Guide for Managing AMORE Website[1]

Ingebjørg A. Iversen

2025-11-13

[1]Thanks to Alina and Heemin for your contributions in developing this guide.

Abstract

This guide provides comprehensive instructions for managing and maintaining the AMORE (Active Monitoring of Oxytocin Research Evidence) website platform. It covers the technical architecture spanning Quarto-based static website development, GitHub version control, Netlify deployment, and ShinyApps.io integration for the interactive Living Meta-Analysis Directory. The guide is designed for lab members adding new LMA project pages and administrators maintaining the platform infrastructure. It includes detailed explanations of the repository structure, file organization, YAML configuration, styling implementation, and deployment processes.

Table of contents

Chapter 1

Introduction

The AMORE (Active Monitoring of Oxytocin Research Evidence) website is a platform for hosting and managing living meta-analyses for oxytocin research. This guide provides comprehensive instructions for lab members and administrators to manage, update, and maintain the website.

Live Website: https://amore-project.org

1.0.1 Who this guide is for:

- Lab members adding new LMA projects
- Website administrators maintaining platform

1.0.2 Important contacts

Ingebjørg A. Iversen

email: iaiversen@hotmail.com, ingebjai@uio.no

Daniel Quintana

email: daniel.quintana@psykologi.uio.no, danielqu@uio.no

1.1 How AMORE website is programmed and hosted

AMORE was built using RStudio and Quarto. Git is used for version control, collaboration and hosting website in GitHub repository. Netlify is deploying the website and accessing the script from the github repository. Shinyapp host the app.R (backend to the directory hosting living meta-analysis on AMORE). Nettskjema delivers contact forms used as medium between platform users and the AMORE team.

1.2 Git, Shinyapp, and netlify

The AMORE platform consists of three interconnected services:

1.2.1 Git (Version Control & Collaboration)

Repository: iaiversen/AMORE-webpage.

Stores all source code. Tracks changes with commit history. Enables collaboration through pull requests.

Important: Never commit sensitive data (API keys, passwords)

1.2.2 GitHub repository

Repository: AMORE-webpage

Repository URL: https://github.com/iaiversen/AMORE-webpage

Created by github account: iaiversen (Ingebjørg Anjadatter Iversen)

Collaborators: dsquintana (Daniel quintana), AMSartorius (Alina Sartorius), HeeminK (Heemin Kang).

What collaborators (Alina, Dan, Heemin) can do:

- Review, comment, approve, merge and close pull requests by external contributors

What only repository owner can do:

- Add/remove collaborators
- Change repository settings (make public/private, etc.)
- Delete the repository
- Transfer ownership

1.2.2.1 Repository License

AMORE-webpage repository has the MIT license. It gives open access. The LICENSE file is in the root folder (AMORE-webpage).

1.2.3 ShinyApps.io (Shiny App Hosting)

Hosts the Living Meta-Analysis directory (app.R) Free tier limitations: limited active hours, connection timeouts URL: https://meta-oxytocin.shinyapps.io/shiny-meta/ Embedded in Living_meta-analysis_Directory.qmd via iframe

1.2.3.1 Shinyapp.io login credentials (only for BNE lab members)

info found In Teams –> manuscript –> AMORE –> Website technicals –> AMORE_credentials.xlsx

1.2.4 Netlify (Website Hosting)

Automatically deploys from GitHub Builds site from _site folder after Quarto render Domain: amore-project.org Configuration in netlify.toml

Deployment workflow:

Make changes locally in RStudio. Test by rendering (quarto render or Render button) Commit and push to GitHub. Netlify automatically rebuilds and deploys. For Shiny app: Deploy separately using rsconnect::deployApp() or push publish button in the right top corner of the terminal.

1.2.4.1 Netlify login credentials (only for BNE lab members)

info found In Teams –> manuscript –> AMORE –> Website technicals –> AMORE_credentials.xlsx

1.3 RStudio and Quarto

RStudio is your primary development environment. Install:

R (version 4.0.0+) from CRAN RStudio Desktop from Posit Quarto CLI - Usually bundled with RStudio, or install from quarto.org

1.3.1 The Different Languages

The AMORE website is built using multiple programming languages and frameworks that work together: Quarto is multilingual, Just specify in the code chunk what language is used.

Languages in use:

- Quarto Markdown (.qmd): Primary content files for pages
- SCSS/CSS: Styling and responsive design
- JavaScript: Interactive functionality and client-side behavior
- R: Shiny app backend, data processing, and setup scripts
- HTML: Embedded within Quarto files for custom components Shell scripts: Deployment and automation tasks

1.4 Nettskjema

Ediorial right for nettskjema forms:**

- ingebjai@uio.no
- danielqu@uio.no

The users reach out to the AMORE team through Nettskjema. They find the nettskjema through links on the AMORE website or directly on the contact page.

The Nettskjema forms are:

1. Contact form AMORE
   - Contact button on home page (index.qmd) transfers platform users to this form.
   - Redirects to the other three forms (Propose your project, Update an existing living meta-analysis or Get help or share feedback).
2. Propose your project
3. Get help or share feedback
4. Update an existing living meta-analysis

Emails of responses to any of these forms goes to:

- ingebjai@student.sv.uio.no
- daniel.quintana@psykologi.no

NB! Dan streamlines the nettskjema responses to the AMORE expert steering committee for evaluation of projects or other inquiries.

If you need editorial right to the nettskjema, reach out to Dan.

Chapter 2

Getting started

This section guides you through setting up your development environment, providing the essential setup instructions.

2.0.1 Good to know before starting

Throughout this guide, you'll see code blocks with different labels that indicate language used, or where/how to run the code. This table explains how to understand the different code blocks for this guide document.

R blocks	yaml blocks	bash blocks	html and SCSS blocks
{r} This code block is used to indicate the content of the code block belongs to a script or file inside the AMORE.Rproj. For this guide it does not equal the language to be used is R.	{yaml} YAML is a format for storing structured information. In AMORE.Rproj, each project file YAML headers contains all the searchable metadata that allows the Shiny app to find, filter, and display projects correctly.	{bash} The bash code blocks indicates commands are for command line interfaces. Bash, powershell and RStudio Terminal are all command line interfaces. (I personally use Windows powershell as my terminal). The blocks contain Git commands that work in all mentioned command interfaces.	{html}, [scss} Html and scss code blocks indicates that the content in the code block is written using the specified language. The languages are used across different files and scripts all withing the AMORE.Rproj project. Scss is mainly found in the styles.scss script

2.1 Prerequisites

- R (Version 4.0.0 or higher) - programming language
- Rstudio - Integrated Development Environment (IDE) for R.
- Quarto - Document publishing system for creating websites or reports
- Github account
- Git version installed on your computer (not browser version).

2.2 Access Methods to Git - three options

Choose the method that matches your access level and preferred workflow:

Method 1: For Collaborators (Command Line in bash or powershell)

Method 2: For External Contributors (Command Line in bash or powershell - Fork Workflow)

Method 3: RStudio Git Integration (For Both Collaborators and Contributors)

2.2.0.1 Understanding Access Levels

Collaborators: - Have direct write access to the AMORE repository - Can push changes directly to the main repository - Changes go live on the website immediately (after Netlify rebuild) - Can review and merge pull requests from external contributors - To become a collaborator: Contact Ingebjørg Iversen (GitHub: iaiversen) to request access

External Contributors: - Work through the fork workflow (Method 2) - Cannot push directly to the AMORE repository - Must submit pull requests for review - Any collaborator can review and merge your contributions - Ideal for one-time contributions or occasional updates

2.2.1 Authentication Setup

Before you can push changes to the repository, you need to set up authentication on your computer. Even though you've been added as a collaborator on GitHub, your computer needs a way to prove your identity when pushing changes. Once configured, you'll be able to push changes to any GitHub repository you have access to.

You have two options: Personal Access Token (PAT) or SSH keys. Choose one method below.

2.2.1.1 Option 1: Personal Access Token (PAT)

Step 1: Generate a Personal Access Token on GitHub

1. Go to GitHub and log in
2. Click your profile picture (top right) → Settings
3. Scroll down to Developer settings (bottom of left sidebar)
4. Click Personal access tokens → Tokens (classic)
5. Click Generate new token → Generate new token (classic)
6. Give your token a descriptive name (e.g., "AMORE-webpage-access")
7. Set expiration (recommend 90 days or 1 year)
8. Under Select scopes, check the `repo` box (this gives full control of private repositories)
9. Scroll down and click Generate token
10. IMPORTANT: Copy the token immediately - you won't be able to see it again!

Step 2: Use the token when pushing

When you try to push for the first time:

```
git push origin main
```

Git will prompt you for credentials: - Username: Enter your GitHub username - Password: Paste your Personal Access Token (NOT your GitHub password)

Step 3: Store credentials (optional but recommended)

To avoid entering the token every time, you can store it using Git Credential Manager:

On Windows:

```
git config --global credential.helper wincred
```

On Mac:

```
git config --global credential.helper osxkeychain
```

On Linux:

```
git config --global credential.helper store
```

After running this command, the next time you enter your token, it will be saved securely.

2.2.1.2 Option 2: SSH Keys - More convenient for frequent use

Step 1: Check if you already have SSH keys

Open PowerShell (Windows) or Terminal (Mac/Linux) and run:

```
ls ~/.ssh
```

If you see files named `id_rsa` and `id_rsa.pub` (or `id_ed25519` and `id_ed25519.pub`), you already have SSH keys and can skip to Step 3.

Step 2: Generate SSH keys (if you don't have them)

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Press Enter to accept the default file location. You can optionally add a passphrase for extra security (or press Enter to skip).

Step 3: Copy your public key

On Windows (PowerShell):

```
cat ~/.ssh/id_ed25519.pub | clip
```

On Mac:

```
cat ~/.ssh/id_ed25519.pub | pbcopy
```

On Linux or if the above don't work:

```
cat ~/.ssh/id_ed25519.pub
```

Then manually copy the output.

Step 4: Add SSH key to GitHub

1. Go to GitHub → Settings (click your profile picture)
2. Click SSH and GPG keys (in the left sidebar)
3. Click New SSH key
4. Give it a title (e.g., "My Work Laptop")
5. Paste your public key in the "Key" field
6. Click Add SSH key

Step 5: Test the SSH connection to GitHub

```
ssh -T git@github.com
```

You should see a message like: "Hi [username]! You've successfully authenticated…"

That's it for initial setup! When you clone repositories using SSH URLs (like 'git@github.com:…'), they'll automatically use your SSH keys.

2.2.2 Method 1: For Collaborators (Command Line in bash or powershell)

If you are a direct collaborator with write access:

```
{bash}
#| label: collaborator-initial-setup
#| description: Clone repository and install dependencies for collaborators
#| eval: false
#| echo: true

# Clone the repository directly
git clone https://github.com/iaiversen/AMORE-webpage.git
cd AMORE-webpage



# Install R dependencies
Rscript Setup.R
```

Daily Workflow:

```
{bash}
#| label: collaborator-daily-workflow
#| description: Standard git workflow for collaborators with write access
#| eval: false
#| echo: true

# Get latest changes before starting work
git pull origin main

# Make your changes
# ... edit files ...

# Stage your changes
git add .

# Commit with descriptive message
git commit -m "Add new project: [Project Name]"
```

```
# Push directly to main repository
git push origin main
```

To check if you're a collaborator: - Visit: https://github.com/iaiversen/AMORE-webpage/settings - If you can access Settings, you're a collaborator - If you get "404 Not Found", use Method 2 below

---

2.2.3 Method 2: or External Contributors (Command Line in bash or powershell - Fork Workflow)

If you are NOT a direct collaborator:

Initial Setup (One Time):

Step 1: Fork the Repository 1. Go to https://github.com/iaiversen/AMORE-webpage 2. Click the Fork button (top right corner) 3. This creates your own copy at: https://github.com/[YOUR-USERNAME]/AMORE-webpage

Step 2: Clone YOUR Fork

```
{bash}
#| label: fork-clone-setup
#| description: Clone your forked repository to your local machine
#| eval: false
#| echo: true

# Clone your fork to your computer
git clone https://github.com/[YOUR-USERNAME]/AMORE-webpage.git
cd AMORE-webpage
```

Step 3: Link to Main Repository

```
{bash}
#| label: fork-upstream-setup
#| description: Link your fork to the original repository as upstream
#| eval: false
#| echo: true

# Add the original repository as "upstream"
git remote add upstream https://github.com/iaiversen/AMORE-webpage.git

# Verify your remotes
git remote -v
# Should show:
# origin    https://github.com/[YOUR-USERNAME]/AMORE-webpage.git (your fork)
# upstream  https://github.com/iaiversen/AMORE-webpage.git (main repo)
```

Understanding Your Fork Setup

When you fork and clone, you're working with TWO GitHub repositories:

Your fork (‘origin‘):

- Your personal copy: ‘https://github.com/[YOUR-USERNAME]/AMORE-webpage‘ - You have full push access here - When you run ‘git push‘, changes go here - Think of this as your workspace

The main AMORE repository (‘upstream‘):

- The official repository: ‘https://github.com/iaiversen/AMORE-webpage‘ - You CANNOT push directly here - You can only pull updates from here - This is where the live website gets updated from

To get your changes into the main AMORE repository: - Go to GitHub and create a Pull Request (see “Creating a Pull Request” below) - AMORE maintainers review and merge your changes - Only then do your changes appear on the live website

To get updates from the main AMORE repository:

```
git pull upstream main
git push origin main  # Update your fork with the latest changes
```

Regular Workflow (After Initial Setup)

Step 1: Before starting new work - sync with upstream

Always get the latest changes from the main repository first:

```
#| label: fork-sync-upstream
#| description: Sync your local repository with the main AMORE repository
#| eval: false
#| echo: true

# Get latest changes from main AMORE repository
git pull upstream main

# Push these updates to your fork
git push origin main
```

Step 2: Make your changes

Edit files, create new content, etc.

Step 3: Commit and push to YOUR fork

```
#| label: fork-commit-push
#| description: Commit changes and push to your fork
#| eval: false
#| echo: true

# Stage your changes
git add .

# Commit with a descriptive message
git commit -m "Add new project page for [project name]"

# Push to YOUR fork (not the main repository)
git push origin main
```

Step 4: Create a Pull Request

1. Go to your fork on GitHub: ‘https://github.com/[YOUR-USERNAME]/AMORE-webpage‘ 2. You’ll see a yellow banner: “This branch is X commits ahead of iaiversen:main” 3. Click **Contribute** → **Open pull request** 4. Fill in the Pull Request form: - **Title:** Brief description of changes (e.g., “Add meta-analysis project page for oxytocin and trust”) - **Description:** Explain what you changed and why 5. Click **Create pull request

Step 5: Wait for review

- AMORE maintainers will review your changes - They may ask for modifications or clarifications - Once approved, they'll merge your changes - Your changes will then appear on the live website after Netlify rebuilds it

2.2.4 Method 3: Method 3: RStudio Git Integration (For Both Collaborators and Contributors)

RStudio has built-in Git integration that provides a visual interface for Git operations. This method works for both collaborators (Method 1) and fork users (Method 2).

Method 3: Prerequisites

- Complete the Authentication Setup section above

- Method 2 users only: Complete the fork setup (fork the repository on GitHub first)

Initial Setup: Clone in RStudio

Step 1: Create a new project from version control

1. In RStudio: File → New Project → Version Control → Git

2. Repository URL:

  - Method 1 (collaborators): ‘https://github.com/iaiversen/AMORE-webpage‘
  - Method 2 (fork users): ‘https://github.com/[YOUR-USERNAME]/AMORE-webpage‘

3. Project directory name: Leave as ‘AMORE-webpage‘

4. Create project as subdirectory of: Choose your location (e.g., ‘Documents/‘)

5. Click Create Project

RStudio will download all files and open the existing ‘AMORE.Rproj‘ project with Git already connected.

Step 2: Set up upstream remote (external collaborators)

If you're using a fork, connect to the main repository:

1. In RStudio: Tools → Terminal → New Terminal

2. Run:

```
git remote add upstream https://github.com/iaiversen/AMORE-webpage.git
```

Daily Workflow in RStudio

You'll see a Git tab in RStudio (usually upper-right pane).

Making and Saving Changes

1. Edit your files in the project

2. Git tab: Check boxes next to files you want to save

3. Click Commit

4. Write a descriptive commit message

5. Click Commit button

6. Click Push

Understanding What “Push” Does

Collaborators:

- Push sends changes directly to the main AMORE repository

- Changes go live on the website after Netlify rebuilds (usually 2-3 minutes)

Fork users:

- Push sends changes only to YOUR fork

- You must create a Pull Request on GitHub to propose changes to the main repository

- See “Creating a Pull Request” section in Method 2

Getting Latest Changes

Ccollaborators:

- Click the Pull button (blue down arrow) in the Git tab

Fork users:

To get updates from the main AMORE repository:

1. In RStudio: Tools → Terminal → New Terminal

2. Run:

```
git pull upstream main
```

Click Push in the Git tab to update your fork

2.2.4.1 Reopening Your Project

After initial setup:

- File → Open Project → Recent Projects** → select AMORE

- Or File → Open Project → navigate to ‘AMORE.Rproj‘

- Or double-click ‘AMORE.Rproj‘ in your file system

RStudio remembers all Git settings - no reconfiguration needed!

2.3 Dependencies

Install all required packages:

Open `Setup.R` and run the entire script.

```
{r}
#| label: Setup script for dependencies
#| description: copy paste into r console
#| eval: false
#| echo: true

#You need to be in the root folder AMORE
source("scripts/Setup.R") # write this in console
# Or just find the script yoursekf and run it.
```

The setup.R script installs:

Core packages:

- `rmarkdown`, `knitr`, `quarto` - Document rendering
- `shiny`, `rsconnect` - Shiny app deployment
- `DT`, `yaml`, `fs`, `httr`, `jsonlite` - Data handling
- `bslib`, `sass` - Styling
- `tinytex` - LaTeX/PDF support

2.3.0.1 Verify Installation

Test that everything works:

```
{bash}
#| label: Verify installation in r console
#| eval: false
#| echo: true

("quarto --version") # Check Quarto system
quarto::quarto_render("index.qmd") # Test rendering a single page
```

Chapter 3

Project structure and file explanations

```
{bash}
#| label: Root folder (AMORE-webpage) and the file structure
#| eval: false
#| echo: true

AMORE-webpage/
├── index.qmd                          # Homepage (root level)
├── _quarto.yml                        # Main configuration file
├── LICENSE
├── README.md
├── .gitignore
├── .node-version
├── package.json
├── netlify.toml                       # Netlify deployment config
├── netlify-setup-quarto.sh            # Quarto setup for Netlify
├── install-quarto.sh                  # Quarto installation script
├── AMORE.Rproj                        # RStudio project file
│
├── .quarto/                           # Quarto build cache (auto-generated)
│   ├── idx/                           # Index files
│   ├── xref/                          # Cross-references
│   ├── listing/
│   ├── preview/
│   └── _freeze/
│
├── assets/
│   ├── favicons/                      # All favicon files
│   │   ├── favicon.ico
│   │   ├── apple-touch-icon.png
│   │   ├── favicon-96x96.png
│   │   ├── site.webmanifest
│   │   └── amore.favicon.ico
│   │
│   ├── images/                        # Logo and visual assets
│   │   ├── logo.unfinished_nowhite- Copy.png
│   │   ├── LiMAs.jpg
│   │   └── [other image files]
│   │
│   └── styles/
│       └── styles.scss                # Main stylesheet
```

```
│
├── pages/                           # All content pages
│   ├── about.qmd
│   ├── contact.qmd
│   ├── guidelines.qmd
│   ├── Living_meta-analysis_Directory.qmd
│   ├── Protocol_checklist.qmd
│   ├── Resources.qmd
│   └── Standardization.qmd
│
├── LMAs/                            # Living Meta-Analysis projects
│   ├── _lma-template.qmd            # Template for new LMAs
│   └── project pages files
│
├── docs/                            # Documentation (working files)
│   ├── _AMORE_Guide.qmd             # Website management guide
│   ├── _recoveryfile.app.qmd
│   └── _recoveryfile.qmd
│
├── scripts/                         # Utility scripts
│   └── Setup.R                      # R dependencies installer
│
├── shiny-meta/                      # Shiny application
│   ├── app.R                        # Main Shiny app
│   ├── deploy.R                     # Deployment script
│   ├── README.md                    # Shiny app documentation
│   ├── .Renviron                    # Saves shiny meta tokens
│   │
│   ├── rsconnect/                   # Shiny deployment config
│   │   └── shinyapps.io/
│   │       └── meta-oxytocin/
│   │           └── shiny-meta.dcf
│   │
│   └── www/                         # Shiny app assets
│       └── amore.favicon.ico
│
└── _site/                           # Generated website (DO NOT EDIT)
    ├── index.html
    ├── search.json
    ├── assets/
    ├── pages/
    ├── LMAs/
    ├── shiny-meta/
    └── site_libs/
```

3.1 Root files

Root files are located directly in the `AMORE-webpage/` folder. These files control the fundamental behavior, configuration, and metadata of the entire website project. They define how the site is built, deployed, and maintained.

Key characteristics of root files:

- Control project-wide settings and behavior
- Required for the website to function properly

- Should be modified carefully as changes affect the entire site
- Most are configuration or metadata files (not content)

3.1.1 index.qmd

The homepage of the AMORE website. This is the first page visitors see when they navigate to https://amore-project.org.

Type: Quarto markdown document (content file)

CSS classes used directly in index.qmd:

- `.content-section` - Main container with transparent background on gradient
- `.header-logo` - Logo image styling (centering, max-width)
- `.subtitle-text` - Centered white subtitle text with specific font size
- `.primary-button` - Call-to-action button styling
- `.contact-buttons` - Button container for centering
- `.BNL-text` - Text styling for lab attribution

Global styles that also affect this page:

- `body` - Gradient animated background (visible behind `.content-section`)
- Typography rules (`p`, `a`, headings)
- `.primary-button:hover` - Hover effects on the CTA button
- Responsive breakpoints that adjust layout on mobile/tablet

3.1.2 _quarto.yml

The main configuration file that controls how the entire website is built and structured.

Type: YAML configuration file

Important notes:

- Changes here affect the entire site
- Always test after editing with `quarto preview`
- Incorrect YAML indentation will break the site
- Navbar order controls menu item order

3.1.3 LICENSE

Defines the legal terms under which the AMORE website code and content can be used, modified, and distributed.

Type: Plain text legal document

License: MIT open source license. MIT license explanations.

3.1.4 README.md

Brief description of the AMORE website project displayed on the GitHub repository page.

Type: Markdown document

Purpose:

- First thing people see on GitHub
- Explains what the project is

3.1.5 .gitignore

Purpose: Tells Git which files and folders to ignore (not track or upload to GitHub).

Type: Plain text configuration file

Purpose:

- Keeps repository clean
- Prevents uploading generated/temporary files
- Reduces repository size
- Protects sensitive information

```
{r}
#| label: .gitignore contents
#| eval: false
#| echo: true

# R files
*.log
*.tex
*.aux
.Rproj.user
.Rproj.user/
.Rhistory
.RData
.Ruserdata
*.Rproj

# Quarto and website specific files
_site/
/.quarto/
*_cache/
*.bak
*_error.scss
*.backup
*.ipynb
/_site/
/_freeze/
/_book/
/*_files/

# Generated HTML files in SOURCE directories (not _site)
```

```
/*.html
/pages/*.html
/LMAs/*.html
/docs/*.html

# Generated dependency folders in SOURCE directories
/*_files/
/pages/*_files/
/LMAs/*_files/
/docs/*_files/



# OS files
.DS_Store
Thumbs.db

# Test files

docs/_recoveryfile.qmd
docs/_recoveryfile.app.qmd

# Generated files
_quarto_internal_scss_error.scss

# Backup files
*.backup
*.bak
```

Why exclude these:

- RStudio files are user-specific
- _site/ is generated, not source code
- Cache files speed up local rendering but aren't needed in repo
- Reduces repository size

Never add to .gitignore:

- Source files (.qmd, .R, .scss)
- Configuration files (_quarto.yml, netlify.toml)
- Assets (images, logos)

3.1.6 .node-version

Specifies which Node.js version to use for building the website.

Type: Plain text version file

Purpose:

- Ensures consistent Node.js version across environments

- Used by Netlify during deployment
- Prevents version compatibility issues

Current content:

- Single line with Node.js version number (e.g., "16")

When to edit:

- Updating Node.js requirements
- Fixing deployment issues related to Node.js version
- Usually leave unchanged unless necessary

3.1.7 Package.json

Defines npm scripts and Node.js dependencies for building and deploying the website.

Type: JSON configuration file

Purpose:

- Contains build scripts for Netlify automated deployment
- Manages project metadata
- Defines commands to install Quarto and render the site

Key sections:

- `name:` - Project identifier "amore-project"
- `version:` - Project version number
- `scripts:` - Build commands
  - `prebuild` - Installs Quarto before building
  - `build` - Renders the Quarto site

3.1.8 netlify.toml

Configuration file for Netlify deployment settings.

Type: TOML configuration file

Purpose:

- Tells Netlify how to build and deploy the website
- Specifies build commands and output directory
- Sets environment variables

Key settings:

- `publish = "_site"` - Where built files are located
- `command = "npm run build"` - Build command to execute
- `NODE_VERSION = "16"` - Node.js version for building

3.1.9 netlify-setup-quarto.sh

Bash script that installs Quarto on Netlify's build servers.

Type: Shell script (`.sh`)

Purpose:

- Downloads and installs specific Quarto version
- Runs during Netlify build process
- Ensures Quarto is available for rendering

Key actions:

- Downloads Quarto v1.3.450 tarball
- Extracts and installs to `~/.local/quarto`
- Creates symbolic link to binary
- Verifies installation

Version note:

- Script uses v1.3.450 for Netlify deployment
- Local development can use newer Quarto versions
- Version mismatch did not cause issued for myself, but if rendering issues, perhaps look into this

Important:

- Called by `package.json` prebuild script
- Must be executable (`chmod +x`)

3.1.10 install-quarto.sh

Bash script that installs Quarto locally (alternative installation method).

Type: Shell script (`.sh`)

Purpose:

- Downloads and installs Quarto v1.6.40
- For local development setup
- Installs to user's home directory

Key actions:

- Downloads Quarto tarball from GitHub

- Extracts to `$HOME/.local/bin.`, a temporary cache directory created during local Quarto installation. The $HOME directory is safe to delete.

  Type: System directory
- Adds to PATH
- Verifies installation with `quarto --version`

When to use:

- Setting up development environment
- Alternative to downloading from Quarto website
- Linux/Unix systems

When to edit:

- Updating to newer Quarto version

Note:

- Different version than `netlify-setup-quarto.sh`
- Not used in deployment (local only)

3.2 .quarto/ folder

Quarto's build cache and internal processing files (auto-generated).

Type: System cache directory

Purpose:

- Stores rendering cache for faster builds
- Contains cross-reference indexes
- Holds frozen computation results
- Internal Quarto processing files

Structure:

Important notes:

- Auto-generated - recreated on each render
- Already excluded in `.gitignore`
- NB! Never edit manually
- Safe to delete if troubleshooting build issues
- Speeds up subsequent renders

When to delete:

- Fixing strange rendering errors
- Clearing cache after major changes
- Will be recreated automatically on next `quarto render`

3.3 assets/ folder

Contains all static files used across the website (images, styles, icons).

Seperated into three subfolders:

- favicorns
  - Contains all browser icons and Progressive Web App (PWA) assets.
  - Utilized in _quarto.yml file
- images
  - all images across the website are saved in this folder
- styles
  - Contains the styles.scss sheet that instructs the rendering process of design rules for the website

3.4 pages/ folder

Contains all main content pages of the website (except homepage).

Type: Content directory

Purpose:

- Houses all `.qmd` files for site pages
- Each file = one webpage
- Linked from navbar in `_quarto.yml`

Files:

- `about.qmd` - About AMORE, mission, team, steering committee
- `contact.qmd` - Contact form links
- `guidelines.qmd` - Step-by-step workflow for publishing LMAs
- `Living_meta-analysis_Directory.qmd` - Embedded Shiny app directory
- `Protocol_checklist.qmd` - Interactive checklist tool
- `Resources.qmd` - External resources and tools
- `Standardization.qmd` - AMORE framework requirements

3.5 LMAs/ folder

Contains all Living Meta-Analysis project pages.

Type: Content directory (project pages)

Purpose:

- Individual `.qmd` file for each LMA project
- Template file for creating new projects
- Fetched by Shiny app for directory listing

YAML metadata structure: Each project file contains metadata that the Shiny app reads:

Important notes:

- Shiny app reads these files via GitHub API
- Metadata must follow template structure
- Files starting with `_` are excluded from rendering

3.6 docs/ folder

Contains internal documentation and working files (not published to website).

Type: Documentation directory

Purpose:

- Working documents for website management
- Draft files and recovery backups
- Internal guides and notes

Files:

- `_AMORE_Guide.qmd` - Website management guide (this document)
- `_recoveryfile.qmd` - Backup/recovery file
- `_recoveryfile.app.qmd` - Backup/recovery file

Important notes:

- Files starting with `_` are not rendered to website
- Already excluded in `.gitignore` (for `.html` versions)
- For internal use only
- Not linked in navbar
- Safe to add more documentation files here

When to use:

- Creating internal documentation
- Draft content before moving to pages/
- Testing new features
- Reference materials for maintainers

3.7 scripts/ folder

Contains utility scripts for project setup and maintenance.

Type: Utility scripts directory

Purpose:

- Automate setup tasks
- Maintenance scripts
- Helper utilities

Files:

- `Setup.R` - Installs all required R packages and dependencies

Key functions in Setup.R:

- Installs R packages (rmarkdown, knitr, shiny, quarto, etc.)
- Installs TinyTeX for PDF generation
- Checks for existing installations before installing
- Loads required libraries

3.8 shiny-meta/ folder

Contains the Shiny application for the Living Meta-Analysis Directory.

Type: Shiny app directory

Purpose:

- Interactive searchable directory of all LMA projects
- Fetches project data from GitHub
- Filters and displays LMA metadata
- Deployed on shinyapps.io

Structure:

```
{bash}
#| label: shiny-meta directory
#| eval: false
#| echo: true

shiny-meta/
├── app.R            # Main Shiny application code
├── deploy.R         # Deployment script for shinyapps.io
├── .Renviron        # Keeps shiny-meta tokens safe
├── README.md        # Shiny app documentation
├── rsconnect/       # Deployment configuration
└── www/             # Static assets (favicon)
```

Key files:

- `app.R` - Complete UI and server logic
- `deploy.R` - Contains shinyapps.io account credentials
- `rsconnect/` - Auto-generated deployment settings
- .Renviron **Security critical file** containing:
  * 'SHINYAPPS_ACCOUNT' - Your shinyapps.io account name * 'SHINYAPPS_TOKEN' - Authentication token for deployment * 'SHINYAPPS_SECRET' - Secret key for deployment

3.8.1 How app.R works:

1. app.R fetches `.qmd` files from `LMAs/` folder via GitHub API
2. Extracts YAML metadata from each project
3. Provides filtering by outcomes, population, methodology
4. Displays paginated results with links to project pages

- Adding New Filter Categories
  - [To be expanded: Updating the Shiny app to include new metadata fields]

3.8.1.1 Deployment:

- Hosted at: https://meta-oxytocin.shinyapps.io/shiny-meta/
- Non-paid server at shinyapps.io, paid server will improve page loading
- Embedded in: `pages/Living_meta-analysis_Directory.qmd`

3.8.1.2 .Renviron

This file is personal with the tokens from the shinyapp account needed to deploy the app. All users that will deploy anew the app.R to the server on the shinyapp account meta-oxytocin must create their own .Renviron file with personal token and secret. Which is collected by login in to the account at shinyapp.io.

The meta-oxytocin account on shinyapp.io from Posit uses unpaid version, to enable multiple account users, the account subscription must be upgraded to professional plan.

Setting up deployment credentials for shinyapp:

1. Create a ‘.Renviron‘ file in the ‘shiny-meta/‘ directory: “‘bash # .Renviron file structure SHINYAPPS_ACCOUNT=your-account-name SHINYAPPS_TOKEN=your-token-here SHINYAPPS_SECRET=you secret-here “‘
2. Get your credentials from shinyapps.io: * Log in to https://www.shinyapps.io/ * Navigate to Account meta-oxytocin → Tokens * Show/Generate token * Copy the account name, token, and secret
3. In R, run ‘deploy.R‘ which will use these credentials

Important notes:

- Updates automatically when new LMA files added to GitHub
- Requires shinyapps.io account for deployment

3.9 _site/ folder

Contains the generated/rendered website files (auto-generated, do not edit).

Type: Build output directory

Purpose:

- Final HTML files ready for deployment
- Compiled CSS and JavaScript
- Copied assets and resources
- Generated during build process

Structure:

```
{bash}
#| label: _site directory
#| eval: false
#| echo: true

_site/
├── index.html          # Rendered homepage
├── search.json         # Search index
├── assets/             # Copied assets
├── pages/              # Rendered page HTML files
├── LMAs/               # Rendered LMA project pages
├── shiny-meta/         # Copied Shiny app files
└── site_libs/          # Quarto dependencies (Bootstrap, JS)
```

Important notes:

- Auto-generated - recreated on every `quarto render`
- Never edit files here - changes will be overwritten
- Already excluded from Git (in `.gitignore`)

Two separate build processes:

Local build (your computer):

- Purpose: Preview and test changes
- Command: `quarto preview` or `quarto render`
- Creates: `_site/` folder locally
- Result: For testing only, not deployed

Netlify build (production):

- Purpose: Deploy live website
- Triggered: Automatically on GitHub push
- Process: Netlify clones repo → runs `npm run build` → generates fresh `_site/`
- Result: This is what visitors see at amore-project.org

Chapter 4

Styling system SCSS

4.0.1 Overview

The AMORE website uses SCSS (Sassy CSS) for styling, which provides more power and flexibility than regular CSS. The main stylesheet is located at `assets/styles/styles.scss`. This file is automatically compiled to CSS when you render the Quarto site.

Here you find scss rules

What styles.scss controls:

- Global variables (colors, fonts)
- Typography system
- Layout components (hero sections, grids, cards)
- Navigation and footer
- Interactive elements (tabs, filters, buttons)
- Page-specific styles (index, contact, project pages)
- Responsive breakpoints (mobile, tablet, desktop). This means the layout adjusts to different screen sizes.
- Animations (gradient shifts, hover effects)

Linked in:

- `_quarto.yml` under `theme:` and `css:`
- Applied to all pages automatically

Important notes:

- Changes affect entire website
- Test thoroughly after edits and before commiting
- Uses SCSS features (variables, nesting)

- Compiled during `quarto render`

File Paths in SCSS

When referencing images or other files in your SCSS, understanding file paths is crucial:

```
{scss}
#| label: file path
#| eval: false
#| echo: true

/* CORRECT - Goes up one folder level from assets/styles/ to assets/ */
background-image: url('../images/logo.png');

/* INCORRECT - Would look in assets/styles/images/ */
background-image: url('images/logo.png');
```

4.0.2 Structure of styles.scss

The file is organized into logical sections marked with comments:

1. Global Variables - Colors, fonts, and sizing used throughout
2. Body & Animations - Background effects and page-level styling
3. Typography - All text styling (headings, paragraphs, links)
4. Components - Reusable pieces (buttons, cards, navigation)
5. Page-Specific - Styles for particular pages (contact, about, etc.)
6. Responsive Design - Adjustments for different screen sizes

- Advanced Customization
  - This section covers advanced modifications for experienced users.
- Custom CSS Modifications
  - [To be expanded: How to modify styles.scss for custom designs]

Chapter 5

Adding new projects

This section guides you through adding a new living meta-analysis project to the AMORE website. Each project requires a `.qmd` file with properly formatted metadata that allows the Shiny app directory to filter and display the project correctly.

5.1 Getting Started:

1. Locate the template: Find `_lma-template.qmd` in the LMAs folder of your project directory
2. Create your project file:
   - Duplicate the template file and rename it with a descriptive name.
   - Naming convention: Use lowercase letters with underscores, no spaces (Correct: `smith_social_cognition.qmd` | Incorrect: `Smith Social Cognition.qmd)`
3. Save location: Keep the new file in the LMAs folder alongside other project files
4. Alternatively: create new .qmd file that you save in the LMAs folder. Write scripts yourself or copy paste from this guide or template the necessary code components that you want to include.

Do NOT push to GitHub repository before the project page is finished. If you need to push your work for version control and to secure your work, add a '_' prefix to the name. Like this `_smith_social_cognition.qmd`. The prefix prevents the page from rendering correctly. The file loses access to the styles instructions from styles.scss, this means that the prefix should only be added after the preview but before pushing to Git repository.

5.1.1 What do you fill in?

1. Mandatory and Optional information
2. Fill inn the brackets [with the information requested inside the brackets] or under headers where you are explicitly told to provide information. Leave the rest of the code as it is. ### Authors or ### Timeline are headers and not comments inside the template and should not be removed.
3. Scripts: Everything under #| echo: true in the {r} code blocks should be included. If you are uncertain use the _lma-template.qmd to visualise what the script should look like.
4. When the script is created, Render to preview

5.1.1.1 Mandatory information (must be included):

- YAML header - Contains all metadata for filtering and display
- Information container - The blue panel at the top with project details (timeline, team, links)
- Abstract - Brief overview of the meta-analysis
- Resources - Links to preregistration, data, scripts, and publications

5.1.1.2 Optional information (include if relevant):

- Current Results
- Methodology Summary
- Inclusion and Exclusion Criteria
- Search Strategy
- Study Characteristics
- References

5.1.2 Format rules

```
{yaml}
#| label: YAML header format rules
#| description: This is not supposed to be inlcuded anywhere! It is just for explanatory purposes.
 ↪  Below is the YAML header as it should be inlcuded.
#| eval: false
#| echo: true

# The YAML format is very important. Incorrect indentation will result in error, or the filter system
 ↪  of the app.R (directory of projects) will be unable to correctly read the neccessary information.

## if a metadata class is irrelevant for the project there are several steps to ignore the class:
oxytocin: #example class
  route: null           # ← Won't show
  dosage: NA            # ← Won't show
  assessment_method:    # ← Empty/missing, won't show
# XinterventionX        # remove class


YAML array format: # All three options should in theory work

# List with dashes without quotes
analytical_framework:
  - Frequentist
  - Bayesian
  - Mixed methods

# List with dashes AND quotes
analytical_framework:
  - "Frequentist"
  - "Bayesian"
  - "Mixed methods"
```

```
# Inline array with brackets AND quotes
analytical_framework: ["Frequentist", "Bayesian", "Mixed methods"]

# ⍰ DO NOT MODIFY the format section at the bottom
# This section is required for proper page rendering:
format:
  html:
    page-layout: full
    toc: true
    # ... rest of format settings
```

5.2 Build the template

5.2.1 Mandatory sections

5.2.1.1 YAML header

```
{yaml}
#| label: YAML header
#| description: This should be at the very beginning of the project file. The filters options are
 ↪  matched with app.R. If you want to include new options this must also be added to the app's filter
 ↪  logic.# When research groups propose new projects through nettskjema "Propose your project", they
 ↪  are instructed to VOLUNTARY choose metadata options for their project to increase their
 ↪  visibility. If they choose not to answer this part of the form leave the sections blank or write
 ↪  null.
#| eval: false
#| echo: true



---
title: "Write title here"
analytical_framework: # Choose one or more
  - "Frequentist"
  - "Bayesian"
  - "Mixed methods"
oxytocin:
  intervention: # if applicable, choose one or more:
    - "Intranasal oxytocin administration"
    - "Oral oxytocin administration"
    - "Intravenous/injection oxytocin administration"
    - "Environmental/behavioral oxytocin manipulation"
    - "Perinatal oxytocin exposure"
  assessment_method: # if applicable, choose one or more:
    - "Behavioral assessment"
    - "Physiological response"
    - "Biological sample collection"
    - "Genetic studies"
    - "Neural/imaging measurement"
  route: # Choose one or more:# if applicable, choose one or more:
    - "Central"
    - "Peripheral"
    - "Various administration routes"
    - "Administration method unspecified"
  dosage: # if applicable, choose one or more:
      - "8 IU"
      - "16 IU"
      - "24 IU"
      - "32 IU"
      - "40 IU"
      - "Variable dosage"
population:
  status: # if applicable, choose one or more:
    - "Healthy"
    - "Clinical"
    - "Mixed"
  age_group: # if applicable, choose one or more:
    - "Children"
    - "Adolescents"
    - "Adults"
    - "Older Adults"
    - "Mixed Age Groups"
  clinical_type: # Describe if applicable (e.g., "Autism Spectrum Disorder", "Depression")
outcomes:
  biological: # if applicable, choose one or more:
    - "Cardiovascular"
    - "Neuroendocrine"
    - "Neurological"
    - "Metabolic"
    - "Immune and Inflammatory"
    - "Pain and Sensory"
    - "Sleep and Circadian"
  psychological_behavioral: # if applicable, choose one or more:
    - "Mood and Emotion"
    - "Cognition and Memory"
    - "Stress and Coping"
    - "Eating and Appetite"
    - "Risk and Decision-Making"
    - "Sleep Behavior and Quality"
    - "Bonding and Attachment"
    - "Trust and Cooperation"
    - "Communication and Empathy"
    - "Aggression and Conflict"
  clinical: # if applicable, choose one or more:
    - "Neurodevelopmental"
    - "Mood Disorders"
    - "Psychotic Disorders"
    - "Addiction and Substance Use"
    - "Eating Disorders"
update-frequency: # Choose one:
  - "Every 6 months"
  - "Every 12 months"
  - "Every 18 months"
  - "Every 24 months"
dois:
  preregistration: "10.XXXX/template-prereg"
  preprint: "10.XXXX/template-preprint"
  publication: "10.XXXX/template-publication"
format: ### NB!  This is important to include ##
  html: ### NB!  This is important to include ##
    page-layout: full ### NB!  This is important to include ##
    toc: true ### NB!  This is important to include ##
    toc-title: "On this page" ### NB!  This is important to include ##
    toc-location: right ### NB!  This is important to include ##
    toc-depth: 3 ### NB!  This is important to include ##
---
```

5.2.1.2 Information container

```
{r}
#| label: information container
#| description: This code chunk is the title and the information box visible under the project hero
 ↪  title. Insert the required information inside the [ ] brackets
#| eval: false
#| echo: true


::: {.project-hero-section}
::: {.project-hero-content}
::: {.project-hero-title}

Write title here

:::
:::
:::

::: {.project-info-panel}
::: {.project-info-grid}

::: {.info-card .authors-card}
### Authors
<div id="author-list">
  <div id="visible-authors">
    <strong>[First Author Name]</strong> ([Affiliation])<br> ## Write authors and
    <strong>[Second Author Name]</strong> ([Affiliation])<br> # affiliations inside
    <strong>[Senior Author Name]</strong> ([Affiliation]) ##### the brackets []
  </div>

  <div id="hidden-authors" style="display:none">
    <strong>[Fourth Author]</strong> ([Affiliation])<br>
    <strong>[Fifth Author]</strong> ([Affiliation])<br>
    <strong>[Sixth Author]</strong> ([Affiliation])<br>
    <strong>[Seventh Author]</strong> ([Affiliation])<br>
    <strong>[Eighth Author]</strong> ([Multiple Affiliations | Separated by Pipes])<br>
    <strong>[Additional Authors as needed...]</strong><br>
  </div>

  <a href="#" id="author-toggle" class="toggle-link" onclick="toggleAuthors(); return false;">+ Show
   ↪  more authors</a>

  <script>
  function toggleAuthors() {
    var hiddenAuthors = document.getElementById("hidden-authors");
    var toggleButton = document.getElementById("author-toggle");

    if (hiddenAuthors.style.display === "none") {
      hiddenAuthors.style.display = "block";
      toggleButton.textContent = "- Show fewer authors";
    } else {
      hiddenAuthors.style.display = "none";
      toggleButton.textContent = "+ Show more authors";
    }
  }
  </script>
</div>
:::

::: {.info-card .timeline-card} ## Fill information inside the [] brackets
### Timeline
**Pre-registered:** [Month DD, YYYY]
**Last update:** [Status of current analysis]
**Next update:** [Planned date or "To be determined" (XX-month update cycle)]
:::

::: {.info-card .identifiers-card} ## Fill information inside the [] brackets
### Identifiers
**DOI:** [https://doi.org/ACTUAL_DOI](https://doi.org/ACTUAL_DOI)

**Keywords:** [keyword1], [keyword2], [keyword3], [methodology], [population]
:::

::: {.info-card .contact-card} ## Fill information inside the [] brackets
### Contact
**Principal Investigator:** [PI Full Name]
**Email:** [pi.email@institution.edu](mailto:pi.email@institution.edu)
:::

:::
:::
```

5.2.1.3 Abstract

```
{r}
#| label: Abstract section of main content area
#| description: This is where you fill in the abstract. It should be the first header after the
 ↪ information container box.
#| eval: false
#| echo: true

## Abstract {#abstract}

write abstract under abstract header
```

5.2.1.4 Resources

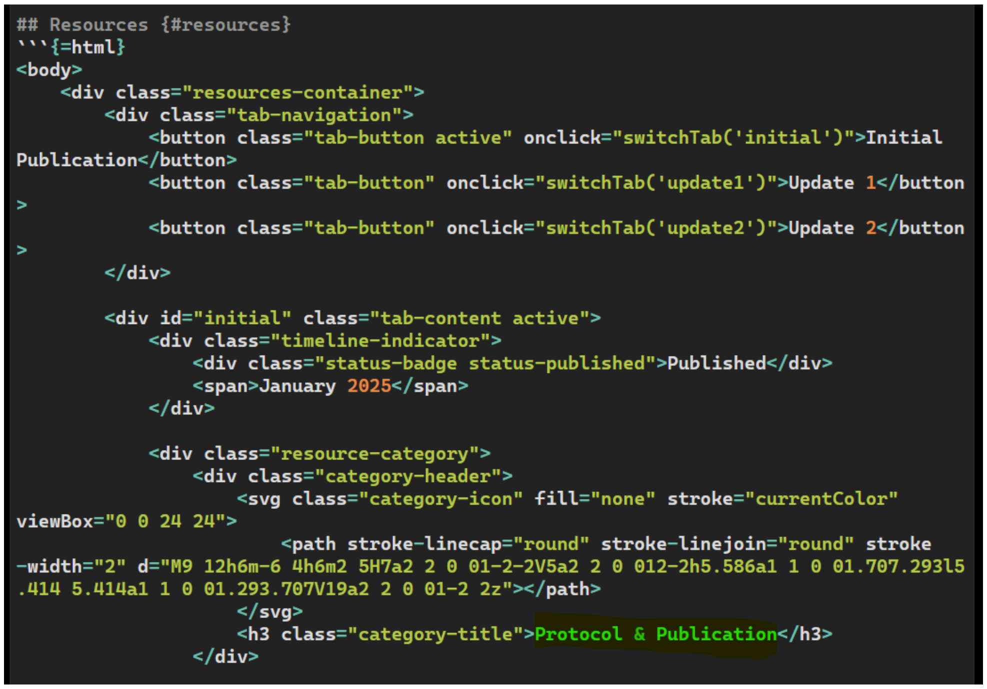
```
{.html}
#| label: Resources section of main content area.
#| description: This code chunk is the resources section. The resources section is written as HTML
 ↪ code, with JavaScript for tab functions.
#| eval: false
#| echo: true

## Resources {#resources}
<body>
    <div class="resources-container">

        <!-- TAB NAVIGATION BUTTONS -->
        <div class="tab-navigation">
            <button class="tab-button active" onclick="switchTab('initial')">Initial
             ↪  Publication</button>
            <button class="tab-button" onclick="switchTab('update1')">Update 1</button>
            <button class="tab-button" onclick="switchTab('update2')">Update 2</button>
        </div>


        <!-- INITIAL PUBLICATION TAB -->
        <div id="initial" class="tab-content active">

            <!-- Timeline indicator for initial publication -->
            <div class="timeline-indicator">
                <div class="status-badge status-published">Published</div>
                <span>[Month Year]</span>
            </div>


            <!-- PROTOCOL & PUBLICATION CATEGORY -->
            <div class="resource-category">
                <div class="category-header">
                    <svg class="category-icon" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                        <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M9
                         ↪  12h6m-6 4h6m2 5H7a2 2 0 01-2-2V5a2 2 0 012-2h5.586a1 1 0 01.707.293l5.414
                         ↪  5.414a1 1 0 01.293.707V19a2 2 0 01-2 2z"></path>
                    </svg>
                    <h3 class="category-title">Protocol & Publication</h3>
                </div>

                <!-- Study Protocol Item -->
                <div class="resource-item">
                    <div class="resource-header">
                        <div class="resource-info">
                            <div class="resource-title">Study Protocol</div>
                            <div class="resource-description">[Write short description of study
                             ↪  protocol, e.g., "Preregistered analysis plan outlining inclusion
                             ↪  criteria and analytical approach"]</div>
                            <div class="resource-links">
                              <a href="[INSERT_FULL_URL_HERE]" class="resource-link" target="_blank">
                                    <div class="link-title">[Link name, e.g., "Preregistration on
                                     ↪  OSF"]</div>
                                    <div class="link-url">[Clean URL, e.g., "osf.io/8fzdy"]</div>
                                </a>
                            </div>
                        </div>
                    </div>
                </div>

                <!-- Published Article Item -->
                <div class="resource-item">
                    <div class="resource-header">
                    <div class="resource-info">
                        <div class="resource-title">Published Article</div>
                       <div class="resource-description">[Write description of published article,
                         ↪  e.g., "Peer-reviewed article published in Molecular Psychiatry"]</div>
                        <div class="resource-links">
                            <a href="[INSERT_FULL_DOI_URL_HERE]" class="resource-link"
                             ↪  target="_blank">
                                <div class="link-title">[Link name, e.g., "Molecular Psychiatry
                                 ↪  Article"]</div>
                             <div class="link-doi">[DOI, e.g., "10.1038/s41380-024-02871-4"]</div>
                            </a>
                        </div>
                    </div>
                </div>
            </div>

            <!-- Preprint Version Item -->
            <div class="resource-item">
                <div class="resource-header">
                    <div class="resource-info">
                        <div class="resource-title">Preprint Version</div>
                        <div class="resource-description">[Write description, e.g., "Open access
                         ↪  preprint with full supplementary materials"]</div>
                        <div class="resource-links">
                           <a href="[INSERT_FULL_URL_HERE]" class="resource-link" target="_blank">
                                <div class="link-title">[Link name, e.g., "OSF Preprint"]</div>
                                <div class="link-url">[Clean URL, e.g., "osf.io/hjyfjygg"]</div>
                            </a>
                        </div>
                    </div>
                </div>
            </div>
        </div>


        <!-- DATA & ANALYSIS CATEGORY -->
        <div class="resource-category">
            <div class="category-header">
                <svg class="category-icon" fill="none" stroke="currentColor" viewBox="0 0 24 24">
                    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M4
                     ↪  7v10c0 2.21 3.589 4 8 4s8-1.79 8-4V7M4 7c0 2.21 3.589 4 8 4s8-1.79 8-4M4
                     ↪  7c0-2.21 3.589-4 8-4s8 1.79 8 4"></path>
                </svg>
                <h3 class="category-title">Data & Analysis</h3>
            </div>

            <!-- Complete Dataset Item -->
            <div class="resource-item">
                <div class="resource-header">
                    <div class="resource-info">
                        <div class="resource-title">Complete Dataset</div>
                        <div class="resource-description">[Short description of data, e.g., "Raw
                         ↪  and cleaned data files with codebook"]</div>
                        <div class="resource-links">
                           <a href="[INSERT_FULL_URL_HERE]" class="resource-link" target="_blank">
                            <div class="link-title">[Link name, e.g., "OSF Data Repository"]</div>
                                <div class="link-url">[Clean URL, e.g., "osf.io/6hsjfg"]</div>
                            </a>
```

```html
                    </div>
                </div>
            </div>
        </div>

        <!-- Analysis Scripts Item -->
        <div class="resource-item">
            <div class="resource-header">
                <div class="resource-info">
                    <div class="resource-title">Analysis Scripts</div>
                  <div class="resource-description">[Description of scripts, e.g., "R scripts
                    ↪  for all meta-analytic models and sensitivity analyses"]</div>
                    <div class="resource-links">
                      <a href="[INSERT_FULL_URL_HERE]" class="resource-link" target="_blank">
                        <div class="link-title">[Link name, e.g., "R Analysis Scripts"]</div>
                           <div class="link-url">[Clean URL or DOI]</div>
                      </a>
                    </div>
                </div>
            </div>
        </div>
    </div>


    <!-- RELATED RESEARCH CATEGORY -->
    <div class="resource-category">
        <div class="category-header">
            <svg class="category-icon" fill="none" stroke="currentColor" viewBox="0 0 24 24">
              <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M13.828
                 ↪  10.172a4 4 0 00-5.656 0l-4 4a4 4 0 105.656 5.656l1.102-1.101m-.758-4.899a4
                 ↪  4 0 005.656 0l4-4a4 4 0 00-5.656-5.656l-1.1 1.1"></path>
            </svg>
            <h3 class="category-title">Related Research</h3>
        </div>

        <!-- Related Research Item -->
        <div class="resource-item">
            <div class="resource-header">
                <div class="resource-info">
                    <div class="resource-title">Theoretical Background</div>
                    <div class="resource-description">[Description, e.g., "Key theoretical
                    ↪  papers informing this meta-analysis"]</div>
                    <div class="resource-links">
                      <a href="[INSERT_FULL_URL_HERE]" class="resource-link" target="_blank">
                           <div class="link-title">[Paper title or description]</div>
                           <div class="link-doi">[DOI, e.g., "10.1234/example"]</div>
                      </a>
                      <a href="[INSERT_FULL_URL_HERE]" class="resource-link" target="_blank">
                           <div class="link-title">[Second paper title or description]</div>
                           <div class="link-doi">[DOI, e.g., "10.5678/example"]</div>
                      </a>
                    </div>
                </div>
            </div>
        </div>
    </div>

</div>
```

```
        <!-- UPDATE 1 TAB -->
        <div id="update1" class="tab-content">

            <!-- Coming soon notice -->
            <div class="coming-soon-notice">
                <strong>Update 1 Coming Soon</strong> • Expected: [Month Year] (updates every [XX]
                 ↪  months)
            </div>

            <!-- Timeline indicator for update 1 -->
            <div class="timeline-indicator">
                <div class="status-badge status-coming-soon">In Development</div>
                <span>Next update within 24 months</span>
            </div>
        </div>


        <!-- UPDATE 2 TAB -->
        <div id="update2" class="tab-content">

            <!-- Future notice -->
            <div class="future-notice">
                <strong>Future Update</strong> • Expected: [Month Year]
            </div>

            <!-- Timeline indicator for update 2 -->
            <div class="timeline-indicator">
                <div class="status-badge status-future">Future</div>
                <span>Long-term update planning</span>
            </div>
        </div>


    </div>


    <!-- JAVASCRIPT FOR TAB SWITCHING -->
    <script>
        function switchTab(tabId) {
            // Hide all tab contents
            const contents = document.querySelectorAll('.tab-content');
            contents.forEach(content => content.classList.remove('active'));

            // Remove active class from all buttons
            const buttons = document.querySelectorAll('.tab-button');
            buttons.forEach(button => button.classList.remove('active'));

            // Show selected tab content
            document.getElementById(tabId).classList.add('active');

            // Add active class to clicked button
            event.target.classList.add('active');
        }
    </script>

</body>
```

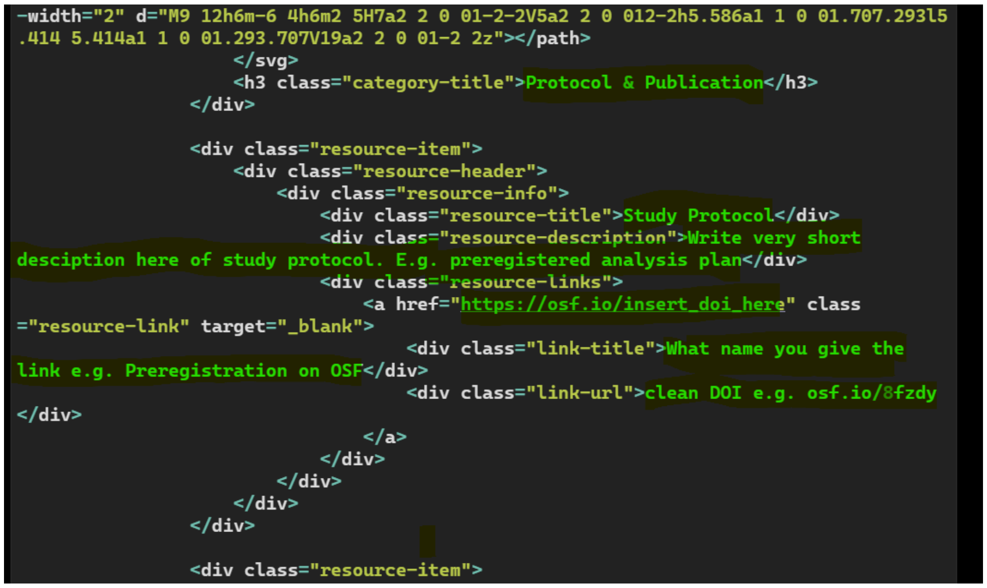
```
</html>
```

Screenshots from the template of the resources html section: The neon green text is where you can or need to customize the html code.



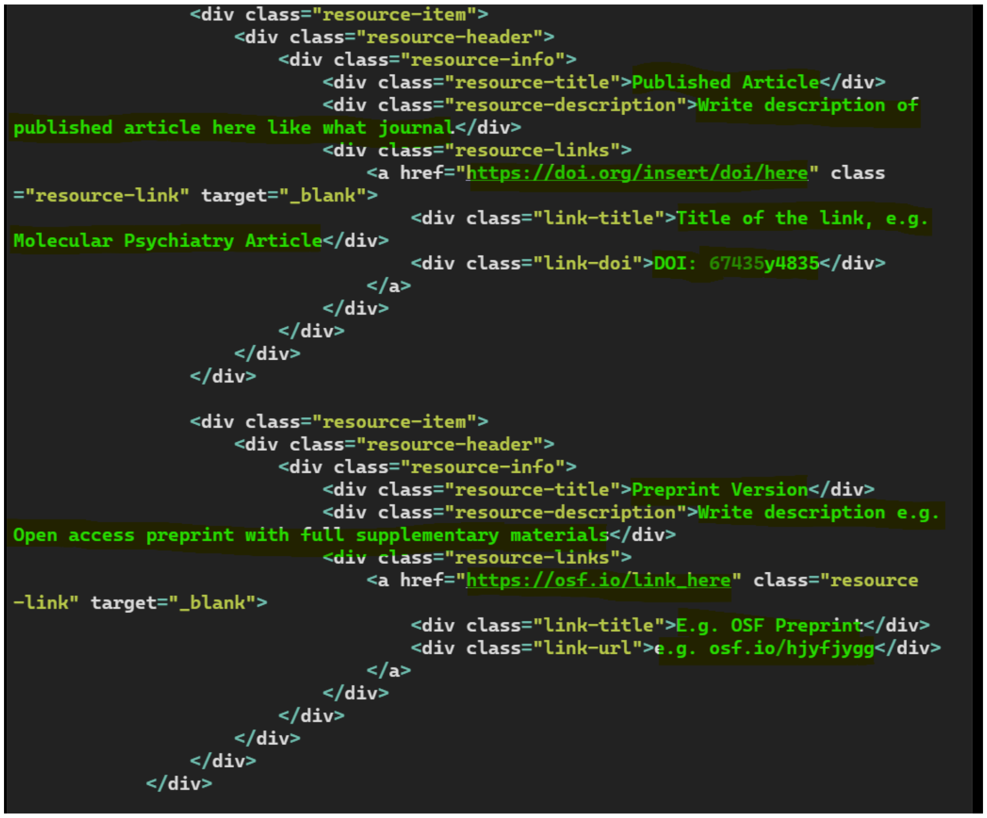
~~~
## Resources {#resources}
```{=html}
<body>
    <div class="resources-container">
        <div class="tab-navigation">
            <button class="tab-button active" onclick="switchTab('initial')">Initial
Publication</button>
            <button class="tab-button" onclick="switchTab('update1')">Update 1</button
>
            <button class="tab-button" onclick="switchTab('update2')">Update 2</button
>
        </div>

        <div id="initial" class="tab-content active">
            <div class="timeline-indicator">
                <div class="status-badge status-published">Published</div>
                <span>January 2025</span>
            </div>

            <div class="resource-category">
                <div class="category-header">
                    <svg class="category-icon" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                        <path stroke-linecap="round" stroke-linejoin="round" stroke
-width="2" d="M9 12h6m-6 4h6m2 5H7a2 2 0 01-2-2V5a2 2 0 012-2h5.586a1 1 0 01.707.293l5
.414 5.414a1 1 0 01.293.707V19a2 2 0 01-2 2z"></path>
                    </svg>
                    <h3 class="category-title">Protocol & Publication</h3>
                </div>
~~~

Figure 5.1: screenshot of resources section. Neongreeen font is where you need to customize the code.



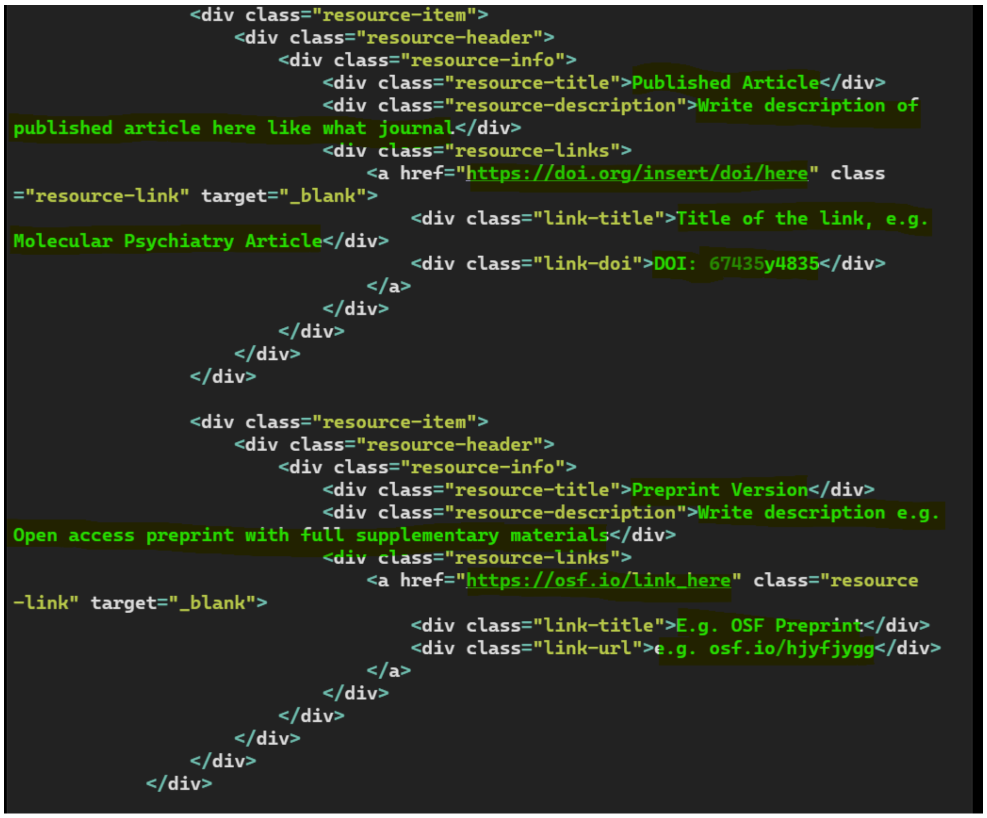
```
-width="2" d="M9 12h6m-6 4h6m2 5H7a2 2 0 01-2-2V5a2 2 0 012-2h5.586a1 1 0 01.707.293l5
.414 5.414a1 1 0 01.293.707V19a2 2 0 01-2 2z"></path>
                    </svg>
                    <h3 class="category-title">Protocol & Publication</h3>
                </div>

                <div class="resource-item">
                    <div class="resource-header">
                        <div class="resource-info">
                            <div class="resource-title">Study Protocol</div>
                            <div class="resource-description">Write very short
desciption here of study protocol. E.g. preregistered analysis plan</div>
                            <div class="resource-links">
                                <a href="https://osf.io/insert_doi_here" class
="resource-link" target="_blank">
                                    <div class="link-title">What name you give the
link e.g. Preregistration on OSF</div>
                                    <div class="link-url">clean DOI e.g. osf.io/8fzdy
</div>
                                </a>
                            </div>
                        </div>
                    </div>
                </div>

                <div class="resource-item">
```

Figure 5.2: screenshot of resources section. Neongreeen font is where you need to customize the code.



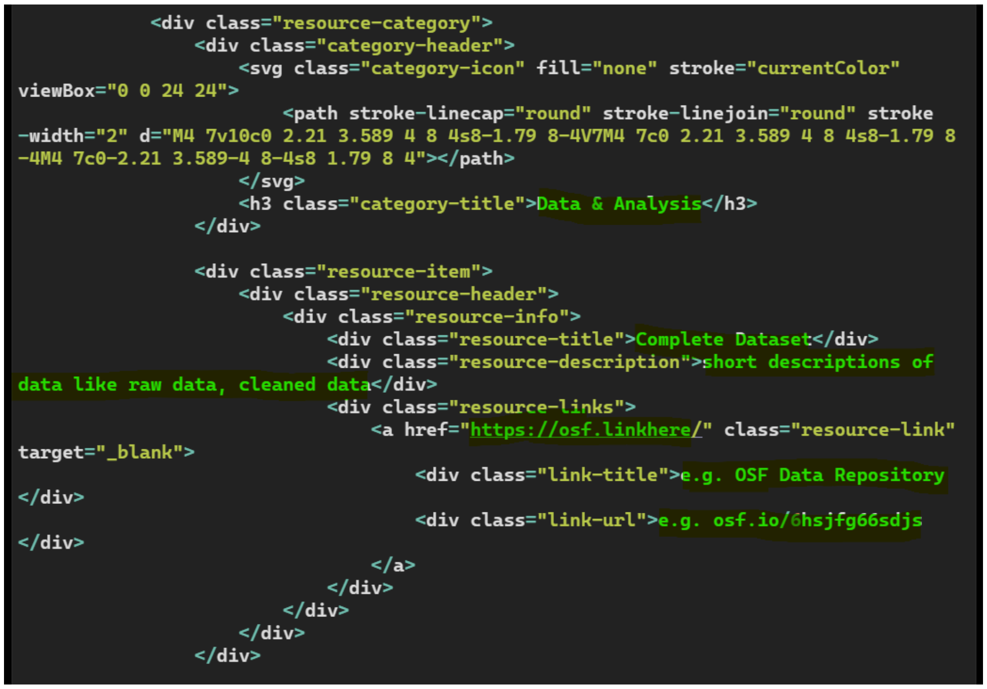
```
                <div class="resource-item">
                    <div class="resource-header">
                        <div class="resource-info">
                            <div class="resource-title">Published Article</div>
                            <div class="resource-description">Write description of
published article here like what journal</div>
                            <div class="resource-links">
                                <a href="https://doi.org/insert/doi/here" class
="resource-link" target="_blank">
                                    <div class="link-title">Title of the link, e.g.
Molecular Psychiatry Article</div>
                                    <div class="link-doi">DOI: 67435y4835</div>
                                </a>
                            </div>
                        </div>
                    </div>
                </div>

                <div class="resource-item">
                    <div class="resource-header">
                        <div class="resource-info">
                            <div class="resource-title">Preprint Version</div>
                            <div class="resource-description">Write description e.g.
Open access preprint with full supplementary materials</div>
                            <div class="resource-links">
                                <a href="https://osf.io/link_here" class="resource
-link" target="_blank">
                                    <div class="link-title">E.g. OSF Preprint</div>
                                    <div class="link-url">e.g. osf.io/hjyfjygg</div>
                                </a>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
```

Figure 5.3: screenshot of resources section. Neongreeen font is where you need to customize the code.



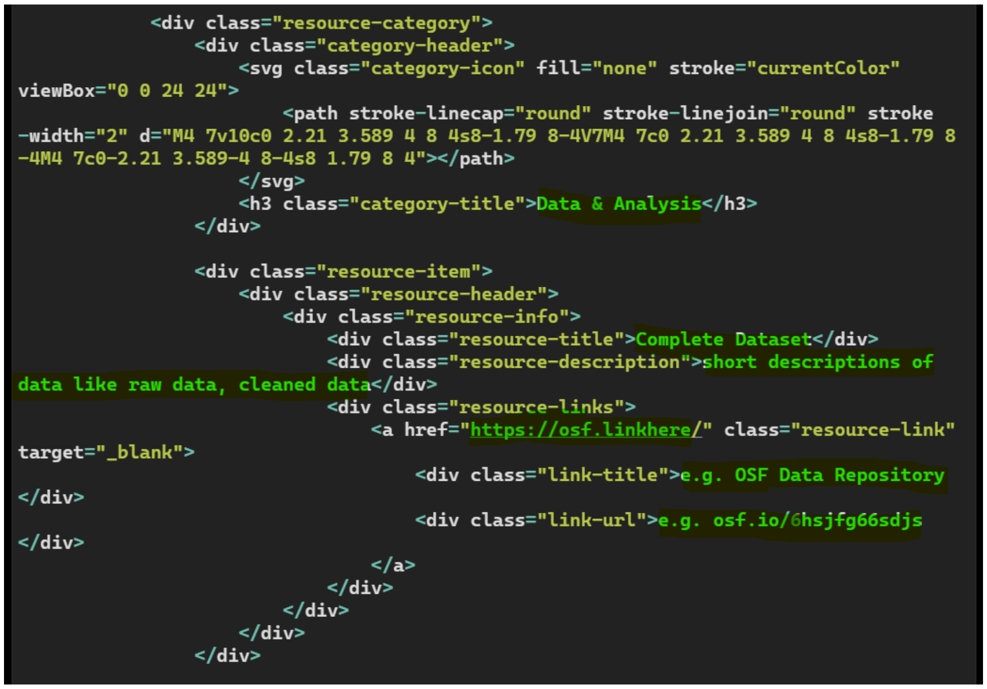
```
                <div class="resource-item">
                    <div class="resource-header">
                        <div class="resource-info">
                            <div class="resource-title">Published Article</div>
                            <div class="resource-description">Write description of
published article here like what journal</div>
                            <div class="resource-links">
                                <a href="https://doi.org/insert/doi/here" class
="resource-link" target="_blank">
                                    <div class="link-title">Title of the link, e.g.
Molecular Psychiatry Article</div>
                                    <div class="link-doi">DOI: 67435y4835</div>
                                </a>
                            </div>
                        </div>
                    </div>
                </div>

                <div class="resource-item">
                    <div class="resource-header">
                        <div class="resource-info">
                            <div class="resource-title">Preprint Version</div>
                            <div class="resource-description">Write description e.g.
Open access preprint with full supplementary materials</div>
                            <div class="resource-links">
                                <a href="https://osf.io/link_here" class="resource
-link" target="_blank">
                                    <div class="link-title">E.g. OSF Preprint</div>
                                    <div class="link-url">e.g. osf.io/hjyfjygg</div>
                                </a>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
```

Figure 5.4: screenshot of resources section. Neongreeen font is where you need to customize the code.



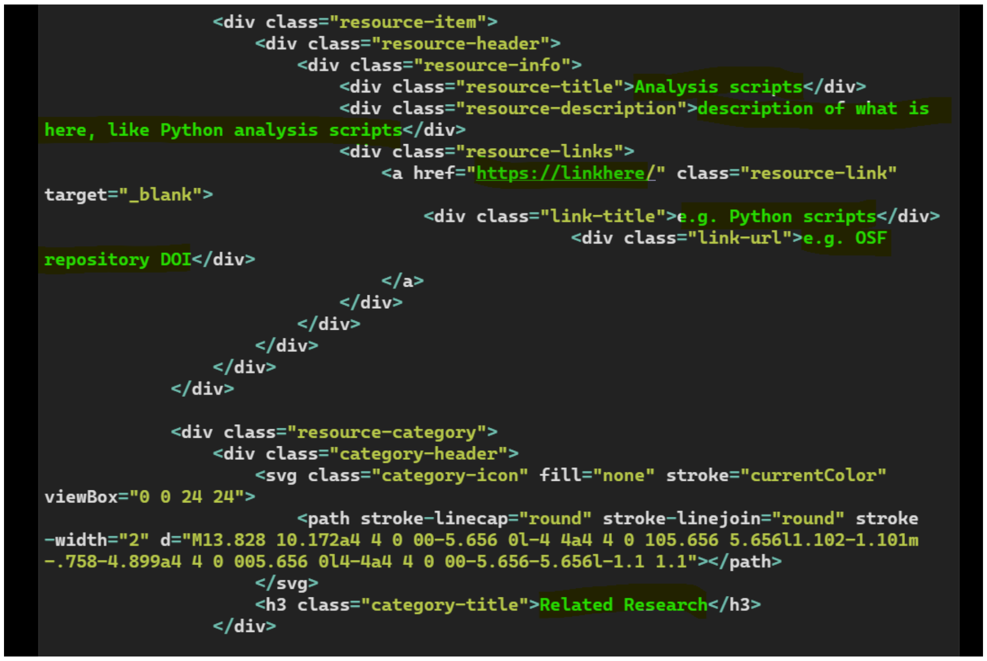
```
            <div class="resource-category">
                <div class="category-header">
                    <svg class="category-icon" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                        <path stroke-linecap="round" stroke-linejoin="round" stroke
-width="2" d="M4 7v10c0 2.21 3.589 4 8 4s8-1.79 8-4V7M4 7c0 2.21 3.589 4 8 4s8-1.79 8
-4M4 7c0-2.21 3.589-4 8-4s8 1.79 8 4"></path>
                    </svg>
                    <h3 class="category-title">Data & Analysis</h3>
                </div>

                <div class="resource-item">
                    <div class="resource-header">
                        <div class="resource-info">
                            <div class="resource-title">Complete Dataset</div>
                            <div class="resource-description">short descriptions of
data like raw data, cleaned data</div>
                            <div class="resource-links">
                                <a href="https://osf.linkhere/" class="resource-link"
target="_blank">
                                    <div class="link-title">e.g. OSF Data Repository
</div>
                                    <div class="link-url">e.g. osf.io/6hsjfg66sdjs
</div>
                                </a>
                            </div>
                        </div>
                    </div>
                </div>
```

Figure 5.5: screenshot of resources section. Neongreeen font is where you need to customize the code.



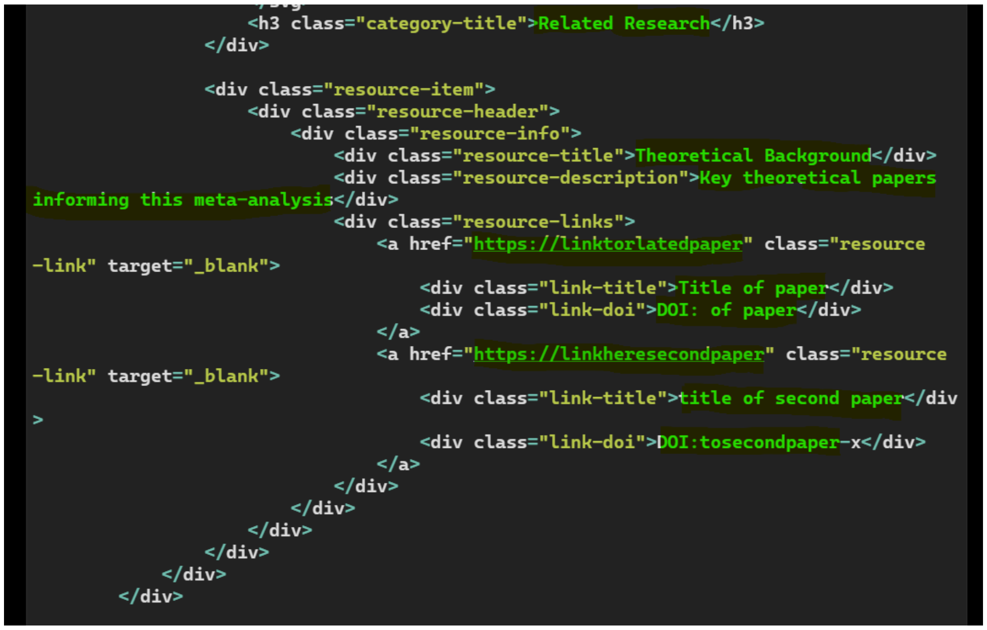
```
            <div class="resource-category">
                <div class="category-header">
                    <svg class="category-icon" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                        <path stroke-linecap="round" stroke-linejoin="round" stroke
-width="2" d="M4 7v10c0 2.21 3.589 4 8 4s8-1.79 8-4V7M4 7c0 2.21 3.589 4 8 4s8-1.79 8
-4M4 7c0-2.21 3.589-4 8-4s8 1.79 8 4"></path>
                    </svg>
                    <h3 class="category-title">Data & Analysis</h3>
                </div>

                <div class="resource-item">
                    <div class="resource-header">
                        <div class="resource-info">
                            <div class="resource-title">Complete Dataset</div>
                            <div class="resource-description">short descriptions of
data like raw data, cleaned data</div>
                            <div class="resource-links">
                                <a href="https://osf.linkhere/" class="resource-link"
target="_blank">
                                    <div class="link-title">e.g. OSF Data Repository
</div>
                                    <div class="link-url">e.g. osf.io/6hsjfg66sdjs
</div>
                                </a>
                            </div>
                        </div>
                    </div>
                </div>
```

Figure 5.6: screenshot of resources section. Neongreeen font is where you need to customize the code.



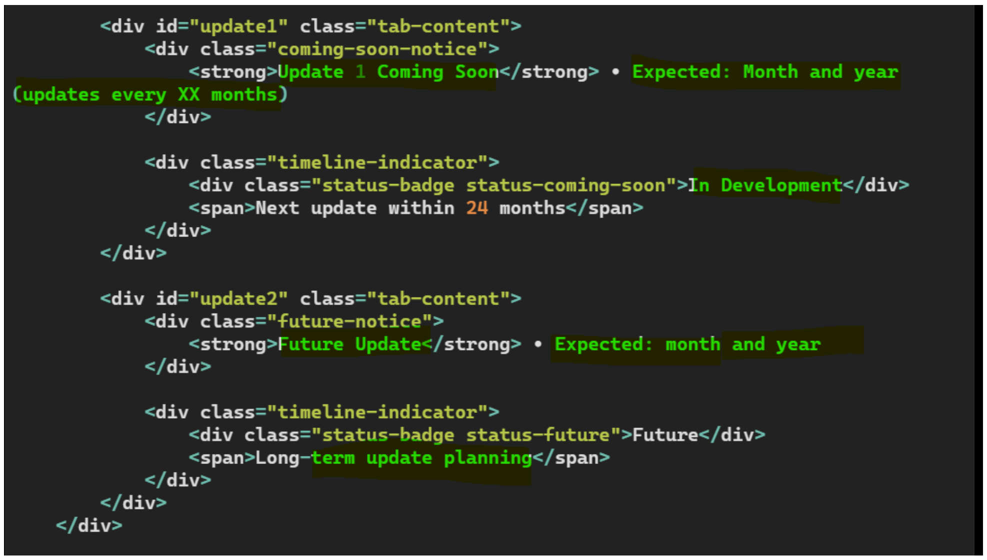
```
                <div class="resource-item">
                    <div class="resource-header">
                        <div class="resource-info">
                            <div class="resource-title">Analysis scripts</div>
                            <div class="resource-description">description of what is
here, like Python analysis scripts</div>
                            <div class="resource-links">
                                <a href="https://linkhere/" class="resource-link"
target="_blank">
                                    <div class="link-title">e.g. Python scripts</div>
                                                  <div class="link-url">e.g. OSF
repository DOI</div>
                                </a>
                            </div>
                        </div>
                    </div>
                </div>
            </div>

            <div class="resource-category">
                <div class="category-header">
                    <svg class="category-icon" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
                        <path stroke-linecap="round" stroke-linejoin="round" stroke
-width="2" d="M13.828 10.172a4 4 0 00-5.656 0l-4 4a4 4 0 105.656 5.656l1.102-1.101m
-.758-4.899a4 4 0 005.656 0l4-4a4 4 0 00-5.656-5.656l-1.1 1.1"></path>
                    </svg>
                    <h3 class="category-title">Related Research</h3>
                </div>
```

Figure 5.7: screenshot of resources section. Neongreeen font is where you need to customize the code.



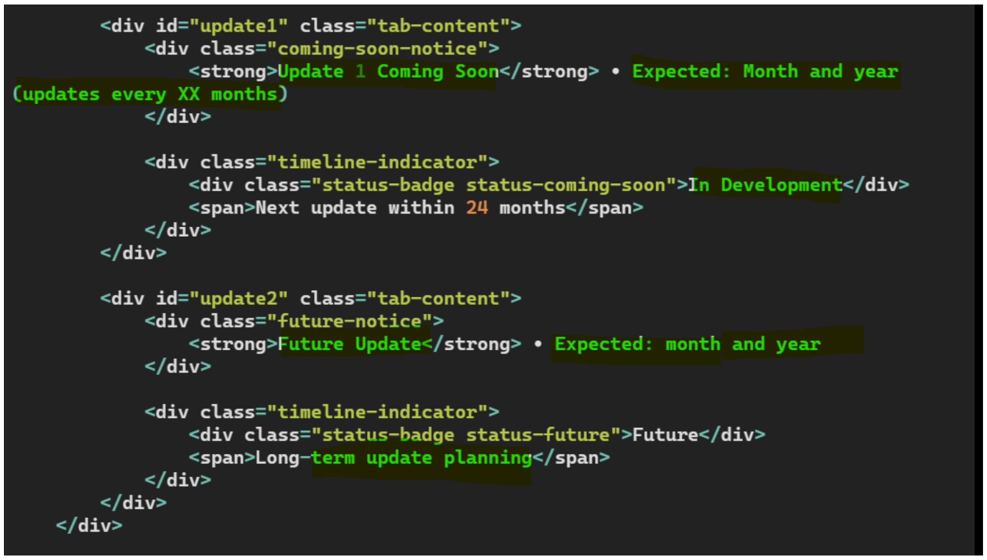
```
                    <h3 class="category-title">Related Research</h3>
                </div>

                <div class="resource-item">
                    <div class="resource-header">
                        <div class="resource-info">
                            <div class="resource-title">Theoretical Background</div>
                            <div class="resource-description">Key theoretical papers
informing this meta-analysis</div>
                            <div class="resource-links">
                                <a href="https://linktorlatedpaper" class="resource
-link" target="_blank">
                                    <div class="link-title">Title of paper</div>
                                    <div class="link-doi">DOI: of paper</div>
                                </a>
                                <a href="https://linkheresecondpaper" class="resource
-link" target="_blank">
                                    <div class="link-title">title of second paper</div
>
                                    <div class="link-doi">DOI:tosecondpaper-x</div>
                                </a>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
```

Figure 5.8: screenshot of resources section. Neongreeen font is where you need to customize the code.



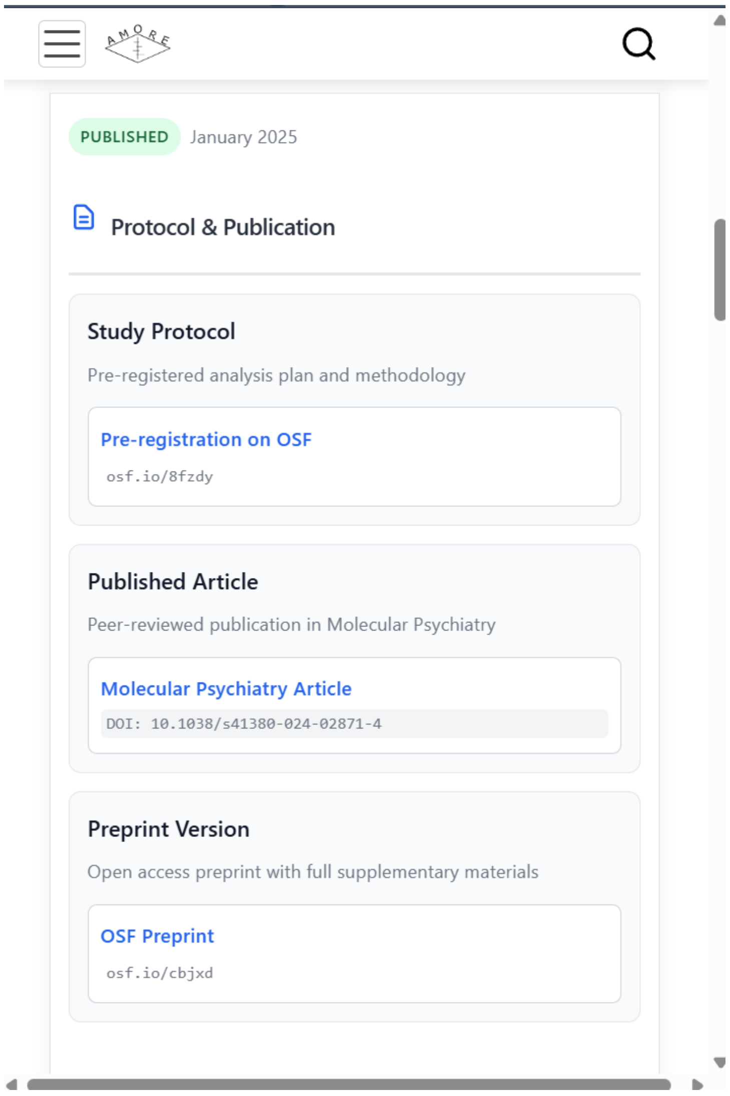
```
        <div id="update1" class="tab-content">
            <div class="coming-soon-notice">
                <strong>Update 1 Coming Soon</strong> • Expected: Month and year
(updates every XX months)
            </div>

            <div class="timeline-indicator">
                <div class="status-badge status-coming-soon">In Development</div>
                <span>Next update within 24 months</span>
            </div>
        </div>

        <div id="update2" class="tab-content">
            <div class="future-notice">
                <strong>Future Update</strong> • Expected: month and year
            </div>

            <div class="timeline-indicator">
                <div class="status-badge status-future">Future</div>
                <span>Long-term update planning</span>
            </div>
        </div>
    </div>
```

Figure 5.9: screenshot of resources section. Neongreeen font is where you need to customize the code.



```html
        <div id="update1" class="tab-content">
            <div class="coming-soon-notice">
                <strong>Update 1 Coming Soon</strong> • Expected: Month and year
(updates every XX months)
            </div>

            <div class="timeline-indicator">
                <div class="status-badge status-coming-soon">In Development</div>
                <span>Next update within 24 months</span>
            </div>
        </div>

        <div id="update2" class="tab-content">
            <div class="future-notice">
                <strong>Future Update</strong> • Expected: month and year
            </div>

            <div class="timeline-indicator">
                <div class="status-badge status-future">Future</div>
                <span>Long-term update planning</span>
            </div>
        </div>
    </div>
```

Figure 5.10: screenshot of resources section. Neongreeen font is where you need to customize the code.



PUBLISHED January 2025

Protocol & Publication

Study Protocol

Pre-registered analysis plan and methodology

Pre-registration on OSF

osf.io/8fzdy

Published Article

Peer-reviewed publication in Molecular Psychiatry

Molecular Psychiatry Article

DOI: 10.1038/s41380-024-02871-4

Preprint Version

Open access preprint with full supplementary materials

OSF Preprint

osf.io/cbjxd

Figure 5.11: Visual of resources section

5.2.2 Optional sections

5.2.2.1 Current results

```
{.html}
#| label: Current results section of main content area
#| description:  HTML code for the current results section for project pages. Pre-built HTML code
 ↪  chunks to mix and match. Different containers to show significant negative or positive findings or
 ↪  null findings. Needs to be wrapped in the div class "results-container"
#| eval: false
#| echo: true


## Current Results {#current-results}
<body>
    <div class="results-container">
```

```
<!-- OVERALL FINDINGS OPTIONS -->


<!-- Option A: No Overall Effect -->
<div class="overall-findings" style="background: #f0f9ff; border-left: 4px solid #3b82f6;">
    <p>The meta-analysis found <strong>no overall significant effect</strong> of oxytocin
     ↪  administration on [OUTCOME] (Hedges' g = [VALUE], 95% CI [[LOWER], [UPPER]], p =
     ↪  [P-VALUE]).
</div>



<!-- Option B: Significant Positive Effect -->
<div class="overall-findings" style="background: #f0fdf4; border-left: 4px solid #10b981;">
    <p>The meta-analysis found a <strong>significant positive effect</strong> of oxytocin
     ↪  administration on [OUTCOME] (Hedges' g = [VALUE], 95% CI [[LOWER], [UPPER]], p =
     ↪  [P-VALUE]).</p>
    <p>This effect was [consistent across domains / varied by subdomain], with [ADDITIONAL
     ↪  CONTEXT].</p>
</div>



<!-- Option C: Significant Negative Effect -->
<div class="overall-findings" style="background: #fef2f2; border-left: 4px solid #ef4444;">
    <p>The meta-analysis found a <strong>significant negative effect</strong> of oxytocin
     ↪  administration on [OUTCOME] (Hedges' g = [VALUE], 95% CI [[LOWER], [UPPER]], p =
     ↪  [P-VALUE]).</p>
    <p>This effect suggests that oxytocin [INTERPRETATION OF NEGATIVE EFFECT].</p>
</div>



<!-- Option D: Mixed/Heterogeneous Effects -->
<div class="overall-findings" style="background: #fffbeb; border-left: 4px solid #f59e0b;">
    <p>The meta-analysis revealed <strong>substantial heterogeneity</strong> in oxytocin's
     ↪  effects on [OUTCOME] (Hedges' g = [VALUE], 95% CI [[LOWER], [UPPER]], I² =
     ↪  [VALUE]%).</p>
    <p>Effects varied significantly by [MODERATOR], suggesting that oxytocin's impact is
     ↪  highly context-dependent.</p>
</div>



<!-- KEY FINDINGS SECTION WRAPPER -->
<div class="key-findings-section">
    <h3 class="key-findings-title">Key Findings by Domain</h3>

    <div class="findings-grid">

        <!-- FINDING CARD OPTIONS -->



        <!-- Card Type A: Significant Positive Effect -->
        <div class="finding-card significant" style="background: #f0fdf4;">
            <div class="finding-domain">
```

```
                [DOMAIN NAME]
                <span class="effect-size-badge positive-effect">Significant Effect</span>
            </div>
            <div class="finding-effect">Hedges' g = [VALUE]</div>
            <div class="finding-significance">p = [P-VALUE] • Statistically significant
             ↪  positive effect</div>
        </div>



        <!-- Card Type B: No Significant Effect -->
        <div class="finding-card non-significant" style="background: #f9fafb;">
            <div class="finding-domain">
                [DOMAIN NAME]
                <span class="effect-size-badge neutral-effect">No Effect</span>
            </div>
            <div class="finding-effect">Hedges' g = [VALUE]</div>
            <div class="finding-significance">No significant effect detected</div>
        </div>



        <!-- Card Type C: Significant Negative Effect -->
        <div class="finding-card negative-significant" style="background: #fef2f2;">
            <div class="finding-domain">
                [DOMAIN NAME]
                <span class="effect-size-badge negative-effect">Significant Negative
                 ↪  Effect</span>
            </div>
            <div class="finding-effect">Hedges' g = [NEGATIVE VALUE]</div>
            <div class="finding-significance">p = [P-VALUE] • Statistically significant
             ↪  negative effect</div>
        </div>



        <!-- Card Type D: Trending/Marginal Effect -->
        <div class="finding-card non-significant" style="background: #fffbeb;">
            <div class="finding-domain">
                [DOMAIN NAME]
                <span class="effect-size-badge neutral-effect">Trending</span>
            </div>
            <div class="finding-effect">Hedges' g = [VALUE]</div>
          <div class="finding-significance">p = [P-VALUE] • Marginally significant trend</div>
        </div>



    </div>
</div>




<!-- BAYESIAN ANALYSIS OPTIONS -->
```

```
       <!-- Option A: Evidence Against Effect (Null Result) -->
       <div class="bayesian-analysis" style="background: linear-gradient(135deg, #dbeafe 0%, #bfdbfe
        ↪  100%);">
           <div class="bayesian-title">
               <svg width="20" height="20" fill="currentColor" viewBox="0 0 20 20">
                  <path fill-rule="evenodd" d="M6 2a2 2 0 00-2 2v12a2 2 0 002 2h8a2 2 0 002-2V4a2 2 0
                    ↪  00-2-2H6zm1 2a1 1 0 000 2h6a1 1 0 100-2H7zm6 7H7a1 1 0 100 2h6a1 1 0 100-2zm0
                    ↪  4H7a1 1 0 100 2h6a1 1 0 100-2z" clip-rule="evenodd"/>
               </svg>
               Bayesian Meta-Analysis Results
           </div>
           <div class="bayesian-content">
               The Robust Bayesian Meta-Analysis provided <strong>moderate support for the absence of
↪  an effect</strong> (BF₁₀ = [VALUE]) and <strong>[moderate/strong] evidence against publication
↪  bias</strong> (BF_PB = [VALUE]). These findings strengthen confidence in the null result.
           </div>
       </div>



       <!-- Option B: Evidence For Effect -->
       <div class="bayesian-analysis" style="background: linear-gradient(135deg, #d1fae5 0%, #a7f3d0
        ↪  100%);">
           <div class="bayesian-title">
               <svg width="20" height="20" fill="currentColor" viewBox="0 0 20 20">
                  <path fill-rule="evenodd" d="M6 2a2 2 0 00-2 2v12a2 2 0 002 2h8a2 2 0 002-2V4a2 2 0
                    ↪  00-2-2H6zm1 2a1 1 0 000 2h6a1 1 0 100-2H7zm6 7H7a1 1 0 100 2h6a1 1 0 100-2zm0
                    ↪  4H7a1 1 0 100 2h6a1 1 0 100-2z" clip-rule="evenodd"/>
               </svg>
               Bayesian Meta-Analysis Results
           </div>
           <div class="bayesian-content">
               The Robust Bayesian Meta-Analysis provided <strong>strong evidence for the
↪  effect</strong> (BF₁₀ = [VALUE]), supporting the frequentist findings. [ADDITIONAL
↪  INTERPRETATION].
           </div>
       </div>




       <!-- Option C: Inconclusive Bayesian Evidence -->
       <div class="bayesian-analysis" style="background: linear-gradient(135deg, #fef3c7 0%, #fde68a
        ↪  100%);">
           <div class="bayesian-title">
               <svg width="20" height="20" fill="currentColor" viewBox="0 0 20 20">
                  <path fill-rule="evenodd" d="M6 2a2 2 0 00-2 2v12a2 2 0 002 2h8a2 2 0 002-2V4a2 2 0
                    ↪  00-2-2H6zm1 2a1 1 0 000 2h6a1 1 0 100-2H7zm6 7H7a1 1 0 100 2h6a1 1 0 100-2zm0
                    ↪  4H7a1 1 0 100 2h6a1 1 0 100-2z" clip-rule="evenodd"/>
               </svg>
               Bayesian Meta-Analysis Results
           </div>
           <div class="bayesian-content">
               The Robust Bayesian Meta-Analysis provided <strong>inconclusive evidence</strong>
↪  (BF₁₀ = [VALUE]), suggesting the data cannot clearly distinguish between the presence and absence
↪  of an effect.
           </div>
       </div>
```

```



        <!-- STATISTICAL NOTES OPTIONS -->

        <!-- Power Analysis Note -->
        <div class="statistical-note" style="background: #f3f4f6; border-left: 3px solid #6b7280;">
            <div class="statistical-note-title">Statistical Power Analysis</div>
            <div class="statistical-note-content">
                Individual studies were generally <strong>underpowered</strong> ([PERCENT]% median
↪   power) to detect the observed summary effect size. This highlights the importance of meta-analytic
↪   approaches for detecting small but meaningful effects.
            </div>
        </div>



        <!-- Heterogeneity Note -->
        <div class="statistical-note" style="background: #fef3c7; border-left: 3px solid #f59e0b;">
            <div class="statistical-note-title">Heterogeneity Analysis</div>
            <div class="statistical-note-content">
                Substantial heterogeneity was observed across studies (I² = [VALUE]%, τ² = [VALUE]),
↪   suggesting that [INTERPRETATION OF HETEROGENEITY SOURCE].
            </div>
        </div>



        <!-- Publication Bias Note -->
        <div class="statistical-note" style="background: #dbeafe; border-left: 3px solid #3b82f6;">
            <div class="statistical-note-title">Publication Bias Assessment</div>
            <div class="statistical-note-content">
                [Evidence of publication bias was/was not detected] using [METHOD]. [INTERPRETATION
↪   AND IMPLICATIONS].
            </div>
        </div>



        <!-- FIGURE SECTION -->
        <div class="figure-section" style="margin: 3rem 0; text-align: center;">
            <div class="figure-container" style="background: white; border: 1px solid #e5e7eb;
             ↪  border-radius: 12px; padding: 2rem; box-shadow: 0 2px 8px rgba(0, 0, 0, 0.05);">
                <img src="[IMAGE_URL]"
                     alt="[DESCRIPTIVE ALT TEXT]"
                     style="max-width: 100%; height: auto; border-radius: 8px; margin-bottom: 1rem;">
                <div class="figure-caption" style="font-size: 0.9rem; color: #4b5563; line-height:
                 ↪  1.6; text-align: left; margin-top: 1rem;">
                    <strong>Figure [NUMBER]: [FIGURE TITLE].</strong> [DETAILED CAPTION DESCRIBING THE
                     ↪  FIGURE AND KEY FINDINGS]
                </div>
            </div>
        </div>


       <!-- No results yet -->
       <div class="current-results-container">
```

```
        <div class="preregistered-notice">
            <p><strong>This is a pre-registered protocol.</strong> Initial analysis has not yet been
             ↪  conducted. Results will be posted when the analysis is completed and the pre-print is
             ↪  published.</p>
        </div>


    </div>
</body>
```

5.2.2.2 Inclusion and exclusion criteria

```
{r}
#| label: Inclusion and Exclusion Criteria
#| description: Here you can create two seperate boxes with the inclusion and exclusion criteria for
 ↪  the meta-analysis.
#| eval: false
#| echo: true


## Inclusion and Exclusion Criteria {#criteria}

::: {.criteria-box}
### Inclusion Criteria

- **Type of Studies:** [Study designs included, e.g., "Randomized controlled trials", "Original
 ↪  research written in English"]
- **Intervention:** [Specific intervention criteria, e.g., "Intranasal oxytocin administration"]
- **Population:** [Target population, e.g., "Healthy adults aged 18-65"]
- **Outcomes:** [Primary and secondary outcomes, e.g., "Social behavior measures", "Validated scales
 ↪  for trust"]
- **Data Requirements:** [Statistical requirements, e.g., "Must report means and standard deviations
 ↪  or effect sizes"]
:::

::: {.criteria-box}
### Exclusion Criteria

- [Specific exclusion criteria relevant to the research question]
- [Additional exclusion criteria]
- [Overlap/duplicate publication handling]
:::
```

5.2.2.3 Methodology summary

```
{r}
#| label: Methodology summary of main content area
#| description: A section handy to explain a summary of the methodology of the living meta-analysis.
#| eval: false
#| echo: true

## Methodology Summary {#methodology}
```

```
This living meta-analysis follows the PRISMA guidelines for systematic reviews.

The quality of included studies will be assessed using [specific quality assessment tool, e.g.,
↪ "Cochrane Risk of Bias tool for RCTs"]. Studies will be independently rated by two reviewers
↪ [describe rating categories and scoring system].

Statistical analyses will be conducted using [statistical software] with the [specific packages].
↪ [Type of effects model] will be used with [specific estimator]. Heterogeneity will be evaluated by
↪ calculating [heterogeneity measures]. Publication bias will be assessed using [bias assessment
↪ methods].
```

5.2.2.4 Search strategy

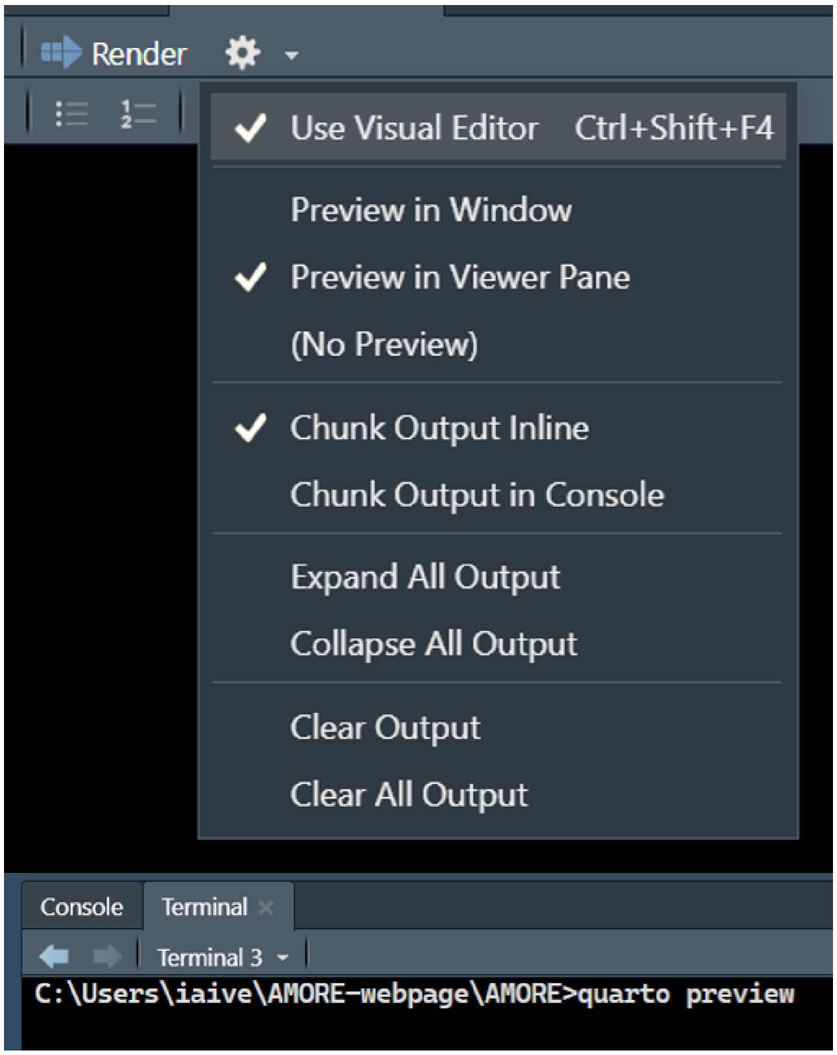
```
{r}
#| label: Search strategy of main content area
#| description: Write the search strings used on different search engines like PubMed or similar for
↪ reproducibility. It is very nice to have the search strings on the project page as this is
↪ apparently not always documented as makes reproducibility more difficult.
#| eval: false
#| echo: true


## Search Strategy {#search-strategy}

::: {.search-strategy-box}
<span class="search-db">write database here:</span>
write [search AND string] here

<span class="search-db">EMBASE:</span>
'Pitocin®' OR 'Syntocinon®' OR 'synthetic oxytocin' AND 'labour/labor' OR 'birth' OR 'perinatal' OR
↪ 'prenatal' OR 'obstetric'

<span class="search-db">Web of Science:</span>
('Pitocin®' OR 'Syntocinon®' OR 'synthetic oxytocin' AND 'labour/labor' OR 'birth' OR 'perinatal' OR
↪ 'prenatal' OR 'obstetric') AND ('delivery' AND 'augment*' OR 'induc*')

<span class="search-db">MEDLINE:</span>
'Pitocin®' OR 'Syntocinon®' OR 'synthetic oxytocin' AND 'labour/labor' OR 'birth' OR 'perinatal' OR
↪ 'prenatal' OR 'obstetric' AND 'antepartum'

:::
```

5.3 Previewing your work

It is very important to preview the work BEFORE pushing to Github repository.

You can preview by clicking the render button located at the the top bar of the RStudio interface, or by writing quarto preview in the terminal.

You can preview in Viewer pane OR in browser version. I recommend adjusting the viewer pane size when previewing to check how the content adjusts to different screen sizes.



```
C:\Users\iaive\AMORE-webpage\AMORE>quarto preview
```

Chapter 6

Troubleshooting

This section proposes different solutions to common errors and problems you may encounter when working on AMORE-website.

6.1 Recovery

Important: All files starting with `_` are in `.gitignore` and won't be committed to version control.

Git ensures version control so earlier version pushed to Git can be retrieved.

6.1.1 Recovery Files

The project includes backup/testing files for safe experimentation:

6.1.1.1 `_recoveryfile.qmd`

- Purpose: Test major page changes before applying to live files
- Usage: Remove `_` to render, test, then re-add `_` to hide
- Location: /docs folder

6.1.1.2 `_recoveryapp.R`

- Purpose: Test Shiny app updates before deploying
- Usage:
    1. Copy `app.R` to `_recoveryapp.R`
    2. Make changes to `_recoveryapp.R`
    3. Test by running app from `_recoveryapp.R`
    4. If successful, copy code back to `app.R`
- Location: `/docs`

6.2 Common Problems and Solutions

6.2.0.1 Authentication Error: “Invalid username or password” or “Password authentication is not supported”

Example of the issue:

```
$ git push origin main
Username for 'https://github.com': user-name@example.com
Password for 'https://user-name@example.com@github.com':
remote: Invalid username or password
fatal: Authentication failed for 'https://github.com/user/repo.git/'
```

This error occurs when authentication is not set up correctly or GitHub cannot verify your credentials.

Common Causes and Solutions:

6.2.0.2 1. Using Password Instead of Personal Access Token (PAT)

GitHub no longer accepts account passwords for Git operations via HTTPS. You must use a Personal Access Token instead.

Solution: If you haven’t set up authentication yet, go back to the Authentication Setup section and follow the instructions for either PAT or SSH.

If you already have a PAT but are still seeing this error, make sure you’re entering the token (not your GitHub password) when prompted for a password.

6.2.0.3 2. Hyphens in Email or Username

Hyphens in email addresses or usernames are valid but can sometimes cause issues if improperly handled by Git.

Solution: Ensure your email is correctly configured in Git:

```
git config --global user.email "user-name@example.com"
git config --global user.name "Your Name"
```

6.2.0.4 3. Incorrect Cached Credentials

If you previously entered wrong credentials, Git may have cached them.

Solution: Clear the cached credentials:

On Windows:

```
git credential-manager-core erase
# Or
git config --global --unset credential.helper
```

On Mac:

```
git credential-osxkeychain erase
host=github.com
protocol=https
[Press Return twice]
```

On Linux:

```
git config --global --unset credential.helper
```

After clearing, the next time you push, Git will prompt for credentials again. Enter your GitHub username and your Personal Access Token (not password).

6.2.0.5 4. Switch from HTTPS to SSH Authentication

If HTTPS continues to cause problems, switching to SSH is often more reliable.

Solution:

First, set up SSH keys (see Authentication Setup - Option 2: SSH Keys in section Getting started), then update your repository's remote URL:

```
cd AMORE-webpage
git remote set-url origin git@github.com:iaiversen/AMORE-webpage.git
```

Or for your fork (Method 2 users):

```
git remote set-url origin git@github.com:[YOUR-USERNAME]/AMORE-webpage.git
```

Verify the change:

```
git remote -v
```

You should see `git@github.com:...` instead of `https://github.com/...`

6.2.0.6 5. Verify Your Authentication Setup

Test your authentication:

For HTTPS (PAT):

```
git ls-remote https://github.com/iaiversen/AMORE-webpage.git
```

If prompted, enter your username and PAT. If it lists branches, authentication works.

For SSH:

```
ssh -T git@github.com
```

You should see: "Hi [username]! You've successfully authenticated…"

6.2.1 YAML Syntax Errors

Symptoms: - Render fails with "YAML error" message - Page shows blank content - Console shows parsing errors

Common Causes:

```
{yaml}
#| label: YAML troybleshooting
#| description: If the YAML header shows error try to see if any of these troubleshooting steps helps
 ↪  solve your problem.
#| eval: false
#| echo: true

# β–Œ Missing quotes
status: Preregistered

# βœ" Correct
status: "Preregistered"

# β–Œ Incorrect array syntax
intervention: "Value 1", "Value 2"

# βœ" Correct
intervention:
  - "Value 1"
  - "Value 2"

# β–Œ Inconsistent indentation
outcomes:
  biological:
  - "Item"  # Wrong indent

# βœ" Correct
outcomes:
  biological:
    - "Item"  # Correct indent
```

Solution: 1. Check R console for specific error line 2. Validate YAML at: http://www.yamllint.com/ 3. Ensure consistent 2-space indentation 4. Use quotes for all string values

6.2.2 Styling Problems

Symptoms: - Colors don't match - Layout broken - Responsive design fails - Changes don't appear

Solution Steps:

6.2.2.1 Inspect element

What is Inspect Element?

Inspect Element is a powerful browser tool that lets you peek "under the hood" of any webpage to see exactly how it's built and styled. Think of it as X-ray vision for websites.

What Inspect Element Shows You:

1. HTML Structure: The actual code that creates the page

2. CSS Styles: All the styling rules affecting each element
3. Which styles are actually applied: When multiple rules conflict, see which one "wins"
4. Box model: Margins, padding, and spacing around elements
5. Console errors: JavaScript errors or warnings

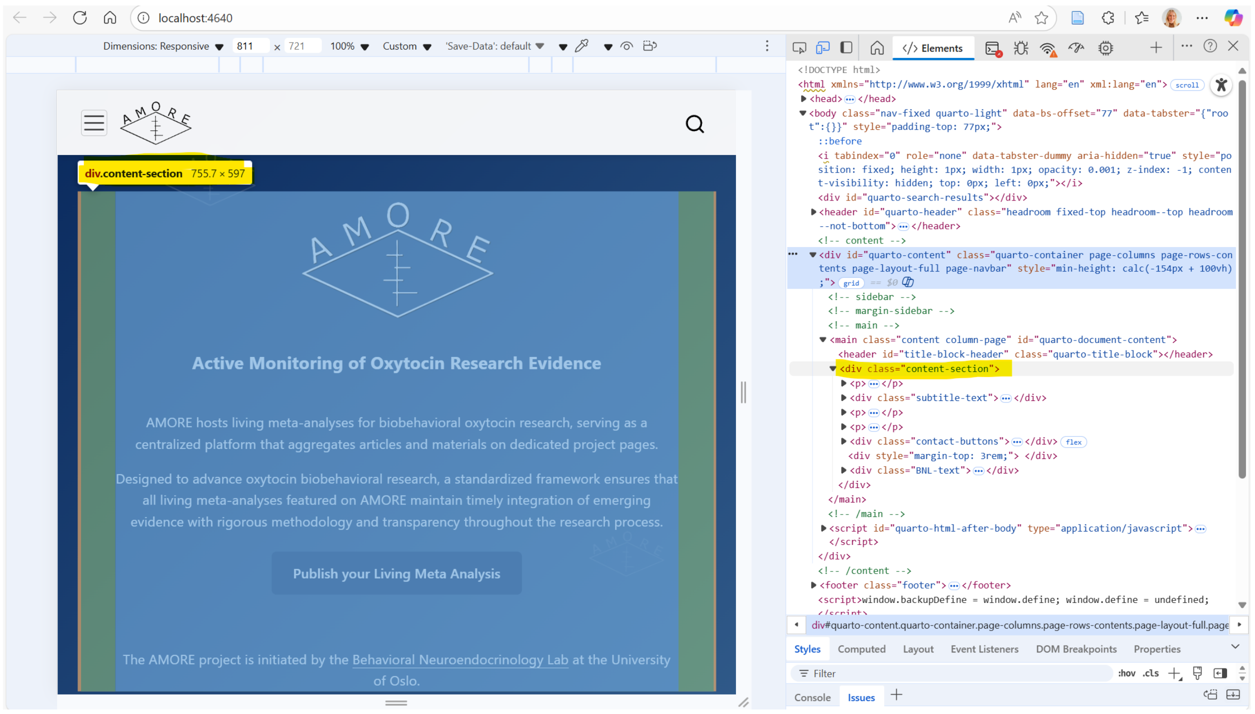
How to Open Inspect Element

In Any Browser:

- Right-click on any element → Select "Inspect" or "Inspect Element"

In RStudio Preview:

1. Render your page (it opens in Viewer or browser)
2. Click "Show in new window" icon (opens in system browser)
3. Now use inspect element as normal
4. Note: RStudio's built-in viewer has limited inspect functionality


localhost:4640
Dimensions: Responsive
811
721
100%
Custom
'Save-Data': default
Elements
div.content-section 755.7 × 597
AMORE
Active Monitoring of Oxytocin Research Evidence
AMORE hosts living meta-analyses for biobehavioral oxytocin research, serving as a centralized platform that aggregates articles and materials on dedicated project pages.
Designed to advance oxytocin biobehavioral research, a standardized framework ensures that all living meta-analyses featured on AMORE maintain timely integration of emerging evidence with rigorous methodology and transparency throughout the research process.
Publish your Living Meta Analysis
The AMORE project is initiated by the Behavioral Neuroendocrinology Lab at the University of Oslo.
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Filter
:hov
.cls
Console
Issues

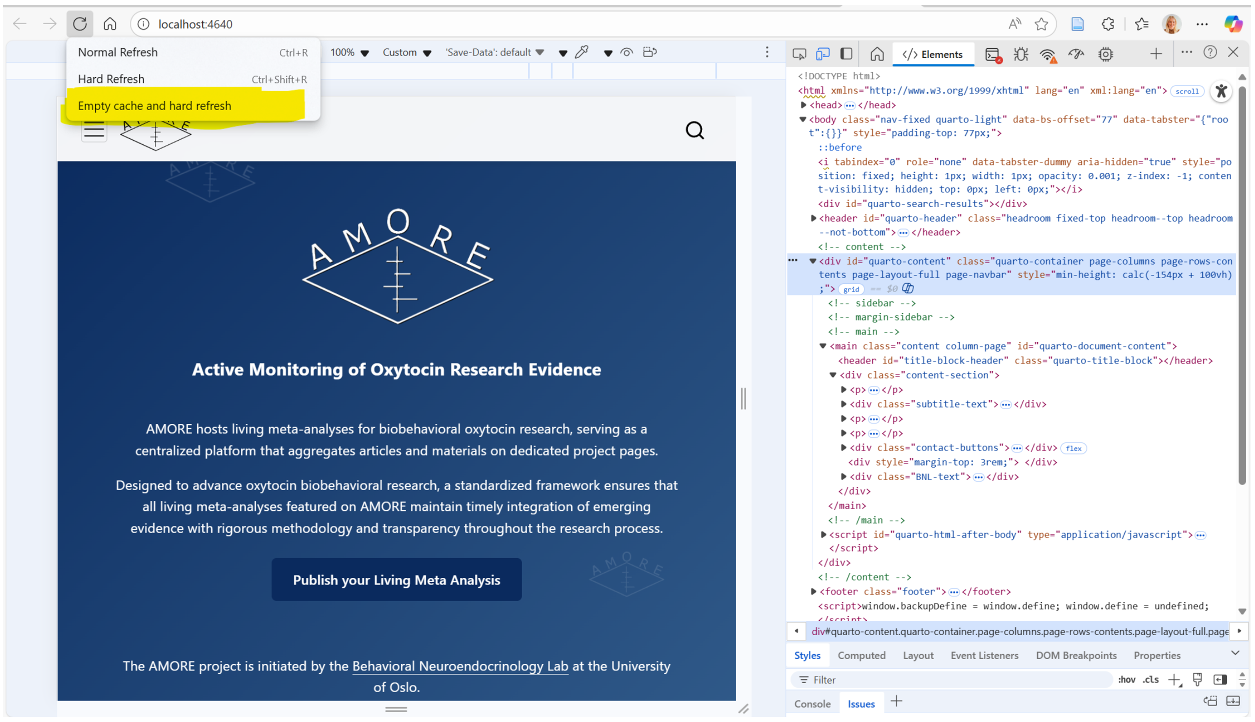
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en"> scroll
▶<head> ··· </head>
▼<body class="nav-fixed quarto-light" data-bs-offset="77" data-tabster="{"roo
  t":{}}" style="padding-top: 77px;">
    ::before
    <i tabindex="0" role="none" data-tabster-dummy aria-hidden="true" style="po
    sition: fixed; height: 1px; width: 1px; opacity: 0.001; z-index: -1; conten
    t-visibility: hidden; top: 0px; left: 0px;"></i>
    <div id="quarto-search-results"></div>
  ▶<header id="quarto-header" class="headroom fixed-top headroom--top headroom
    --not-bottom"> ··· </header>
    <!-- content -->
···▼<div id="quarto-content" class="quarto-container page-columns page-rows-con
    tents page-layout-full page-navbar" style="min-height: calc(-154px + 100vh)
    ;"> grid == $0
      <!-- sidebar -->
      <!-- margin-sidebar -->
      <!-- main -->
    ▼<main class="content column-page" id="quarto-document-content">
        <header id="title-block-header" class="quarto-title-block"></header>
      ▼<div class="content-section">
        ▶<p> ··· </p>
        ▶<div class="subtitle-text"> ··· </div>
        ▶<p> ··· </p>
        ▶<p> ··· </p>
        ▶<div class="contact-buttons"> ··· </div> flex
          <div style="margin-top: 3rem;"> </div>
        ▶<div class="BNL-text"> ··· </div>
        </div>
      </main>
      <!-- /main -->
    ▶<script id="quarto-html-after-body" type="application/javascript"> ···
      </script>
    </div>
    <!-- /content -->
  ▶<footer class="footer"> ··· </footer>
    <script>window.backupDefine = window.define; window.define = undefined;
div#quarto-content.quarto-container.page-columns.page-rows-contents.page-layout-full.page
```

Figure 6.1: Inspect element in browser

6.2.2.2 Cache Issues:

1. Save all files in RStudio
2. Restart R: `Session >> Restart R`

3. Render to browser preview - Right-click page Inspect Element
4. Clear browser cache: - - Right-click refresh button - “Empty Cache and Hard Refresh”
5. Re-render page



Active Monitoring of Oxytocin Research Evidence

AMORE hosts living meta-analyses for biobehavioral oxytocin research, serving as a centralized platform that aggregates articles and materials on dedicated project pages.

Designed to advance oxytocin biobehavioral research, a standardized framework ensures that all living meta-analyses featured on AMORE maintain timely integration of emerging evidence with rigorous methodology and transparency throughout the research process.

Publish your Living Meta Analysis

The AMORE project is initiated by the Behavioral Neuroendocrinology Lab at the University of Oslo.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en"> scroll
 ▶<head> ··· </head>
 ▼<body class="nav-fixed quarto-light" data-bs-offset="77" data-tabster="{"roo
  t":{}}" style="padding-top: 77px;">
    ::before
    <i tabindex="0" role="none" data-tabster-dummy aria-hidden="true" style="po
    sition: fixed; height: 1px; width: 1px; opacity: 0.001; z-index: -1; conten
    t-visibility: hidden; top: 0px; left: 0px;"></i>
    <div id="quarto-search-results"></div>
   ▶<header id="quarto-header" class="headroom fixed-top headroom--top headroom
    --not-bottom"> ··· </header>
    <!-- content -->
 ··· ▼<div id="quarto-content" class="quarto-container page-columns page-rows-con
    tents page-layout-full page-navbar" style="min-height: calc(-154px + 100vh)
    ;"> grid == $0
      <!-- sidebar -->
      <!-- margin-sidebar -->
      <!-- main -->
     ▼<main class="content column-page" id="quarto-document-content">
        <header id="title-block-header" class="quarto-title-block"></header>
       ▼<div class="content-section">
         ▶<p> ··· </p>
         ▶<div class="subtitle-text"> ··· </div>
         ▶<p> ··· </p>
         ▶<p> ··· </p>
         ▶<div class="contact-buttons"> ··· </div> flex
          <div style="margin-top: 3rem;"> </div>
         ▶<div class="BNL-text"> ··· </div>
        </div>
      </main>
      <!-- /main -->
     ▶<script id="quarto-html-after-body" type="application/javascript"> ···
      </script>
    </div>
    <!-- /content -->
   ▶<footer class="footer"> ··· </footer>
    <script>window.backupDefine = window.define; window.define = undefined;
```

div#quarto-content.quarto-container.page-columns.page-rows-contents.page-layout-full.page

6.2.2.3 CSS Specificity:

```
{scss}
#| label: scss troubleshooting
#| description: see if changing specificity will help
#| eval: false
#| echo: true

/* Too general - might not apply */
.box {
  color: red; }

/* More specific - will apply */
.content-section .box {
  color: red !important;  /* Use !important sparingly */ }
```

6.2.2.4 Syntax Errors:

```
{scss}
#| label: scss troubleshooting
#| description: look for syntax error
#| eval: false
#| echo: true

/* Missing semicolon */
.class {
  color: red
  margin: 10px; }

/* Correct */
.class {
  color: red;
  margin: 10px; }
```

6.3 Recovering Previous Versions

6.3.0.1 Undo uncommitted changes:

```
{bash}
#| label: Undo committed changes with git command
#| description: write code into command line interface (Bash, Powershell, RStudio Terminal)
#| eval: false
#| echo: true

# Discard changes to specific file
git checkout -- filename.qmd

# Discard all uncommitted changes
git reset --hard HEAD
```

6.3.0.2 Restore file to previous version:

```
{bash}
#| label: Restore file to previous versions with git command
#| description: write code into command line interface (Bash, Powershell, RStudio Terminal)
#| eval: false
#| echo: true

# Find the commit hash
git log --oneline

# Restore file from specific commit git checkout [commit-hash] -- filename.qmd

# Example:
git checkout a1b2c3d -- LMAs/project.qmd
```

6.4 Recovering from Mistakes

6.4.0.1 Accidentally deleted file:

```
{bash}
#| label: Version control with Git - recover accidentally deleted file
#| description: write code into command line interface (Bash, Powershell, RStudio Terminal).
#| eval: false
#| echo: true

# Restore from last commit
git checkout HEAD -- filename.qmd
```

6.4.0.2 Committed wrong files:

```
{bash}
#| label: Version control with Git - commited wrong file
#| description: write code into command line interface (Bash, Powershell, RStudio Terminal)
#| eval: false
#| echo: true

# Undo last commit, keep changes
git reset --soft HEAD~1

# Re-stage correct files
git add correct-files.qmd git commit -m "Corrected commit"
```

6.4.0.3 Pushed broken code:

```
{bash}
#| label: Version control with Git - go back to last committed version
#| description: write code into command line interface (Bash, Powershell, RStudio Terminal)
#| eval: false
#| echo: true

# Revert to previous working commit
git revert [commit-hash] git push origin main
```

Chapter 7

Website maintenance

This section provides the correct information needed to maintain a smooth running website. The section provides instructions on updating packages and installations necessary for the project, and the necessary maintenance of third party services used to host and deploy AMORE.

7.1 Third-Party Services Overview

AMORE's infrastructure relies on several third-party services. Each service has specific maintenance requirements and renewal schedules.

7.1.1 Netlify

- Purpose: Domain hosting and website deployment
- Cost: Domain name (annual payment required), deployment (free tier)
- Website: https://www.netlify.com

7.1.1.1 Maintenance Tasks:

- Renew domain name annually (Needs updated billing information, billing can be autorenewed after billing information is updated)
- Monitor build logs for deployment errors
- Check deploy settings if website fails to update

7.1.2 ShinyApps.io

- Purpose: Hosting the Living Meta-Analysis Directory Shiny application
- Account: meta-oxytocin
- Cost: Free tier
- Dashboard: https://www.shinyapps.io/admin/#/dashboard
- URL: https://meta-oxytocin.shinyapps.io/shiny-meta/

7.1.2.1 Maintenance Tasks:

- Monitor app usage (free tier has limits, might be necessary with paid version to improve server capacity). Free tier usage limits:
  - 25 active hours per month
  - 5 applications maximum
  - Automatic sleep after 15 minutes of inactivity
- Check error logs if app becomes unresponsive
- Redeploy if changes are made to `app.R`

7.1.3 GitHub

- Purpose: Version control and source code repository
- Repository: https://github.com/iaiversen/AMORE-webpage
- Cost: Free

7.1.3.1 Maintenance Tasks:

- Regularly commit and push changes
- Review pull requests if collaborators
- Keep repository organized (clear branch structure)
- Monitor repository size (GitHub has storage limits)

7.1.4 Nettskjema

- Purpose: Contact forms and data collection
- Provider: University of Oslo
- Cost: Free (institutional service)
- Renewal: Requires renewal after a set period of active years (check in Nettskjema)

7.1.4.1 Maintenance Tasks:

- Ensure form links are working
- Update form fields if project requirements change
- Export and back up form responses periodically
- Monitor embedded contact forms, perhaps better to change to old solution if embedded solution is loading to slow, or shows other errors. Follow instructions below to change back to old version.

How to change contact form solution:

Step 1. Move to folder /pages

Step 2. Rename contact.qmd to _embedded_contact.qmd

Step 3. Rename _redirected_contact.qmd to contact.qmd

Step 4. Push changes to Github repository AMORE-webpage

Nothing else needs to be done. quarto.yml will fetch the file named contact.qmd. The _redirected_contact.qmd redirects the website users to nettskjema's own website to fill the form.

7.1.5 Outlook.com Email

- Purpose: Login credentials for Netlify and ShinyApps.io
- Importance: Critical - this email provides access to necessary services

7.1.5.1 Maintenance Tasks:

- Keep email account secure
- accounts can be transferred to more secure users if neccessary.This can be done inside netlify and shinyapps.io. The procedures are easy.
  - in Netlify go to Profile - edit settings –> insert new profile credentials
  - In shinyapps.io go to Account - Settings - Transfer Account
    - Only necessary changes is later to change the .Renviron tokens and password if the meta-oxytocin account is transferred to a new Posit user.
- Monitor for service notifications
- Keep inbox organized to avoid missing important alerts (add profile on local computer or log in to browser version every now and then to monitor).

7.1.6 Zenodo

- Purpose: Archived backups of GitHub repository
- Website: https://zenodo.org
- User: iaiversen
- Cost: Free
- Features:
  - Each archive receives a unique DOI
  - Permanent preservation of repository snapshots
  - Citable versions for academic references
- Maintenance Tasks:
  - Create new archive after major release

7.2 Login Credentials Management

7.2.1 Storage Location

All login credentials are stored in:

- Location: Teams folder → AMORE → Website technicals
- Filename: `AMORE_credentials.xlsx`
- Access: Restricted to Teams member of BNE lab

7.3 Automated Maintenance Script

7.3.1 Overview

The AMORE project includes a comprehensive maintenance script (`scripts/Maintenance.R`) that automates the process of updating R packages and testing the website functionality. This script is designed to make maintenance safer and more systematic by:

- Creating automatic backups before changes
- Offering different update strategies based on risk tolerance
- Running comprehensive post-update tests
- Generating detailed maintenance logs

7.3.2 Explanation of script:

The script will run interactively, prompting you for decisions at key points.

Important: Always commit all your work to Git before running maintenance updates.

Section 1: Pre-Update Checks and Backup

The script first performs safety checks:

1. Git Status Check: Ensures your working directory is clean
2. Package Backup: Creates a snapshot of all current package versions
   - Saved as `docs/packages-backup-YYYYMMDD.csv`
   - Allows you to rollback if something breaks
3. Outdated Packages: Identifies which packages need updating
   - Shows current vs. available versions
   - Flags major version updates (higher risk)
4. Critical Packages: Highlights AMORE-essential packages

Section 2: Package Updates

You'll be presented with 6 update strategies

Option	Strategy	Risk Level	Recommended For
0	Skip updates (just test)	None	Testing current setup
1	Critical AMORE packages only	Lowest	Regular maintenance
2	Critical + high priority	Low	Recommended approach
3	All packages except major versions	Medium	Comprehensive updates
4	Everything including major versions	High	When major changes needed
5	Custom selection	Variable	Advanced users

Section 3: Post-Update Testing

After updates, the script runs comprehensive tests:

1. Basic Page Rendering: Tests `index.qmd` renders correctly
2. Project Page Rendering: Tests an LMA file renders correctly
3. Full Site Render: Renders entire website (can take a few minutes)
4. Shiny App Loading: Verifies the Shiny app loads without errors
5. ShinyApps.io Connection: Tests deployment credentials (if configured)
6. Link Checking: Optionally scans for broken links (can be slow)