How to build an acceptable Job Market Website with in RStudio

Jakob Miethe

5th of July

What is blogdown?

- an R package that creates static websites
- no server sided scripts needed, you can upload it anywhere and also look at it on your desktop
- I found no compatability issues between Ubuntu and Windows
- go-to resource: https://bookdown.org/yihui/blogdown/
- fortunately, we don't have to understand everything happening here to use blogdown
- we write in markdown and build a website using hugo

Yihui Xie (2019). blogdown: Create Blogs and Websites with R Markdown. R package version 0.13.

Yihui Xie, Alison Presmanes Hill, and Amber Thomas (2017). blogdown: Creating Websites with R Markdown. Chapman and Hall/CRC. ISBN 978-0815363729

What is R Markdown?

- "markdown" is a document format (and syntax) used mostly (I think) for code descriptions or github cover pages
- "R markdown" integrates this with R and significantly extends it to reports, websites, slides (these here for example), dashbords and many more; uses knitr, inspired by Pandoc
- can include R code (all kinds) as well as text (also Latex code)
- go-to resource: https://bookdown.org/yihui/rmarkdown/
- for blogdown, we just need to write text (with some hightlights and gimmicks) as you see in the examples

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2019. Rmarkdown: Dynamic Documents for R. https://CRAN.R-project.org/package=rmarkdown.

What is Hugo?

- an open source static site generator written in Go
- source: https://gohugo.io/
- advertising says its fast and stable, I have no comparison but also not a single complaint so far
- also no issues with DSGVO on static pages
- can be hosted using Netlify, Heroku, GoDaddy, DreamHost, GitHub Pages, GitLab Pages, Surge, Aerobatic, Firebase, Google Cloud Storage, Amazon S3, Rackspace, Azure, CloudFront, and probably more
- I use netlify

Lets jump right in

```
install.packages("blogdown")
```

```
blogdown::install_hugo()
```

then File -> New Project choose a folder

wait, then have a look a the folder, then:

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

wait and have a look at the folder. What just happened?

if you have trouble try an empty theme:

```
blogdown::new_site()
```

other themes: https://themes.gohugo.io/

What is in this folder now?

- a fully functional website. click "show in viewer" in Rstudio to see it
- the folders include many files that I never went through but that you can edit when you found the answer to a question on the hugo forum for example
- .toml files are configuration files, you will use "config.toml" extensively: it defines the parameters of your websites
- .md files are markdown files we use for content
- .css files are used in your theme (currently: hugo academic), changing them is possible (find help online) but risky, because they do get changed in updates (when you download the theme again)
- there are also .Rmd files, .content files, .json files that I didn't need to touch so far

The basic structure

- "config" includes configuration .tomls
- "content" is where you can put your files (such as papers, cv), and your .md files, **this is the main folder** with the site resources we will change
- "public" is your site as you will post it with .html files that blogdown/hugo created. Unless you are in crazy last minute workaround mode, only R should touch that folder
  - You can though: for example to change your pdfs without recompiling the website. Make sure you also change them in "content" which will override "public" when you compile again
- "themes" this is where we donwloaded the academic theme. If you don't like something about the theme, you can go into the .css files here and change it (I decreased the font size)
- I don't use the other folders (but I have the suspicion that "static" might be a better place for pdfs than content?)

Your input

most of the time you will only touch:

1. yoursitename.Rproj, in my case BasicWebsite.Rproj to load the project into RStudio
2. config.toml to provide the site parameters
3. the content folder to provide all content

Lets have a look at the website Hugo created

Clean up

before we go into detail, take out the advertising so we only see the website:

- look in content/home/ which is the default place for content
- Our adds are demo.md and hero.md. Open both and change "active = true" to "active = false".
- This is how you in/exclude content (in this case "widgets") in your website. We'll talk about these widgets later.

if RStudio doesn't update automatically when you save, run:

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

I hade some issues with run_site() but if it works, its great

An academic job market website needs very little

1. a Menu
2. some content
    - text with links, colors, other tweaks
    - pdfs
    - maybe a video

that's pretty much it

1. The Menu

we see the menu here: “config/_default/menus.toml”, in other sites (like mine which is largely based on an older site of Alison Hill) its in the config.toml

- here you can see all menu entries
- the "name" will appear in the menu
- with the url you can link directly to .md files ("widgets") in the home folder or to subfolders
  - if you use a hash, the file will appear on the same page (infinity scroll)
  - if you link to it with /yourmdfile, it will go on a separate page
  - For subfolders simply use /yourfolder/yourmdfile
  - You can also directly link to pdfs, for example: url = "/cv/cv_miethejakob.pdf"
- "weight" orders the menu entries

2. The content (remainder of this talk)

1. infinity scroll or not?
2. how to organize content
3. how to write content

To scroll or not to scroll?

- the Hugo template is an infinity scroll website
- but it also includes links to other pages in the menu (as we know already)
- compare "courses" to the other links
- active widgets can be called in the url (even if you didn't link to them in the menu)

When I started, I based my website on the one of Alison Hill which used infinity scroll back then. She has also moved away from it! https://alison.rbind.io/

How to organize content: No infinity scroll

Expecting time constrained job market recruiters who are used to google scholar, I wanted to emulate the page structure there. I also thought that in the long run (when I'll have many projects, teaching material etc there), a compartmentalized website makes more sense for me.

There are basically four ways that I have found for subsites:

1. 'master widgets' combine several .md files from one folder with certain visualization settings. This is the way hugo academic is meant to be built. It felt a little overkill for a job market website, however.
2. lists of .md files without a widget
3. a direct link to a document (your cv for example)
4. a workaround I use until I figure the rest out: one .md file in which I write content directly without links etc

What are widgets?

- the hugo academic theme provides "widgets" to organize your content
- these are basically layout templates for your sites
- see an overview here: https://sourcethemes.com/academic/docs/page-builder/
- when I built my page there were much less, I didn't take much advantage of it
- each of our .md files chooses a widget (check)
- the content/courses page is organized differently (I haven't checked it)

Content 1: A master widget

- have a look at the "Projects" link, it shows 2 pages
- they are in /content/project/
- look at the 'master widget' in /content/home/ called projects.md
- do we already understand some stuff in the preamble there?
- the .md also defines tags for example that you could give your different projects (look at the internal project to see this)
- where does the file define which .mds to use? [content] and page_type. Try changing that and see what graphical parameters the widget defines, irrespective of the other .md files it then links
- now you could just load more .md files into that folder and they will show up inside projects on the website.

Content 2: Listing .md files

- instead of having the "projects.md" widget organize the content, lets simply create a new folder in content named "research" and run the following code:

```
blogdown::new_content(path="/research/researchproject1.md")
blogdown::new_content(path="/research/researchproject2.md")
blogdown::new_content(path="/research/researchproject3.md")
```

- lets go into the menu (or config) .toml and add to the menus:

```
[[main]]
  name = "Research"
  url = "research/"
  weight = 51
```

prettier with a master widget but works

Content 3: A direct link

This is simple: In the menu .toml you simply define the link:

```
url = "/cv/cv_miethejakob.pdf"
```

keep in mind that files are also uploaded if you didn't provide a menu entry and can be accesses with the correct url. Also notice that I created a folder "cv" in my content folder in which I put the pdf

I was also able to out of the box link to a video. You can also do that inside markdown pages of course.

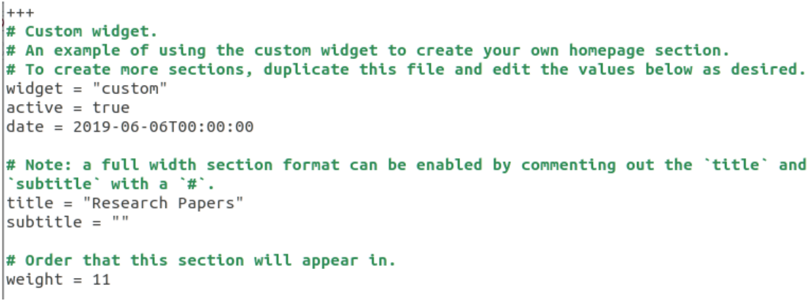
Content 4: A simple .md file

This is a workaround but its what got me closest to the standard job market website fast. In the standard hugo academic logic, each paper should be its own project/widget/folder/etc

- my workaround: one .md that I use as a page
- copy or create an .md file
- lets look at the preamble and at the markdown code

What I didn't solve (probably have to go into the theme css files where the custom widget is defined): I'd like to get rid of the dates. . .

Content 4a: The preamble

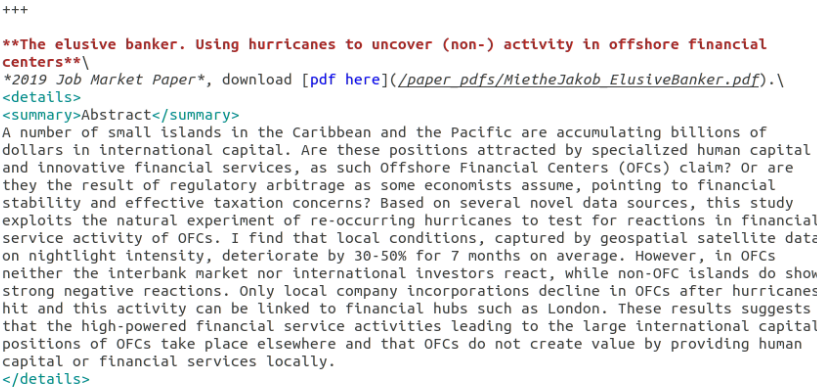


```
+++
# Custom widget.
# An example of using the custom widget to create your own homepage section.
# To create more sections, duplicate this file and edit the values below as desired.
widget = "custom"
active = true
date = 2019-06-06T00:00:00

# Note: a full width section format can be enabled by commenting out the `title` and
`subtitle` with a `#`.
title = "Research Papers"
subtitle = ""

# Order that this section will appear in.
weight = 11
```

Content 4b: The markdown code



```
+++

**The elusive banker. Using hurricanes to uncover (non-) activity in offshore financial
centers**\
*2019 Job Market Paper*, download [pdf here](/paper_pdfs/MietheJakob_ElusiveBanker.pdf).\
<details>
<summary>Abstract</summary>
A number of small islands in the Caribbean and the Pacific are accumulating billions of
dollars in international capital. Are these positions attracted by specialized human capital
and innovative financial services, as such Offshore Financial Centers (OFCs) claim? Or are
they the result of regulatory arbitrage as some economists assume, pointing to financial
stability and effective taxation concerns? Based on several novel data sources, this study
exploits the natural experiment of re-occurring hurricanes to test for reactions in financial
service activity of OFCs. I find that local conditions, captured by geospatial satellite data
on nightlight intensity, deteriorate by 30-50% for 7 months on average. However, in OFCs
neither the interbank market nor international investors react, while non-OFC islands do show
strong negative reactions. Only local company incorporations decline in OFCs after hurricanes
hit and this activity can be linked to financial hubs such as London. These results suggests
that the high-powered financial service activities leading to the large international capital
positions of OFCs take place elsewhere and that OFCs do not create value by providing human
capital or financial services locally.
</details>
```

This is a workaround

- I simply continued in the same .md with the next project and had one .md for research, teaching, and policy
- the other two things in my menu are a direct link (cv) and an infinity scroll link (contact) that is another .md we can look at
- this is very limited compared to the rich widget structure (where you can have folders full of stuff for each project)
- but its sufficient for a job market website
- for much more fleshed out website thats useful long term, check out https://alison.rbind.io/

I still chose this over google scholar because as you saw, I can easily move to a more complicated website when I have enough material so it makes sense and I have full control over that content.

Final content: The “about” widget

this is the first thing your visitor sees, add:

- welcome text
- research interests
- references

I never managed to make the image square but it should be possible in the theme css files I would think.

Ingredients summary

my website is very simple. It only consists of:

1. the config .toml with the menu and some options (I got rid of all the social media links for example)
2. three .md files I put in the content folder (research, teaching, policy)
3. an about and a contact widget in the home folder
4. pdfs of my cv and my papers (and a video at the time)

Thats it!

How to publish your website

simply publish the public folder, I'll show you how to do this with netlify.

Presentation

This presentation is written in R markdown, output: beamer_presentation. Can we have an R group on how to make these pretty and combine them with other beamer templates such as metropolis or the DIW gc template?