Website Development with R

Sophia Shalhout and David Miller

11/23/2019

Overview

- This working document is a rough user guide on how to create a website using R via a package called `blogdown`. It will also walk you through opening in account on Netlify, which will host your website.
- There are numerous resources for webpage development, including an amazing bookdown called "blogdown: Creating Websites with R Markdown" (https://bookdown.org/yihui/blogdown/). We've created this resource as we have tried to provide details and pearls that we struggled with and we belive these may be helpful to others.
- This is a ongoing, 'living' document that is constantly being edited and expanded. Please provide comments to allow us to optimize this document for all R users.

Getting Started

Load Packages for RMarkdown

```
tinytex::install_tinytex()
```

Github

1. Register a GitHub account (https://github.com)
2. Create a new Repository ("+" sign in the upper right hand corner)

3. Create a name for the repo and remember this as it will be used

4. Copy the github repo address

R

Starting Your Project

1. Create a new project in R using version control

- File -> New Project -> Version Control -> Git (clone a project from a Git repository)
- Enter the Git Repository URL:e.g. https://www.github.com/DM-Miller/themillerlab
- Create Project (open in new session would be my recommendation

2. Install blogdown

```
if (!requireNamespace("devtools")) install.packages("devtools")
devtools::install_github("rstudio/blogdown")
```

3. Load blogdown

```
library(blogdown)
```

4. Create a new_site

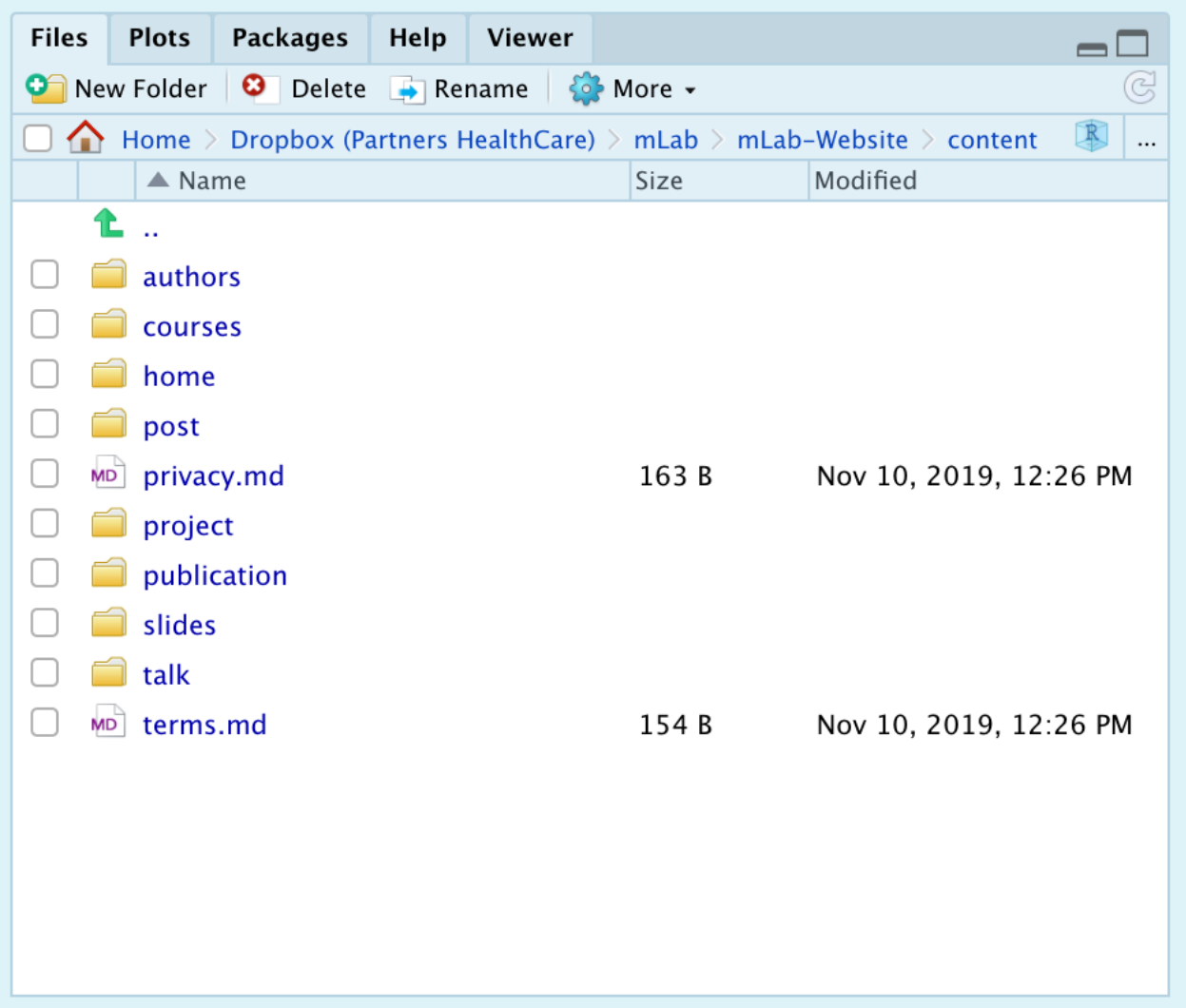
```
new_site()
```

- this will install Hugo and download new files

- The default theme is lithium
- If you want a different one (e.g. hugo-academic) add the following into the function:

```
new_site(theme = "gcushen/hugo-academic")
```

- Once you do this it will download all of the necessary files into your working directory
  - for example:



Build Your Site

build_site()

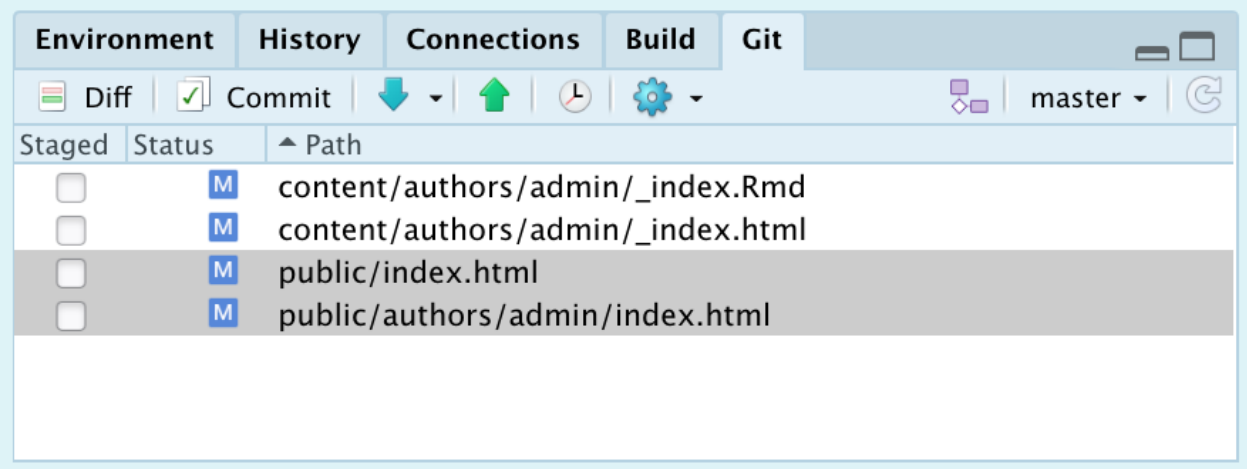
Use the function `build_site()` in the `blogdown` package

```
blogdown::build_site()
```

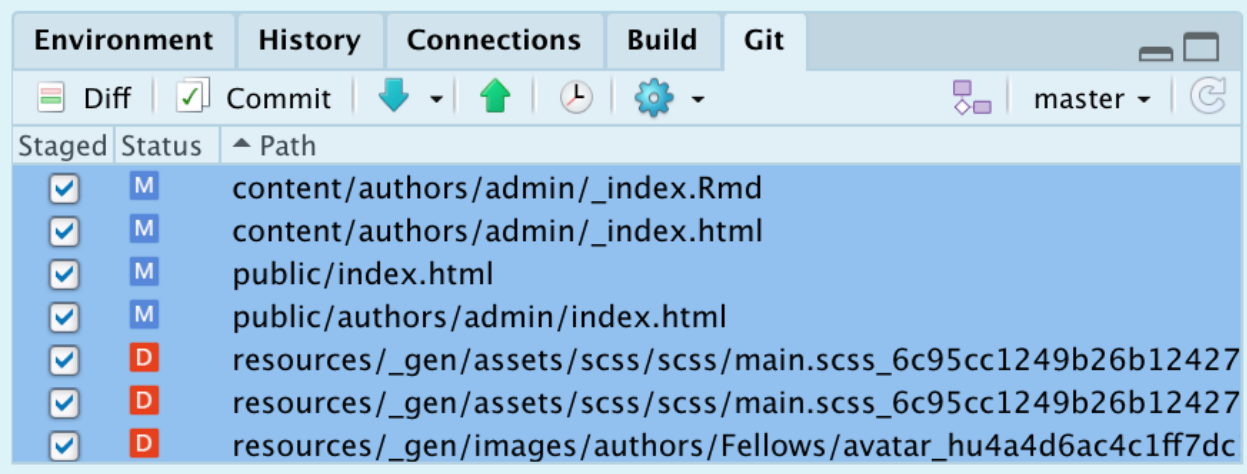
serve_site()

```
blogdown::serve_site()
```

This function allows you to serve your website locally so you can view it in the "Viewer" tab in the "Files Plots Packages Help Viewer" pane ## Commit Changes to GitHub 1. Once you've set up your version control to github, each change that you make will be tracked by R. These changes will be cataloged in the Git Pane.

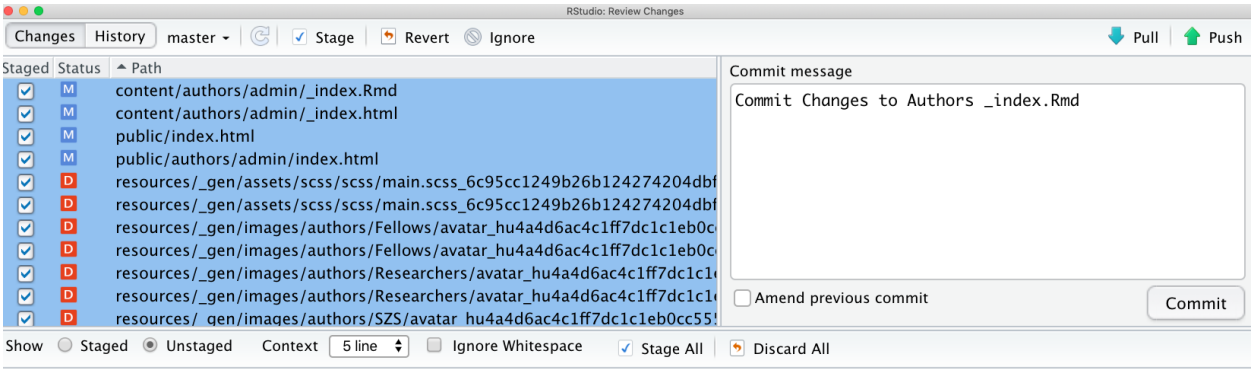


2. To commit them to GitHub, select all of the files and click on one of the check boxes.

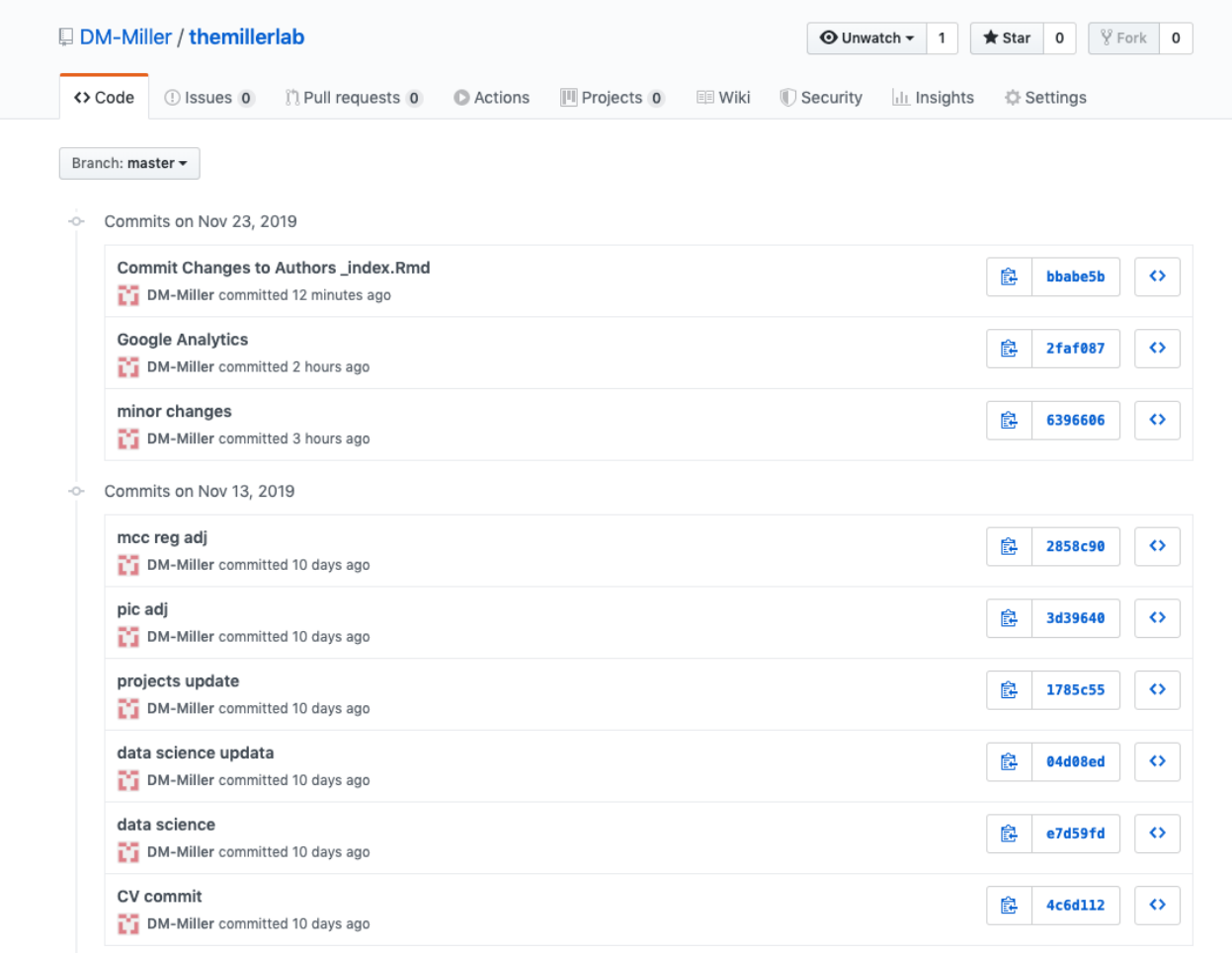


3. Then click on the "Commit" button in teh above pane

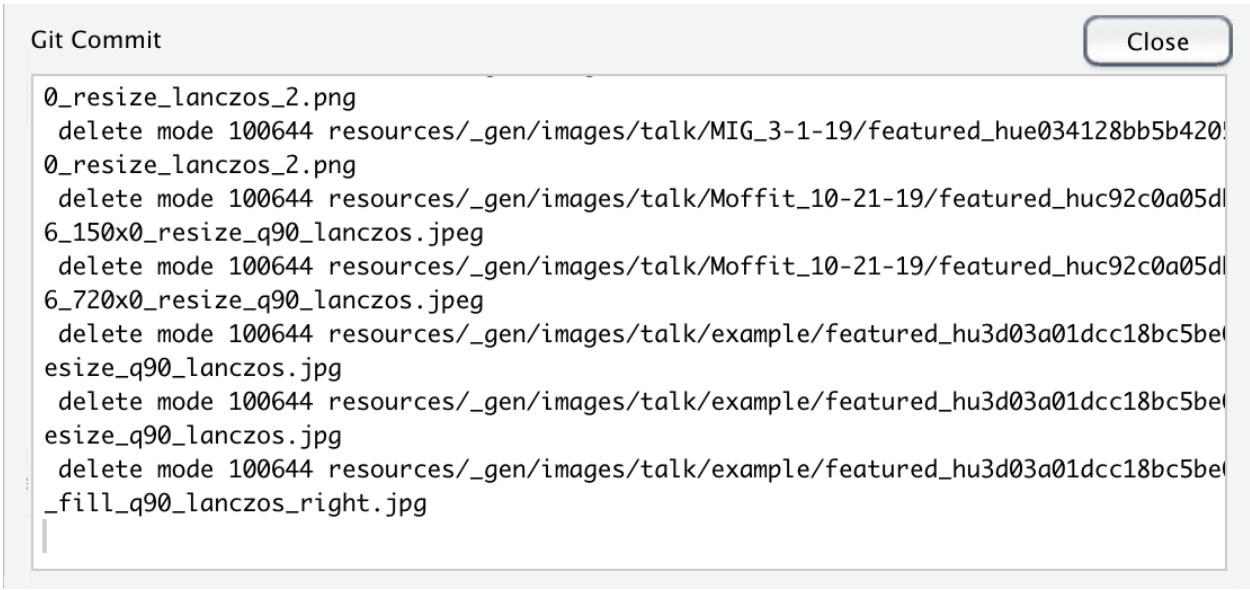
4. That will bring up a new window which will allow you to review the changes you've made. Write a message in the "Commit Message" Box in order to annotate the changes.



* Of note, these changes will show up in GitHub under "Commit".



5. Then press "Commit". After you do that a new window will pop up called "Git Commit".



6. Then press "Close". Press "Push" in the Review Changes Pane, which will push the changes to GitHub.

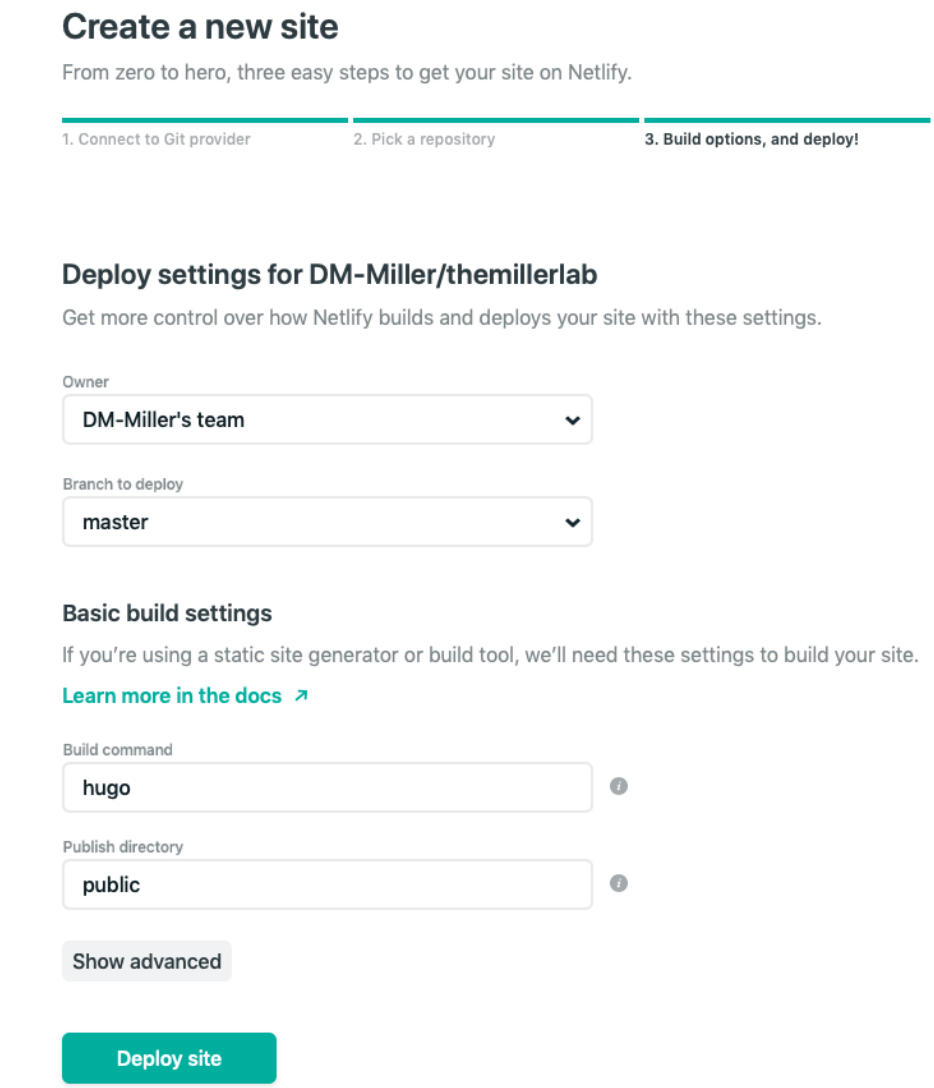
This window will pop up. Once it has been pushed, press “Close”. That should be it.

Netlify

Overview

- Netlify is a is a San Francisco-based cloud computing company that offers hosting and serverless backend services for static websites (source: https://en.wikipedia.org/wiki/Netlify).
- There are other options. We’ve used it to host our website.
- Here are a few steps to get started:

1. Go to netlify (https://www.netlify.com) and sign up
2. We would recommend signing up with your GitHub account, especially if your are doing to deploy your website from this account
3. Launch a New Site from Git
4. Select Continuous Deployment from GitHub
5. Pick your GitHub Repository for which you want Continuous Deployment
6. Modify Settings (here’s an example)



7. Select Deploy Site 8. Select a Domain Name (you can purchase one from Netlify if it is available)

Designing a Website Using Hugo-Academic

Overview

- Hugo-Academic is a very reasonable theme for a academic website
  - for an example see: https://www.themillerlab.io

Configurating the Website

Overview

- There are several config files that you need to know about

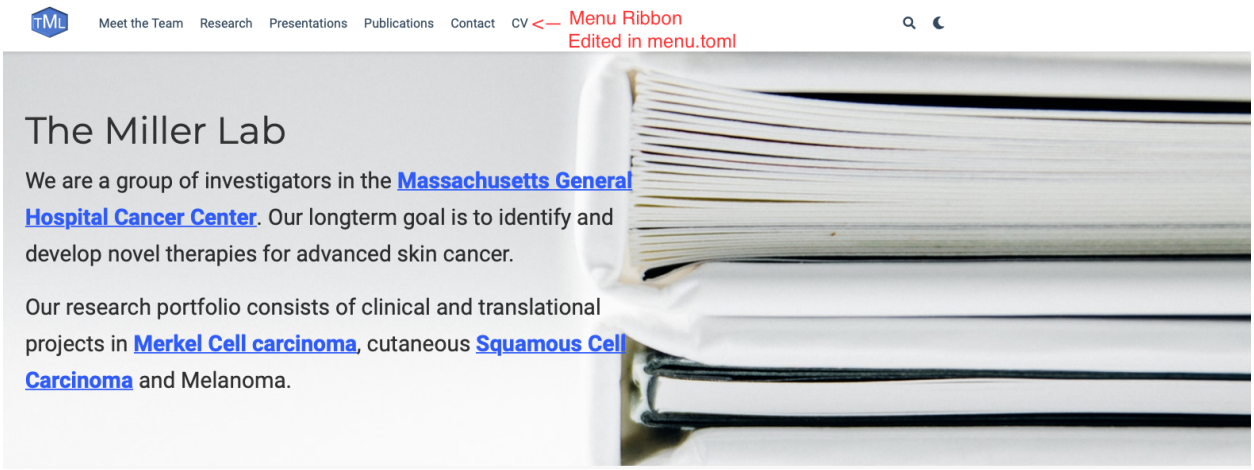
config.toml

- This file is where you will create the Title of your website
  # Title of your site
  title = “The Miller Lab”

menus.toml

- Location
  - config/_default/menus.toml

- This is where you will control which items go on the top menu ribbon on your website.



* This is the default

[[main]]
name = “Demo”
url = “#hero”
weight = 10

[[main]]
name = “Posts”
url = “#posts”
weight = 20

[[main]]
name = “Projects”
url = “#projects”
weight = 30

[[main]]
name = “Publications”
url = “#featured”
weight = 40

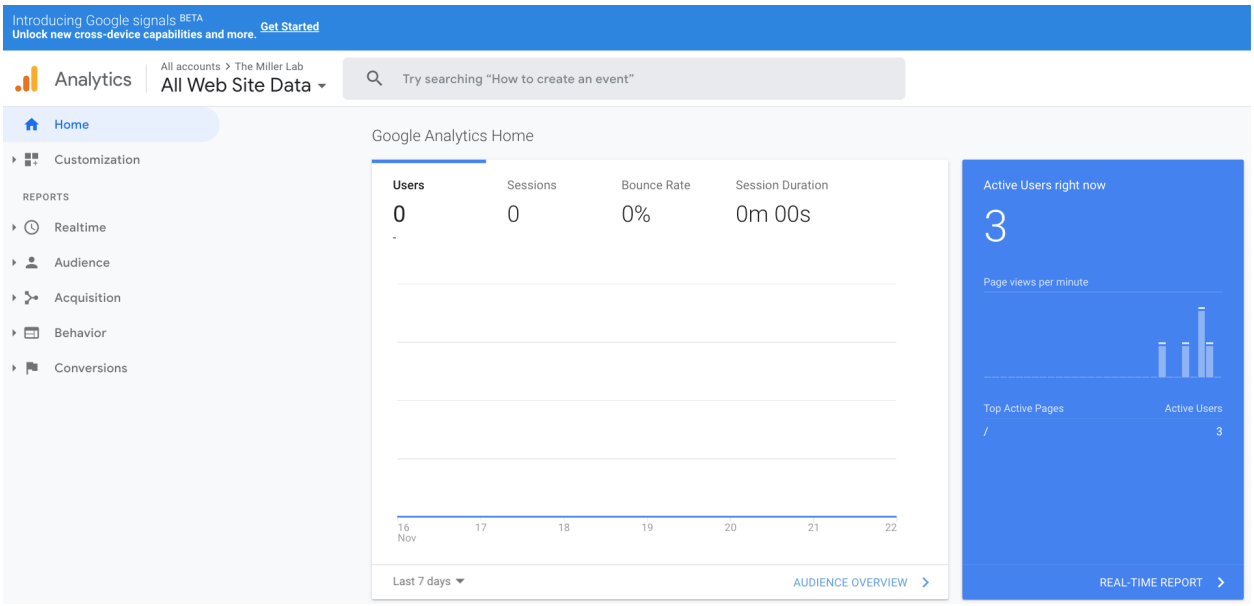
[[main]]
name = “Courses”
url = “courses/”
weight = 50

[[main]]
name = “Contact”
url = “#contact”
weight = 60

params.toml

- Location
  - config/_default/params.toml

- This is where you will control which items such as
  - Logo
  - Contact Details (email, addresses, etc)
  - Social Media Details
  - Configuration of publication pages
  - Marketing such as Google Analytics



Widgets in Hugo Academic

Overview

- The academic theme has lots of widgets; these are in your home folder; look fofr “active = “ and assign true or false depending on whether or not you want them “active = false” will hide the widget

- The Hugo project uses +++ to denote demarcation for TOML documents. It is a way for them to have a single text file that consists of TOML in the “frontmatter” (document meta-data), and a second “document” that is the body of page content.

Star

Octicon-start and Get Started Clicker

Hero widget

- This is the first widget worth adjusting
- This is where you change the title of the
- Within the TOML, you can add text using
  [cta_note]
  label = 'Latest release'

[cta]
url = "https://sourcethemes.com/academic/docs/install/"
label = "Get Started"
icon_pack = "fas"
icon = "download"

[cta_alt]
url = "https://sourcethemes.com/academic/"
label = "View Documentation"

People Widget

- Under Content -> Home -> people.md
- This will allow you place a "Meet the Team" segment on the web page
- User_groups = [ ] is key
- Within the [ ] you should specify the names of the subfolders in the "authors"
- For example [content]
  Choose which groups/teams of users to display. Edit `user_groups` in each user's profile to add them to one or more of these groups.
  user_groups = ["Fellows", "Students"]
  Fellows and Students are subfolders
  Within those subfolders the info for the site is placed in a .md or .Rmd file called "_index.md" or "_index.Rmd"

Talk Widget

- To edit content go to content -> talk
  - Then edit
- Example code that was embedded in default Talk widget page {{% alert note %}}
  Click on the **Slides** button above to view the built-in slides feature.
  {{% /alert %}}
- Slides can be added in a few ways:
  - **Create** slides using Academic's
    Slides feature and link using `slides` parameter in the front matter of the talk file

  - **Upload** an existing slide deck to `static/` and link using `url_slides` parameter in the front matter of the talk file

- Example from the miller lab website
  - I created a folder in "static" called "slides"
  - Then I put the pdf of "MIG_3-1-19.pdf" in that folder
  - Then in the index.md file under content->talk->MIG_3-1-19 I used the following
    url_slides: "slides/MIG_3-1-19.pdf"
  - I also removed "example" from line 29
  - Slides: " "
  - This was key, otherwise it didn't work

- **Embed** your slides (e.g. Google Slides) or presentation video on this page using shortcodes.
- Further talk details can easily be added to this page using `*Markdown* and $\rm \LaTeX$ math code.`

Projects Widget

- Md document is in the Home folder "projects.md"
  - content -> home -> projects.md

- There is a built in mechanism to link your active projects
- The details for your project, however is located:
  - Content -> project
  - Within thei "project" folder you can add new subfolders of the different ongoing projects
- Toolbar
  - There is a tool bar feature that you can edit, e.g.
    [[content.filter_button]]
    name = "Data Science"
    tag = "data-science"

- projects subfolder
  - As mentioned above, the actual descriptions of the projects are in
  - Content -> project
  - Here's an example
  - Content -> project -> data-science
- With the "data-science" subfolder, there are two key files
  - Featured.jpg (the picture that will be feature)
    Index.md
- Index.md for projects
  - Here put the data you want in the YAML
- Pay attention to
  - Tags:
  - -data-science
  - This will link this file to the tool bar in the projects page

Other Pearls

Links in Hugo

- The default Hugo target directory for your built website is public/
  - Therefore, if you want to link to a file with this format
  - `[Merkel Cell Carcinoma tumor Registry](MCC_Registry.Rmd)`
  - Then the MCC_Registry.Rmd file must be in the public folder; this tripped me up for hours!
  - This is considere a permalink

Netlify TOML

- I Ran into a problem where I could not deploy from GitHub to Netlify
- I ran the error message into google and Alison Hill recommended that I create a netlify.toml
- https://alison.rbind.io/post/2019-02-19-hugo-netlify-toml/
- The key was addint the hugo_version to the latest version I was workin on (see below)
- Created netlify.toml in R in the workin directly (see content below)
- I then pushed it to GitHub and was all set!
- This is the code I got from Alison Hill's website

  [build]
  publish = "public"
  command = "hugo"

  [context.production.environment]
  HUGO_VERSION = "0.59.1"
  HUGO_ENV = "production"
  HUGO_ENABLEGITINFO = "true"

  [context.branch-deploy.environment]
  HUGO_VERSION = "0.59.1"

  [context.deploy-preview.environment]
  HUGO_VERSION = "0.59.1"

  [context.deploy-preview]
  command = "hugo -b $DEPLOY_PRIME_URL –buildFuture"

  [context.branch-deploy]
  command = "hugo -b $DEPLOY_PRIME_URL –buildFuture"‘

Adding images to your website

- I added an img folder to the home folder and placed the file images in that folder
- Then within the Rmd file within the home folder I used this code
- `![](/img/DMM_pic.jpg)`

Link to a PDF of your resume/CV from the menu

- To enable, copy your resume/CV to `static/files/cv.pdf` and uncomment the lines below
- Direct the navigation bar to your CV by editing the menus.toml file
  - Wd -> config -> _default-> menus.toml [[main]]
    name = "CV"
    url = "files/DMM_CV.pdf"
    weight = 70

```
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_3.6.1  magrittr_1.5    tools_3.6.1     htmltools_0.4.0
##  [5] yaml_2.2.0      tinytex_0.17    Rcpp_1.0.3      stringi_1.4.3
##  [9] rmarkdown_1.17  knitr_1.26      stringr_1.4.0   xfun_0.11
## [13] digest_0.6.23   rlang_0.4.1     evaluate_0.14
```