Categories: Newsletter Issue 2020:3

Create a Simple and a Free Job Market Website

By: Alex Hollingsworth

With the job market approaching, we thought it would be nice to provide a tutorial on how to create simple and free job market website. Having a clear, concise, and simple to navigate job market website is of particular importance to health economists since we usually apply to jobs across a few different disciplines and schools.

There are already a number of nice tutorials on how to create an academic website, but most focus on established academics. There are also a number of paid-for options that work nicely. This tutorial is designed for new job candidates that want to create a simple job market website using entirely free tools. We will use freely available software R, Hugo, and Netlify. You can check out a preview of the website we will make here: https://job-candidate.netlify.app.

Why focus on job market candidates?

There is already a lot of advice out there about how to make a website. There's even advice geared towards a building a general academic website (See https://www.dsquintana.blog/free-website-in-r-easy/). While much of this advice is quite good, I think it slightly misses the mark for new graduate students seeking their first job. This is because most of the existing material is suited for *established* academics that are not actively seeking jobs.

Established academics have lots of published papers and many other accomplishments/efforts to be outlined. Since they have so much content, many of the themes/tutorials for academic websites separate content into pages like `Research`, `Teaching`, and `Conferences`. New PhD students seeking their first job have don't need complex websites with many pages.

More importantly, the website's target audience and goal are both different for established academics than first time job market candidates. The target audience for an established academic's website is an interested party (colleague, reporter, conference attendee, etc.) that has sought out this specific website, perhaps through a Google search of the academic's name. The goal is simply to provide relevant information to this interested party. If you're a graduating PhD student seeking your first job, your target audience are those on the hiring committees at the jobs you applied to and your goal is simple. **To get hired.**

Always try to think about how your website is structured from the perspective of your goal and target audience. Those visiting an established academic's site may be willing to poke around and spend some time to find the specific content they are looking for. Busy hiring committee members may have 400+ job candidate websites to look through and might not be willing to explore every nook and cranny of your site. It's possible that adding a fancy bell or whistle to your website may make it harder for a person to find something they were looking for. If you're a job market candidate, you want to make it easy for time-crunched hiring committee members to find the most relevant and important information about you that increases the chances of you being hired.

What should my website look like?

Think of a job-market website as a better, and less-standardized, version of a CV. It likely won't be printed out so you can use hyperlinks and color; you'll often include a picture of yourself; you can omit/add sections as you see fit; and you can express creativity/personality if you want to.

If you think about a job market website as type of electronic CV, then Sarah Jacobson's advice (https://twitter.com/SarahJacobsonEc/status/1026231483638460417) for CVs applies well. You want the site to be a clean and compact conveyor of relevant information. Make sure the most important information is at the top. Don't include the unimportant. A starting guide for the order of importance is:

1. Name

2. Contact information
3. Education
4. Research interests
5. Publications
6. Working papers
7. Work in progress
8. Teaching
9. Anything else (e.g., grants, cool pictures you took, recipes, your travel blog)

What should you include in the `Anything else` category? I agree with Sarah that it's stuff you want discussed when you are not in the room. So no need to include Stata/Word/Latex/Favorite Software experience unless it is truly noteworthy. However, if you want to show off a cool set of hiking pictures that you think really reflect your personality well, why not? This is your website, there are no actual rules, just keep the end goal in mind.

When is a paper a `Working Paper` v. a `Work in Progress`? My personal rule is that a paper goes in the Working Papers section if it is posted online at a place like SSRN or SocArXiv. Most of the time I consider a paper to be a working paper around the same time that I would be willing to submit it a journal. There is no one-size fits all rule here, but my advice would be to have a link to the paper if you consider it to be a working paper.

I completely agree with Sarah's advice that it is not a great idea to post loose/very preliminary ideas in `Works in Progress`. You may be asked detailed questions about work you put in this section during an interview. Basically, don't use this section as a brainstormed list of pie-in-the-sky ideas that you haven't actually started unless you're ok revealing that.

Your job market website should also contain a link to your CV, job market paper, and any other publications/working papers you want people to see. Ideally these links would not take you to an external site. I have heard stories that CVs or papers hosted on Google Drive/Dropbox may not be accessible in all parts of the world. You can avoid issues like this by simply hosting these documents on your own site.

Lastly, as a goal, try to keep the site very simple and avoid unnecessary bells and whistles. You want it to look good on mobile and across various browsers. Keeping it simple maximizes the chances that it looks good when used by these different technologies. I try to keep my site on one single page.

Making the site.

You can absolutely create a successful job market website using other tools, like wordpress, squarespace, or Google. But if you're interested in making a quick and free site using R, Hugo, and Netlify, read-on.

Prerequisite installs

1. You'll need to download R (https://www.r-project.org) and R-studio (https://rstudio.com/products/rstudio/download/) since we will be using these to interact with the bones of your website. Note: You do not need strictly need R for this, but the bookdown package is fantastic and simplifies the whole process.
2. After you install both, open RStudio and install the blogdown package by typing `install.packages("blogdown")` in the console.
3. Visit https://github.com/hollina/template-job-market-website and select `Download ZIP` in the drop-down menu on the green code button.
4. Extract the zip file and place it in the place you'd like to store your website. Consider renaming the folder if you'd like.
5. Open the file `job-market-website.Rproj` from this directory. This should launch RStudio.
6. Type `library(blogdown)` in RStudio's console.
7. Install Hugo by typing `blogdown::install_hugo(force = TRUE)` in the console (this could take a little while).
8. Launch the default website by typing `blogdown::serve_site()` in the console.

- This will launch a mobile view of the website in RStudio and provide you with a local url on which you can view the website. This will be something like http://127.0.0.1:4321, which can be put into a browser on your computer while the Hugo server is running.
- The launched website should look just like this: https://job-candidate.netlify.app.

- Don't worry it is not posted publicly yet. Here you can explore what the website looks like and make sure that you're happy with everything before you post it online.
- As you change various features in the code (e.g., your name), you should see the changes immediately in the local version of your website. This is a neat feature since it allows you to test everything out before you publicly post changes. Note: changing the static files (pictures/PDFs) may require you to restart the server before you see them in the preview.
- Once you're done tinkering, you can stop the server by typing `blogdown::stop_server()` in the console.
- I recommend leaving the server running while you're customizing since you'll get to see immediately how any changes affect your website.

Customize

This theme is a slightly modified version of the very popular `hugo-academic` theme. The defaults for `hugo-academic` area bit too busy for a simple job market website. Moreover, the default theme doesn't include sections for teaching or working papers. Chase Eck, a job market candidate at the University of Arizona, introduced a few simple modifications to the theme to contain these sections and to simplify the rest. You can check out his website here: https://chase-eck.github.io.

I used his job market website as a starting point to create a modified theme, which is what you downloaded from GitHub. If you want to see all of the edits made to the default `hugo-academic` theme, you can check out the Appendix section of this article.

Now the fun begins. You get to make the website your own. Let's start by adding personal details.

Add a picture

- Place your profile picture in the `content/authors/admin` directory.
- Save the picture as `avatar.jpg`.

Add your CV

- Add your CV to the static/files/ directory. If the file name of your pdf is not `cv.pdf` then you will need to alter the file name of your pdf or change the code to reflect your different file name.
    - e.g., you'd need to change this in `config/_default/menus.toml` and in `content/authors/admin/_index.md`

Edit your personal information

- Open the `config/_default/params.toml` file
    - Edit lines 90-299 as you see fit.
    - You should certainly change
        - The title of your website to be your name (line 9)
        - Organization to be your current place of study/employment (line 48)
        - The site description (line 51)
        - Your email address (line 91)
        - Your phone number (line 92)
        - Your address (line 96)
        - Add your twitter handle to line 130 of
        - Replace `name = "DM Me"` in line 115 to be `name = "whatever your twitter handle is here"`
        - If you have a Google analytics number (for recording site-traffic) , add that to line 284
- Open the `content/authors/admin/_index.md` file
    - Change the title (line 3) to be your name
    - Change your role (line 13) to be correct (e.g., `role: PhD candidate in Economics`)
    - Change your education/institutions

- line 17: your current institutuion (e.g., `name: Big Ten University`)
- lines 30-32: your PhD granting institutuion
- lines 33-35: your masters granting institutuion
- etc

- Add a link to your email address on line 48
- Add a link to your twitter profile on line 51
- Add your google scholar link on line 54
- Add your github link on line 57
- Make sure the path and filename for your cv is correct on line 63.
- Edit your interests on lines 24 and 25
- Write a longer (but not too long) bio on line 75 for the front page
- Write a shorter bio for line 21

For most things if you comment them out, then they disappear, but be sure this is the case by looking at the local preview of your site.

Add your working papers and publications

- Add any PDFs you want to share to the `static/files/` directory.
- In the `content/publication` directory you'll notice that there's a folder for each paper on the webpage.
- All you need to do is create a separate folder for each paper you'd like on your webpage and in each folder make an index.md file.
- Everything should be really straightforward.
  - Just make sure the publication type is set to 3 if this is a working paper and 2 if it is published.
  - Also you should set the `url_pdf` to point to the specific pdf file (`files/pdfname.pdf`) or url for the paper.
- You can be really fancy here and include things like images, slides, bibtex templates, references, etc. But you can also keep it simple with a list of authors, journal names, and a paper title.
- Play around with the different settings to make sure it's how you'd like.

Note: If you don't have any publications yet (which is the norm for a first time job market applicant!) then you should change line 8 of `content/publications.md` to be `active = false` instead of `active = true`. You should also remove the publications link from the drop down menu on the top bar by opening `config/_default/menus.toml` and either deleting or commenting out the following code:

```
[[main]]
name = “Publications”
url = “#publications”
weight = 20
```

Edit the teaching section

- Edit the `content/home/teaching.md` file to reflect courses you've taught or TA'd.

Publish

Now it's time to publish your website!

First do one last build of the website using RStudio. This updates the files in your `public` folder, the contents of which is what you will be actually sharing with the world as your website.

1. Open RStudio by opening the `job-market-website.Rproj` file.
2. Type `library(blogdown)` inRStudio's console.
3. Build your site by typing `blogdown::serve_site()` in the console.
4. Check out the mobile and browser preview.
5. If you're happy with the site, shut down the `blogdown::stop_server()` command.

Ok. Now you can decide where/how you want to host your website. The contents of the public directory are a complete website that you can host using any number of services. I use Hugo to create my own website and I host it using GitHub pages. I know people that host the website using university servers. In this tutorial, we want to keep it as simple as possible so we will use Netlify.

- You'll need to create a Netlify account by visiting https://www.netlify.com.
- Once you create an account and log-in, you should visit your `Sites`.
- You won't have any yet. To make your own just drag the contents of your public window to the marked area
- Your site should build automatically and will be given a default (and often weird) url
- You can change the url in the `General`-> `Site Details` section
- Note: If you want to use a custom domain name (e.g. yourfullname.com) you can set that up using Netlify or another company like Namecheap. You will need to pay someone if you want this option. It's fairly simple to do. Here's a guide on how to implement a custom domain purchased from Namecheap using Netlify, https://www.codementor.io/@ekunolaeasybuoy/setting-up-domain-with-namecheap-netlify-12yiksvig6.

Last, go to your new website! Make sure it looks good on mobile/the computer. And make changes as you see fit.

Remember, every time you make an edit you'll need to

1. open the `job-market-website.Rproj` file
2. rebuild the site using the blogdown commands
3. drag the newly updated public folder to netlify

It's really simple once you get the hang of it.

Enjoy!

Appendix

For the curious, here are the major changes made to the default `hugo-academic` theme. You don't need to do any of this if you downloaded the theme from https://github.com/hollina/template-job-market-website, but here it is.

Turn most widgets off

We set most of the widgets to be turned off by making sure that the preamble active is set tofalse (i.e., `active = false`).

We did this for:

- accomplishments.md
- demo.md
- experience.md
- featured.md
- people.md
- posts.md
- projects.md
- skills.md
- slider.md
- tags.md
- talks.md
- gallery/index.md
- hero.md

Modify the publications section

Steps to edit `publications.md` yourself:

- Navigate to `content/home`.
    - Change line 10 from `title = "Recent Publications"` to `title = "Publications"`
    - Change line 30 from `publication_type = ""` to `publication_type = "2"`
    - Change line 40 from `view = 2` to `view = 3`

  - Delete lines 69-73

- Save `publications.md`

Create working papers section

Steps to create `working_papers.md` yourself:

- Navigate to `content/home`.
- Create an additional copy of the `publications.md` file
- Rename it to be `working_papers.md`
- Open `working_papers.md`
  - Change line 8 from `weight = 90` to `weight = 100`. This will put working papers below publications. You can always switch these if you'd prefer.
  - Change line 10 from `title = "Publications"` to `title = "Working Papers"`
  - Change line 30 from `publication_type = "2"` to `publication_type = "3"`
- Save `working_papers.md`

Remove embedded email box at the bottom of the main page

- Open `content/home/contact.md`
- Change line 18 to be `email_form = 0`

Add teaching section

Steps to create `teaching.md` yourself:

- Open a new blank file in your favorite editor like Atom or Sublime Text.
- Paste in the text below (with ## separating each course)
- Save the file as `` ` ``content/home/teaching.md

```
+++
# 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 = 2020-08-09

# Note: a full width section format can be enabled by commenting out the `title` and `subtitle` with a `#`.
title = “Teaching”
subtitle = “Average instructor rating: 4.8/5”

# Order that this section will appear in.
weight = 120

+++

### Econ 235: Health Economics
Instructor of record: Fall 2019

MTH 101: Basic Statiscis
TA: Fall 2018, 2017, 2016
```

Reorder the navigation bar

- Open `config/_default/menus.toml`
- Be sure that `"files/cv.pdf"` directs to cv file (i.e., change `cv.pdf` to be the same filename as your cv you just put in the `files` directory)

```
[[main]]
name = “CV”
url = “files/cv.pdf”
weight = 10

[[main]]
name = “Publications”
url = “#publications”
weight = 20

[[main]]
name = “Working papers”
url = “#working-papers”
weight = 30

[[main]]
name = “Teaching”
url = “#teaching”
weight = 40

[[main]]
name = “Contact”
url = “#contact”
weight = 50
```

Remove busy features from home page

- Open `config/_default/params.toml`
- Turn off all other location features by commenting out (adding a # to the start of the line)
    - coordinates (line 100), directions (line 103), office hours (line 107), and appointment url (line 110)
- Turn off the map feature by setting line 276 to be `engine = 0`
- Turn off all social media icons that I do not use or want strangers to have access to (e.g. Skype)
    - Comment out lines 116, 117, and 118 by adding a # at the beginning of the line
- change line 153 from `reading_time = true` to `reading_time = false`
- change line 298 from `ai = false` to `ai = true`
- change line 19 from `day_night = true` to `day_night = false` to remove the dark theme.

Useful links

- Open config/_default/params.toml
- Turn off all other location features by commenting out (adding a # to the start of the line)
    - coordinates (line 100), directions (line 103), office hours (line 107), and appointment url (line 110)
- Turn off the map feature by setting line 276 to be engine = 0
- Turn off all social media icons that I do not use or want strangers to have access to (e.g. Skype)
    - Comment out lines 116, 117, and 118 by adding a # at the beginning of the line
- change line 153 from reading_time = true to reading_time = false
- change line 298 from ai = false to ai = true
- change line 19 from day_night = true to day_night = false to remove the dark theme.

Useful links

Check out this guide, https://www.dsquintana.blog/free-website-in-r-easy/, which served as the basis for this tutorial. It is geared towards those with at least a few publications under their belt and many presentations, but is worth checking out.

ASHEcon

1100 Vermont Avenue NW Suite 650
Washington, DC 20005
P: (202) 737 6608
F: (202) 496 0134
Email: info@ashecon.org

Search …