Building a website using blogdown in R

Alistair Bailey

May 18 2018

Contents

List of Tables

List of Figures

Summary

These are my instructions for how to build a website using R. The inspiration came from a series of tweets by Dan Qunitana about how to build an academic website using the blogdown package (Xie, 2018).

Following these instructions you can build a website for Bibi the Cat, using the Hugo academic theme but you can of course make a website for anything you like and use any of the many themes available.

These instructions assume you have R (R Core Team, 2018) and Rstudio installed and are reasonably comfortable using these tools.

For much more detail check out the fantastic blogdown book.

Chapter 1

Getting started

1.1 Installation

First you'll need to install the blogdown package:

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

Then use blogdown to install the static site generator Hugo:

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

1.2 Installing JabRef

If you are going to link academic publications on your website you will probably find it useful to have JabRef installed to create the necessary files you need.

JabRef is an open source bibliography reference manager.

This is not the only way to do things, but it's what I'm familiar with and is free.

See the publications section for the full details.

1.3 Other files

If you want to build Bibi the Cat's website you'll need these files.

Chapter 2

Creating and deploying an initial website

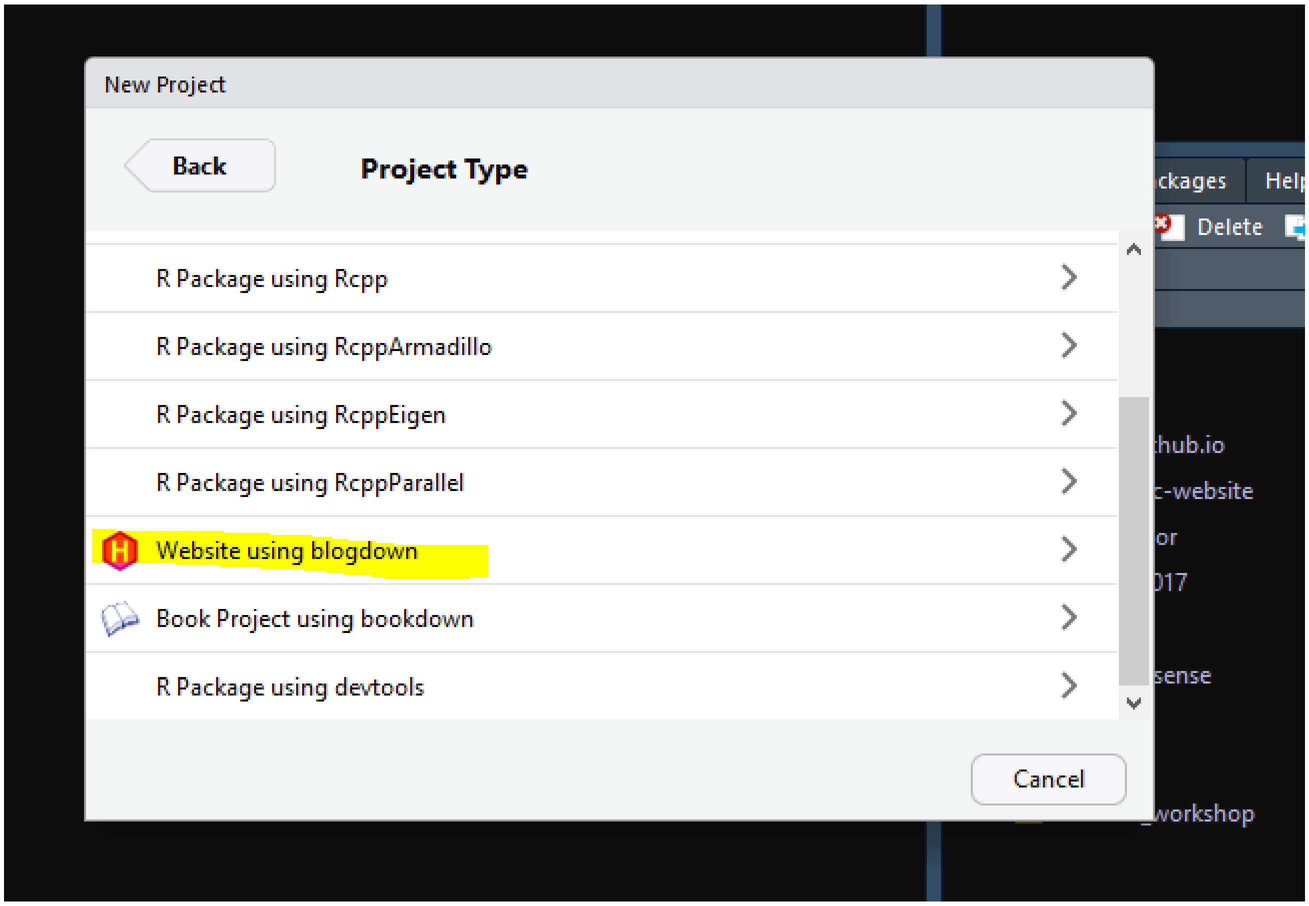
2.1 Creating a website in R

We're going to create a website for Bibi the Cat aka The Tiny Tiger.

Here she is:

Go to `File > New Project > New directory` and then scroll down and choose **Website using blogdown**.

Figure 2.1: Bibi the Cat



New Project

Back

Project Type

R Package using Rcpp

R Package using RcppArmadillo

R Package using RcppEigen

R Package using RcppParallel

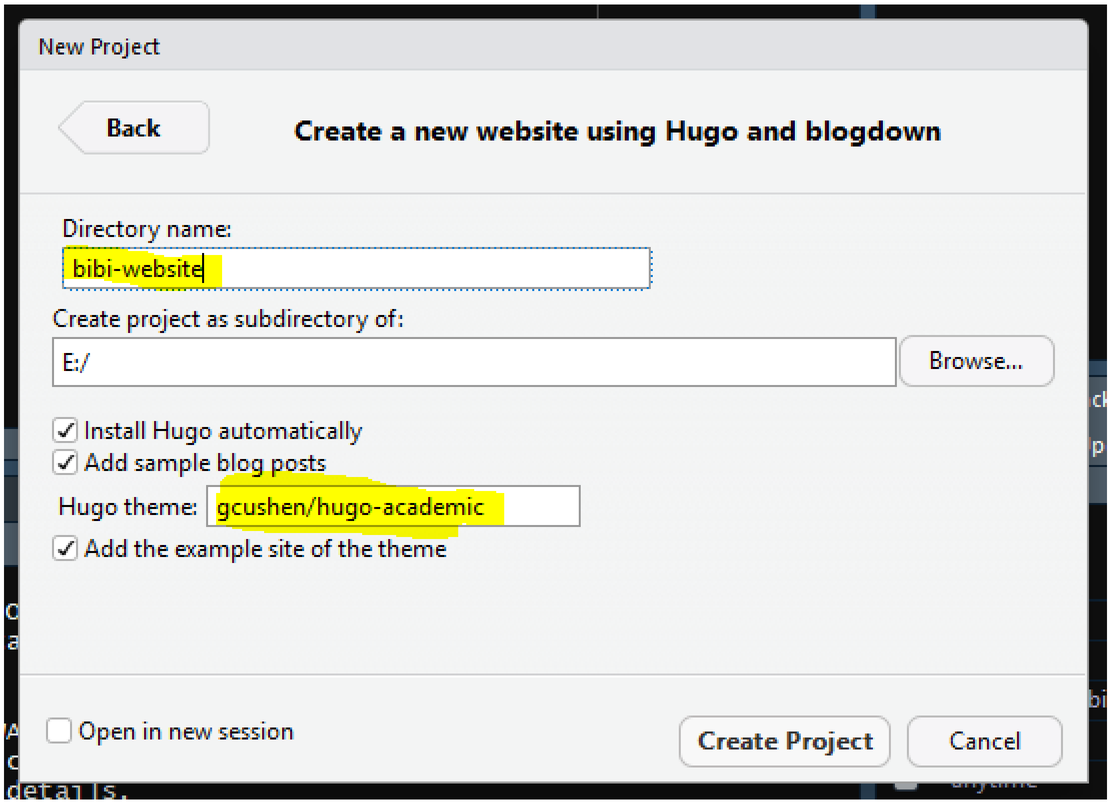
Website using blogdown

Book Project using bookdown

R Package using devtools

Cancel

This will then take you to the another screen where you can choose the directory for the



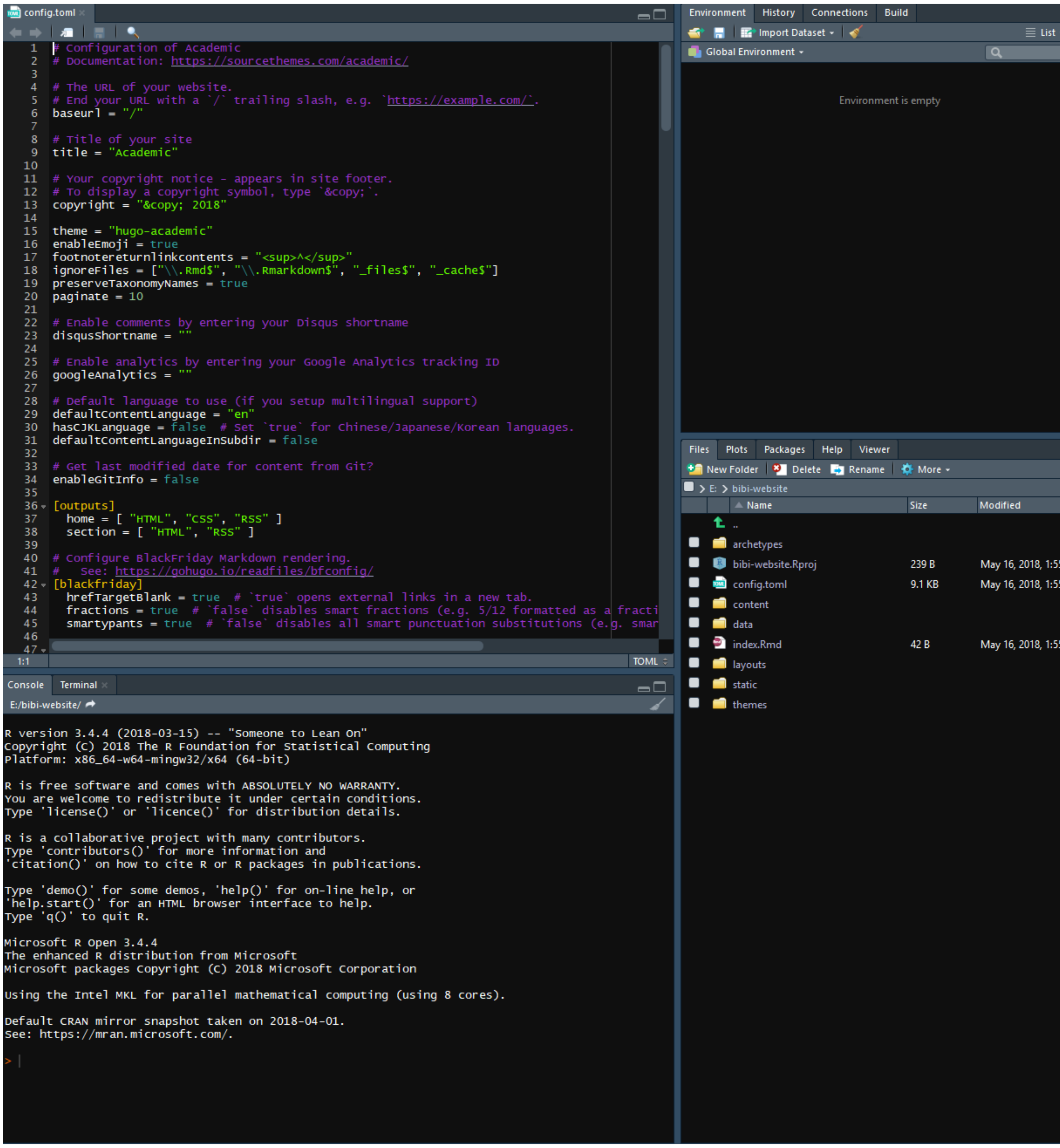
Directory name: bibi-website
Create project as subdirectory of: E:/
☑ Install Hugo automatically
☑ Add sample blog posts
Hugo theme: gcushen/hugo-academic
☑ Add the example site of the theme
☐ Open in new session

Figure 2.2: Create a new directory with a suitable name with no spaces and choose a theme, here we're using `gchusen/hugo-academic`.

website and we choose the theme.

Given Bibi's interests in rheology we'll be using the hugo-academic theme, this is selected by entering `gchusen/hugo-academic` in the box as shown in Figure 2.2

Then `Create Project` and it should download the necessary files, change the working directory to the one created and you should see something like this:



```
# Configuration of Academic
# Documentation: https://sourcethemes.com/academic/

# The URL of your website.
# End your URL with a `/` trailing slash, e.g. `https://example.com/`.
baseurl = "/"

# Title of your site
title = "Academic"

# Your copyright notice - appears in site footer.
# To display a copyright symbol, type `&copy;`.
copyright = "&copy; 2018"

theme = "hugo-academic"
enableEmoji = true
footnotereturnlinkcontents = "<sup>^</sup>"
ignoreFiles = ["\\.Rmd$", "\\.Rmarkdown$", "_files$", "_cache$"]
preserveTaxonomyNames = true
paginate = 10

# Enable comments by entering your Disqus shortname
disqusShortname = ""

# Enable analytics by entering your Google Analytics tracking ID
googleAnalytics = ""

# Default language to use (if you setup multilingual support)
defaultContentLanguage = "en"
hasCJKLanguage = false  # Set `true` for Chinese/Japanese/Korean languages.
defaultContentLanguageInSubdir = false

# Get last modified date for content from Git?
enableGitInfo = false

[outputs]
  home = [ "HTML", "CSS", "RSS" ]
  section = [ "HTML", "RSS" ]

# Configure BlackFriday Markdown rendering.
#   See: https://gohugo.io/readfiles/bfconfig/
[blackfriday]
  hrefTargetBlank = true  # `true` opens external links in a new tab.
  fractions = true  # `false` disables smart fractions (e.g. 5/12 formatted as a fracti
  smartypants = true  # `false` disables all smart punctuation substitutions (e.g. smar
```

```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Microsoft R Open 3.4.4
The enhanced R distribution from Microsoft
Microsoft packages Copyright (C) 2018 Microsoft Corporation

Using the Intel MKL for parallel mathematical computing (using 8 cores).

Default CRAN mirror snapshot taken on 2018-04-01.
See: https://mran.microsoft.com/.

>
```

Name	Size	Modified
..		
archetypes		
bibi-website.Rproj	239 B	May 16, 2018, 1:5
config.toml	9.1 KB	May 16, 2018, 1:5
content		
data		
index.Rmd	42 B	May 16, 2018, 1:5
layouts		
static		
themes		

There are a bunch of folders and files, and the script editor pane is open with the `confog.toml` file open for editing.

As example files are provided we can build a website immediately using:

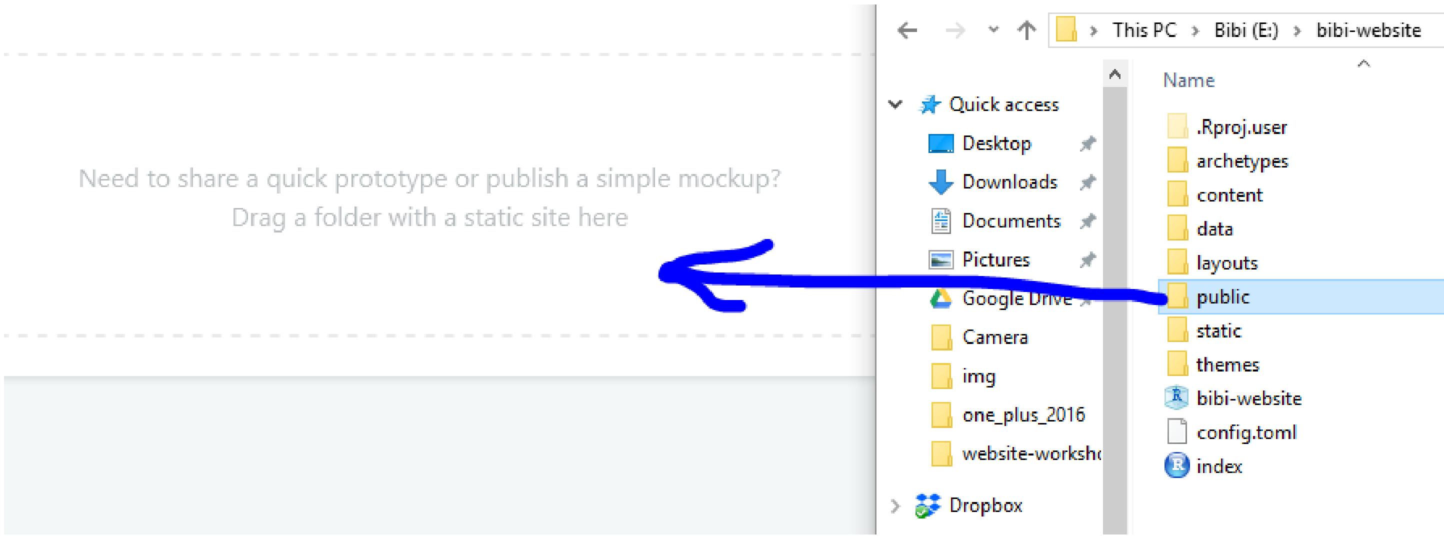
```
blogdown::serve_site()
```

And we should see the example site open in the `Viewer` pane.

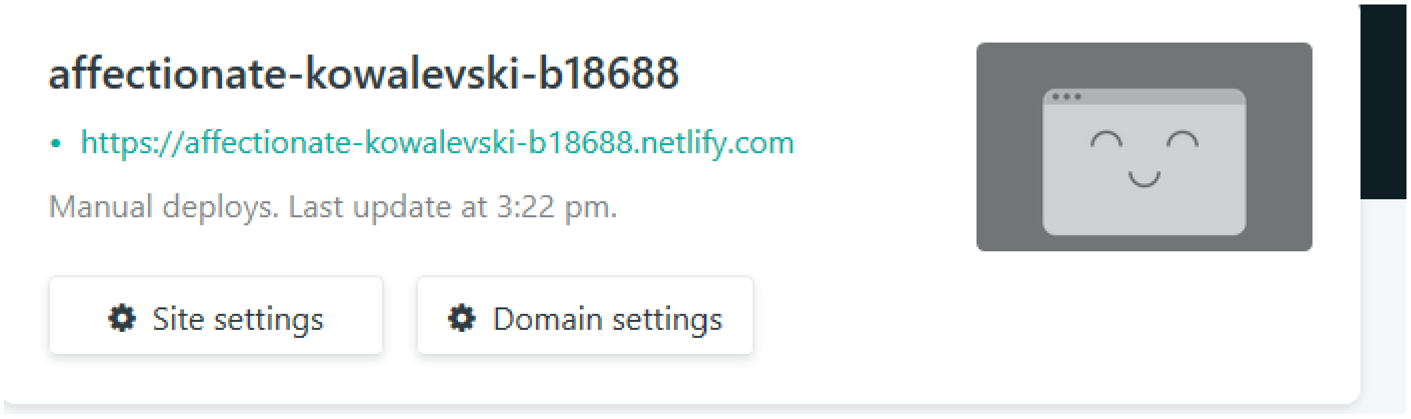
2.2 Deployment

The simplest way to delpoy our website is to use netlify.

Create and account or connect via another account such as GitHub and then from the `Sites` tab that should appear if you click on your name, drag and drop the public folder from the directory into the box like so:



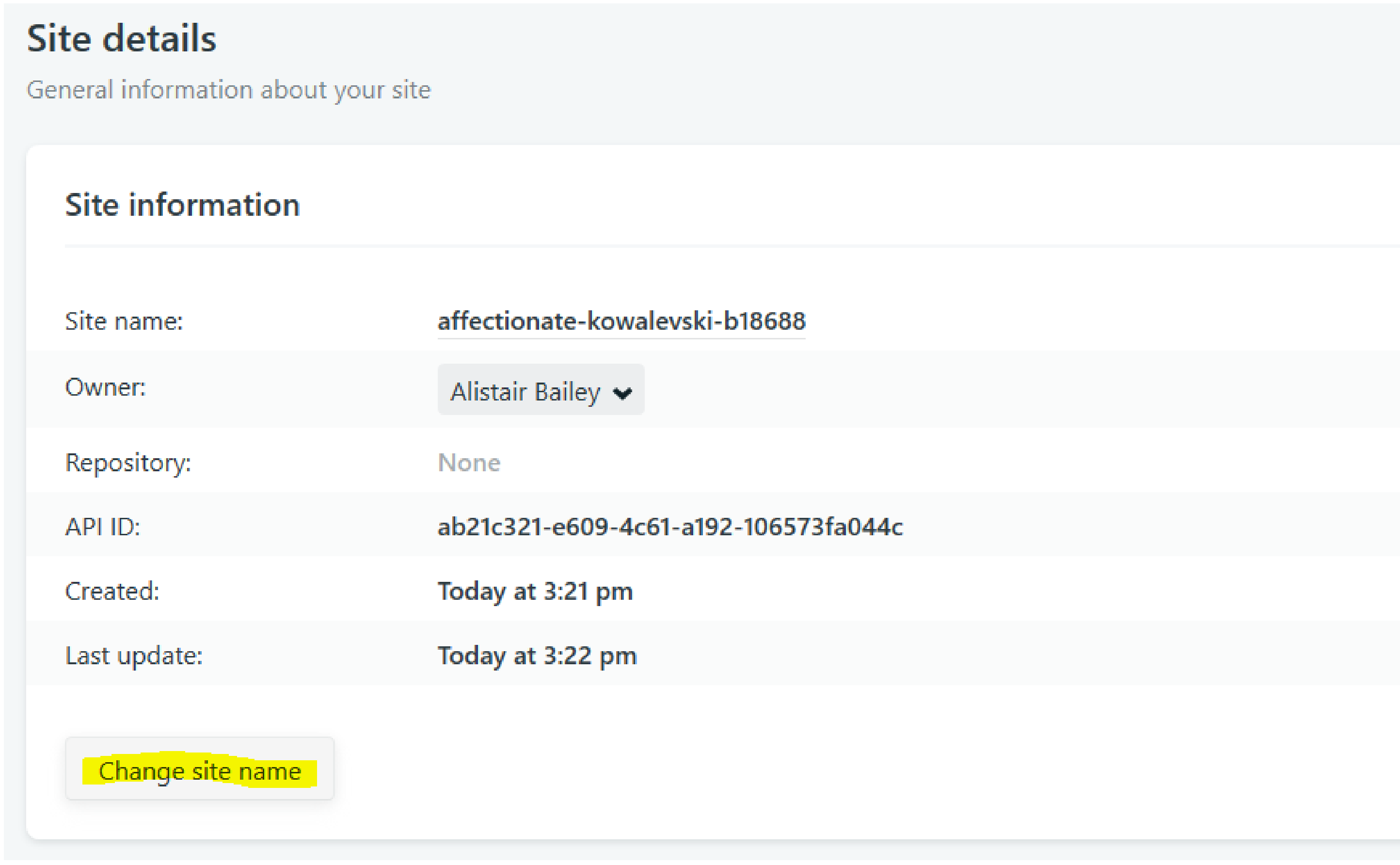
Assuming that goes ok, you'll then see a randomly generated name for your new site. Click on `Site settings` to change the name to whatever you wish.


affectionate-kowalevski-b18688

https://affectionate-kowalevski-b18688.netlify.com

Manual deploys. Last update at 3:22 pm.

You should see the option to change the name like so. Click and follow the instructions.


Site details

General information about your site

Site information

Site name:	affectionate-kowalevski-b18688
Owner:	Alistair Bailey
Repository:	None
API ID:	ab21c321-e609-4c61-a192-106573fa044c
Created:	Today at 3:21 pm
Last update:	Today at 3:22 pm

Click on your name to get back to `Sites` and then click on the name of your website to view it.

Congratulations, you've created and deployed a website.

Next, we'll go back into R to learn how to change the content.

Chapter 3

Adding content to the site

3.1 The file structure in R

The folder containing the published website as we saw in the last chapter is the `public` folder.

The `config.toml` file is where we set the global configurations for the site.

For detail see the TOML syntax blogdown chapter, but most of what we're going to change is quite straightoforward, see Configuration

The `content` folder contains subdirectories containing the files that we create or edit for the sections on the website e.g. publications or project pages.

Images and other files we might want (such a CV) go in the `static` folder and sub-folders respectively. These will then be copied to the `public` folder when we build the site.

We don't need to touch the other folders for the purposes of this tutorial, but as before the blogdown book has all the details.

3.2 Configuration

Here we'll configure the `config.toml` file.

1. First we'll change the title:

```
# Title of your site
title = "Professor Bibi Cat"
```

2. Then we change the details:

```
  # Your details.
  name = "Bibi the Cat"
  role = "Professor of Chaos Theory and Practice"

  # Organizations/Affiliations.
  #   Separate multiple entries with a comma, using the form:
  # `[ {name="Org1", url=""}, {name="Org2", url=""} ]`.
  organizations = [ { name = "Feline University", url = "" } ]
```

3. Next we change the avatar picture by copying an image to `static/img` and either calling it `potrait.jpg` or changing the name in the `config.toml` file. We'll also change the other details, deleting anything we don't want:

```
  gravatar = false  # Get your avatar from Gravatar.com? (true/false)
  avatar = "portrait.jpg"  # Specify an avatar image (in `static/img/` folder)
                           # or delete value to disable avatar.
  email = "bibi@example.org"
  address = "Red Fleecy Blanket, Southampton"
  office_hours = "Whenever I'm hungry"
  phone = ""
  skype = ""
  telegram = ""
```

4. Then we'll change the social media icons to include ORCID, this uses the `ai` icon pack, rather than the `fa` icon pack:

```
[[params.social]]
    icon = "orcid"
    icon_pack = "ai"
    link = "https://orcid.org/0000-000X-XXXX-XXXX"

  [[params.social]]
    icon = "twitter"
    icon_pack = "fa"
    link = "//twitter.com/bibi-the-cat"
```

5. We can add, move or remove links that appear on the homepage of the website. Bibi is too busy sleeping to write blog posts or do any teaching, but she would like to promote her CV which we'll add to the `static/files` folder:

```
[[menu.main]]
  name = "Home"
  url = "#about"
  weight = 1

[[menu.main]]
  name = "Publications"
  url = "#publications"
  weight = 2

[[menu.main]]
  name = "CV"
  url = "/files/cv.pdf"
  weight = 3

[[menu.main]]
```

```
  name = "Projects"
  url = "#projects"
  weight = 4


[[menu.main]]
  name = "Contact"
  url = "#contact"
  weight = 6
```

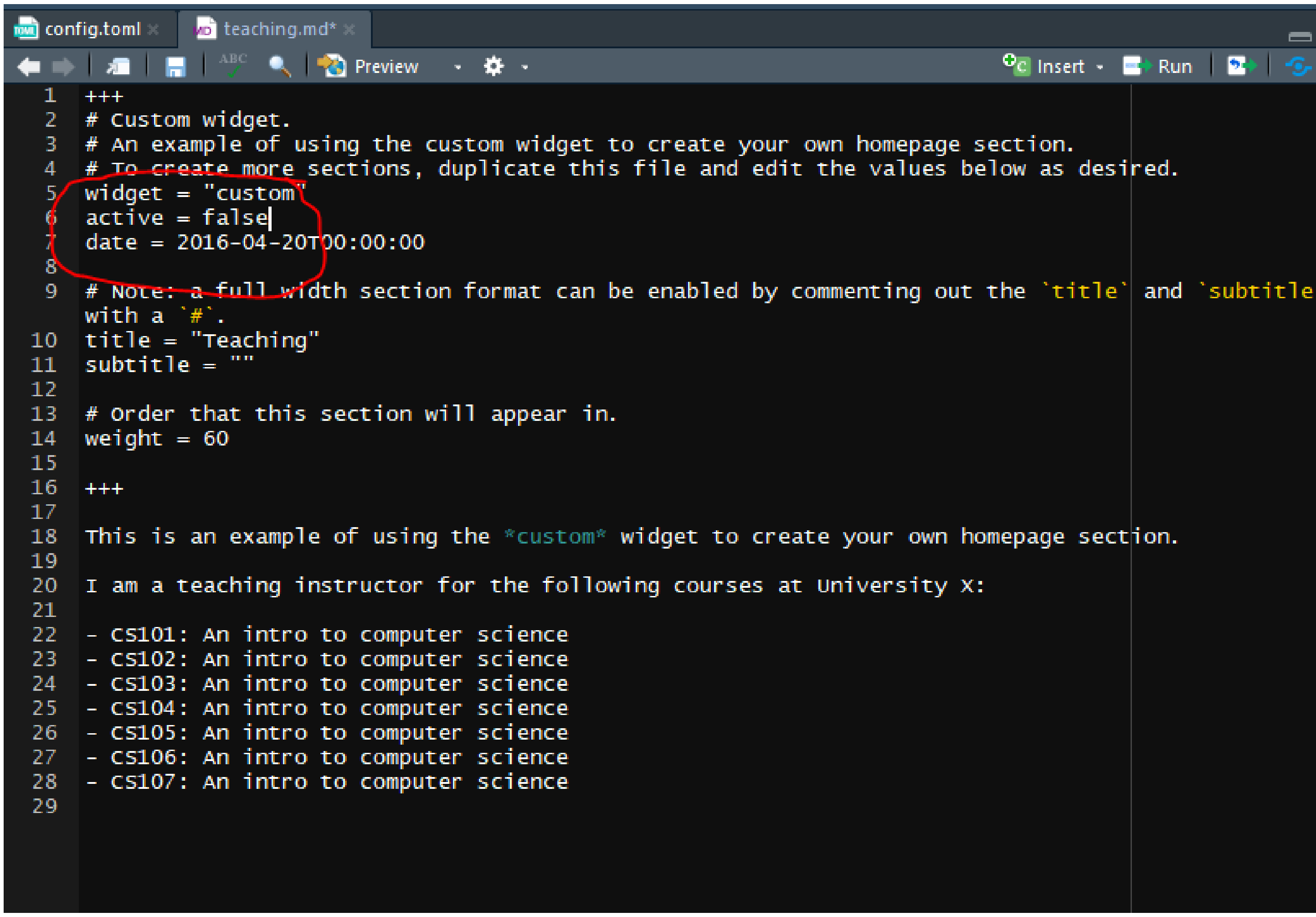
Explore to find out what else you can change, such as the publication format.

3.3 Choosing sections and editing the biography

In the `content/home` folder are a series of files which configure the sections widgets.

To turn a section off, open the relevant file and change `active = true` to `active = false`.

For example, Bibi is far too busy sleeping to do any teaching, so we'll turn of the teaching widget by opening `teaching.md` and changing the `active` status.



```
+++
# 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 = false
date = 2016-04-20T00:00:00

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

# Order that this section will appear in.
weight = 60

+++

This is an example of using the *custom* widget to create your own homepage section.

I am a teaching instructor for the following courses at University X:

- CS101: An intro to computer science
- CS102: An intro to computer science
- CS103: An intro to computer science
- CS104: An intro to computer science
- CS105: An intro to computer science
- CS106: An intro to computer science
- CS107: An intro to computer science
```

Let's do this for `hero`, `publications_selected`, `posts`,`talks` and `teaching`.

And now look at the updated site. `hero` controlled the top banner, and `publications` is where the link on our menu bar links to.

3.4 Editing section content

The template files in `content/home` are written in markdown, lightweight markup language, where for example # indicates Heading 1 and ## Heading 2. See the markdown cheatsheet to quickly understand the syntax.

You can also write Rmarkdown files here, we're not going to, but see here for details.

Starting with the `about` file, the bit between the +++ symbols is for the `about` widget that creates the interests and education bit on the homepage.

```
+++
# About/Biography widget.
widget = "about"
active = true
date = 2016-04-20T00:00:00

# Order that this section will appear in.
weight = 5

# List your academic interests.
[interests]
  interests = [
    "Sleeping",
    "Cardboard boxes and bags",
    "Dreamies"
  ]

# List your qualifications (such as academic degrees).
[[education.courses]]
  course = "PhD in Causing Chaos"
  institution = "University of Life"
  year = 2012

[[education.courses]]
  course = "MEng in Cardboard Box Destruction"
  institution = "University of Life"
  year = 2009
```

```
[[education.courses]]
  course = "BSc in Covering Everything in Hair"
  institution = "University of Life"
  year = 2008

+++

# Biography

Bibi the Cat is a Professor in Chaos Theory and Practice. 90% of her time is
spent snoozing, whilst she devotes the other 10% to destroying things and
eating tasty treats. Don't call her, she'll call you.
```

3.5 Creating Projects content

Now if we go up to the `content` directory you'll see we have folders for `projects` and `publication`.

Let's go into `content/project` and open `deep-learning.md` and edit it, starting with the widget section to change:

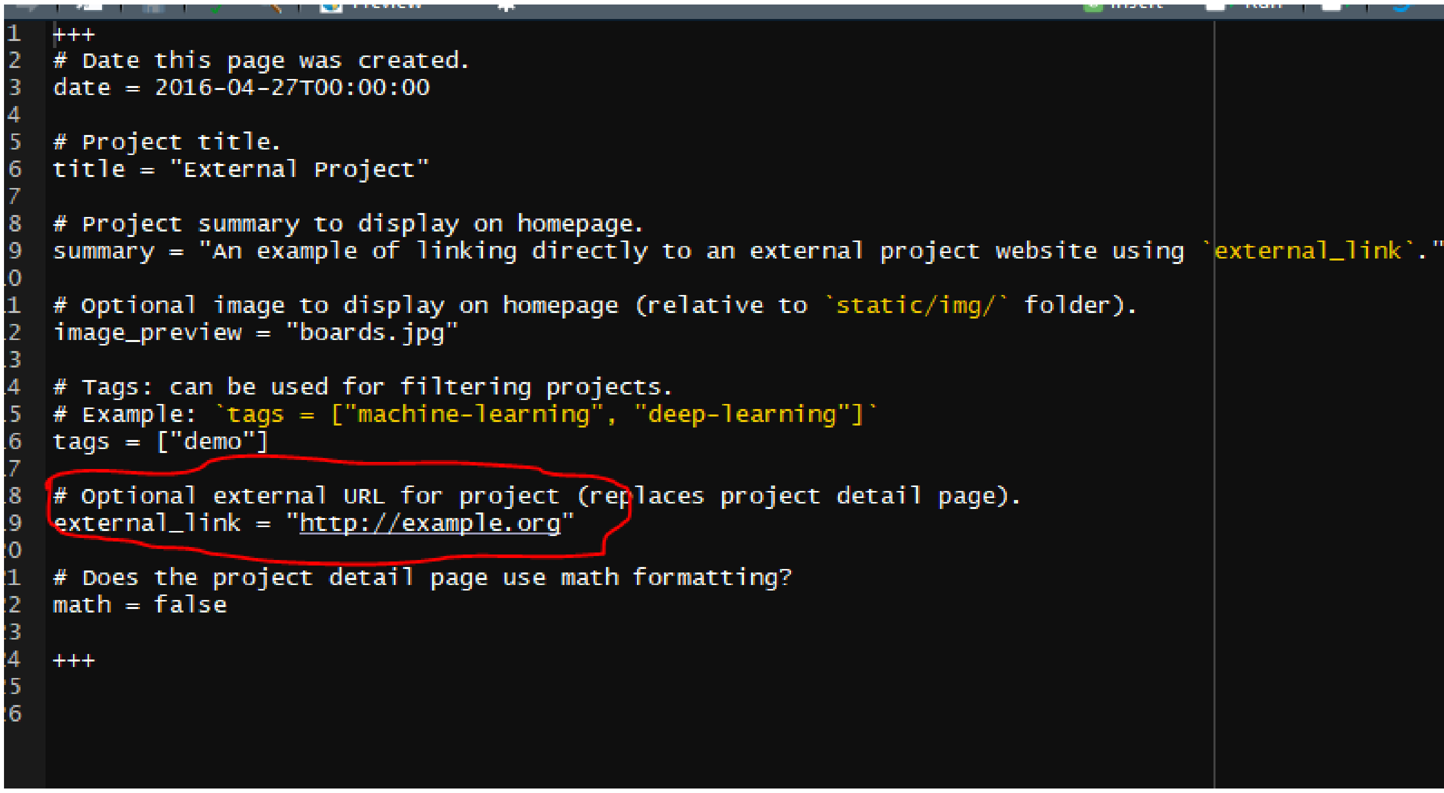
1. the date
2. the title
3. the summary
4. the preview image to the one in `static/img`
5. the tags
6. the header image also to ehe one in `static/img`

And then write whatever we want to about the project, below the `+++` , I've added some markdown for another image also in the `static/img/` folder so we have this:

```
 +++
# Date this page was created.
date = 2018-05-17T00:00:00

# Project title.
title = "Bags"

# Project summary to display on homepage.
summary = "Bibi loves to get into bags"

# Optional image to display on homepage (relative to `static/img/` folder).
image_preview = "bibi-bag.jpg"

# Tags: can be used for filtering projects.
# Example: `tags = ["machine-learning", "deep-learning"]`
tags = ["bags"]

# Optional external URL for project (replaces project detail page).
external_link = ""

# Does the project detail page use math formatting?
math = false

# Optional featured image (relative to `static/img/` folder).
[header]
image = "bibi-bag.jpg"
caption = "Bibi in a bag"

+++
I love bags, but also boxes. In fact anything I can get inside, especially
if you don't want me to.


![Bibi in a box](/img/bibi-box.jpg)
```

I then save this as `bags.md` and delete the `deep-learning.md` file.

I'll leave you to explore the external link example, but it requires editing the widget as before for title, date and images, and then changing the link to your external webpage.



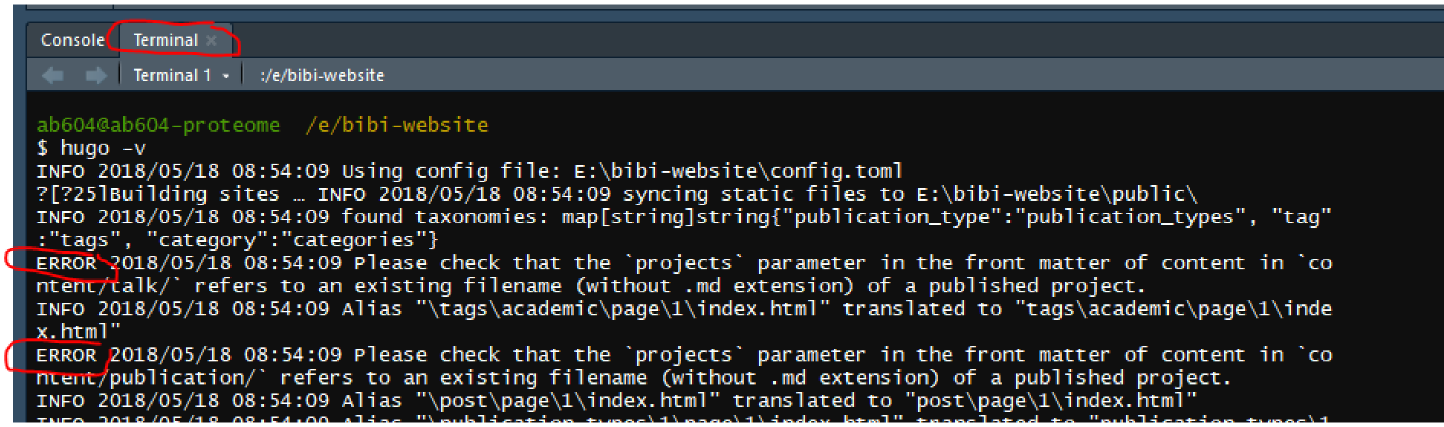
```
+++
# Date this page was created.
date = 2016-04-27T00:00:00

# Project title.
title = "External Project"

# Project summary to display on homepage.
summary = "An example of linking directly to an external project website using `external_link`."

# Optional image to display on homepage (relative to `static/img/` folder).
image_preview = "boards.jpg"

# Tags: can be used for filtering projects.
# Example: `tags = ["machine-learning", "deep-learning"]`
tags = ["demo"]

# Optional external URL for project (replaces project detail page).
external_link = "http://example.org"

# Does the project detail page use math formatting?
math = false

+++
```

In the next chapter we'll look at adding publications to the site.

3.6 Troubleshooting

When you start changing things, you may find that the site stops automatically updating. This indicates an error.

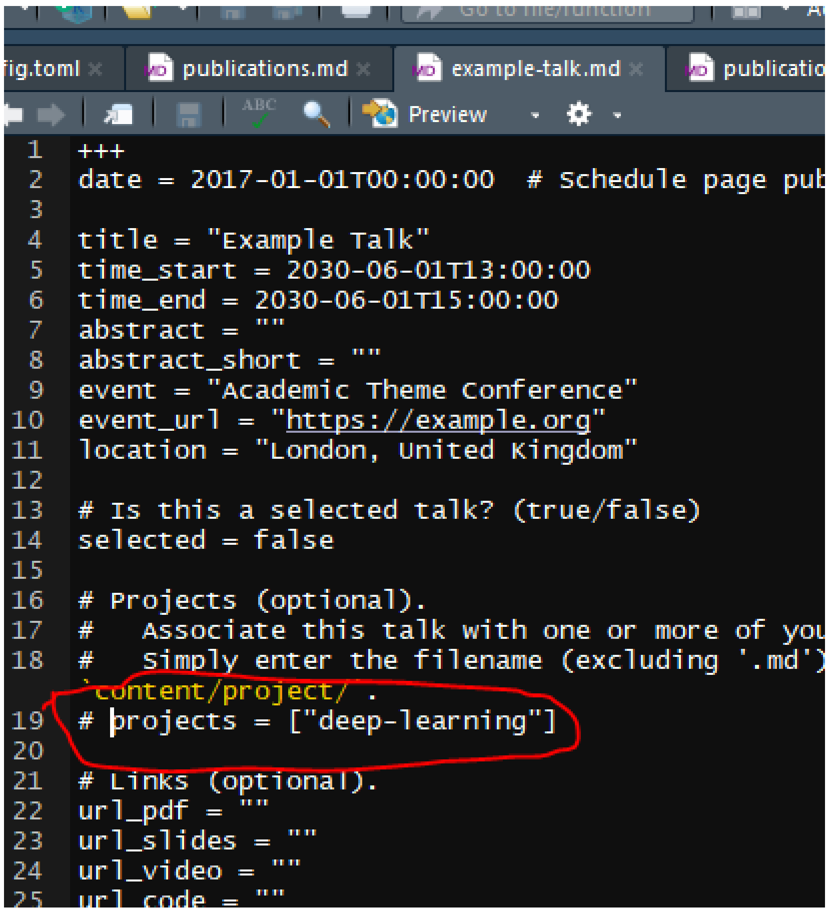
To find out what is wrong, go to the `Terminal` tab in Rstudio and type `hugo -v` and try to figure it out.



```
ab604@ab604-proteome  /e/bibi-website
$ hugo -v
INFO 2018/05/18 08:54:09 Using config file: E:\bibi-website\config.toml
?[?25lBuilding sites … INFO 2018/05/18 08:54:09 syncing static files to E:\bibi-website\public\
INFO 2018/05/18 08:54:09 found taxonomies: map[string]string{"publication_type":"publication_types", "tag"
:"tags", "category":"categories"}
ERROR 2018/05/18 08:54:09 Please check that the `projects` parameter in the front matter of content in `co
ntent/talk/` refers to an existing filename (without .md extension) of a published project.
INFO 2018/05/18 08:54:09 Alias "\tags\academic\page\1\index.html" translated to "tags\academic\page\1\inde
x.html"
ERROR 2018/05/18 08:54:09 Please check that the `projects` parameter in the front matter of content in `co
ntent/publication/` refers to an existing filename (without .md extension) of a published project.
INFO 2018/05/18 08:54:09 Alias "\post\page\1\index.html" translated to "post\page\1\index.html"
```

Here, I've removed `deep-learning.md` from the project folder, but the error indicated there are references to it in other files that I needed to amend.

For the first error, I opened the `content/talk/` folder and in the `example-talk.md` file I see that line 19 has `projects = ["deep-learnig"]` . So I commented it out with a # symbol.



```
+++
date = 2017-01-01T00:00:00  # Schedule page pub

title = "Example Talk"
time_start = 2030-06-01T13:00:00
time_end = 2030-06-01T15:00:00
abstract = ""
abstract_short = ""
event = "Academic Theme Conference"
event_url = "https://example.org"
location = "London, United Kingdom"

# Is this a selected talk? (true/false)
selected = false

# Projects (optional).
#   Associate this talk with one or more of you
#   Simply enter the filename (excluding '.md')
`content/project/`.
# projects = ["deep-learning"]

# Links (optional).
url_pdf = ""
url_slides = ""
url_video = ""
url_code = ""
```

Follow the same approach for all errors until when you run `hugo -v` there are no more, and the site should now build.

Chapter 4

Creating publication files

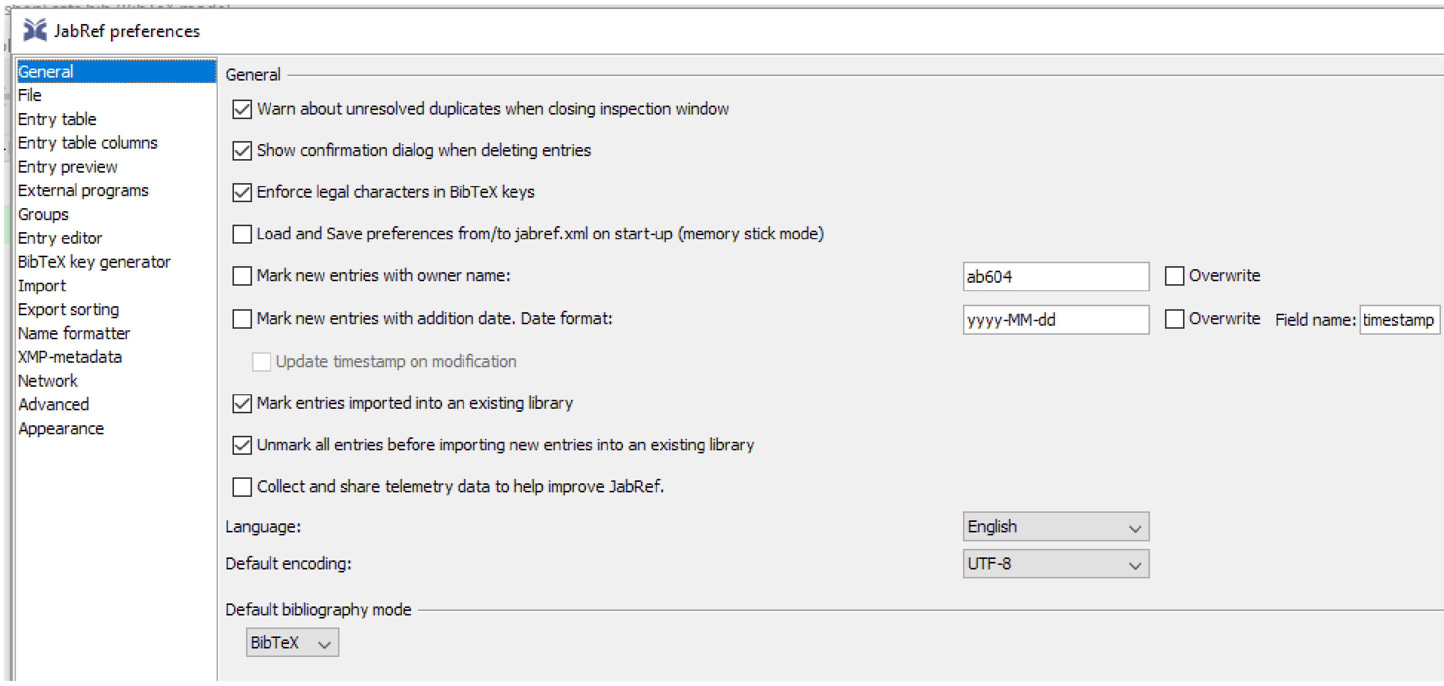
This is probably the most fiddly part. In the `content/publication` folder we need `.md` files for each publication we want to add to our site.

Bibi being quite a lazy cat only has one publication, but we'll look at how to automate the process for many.

4.1 Citation format

To add publication citations, we first need them in `bibtex` format. This can be done using JabRef.

1. Open JabRef and go to `Options > Preferences > General` and ensure `Default encoding` is set to `UTF-8`



General

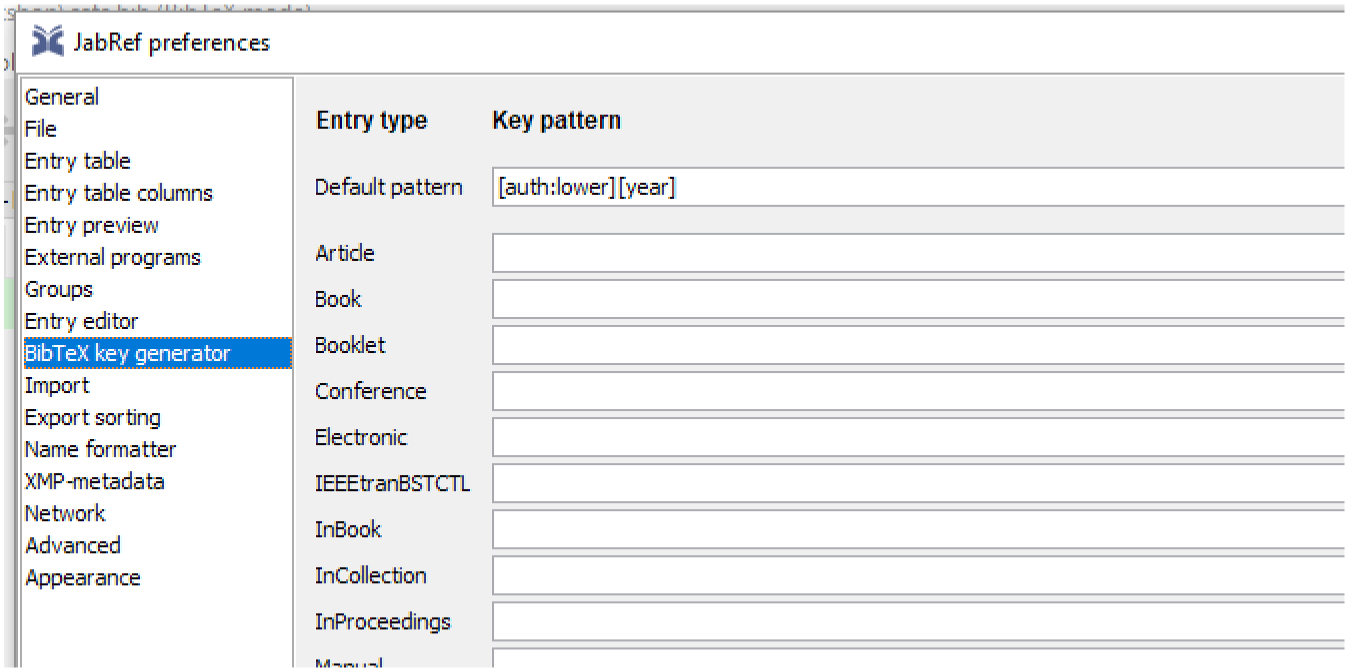
- [✓] Warn about unresolved duplicates when closing inspection window
- [✓] Show confirmation dialog when deleting entries
- [✓] Enforce legal characters in BibTeX keys
- [ ] Load and Save preferences from/to jabref.xml on start-up (memory stick mode)
- [ ] Mark new entries with owner name: ab604 [ ] Overwrite
- [ ] Mark new entries with addition date. Date format: yyyy-MM-dd [ ] Overwrite Field name: timestamp
- [ ] Update timestamp on modification
- [✓] Mark entries imported into an existing library
- [✓] Unmark all entries before importing new entries into an existing library
- [ ] Collect and share telemetry data to help improve JabRef.

Language: English

Default encoding: UTF-8

Default bibliography mode: BibTeX

2. Then go to `Options > Preferences > Bibtex key generator` and set this to `[auth:lower][year]` and then `OK`.

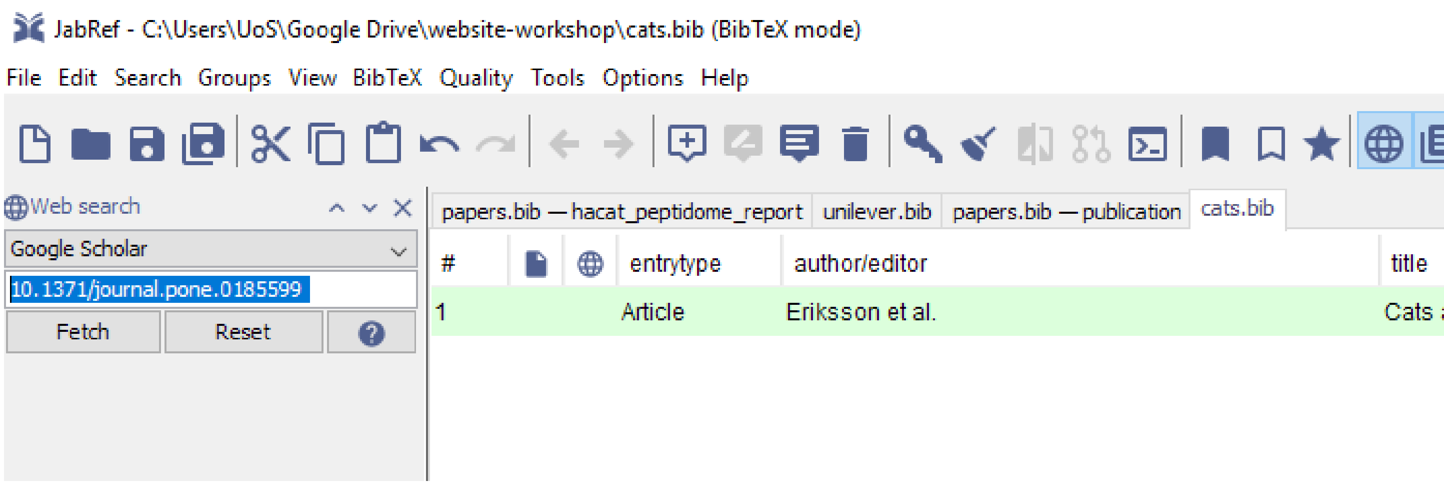


Entry type	Key pattern
Default pattern	[auth:lower][year]
Article	
Book	
Booklet	
Conference	
Electronic	
IEEEtranBSTCTL	
InBook	
InCollection	
InProceedings	

3. Then create a library `File > New Bibtex Library`.

4. If you have exported your publications from EndNote or other software as bibtex you can import this into your new library and then highlight the list and click on the key icon to generate bibtex keys.

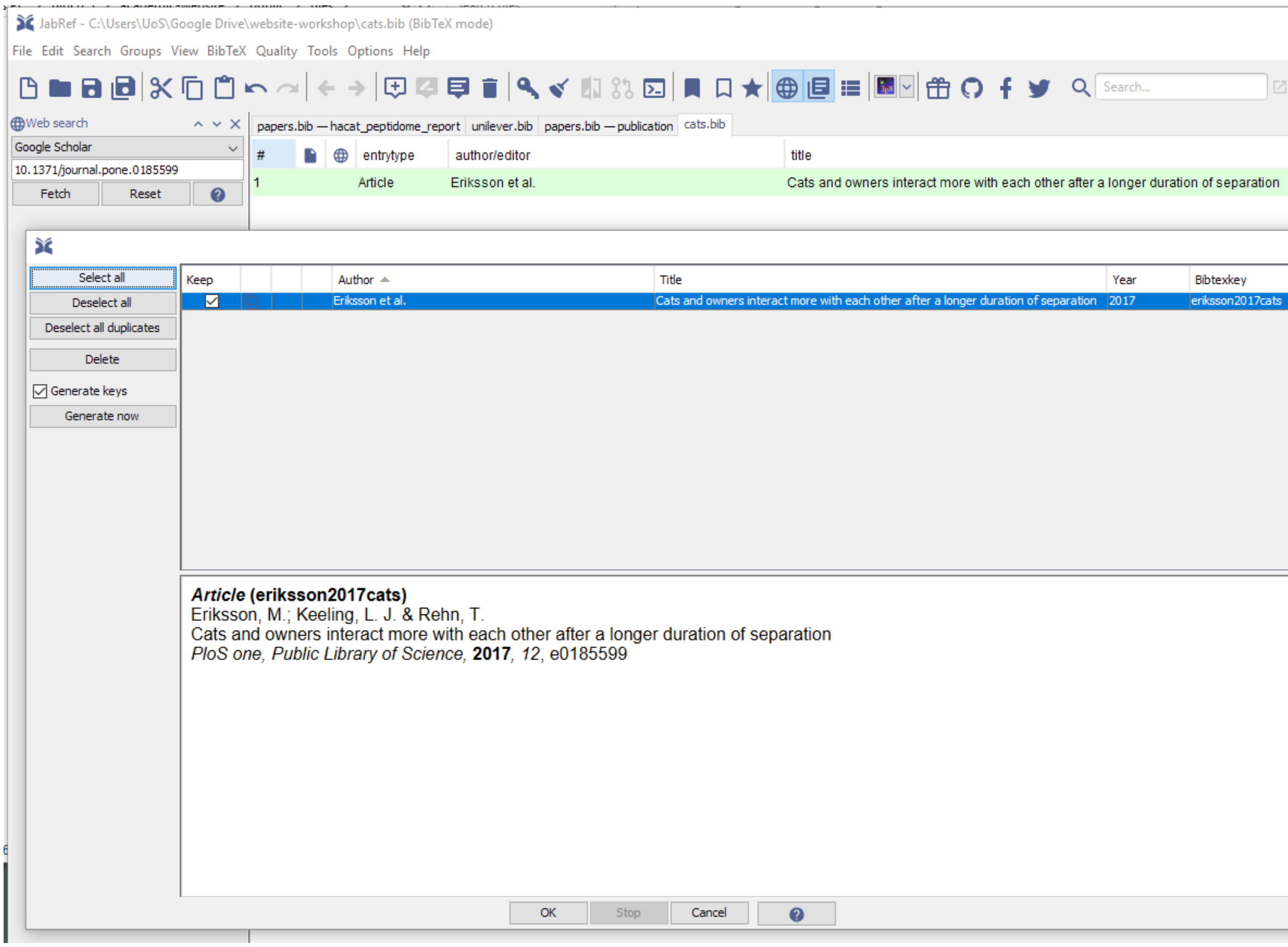
5. Alternatively, we can generate the list using the search function, here I entered a DOI and searched with Google Scholar:


JabRef - C:\Users\UoS\Google Drive\website-workshop\cats.bib (BibTeX mode)
File Edit Search Groups View BibTeX Quality Tools Options Help
Web search
Google Scholar
Fetch
Reset
papers.bib — hacat_peptidome_report
unilever.bib
papers.bib — publication
cats.bib

10.1371/journal.pone.0185599

#			entrytype	author/editor	title
1			Article	Eriksson et al.	Cats

Then I selected the publication and generated the key.


JabRef - C:\Users\UoS\Google Drive\website-workshop\cats.bib (BibTeX mode)
File Edit Search Groups View BibTeX Quality Tools Options Help
Web search
Google Scholar
10.1371/journal.pone.0185599
Fetch
Reset
papers.bib — hacat_peptidome_report
unilever.bib
papers.bib — publication
cats.bib
Select all
Deselect all
Deselect all duplicates
Delete
Generate keys
Generate now
OK
Stop
Cancel

#	entrytype	author/editor	title
1	Article	Eriksson et al.	Cats and owners interact more with each other after a longer duration of separation

Keep	Author	Title	Year	Bibtexkey
✓	Eriksson et al.	Cats and owners interact more with each other after a longer duration of separation	2017	eriksson2017cats

***Article* (eriksson2017cats)**
Eriksson, M.; Keeling, L. J. & Rehn, T.
Cats and owners interact more with each other after a longer duration of separation
PloS one, *Public Library of Science*, **2017**, *12*, e0185599

6. Now we save the bibtex library. Here it is `cats.bib`. It's just another text file so can be viewed in any text editor.

4.2 Converting Bibtex files

Converting to markdown requires some effort, fortunately Lorenzo Buesetto wrote a function that required a small hack to work with JabRef.

I found the function still may output files that need a bit of cleaning up, but generally it works really well.

We need the `tidyverse`, `RefManageR`, `anytime` packages installed.

```
bibtex_2academic <- function(bibfile,
                             outfold,
                             abstract = FALSE,
                             overwrite = FALSE) {
                             require(RefManageR)
                             require(dplyr)
                             require(stringr)
                             require(anytime)

                             # Import the bibtex file and convert to data.frame
                             mypubs   <-
                             ReadBib(bibfile, check = "warn",
                                     .Encoding = "UTF-8") %>%
                             as.data.frame()

                             # assign "categories" to the different types of
                             # publications
                             mypubs   <- mypubs %>%
                             dplyr::mutate(
                             pubtype = dplyr::case_when(
                             bibtype == "Article" ~ "2",
                             bibtype == "Article in Press" ~ "2",
                             bibtype == "InProceedings" ~ "1",
                             bibtype == "Proceedings" ~ "1",
                             bibtype == "Conference" ~ "1",
                             bibtype == "Conference Paper" ~ "1",
                             bibtype == "MastersThesis" ~ "3",
                             bibtype == "PhdThesis" ~ "3",
                             bibtype == "Manual" ~ "4",
```

```r
bibtype == "TechReport" ~ "4",
bibtype == "Book" ~ "5",
bibtype == "InCollection" ~ "6",
bibtype == "InBook" ~ "6",
bibtype == "Misc" ~ "0",
TRUE ~ "0"
)
)

# create a function which populates the md template
# based on the info
# about a publication
create_md <- function(x) {
# define a date and create filename by appending date
# and start of title
if (!is.na(x[["year"]])) {
x[["date"]] <- paste0(x[["year"]], "-01-01")
} else {
x[["date"]] <- "2999-01-01"
}

filename <- paste(
x[["date"]],
x[["title"]] %>%
str_replace_all(fixed(" "), "_") %>%
str_remove_all(fixed(":")) %>%
str_sub(1, 20) %>%
paste0(".md"),
sep = "_"
)
```

```
# start writing
if (!file.exists(file.path(outfold, filename)) |
overwrite) {
fileConn <- file.path(outfold, filename)
write("+++", fileConn)

# Title and date
write(paste0("title = \"", x[["title"]], "\""),
fileConn,
append = T)
write(paste0("date = \"", anydate(x[["date"]]), "\""),
fileConn,
append = T)

# Authors. Comma separated list, e.g. `["Bob Smith",
# "David Jones"]`.
auth_hugo <-
str_replace_all(x["author"], " and ", "\", \"")
auth_hugo <-
stringi::stri_trans_general(auth_hugo, "latin-ascii")
write(paste0("authors = [\"", auth_hugo, "\"]"),
fileConn,
append = T)

# Publication type. Legend:
# 0 = Uncategorized, 1 = Conference paper,
# 2 = Journal article
# 3 = Manuscript, 4 = Report, 5 = Book,  6 = Book
# section
write(paste0("publication_types = [\"", x[["pubtype"]],
             "\"]"),
fileConn,
append = T)

# Publication details: journal, volume, issue,
# page numbers and doi link
publication <- x[["journal"]]
if (!is.na(x[["volume"]]))
publication <- paste0(publication,
", (", x[["volume"]], ")")
if (!is.na(x[["pages"]]))
publication <- paste0(publication,
", _pp. ", x[["pages"]], "_")
if (!is.na(x[["doi"]]))
publication <- paste0(publication,
", ",
paste0("https://doi.org/",
x[["doi"]]))

write(paste0("publication = \"", publication, "\""),
fileConn,
append = T)
write(paste0("publication_short = \"",
publication,
"\""),
fileConn,
append = T)

# Abstract and optional shortened version.
```

```r
if (abstract) {
write(paste0("abstract = \"", x[["abstract"]], "\""),
fileConn,
append = T)
} else {
write("abstract = \"\"",
fileConn,
append = T)
}
write(paste0("abstract_short = \"", "\""),
fileConn,
append = T)

# other possible fields are kept empty. They can be
# customized later by
# editing the created md

write("image_preview = \"\"",
fileConn,
append = T)
write("selected = false", fileConn, append = T)
write("projects = []", fileConn, append = T)
write("tags = []", fileConn, append = T)
#links
write("url_pdf = \"\"", fileConn, append = T)
write("url_preprint = \"\"",
fileConn,
append = T)
write("url_code = \"\"", fileConn, append = T)
write("url_dataset = \"\"",
fileConn,
append = T)
write("url_project = \"\"",
fileConn,
append = T)
write("url_slides = \"\"", fileConn, append = T)
write("url_video = \"\"", fileConn, append = T)
write("url_poster = \"\"", fileConn, append = T)
write("url_source = \"\"", fileConn, append = T)
#other stuff
write("math = true", fileConn, append = T)
write("highlight = true", fileConn, append = T)
# Featured image
write("[header]", fileConn, append = T)
write("image = \"\"", fileConn, append = T)
write("caption = \"\"", fileConn, append = T)

write("+++", fileConn, append = T)
}
}
# apply the "create_md" function over the
# publications list to generate
# the different "md" files.

apply(
mypubs,
FUN = function(x)
create_md(x),
                             MARGIN = 1
                             )
                             }
```

To use this function, save it as `bibtex_2academic.R` and then load the fcuntion into your R environment using `source("bibtex_2academic.R")`.

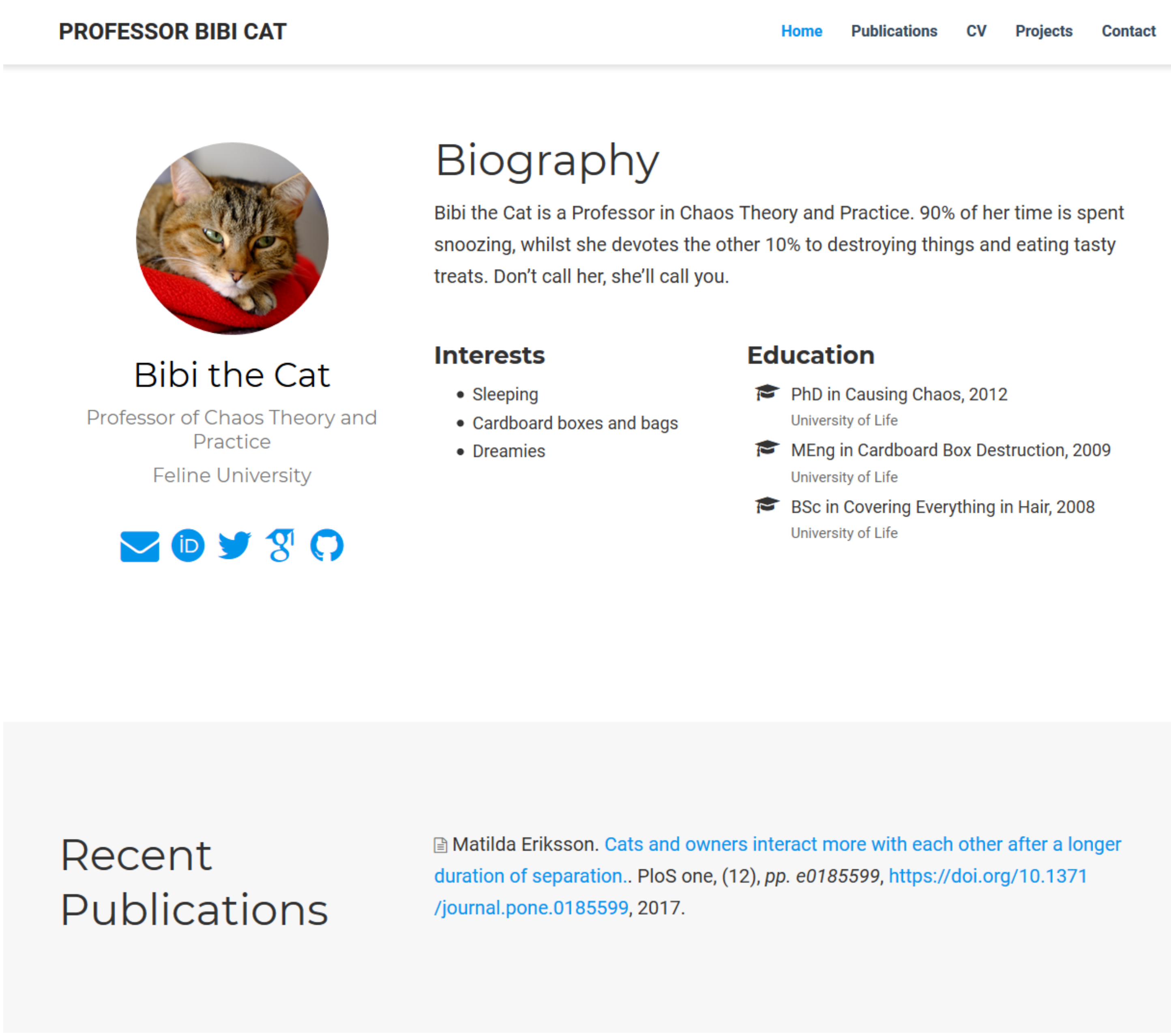
Then assuming you have a JabRef outputted Bixbtex file, here `cats.bib` we need to assign variables for the bibtex file and the output location, which in this case will be `content/publication`. Then we use these variables as arguement to the conversion function:

```
# Bibtex file in my directory
my_bibfile <- "cats.bib"
# Where I want the markdown output to go
outfold <- "content/publication"
# Use the conversion function
bibtex_2academic <- function(my_bibfile,
                             outfold,
                             abstract = FALSE,
                             overwrite = FALSE
                             )
```

All being well, we should now have a markdown file for each publication (we only had one in this example) in the `content/publication`. It may need some manual tweaking if the format on the webpage isn't quite right.

We can remove the example files that came with the template.

Bibi should now have her website configured and what's left is to re-deploy the completed version which should look like this:



Biography

Bibi the Cat is a Professor in Chaos Theory and Practice. 90% of her time is spent snoozing, whilst she devotes the other 10% to destroying things and eating tasty treats. Don't call her, she'll call you.

Bibi the Cat

Professor of Chaos Theory and Practice

Feline University

Interests

- Sleeping
- Cardboard boxes and bags
- Dreamies

Education

- PhD in Causing Chaos, 2012
  University of Life
- MEng in Cardboard Box Destruction, 2009
  University of Life
- BSc in Covering Everything in Hair, 2008
  University of Life

Recent Publications

Matilda Eriksson. Cats and owners interact more with each other after a longer duration of separation.. PloS one, (12), *pp. e0185599*, https://doi.org/10.1371/journal.pone.0185599, 2017.

Chapter 5

Automating deployment with GitHub

In Chapter 2 we used drag and drop of the public folder, but if you are familiar with version control (and if you aren't it's definitely worth learning), then you can automate deployment using GitHub.

The details are in the blogdown book, but essentially you can link netlify to your GitHub account and deploy to netlify from there. This means that anytime you update your site in Rstudio, you can push your files to GitHub and your site will be deployed automatically.

Basically, this means you are tracking your changes and deploying in one step, which is both quicker and means it's easy to revert should you need to.

References

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Xie, Y. (2018). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.7.