

# TUTORIAL

How to customize the 'Exploits\_predictions' GitPage

Emmerson Wilson

20/02/25

## CONTENT

This presentation will walk through using and customizing the “exploits\_predictions” GitPage

- Using Github
- Parts of a Quarto Webpage
- The website itself!
  - Home tab
  - Slider tab
  - Map tab
- Getting it online

# GITHUB BASICS

Somewhere to store, share, collaborate on code/software

# REPOSITORIES

This is where you store all your code/data/versions etc

All of the information for this website is at my repository: [https://github.com/emmwilson/exploits\\_predictions](https://github.com/emmwilson/exploits_predictions)

The screenshot shows the GitHub interface for the repository 'exploits\_predictions' by user 'emmwilson'. The repository is public and has 1 branch, 0 tags, 0 stars, and 0 forks. The main branch is 'main'. The repository description is 'predictions for salmon resources throughout Exploits watershed'. The repository contains several files and folders, including '\_extensions/r-wasm/live', '\_site', 'data', 'docs', '.DS\_Store', and '.Rhistory'. The repository was last committed 3 days ago by 'emmwilson'.

emmwilson / exploits\_predictions

Search: Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

exploits\_predictions Public

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file Code

emmwilson add comments to code 28462b0 · 3 days ago 24 Commits

_extensions/r-wasm/live	_files and .files	3 days ago
_site	_files and .files	3 days ago
data	got png to load!	3 days ago
docs	add comments to code	3 days ago
.DS_Store	got png to load!	3 days ago
.Rhistory	add comments to code	3 days ago

About

predictions for salmon resources throughout Exploits watershed

Activity 0 stars 1 watching 0 forks

Releases

No releases published

[Create a new release](#)

## **YOUR OWN VERSION**

I don't want you changing this version of the website

So let's make you your own!

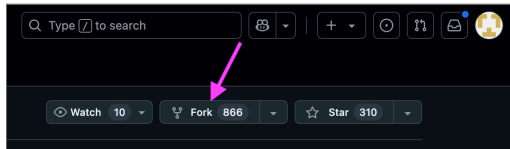
To do this we:

1. fork the repository to your Git account
2. clone the forked repository to your computer



## FORK

In the top right corner of  
[emmwilson/exploits\\_predictions](#)  
click fork




Then give your repository its own name and description

### Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

*Required fields are marked with an asterisk (\*).*

**Owner \***  
 emmwilson

**Repository name \***  
quarto-web

✔️ **quarto-web is available.**

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

**Description** (optional)

Quarto website

☒ **Copy the** `main` **branch only**  
Contribute back to quarto-dev/quarto-web by adding your own branch. [Learn more.](#)

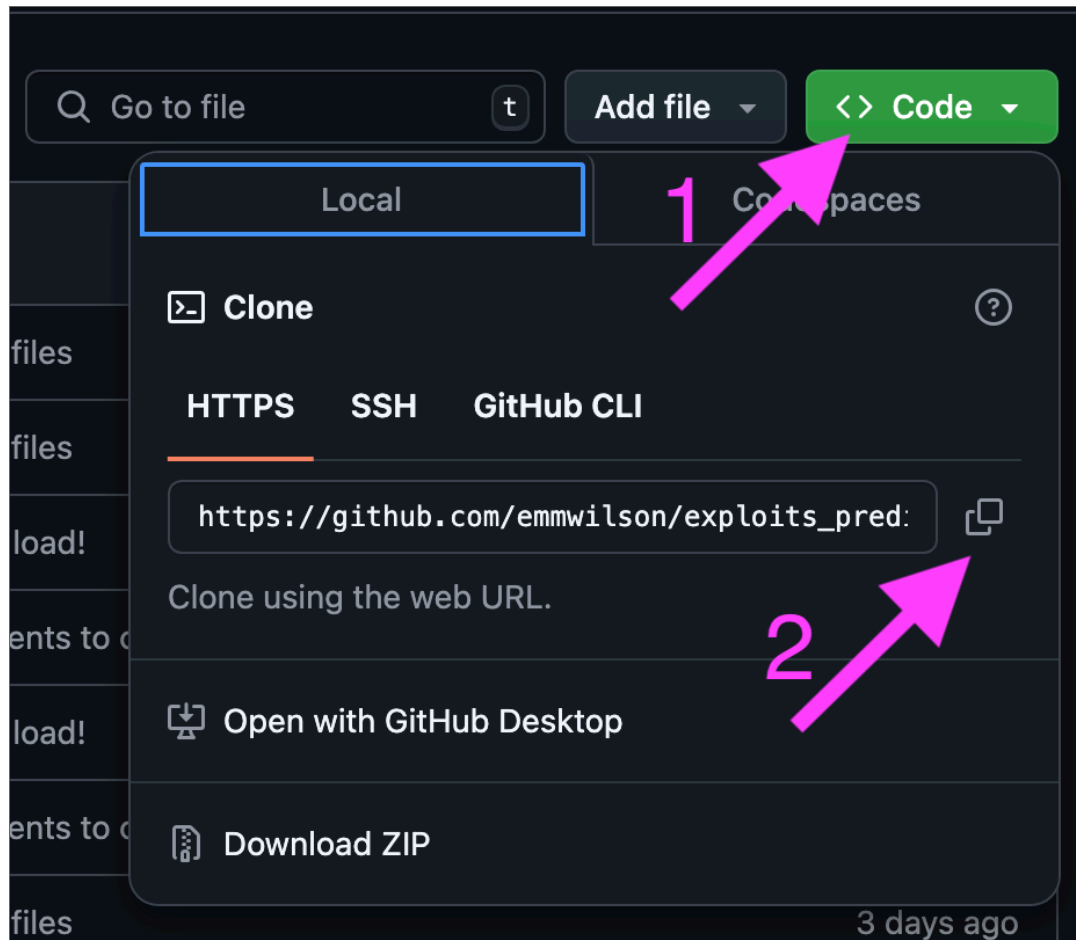
ⓘ You are creating a fork in your personal account.

Create fork

## CLONE

Now to get the repository onto your computer to work on  
We will do this using RStudio

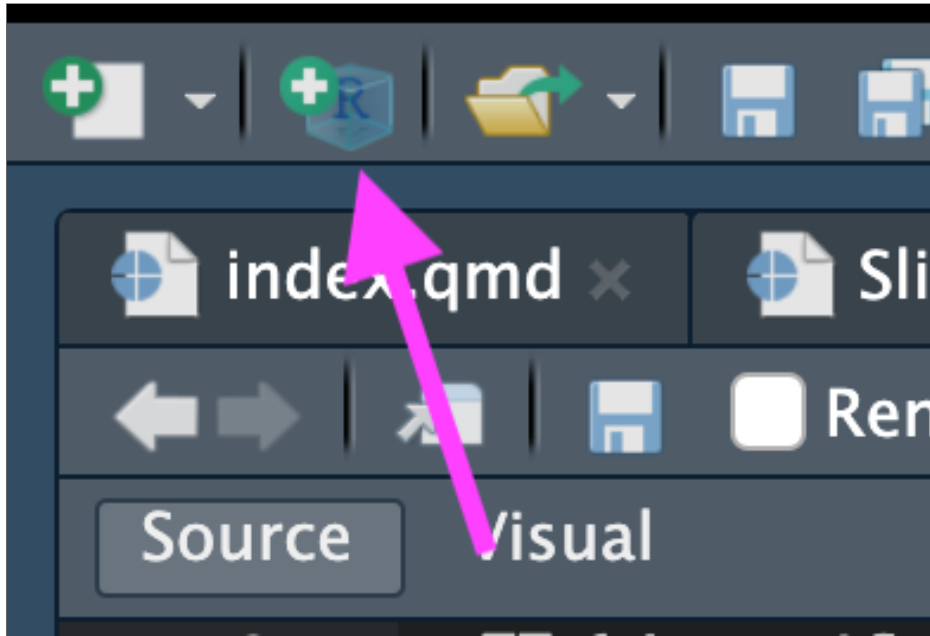
1. In GitHub (in your own version of the repository) copy the url



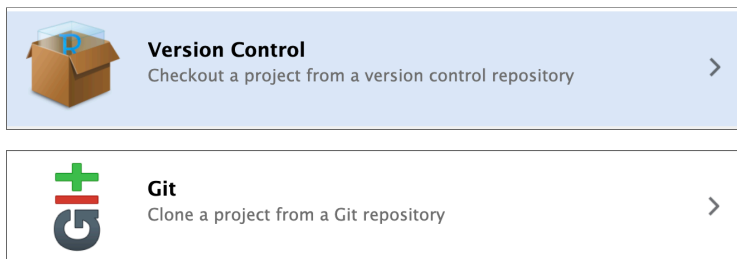
## CLONE

### 2. Create a new project

i.



### ii. Select Version control, using GIT





## CLONE


2. Create a new project

iii. Paste in the URL

iv. Give the repository a folder name on your computer, and choose where to store it

New Project Wizard

[Back](#) **Clone Git Repository**



Repository URL:

Project directory name:

Create project as subdirectory of:  
 [Browse...](#)

☐ Open in new session

[Create Project](#) [Cancel](#)

## **GIT IN RSTUDIO**

Now as you work on the project you can keep track of your changes using GitHub

This is done by:

1. Commit: creates a snapshot of the repository with the specified changes
2. Push: send the changes to the repository online



## RESOURCES

Skipped over installing git:

<https://github.com/git-guides/install-git>

Other helpful places to look for instructions

<https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>

<https://happygitwithr.com/rstudio-git-github.html>

# PARTS OF A QUARTO WEBPAGE

## **.QMD FILES**

Each of these files specifies a different page on the website.

For example:

.

## **\_QUARTO.YML**

Can specify:

- How the page gets rendered
- Which .qmd files get included as pages
- Universal themes

## **OTHERS**

There are also other files that are important to building a quarto webpage:

- styles.css
- \_extensions folder
- \_site folder
- etc...

But we don't need to alter any of them so we will skip over

## THE WEBSITE ITSELF

[https://emmwilson.github.io/exploits\\_predictions/](https://emmwilson.github.io/exploits_predictions/)



These next few slides go over which parts of the different qmd files you will need to customize to get this website to show your results

- index.qmd
- Sliders.qmd
- Exploits\_map.qmd

## INDEX

Simply type in the information on each line.

- about... :Describe the results that will show on each tab
- Affiliated manuscript: if applicable fill this out, if not you can delete that line
- Authors, Affiliations, Funding: add applicable information

# Salmon resource predictions

PUBLISHED  
14/02/25

Use the top navigation bar to explore the following outputs:

Slider predictions: about...

Map predictions: about...

---

Affiliated manuscript:

Authors:

Affiliations:

Funding:

App developement: Emmerson Wilson

## SLIDERS

Creates a page where you can select predictor variable values and see predicted levels of response variables in a bar graph

There are several parts you will need to customize:

- png paths
  - specify paths(lines: 38 - 41)
  - create table with paths (lines: 81 - 84)
- set up sliders (lines: 46 - 59)
- create table with predictions (lines: 67 - 78)

## PNG PATHS

For the response variables

1. save pngs to the “data” folder
2. add a line for your specific image  
(or change out for one you aren’t using; lines: 38 - 41)

```
258 imageURL_invert = FileAttachment("/data/invert 2.png").url()
259 imageURL_peri = FileAttachment("/data/periphyton.png").url()
260 imageURL_invert_div = FileAttachment("/data/invert_div.png").
261 imageURL_par_len = FileAttachment("/data/par_len.png").url()
```

## PNG PATHS

For the response variables

3. create table with paths (lines 81 - 84)

```
276 // create a table that stores the image paths
277 images = aq.table({
278   'response_type': ["Invertebrate biomass ()", "Invertebrate diversity", "Par length (cm)", "Periph
279   'image_path': [imageUrl_invert, imageUrl_invert_div, imageUrl_par_len, imageUrl_peri] //order the
280 })
```

## SET UP SLIDERS

For the predictor variables

Add in name of variable, range, starting value, step, and label (lines: 46 - 59)

```
300 viewof wetted = Inputs.range(  
301   [0, 10],  
302   {value: 5, step: 0.1, label: "Wetted width (units):"}  
303 );  
304 // example: substrate size  
305 viewof substrate = Inputs.range(  
306   [0, 10],  
307   {value: 5, step: 0.1, label: "Substrate size (units):"}  
308 );  
309 // example: percent lakes  
310 viewof percent = Inputs.range(  
311   [0, 20],  
312   {value: 10, step: 0.1, label: "Percent lake (units):"}  
313 );
```

## CREATE TABLE WITH PREDICTIONS

Uses the inputs from sliders to calculate the predicted response variables (lines: 67 - 78):

1. Need to make sure predictor variables match sliders

```
331 function myFunction(wetted, substrate, percent) {  
332   return aq.table({  
333     'wetted': [wetted, wetted, wetted, wetted],  
334     'substrate': [substrate, substrate, substrate, substrate],  
335     'percent': [percent, percent, percent, percent],  
336     'response_type': ["Invertebrate biomass (units)", "Invertebrate diversity", "Par length (cm)", "P  
337     'response_predict': [0.83-0.095*wetted+0.03*substrate, 0.44+0.026*wetted+0.32*percent, 0.7+0.0016  
338   })  
339 }  
340  
341 predictions_ps = myFunction(wetted, substrate, percent)
```



## CREATE TABLE WITH PREDICTIONS

Uses the inputs from sliders to calculate the predicted response variables:

2. Give response variables names (same as those for png files)

```
359 function myFunction(wetted, substrate, percent) {  
360   return aq.table({  
361     'wetted': [wetted, wetted, wetted, wetted],  
362     'substrate': [substrate, substrate, substrate, substrate],  
363     'percent': [percent, percent, percent, percent],  
364     'response_type': ["Invertebrate biomass (units)", "Invertebrate diversity", "Par length (cm)", "P  
365     'response_predict': [0.83-0.095*wetted+0.03*substrate, 0.44+0.026*wetted+0.32*percent, 0.7+0.0016  
366   })  
367 }  
368  
369 predictions_ps = myFunction(wetted, substrate, percent)
```

## CREATE TABLE WITH PREDICTIONS

Uses the inputs from sliders to calculate the predicted response variables:

### 3. Specify calculations of response variables from predictor variables

```
389 function myFunction(wetted, substrate, percent) {  
390   return aq.table({  
391     'wetted': [wetted, wetted, wetted, wetted],  
392     'substrate': [substrate, substrate, substrate, substrate],  
393     'percent': [percent, percent, percent, percent],  
394     'response_type': ["Invertebrate biomass (units)", "Invertebrate diversity", "Par length (cm)", "P  
395     'response_predict': [0.83-0.095*wetted+0.03*substrate, 0.44+0.026*wetted+0.32*percent, 0.7+0.0016  
396   })  
397 }  
398  
399 predictions_ps = myFunction(wetted, substrate, percent)
```

## Sliders

Set the below variable values to see predictions of several response variables shown in the bar graph

## MAP

Creates a map of the Exploits Watershed with predicted levels of response variables for each reach that has been sampled.

There are several parts you will need to customize:

- importing layers (lines: 38 - 50)
- create predictions (lines: 60-64)
- colour palettes (lines: 70-72)
- popups (lines: 80-86)
- map (lines: 92 - 138 \*\*but only lines that need changing are 117-130)

## IMPORT SPATIAL DATA

(lines: 38 - 50)

```
1 #import layers
2 ## river extent
3 exploits_drainage <- vect("data/spatial data/exploits_watershed.shp") |>
4   project("EPSG:4326 - WGS 84")
5
6 ## all rivers
7 exploits_river <- vect("data/spatial data/OrderSlopeSegJan22.shp") |>
8   project("EPSG:4326 - WGS 84")
9
10 ## lakes
11 lakes <- vect("data/spatial data/Exploits_lakes.shp") |>
12   project("EPSG:4326 - WGS 84")
13
14 ## data from river reaches sampled
15 field_reaches <- vect("data/spatial data/exploits_reaches_with_sites.shp") |>
16   project("EPSG:4326 - WGS 84")
```

## CREATE PREDICTIONS

based on data from sampled reaches (lines: 60-64)

```
1 # use river reach data to predict outcomes
2 predictions_reaches <- field_reaches
3   ## predict responses based on data from field at each reach
4 predictions_reaches$invert <- 0.43-0.015*predictions_reaches$avg_wet_wi+0.03*predictions_reaches$D
5 predictions_reaches$embed <- 0.43-0.015*predictions_reaches$avg_wet_wi+0.03*predictions_reaches$D
```

## COLOUR PALETTES

Add palettes as you add response variables (lines: 70-72)

```
1 #colour palettes for predictions
2 invert.pal <- colorNumeric(c("#f47c3c", "#d9534f"), domain = predictions_reaches$invert, na.color = "white")
3 embed.pal <- colorNumeric(c("#e9c602", "#93c54b"), domain = predictions_reaches$embed, na.color = "white")
```

## POP UPS

Lists out the data that will show up when you click on a reach (lines: 80-86)

```
1   label_text <- glue("<b><u>{predictions_reaches$stream_nam}</u></b>")
2   "<b>Embeddedness: </b> {predictions_reaches$embed}<br/>", # predict
3   "<b>Invertebrate biomass: </b> {predictions_reaches$invert}<br/>",
4   "<b>Wetted width: </b> {predictions_reaches$avg_wet_wi}<br/>", # p
5   "<b>Catchment: </b> {predictions_reaches$catchment_}<br/>", # pred
6   "<b>Substrate size: </b> {predictions_reaches$D50_pred}<br/>") |>
```

Looks like

.



## CODE FOR THE MAP

(lines: 92 - 138 \*\*but only lines that need changing are 117-130)

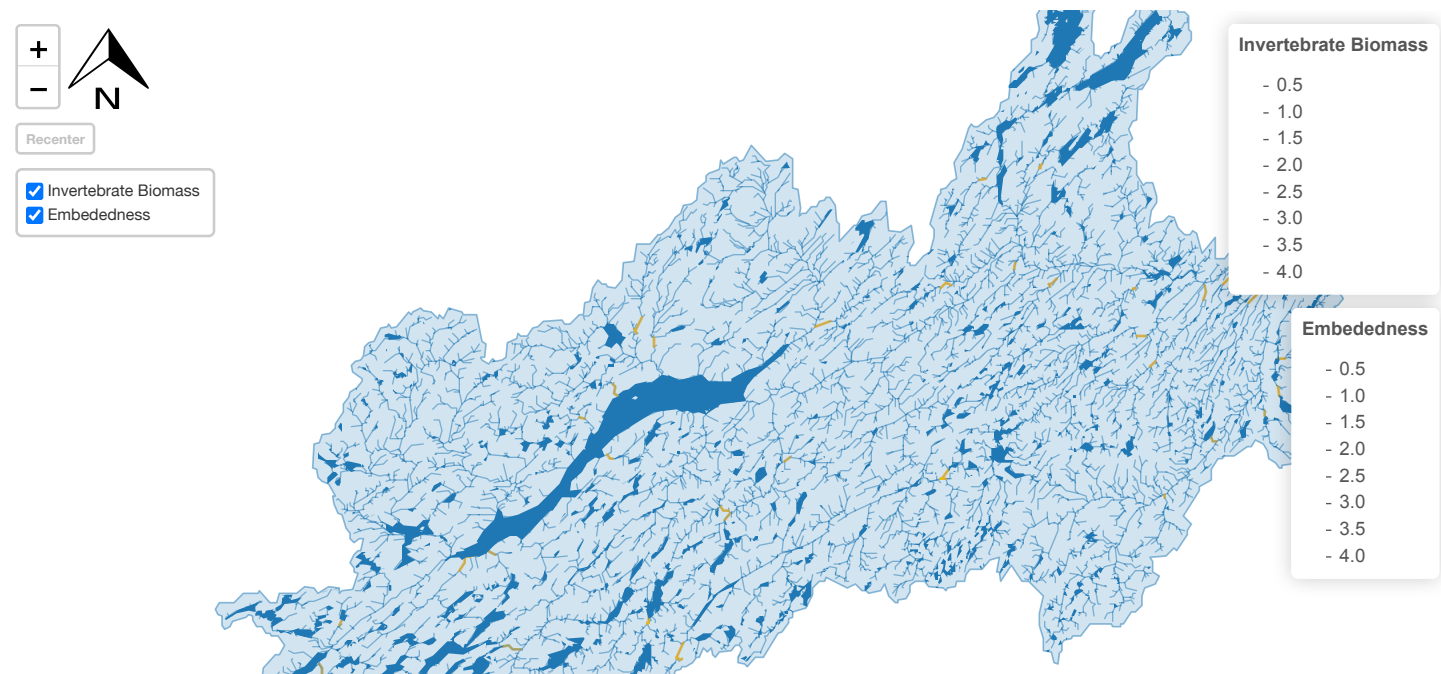
```
1 exploits_map <- leaflet(width = 1900, height = 1600) |>
2   addProviderTiles('Esri.WorldImagery') |> # adds base map
3
4   # open map to this screen
5   setView(-56.65499, 48.70449, zoom = 9) |>
6
7   #can recenter to that location by clicking button
8   addHomeButton(ext = c(-58.9311,47.9605, -53.830,49.3532), position = "topleft", group = "Rece
9
10  # create layers to add vectors to
11  addMapPane("drainage", zIndex = 410) |>
12  addMapPane("river", zIndex = 415) |>
13  addMapPane("reaches", zIndex = 420) |>
14  addMapPane("lakes", zIndex = 425) |>
15
16  # add polygons and lines
17  ## exploits drainage area
18  addPolygons(data = exploits_drainage, color = "#1f78b4", weight = 1.25, fillColor = "#1f78b4"
19
```



## Exploits watershed predictions

Predictions of invertebrate biomass and embededness for the sampled reaches of the Exploits watershed.

Click on a reach that has been sampled (highlighted in orange-red or yellow-green) to see the predictions and underlying characteristics



## GETTING THE SITE ONLINE

We use GitPages to host the interactive webpage

## **A FEW THINGS YOU NEED FIRST**

In terminal run this line of code:

- quarto add r-wasm/quarto-live

(there should be a terminal tab beside the RStudio console tab)

## **SOURCE FOLDER**

GitPages needs to know where to look to render the webpage. In our case we are using the docs folder.

Go to GitHub -> Your repository -> Settings (along the top)-> Pages (along left)

Make sure your deployment is configured like so:

.

## **RENDER LOCALLY**

To get the docs folder ready we need to render everything locally

To do that click

.

along the top of RStudio for each qmd file

This should open up each page in the browser

And will create the necessary files in the docs folder

## **COMMIT AND PUSH**

Now we need to commit and push all the changes we've made

1. In RStudio: go to the Git tab (one along top or one by the environment tab)
2. Select

## COMMIT AND PUSH

### 3. Commit:

i. Select the files and folders that show up in the top left console

ii. Add a commit message and select “Commit”

**\*\* Important \*\*** There is a limit to how much you can commit and push at one time!

If you are adding any spatial data, do a separate commit and push for that file

And be conservative about how much you commit at one time

4. After it has finished with the commit click Push





## PAGES BUILD AND DEPLOYMENT

GitHub will now build and deploy your webpage!

Once it has finished you can go to:

Settings -> Pages

and you should have the option to visit the site

[https://emmwilson.github.io/exploits\\_predictions/](https://emmwilson.github.io/exploits_predictions/)



**CONGRATS! YOU'VE CUSTOMIZED THE WEBPAGE!**



## EXTRA RESOURCES

### QUARTO:

basics of quarto pages: <https://quarto.org/docs/websites/> info on using quarto and GitPages:  
<https://quarto.org/docs/publishing/github-pages.html>

### FOR SLIDERS

using observable js with quarto: <https://quarto.org/docs/dashboards/interactivity/observable.html> all the options available with Plot: <https://observablehq.com/plot/what-is-plot> going from ggplot to Plot:  
<https://observablehq.com/@observablehq/plot-from-ggplot2>

### USING LEAFLET

how to use leaflet to make maps in R: <https://rstudio.github.io/leaflet/articles/leaflet.html>