

All materials at:

[github.com/paulbradshaw/
parameterisation](https://github.com/paulbradshaw/parameterisation)



Personalisation

One dataset > 150 webpages with one click

ME! ME! ME!

Who are we?

...





BBC England Data Unit & BBC Shared Data Unit

The BBC England data unit uses data to find, tell, and illustrate news affecting England. The BBC Shared Data Unit helps

30 followers Birmingham, UK <https://uk.pinterest.com/bbcenglan...>

Overview

Repositories 296

Projects

Packages

Teams

People 11

Settings

Pinned

Customize pins

shared-data-unit Public

An explanation of the shared data unit and what they do

41 8

unsolved-crime Public

More than two thirds of thefts are 'never solved'

2 3

libraries Public

Data used in the story 'Libraries lose a quarter of staff as hundreds close'

Python 5 4

NHS-litigation Public

Forked from Alex-Homer/NHS-litigation

Cost legacy of decades-old NHS blunders begins to rise

music-festivals Public

Festivals dominated by male acts, study shows, as Glastonbury begins

Python 8 2

art-uk Public

Illuminating facts about the UK's art collection

7 2

<https://github.com/BBC-Data-Unit>

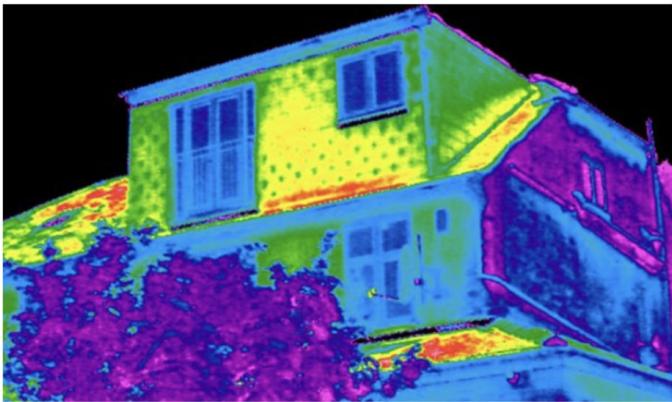
What we'll cover

Why we generated 150-page websites for partners

How to do it: parameterisation + GitHub Pages

Tips and tricks





What the pack contains:

We sourced data from a variety of sources including the UK census and open data on EPC certification, as well as looking at information from the various nation housing surveys.

We extracted a series of measures for each local authority in England, Wales and Scotland to determine:

- For each local authority area, the number of homes in each EPC band
- The proportion of homes currently falling short of the government target to make homes energy efficient (homes currently in the D to G bands)

How energy efficient are our homes?

BBC Local News partnerships

Mailbox | BBC Birmingham | B1 1AY

Energy efficiency in private rented accommodation FOR SHARING

A1 | Local authority

A	B	C	D	E	F	G
Local authority	Inspection s rating properties below C in 2018-2022	Inspections 2018-22 where property could not achieve a "potential" C rating	Proportion using portable heaters as secondary heating	% of recommendations related to improving electricity	% of recommendations related to improving heating and water	% of recommendations related to improving insulation
Isles of Scilly	90%	18%	56%	15%	34%	33%
Ryedale	85%	4%	70%	20%	27%	35%
Eden	85%	6%	66%	20%	29%	36%
Isle of Anglesey	85%	11%	55%	20%	31%	37%
Derbyshire Dales	82%	3%	62%	20%	29%	35%
Ceredigion	80%	7%	45%	19%	32%	38%
Gwynedd	79%	6%	45%	17%	31%	39%
North Norfolk	79%	6%	54%	19%	30%	36%
Craven	78%	4%	62%	20%	30%	36%
Ribble Valley	78%	5%	52%	21%	30%	36%
Cotswold	77%	7%	61%	19%	28%	36%

LOCAL NEWS PARTNERSHIPS

Police misconduct

This story is available for use by the BBC's local news partners. Please do not share outside of the network. It is under strict embargo.

What's the story?

We've looked at the effectiveness of the police complaints system for

The Independent Office for Police Complaints (IOPC) succeeded the Independent Police Complaints Commission (IPCC) in 2012 to improve police accountability and make more accountability.

The watchdog has powers to ensure police forces investigate when it has concerns about how officers have behaved.

Ultimately, police disciplinary panels have the final say on sanction.

We've analysed:

1. Three years of yearly outcome reports published by the IOPC
2. More than 800 published reports on its website

Yearly outcome reports

Over the past three years, some 418 misconduct cases were held by force panels.

The force panels subsequently found officers or staff to have committed misconduct.

In misconduct cases, panels gave 18 final written warnings, 57 written warnings and 12 cases.

Force analysis: West Midlands Police

[Back to main page](#)

This story is available for use by the BBC's local news partners. Please do not share outside of the network. It is under strict embargo.

We've analysed reports into investigations published into the conduct of officers and staff at West Midlands Police since the IOPC was formed in 2018.

In 16 of those reports, investigators found at least one officer or member of police staff had a case to answer for misconduct, and advised the force to hold a disciplinary panel.

In these cases, the IOPC identified **17 officer(s)** and **6 police staff** as having a case to answer for misconduct.

1 officer(s) and **0 staff** had retired or left the force by the time the tribunal was held.

When the case was heard by the force's tribunal, misconduct was proven against **10 officer(s)** and **6 staff**.

The force panel cleared **7 officer(s)** and **0 staff**

The outcomes were as follows:

- **9 officers** and **0 staff** faced no further action.
- **7 officers** and **6 staff** were given management action.
- **0 officers** and **0 staff** were given a written warning.
- **1 officers** and **0 staff** were dismissed.

1 of the cases referred to death or serious injury having occurred shortly after contact with police

LOCAL NEWS PARTNERSHIPS

LOCAL NEWS PARTNERSHIPS



Children's language needs rise post lockdown

This story is available for use by the BBC's local news partners. Please do not share outside of the network. It is under strict embargo until November 7 at 0001.

Use the dropdown navigation menus across the top of the page to see the data for a specific local authority

What's the story?

We've looked at the numbers of children needing support for 'Speech, language and communication needs' in primary schools. In particular, children in Year 1, who were at pre-school age during the lockdowns of the 2020/21 academic year.

Speech and language is the biggest category of support, and has experienced one of the biggest increases.

The number of children needing speech and language support analysis shows.

The increase is partly down to the effect of lockdowns.

The Royal College of Speech and Language Therapists

LOCAL NEWS PARTNERSHIPS



Local authority analysis: Birmingham

Analysing the impact of the pandemic on children's speech development

Children entering school in the year after the first lockdown are showing a big increase in speech, language and communication (SLC) needs, according to analysis of DfE data. Find out how the numbers changed in Birmingham.

In **Birmingham** the number of children needing support for speech, language and communication in Year 1 rose from **765** in 2015/16 to **1114** in 2021/22.

The number of children needing support for speech, language and communication in Year 1 in 2021/22 **rose by 19.02%** compared to the previous academic year. This is higher than the national rise of **9.81%**. The authority ranks **7** in the region out of 14 authorities.

Explore your area

senspeech.github.io/website/Liverpool.html

Show 10 entries

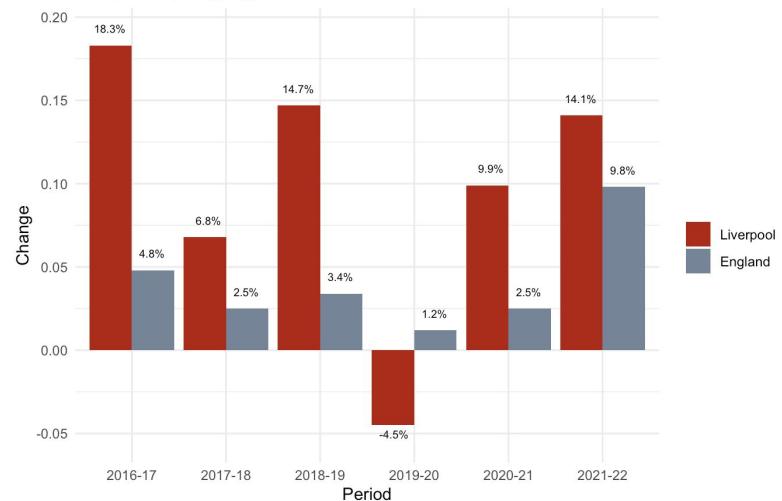
Search:

Numbers of children in year 1 needing speech, language and communication support in each authority. Analysis by the BBC Shared Data Unit

Authority	2015/16	2016/17	2017/18	2018/19	2019/20	2020/21	2021/22
All	All	All	All	All	All	All	All
10 Lancashire	446	507	521	588	628	576	622
11 Liverpool	322	381	407	467	446	490	559
12 Manchester	305	408	434	430	485		
15 Salford	240	295	298	276	276		
3 Bolton	142	178	218	214	260		
13 Oldham	201	198	178	255	215		
1 Blackburn with Darwen	181	200	199	216	224		
7 Cumbria	207	181	177	191	223		
22 Wigan	133	159	174	206	207		
23 Wirral	164	162	226	219	244		

Chart: year-on-year change in children needing speech and communication support

Year-on-year change in year 1 pupils needing SEN support for speech, language and communication



The steps:

- Produce a ‘template’ analysis for a given area
- Produce an index (home) page
- Render multiple .md versions of that analysis - one for each region
- Render HTML versions
- Publish as a website online (GitHub Pages)
- Debug, clean and tweak the code



Go to this URL

[github.com/paulbradshaw/
parameterisation](https://github.com/paulbradshaw/parameterisation)

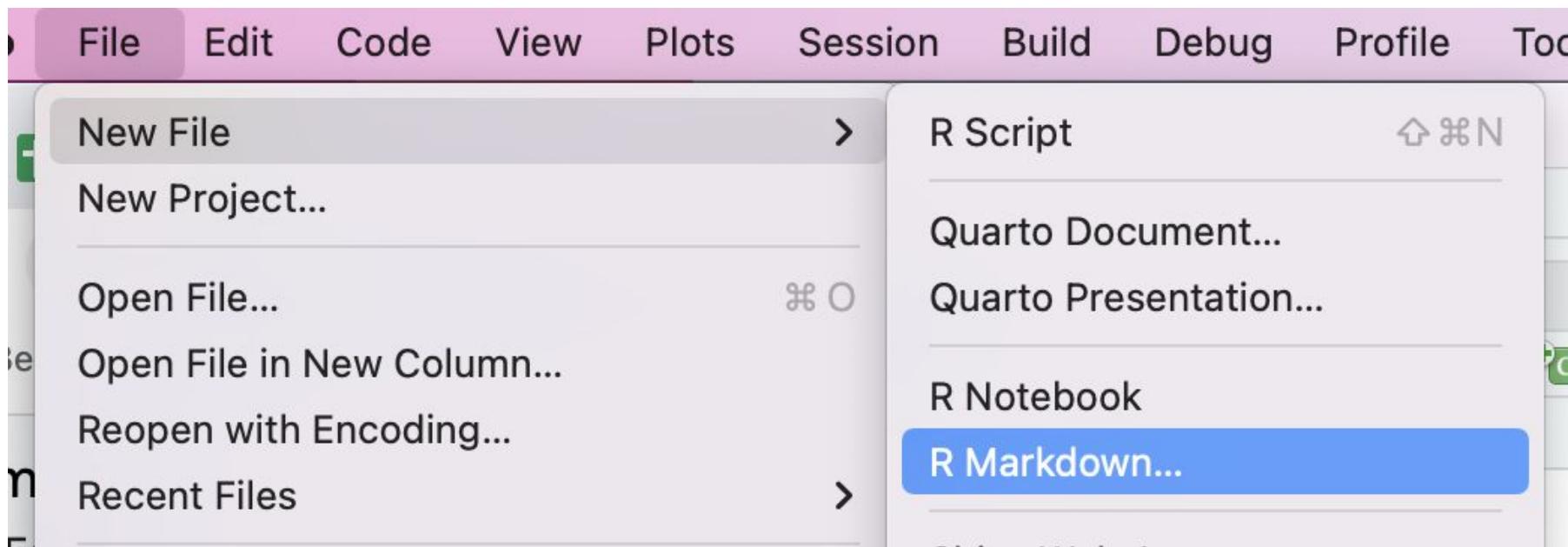
and: <https://drive.google.com/drive/folders/17WtozAfB1frGZmLOJbSbJooP1WRDIIfY1?usp=sharing>

The principles:

- Create a parameter — this should be what you want to generate a different page for (e.g. region, police force, category, politician)
- Limit code output to what should be in the HTML file (visible elements)
- Loop through list and call a ‘knitting’ function to create a file for each

R Markdown files: a quick introduction

Create an R Markdown file



Give a title - and HTML

New R Markdown

-  Document
-  Presentation
-  Shiny
-  From Template

Title: 01template

Author: BIGNEWS

Date: 2023-05-25

Use current date when rendering document

Default Output Format:

HTML
Recommended format for authoring (you can switch to PDF or Word output anytime).

PDF
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

Knit - and title it again

The screenshot shows the RStudio interface with an R Markdown file named "01template.Rmd" open. The code editor displays the following R Markdown code:

```
1 ---  
2 title: "01template"  
3 author: "BIGNEWS"  
4 date: "2023-05-25"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple te
```

A red box highlights the "Knit" button in the toolbar, which has a blue knitting icon. A red arrow points from this button to a tooltip message: "Knit the file – you'll be asked to name it again". A modal dialog box is displayed, prompting the user to save the file. The "Save As:" field contains "01template.Rmd", the "Where:" field is set to "dataharvestparam", and there are "Cancel" and "Save" buttons at the bottom.

Markdown will generate HTML:

- Headings: # or ## or ###
- Italic, bold: * or ** or *** before and after
- Links: [link text] (<http://url>)
- Indented text: >
- Bullet list items: *
- Code blocks: ```\${r} above and ```` below
- Inline code: ` before and after

Notebook 1: the template

The YAML header

```
title: "01template"  
author: "BIGNEWS"  
date: "2023-05-25"  
output: html_document
```



Add a parameter

```
---
```

```
title: "01template"
author: "BIGNEWS"
date: "2023-05-25"
output: html_document
params:
  reg: "Anyville"
---
```



Change the title

```
title: |  
  Regional analysis: `r params$reg`  
author: "BIGNEWS"  
date: "2023-05-25"  
output: html_document  
params:  
  reg: "Anyville"
```

What have we added?

```
---
```

title: |
 Regional analysis: `r params\$reg`

author: "BIGNEWS"

date: "2023-05-25"

output: html_document

params:
 reg: "Anyville"

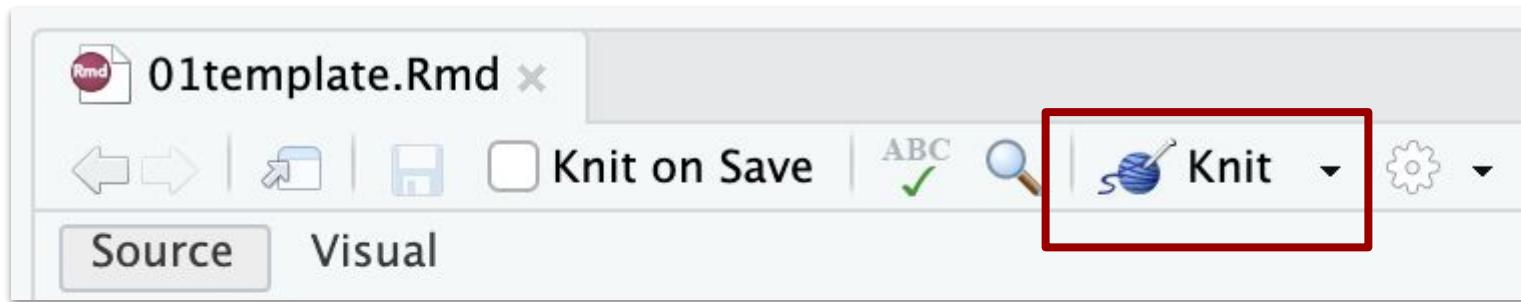
```
---
```

We define a list variable called 'params'

Within that list an element called 'reg' with a value of 'Anyville' is created. We set it to that value for the purpose of testing.

The title fetches the 'params' list and then drills down into the 'reg' element within that, and fetches its value

Knit to see how it's changed!



Delete unwanted info

```
title: |
  Regional analysis: `r params$reg`
output: html_document
params:
  reg: "Anyville"
---
```

Add your own Markdown

```
```{r setup, include=FALSE}  
knitr::opts_chunk$set(echo = TRUE)
```
```

Code block
with knit
settings (leave
for now)

Analysing stop and search

Police have been given new powers to stop and
search potential criminals - but are the
existing powers effective - and used fairly?
Find out how the numbers changed in

Level 1
header
(1 hash, then
space)

Use the parameter

```
# Analysing stop and search
```

Police have been given new powers to stop and search potential criminals - but are the existing powers effective - and used fairly? Find out how the numbers changed in `r params\$reg`.

Inline R code:
‘calls’ the variable
params\$reg

Code blocks:

```
```{r}
```

```
Code goes here
```

```
#add comments to explain
```

```
```
```

- Code blocks and any messages/warnings will be visible in the HTML version — unless you specify otherwise in the `Knitr` settings

Code: import packages

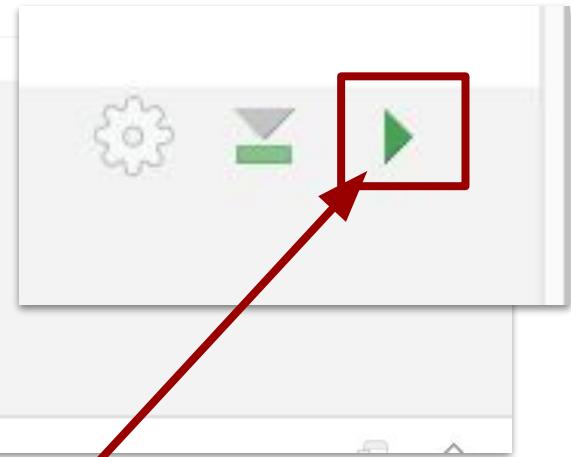
```
```{r import packages}  
#import the tidyverse
library(tidyverse)
#The datatables package
library(DT)
...``
```

Another code block -  
we've named it 'import  
packages'

Code to import  
two packages -  
with comments

# Run the code!

```
0
1 ```{r import packages}
2 #import the tidyverse
3 library(tidyverse)
4 #The datatables package
5 library(DT)
6
7 ````
```



Press play!

# Import the data

```
```{r import data}  
#store the URL  
fileloc =  
"https://docs.google.com/spreadsheets/d/e/2PACX-  
1vR2-w4JWtqaDmOBp7CxZpMaqsa-2Tm4vUVs1Of5_NVfiTIa  
vYEJWHWOv-4sGJ9VUU1x8eJPnULECeYW/pub?gid=0&singl  
e=true&output=csv"
```

Another code block

Store the URL of the CSV

```
#Tidyverse's read_csv function imports from it  
data = read_csv(fileloc)  
...
```

Read the CSV into a dataframe called 'data'

When knitted...

Analysing stop and search

Police have been given new powers to stop and search potential criminals - but are the numbers changed in Anyville.

```
#import the tidyverse which includes dplyr, that we'll use  
library(tidyverse)
```

```
## — Attaching packages —————— t  
## ✓ ggplot2 3.4.0      ✓ purrr   1.0.1  
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10  
## ✓ tidyr   1.2.1      ✓ stringr 1.5.0  
## ✓ readr   2.1.3      ✓forcats 0.5.2  
## — Conflicts —————— tidyver
```

Change Knitr settings

```
```{r setup, include=FALSE}  
knitr::opts_chunk$set(echo = FALSE,
 warning = F,
 message = F)
```

```
```
```

```
# Analysing stop and search
```

Police have been given new powers to stop and search potential criminals - but are the

Change echo= to FALSE (or F) to hide code from output

Add two parameters to suppress messages & warnings

Use parameter to filter

```
```{r filter and subset}
#filter to those for this la, and specified cols
subsetdf <- data %>%
 dplyr::filter(region == params$reg)
```

```



Parameter
must match
value in data to
act as filter

Extract figures to use

```
```{r extract figures}
#store the figures for the earliest and latest years
fig21 <- subsetdf$stops_2021
fig22 <- subsetdf$stops_2022

#calculate the change in this region
localchange <- (fig22-f1g21)/f1g21
#calculate the change across all regions
nationalchange <-
 (sum(data$stops_2022)-sum(data$stops_2021))/sum(data$stops_
2021)
...````
```

# Template text + variables

```
```{r create custom string}
#set a variable to 'rose' or 'dropped' depending on relation of
earlier figure to newer one
changetype <- ifelse(isTRUE(fig22 > fig21), "rose", "dropped")

#create a custom string with that
customstring1 <- paste0("In **",params$reg,"** the number of stops
",changetype," from **",fig21,"** in 2021 to **",fig22,"** in
2022.")
```
`r customstring1`
```

Code block: creates a text string, inserting variables for particular region

Inline code: 'calls' that string variable

# Another line...

```
```{r create another string}
nationalcomparison <- ifelse(localchange>nationalchange, "higher
than", "below the")

#create a custom string with that
customstring2 <- paste0("The number of people being stopped in 2022
**",changetype," by ",round(localchange*100,1),"%"** compared to the
previous year. This is ",nationalcomparison," the national rise of
**",round(nationalchange*100, 1),"%"** .)
```
`r customstring2`
```

Code block: creates a text string, inserting variables for particular region

Inline code: 'calls' that string variable

# That custom string broken down

---

```
paste0(
 "The number of people being stopped in 2022
 **",
 changetype,
 " by ",
 round(localchange*100,1),
 "%** compared to the previous year. This is ",
 nationalcomparison,
 " the national rise of **",
 round(nationalchange*100, 1),
 "%**.")
```

# Add an interactive table

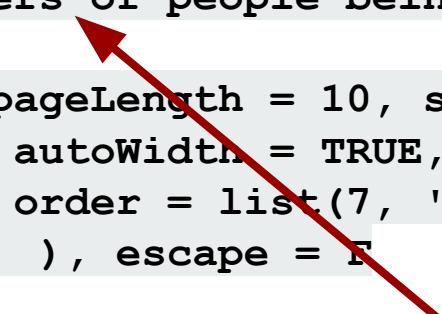
```
Explore your area
```



Add a heading before the table

```
```{r table}
#Create the datatable. Add a caption if you want
DT::datatable(data,
              style = 'bootstrap',
              caption = 'Numbers of people being stopped and searched',
              filter = 'top',
              options = list(pageLength = 10, scrollX=TRUE,
                            autoWidth = TRUE,
                            order = list(7, 'desc') #order by col 7
                            ), escape = FALSE
            )
```

```



Customise these options

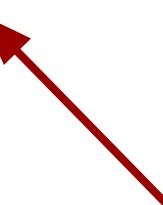
# Tip: clean headings first

```
Explore your area
```

```
```{r clean_headings}
#clean column headings
colnames(subsetdf) <-
str_replace(colnames(subsetdf), "_" , " ")
```

```

Column names often use underscores or other characters which are better replaced with spaces when used publicly. Use this code *before* the table is generated.



# Tip: reduce file size

```

```

```
title: |
 Regional analysis: `r params$reg`
output:
 html_document:
 selfContained: false
 lib_dir: site/libs
params:
 reg: "Anyville"
```

By default each HTML document contains all the JavaScript it needs - but this makes it huge. We stop it doing that.

It now needs to know where to put any files (JavaScript, CSS) that the HTML document will need, so we specify a folder called 'libs' inside a 'site' folder

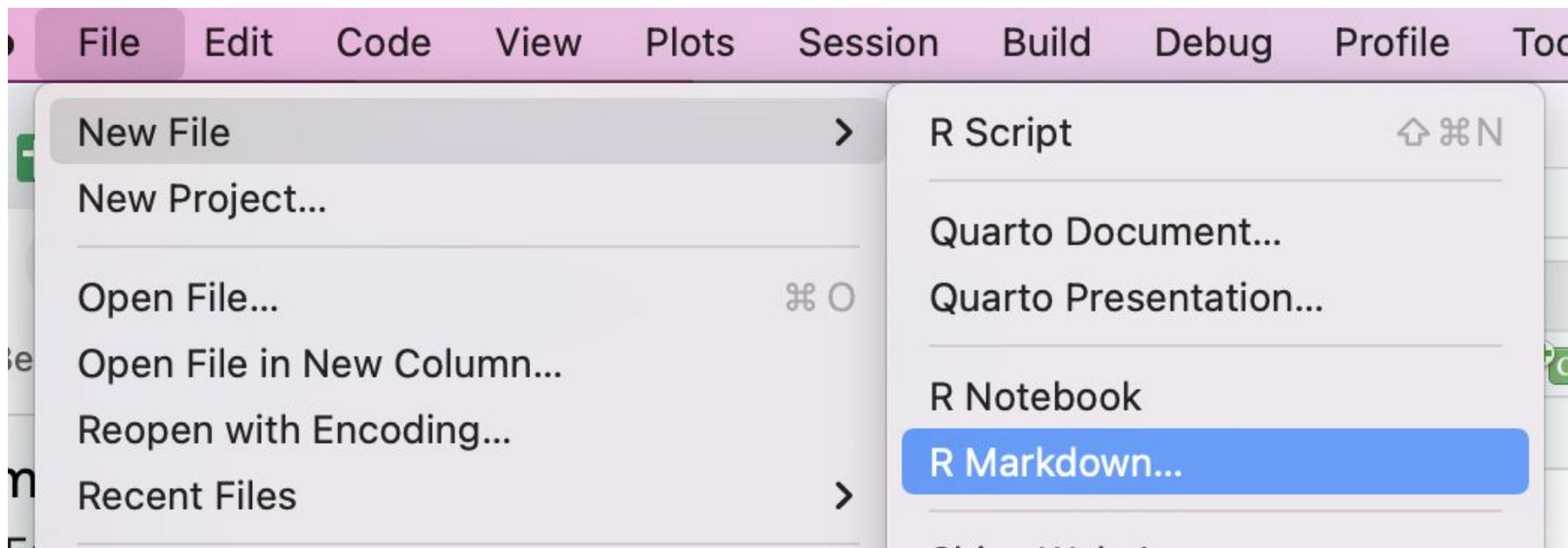
```

```

# **Notebook 2: Index.Rmd**

## **(in the /site folder)**

# Another R Markdown file

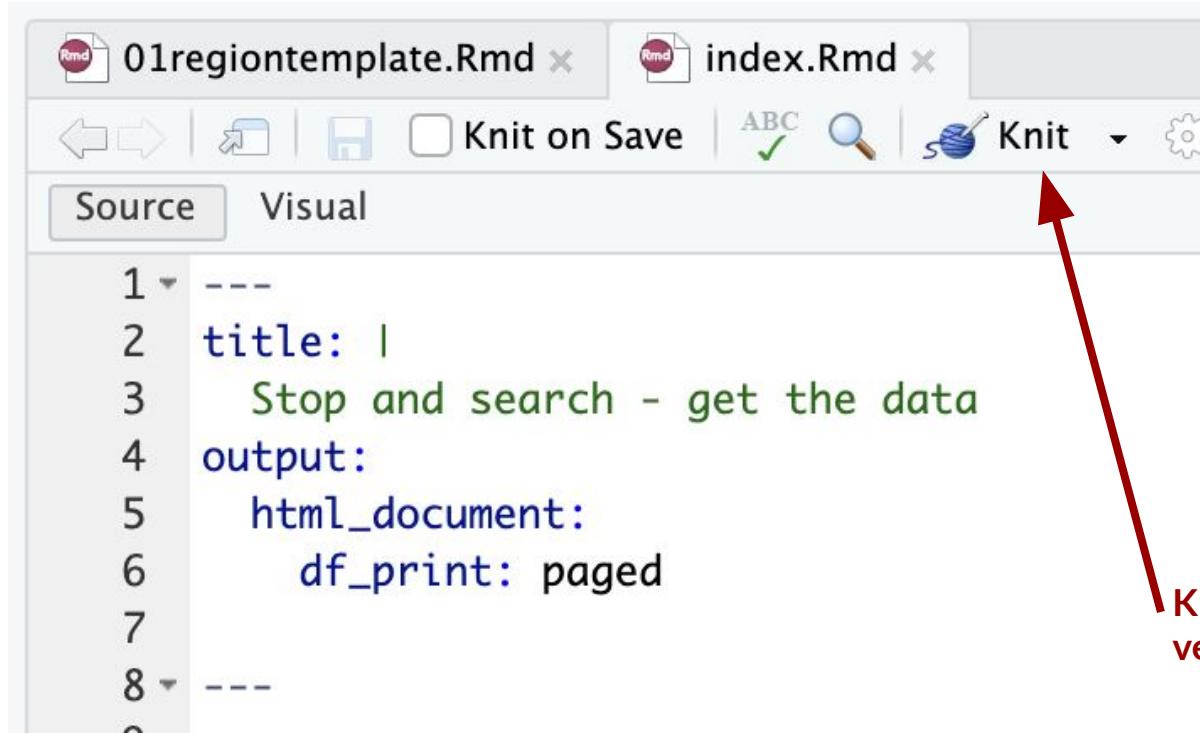


# This is the site's index.html page

---

- **.Rmd file NEEDS TO BE IN SITE FOLDER**
- **Use markdown to create the content for your page — no need for R code blocks**
- **Headings: # or ## or ###**
- **Italic, bold: \* or \*\* or \*\*\* before and after**
- **Links: [link text] (<http://url>)**
- **Indented text, e.g. quotes: >**
- **Bullet list items: \***

# Index.Rmd



01regiontemplate.Rmd x index.Rmd x

ABC Knit

Source Visual

```
1 ---
2 title: |
3 Stop and search - get the data
4 output:
5 html_document:
6 df_print: paged
7 ---
8 ---
```

Knit the file – it should generate a HTML version called index.html

# The YAML

```

```

```
title: |
 Stop and search - get the data
output:
 html_document:
 df_print: paged

```

# Example Markdown

\*\*\*This story is available for use by NEWSORG's news partners. Please do not share outside of the network. It is under strict embargo until November 7 at 0001.\*\*\*

\*\*\*Use the dropdown navigation menus across the top of the page to see the data for a specific local authority\*\*\*

# What's the story?

Here's some text for the first part of the page.

## What the experts say

Blah blah.

# Notebook 3: rendering markdown files

# This will render multiple files

---

- It will use the template Rmd file as the basis
- It needs a list of regions, and will generate one for each

# Import the data - again

```
```{r import data}
#store the URL
fileloc =
"https://docs.google.com/spreadsheets/d/e/2PACX-1vR2-w4JWtqaDmOBp7CxZpMaqsa-2Tm4vUVs1Of5\_NVfiTIavYEJWHWOv-4sGJ9VUUlx8eJPnULECeYW/pub?gid=0&single=true&output=csv"
```

```
data = read_csv(fileloc)
#just get one column - it's now a vector
regions <- data$region
```

```

This bit is new. We store a list of regions to loop through

# Loop and render

```
```{r generate md files}
#store the location of the template
paramsfile <- "01template.Rmd"
#loop through all regions
for (r in regions) {
  rmarkdown::render(paramsfile, params = list(reg = r),
output_file =
paste(sep="", 'site/' ,stringr::str_replace_all(r,
", "-"), '.md'),
  envir = parent.frame()))
}
```

```

Here is where we set the  
params\$reg to the region  
currently being looped through



# (Troubleshooting)

```
```{r generate md files}
#store the location of the template
paramsfile <- "01template.Rmd"
#loop through all regions
for (r in regions) {
  rmarkdown::render(paramsfile, params = list(reg = r),
output_file =
paste(sep="", 'site/' stringr::str_replace_all(r,
", "-"), '.md'),
  envir = parent.frame())
}
...``
```

This will cause an error if you haven't set the YAML to create the /site folder

```
30
31 - ````{r generate md files}
32 #store the location of the paramsfile to use below
33 paramsfile <- "01template.Rmd"
34 #loop through all regions
35 for (r in regions) {
36   print(r)
37   rmarkdown::render(paramsfile, params = list(reg = r), output_file
38   paste(sep="", 'site/', stringr::str_replace_all(r, " ", "-"), '.md'),
39   envir = parent.frame())
40 }
```

```
[1] "Badtown"
```

```
Error: The directory 'site' does not exist.
```

↑ Show

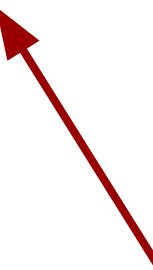
Solutions:

- **Create /site folder, or:**
- **Edit YAML in template so it specifies a lib_dir**

Notebook 4: rendering HTML files and navigation

Make a list of files

```
```{r get file names}
#get the names of all the html files
filenames <- list.files("site")
````
```



The files were all rendered into a folder called 'site' so we are asking for a list of files in that folder

Generate the YAML...

```
```{r generate yaml}
#store the string we want to start out yaml file with
yamlstring <- 'name: "reg"
navbar:
 title: "Stop and search"
 left:
 - text: "Regions"
 menu: '
````
```

This text will be used in the navigation menu that's created

The menu part is going to be filled in next...

...gather links for it...

```
#create an empty vector to store all the strings we're about to create
strvec <- c()
#loop through the filenames
for (i in filenames){
  if(substring(i,nchar(i)-2,nchar(i)) == ".md"){
    #replace spaces with dashes, and replace the file extension with .html
    htmlversion <- gsub(" ","-",gsub(".md",".html",i))
    #get the name by removing the file extension.
    textversion <- gsub(".md","",i)
    #create a string for the YAML file by inserting those
    fullstring <- paste0(
      - text: '',textversion,''
      href: ',htmlversion)
    strvec <- c(strvec,fullstring) #add to the collection of strings
  }
}
```

What's happening?

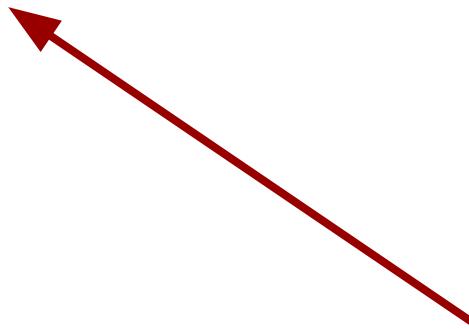
- We create an empty list
- Loop through each filename and for each:
- Extract the name to create link text
- Generate a .html version of its filename to create URL for it
- Add to the empty list

...and write as a .yml file

```
#add the initial string
strvec <- c(yamlstring, strvec)
#create a yaml file with that string of text in it
write(strvec, file = "site/_site.yml")
```

Render the HTML pages

```
```{r render site}  
#now render the site
rmarkdown::render_site("site")
```
```



It will look in the /site folder for the .yml file as the basis for rendering

| | | | |
|---|---|--|----------------|
| ▼ | site | | Today at 17:08 |
| ▼ | _site | | Today at 17:08 |
| |  Anyville.html | | Today at 17:08 |
| |  Badtown.html | | Today at 17:08 |
| |  Goodstat.html | | Today at 17:08 |
| |  index.html | | Today at 17:08 |
| > |  libs | | Today at 17:08 |
| > |  site_libs | | Today at 17:08 |
| |  Somecity.html | | Today at 17:08 |
| |  _site.yml | | Today at 17:08 |
| |  Anyville.md | | Today at 17:08 |
| |  Badtown.md | | Today at 17:08 |
| |  Goodstat.md | | Today at 17:08 |
| |  index.Rmd | | Today at 17:06 |
| > |  libs | | Today at 17:08 |
| |  Somecity.md | | Today at 17:08 |

Create a website using GitHub Pages

Put website files in a 'docs' folder — and upload to GitHub

- Cannot upload more than 100 files so upload in 3 batches:
- .html files first
- Then the /libs folder (the table uses these files)
- Then the /site_libs folder (navigation uses these)

 General

Access

 Collaborators

 Moderation options

Code and automation

 Branches

 Tags

 Rules

 Actions

 Webhooks

 Environments

 Codespaces

 Pages

GitHub Pages

[GitHub Pages](#) is designed to host your personal, organization, or project pages from a GitHub repository.

Build and deployment

 Source

Deploy from a branch ▾

Branch

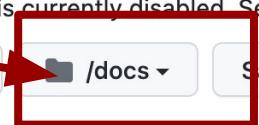
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more](#).

 main ▾

 /docs ▾

Save

 Settings



More on GitHub Pages

pages.github.com

Debugging: cleaning the HTML files

This HTML will need fixing

- <!DOCTYPE html> on page
 - Menu doesn't work on region pages
 - Menu might be too long for index page
-
- *(Other things you could do but not worth the time include: change local CSS/JS links to remote ones; remove duplicate links; etc)*



main ▾

parameterisation / rfiles / 05cleaning.Rmd ▾

 Go to file

paulbradshaw Add files via upload

7cd5b0f · 3 min

Code

Blame

181 lines (146 loc) · 5.36 KB

Raw



```
1 ---  
2   title: "05cleaning"  
3   output: html_document  
4 ---  
5  
6  
7 ## Clean the HTML outputs  
8  
9 The pages have some HTML which needs to be removed because it is being rendered as paragraph text: `<p>&lt;!DOCTYPE html&gt;</p>`.  
10  
11 We open the HTML file in a text editor and search for that text – the second instance – to identify which line it's on.  
12  
13 Then the code below seeks out that line and replaces it.|  
14  
15 ### Test on one file first  
16  
17  
18 ```{r list html files}  
19 #get the names of all the html files  
20 htmlfiles <- list.files("site/_site")  
21 htmlfiles[1]  
22 #read in the first one  
23 testfile <- readr::read_lines(paste0("site/_site/",htmlfiles[1]))
```

<https://github.com/paulbradshaw/parameterisation/blob/main/rfiles/05cleaning.Rmd>



Can you explain why the dropdown menu generated by this code doesn't work

```
<div class="container-fluid main-container">
```

```
<div class="navbar navbar-default navbar-fixed-top">
<div class="container">
  <div class="navbar-header">
    <button type="button" class="navbar-toggle c
```

Firstly, there seems to be a redundancy in the `data-target` attribute of the button element and the `id` attribute of the navbar collapse element. You have both `'data-target="#navbar'"` and 'data-bs-target="#navbar'"`. The 'data-target' attribute is used in Bootstrap 3, while the 'data-bs-target' attribute is used in Bootstrap 4 and later versions. Since you're using Bootstrap 4 syntax ('data-bs-toggle'), you should remove 'data-target="#navbar'"` and keep 'data-bs-target="#navbar'"`.`

Secondly, you didn't include the necessary JavaScript and CSS files for Bootstrap to work properly. Make sure you have included the Bootstrap CSS and JS files in your HTML code. Here's an example of how you can include them:

```
html
Copy code

<!-- CSS file -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">

<!-- JavaScript files -->
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js">
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.js">
```

BBC-Data-Unit / police_misconduct

Public

forked from paullynchmedia/police_misconduct

<> Code Pull requests Action

main police_misconduct

paulbradshaw Add files via upload

This branch is 17 commits ahead of paull

Name

..

site

00instructionsForUse.Rmd

01forcetemplateMDversion.Rmd

02parameterisationMDversion.Rmd

03renderAndClean.Rmd

iopc_widetolong.R

readme.md

```
125 We also need to clean the dropdown menu in the index page: by default this lists all the forces but is too long to fit on one screen
126
127 So we need to split it into two menus instead.
128
129 ```{r clean index menu}
130 #the index file should be the 20th in the previous list
131 htmlfiles[20]
132 #read it in
133 indexfile <- readr:::read_lines("site/_site/index.html")
134 #line 257 should be the title of the dropdown, 'Forces'
135 indexfile[257]
136 #change it
137 indexfile[257] <- "Forces A-M"
138 indexfile[257]
139 #line 325 should be the beginning of the first N force, Norfolk
140 indexfile[325:326]
141 #add HTML which closes the first part of the list and begins a second list with a dropdown button
142 #code taken from https://github.com/sduiopc/test1/blob/8938ef49cf45eb5cb67ab73974c8bdbf33aee4c5/index.html
143 #which is the version where I did this manually
144 indexfile[325] <- '</ul></li></ul><ul class="nav navbar-nav navbar-right"></ul><ul class="nav navbar-nav"><li class="dropdown">
145 indexfile[325:326]
146 #save file
147 write(x = indexfile, file=paste0("site/_site/","index.html"))
148 #remove the variable
149 rm(indexfile)
150 ...
```

Update readme.md

https://github.com/BBC-Data-Unit/police_misconduct/tree/main/rfiles

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[S](#)[main](#)[child-speech / parameterisation](#)[paulbradshaw](#) Add files via upload

Name

Last commit message



..

[01compileSENdata.Rmd](#)

Add files via upload

[01latemplate.Rmd](#)

Add files via upload

[02rendering.Rmd](#)

Add files via upload

[03renderandclean.Rmd](#)

Add files via upload

[SEN_language_covid.Rproj](#)

Add files via upload

[index.Rmd](#)

Add files via upload

[readme.md](#)

Create readme.md

<https://github.com/BBC-Data-Unit/child-speech/tree/main/parameterisation>

All that in a nutshell

- Parameterisation allows you to generate multiple files from a single template
- This can be used to build a website with a page for every region/category/row in your data
- Use GitHub Pages to easily publish that - but allow time for tweaking and problem-solving

More:

- R Markdown: The Definitive Guide has a chapter on parameterised reports