CrumpLab / LabJournalWebsite Public

An R Markdown website template for a lab journal

☆ **11** stars **356** forks

☆ Star

Watch ▾

<> **Code** Issues Pull requests 9 Actions Projects Security Insigh

master

CrumpLab added page ... on Jan 28, 2019 8

View code

README.md

Releases

No releases published

Packages

No packages published

Environments 1

github-pages Active

LabJournalWebsite

An R Markdown website template for a lab journal https://crumplab.github.io/LabJournalWebsite/index.html

To use this, click the download as .zip button to start from a local computer, or skip down to the github example to fork this and get started that way.

The unzipped folder contains all of the files you need to compile a website in R Markdown. This should all work fine if you have the latest version of R and R-studio installed.

Steps for compiling on your local computer

1. Make sure R and R-studio are installed
2. Make sure the rmarkdown package is installed in R-studio. Open R-studio, click the packages tab in the lower left hand corner, click install packages, type in rmarkdown, make sure "install dependencies" is clicked on, then press install. Close R-studio.
3. Navigate to the folder you just downloaded, open the 'LabJournalWebsite.Proj' file. This should automatically open R-studio, and your current working environment will be inside this project. That means everything you save will be auto saved to this folder (unless you tell R-studio to save something somewhere else.
4. Inside R-studio you should see a files tab in the bottom right hand corner. Most files you click will be opened up as text files in the R-studio editor. Click the "Index.Rmd" file.
5. To compile the entire website, find the build tab in the top right hand corner. You should see the option to "build website". Click this. The website should be built.
6. After the website is built, you should be able to see it in the R-studio browser. There is a little button (blue arrow with a little browser icon) that allows you to

pop the website into your default web-browser. This way you can look at the website in your browser.

Important: After compilation, all of the files for displaying your website are saved in the folder where your R project resides. When you look at these in a browser (for example, by going to the folder and dragging the index.html file into a browser), you are loading from your disk. Only you will be able to see the website, because it is on your hard-drive. You need to upload to a web server to serve the webpage on the internet.

Steps for serving your webpage using github pages.

This is the source code repository for making the webpage in R-studio. At the same time, the resulting website is being served from this repository at this link https://crumplab.github.io/LabJournalWebsite/index.html.

Every github repository has the capability of serving html files (web page files) contained in the repository, this is called github pages. How this works depends a little bit on the specific repository you are using. For this repository. The webpage is served from the docs folder. The example files are set so that when you compile the example in R-studio, the output automatically goes into the docs folder. As a result, when you have these files in a github repository, github will serve the html files in your docs folder as a website.

Steps

1. For this repo to your github (press the fork button in top right hand corner, then choose your github account)
2. You should now see a copy of this repo in your github account
3. Click the Settings (also near top right), scroll down to Github Pages options
4. click the optino to serve from docs folder
5. You should see a little green message above the github pages options with a link to your new webpage.

Editing webpage and serving on github

1. download github desktop
2. make sure it is connected to your account
3. clone the website repo to your local computer
4. Open up the project file in the folder for your repo on your local computer (.rproj file)
5. Edit the .rmd files in R-studio
6. Recompile website (build website when index.rmd is loaded), or knit individual

.rmd files

7. send your changes back to the online github repository (note this can be done in github desktop, or directly in R-studio, in R-studio you will see a git tab if you are working in a git repo. Click the git tab, click the diff button, which will show you if there are any new changes. Click each of the files that you want to commit. Write a short note to describe the changes. Press the commit button. Wait a couple seconds, your changes should now be served on your website).