Course Name

August, 2025

Table of Contents

# About this Course

Check out the [ITN website](https://www.itcrtraining.org/resources) for more ITN resources.

## 0.1 Available course formats

This course is available in multiple formats which allows you to take it in the way that best suites your needs. You can take it for certificate which can be for free or fee.

* The material for this course can be viewed without login requirement on this [Bookdown website](https://hutchdatascience.org/reproducibility_capstone/). This format might be most appropriate for you if you rely on screen-reader technology.
* This course can be taken on [Coursera for certification for a fee here](https://www.coursera.org/learn/capstone-reproducible-science) (this course can also be audited for free on Coursera).
* Our courses are open source, you can find the [source material for this course on GitHub](https://github.com/fhdsl/reproducibility_capstone).

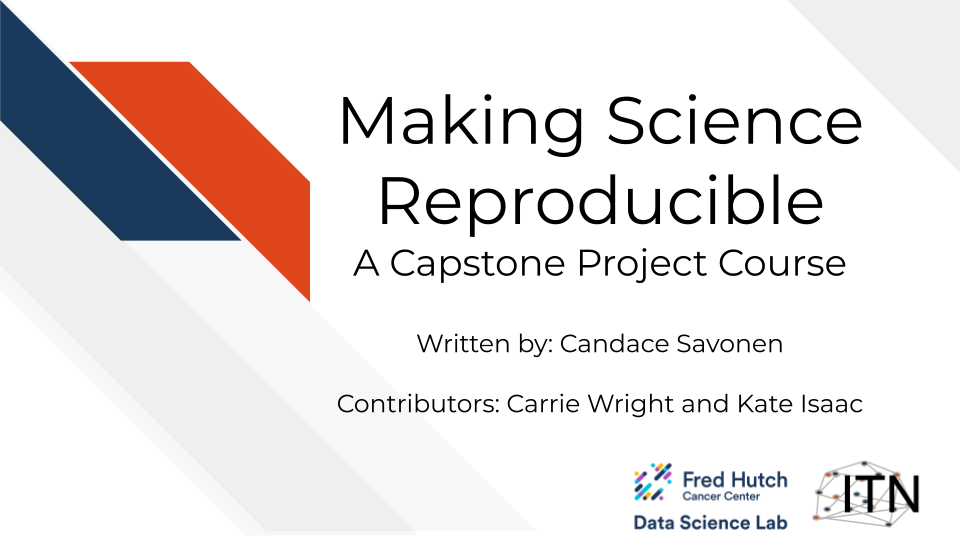
# 1 Introduction Test

[Here is a link at the very beginning to test](https://github.com/). And need to make some edits to retrigger rendering of course.

## 1.1 Motivation

Cancer research in the 21st century is increasingly data-driven, with researchers generating and analyzing vast amounts of genomic, proteomic, imaging, and clinical data. The ability to effectively manage, analyze, and share these complex datasets has become as crucial as traditional laboratory skills. However, many cancer researchers find themselves working with computational tools and workflows that were not developed using robust software engineering practices. These tools often suffer from poor documentation, lack of version control, inadequate testing, and limited interoperability. Such limitations hinder collaboration, impede scalability, and make it difficult to reproduce results reliably.

The modern cancer informatics landscape demands not just analytical skills, but also proficiency in reproducible computational practices. Research has shown that computational analyses are often difficult or impossible to reproduce without direct contact with the original researchers ([**BeaulieuJones2017?**](#ref-BeaulieuJones2017)). This reproducibility crisis extends beyond academic concerns—it directly impacts the translation of research findings into clinical applications and the overall efficiency of cancer research efforts.



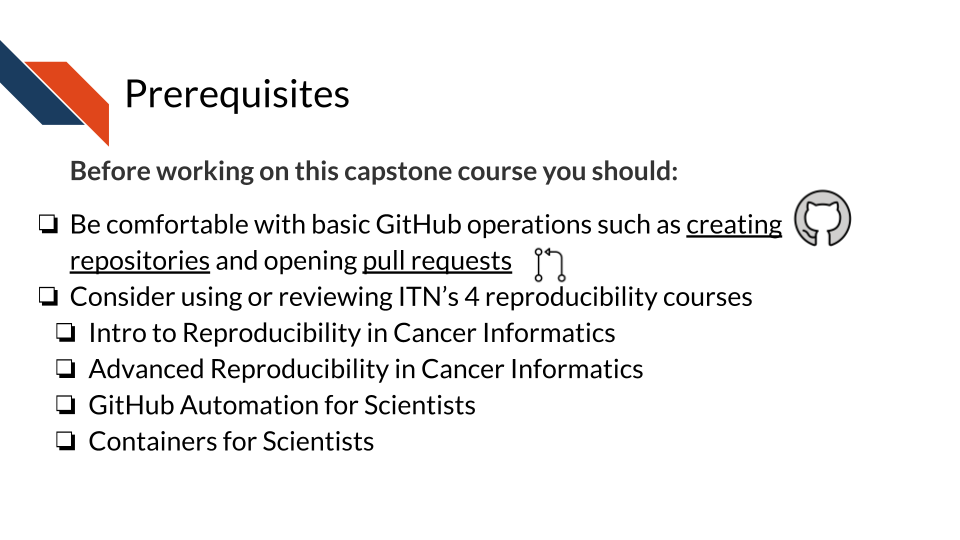
## 1.2 Target Audience

The course is intended for cancer researchers, bioinformaticians, and data scientists who work with computational analyses in cancer research contexts and want to enhance the reproducibility, efficiency, and shareability of their work.

This capstone course builds on foundational concepts from the following ITCR Training Network’s courses:

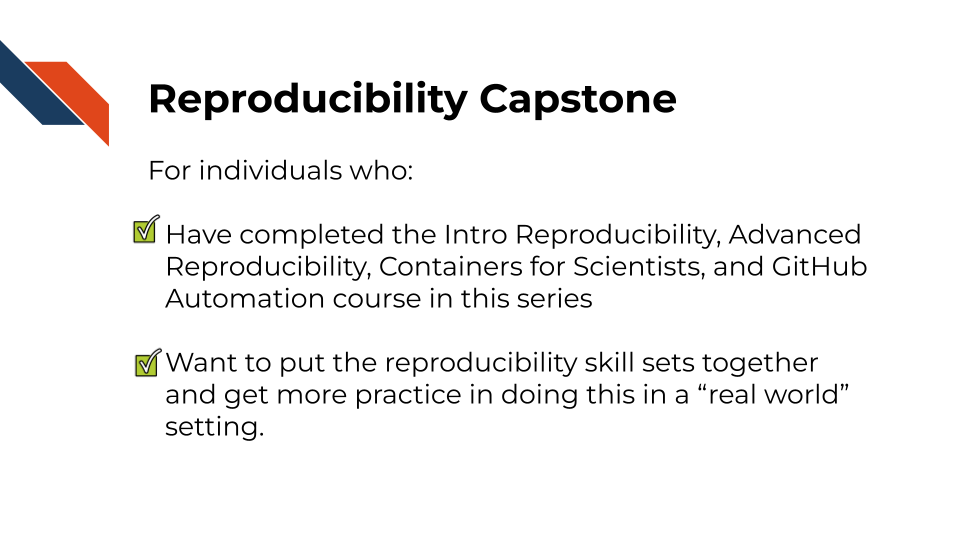
* [Intro to Reproducibility in Cancer Informatics](https://hutchdatascience.org/ITN_course_search/Reproducibility_in_Cancer_Informatics_coursePage.html)
* [Advanced Reproducibility in Cancer Informatics](https://hutchdatascience.org/ITN_course_search/Adv_Reproducibility_in_Cancer_Informatics_coursePage.html)
* [Containers for Scientists](https://hutchdatascience.org/ITN_course_search/Containers_for_Scientists_coursePage.html)
* [GitHub Automation for Scientists](https://hutchdatascience.org/ITN_course_search/GitHub_Automation_for_Scientists_coursePage.html)

Students should be comfortable with basic GitHub operations, including creating repositories, making commits, and opening pull requests before beginning this capstone.



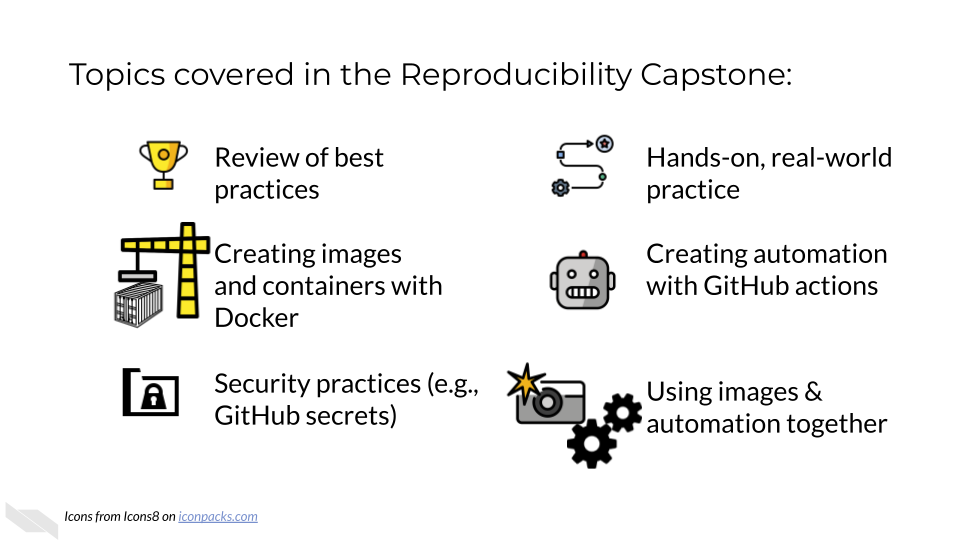
*This capstone course is written for individuals who:*

* Conduct computational analyses for cancer research, including genomics, proteomics, imaging, or clinical data analysis.
* Have completed the ITN reproducibility courses mentioned above.
* Want to put the reproducibility skill sets together and get more practice doing this in a “real world” setting.

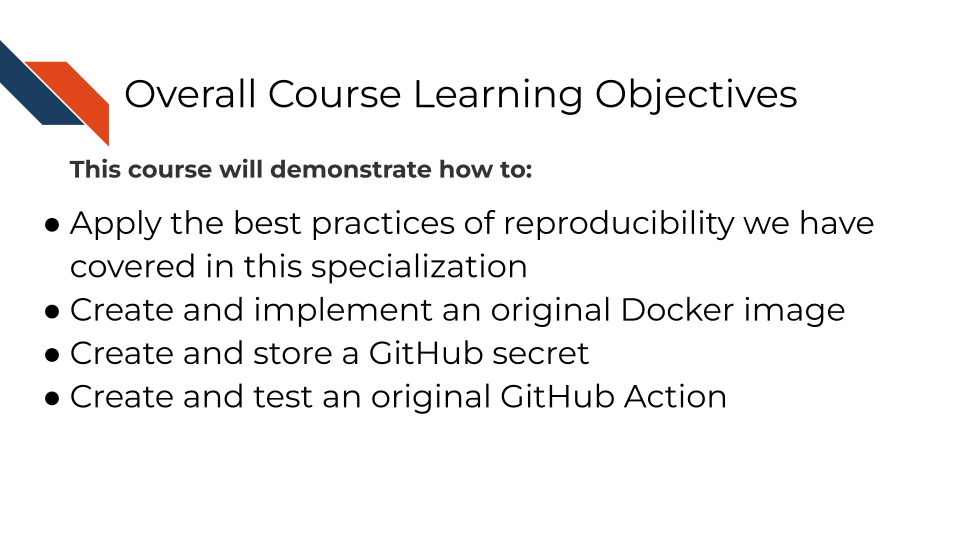


## 1.3 Curriculum

The course covers practical application of containerization and automation technologies specifically tailored for cancer informatics workflows. Unlike introductory courses that focus on individual concepts, this capstone emphasizes integration and **real-world application** of multiple technologies working together.



## 1.4 Objectives



Connecting the Learning Objectives to the Prerequisite courses

The [Intro to](https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/no_toc/) and [Advanced Reproducibility in Cancer Informatics](https://jhudatascience.org/Adv_Reproducibility_in_Cancer_Informatics/no_toc/) courses both provide an overview of **best practices** such as

* Utilizing [organization in your project](https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/no_toc/organizing-your-project.html)
* Composing [exploratory analyses or reports in Notebooks](https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/no_toc/using-notebooks.html)
* [Documenting analyses and software](https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/no_toc/documenting-analyses.html),
* Using [version control and branches](https://jhudatascience.org/Adv_Reproducibility_in_Cancer_Informatics/no_toc/using-version-control-with-github.html) for code
* [Data sharing (when appropriate)](https://jhudatascience.org/Adv_Reproducibility_in_Cancer_Informatics/no_toc/providing-data.html)
* [Requesting code review](https://jhudatascience.org/Adv_Reproducibility_in_Cancer_Informatics/no_toc/engaging-in-code-review---as-an-author.html)
* [Using automation](https://jhudatascience.org/Adv_Reproducibility_in_Cancer_Informatics/no_toc/automation-as-a-reproducibility-tool.html).
* [Software versioning](https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/no_toc/managing-package-versions.html)
* [Using containers for managing software](https://jhudatascience.org/Adv_Reproducibility_in_Cancer_Informatics/no_toc/launching-a-docker-image.html)

While the intro and advanced reproducibility courses provide overviews about the more advanced topics like automation and containers, the [GitHub Automation for Scientists](https://hutchdatascience.org/GitHub_Automation_for_Scientists/no_toc/index.html) and [Containers for Scientists](https://hutchdatascience.org/Containers_for_Scientists/no_toc/) courses explore these topics, respectively, in more depth. Automation can be used as a continuous integration, continuous deployment tool such that automatic tests are set up to run whenever code is updated. Other automations can occur on a schedule or due to other events happening (like opening a pull request on GitHub).

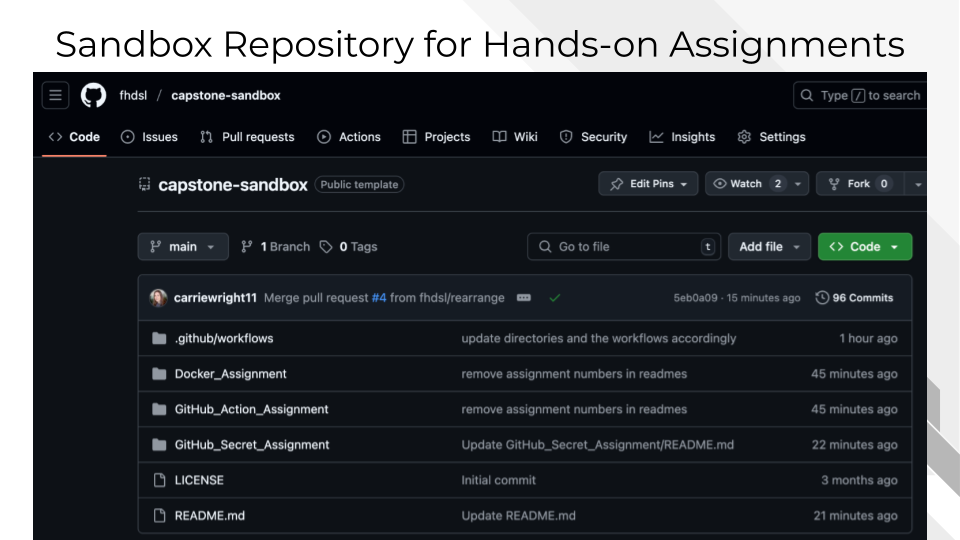
The “create and implement an original **Docker image**” learning objective is putting into practice the theory from the [Containers for Scientists course](https://hutchdatascience.org/Containers_for_Scientists/no_toc/). Containers are used to package software with defined versions together so that a computing environment can be consistently and clearly defined as well as easily shared or used by many people.

Context on what a **GitHub Secret** is, why you would use secrets, and instructions on creating and storing a GitHub Secret are [covered briefly in the GitHub Automation for Scientists course](https://hutchdatascience.org/GitHub_Automation_for_Scientists/github-action-variables.html#setting-and-grabbing-secrets), but additional context is provided at the beginning of the GitHub Secrets exercise within this capstone.

The “create and test an original **GitHub Action**” learning objective is putting into practice the theory from the [GitHub Automation for Scientists course](https://hutchdatascience.org/GitHub_Automation_for_Scientists/no_toc/index.html). In addition, depending on the automation that is created, learners may need to incorporate use of a GitHub Secret and a Docker image.

**Assessment Approach:**

The capstone uses a practical, project-based assessment model where students work with a provided computational sandbox environment to complete real-world tasks. Rather than traditional exams, students demonstrate mastery by successfully implementing containerized workflows and automated pipelines that pass automated evaluation criteria.



**Custom Container Development Assignment** - Students create Docker containers to support specific cancer analysis requirements, demonstrating understanding of containerization principles and bioinformatics software management.

**Working with GitHub Secrets Assignment** – Students create a personal access token, store the token as a GitHub secret in a repository and access that secret in a GitHub Actions workflow, displaying application of best practices in security when handling sensitive information.

**Automated Workflow Implementation Assignment** - Students create GitHub Actions that automatically execute and validate cancer data analysis pipelines, showing proficiency in continuous integration concepts and workflow automation.

Each assignment utilizes automated evaluation systems that provide immediate feedback and validation codes upon successful completion, simulating real-world development environments where code must pass automated testing before deployment.

**Goal of this course:** Enable cancer researchers to create, maintain, and share reproducible computational workflows that meet the evolving standards of modern cancer informatics while enhancing collaboration and accelerating scientific discovery.

**What is not the goal:** This course does not aim to teach cancer biology, statistical analysis methods, or introductory programming concepts. Students are expected to bring domain expertise and basic computational skills to the course, focusing instead on learning to apply advanced reproducibility technologies to their existing research workflows.

## 1.5 How to use the course

This capstone course is designed as a hands-on, practical experience that builds directly on the foundational knowledge from previous ITCR Training Network courses. Students should approach the course with an active learning mindset, ready to experiment, troubleshoot, and iterate on their solutions.

**Working with the Course Sandbox:** The course utilizes a GitHub-based sandbox environment that simulates real-world research collaboration workflows. Students will work with the template repository, create branches for different assignments, and receive automated feedback through pull request evaluations. This approach mirrors modern software development practices and prepares students for collaborative research environments.

**Integration Focus:** Each assignment builds on previous work and integrates multiple technologies. Students should be prepared to synthesize knowledge from containerization, automation, version control, and domain-specific cancer informatics concepts. The goal is to create holistic solutions rather than demonstrate isolated technical skills.

**Real-world Application:** Throughout the course, students are encouraged to consider how the techniques they’re learning apply to their own research contexts. The assignments use cancer informatics examples, but the principles and technologies transfer to a wide range of computational research domains.

We also recommend that students leverage the [broader ITCR Training Network community and resources](https://www.itcrtraining.org/) to enhance their learning experience and build lasting professional connections in the cancer informatics community.

## 1.6 Where do you need to invest?

This course aims to help you apply what we have learned in the previous courses to a real world application. What you build in this capstone is completely up to you, but for your own benefit we suggest you try to make something you will use in your work beyond this specialization.

Think about work on GitHub that you are doing manually, or a project where you are struggling to easily reproduce the results. Those may be ripe opportunities for you to build something. However, keep in mind that it’s OK to start small and build up to these goals as you learn.

Use a few minutes for introspective reflection about where you are spending a lot of time and effort to reproduce something that could use some reproducibility investment (like a Github Action or Docker image). Consider attempting to develop such reproducibility investments for these projects in your upcoming assignments.

# 2 Introduction

## 2.1 Motivation

## 2.2 Target Audience

The course is intended for …

## 2.3 Curriculum

The course covers…

devtools::session\_info()

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value  
## version R version 4.3.2 (2023-10-31)  
## os Ubuntu 22.04.4 LTS  
## system x86\_64, linux-gnu  
## ui X11  
## language (EN)  
## collate en\_US.UTF-8  
## ctype en\_US.UTF-8  
## tz Etc/UTC  
## date 2025-08-07  
## pandoc 3.1.1 @ /usr/local/bin/ (via rmarkdown)  
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date (UTC) lib source  
## bookdown 0.43 2025-04-15 [1] CRAN (R 4.3.2)  
## cachem 1.0.8 2023-05-01 [1] RSPM (R 4.3.0)  
## cli 3.6.2 2023-12-11 [1] RSPM (R 4.3.0)  
## devtools 2.4.5 2022-10-11 [1] RSPM (R 4.3.0)  
## digest 0.6.34 2024-01-11 [1] RSPM (R 4.3.0)  
## ellipsis 0.3.2 2021-04-29 [1] RSPM (R 4.3.0)  
## evaluate 1.0.4 2025-06-18 [1] CRAN (R 4.3.2)  
## fastmap 1.1.1 2023-02-24 [1] RSPM (R 4.3.0)  
## fs 1.6.3 2023-07-20 [1] RSPM (R 4.3.0)  
## glue 1.7.0 2024-01-09 [1] RSPM (R 4.3.0)  
## htmltools 0.5.7 2023-11-03 [1] RSPM (R 4.3.0)  
## htmlwidgets 1.6.4 2023-12-06 [1] RSPM (R 4.3.0)  
## httpuv 1.6.14 2024-01-26 [1] RSPM (R 4.3.0)  
## knitr 1.50 2025-03-16 [1] CRAN (R 4.3.2)  
## later 1.3.2 2023-12-06 [1] RSPM (R 4.3.0)  
## lifecycle 1.0.4 2023-11-07 [1] RSPM (R 4.3.0)  
## magrittr 2.0.3 2022-03-30 [1] RSPM (R 4.3.0)  
## memoise 2.0.1 2021-11-26 [1] RSPM (R 4.3.0)  
## mime 0.12 2021-09-28 [1] RSPM (R 4.3.0)  
## miniUI 0.1.1.1 2018-05-18 [1] RSPM (R 4.3.0)  
## pkgbuild 1.4.3 2023-12-10 [1] RSPM (R 4.3.0)  
## pkgload 1.3.4 2024-01-16 [1] RSPM (R 4.3.0)  
## profvis 0.3.8 2023-05-02 [1] RSPM (R 4.3.0)  
## promises 1.2.1 2023-08-10 [1] RSPM (R 4.3.0)  
## purrr 1.0.2 2023-08-10 [1] RSPM (R 4.3.0)  
## R6 2.5.1 2021-08-19 [1] RSPM (R 4.3.0)  
## Rcpp 1.0.12 2024-01-09 [1] RSPM (R 4.3.0)  
## remotes 2.5.0 2024-03-17 [1] CRAN (R 4.3.2)  
## rlang 1.1.6 2025-04-11 [1] CRAN (R 4.3.2)  
## rmarkdown 2.25 2023-09-18 [1] RSPM (R 4.3.0)  
## sessioninfo 1.2.2 2021-12-06 [1] RSPM (R 4.3.0)  
## shiny 1.8.0 2023-11-17 [1] RSPM (R 4.3.0)  
## stringi 1.8.3 2023-12-11 [1] RSPM (R 4.3.0)  
## stringr 1.5.1 2023-11-14 [1] RSPM (R 4.3.0)  
## urlchecker 1.0.1 2021-11-30 [1] RSPM (R 4.3.0)  
## usethis 2.2.3 2024-02-19 [1] RSPM (R 4.3.0)  
## vctrs 0.6.5 2023-12-01 [1] RSPM (R 4.3.0)  
## xfun 0.52 2025-04-02 [1] CRAN (R 4.3.2)  
## xtable 1.8-4 2019-04-21 [1] RSPM (R 4.3.0)  
## yaml 2.3.10 2024-07-26 [1] CRAN (R 4.3.2)  
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library  
##   
## ──────────────────────────────────────────────────────────────────────────────

# 3 A new chapter

If you haven’t yet read the getting started documentation pages; [start there](https://www.ottrproject.org/getting_started.html).

To see the rendered version of this chapter and the rest of the template, see here: <https://ottrproject.org/OTTR_Template/>.

## 3.1 Learning Objectives

Every chapter also needs Learning objectives that will look like this:

This chapter will cover:

* {You can use <https://tips.uark.edu/using-blooms-taxonomy/> to define some learning objectives here}
* {Another learning objective}

## 3.2 Libraries

For this chapter, we’ll need the following packages attached:

\*Remember to add [any additional packages you need to your course’s own docker image](https://github.com/ottrproject/OTTR_Template/wiki/Using-Docker#starting-a-new-docker-image).

library(magrittr)

## 3.3 Topic of Section

You can write all your text in sections like this, using ## to indicate a new header. you can use additional pound symbols to create lower levels of headers.

See [here](https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf) for additional general information about how you can format text within R Markdown files. In addition, see [here](https://pandoc.org/MANUAL.html#pandocs-markdown) for more in depth and advanced options.

### 3.3.1 Subtopic

Here’s a subheading (using three pound symbols) and some text in this subsection!

## 3.4 Code examples

You can demonstrate code like this:

output\_dir <- file.path("resources", "code\_output")  
if (!dir.exists(output\_dir)) {  
 dir.create(output\_dir)  
}

And make plots too:

hist\_plot <- hist(iris$Sepal.Length)



You can also save these plots to file:

png(file.path(output\_dir, "test\_plot.png"))  
hist\_plot

## $breaks  
## [1] 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0  
##   
## $counts  
## [1] 5 27 27 30 31 18 6 6  
##   
## $density  
## [1] 0.06666667 0.36000000 0.36000000 0.40000000 0.41333333 0.24000000 0.08000000  
## [8] 0.08000000  
##   
## $mids  
## [1] 4.25 4.75 5.25 5.75 6.25 6.75 7.25 7.75  
##   
## $xname  
## [1] "iris$Sepal.Length"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

dev.off()

## png   
## 2

## 3.5 Image example

How to include a Google slide. It’s simplest to use the ottrpal package:



But if you have the slide or some other image locally downloaded you can also use HTML like this:

## 3.6 Video examples

You may also want to embed videos in your course. If alternatively, you just want to include a link you can do so like this:

Check out this [link to a video](https://www.youtube.com/embed/VOCYL-FNbr0) using markdown syntax.

### 3.6.1 Using knitr

To embed videos in your course, you can use knitr::include\_url() like this: Note that you should use echo=FALSE in the code chunk because we don’t want the code part of this to show up. If you are unfamiliar with [how R Markdown code chunks work, read this](https://rmarkdown.rstudio.com/lesson-3.html).

## `google-chrome`, `chromium-browser` and `chrome` were not found. Try setting the `CHROMOTE\_CHROME` environment variable to the executable of a Chromium-based browser, such as Google Chrome, Chromium or Brave or adding one of these executables to your PATH.

### 3.6.2 Using HTML

### 3.6.3 Using knitr

### 3.6.4 Using HTML

## 3.7 Website Examples

Yet again you can use a link to a website like so:

[A Website](https://yihui.org)

You might want to have users open a website in a new tab by default, especially if they need to reference both the course and a resource at once.

[A Website](https://yihui.org)

Or, you can embed some websites.

### 3.7.1 Using knitr

This works:

### 3.7.2 Using HTML

If you’d like the URL to show up in a new tab you can do this:

<a href="https://www.linkedin.com" target="\_blank">LinkedIn</a>

## 3.8 Citation examples

We can put citations at the end of a sentence like this ([Allaire et al. 2021](#ref-rmarkdown2021)). Or multiple citations Xie, Allaire, and Grolemund ([2018](#ref-Xie2018)).

but they need a ; separator ([Allaire et al. 2021](#ref-rmarkdown2021); [Xie, Allaire, and Grolemund 2018](#ref-Xie2018)).

In text, we can put citations like this Allaire et al. ([2021](#ref-rmarkdown2021)).

## 3.9 Stylized boxes

Occasionally, you might find it useful to emphasize a particular piece of information. To help you do so, we have provided css code and images (no need for you to worry about that!) to create the following stylized boxes.

You can use these boxes in your course with either of two options: using HTML code or Pandoc syntax.

### 3.9.1 Using rmarkdown container syntax

The rmarkdown package allows for a different syntax to be converted to the HTML that you just saw and also allows for conversion to LaTeX. See the [Bookdown](https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html) documentation for more information ([Xie, Dervieux, and Riederer 2020](#ref-Xie2020)). Note that Bookdown uses Pandoc.

::: {.notice}  
Note using rmarkdown syntax.  
  
:::

Note using rmarkdown syntax.

As an example you might do something like this:

Please click on the subsection headers in the left hand navigation bar (e.g., 2.1, 4.3) a second time to expand the table of contents and enable the scroll\_highlight feature ([see more](introduction.html#scroll-highlight))

### 3.9.2 Using HTML

To add a warning box like the following use:

<div class = "notice">  
Followed by the text you want inside  
</div>

This will create the following:

Followed by the text you want inside

Here is a <div class = "warning"> box:

Note text

Here is a <div class = "github"> box:

GitHub text

Here is a <div class = "dictionary"> box:

dictionary text

Here is a <div class = "reflection"> box:

reflection text

Here is a <div class = "wip"> box:

Work in Progress text

## 3.10 Dropdown summaries

You can hide additional information in a dropdown menu

Here’s more words that are hidden.

## 3.11 Print out session info

You should print out session info when you have code for [reproducibility purposes](https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/managing-package-versions.html).

devtools::session\_info()

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value  
## version R version 4.3.2 (2023-10-31)  
## os Ubuntu 22.04.4 LTS  
## system x86\_64, linux-gnu  
## ui X11  
## language (EN)  
## collate en\_US.UTF-8  
## ctype en\_US.UTF-8  
## tz Etc/UTC  
## date 2025-08-07  
## pandoc 3.1.1 @ /usr/local/bin/ (via rmarkdown)  
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date (UTC) lib source  
## bookdown 0.43 2025-04-15 [1] CRAN (R 4.3.2)  
## cachem 1.0.8 2023-05-01 [1] RSPM (R 4.3.0)  
## chromote 0.5.1 2025-04-24 [1] CRAN (R 4.3.2)  
## cli 3.6.2 2023-12-11 [1] RSPM (R 4.3.0)  
## devtools 2.4.5 2022-10-11 [1] RSPM (R 4.3.0)  
## digest 0.6.34 2024-01-11 [1] RSPM (R 4.3.0)  
## dplyr 1.1.4 2023-11-17 [1] RSPM (R 4.3.0)  
## ellipsis 0.3.2 2021-04-29 [1] RSPM (R 4.3.0)  
## evaluate 1.0.4 2025-06-18 [1] CRAN (R 4.3.2)  
## fansi 1.0.6 2023-12-08 [1] RSPM (R 4.3.0)  
## fastmap 1.1.1 2023-02-24 [1] RSPM (R 4.3.0)  
## fs 1.6.3 2023-07-20 [1] RSPM (R 4.3.0)  
## generics 0.1.3 2022-07-05 [1] RSPM (R 4.3.0)  
## gitcreds 0.1.2 2022-09-08 [1] RSPM (R 4.3.0)  
## glue 1.7.0 2024-01-09 [1] RSPM (R 4.3.0)  
## hms 1.1.3 2023-03-21 [1] RSPM (R 4.3.0)  
## htmltools 0.5.7 2023-11-03 [1] RSPM (R 4.3.0)  
## htmlwidgets 1.6.4 2023-12-06 [1] RSPM (R 4.3.0)  
## httpuv 1.6.14 2024-01-26 [1] RSPM (R 4.3.0)  
## httr 1.4.7 2023-08-15 [1] RSPM (R 4.3.0)  
## jsonlite 1.8.8 2023-12-04 [1] RSPM (R 4.3.0)  
## knitr 1.50 2025-03-16 [1] CRAN (R 4.3.2)  
## later 1.3.2 2023-12-06 [1] RSPM (R 4.3.0)  
## lifecycle 1.0.4 2023-11-07 [1] RSPM (R 4.3.0)  
## magrittr \* 2.0.3 2022-03-30 [1] RSPM (R 4.3.0)  
## memoise 2.0.1 2021-11-26 [1] RSPM (R 4.3.0)  
## mime 0.12 2021-09-28 [1] RSPM (R 4.3.0)  
## miniUI 0.1.1.1 2018-05-18 [1] RSPM (R 4.3.0)  
## ottrpal 2.0.0 2025-08-04 [1] Github (ottrproject/ottrpal@e4cfa82)  
## pillar 1.9.0 2023-03-22 [1] RSPM (R 4.3.0)  
## pkgbuild 1.4.3 2023-12-10 [1] RSPM (R 4.3.0)  
## pkgconfig 2.0.3 2019-09-22 [1] RSPM (R 4.3.0)  
## pkgload 1.3.4 2024-01-16 [1] RSPM (R 4.3.0)  
## processx 3.8.3 2023-12-10 [1] RSPM (R 4.3.0)  
## profvis 0.3.8 2023-05-02 [1] RSPM (R 4.3.0)  
## promises 1.2.1 2023-08-10 [1] RSPM (R 4.3.0)  
## ps 1.7.6 2024-01-18 [1] RSPM (R 4.3.0)  
## purrr 1.0.2 2023-08-10 [1] RSPM (R 4.3.0)  
## R6 2.5.1 2021-08-19 [1] RSPM (R 4.3.0)  
## Rcpp 1.0.12 2024-01-09 [1] RSPM (R 4.3.0)  
## readr 2.1.5 2024-01-10 [1] RSPM (R 4.3.0)  
## remotes 2.5.0 2024-03-17 [1] CRAN (R 4.3.2)  
## rlang 1.1.6 2025-04-11 [1] CRAN (R 4.3.2)  
## rmarkdown 2.25 2023-09-18 [1] RSPM (R 4.3.0)  
## rprojroot 2.1.0 2025-07-12 [1] CRAN (R 4.3.2)  
## rvest 1.0.4 2024-02-12 [1] CRAN (R 4.3.2)  
## sessioninfo 1.2.2 2021-12-06 [1] RSPM (R 4.3.0)  
## shiny 1.8.0 2023-11-17 [1] RSPM (R 4.3.0)  
## spelling 2.3.1 2024-10-04 [1] CRAN (R 4.3.2)  
## stringi 1.8.3 2023-12-11 [1] RSPM (R 4.3.0)  
## stringr 1.5.1 2023-11-14 [1] RSPM (R 4.3.0)  
## tibble 3.3.0 2025-06-08 [1] CRAN (R 4.3.2)  
## tidyr 1.3.1 2024-01-24 [1] RSPM (R 4.3.0)  
## tidyselect 1.2.0 2022-10-10 [1] RSPM (R 4.3.0)  
## tzdb 0.4.0 2023-05-12 [1] RSPM (R 4.3.0)  
## urlchecker 1.0.1 2021-11-30 [1] RSPM (R 4.3.0)  
## usethis 2.2.3 2024-02-19 [1] RSPM (R 4.3.0)  
## utf8 1.2.4 2023-10-22 [1] RSPM (R 4.3.0)  
## vctrs 0.6.5 2023-12-01 [1] RSPM (R 4.3.0)  
## webshot2 0.1.2 2025-04-23 [1] CRAN (R 4.3.2)  
## websocket 1.4.4 2025-04-10 [1] CRAN (R 4.3.2)  
## xfun 0.52 2025-04-02 [1] CRAN (R 4.3.2)  
## xml2 1.3.6 2023-12-04 [1] RSPM (R 4.3.0)  
## xtable 1.8-4 2019-04-21 [1] RSPM (R 4.3.0)  
## yaml 2.3.10 2024-07-26 [1] CRAN (R 4.3.2)  
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library  
##   
## ──────────────────────────────────────────────────────────────────────────────

# About the Authors

These credits are based on our [course contributors table guidelines](https://www.ottrproject.org/more_features.html#giving-credits-to-contributors).

| Credits | Names |
| --- | --- |
| **Pedagogy** |  |
| Lead Content Instructor(s) | [FirstName LastName](link%20to%20personal%20website) |
| Lecturer(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | Delivered the course in some way - video or audio |
| Content Author(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | If any other authors besides lead instructor |
| Content Contributor(s) (include section name/link in parentheses) - make new line if more than one section involved | Wrote less than a chapter |
| Content Editor(s)/Reviewer(s) | Checked your content |
| Content Director(s) | Helped guide the content direction |
| Content Consultants (include chapter name/link in parentheses or word “General”) - make new line if more than one chapter involved | Gave high level advice on content |
| Acknowledgments | Gave small assistance to content but not to the level of consulting |
| **Production** |  |
| Content Publisher(s) | Helped with publishing platform |
| Content Publishing Reviewer(s) | Reviewed overall content and aesthetics on publishing platform |
| **Technical** |  |
| Course Publishing Engineer(s) | Helped with the code for the technical aspects related to the specific course generation |
| Template Publishing Engineers | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/), [Ava Hoffman](https://www.avahoffman.com/) |
| Publishing Maintenance Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Technical Publishing Stylists | [Carrie Wright](https://carriewright11.github.io/), [Ava Hoffman](https://www.avahoffman.com/), [Candace Savonen](https://www.cansavvy.com/), [Katherine Cox](https://katherinecox.github.io/) |
| Package Developers ([ottrpal](https://github.com/ottrproject/ottrpal)) | [Candace Savonen](https://www.cansavvy.com/), [Ava Hoffman](https://www.avahoffman.com/), [Howard Baek](https://www.linkedin.com/in/howard-baik/), [Kate Isaac](https://kweav.github.io/), [Carrie Wright](https://carriewright11.github.io/), [John Muschelli](https://johnmuschelli.com/) |
| **Art and Design** |  |
| Illustrator(s) | Created graphics for the course |
| Figure Artist(s) | Created figures/plots for course |
| Videographer(s) | Filmed videos |
| Videography Editor(s) | Edited film |
| Audiographer(s) | Recorded audio |
| Audiography Editor(s) | Edited audio recordings |
| **Funding** |  |
| Funder(s) | Institution/individual who funded course including grant number |
| Funding Staff | Staff members who help with funding |

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value  
## version R version 4.3.2 (2023-10-31)  
## os Ubuntu 22.04.4 LTS  
## system x86\_64, linux-gnu  
## ui X11  
## language (EN)  
## collate en\_US.UTF-8  
## ctype en\_US.UTF-8  
## tz Etc/UTC  
## date 2025-08-07  
## pandoc 3.1.1 @ /usr/local/bin/ (via rmarkdown)  
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date (UTC) lib source  
## bookdown 0.43 2025-04-15 [1] CRAN (R 4.3.2)  
## cachem 1.0.8 2023-05-01 [1] RSPM (R 4.3.0)  
## cli 3.6.2 2023-12-11 [1] RSPM (R 4.3.0)  
## devtools 2.4.5 2022-10-11 [1] RSPM (R 4.3.0)  
## digest 0.6.34 2024-01-11 [1] RSPM (R 4.3.0)  
## ellipsis 0.3.2 2021-04-29 [1] RSPM (R 4.3.0)  
## evaluate 1.0.4 2025-06-18 [1] CRAN (R 4.3.2)  
## fastmap 1.1.1 2023-02-24 [1] RSPM (R 4.3.0)  
## fs 1.6.3 2023-07-20 [1] RSPM (R 4.3.0)  
## glue 1.7.0 2024-01-09 [1] RSPM (R 4.3.0)  
## htmltools 0.5.7 2023-11-03 [1] RSPM (R 4.3.0)  
## htmlwidgets 1.6.4 2023-12-06 [1] RSPM (R 4.3.0)  
## httpuv 1.6.14 2024-01-26 [1] RSPM (R 4.3.0)  
## knitr 1.50 2025-03-16 [1] CRAN (R 4.3.2)  
## later 1.3.2 2023-12-06 [1] RSPM (R 4.3.0)  
## lifecycle 1.0.4 2023-11-07 [1] RSPM (R 4.3.0)  
## magrittr 2.0.3 2022-03-30 [1] RSPM (R 4.3.0)  
## memoise 2.0.1 2021-11-26 [1] RSPM (R 4.3.0)  
## mime 0.12 2021-09-28 [1] RSPM (R 4.3.0)  
## miniUI 0.1.1.1 2018-05-18 [1] RSPM (R 4.3.0)  
## pkgbuild 1.4.3 2023-12-10 [1] RSPM (R 4.3.0)  
## pkgload 1.3.4 2024-01-16 [1] RSPM (R 4.3.0)  
## profvis 0.3.8 2023-05-02 [1] RSPM (R 4.3.0)  
## promises 1.2.1 2023-08-10 [1] RSPM (R 4.3.0)  
## purrr 1.0.2 2023-08-10 [1] RSPM (R 4.3.0)  
## R6 2.5.1 2021-08-19 [1] RSPM (R 4.3.0)  
## Rcpp 1.0.12 2024-01-09 [1] RSPM (R 4.3.0)  
## remotes 2.5.0 2024-03-17 [1] CRAN (R 4.3.2)  
## rlang 1.1.6 2025-04-11 [1] CRAN (R 4.3.2)  
## rmarkdown 2.25 2023-09-18 [1] RSPM (R 4.3.0)  
## sessioninfo 1.2.2 2021-12-06 [1] RSPM (R 4.3.0)  
## shiny 1.8.0 2023-11-17 [1] RSPM (R 4.3.0)  
## stringi 1.8.3 2023-12-11 [1] RSPM (R 4.3.0)  
## stringr 1.5.1 2023-11-14 [1] RSPM (R 4.3.0)  
## urlchecker 1.0.1 2021-11-30 [1] RSPM (R 4.3.0)  
## usethis 2.2.3 2024-02-19 [1] RSPM (R 4.3.0)  
## vctrs 0.6.5 2023-12-01 [1] RSPM (R 4.3.0)  
## xfun 0.52 2025-04-02 [1] CRAN (R 4.3.2)  
## xtable 1.8-4 2019-04-21 [1] RSPM (R 4.3.0)  
## yaml 2.3.10 2024-07-26 [1] CRAN (R 4.3.2)  
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library  
##   
## ──────────────────────────────────────────────────────────────────────────────

# 4 References

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2021. *Rmarkdown: Dynamic Documents for r*. <https://github.com/rstudio/rmarkdown>.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. <https://bookdown.org/yihui/rmarkdown>.

Xie, Yihui, Christophe Dervieux, and Emily Riederer. 2020. *R Markdown Cookbook*. Boca Raton, Florida: Chapman; Hall/CRC. <https://bookdown.org/yihui/rmarkdown-cookbook>.