# Building a Portfolio with Github and Quarto

Jason Bryer, Ph.D.

2025-02-19

# Static vs Dynamic Websites

*Static* websites displays the same pre-built content to every visitor…

While a **dynamic** websites generate content on the fly based on user interactions.

An important advantages of *static* websites include faster load times, less bandwidth, and cheaper hosting (often free).

# Static Website Frameworks

There are a number of static website frameworks including:

- Hugo
  - Lots of free themes
  - `blogdown` R package
  - Examples: my personal website and CV site
- Jekyll
  - Default framework for Github Pages
  - Lots of free themes
- Quarto
  - Relatively new but language agnostic.
  - Designed to work well with R, Python, Julia, SQL, and any language used by data scientists and programmers.
  - Example: DATA 606 course site, Visual Stats book

# Markdown

A common feature of most of the static website frameworks is that the site content is moslty written in Markdown. Markdown, originally created by John Gruber, is a lightweight markup language for creating formatted text using a plain-text editor. The idea is to write with minimal *markup* and let the website/document creator handle the specific styling.
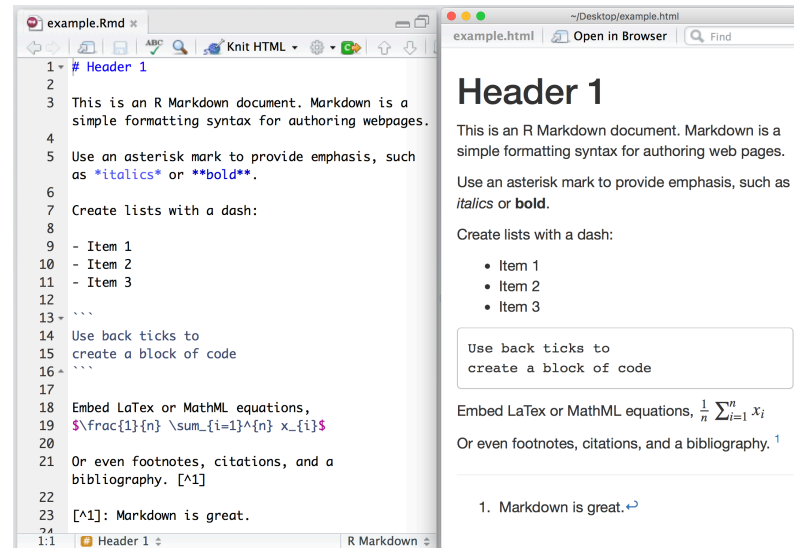


Here are some additional resources for learning markdown:

- Markdown Guide

- R Markdown quick tour

- Markdown for revealjs (this slide deck)

# Literate programming

Donald Knuth introduced a programming framework in 1984 called literate programming. The core idea is that a computer program is written in plain language with interspersed (i.e. embedded) code snippets that implement what is described in the plain text. This has been the foundation for researchers conducting reproducible research. For example, data scientists maintain one document that can contains the description of analyses along with the code that performs the analyses. When the document is rendered, all the code is executed and the output (e.g. tables, figures) are embedded within the final document (e.g. PDF, Word, HTML).

RMarkdown was an extension to allow for the embedding of R code within markdown documents.This was initially implemented in the knitr and later allowed for embedding of other languages including Python and SQL.

Quarto extends the ideas of RMarkdown but removes the requirement of R for rendering documents.

Both Quarto and RMarkdown use the following format to embed code that will be executed when the document is rendered:

```
1  ```{LANGUAGE, OPTIONS}
2  CODE GOES HERE
3  ```
```

Where LANGUAGE is the programming language (e.g. r, python, sql, bash, etc.).

# Common Code Chunk Options

Full list of options available here: https://quarto.org/docs/computations/execution-options.html

- `eval` - Evaluate the code chunk (if false, just echos the code into the output).

- `echo` - Include the source code in output

- `output` - Include the results of the code (`true`, `false`, or `asis`).

- `warning` - Include warnings in the output.

- `error` - Include errors in the output.

- `include` - Catch all for preventing any output (code or results) from being included (e.g. `include: false` suppresses all output from the code block).

- `fig-cap` - Figure caption.

- `fig-alt` - Alt text for the figure.

- `fig.align` - Figure alignment (`center`, `left`, `right`)

# Getting Started

Creating a new Quarto website from the command line:

```
1  quarto create project website mysite
```

Or you can clone the repository from this talk:

```
1  git clone https://github.com/jbryer/portfoliotalk.git
```
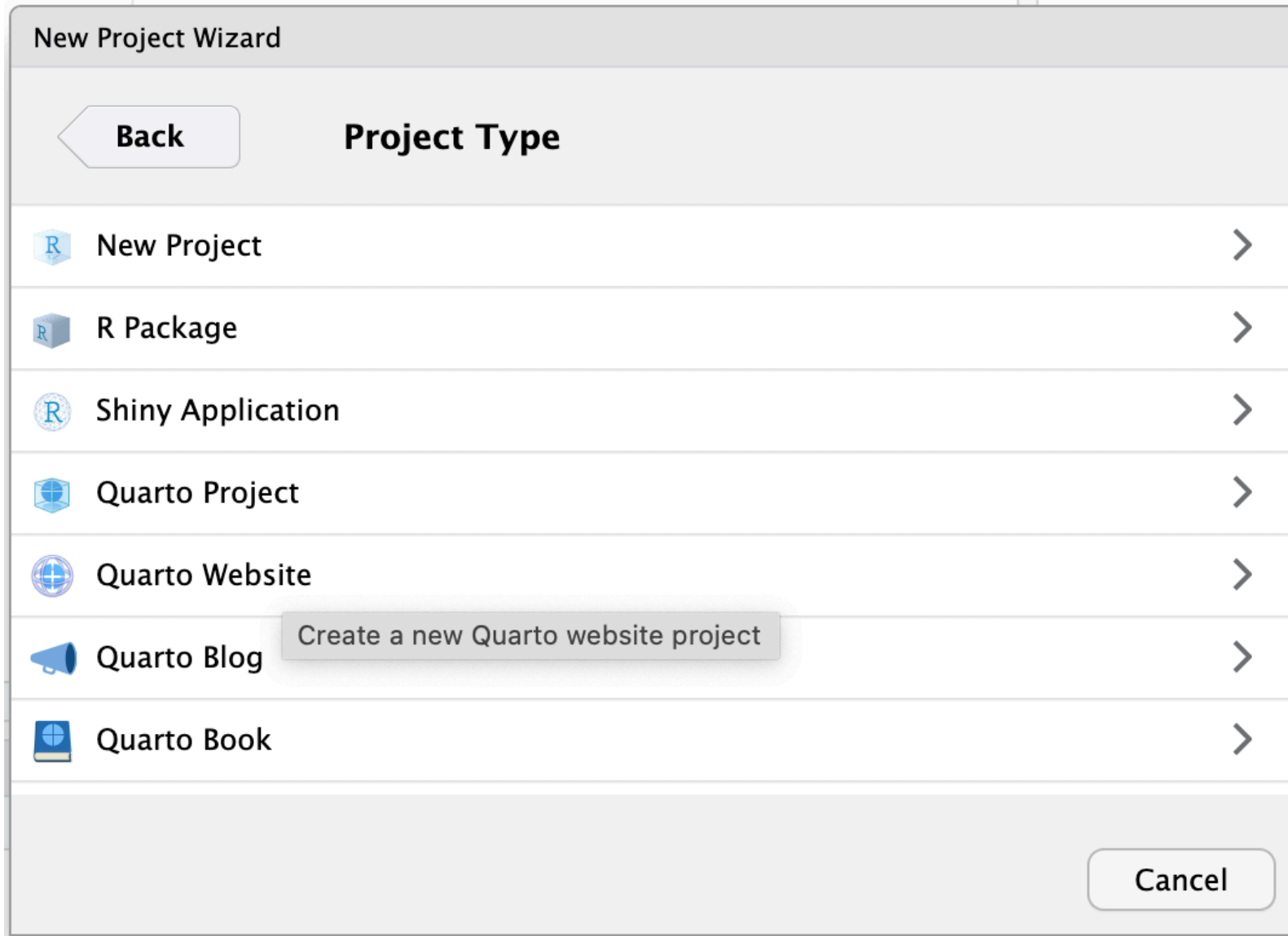
More detailed directions are available on the Quarto website including using VSCode and Rstudio.

# Getting Started with RStudio
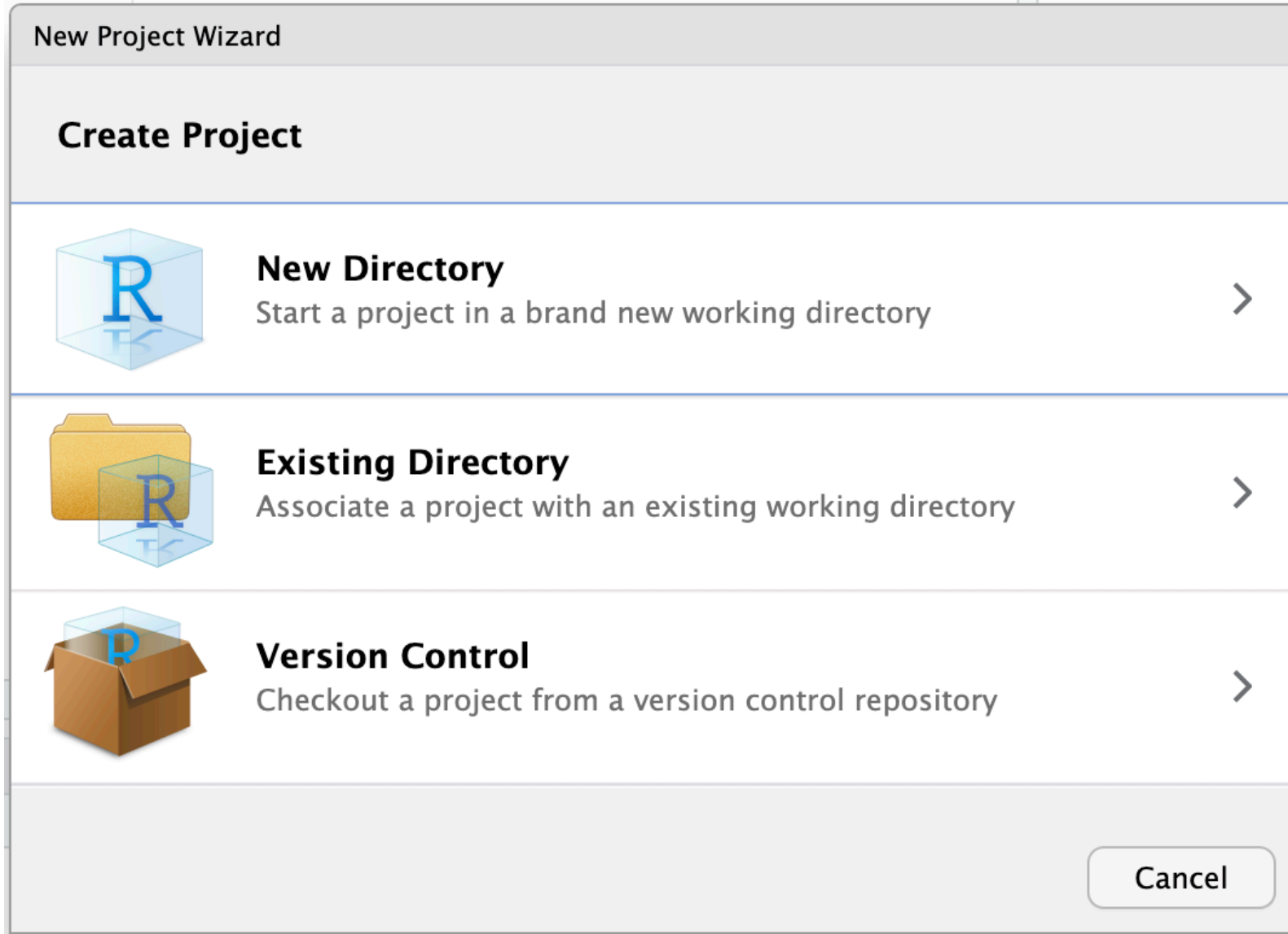
# Getting Started with RStudio (cont.)

New Project Wizard

**Back**    **Project Type**

| | |
|---|---|
|  New Project | > |
|  R Package | > |
|  Shiny Application | > |
|  Quarto Project | > |
|  Quarto Website | > |
|  Quarto Blog | > |
|  Quarto Book | > |

Create a new Quarto website project

Cancel

# Getting Started with RStudio (cont.)
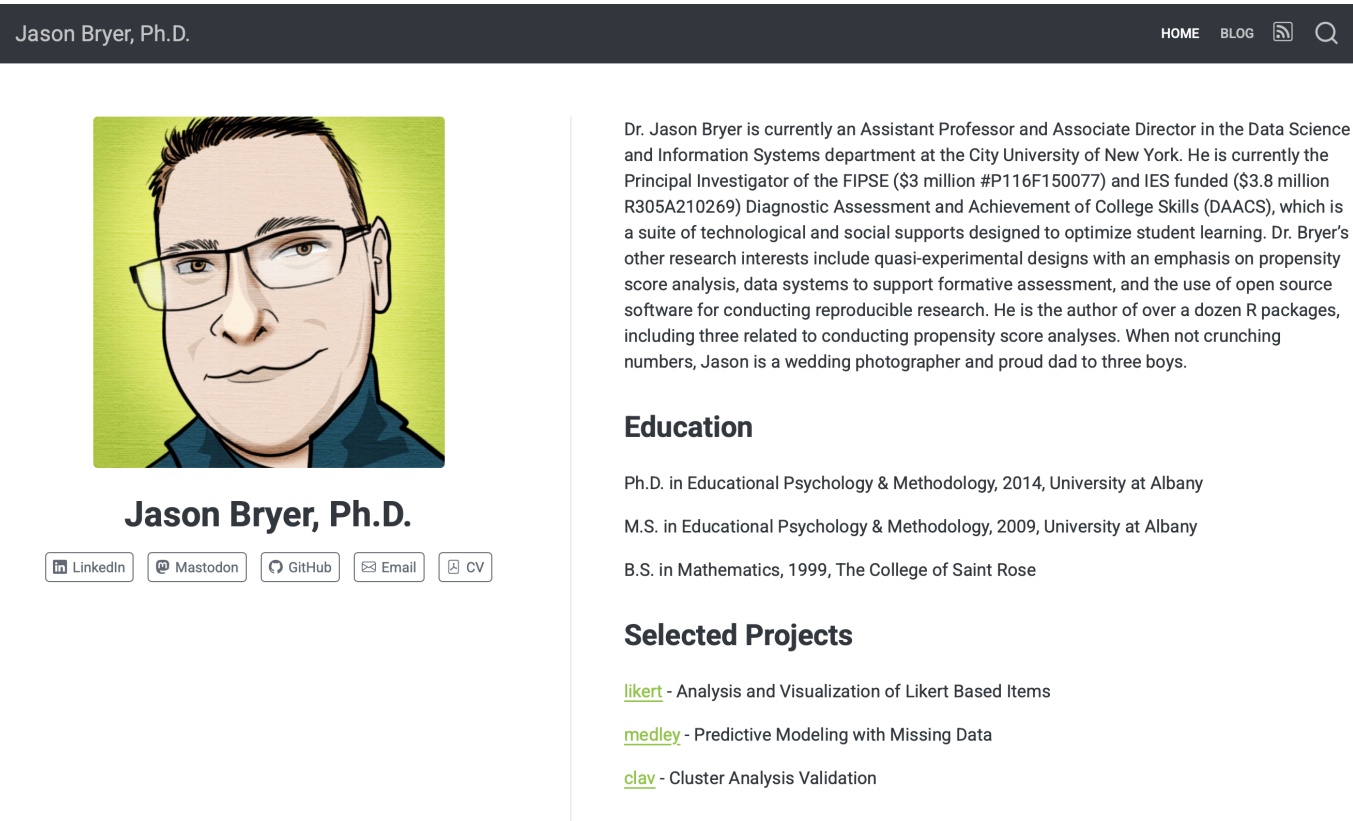
# Rendering and previewing

You can then preview the website using (this will run a small webserver):

```
1  quarto preview
```

Or to render the entire webiste:

```
1  quarto render
```

# Customizing your website

The `_quarto.yml` file contains a lot of options to customize your website. Full list of options is available here: https://quarto.org/docs/reference/formats/html.html

```
 1  project:
 2    type: website
 3    output-dir: docs
 4    render:
 5      - "*.qmd"
 6      - "!slides/"
 7    post-render:
 8      - "cp -rnv slides/. docs/slides/"
 9
10  website:
11    title: "Jason Bryer, Ph.D."
12    site-url: https://jbryer.github.io/portfoliotalk
13    description: "Sample portfolio website"
14    open-graph:
15      locale: en_US
16    twitter-card:
17      creator: "@jbryer"
18    navbar:
19      right:
20        - text: "Home"
21          href: index.html
22        - text: "Projects"
23          href: projects.html
24        - text: "Blog"
25          href: blog.html
26        - icon: rss
27          href: blog.xml
28    page-footer:
29      left: "Site created with Quarto by Jason Bryer."
30
```

# Home page

The `index.qmd` file is our homepage.

```
1   ---
2   title: "Jason Bryer, Ph.D."
3   about: # More info about this page type: https://quarto.org/docs/websites/website-about.html
4     template: trestles # Options include: jolla, trestles, solana, marquee, broadside
5     image: Headshot_Cartoon.jpg
6     links:
7       - text: LinkedIn
8         icon: linkedin
9         url: "https://www.linkedin.com/in/jasonbryer/"
10      - text: Mastodon
11        icon: mastodon
12        url: "https://vis.social/@jbryer"
13      - text: GitHub
14        icon: github
15        url: "https://github.com/jbryer"
16      - text: Email
17        icon: envelope
18        url: "mailto:jason.bryer@cuny.edu"
19      - text: CV
20        icon: file-pdf
21        url: "https://github.com/jbryer/CV/blob/master/Bryer_CV.pdf?raw=true"
22  comments: false
23  ---
24  ::: {.column-page}
25
26
27  Dr. Jason Bryer is currently an Assistant Professor and Associate Director in the [Data Science and Information Systems](h
28
29
30  ## Education
```

# Additional Pages

It is easy to add additional pages. The following can serve as a template for new pages.

```
 1  ---
 2  title: "Blog"
 3  listing:
 4    contents: projects
 5    sort: "title"
 6    type: grid # Options include default, grid, table
 7    categories: true
 8    sort-ui: true
 9    filter-ui: true
10  page-layout: full
11  title-block-banner: false
12  ---
13  ::: {.column-page}
14
15  :::
```

Remember, for a page to show up in the navigation bar it needs to be added to the `_quarto.yml` file as well:

```
 1  navbar:
 2    right:
 3      - text: "Home"
 4        href: index.html
 5      - text: "Projects"
 6        href: projects.html
 7      - text: "Blog"
 8        href: blog.html
 9      - icon: rss
10        href: blog.xml
```

# Blog posts

Blog posts can be placed in the `/posts` directory. Here is an example QMD file for a blog post:

```
1  ---
2  title: "Building a portfolio with Github and Quarto"
3  description: "Slides for a talk on how to build a portfolio website using Github"
4  date: "2025-02-19"
5  categories: ["Github", "Quarto"]
6  image: 2025-02-19-Github_Portfolio.png
7  ---
8
9  This is a blog post...
```

The `blog.qmd` file builds the index of posts. Check that file as there are a few things you can customize.

```
1  ---
2  title: "Blog"
3  listing:
4    contents: posts
5    sort: "date desc"
6    type: default # Options include default, grid, table
7    categories: true
8    sort-ui: true
9    filter-ui: true
10   # image-height: "0"
11   feed:
12     categories: ["Quarto", "Github"]
13 page-layout: full
14 title-block-banner: false
15 ---
16 ::: {.column-page}
17
18 :::
```

Note that Quarto will automatically build an RSS feed, in this case called `blog.xml`.

# Github Pages

In order to ensure our project is ready to publish to Github, we need to do a couple of things. First, we need to change the output directory for when Quarto renders the website to `docs/`. In the `_quarto.yml` file, change the `outout-dir` parameter:

```
1  project:
2    type: website
3    output-dir: docs
```

Second, Github pages defaults to Jekyll for rendering. To tell Github to not use Jekyll we need to add a file, `.nojekyll`, to the root of our project. Run the following in the terminal appropriate for your platform:

Mac/Linux

```
1  touch .nojekyll
```

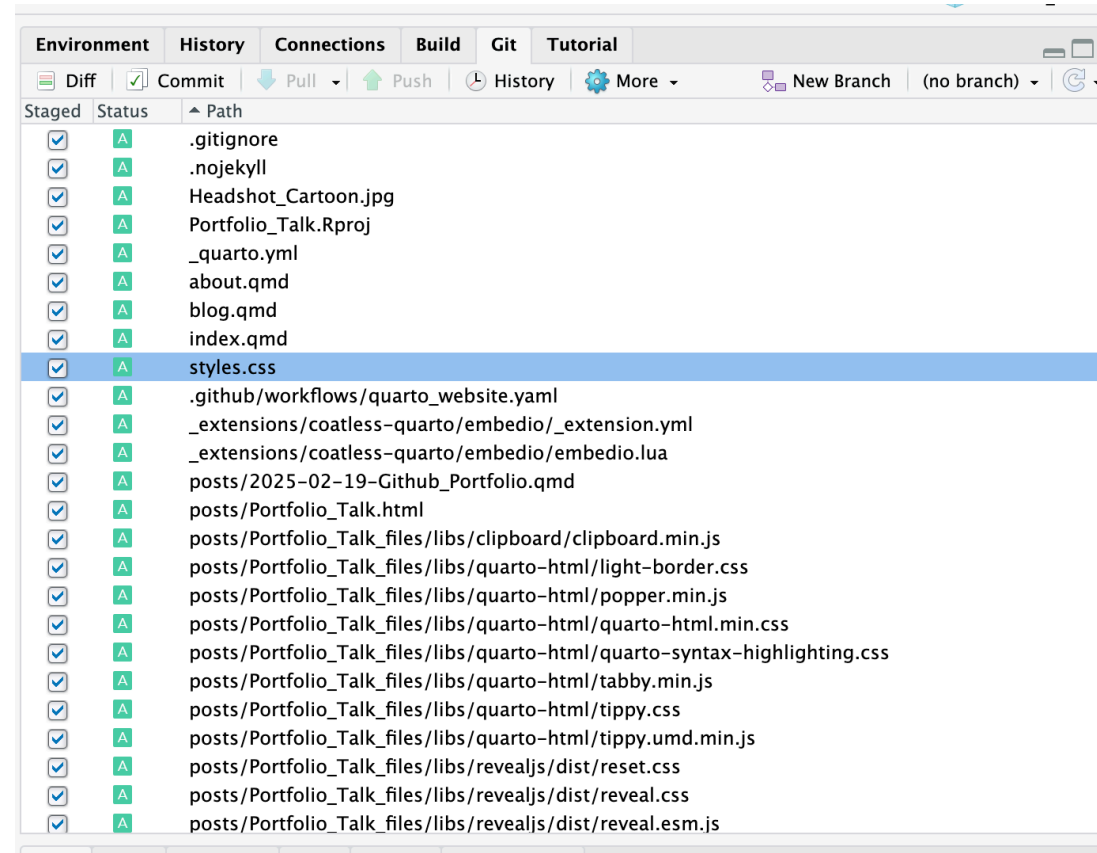Windows

```
1  copy NUL .nojekyll
```

# Creating a Local Git Repo

The following (terminal) commands will create, add, then commit our files to a local git repository:

```
1  git init -b main                 # Initialize Git repo.
2  git add .                        # Add the files to the repo
3  git commit -m "Initial version"  # Commit the files to the repo
```

Or use the RStudio git integration:

# Publishing to Github

1. Got to https://github.com/new

2. Create a new repository on Github (see figure to the right). Note that it is public and we did not initialize anything else (e.g. readme, .gitignore, license, etc.).

3. The resulting page will have directions, but for now just copy the repository URL to be used in the next step.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

*Required fields are marked with an asterisk (*).*

**Repository template**

No template ▾

Start your repository with a template repository's contents.

**Owner ***     **Repository name ***

🧑 jbryer ▾   /   portfoliotalk

✅ **portfoliotalk is available.**

Great repository names are short and memorable. Need inspiration? How about **symmetrical-parakeet** ?

**Description** (optional)

Creating a portfolio with Github|

○ 📖 **Public**
    Anyone on the internet can see this repository. You choose who can commit.

○ 🔒 **Private**
    You choose who can see and commit to this repository.

**Initialize this repository with:**

☐ **Add a README file**
    This is where you can write a long description for your project. Learn more about READMEs.

**Add .gitignore**

.gitignore template: None ▾

Choose which files not to track from a list of templates. Learn more about ignoring files.

**Choose a license**

License: None ▾

A license tells others what they can and can't do with your code. Learn more about licenses.

ⓘ You are creating a public repository in your personal account.

**Create repository**

# Pushing to Github (cont.)

4. Run the following command in the terminal. Note that you need to change the URL to your repository from step 3.

```
1  git remote add origin https://github.com/jbryer/portfoliotalk.git
```

5. Verify the repository was created correctly.

```
1  git remote -v
```

6. Push the local repository to Github.

```
1  git push -u origin main
```
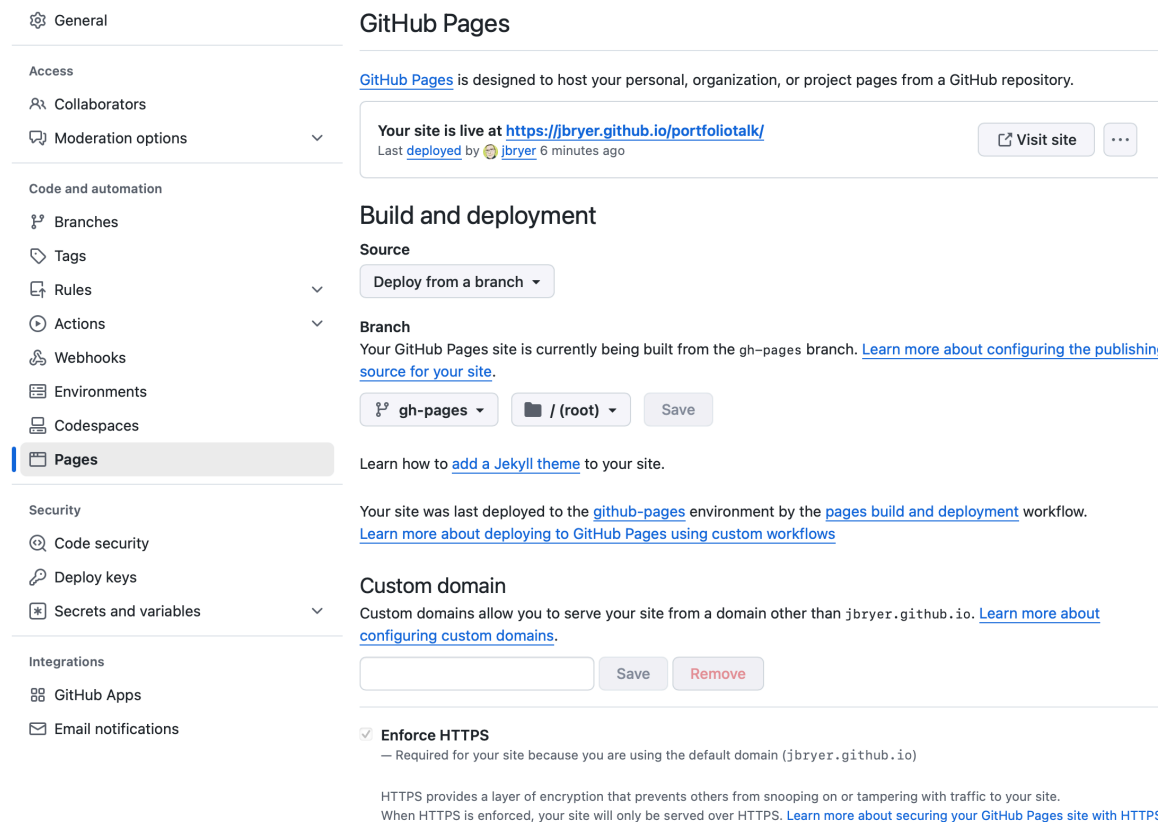
The source code for your website should not be on Github.

# Deploying website to Github

Now we can publish the website to Github. Note that this publishes to the gh-pages branch.

```
1   quarto publish gh-pages
```

We can verify the site is configured correctly here:

https://github.com/jbryer/portfoliotalk/settings/pages

# Custom Domain (optional)

You can register a custom domain at many places. I recommend Hover since it is easy to use and reasonably priced. Important steps: create four A records pointing to Github's IP addresses: 185.199.108.153, 185.199.109.153, 185.199.110.153, and 185.199.111.153.

For subdomains, create CNAME record pointing to GITHUB_USER.github.io where GITHUB_USER is your Github username.

**hover**    Domains    Emails    Nameservers    Forwards    Settings                    Help    Sign Out

View all domains

Jump to another domain

## bryer.org                    OVERVIEW    DNS    CONNECT    FORWARDS    EMAIL    ADVANCED

ADD A RECORD    Filter your records                    Bulk edit: Select

| | TYPE | HOST ↑ | VALUE | TTL | ADDED BY | | |
|---|---|---|---|---|---|---|---|
| | A | @ | 185.199.108.153 | 15 Minutes | | EDIT | ✕ |
| | A | @ | 185.199.109.153 | 15 Minutes | | EDIT | ✕ |
| | A | @ | 185.199.110.153 | 15 Minutes | | EDIT | ✕ |
| | A | @ | 185.199.111.153 | 15 Minutes | | EDIT | ✕ |
| | CNAME | visualstats | jbryer.github.io | 15 Minutes | | EDIT | ✕ |
| | CNAME | www | jbryer.github.io | 15 Minutes | | EDIT | ✕ |

Detailed directions on using a custom domain are available on Github's website

# Github Actions

We can automate the generation and deployment of our website using Github actions. The file below is located in the `.github/workflows/` directory.

```
1  on:
2    push:
3      branches: [main, master]
4
5  name: Render and Publish Quarto WEbsite
6
7  permissions:
8      contents: write
9      pages: write
10
11 jobs:
12   build-deploy:
13     runs-on: ubuntu-latest
14
15     steps:
16       - name: Check out repository
17         uses: actions/checkout@v4
18
19       - name: Set up Quarto
20         uses: quarto-dev/quarto-actions/setup@v2
21         env:
22           GH_TOKEN: ${{ secrets.GITHUB_TOKEN }}
23         with:
24           # To install LaTeX to build PDF book
25           tinytex: true
26           # uncomment below and fill to pin a version
27           # version: SPECIFIC-QUARTO-VERSION-HERE
28
29       # add software dependencies here and any libraries
```

# Github Actions

You can check the status of your Github actions here: https://github.com/jbryer/portfoliotalk/actions

# Thank you!

Links:

- Github repository: https://github.com/jbryer/portfoliotalk

- Example website: https://jbryer.github.io/portfoliotalk/

- Slide deck:
  https://jbryer.github.io/portfoliotalk/slides/Portfolio_Talk.htr



Additional Resources:

- Tom Mock's workshop materials from 2022 rstudio::conf

- Quarto Project Website

- Gallery of Quarto Projects (including websites)