PharmaSUG 2024 - Paper DV - 456

An introduction to Quarto: A Versatile Open-source Tool for Data Reporting and Visualization

Joshua J. Cook, M.S., ACRP-PM, CCRC

ABSTRACT

In the collaborative landscape of data analysis, a common frustration among analysts stems from the need to integrate and harmonize different programming languages within a team. Teams often comprise interdisciplinary researchers, each with their unique programming preferences and expertise, leading to complexities in project integration and continuity. The difficulty in compiling and executing data projects cohesively can hinder efficiency and impede the delivery of coherent, multi-faceted data insights. This challenge necessitates a solution that can bridge the gaps between varying coding languages and methodologies to streamline team collaboration and project completion.

Addressing this issue, the point of this paper is to present Quarto as an innovative solution that can unify the diverse programming approaches within a team. Quarto stands out by offering extensive cross-language support, enabling the integration of code from multiple languages into a singular, dynamic report. This versatile reporting system is tailored for the pharmaceutical and biotech industries, facilitating the creation of comprehensive reports and visualizations that cater to stakeholders at all technical levels. With Quarto, consolidating code, narrative text, and outputs into one document is seamless, accommodating outputs in various formats and types such as HTML, PDF, Word, Typeset, Markdown, PowerPoint, dashboards, websites, manuscripts, and even entire books. This paper serves as an introduction to Quarto's capabilities, highlighting its role in enhancing collaboration and efficiency in data science projects across the spectrum of technical expertise.

INTRODUCTION

DATA SCIENCE

There are many ways to define *data science*, but IBM has defined it as:

> data science combines math and statistics, specialized programming, advanced analytics, artificial intelligence (AI) and machine learning with specific subject matter expertise to uncover actionable insights hidden in an organization's data. These insights can be used to guide decision making and strategic planning ("What is Data Science?" n.d.).

Thus, data science has emerged as a cornerstone of innovation and strategic decision-making across industries, from healthcare and pharmaceuticals to finance and technology. Its importance cannot be overstated, as it enables organizations to derive meaningful insights from vast and complex datasets, driving efficiency, innovation, and competitive advantage (Davenport and Patil 2012). However, the power of data science is fully realized only when insights are effectively communicated to stakeholders through comprehensive reporting and visualization. These processes are crucial for translating complex data analyses into actionable intelligence that can guide business strategies and scientific discoveries (Knaflic 2015). The general "game" of data science, as described by Wickham and Grolemund, is summarized in Figure 1 (Wickham and Grolemund 2016).

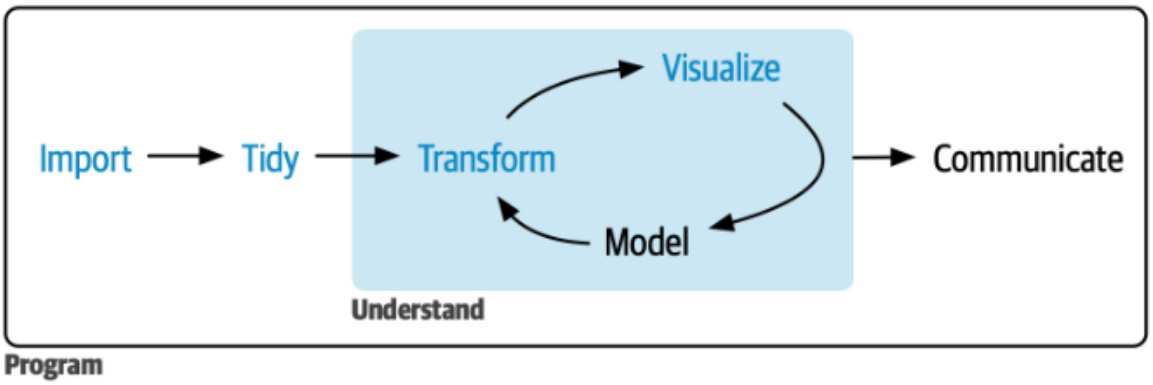

Figure 1. The “whole game” of data science, from the initial import of raw data to the process of communicating findings to stakeholders. Each phase of this process could be approached differently depending on the background and skillset of an individual data scientist. To enable interdisciplinary collaboration, it is essential that flexible tools and systems be introduced and made the standard for reporting processes.

Despite the clear benefits, the interdisciplinary nature of data science teams often introduces significant challenges. Data scientists come from various backgrounds, wielding expertise in different programming languages and methodologies. For instance, some may prefer R for its statistical and data visualization capabilities, while others might lean towards Python for its versatility and extensive library support (Wickham and Grolemund 2016; VanderPlas 2016). This diversity, although valuable for bringing a breadth of perspectives and skills to a project, complicates collaboration and integration efforts. The need to consolidate analyses conducted in different programming languages into a cohesive report can hinder efficiency and delay the delivery of insights.

Moreover, the necessity for an interdisciplinary approach to data science projects is becoming increasingly clear. As data problems grow in complexity, leveraging the strengths of diverse programming languages and methodologies is essential for developing comprehensive solutions (Blei and Smyth 2017). This calls for tools and systems that can facilitate seamless collaboration among data scientists with varied expertise.

QUARTO

In this context, Quarto (often described as the next generation of R Markdown) emerges as a promising solution capable of bridging the gaps between different programming languages and enabling the efficient integration of diverse methodologies. By supporting extensive cross-language operations, Quarto allows teams to collaborate more effectively, streamlining the process of compiling and executing data projects. This paper aims to introduce readers to Quarto's potential to enhance interdisciplinary collaboration in data science, particularly focusing on its utility for reporting and visualization in the pharmaceutical and biotech industries. For the purposes of this paper, **Quarto Project Types** refer to a type of organization of files (ex: report, website, presentation) while **Quarto Project Formats** refer to the output file type of a document (ex: HTML, PDF, DOCX).

MARKDOWN REPORTS – THE MOST BASIC QUARTO PROJECT TYPE

SETUP IN RSTUDIO AND QUARTO PROJECTS

Quarto offers several options regarding Integrated Development Environments **(IDEs),** including:

- **Visual Studio (VS) Code.**
- **Jupyter.**
- **RStudio.**

- **Neovim.**
- **Basic text editors.**

Why would someone using a notebook-style system, with markdown and code cells, like Jupyter want to use Quarto? Because Quarto still offers a vast array of enhancements including cross-references, multiple output formats, and document creation versatility!

Feel free to start with your tool of choice, but this demonstration utilized **RStudio®** as the primary **IDE.** RStudio is also developed by Posit (RStudio Desktop - Posit), and thus comes with Quarto already built-in. Tool-specific instructions can be found within the Quarto documentation if you prefer a different tool, such as Jupyter or VS Code (Quarto – Get Started).

As with any project in RStudio®, a primary folder should be setup that will be used as the working directory. This is where the Quarto project (*.Rproj*) file will be housed, as well as any required subfolders. Quarto Projects are directories that provide a central location for all files, images, and outputs. Furthermore, Quarto Projects provide:

- Shared YAML configuration and metadata (ex: tables of contents and bibliographies).
- Organized rendering of multiple project formats from a single file (ex: PDF and HTML) and multiple project types (ex: websites and books).
- Package and environment version control (for regulatory purposes).
- Directory-specific settings and preferences (ex: code formatting and view panes).
- Virtual Environments.

There are *many* Quarto project types available, but for this first example, we will create a new **Quarto Project** (in the form of a report) within RStudio® using the "Create Project" button in the toolbar, as shown in Figure 2 and Figure 3. Choose any name and location for your project, just make sure you can remember and find it!

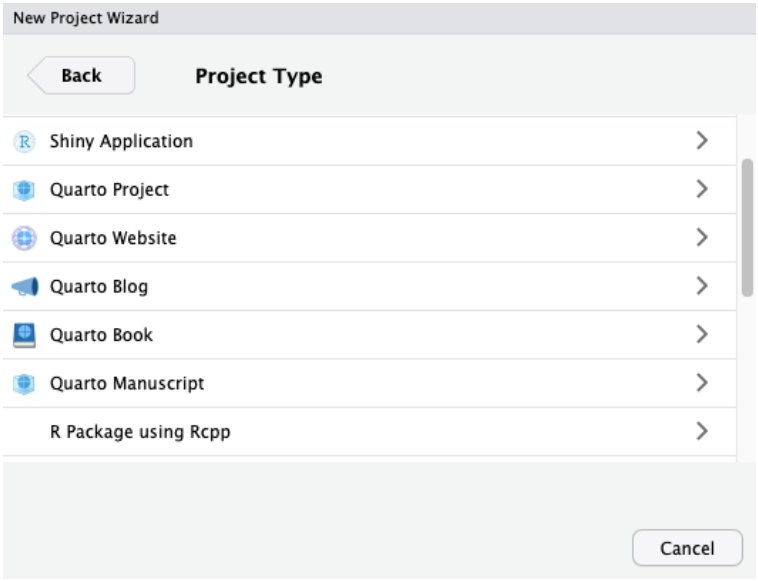

Figure 2. An example of some of the new Quarto project types that can be created within RStudio® selecting the "Create Project" button.

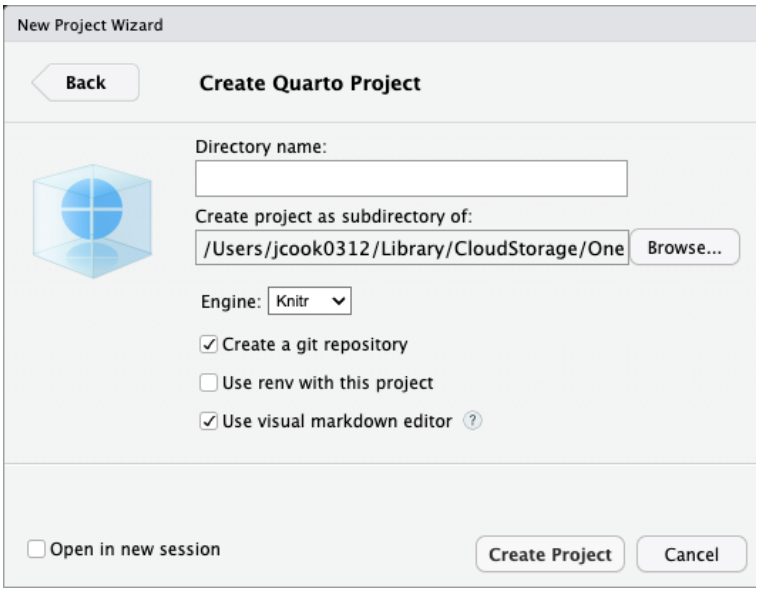

Figure 3. The options when setting up a new Quarto Project. The directory can be any name, and we recommend starting with the visual markdown editor. Git repository use is optional, but highly recommended if this project is collaborative or if the output of the analyses will need to be shared via the web.

These examples use the following:

```
C:\PharmaSUG2024\QuartoIntro.Rproj
C:\PharmaSUG2024\QuartoIntro.qmd
```

Create subordinate folders for resources that will be required by markdown, such as data and images:

```
C:\PharmaSUG2024\data\
C:\PharmaSUG2024\images\
```

Your Quarto Project will open to a template Quarto Markdown *(.qmd)* file with a YAML header, some markdown text, and an example R code chunk, as shown in Figure 4. If another programming language is preferred, {r} can be changed to {python}, etc. The tool bar gives text-editing options for markdown and allows for the creation of more chunks as needed. This is just the basic layout, and many more components can be added to a Quarto Markdown to enrich content.

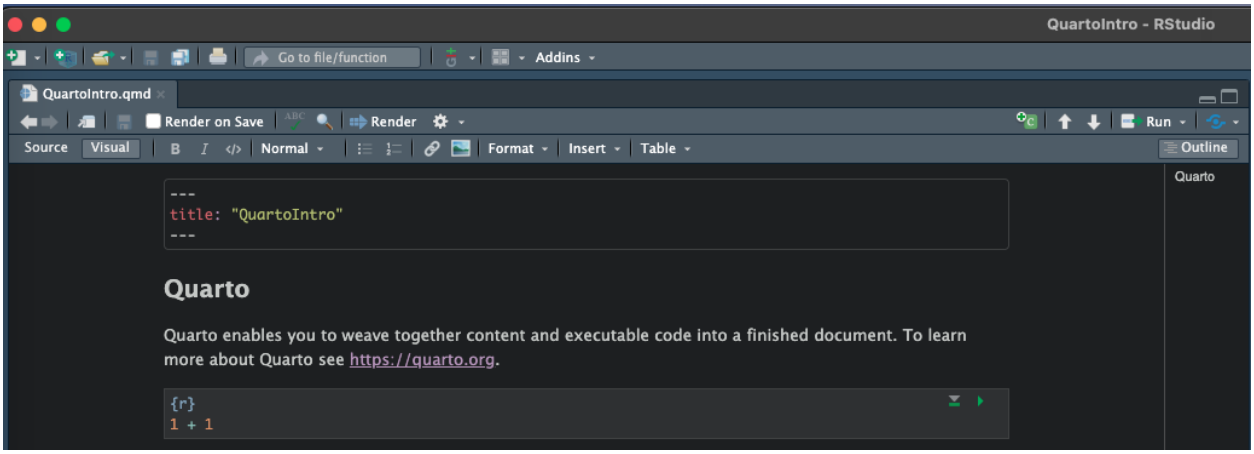

Figure 4. The basic template of Quarto Projects, which includes a YAML header, some markdown text, and an example R code cell.

For this example report, we are going to use only three packages which are installed and imported– `tidyverse`, which is used for data wrangling and visualization, `gt`, which is used for gold-standard table generation, and `quarto`, which is the open-source publishing system used throughout this paper The code for this can be placed in the initial R code chunk. Finally, an example dataset is imported:

```
packages <- c("tidyverse", "gt", "quarto")

if (any(!sapply(packages, requireNamespace, quietly = TRUE))) {
  install.packages(packages[!sapply(packages, requireNamespace, quietly
= TRUE)])
}

library(tidyverse)
library(gt)
library(quarto)

clinical_trials_fl <- readRDS("query_results_clean_final.rds")
```

EXAMPLE DATA

This paper utilizes a publicly available dataset that has been extracted from the Aggregate Analysis of ClinicalTrials.gov **(AACT)** database ("Clinical Trials Transformation Initiative" n.d.). This dataset, `clinical_trials_fl`, was generated by querying the AACT database using open-source tools like R, RStudio, DBI, PostgreSQL, the tidyverse suite, and gt (also accomplished using a Quarto environment). Specifically, this dataset represents all the clinical trials registered and completed within the state of Florida of the United States between 23MAR2014-23MAR2024 (10 years). This dataset currently includes 73,500 observations of 39 variables, relating to both the protocol designs and the completion of the clinical trials, which are detailed in Table 1. All conditions and interventions are included for analysis, and the dataset was cleaned and wrangled by the authors for use in this report.

Index	Variable	Meaning
1	nct_id	(primary key) NCT ID is a unique identification code given to each clinical study registered on ClinicalTrials.gov. The format is the letters "NCT" followed by an 8-digit number (for example, NCT00000419).
2	start_date	Registration date of the study.
3	completion_date	Completion date of ALL endpoints of the study.
4	study_type	Type of study (Interventional, Observational, Expanded Access).
5	acronym	Acronym of the study (if applicable).
6	baseline_population	The intended source of the enrollment population (ex: clinic, associated hospital).
7	brief_title	Brief title of the study that was analyzed.
8	official_title	Official study (protocol) title.
9	overall_status	Overall status of the study (Completed, Terminated, Enrolling, etc.).
10	phase	Phase of clinical development (Early, I-IV, combinations).

11	enrollment	Enrollment numbers. According to the planned/actual enrollment information provided in the protocol section of the study record.
12	enrollment_type	Describes if the enrollment number is planned or actual.
13	source	The sponsor of the study.
14	number_of_arms	The number of treatment arms in the study.
15	why_stopped	If stopped prematurely, an explanation of why.
16	has_expanded_access	Logical indicating if EA was utilized.
17	is_fda_regulated_drug	Logical indicating if the study involved an FDA regulated drug.
18	is_fda_regulated_device	Logical indicating if the study involved an FDA regulated device.
19	site	Site name (usually multi-site).
20	city	Location of the site.
21	state	Location of the site (all Florida in this dataset).
22	zip	Location of the site.
23	country	Location of the site (all United States in this dataset).
24	number_of_facilities	Number of facilities in the U.S.
25	actual_duration	Duration (in months) from start_date until the completion of the primary endpoint.
26	were_results_reported	Logical indicating if results were submitted to ClinicalTrials.gov.
27	has_us_facility	Logical indicating if the study had a site in the U.S. (all true in this dataset).
28	has_single_facility	Logical indicating if the study had only one site.
29	minimum_age_num	Minimum age enrolled in the study.
30	maximum_age_num	Maximum age enrolled in the study.
31	minimum_age_unit	Unit of minimum age.
32	maximum_age_unit	Unit of maximum age.
33	number_of_primary_outcomes_to_measure	Number of primary outcomes listed in the protocol.
34	number_of_secondary_outcomes_to_measure	Number of secondary outcomes listed in the protocol.
35	number_of_other_outcomes_to_measure	Number of teritary+ outcomes listed in the protocol.
36	condition_name	Name of the studied condition (indication).

37	condition_name_lower_case	Lowercase name of the studied condition (indication).
38	duration_months	Full duration of the study (in months) from start_date to completion_date.
39	duration_years	Full duration of the study (in years) from start_date to completion_date.

Table 1. All Variables Included in the `clinical_trials_fl` dataset extracted from the AACT database ("Clinical Trials Transformation Initiative" n.d.).

YAML

Every Quarto Markdown *(.qmd*) file begins with a **YAML** (Yet Another Markup Language) header, which is a versatile human-readable data serialization language. YAML is primarily used in configuration files and data representation in the form of **maps** (key-value pairs) and **lists** (sequences of values). YAML is delineated with three dashes (---) and provides the configuration of a Quarto document with many fields available for specification depending on the quarto project format (ex: HTML and PDF). Some examples of YAML fields that we have found especially helpful include:

- **Title/subtitle** – titles of the document.
- **Date** – date of the document (*can be "today").*
- **Author** – listing of authors for the document.
- **Abstract** – provides a summary of the document.
- **Theme** – theme of the entire document (*many formats available*).
- **Toc** – table of contents (*many formats available*).
- **Bibliography** – document bibliography (*BibTeX or CSL*).
- **Crossref** – configuration for cross-reference labels and prefixes.
- **Format** – designates the output format (*HTML by default*).
- **Embed-resources** - incorporates external dependencies into a standalone HTML file (i.*e., self-contained files*).

For the full list of YAML options, please refer to the Quarto documentation (Quarto – Reference). Let's create our first YAML header, with as many of these fields as possible for customization:

```
---
title: |
  ![](images/logo.png){fig-align="left" width="273"}  \
    \
    PharmaSUG 2024: Introduction to Quarto
subtitle: "A Versatile Open-source Tool for Data Reporting and
Visualization "
date: "today"
author: "Joshua J. Cook, M.S., ACRP-PM, CCRC"
abstract: "In the collaborative landscape of data analysis, a common
frustration among analysts stems from the need to integrate and
harmonize different programming languages within a team. Teams often
comprise interdisciplinary researchers, each with their unique
programming preferences and expertise, leading to complexities in
```

```
project integration and continuity. The difficulty in compiling and
executing data projects cohesively can hinder efficiency and impede the
delivery of coherent, multi-faceted data insights. This challenge
necessitates a solution that can bridge the gaps between varying coding
languages and methodologies to streamline team collaboration and
project completion. Addressing this issue, the point of this paper is
to present Quarto as an innovative solution that can unify the diverse
programming approaches within a team. Quarto stands out by offering
extensive cross-language support, enabling the integration of code from
multiple languages into a singular, dynamic report. This versatile
reporting system is tailored for the pharmaceutical and biotech
industries, facilitating the creation of comprehensive reports and
visualizations that cater to stakeholders at all technical levels. With
Quarto, consolidating code, narrative text, and outputs into one
document is seamless, accommodating outputs in various formats such as
HTML, PDF, Word, Typeset, Markdown, PowerPoint, dashboards, websites,
manuscripts, and even entire books. This paper serves as an
introduction to Quarto's capabilities, highlighting its role in
enhancing collaboration and efficiency in data science projects across
the spectrum of technical expertise."

theme: minty

toc: true

bibliography: references.bib

csl: asa.csl

format: html

embed-resources: true

---
```

Importantly, we specified a few *dependencies* that must be present in the working directory for the document to render, including a logo image within the title, a bibliography file *(.bib),* and a citation style language document *(.csl).* These dependencies will be built-in to the HTML file which can be shared to other users. Not embedding the dependencies will require attaching them manually when the file is shared. The citation style here follows the American Statistical Association (**ASA**), but many more styles are available for use through Creative Commons (citation-style-language/styles: Official repository for Citation Style Language (CSL) citation styles. (github.com)) and Zotero (Zotero Style Repository). The document generated by rendering (by pressing the "Render" button in the toolbar) our YAML is shown in Figure 5. References and bibliography are not shown but will be utilized in the next steps of the document.

Table of contents

This is a Markdown Section

References

PharmaSUG 2024: Introduction to Quarto

A Versatile Open-source Tool for Data Reporting and Visualization

AUTHOR
Joshua J. Cook, M.S., ACRP-PM, CCRC

PUBLISHED
March 28, 2024

ABSTRACT

In the collaborative landscape of data analysis, a common frustration among analysts stems from the need to integrate and harmonize different programming languages within a team. Teams often comprise interdisciplinary researchers, each with their unique programming preferences and expertise, leading to complexities in project integration and continuity. The difficulty in compiling and executing data projects cohesively can hinder efficiency and impede the delivery of coherent, multi-faceted data insights. This challenge necessitates a solution that can bridge the gaps between varying coding languages and methodologies to streamline team collaboration and project completion. Addressing this issue, the point of this paper is to present Quarto as an innovative solution that can unify the diverse programming approaches within a team. Quarto stands out by offering extensive cross-language support, enabling the integration of code from multiple languages into a singular, dynamic report. This versatile reporting system is tailored for the pharmaceutical and biotech industries, facilitating the creation of comprehensive reports and visualizations that cater to stakeholders at all technical levels. With Quarto, consolidating code, narrative text, and outputs into one document is seamless, accommodating outputs in various formats such as HTML, PDF, Word, Typeset, Markdown, PowerPoint, dashboards, websites, manuscripts, and even entire books. This paper serves as an introduction to Quarto's capabilities, highlighting its role in enhancing collaboration and efficiency in data science projects across the spectrum of technical expertise.

Figure 5. The document generated by our YAML specifications. References and bibliography are not shown but will be utilized in the next steps of the document.

MARKDOWN AND CODE CHUNKS

Quarto documents offer both **markdown** (which can capture formatted headers, text, images, bibTeX references, markdown tables, and even LaTeX markup), as well as code chunks (which execute code). The syntax associated with markdown is also outside the scope of this paper, but there are many free and publicly available resources to help with writing markdown syntax (Quarto – Markdown Basics). Markdown and code can be written in "Source" (i.e., raw markdown) or "Visual" (i.e., natural text) view modes within RStudio®. An example of each view is shown in Figure 6 (Source) and Figure 7 (Visual).

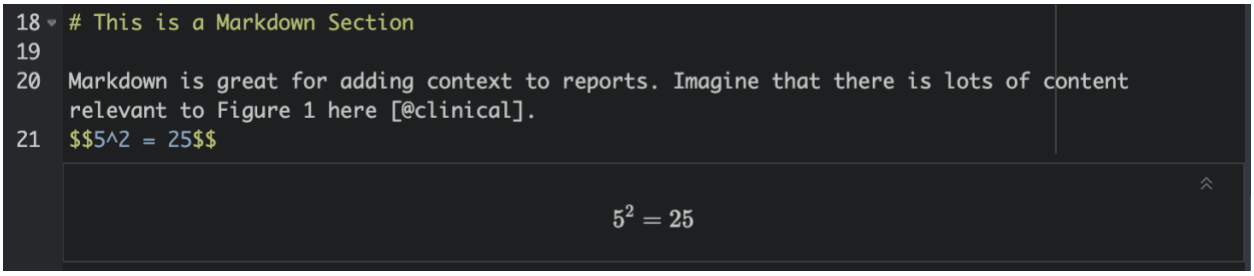

Figure 6. An example of the "Source" view mode within RStudio®. The source view offers a traditional raw markdown view.

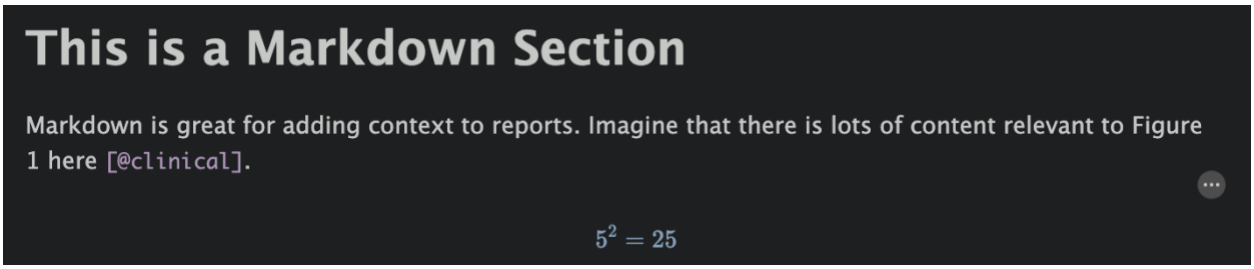

Figure 7. An example of the "Visual" view mode within RStudio®. The visual view offers a modern natural text view.

Code chunks are delineated with three dashes backticks (```) and a coding language {r}. Languages available include:

- **R.**
- **Python.**
- **Bash.**
- **Rcpp.**
- **SQL.**
- **D3.**
- **Stan.**
- **Julia.**

Each language must be installed on the local machine, and some languages such as Python can utilize tools like `reticulate` for added interoperability with R ("Interface to Python" n.d.). Importantly, a mix of programming languages can be called in a single Quarto document (given that they are all installed), thus enabling programmers from diverse backgrounds to collaborate on the same project. In the same notion, the strengths of each language (ex: R for statistical analysis, Python for machine learning) can be utilized to maximize the efficiency and versatility of the data scientist.

Code chunks can utilize a variety of **code chunk options** with similar syntax as YAML, identified by "#|" at the beginning of each line. Let's take the setup code from earlier and build it into our first code chunk:

```
{r}
#| label: Setup
#| echo: false
#| warning: false

packages <- c("tidyverse", "gt", "quarto")

if (any(!sapply(packages, requireNamespace, quietly = TRUE))) {
```

```
      install.packages(packages[!sapply(packages, requireNamespace, quietly
= TRUE)])
    }

    library(tidyverse)
    library(gt)
    library(quarto)

    clinical_trials_fl <- readRDS("query_results_clean_final.rds")
```

In this example, there are three code chunk options being called 1) assignment of a label to the chunk for cleaner coding, 2) echo is disabled so any output generated is hidden from the rendered report, and 3) any warnings are also being hidden from the report. These are being called to hide the package install and library loading outputs.

For the first *content* code chunk, let's summarize our clinical trial data for trial counts per year and make a line chart to visualize the data:

```
    {r}
    #| label: Figure_1
    #| fig-cap: "Figure 1. Successfully Completed Clinical Trials in
Florida from 2014-2023"

    # Counts by Year
    clinical_trials_fl_yearly_summary <- clinical_trials_fl %>%
      mutate(year = year(completion_date)) %>%
      group_by(year) %>%
      summarise(count = n_distinct(nct_id)) %>%
      filter(year >= 2014 & year <= 2023)

    # Line chart
    ggplot(clinical_trials_fl_yearly_summary, aes(x = year, y = count)) +
      geom_line(color = "red", size = 1) +
      geom_point(color = "red", size = 2) +
      scale_x_continuous(breaks = 2014:2023) +
      labs(x = "Year",
           y = "Successfully Completed Clinical Trials") +
      theme_minimal()
```

Here, a label is once again assigned, but we are also including a code chunk option to provide a caption to the output from the code chunk, which we refer to as Figure_1. The markdown text as well as the rendered output from this code chunk is shown in Figure 8.

This is a Markdown Section

Markdown is great for adding context to reports. Imagine that there is lots of content relevant to Figure 1 here (“Clinical trials transformation initiative” n.d.).

```
# Counts by Year
clinical_trials_fl_yearly_summary <- clinical_trials_fl %>%
  mutate(year = year(completion_date)) %>%
  group_by(year) %>%
  summarise(count = n_distinct(nct_id)) %>%
  filter(year >= 2014 & year <= 2023)

# Line chart
ggplot(clinical_trials_fl_yearly_summary, aes(x = year, y = count)) +
  geom_line(color = "red", size = 1) +
  geom_point(color = "red", size = 2) +
  scale_x_continuous(breaks = 2014:2023) +
  labs(x = "Year",
       y = "Successfully Completed Clinical Trials") +
  theme_minimal()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
```

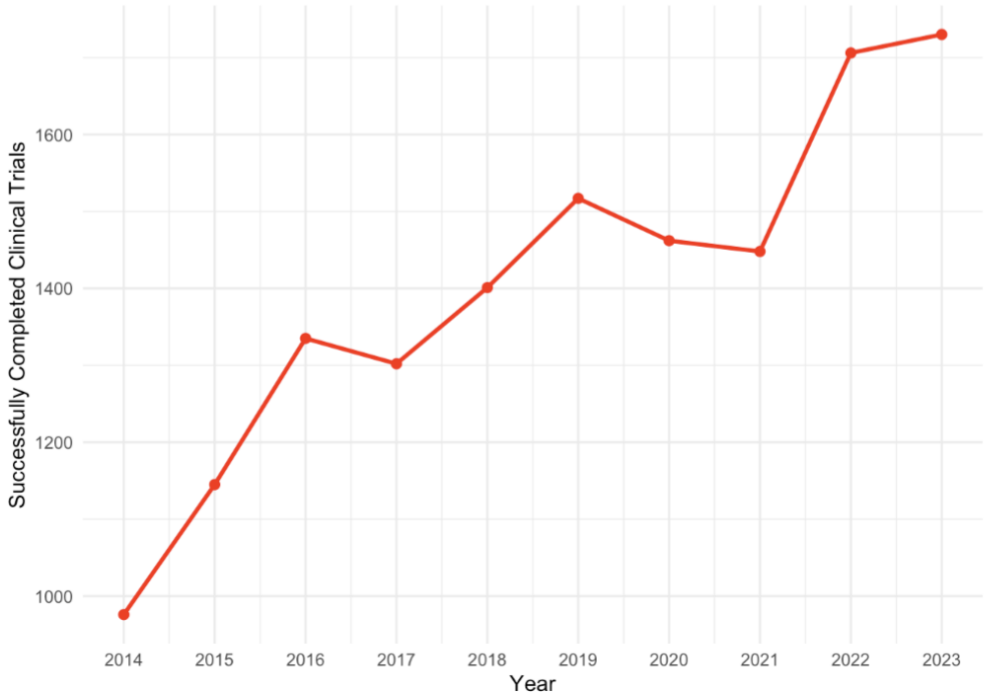

Figure 1. Successfully Completed Clinical Trials in Florida from 2014-2023

Figure 8. The markdown text as well as the rendered output from our example code chunk.

QUARTO PROJECT FORMATS

To change the quarto project *formats*, the only setting that must be changed is the "format" field of the YAML header. In our previous example, HTML was the format. Let's change that to PDF and re-render the document:

```
---
title: |
  ![](images/logo.png){fig-align="left" width="273"}  \
    \
    PharmaSUG 2024: Introduction to Quarto
subtitle: "A Versatile Open-source Tool for Data Reporting and
Visualization "
date: "today"
author: "Joshua J. Cook, M.S., ACRP-PM, CCRC"
abstract: "In the collaborative landscape of data analysis, a common
frustration among analysts stems from the need to integrate and
harmonize different programming languages within a team. Teams often
comprise interdisciplinary researchers, each with their unique
programming preferences and expertise, leading to complexities in
project integration and continuity. The difficulty in compiling and
executing data projects cohesively can hinder efficiency and impede the
delivery of coherent, multi-faceted data insights. This challenge
necessitates a solution that can bridge the gaps between varying coding
languages and methodologies to streamline team collaboration and
project completion. Addressing this issue, the point of this paper is
to present Quarto as an innovative solution that can unify the diverse
programming approaches within a team. Quarto stands out by offering
extensive cross-language support, enabling the integration of code from
multiple languages into a singular, dynamic report. This versatile
reporting system is tailored for the pharmaceutical and biotech
industries, facilitating the creation of comprehensive reports and
visualizations that cater to stakeholders at all technical levels. With
Quarto, consolidating code, narrative text, and outputs into one
document is seamless, accommodating outputs in various formats such as
HTML, PDF, Word, Typeset, Markdown, PowerPoint, dashboards, websites,
manuscripts, and even entire books. This paper serves as an
introduction to Quarto's capabilities, highlighting its role in
enhancing collaboration and efficiency in data science projects across
the spectrum of technical expertise."
theme: minty
toc: true
bibliography: references.bib
csl: asa.csl
```

```
format: pdf
embed-resources: true
---
```

The output format is now in PDF, as shown in Figure 9. Importantly, **not all YAML fields are compatible between formats.** In this example, the image in the title had to be removed to correctly render the PDF.

PharmaSUG 2024: Introduction to Quarto

A Versatile Open-source Tool for Data Reporting and Visualization

Joshua J. Cook, M.S., ACRP-PM, CCRC

2024-03-29

In the collaborative landscape of data analysis, a common frustration among analysts stems from the need to integrate and harmonize different programming languages within a team. Teams often comprise interdisciplinary researchers, each with their unique programming preferences and expertise, leading to complexities in project integration and continuity. The difficulty in compiling and executing data projects cohesively can hinder efficiency and impede the delivery of coherent, multi-faceted data insights. This challenge necessitates a solution that can bridge the gaps between varying coding languages and methodologies to streamline team collaboration and project completion. Addressing this issue, the point of this paper is to present Quarto as an innovative solution that can unify the diverse programming approaches within a team. Quarto stands out by offering extensive cross-language support, enabling the integration of code from multiple languages into a singular, dynamic report. This versatile reporting system is tailored for the pharmaceutical and biotech industries, facilitating the creation of comprehensive reports and visualizations that cater to stakeholders at all technical levels. With Quarto, consolidating code, narrative text, and outputs into one document is seamless, accommodating outputs in various formats such as HTML, PDF, Word, Typeset, Markdown, PowerPoint, dashboards, websites, manuscripts, and even entire books. This paper serves as an introduction to Quarto's capabilities, highlighting its role in enhancing collaboration and efficiency in data science projects across the spectrum of technical expertise.

Table of contents

Figure 9. The PDF output of the rendered document. Importantly, not all YAML fields are compatible between formats. In this example, the image in the title had to be removed to correctly render the PDF.

One of the highlights of Quarto is its ability to generate multiple quarto project formats simultaneously (given that all YAML fields are compatible). In this last example, both HTML and PDF are being generated simultaneously from the same single source (*.qmd*):

```
    ---
    title: "PharmaSUG 2024: Introduction to Quarto"
    subtitle: "A Versatile Open-source Tool for Data Reporting and
Visualization "
    date: "today"
    author: "Joshua J. Cook, M.S., ACRP-PM, CCRC"
    abstract: "In the collaborative landscape of data analysis, a common
frustration among analysts stems from the need to integrate and harmonize
different programming languages within a team. Teams often comprise
interdisciplinary researchers, each with their unique programming
preferences and expertise, leading to complexities in project integration
and continuity. The difficulty in compiling and executing data projects
cohesively can hinder efficiency and impede the delivery of coherent,
multi-faceted data insights. This challenge necessitates a solution that
can bridge the gaps between varying coding languages and methodologies to
streamline team collaboration and project completion. Addressing this
issue, the point of this paper is to present Quarto as an innovative
solution that can unify the diverse programming approaches within a team.
Quarto stands out by offering extensive cross-language support, enabling
the integration of code from multiple languages into a singular, dynamic
report. This versatile reporting system is tailored for the pharmaceutical
and biotech industries, facilitating the creation of comprehensive reports
and visualizations that cater to stakeholders at all technical levels. With
Quarto, consolidating code, narrative text, and outputs into one document
is seamless, accommodating outputs in various formats such as HTML, PDF,
Word, Typeset, Markdown, PowerPoint, dashboards, websites, manuscripts, and
even entire books. This paper serves as an introduction to Quarto's
capabilities, highlighting its role in enhancing collaboration and
efficiency in data science projects across the spectrum of technical
expertise."
    theme: minty
    toc: true
    bibliography: references.bib
    csl: asa.csl
    format:
      html:
        toc: true
      pdf:
        toc: true
    embed-resources: true
```

COLLABORATION AND OTHER QUARTO PROJECT TYPES

COLLABORATION WITH GIT AND GITHUB

Git is a distributed version control system (**VCS**) that is commonly used for tracking changes in files and for collaboration during code development ("About Git" n.d.). **GitHub** is a website and service that hosts Git repositories (i.e., hubs) where developers can store, share, and collaborate on projects ("GitHub features" n.d.). Git integration is built-in to RStudio® (as shown earlier in Figure 3) and is a key medium through which Quarto can be used *collaboratively*. Users can sync the same Quarto Project, along with any file dependencies, across systems to collaborate on projects. The full capabilities of Git and GitHub are outside the scope of this paper (see Recommended Reading), but a simplified workflow for collaboration with Quarto is provided in Figure 10.

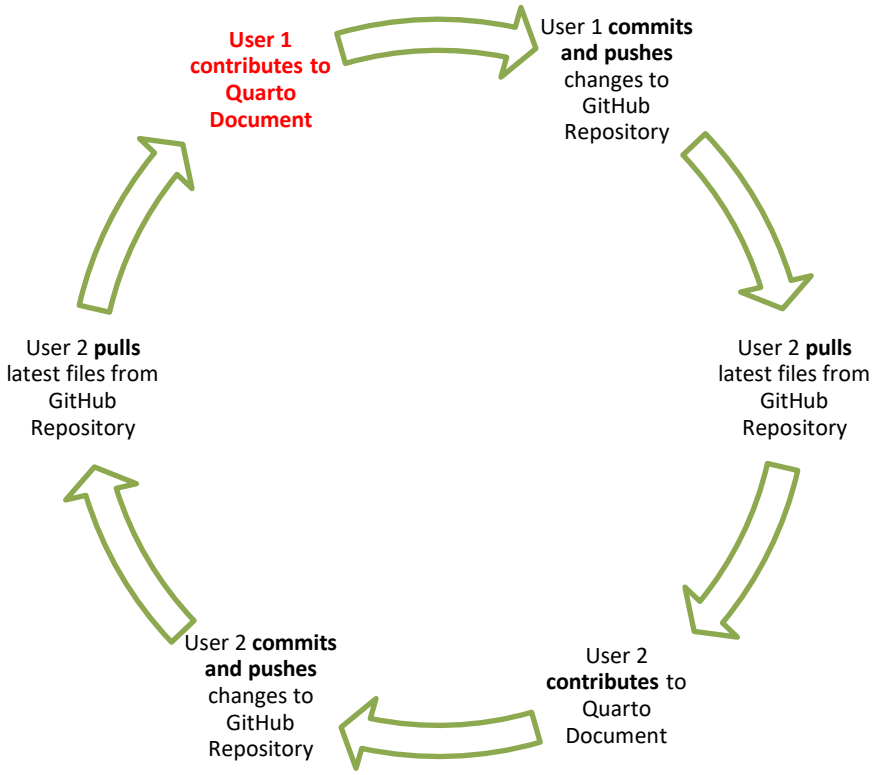

Figure 10. A simplified workflow for collaboration with Quarto via GitHub.

OTHER QUARTO PROJECT TYPES

This paper provided an overview of Quarto Projects, which take the form of a report (in this case HTML and PDF quarto project formats). However, there are many other quarto project formats that those reports can be output to (ex: DOCX or typst). Furthermore, Quarto supports many other project types that are built using the same system and similar syntax as reports:

- **Presentations** (*including Revealjs, PowerPoint, and Beamer*).
- **Dashboards**.
- **Websites**.
- **Books**.
- **Manuscripts**.

A separate paper could be written on each of these Quarto project types, and thus exploration of each is outside the scope of this paper (see Recommended Reading). However, it is important for new Quarto users to be aware that these project types exist so they can consider project efficiency during their work. For example, if Quarto Projects are setup correctly, one could write a presentation on their topic simultaneously with the writing of the analytical report and draft manuscript. Systems such as these would offer a new approach to efficiency in data science, offering expedited analytical reporting and contextual dissemination in multiple output types and formats all from a single source document.

CONCLUSION

Quarto is an innovative solution that has the extraordinary potential to unify the diverse programming approaches within a team. Quarto stands out by offering extensive cross-language support, enabling the integration of code from multiple languages into a singular, dynamic report (along with many other project types). This versatile reporting system is tailored for the pharmaceutical and biotech industries, facilitating

the creation of comprehensive reports and visualizations that cater to stakeholders at all technical levels. This paper served as an introduction to Quarto's capabilities, highlighting its role in enhancing collaboration and efficiency in data science projects across the spectrum of technical expertise.

REFERENCES

"About Git" (n.d.). *GitHub Docs*, Available athttps://docs.github.com/en/get-started/using-git/about-git.

Blei, D. M., and Smyth, P. (2017), "Science and data science," *Proceedings of the National Academy of Sciences of the United States of America*, 114, 8689–8692. https://doi.org/10.1073/pnas.1702076114.

"Clinical Trials Transformation Initiative" (n.d.). *CTTI*, Available athttps://ctti-clinicaltrials.org/.

Davenport, T. H., and Patil, D. J. (2012), "Data scientist: the sexiest job of the 21st century," *Harvard Business Review*, 90, 70–76, 128.

"GitHub features: the right tools for the job" (n.d.). *GitHub*, Available athttps://github.com/features.
"Interface to Python" (n.d.). Available athttps://rstudio.github.io/reticulate/.

Knaflic, C. N. (2015), *Storytelling with Data: A Data Visualization Guide for Business Professionals*, John Wiley & Sons.

VanderPlas, J. (2016), *Python Data Science Handbook: Essential Tools for Working with Data*, O'Reilly Media, Inc.

"What is Data Science? | IBM" (n.d.). Available athttps://www.ibm.com/topics/data-science.

Wickham, H., and Grolemund, G. (2016), *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*, O'Reilly Media, Inc.

ACKNOWLEDGMENTS

The author thanks PharmaSUG for their support through the PharmaSUG New Professionals Scholarship Program, and his mentors, Kirk Paul Lafler (@sasNerd), and Dr. Achraf Cohen, for their ongoing mentorship and support. The author also thanks Richann Watson, Louise Hadden, and Troy Martin Hughes, for their general guidance and mentorship.

RECOMMENDED READING

- Get started with Quarto | Mine Çetinkaya-Rundel (youtube.com)
- RStudio Desktop - Posit
- Quarto – Get Started
- Quarto – Reference
- Quarto – Markdown Basics
- citation-style-language/styles: Official repository for Citation Style Language (CSL) citation styles.

(github.com)

- Zotero Style Repository
- Interface to Python • reticulate (rstudio.github.io)
- Features | GitHub
- skills/introduction-to-github: Get started using GitHub in less than an hour.
- Quarto – Guide

CONTACT INFORMATION

Joshua J. Cook, M.S., ACRP-PM, CCRC is a recent graduate of Wake Forest University (WFU) where he earned his Master of Science for Clinical Research Management. He is a current graduate student at the University of West Florida (UWF) studying Data Science while working at the university as an Adjunct Professor. Joshua worked in the field of clinical research for nearly three years, starting in neurology clinical trials and then specializing in orthopedic regenerative medicine as a Research Quality Analyst. He has published his undergraduate honors thesis, entitled “Endurance exercise-mediated metabolic reshuffle attenuates high-caloric diet-induced non-alcoholic fatty liver disease” in the *Annals of Hepatology* and has recently submitted several orthopedic research papers to various journals. He has also presented his research at over ten unique conferences at the local, state, and national levels with topics spanning from the impact of blood sugar on Alzheimer’s Disease to publication metric tracking with R and Microsoft Power BI®. Joshua has developed a passion for bench-to-bedside research and aims to synthesize his knowledge of the biomedical sciences, clinical research, and data science to become a physician-scientist capable of integrating clinical care with clinical research in a way that maximizes evidence-based care options for his patients.

Your comments and questions are valued and encouraged. Contact the author at:

Joshua J. Cook, M.S., ACRP-PM, CCRC

Adjunct Professor / Graduate Student / Clinical Researcher / Data Scientist

Website: https://joshua-j-cook-portfolios.netlify.app/

Email: jcook0312@outlook.com
