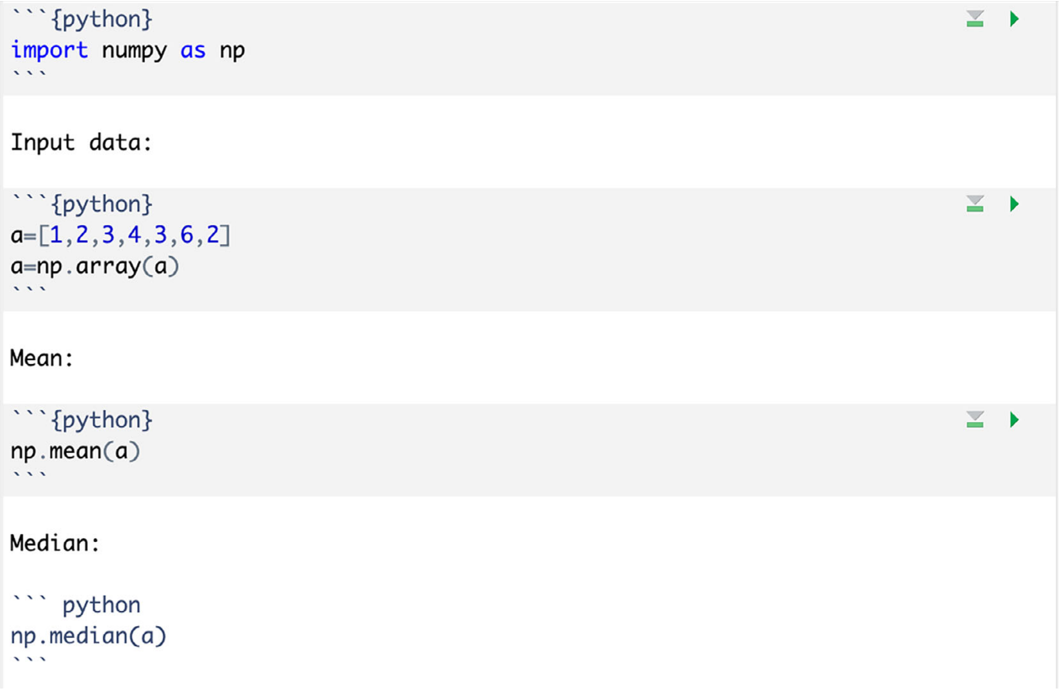
2.4 | Example template and hosting resources

In order to act as a template for any reader wishing to try out Quarto, the Quarto file used to produce the outputs in this section is available with this document.

Any outputs produced by Quarto will need to be hosted in a suitable place that is accessible to students. This is particularly the case for HTML outputs/webpages because Virtual Learning Environments do not usually support such outputs. It may be useful to note that Quarto provides a free publishing service, Quarto Pub, for content produced by Quarto, including webpages.

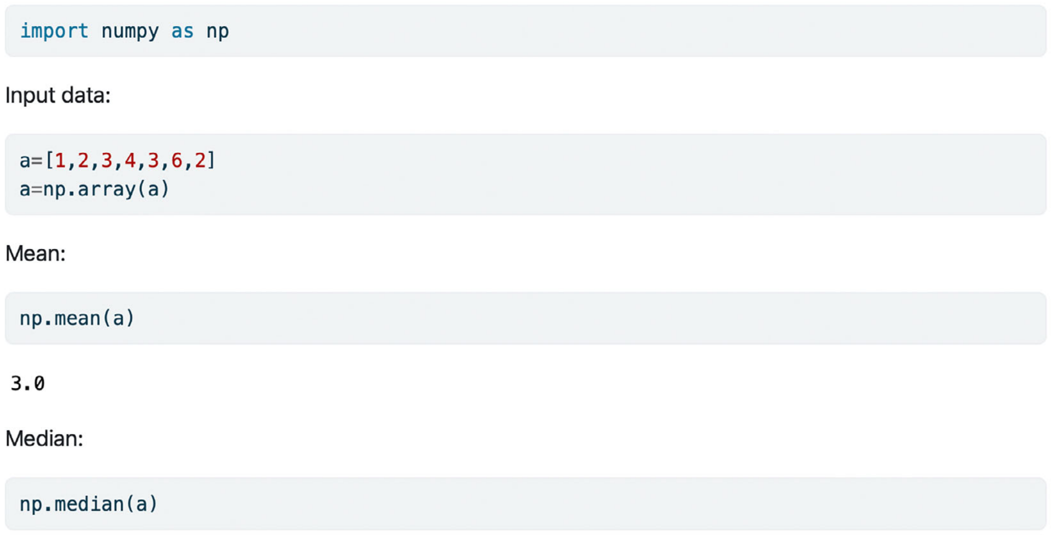
2.4.1 | Recommendations for beginners

- Download the latest version of Quarto;
- Download the latest version of your chosen editor (e.g., R Studio, VS Code, and JupyterLab);
- First create a basic text Quarto output;
- Slowly introduce more advanced features, for example, code blocks and interactive features, and continuously check outputs;
- If your preferred output type is HTML, plan in advance where this will be hosted and consider a backup host in case of any server issues;
- Quarto is developing quickly, therefore, it is advised to keep the system and your editor continuously updated.



~~~
```{python}
import numpy as np
```

Input data:

```{python}
a=[1,2,3,4,3,6,2]
a=np.array(a)
```

Mean:

```{python}
np.mean(a)
```

Median:

``` python
np.median(a)
```
~~~

FIGURE 6 Example of text and Python code block in Quarto. [Colour figure can be viewed at wileyonlinelibrary.com]



```
import numpy as np
```

Input data:

```
a=[1,2,3,4,3,6,2]
a=np.array(a)
```

Mean:

```
np.mean(a)
```

```
3.0
```

Median:

```
np.median(a)
```

FIGURE 7 Output of the code in Figure 6 when rendered in Quarto. [Colour figure can be viewed at wileyonlinelibrary.com]

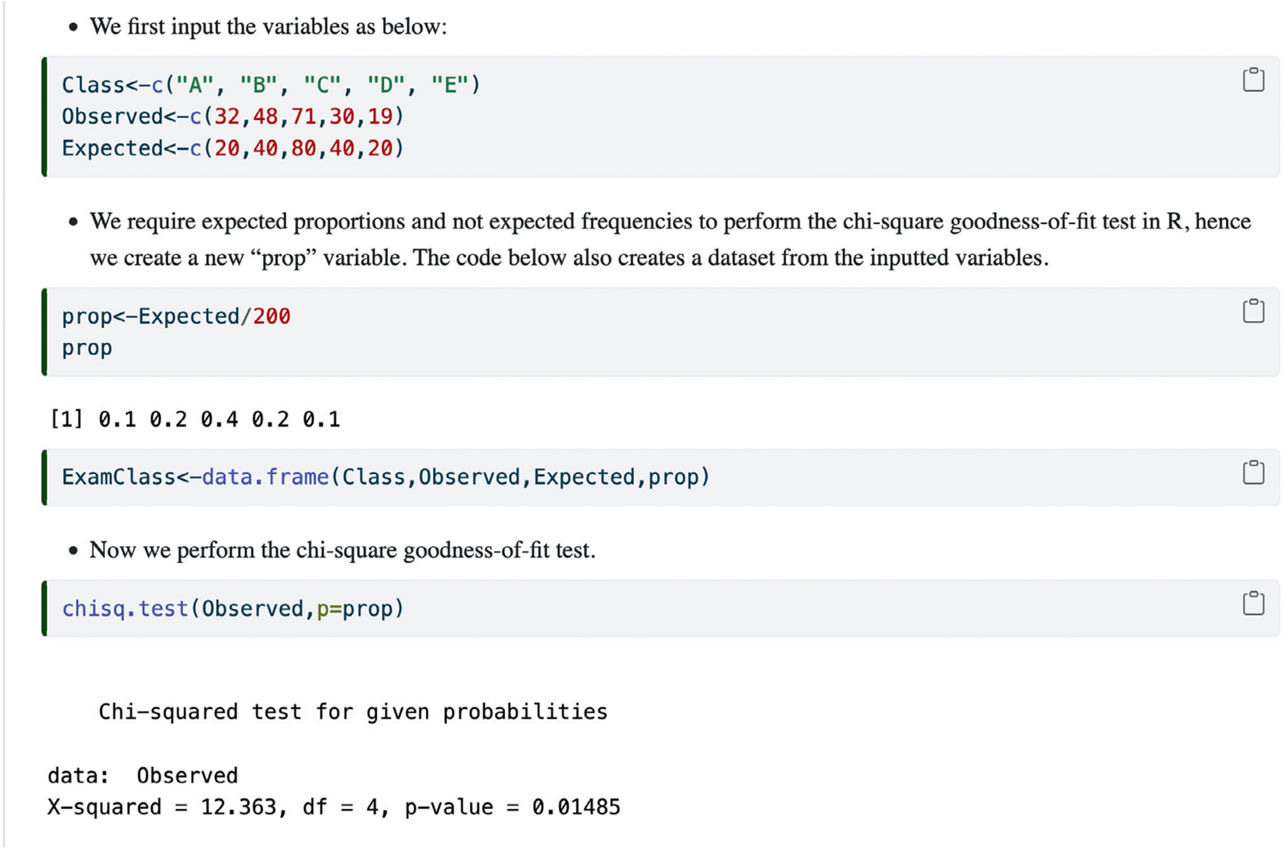
3 | THE AUTHOR'S EXPERIENCES AND STUDENT FEEDBACK

As mentioned in Section 1, the author started using Quarto to produce teaching resources to try to find a suitable balance between students spending significant time writing out code (and hence not covering as much material) and students being provided with the working code (and hence the risk of students not understanding the coding steps). Even though this initiative was approached primarily from an R perspective, the author had not used R Markdown before and decided to move straight to Quarto, as it was the most up-to-date system and because of its benefits over R Markdown, as highlighted in Section 1. The author has primarily used Quarto to create R lab teaching resources for a second-year statistics module; however, he has also used the system while contributing to teaching Python to master's-level data science students and for producing interactive outputs using Observable JS. The section will focus on the author's experiences of using Quarto to produce high-quality, interactive R-focused lab teaching materials for his second-year Statistical Data Analysis module. For this module, the most important was the ability to combine plain text, LaTeX code, and R code blocks all in one HTML output. An example of an HTML output produced by Quarto is given in Figure 8:

This example highlights how the output breaks up the example into blocks of explanatory text and small blocks of R code. Students can copy each block of R code across to their editor using the copy symbol in the top right corner of each code block. Students can also check the outputs they obtain with the ones made available in the document. An important point to note here is that students do not need to consume too much time writing out each line of code manually; they can just copy the individual code blocks, but crucially, this must be done one block of code at a time. This should encourage students to run and understand each small block of code, comparing it with the provided outputs. Considering the other lab teaching extreme of providing all example code in advance, this technique allows the lab instructor to keep a handle on the pace of the lab by the design of the output, in particular how the code is broken up and weaved with other important information.

For each week's lab, a webpage was produced in a similar format to the example in Figure 8. In addition to these individual resources for each week, Quarto was used to create a website that acts as a hub for all of the lab teaching resources for this module. This does involve more advanced techniques, but Quarto makes the process more straightforward, creating all of the website files required to host such a website on your server. Not only are all of the lab teaching resources available in one place with this approach, but there are also further pedagogical benefits, including a search function and the ability to provide topic summaries for each lab class. See Figure 9 for a snippet of the website created with Quarto. These features should be beneficial to students for revision purposes.

Note in Figure 9 the search bar at the top left of the page—this allows students to search for a particular term or topic and hence allows them to find the relevant resources quickly. The individual lab sheets are accessed



- We first input the variables as below:

```
Class<-c("A", "B", "C", "D", "E")
Observed<-c(32,48,71,30,19)
Expected<-c(20,40,80,40,20)
```

- We require expected proportions and not expected frequencies to perform the chi-square goodness-of-fit test in R, hence we create a new "prop" variable. The code below also creates a dataset from the inputted variables.

```
prop<-Expected/200
prop
```

```
[1] 0.1 0.2 0.4 0.2 0.1
```

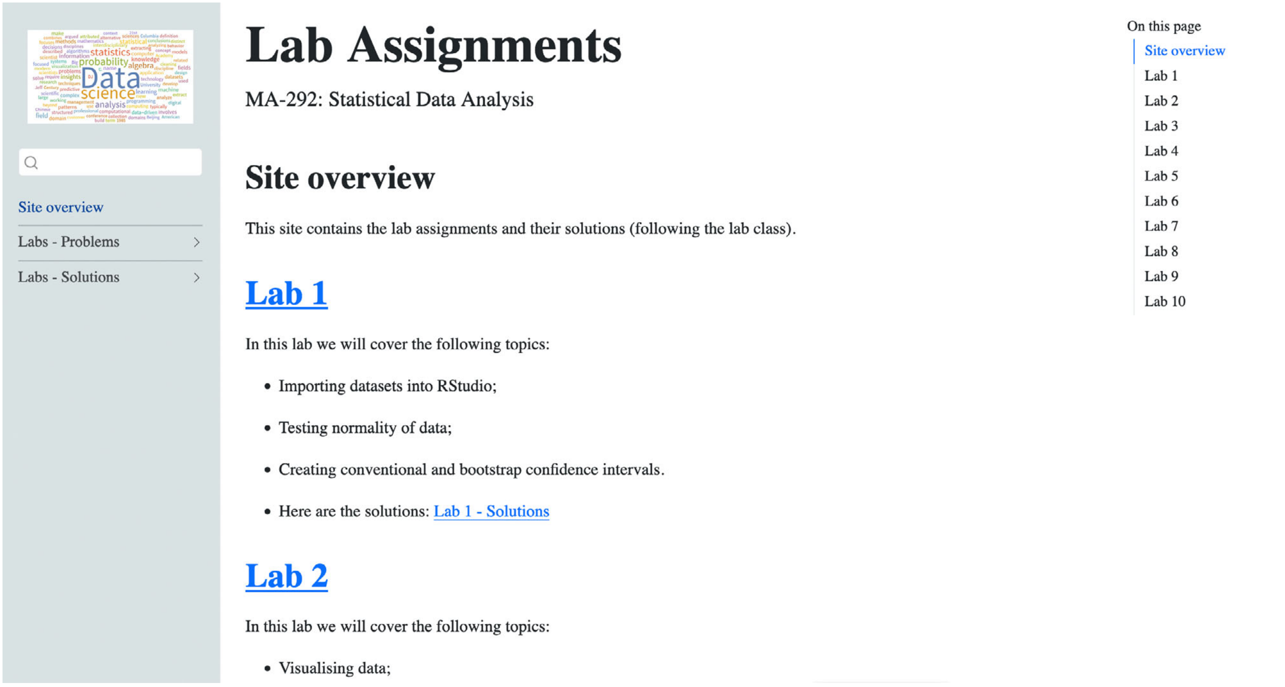
```
ExamClass<-data.frame(Class,Observed,Expected,prop)
```

- Now we perform the chi-square goodness-of-fit test.

```
chisq.test(Observed,p=prop)
```

```

	Chi-squared test for given probabilities

data:  Observed
X-squared = 12.363, df = 4, p-value = 0.01485
```

FIGURE 8 Example HTML output using Quarto. [Colour figure can be viewed at wileyonlinelibrary.com]



Lab Assignments

MA-292: Statistical Data Analysis

Site overview

This site contains the lab assignments and their solutions (following the lab class).

Lab 1

In this lab we will cover the following topics:

- Importing datasets into RStudio;
- Testing normality of data;
- Creating conventional and bootstrap confidence intervals.
- Here are the solutions: Lab 1 - Solutions

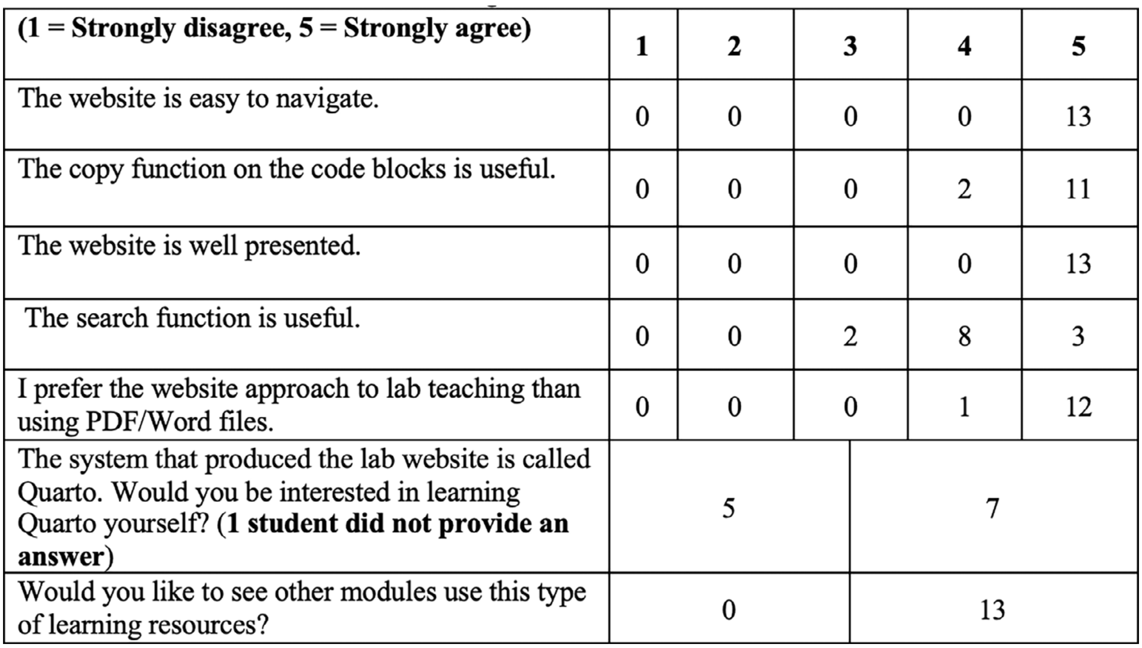
Lab 2

In this lab we will cover the following topics:

- Visualising data;

FIGURE 9 Homepage for lab teaching website. [Colour figure can be viewed at wileyonlinelibrary.com]

by following the relevant lab number (in bold) on the webpage, and then solutions are made available after each lab class by following the solutions link. For any readers interested in investigating the website further, it can be found by following the link: https://math.swan.ac.uk/kevans/MA-292_25/.



(1 = Strongly disagree, 5 = Strongly agree)	1	2	3	4	5
The website is easy to navigate.	0	0	0	0	13
The copy function on the code blocks is useful.	0	0	0	2	11
The website is well presented.	0	0	0	0	13
The search function is useful.	0	0	2	8	3
I prefer the website approach to lab teaching than using PDF/Word files.	0	0	0	1	12
The system that produced the lab website is called Quarto. Would you be interested in learning Quarto yourself? (**1 student did not provide an answer**)	5			7	
Would you like to see other modules use this type of learning resources?	0			13	

FIGURE 10 Results of the student questionnaire.

The author's conclusion from teaching R lab classes using Quarto is that it can be used to produce high-quality, interactive resources that allow the pace of the lab teaching to be better controlled while at the same time providing digestible chunks of new code and materials to students.

3.1 | Student feedback

3.1.1 | Informal feedback

Using students' queries during second-year statistics lab sessions during the 23/24 and 24/25 academic years as a guide, it was found that students were trying to understand the individual steps of the worked examples, and hence, they appeared better prepared for the associated exercises. After the introduction of the Quarto-produced resources, students seemed to have fewer issues with R coding and evaluating R outputs. These cohorts of second-year students were taught using Quarto were taught using more traditional techniques in their first year of study. Anecdotally, many of these students preferred the new approach; many expressed this verbally during the lab classes. In the 23/24 academic year, one student verbally stated that "it is good to have a hub for all lab materials because it helps with revision." The same student was interested in how the materials were coded.

3.1.2 | Formal feedback

In addition, formal feedback was obtained from the students during designated lab sessions, which took place in person during the second semester of the 2024/25 academic year. Thirteen second-year students took part in the survey. Ethical approval was obtained by participating students opting in and signing an ethical approval form. This form was also approved by Swansea University prior to the survey.

The questionnaire provided to students can be found in Figure 10. The entries in the cells denote the number of participants who provided that answer for each question.

3.1.3 | The following open-ended questions were also asked

- *What did you like the most about the website?*
- *What did you dislike the most about the website?*
- *Would you like to see any changes or additional features to the website?*
- *Additional comments:*

In summary, all students strongly agreed with the statements that the website is easy to navigate and that it is well presented. All students either agreed or strongly agreed with the statements that the copy function on the code blocks is useful and that they prefer the website approach to lab teaching over using PDF/Word files. Eleven out of the thirteen student participants either agreed or strongly agreed with the statements that the search function is useful, with the remaining two students neither agreeing nor disagreeing with this statement. It was expected for more students to strongly agree with this statement; however, the survey took place before the final lab-based assignment for the module, which is where the search function would likely be most useful for finding specific topics; therefore, it would be interesting to survey students again about this statement at the end of the module. Seven out of twelve students (one student did not provide an answer) expressed an interest in learning about Quarto itself; however, all 13 participants would like to see other modules use similar types of learning resources.

With respect to the open-ended questions, students commented on the clarity of the website, its ease of use and all resources being in one place as being what they liked most about the website. For example, one student wrote, "*All the content is in one place and the ability to copy and paste example code is extremely helpful*." With respect to what students disliked most about the website, two students stated that they didn't like having to go back to the home page before visiting another lab sheet. This is an excellent point, and one that was not realized when creating the webpage. This feature will be investigated in future versions of the resources. With respect to changes or additional features to the website, there was a remark about the ability to copy equations and to include video tutorials and a formula page. These remarks will be taken into account in any future developments of the website. Finally, all remarks entered under the additional comments option were positive, including:

> Having used Word and R-script files with exercises on as well in first year, I find this method a lot more helpful than using Word. Everything is easy to navigate, it is all complied in one place and the copy function is convenient. I don't really have any negatives about it.

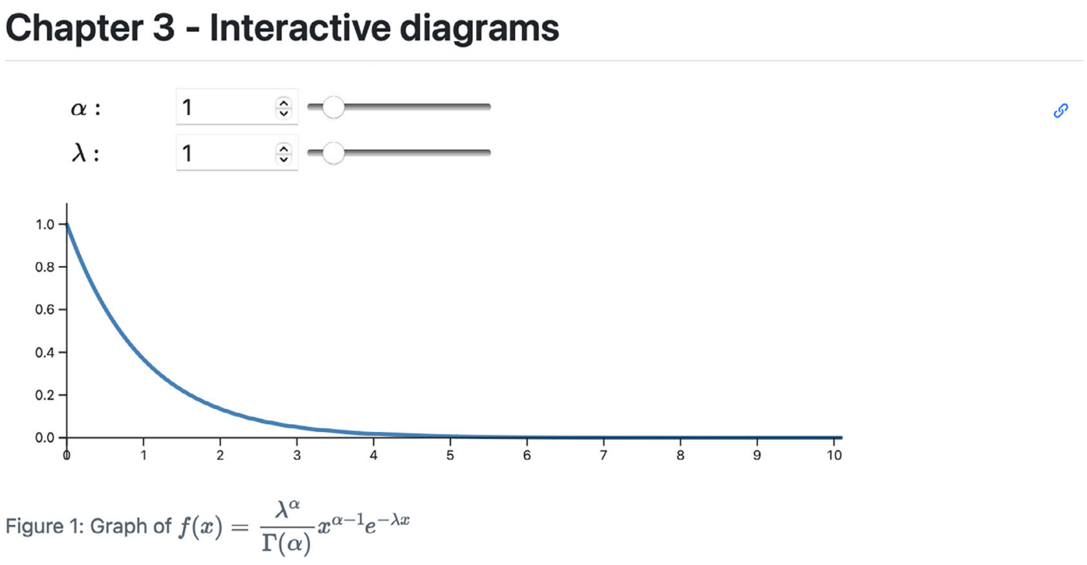
and

> It is very accessible and engaging, making it easy to learn.

4 | FUTURE WORK

Based on student feedback highlighted in the previous section, it would be interesting to survey students again, but following their lab assignment, in order to determine whether more students find the search function of the website more useful for revision. In addition, future improvements to the website will include the ability to move to all lab sheets from any point on the website. Given that all participants would like to see similar resources produced by Quarto used in other modules, this is an area of development that will be investigated.

Thus far, Quarto has only been used to produce outputs for students to use in their learning; that is, the



Chapter 3 - Interactive diagrams

α : 1

λ : 1

Figure 1: Graph of $f(x) = \dfrac{\lambda^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}$

FIGURE 11 HTML output of interactive diagram created using Observable JS. [Colour figure can be viewed at wileyonlinelibrary.com]

students have not interacted with the Quarto system. However, seven students expressed an interest in learning how to use Quarto itself in the survey. A further interesting extension would be to teach students to use Quarto, that is, as a learning objective in and of itself. This has been investigated with R Markdown; for example, see Reference 13.

Finally, following the presentation and demonstration on Quarto delivered at *UKCOTS: Teaching Statistics in HE*, a member of the audience asked whether multiple-choice questions could be incorporated into outputs to further increase their interactivity, which is an interesting concept. This can be achieved by using Quarto extensions, for example, *naquiz*.

A further potential advantage of Quarto is to produce interactive animations/simulations. As stated in Reference 14, animations and simulations work by extending the visual power of static graphical representations to help participants comprehend the properties of statistical distributions more easily. The most effective animations/simulations are also interactive and allow students to change variables and experience the effects of these changes.[15]

Observable JS is especially well suited for interactive data exploration and analysis; however, it can also be used to produce interactive diagrams. In particular, the following interactive diagram for the gamma distribution has been produced in Quarto using Observable JS. Only the output will be highlighted in this article due to the length of the code involved in creating this output.

This interactive diagram was created with the help of the inbuilt library D3 and the Math object, in addition to the external package *math.fn* from npm; see Reference 16. D3 is a JavaScript library for bespoke data visualization; see Reference 17 for further information. Math *contains static properties and methods for mathematical constants and functions*.[18] Please note that in the output highlighted in Figure 11, the parameters α and λ of the gamma distribution can be varied to see the effect this has on the graph of the function.

5 | CONCLUSION

Quarto is a system that allows for the flexibility of producing a variety of outputs from one source file. It provides the opportunity to incorporate more advanced features, including code blocks, code execution, and interactivity. The author's experiences with the system are very positive, and student feedback has confirmed that the teaching resources produced by Quarto have been generally well received. The author looks forward to developing further interactive teaching resources with the system.

ACKNOWLEDGMENTS

The author would like to thank Dr. Dmitri Finkelshtein for his support and guidance on using Quarto. In addition, an acknowledgment is given to the organizers of the 2024 UK Conference on Teaching Statistics (UKCOTS) for the opportunity to present on the Quarto system.

ORCID

Kristian Evans https://orcid.org/0009-0008-7043-1961

REFERENCES

1. rOpenSci. rOpenSci Educators Collaborative: What are the challenges when teaching science with R? R Bloggers. https://www.r-bloggers.com/2018/07/ropensci-educators-collaborative-what-are-the-challenges-when-teaching-science-with-r/.
2. Posit. Quarto. https://quarto.org/.
3. A. Reid and P. Petocz, *Students' conceptions of statistics: A phenomenographic study*, J. Stat. Educ. **10** (2002), no. 2. https://doi.org/10.1080/10691898.2002.11910662.
4. M. Cone. The Markdown Guide: Getting Started. https://www.markdownguide.org/getting-started/.
5. M. Dogucu, A. A. Johnson, and M. Ott, *Framework for accessible and inclusive teaching materials for statistics and data science courses*, J Statis Data Sci Educ **31** (2023), no. 2, 144–150. https://doi.org/10.1080/26939169.2023.2165988.
6. A. Berg and N. Hawila, *Some teaching resources using R with illustrative examples exploring COVID-19 data*, Teach. Stat. **43** (2021), S98–S109. https://doi.org/10.1111/test.12258.
7. S. Finch, I. Gordon, and C. Patrick, *Taking the aRghhhh out of teaching statistics with R: Using R markdown*, Teach. Stat. **43** (2021), no. S143, S143–S147. https://doi.org/10.1111/test.12251.
8. Observable, Inc. Learn Just Enough JavaScript: Introduction. https://observablehq.com/@observablehq/learn-javascript-introduction?collection=%40observablehq%2Ftutorial.
9. Observable, Inc. Observable. https://observablehq.com/.
10. RSS Teaching Statistics Section. UKCOTS. https://www.ukcots.org/.
11. RSS Teaching Statistics Section. UKCOTS-2024-Presentations/Thursday Presentations/Using Software (1145–1300). https://github.com/rssteachingstats/UKCOTS-2024-Presentations/tree/main/Thursday%20Presentations/Using%20Software%20(1145-1300).
12. YAML. https://yaml.org/.
13. B. Baumer, M. Cetinkaya, A. Bray, L. Loi, and N. Horton, *R markdown: Integrating a reproducible analysis tool into introductory statistics*, Technol. Innov. Stat. Educ. **8** (2014). https://doi.org/10.5070/T581020118.
14. E. A. Boyle, E. W. MacArthur, T. M. Connolly, et al., *A narrative literature review of games, animations and simulations to teach research methods and statistics*, Comput. Educ. **74** (2014), 1–14. https://doi.org/10.1016/j.compedu.2014.01.004.
15. P.-Y. Wang, B. K. Vaughn, and M. Liu, *The impact of animation interactivity on novices' learning of introductory statistics*, Comput. Educ. **56** (2011), no. 1, 300–311. https://doi.org/10.1016/j.compedu.2010.07.011.
16. npm, Inc. npm. https://www.npmjs.com/.
17. M. Bostock and Observable, Inc. D3. https://d3js.org/.

18. MDN Web Docs. Math. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: K. Evans, *Innovative and interactive statistics teaching using Quarto*, Teach. Stat. (2025), 1–11, DOI 10.1111/test.12409