

```
library(blogdown)

rladies_global %>%
  filter(city == 'Tbilisi')
```



Creating a website / blog

Steps to take

1. Install blogdown package
2. Choose a Hugo (site generator) theme
3. Customise your site
4. Serve site
5. Publish your site, eg through Github Pages or Netlify (not covered in this meetup)

Blogdown + Hugo

Creating a website/ blog has been made relatively easy with [blogdown](#) package (using rmarkdown) and [Hugo](#) site generator.



+



= blogdown



Awesome blogdown

Full list and how to add own blog to the list on the Awesome Blogdown [site](#) and its [github](#)

Blogdown [demo site](#)

Blogdown + Hugo

- Building a static website - website only consists of static files such as HTML, CSS, JavaScript, and images, etc
- Host on any website (does not require PHP)
- Generated from (R) Markdown documents - can turn posts into PDF, easily reproducible, etc.
- Different ways to generate the site (Hugo, Jekyll)

Read more here: <https://bookdown.org/yihui/blogdown/>



Install blogdown

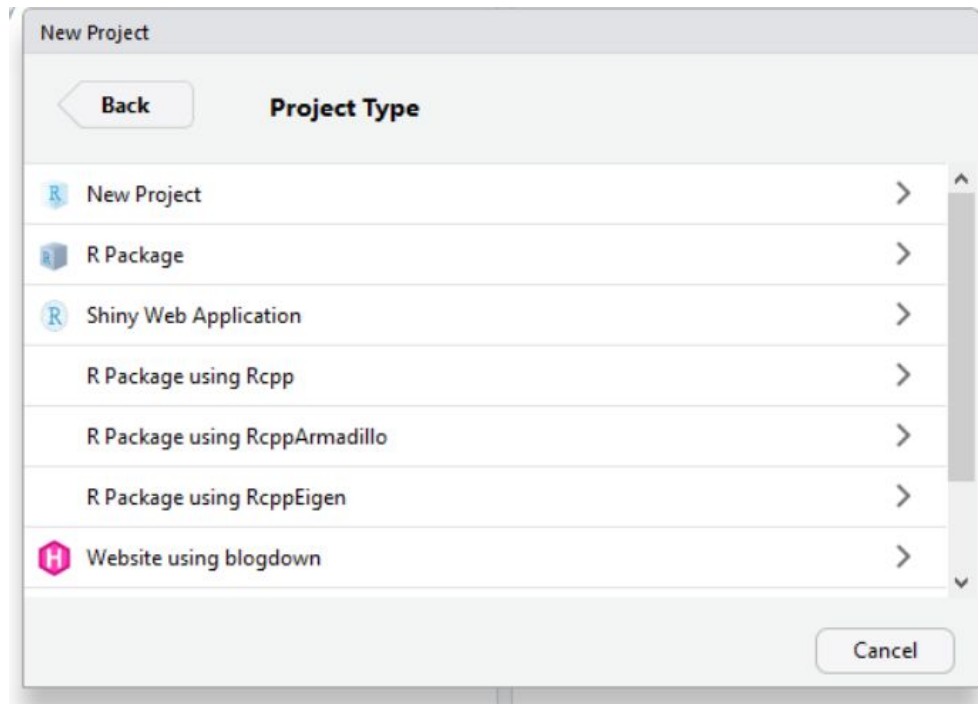
```
install.packages("blogdown")
```

```
library(blogdown)
```

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

Or:

- 1) File → New Project → New Directory → New project
- 2) → Website using blogdown (if have blogdown installed)



Choose the design or layout (Hugo theme)



1. Blogdown automatically installs Hugo
2. Choose a Hugo theme:
<https://themes.gohugo.io/>
(default: Lithium)
3. To insert the chosen theme in RStudio, copy the respective link, e.g.
“gcushen/hugo-academic”



The image shows the 'New Project' dialog in RStudio, specifically the 'Create a new website using Hugo and blogdown' tab. The dialog has a 'Back' button and a title bar. It contains several input fields and checkboxes. The 'Directory name' field is empty. The 'Create project as subdirectory of:' field is set to '~/RLadies' with a 'Browse...' button. There are three checked checkboxes: 'Install Hugo automatically', 'Add sample blog posts', and 'Add the example site of the theme'. The 'Hugo theme' field is set to 'yihui/hugo-lithium-theme' and is highlighted with a dashed border. At the bottom, there is an unchecked checkbox for 'Open in new session' and two buttons: 'Create Project' and 'Cancel'.

New Project

Back Create a new website using Hugo and blogdown

Directory name:

Create project as subdirectory of:

~/RLadies Browse...

☒ Install Hugo automatically

☒ Add sample blog posts

Hugo theme: yihui/hugo-lithium-theme

☒ Add the example site of the theme

☐ Open in new session

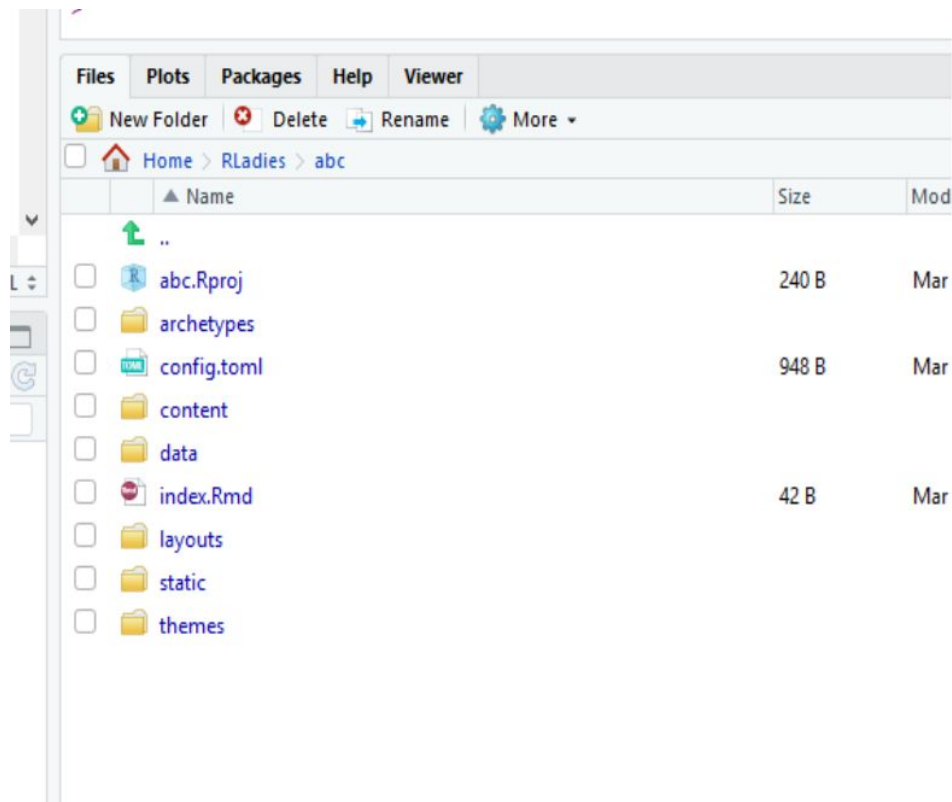
Create Project Cancel

Structure of the site



- Content of the website is created in the files of the project. The structure of each theme may vary.
- If using a theme connected to a github repo → don't edit the theme folder directly.

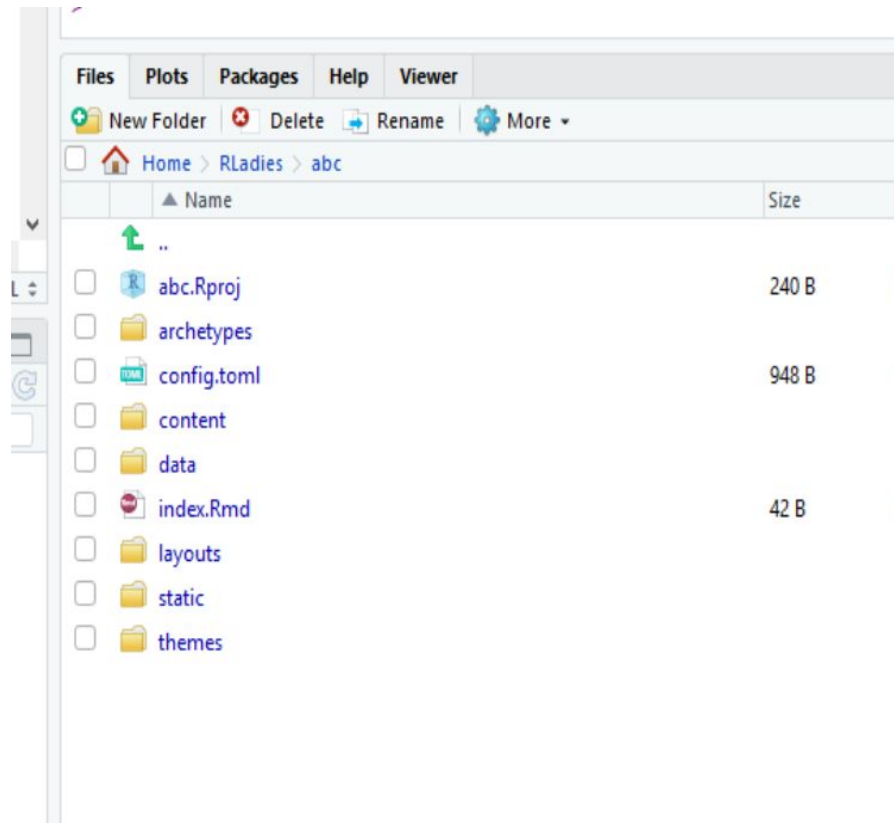
Take a look into the folders!



Customise your site



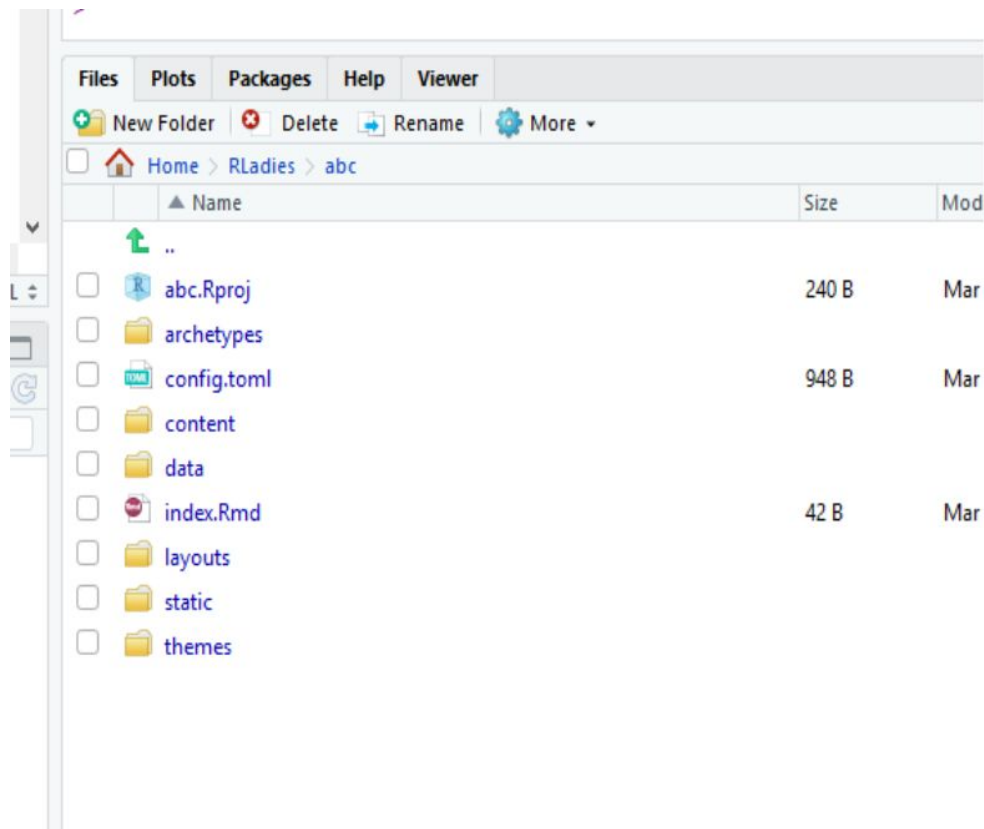
1. *config.toml* - the main structure of your site
2. *Content* folder includes your posts
3. Everything in the *public* folder will be published on the site
4. *Static* - the core of your site, preserves static content (images, CSS, JavaScript)
5. *Themes* - replicates files in the project root



Empty folders in the root



1. *archetypes*
2. *data*
3. *layouts*
4. *static*

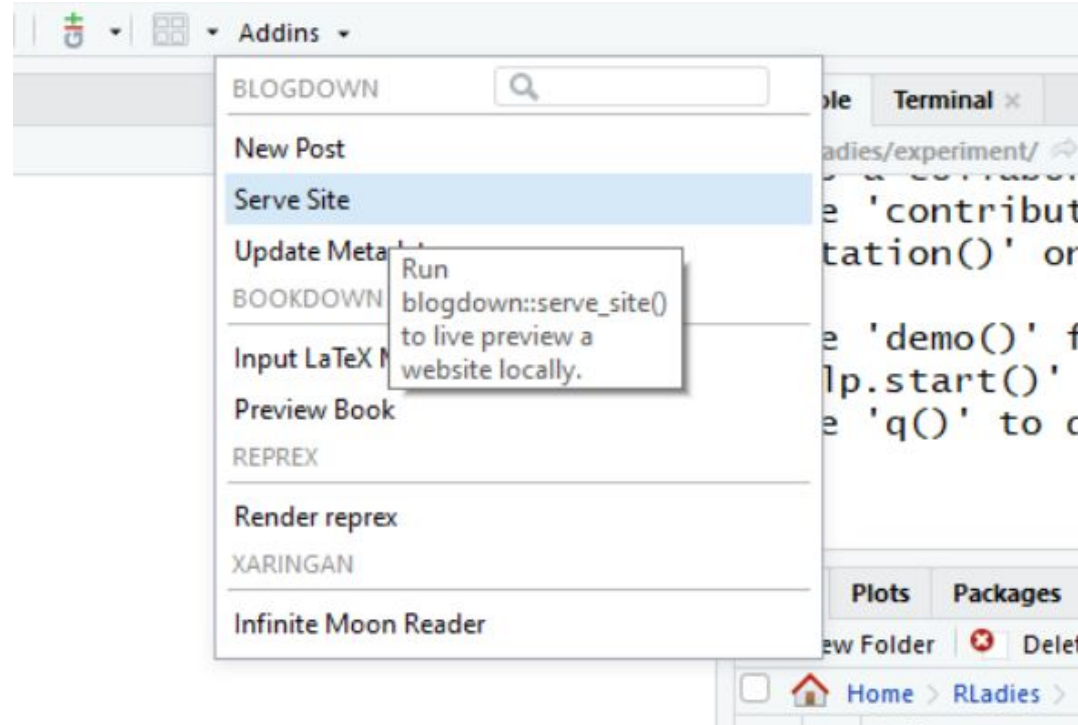


Serve site (Addins)

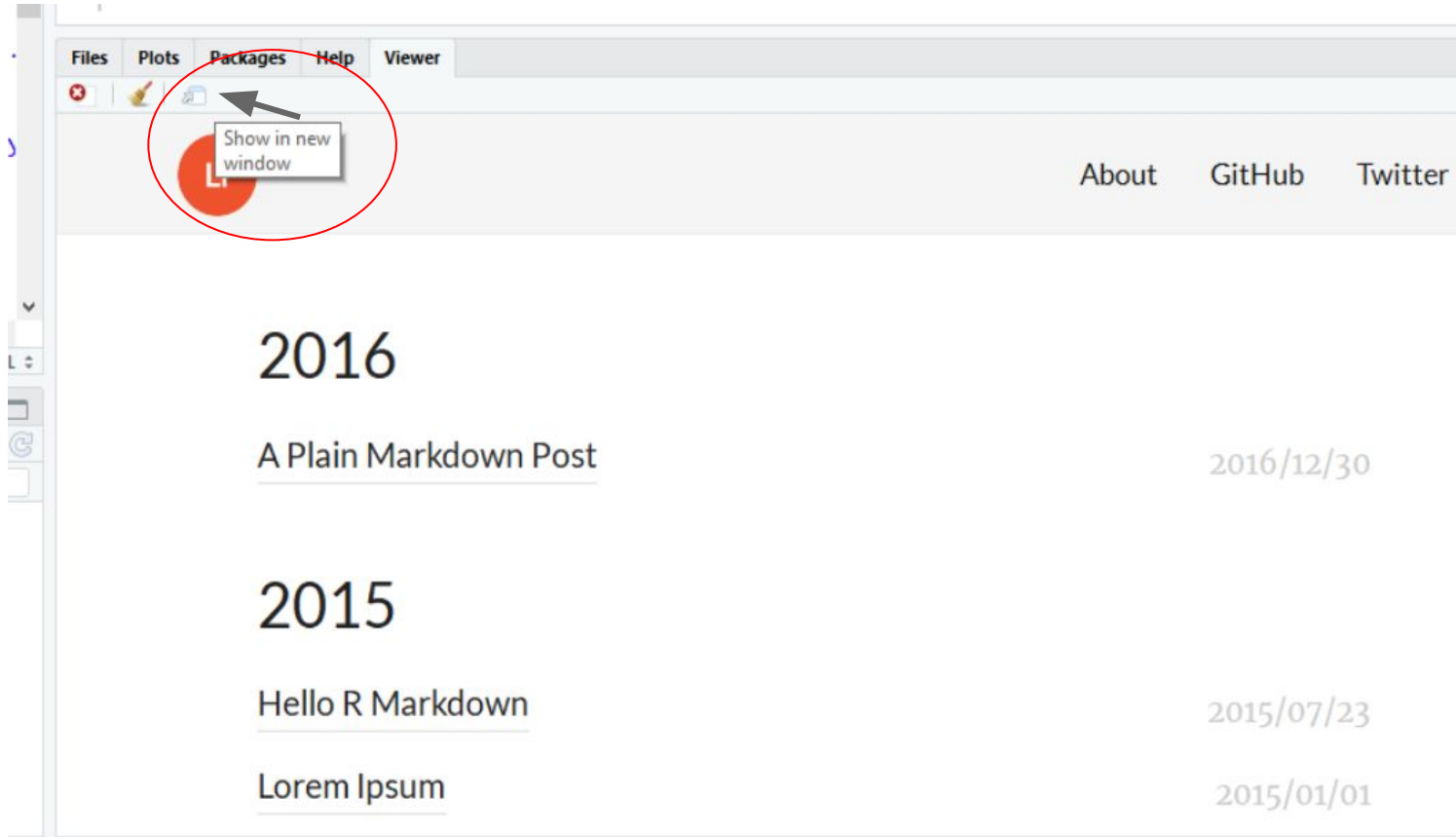
Choose the serve site option from the drop-down menu of Addins.

Or:

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

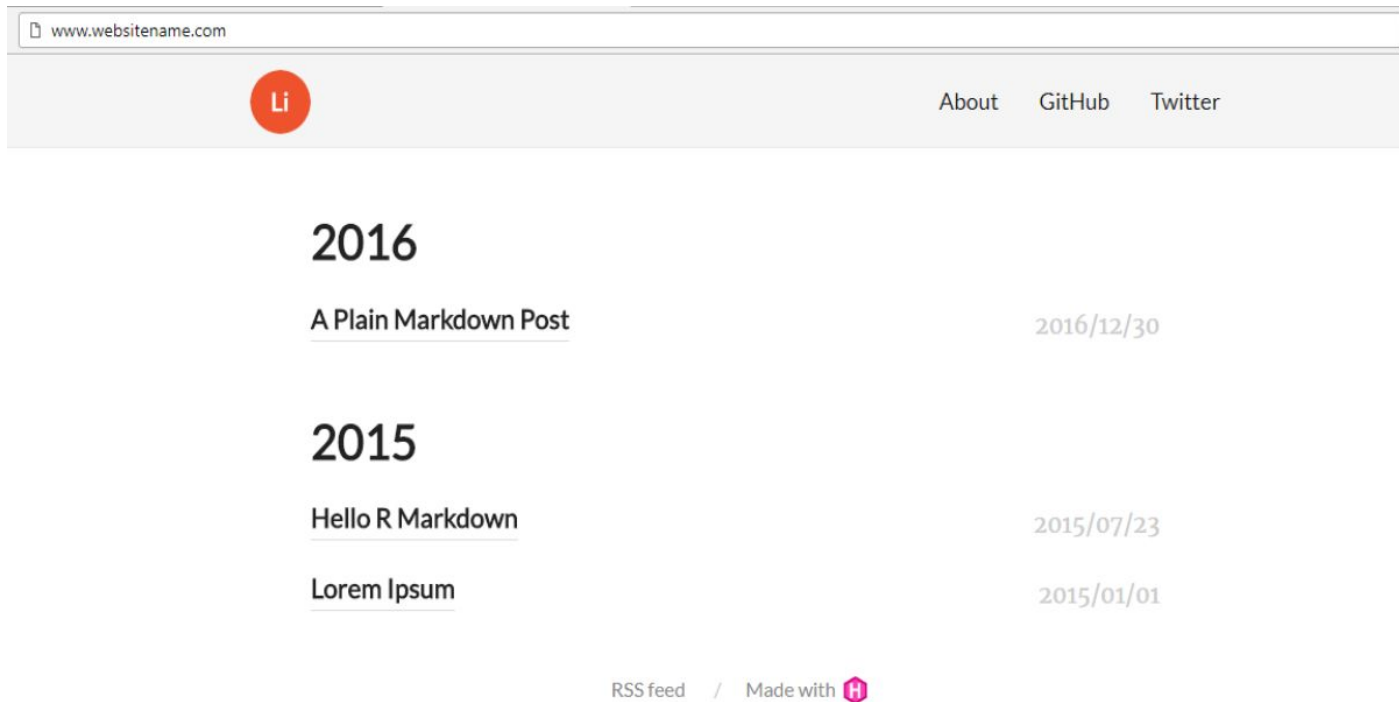


View your site (RStudio Viewer)





View your site (browser)



Download files for today's meetup



[https://github.com/rladies/meetup-presentations_tbilisi
/tree/master/Website](https://github.com/rladies/meetup-presentations_tbilisi/tree/master/Website)

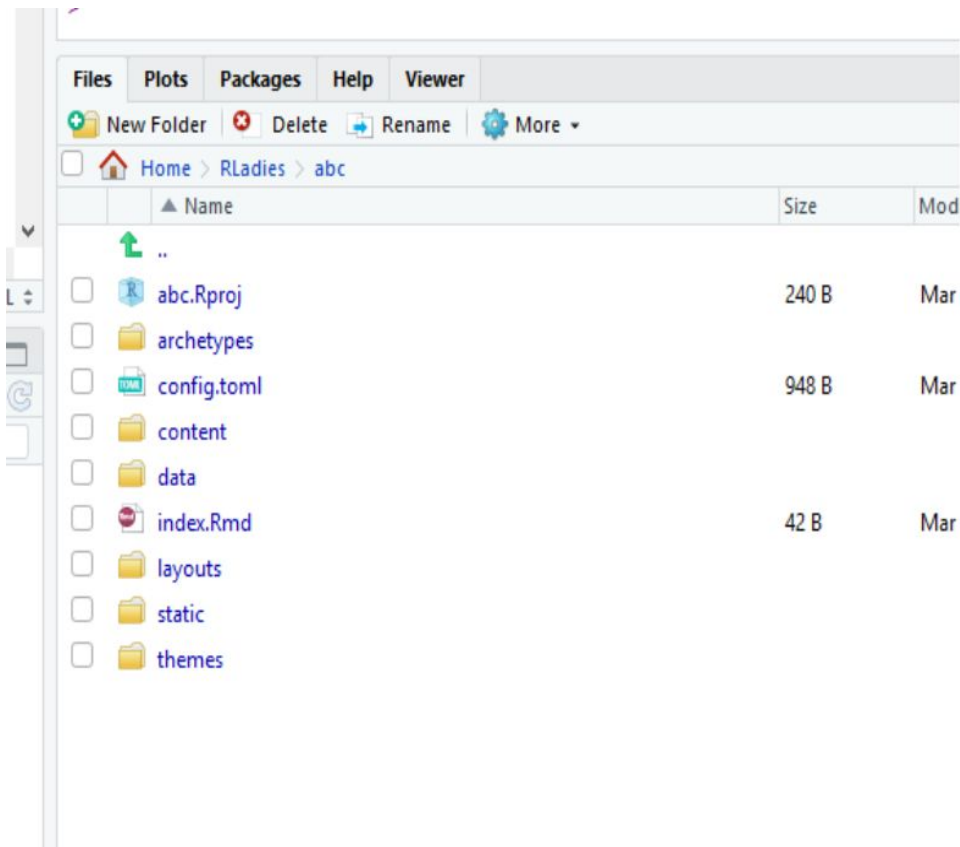


Let's create a website for R-Ladies Tbilisi


1. File → New Project → New Directory → Website using blogdown
2. Name and choose the directory (folder), eg “rldiestbilisi”
3. Keep the default Lithium theme
4. Create website

Relying very much on Alison Hills' [code](#) and [video](#)

Take a look into the folders



Open the config.toml file from RStudio project files (1/2)

1. Baseurl = “/ ” → will have the link of your website
2. Title = “Give your website a title here”
3. Theme → keep the name of the theme you chose to use (the structure of the config.toml would be different for another theme)
4. enableEmoji = true (you will see emojis  on your site!)
5. Under [params.logo] can specify the file of the logo you want to use for your site. This logo file has to be located in the static/ images folder.

config.toml file (2/2)



```
[[menu.main]]
```

```
name = "ჩვენს შესახებ"
```

```
pre = ""
```

```
url = "/about/"
```

```
weight = 1
```

name of the specific page
(visible in the header)

“pre” specifies the symbol/ icon
associated with page (e.g Twitter
; from <https://fontawesome.com/>)






“url” specifies the folder
where content is taken from

“Weight” indicates the order of appearance
of page (1st, 3rd) in the header

Getting font awesome



- Create a folder called “css” in your project’s “static” folder
- Copy-paste the file “font-awesome.css” from the shared materials to your project’s “static/css” folder
- Create a folder in your “layouts” folder called “partials”
- Move “head_custom.html” and “nav.html” files to “partials” folder
- Under [params] in config.toml file, add a line:
`customCSS = ["css/font-awesome.min.css"]`
- Fill the “pre” - field with respective icon code
“fa-music”  “fa-envelope-o”  “fa-user” 

Adding a logo



1. Copy jpg/png file “rlogo” from the shared materials to your project’s “static/images” folder
2. Under `[params.logo]` in config.toml file change the “url”:
`url = "rlogo.jpg" / url = "rlogo.png"`

Adding site content (1)



1. Change the “About” page content
 - a. Go to the “/content” folder in your project
 - b. Open “about.md” file in R
 - c. Edit the file by adding the following info:

```
# About this site
```

```
This site was created based on Alison P. Hill's
```

```
[**materials**]
```

```
(https://github.com/apreshill/blogdown-workshop)
```

Adding site content (2)



1. Copy file “2018-03-05-r-markdown.rmd” from the shared materials to your project’s “content/post” folder
2. Delete the default r-markdown post file
3. (We keep the other default posts in here for now)

Adding site content (3)



1. Add one `[[menu.main]]` to get posts back
2. Copy the “index.html” file from the shared materials to your project’s “layout” folder
3. Let’s have a look at what is in there - open the index file through RStudio
 - a. It specifies a new variable called “chapter” (not part of Hugo, created by a user there)

Last touches



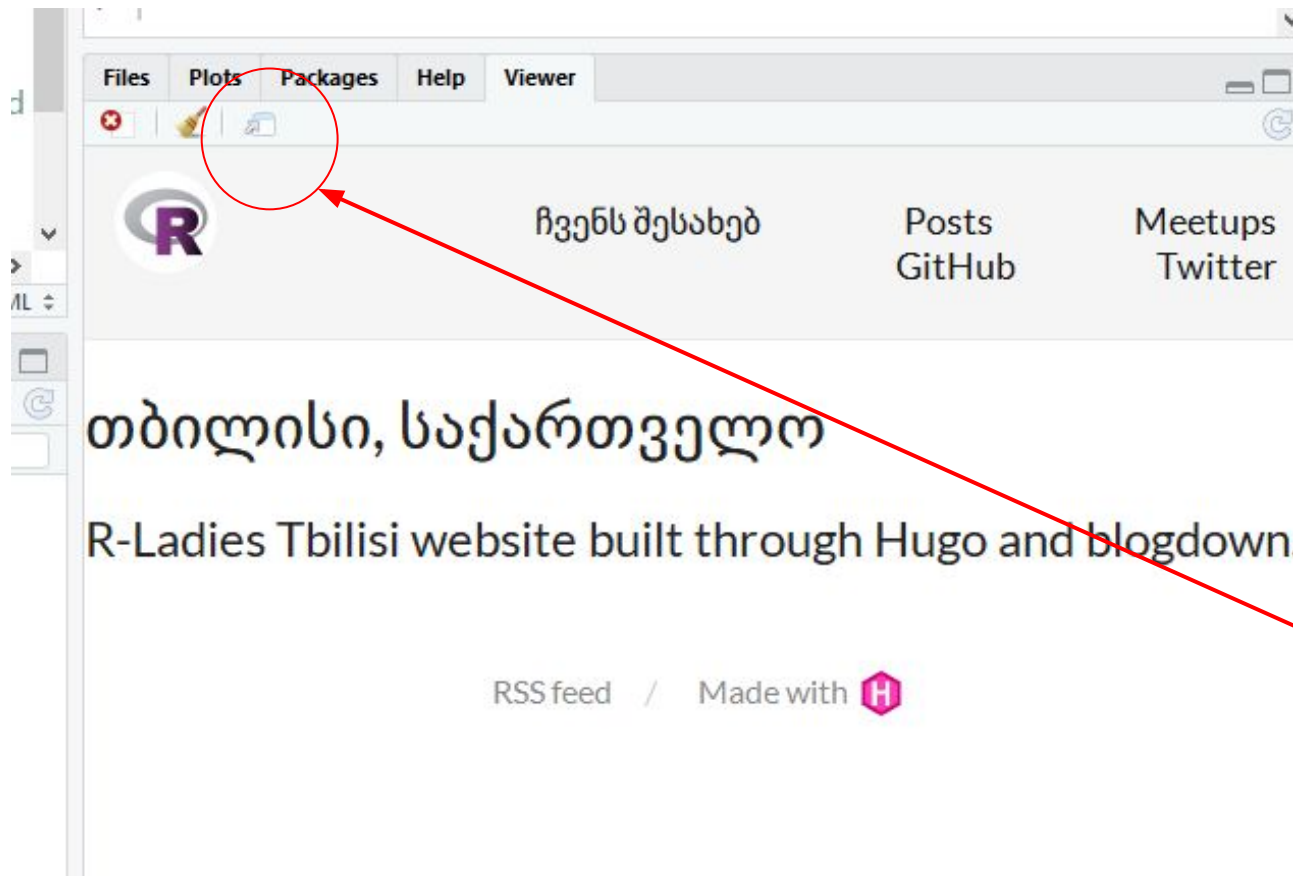
- Change the “About” to “ჩვენს შესახებ” in `[[menu.main]]`
 - If it won't work just by typing so, type it elsewhere (Notepad) and then copy paste to config.toml file
- Under `[params]` in config.toml file add:

```
chapter = "თბილისი, საქართველო"
```



This is possible thanks to creating a “chapter” variable in the index file (located in the layout folder)

Show in new window (RStudio Viewer)



View
bigger!

End-result in a browser



ჩვენს შესახებ



Posts



Meetups



GitHub



Twitter

თბილისი, საქართველო

R-Ladies Tbilisi website built through Hugo and blogdown.

RSS feed



Made with



Publishing online

1. Connect your RStudio to Github first (1 x per computer).

You can read how to do that [here](#).

2. Commit and push to Git
3. Deploy through Netlify or Github Pages (must have an account first).

Read about deployment [here](#).

Continuous deployment (Netlify)



Link your site to a Git repository

From zero to hero, three easy steps to get continuous deployment for your site.

1. Connect to Git provider

2. Select repository

3. Configure your build

Continuous Deployment: GitHub

Choose the repository you want to link to your site on Netlify. When you push to Git, we run your build tool of choice on our servers and deploy the result.

 rladiestbilisi

 Search repo

 rladiestbilisi/blogdown-workshop



Specify Hugo variable



1. Find out the latest Hugo version

```
blogdown::hugo_version()
```

2. Enter hugo build variable (in Netlify)

The screenshot shows the Netlify dashboard interface. On the left is a sidebar menu with options: General, Build & deploy (highlighted), Continuous Deployment, Single page apps, Post processing, Deploy notifications, Domain management, Functions, Forms, Identity, and Access control. The main content area is divided into two sections. The top section, 'Deploy settings', contains a table with the following data: Repository (https://github.com/rladiestbilisi/experiment), Build command (hugo), Publish directory (public), Production branch (master), Branch deploys (Deploy only the production branch and its deploy previews), and Public deploy logs (Logs are public). Below this table is a link 'Learn more about common configuration directives in the docs' and an 'Edit settings' button. The bottom section, 'Build environment variables', has a table with one row: Key (hugo) and Value (0.19). Below this table is a 'New variable' button.

Deploy settings	
Repository:	https://github.com/rladiestbilisi/experiment
Build command:	hugo
Publish directory:	public
Production branch:	master
Branch deploys:	Deploy only the production branch and its deploy previews
Public deploy logs:	Logs are public

[Learn more about common configuration directives in the docs](#)

Edit settings

Build environment variables	
Key	Value
hugo	0.19

New variable



Deploy website

More sources

[blogdown: Creating Websites with R Markdown - book](#) by
Yihui Xie, Amber Thomas, Alison Presmanes Hill
Alison Hill [blogdown workshop](#) and [video](#)
Emily Zabor [tutorial “Creating websites in R”](#)
Mara Averick [“Keeping up with blogdown”](#)
Claudia Vitolo [“Make your own blog”](#)
Marie Dussault [“Building your blog using blogdown”](#)
Netlify and Hugo



Thank you!