

Remotely Sensing Cities and Environments

Lecture 2: Portfolio tools: Xaringan and Quarto

02/02/2022 (updated: 16/01/2024)

 a.maclachlan@ucl.ac.uk

 [@andymaclachlan](https://twitter.com/andymaclachlan)

 [@andrewmaclachlan](https://github.com/andrewmaclachlan)

 Centre for Advanced Spatial Analysis, UCL

 [PDF presentation](#)

This lecture and practical are solely about getting started
with the two data science tools we will use



1. Xaringan for group presentations*
2. Quarto for individual portfolios



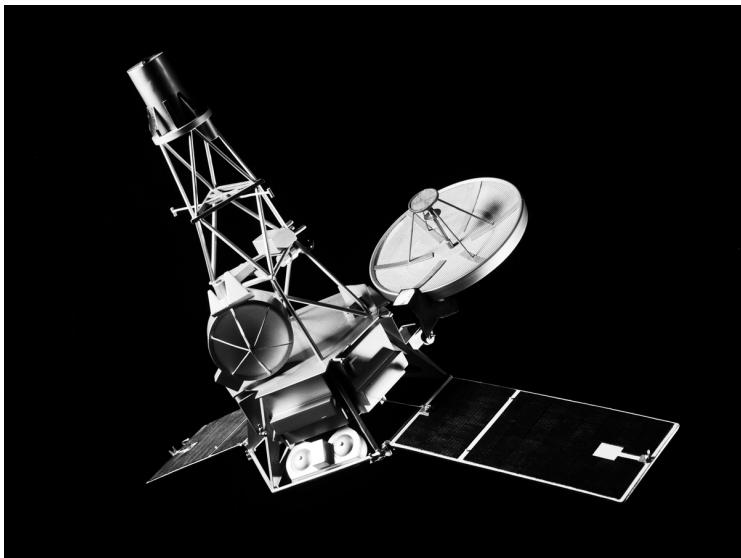
How to use the lectures

- Slides are made with [xaringan](#)
-  In the bottom left there is a search tool which will search all content of presentation
- Control + F will also search
- Press enter to move to the next result
-  In the top right let's you draw on the slides, although these aren't saved.
- Pressing the letter [o](#) (for overview) will allow you to see an overview of the whole presentation and go to a slide
- Alternatively just typing the slide number e.g. 10 on the website will take you to that slide
- Pressing alt+F will fit the slide to the screen, this is useful if you have resized the window and have another open - side by side.



Why are you doing this?

- Employers want to know you can do data science --
- this will let you **show** them as opposed to just
telling them.
- Once you get set up it's actually fairly easy to use,
the setup is the hardest part.
- Collaborating via Git and GitHub is an essential
skill that is difficult to learn outside a team setting.
- I want you to have a Portfolio of some of your work.
- Let's you show and run code in a presentation.



Mariner-C Spacecraft Model. Source:[Original from NASA](#). Digitally enhanced by [rawpixel](#).

Sessions this week

- A bit different....

Today

- 30 minutes of going through Xaringan
- 30 minutes + homework of you creating a **small** presentation
- This will be part of the assessment this week

Thursday

- 30 minutes of going through Quarto
- 1 hour creating a Quarto site
- This forms the template for the main assessment component of the module

By the end of the week you'll have an online portfolio!





Xaringan = shar-in-gen or [ʃæ.ˈrin.gæn]..

Xaringan = shar-in-gen



Strongest Sharingan of the List | Ninja World. [Source:Pinterest](#)

A lot of the concepts covered today are the same for Quarto,
which we cover later in the week



Xaringan

This presentation provides **an outline of the tools**...some more information is provided in the practical.

- 🧑 Xaringan = shar-in-gen or [ʃæ.'riŋ.gæn]..
- Created by Yihui Xie....also created Bookdown...
- Install the package from GitHub
- You may also need the remotes package

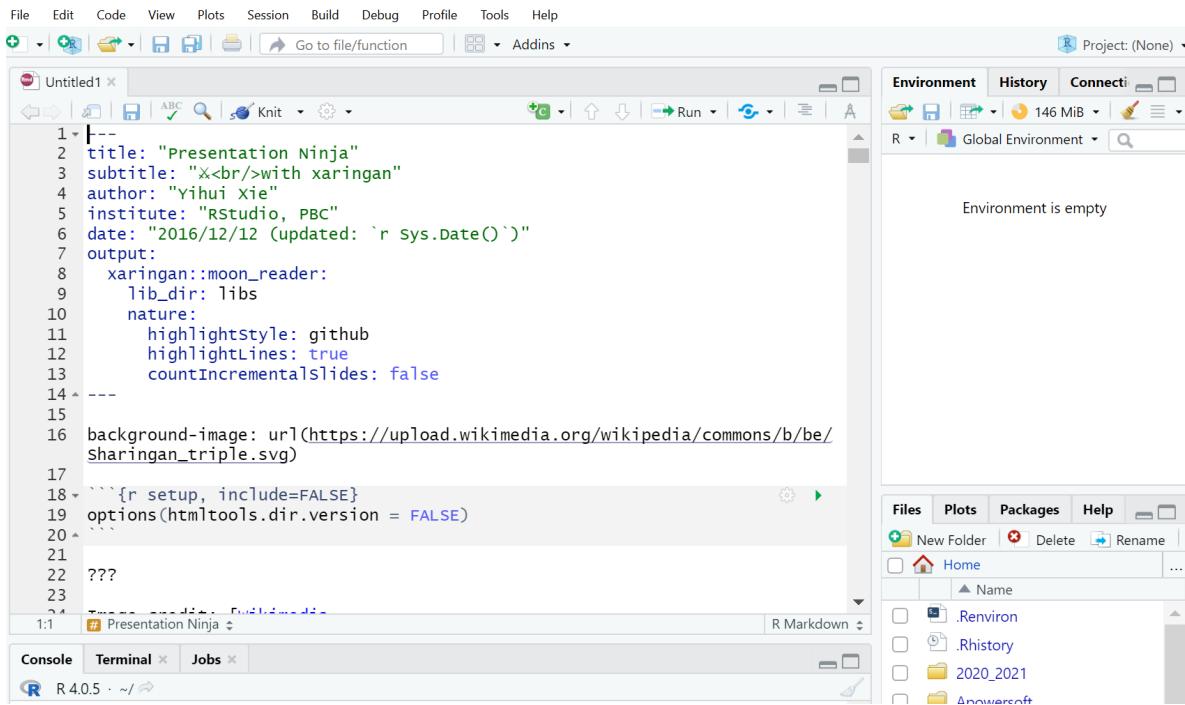
```
remotes::install_github("yihui/xaringan")
```

- In RStudio:
 - Make a new project
 - File -> New File -> R Markdown -> From Template -> Ninja Presentation



Xaringan 2

- Make a new project > click Knit



```
1 ---  
2 title: "Presentation Ninja"  
3 subtitle: "x<br/>with xaringan"  
4 author: "Yihui Xie"  
5 institute: "RStudio, PBC"  
6 date: "2016/12/12 (updated: `r sys.Date()`)"  
7 output:  
8   xaringan::moon_reader:  
9     lib_dir: libs  
10    nature:  
11      highlightstyle: github  
12      highlightLines: true  
13      countIncrementalSlides: false  
14 ---  
15  
16 background-image: url(https://upload.wikimedia.org/wikipedia/commons/b/be/sharingan\_triple.svg)  
17  
18 ``{r setup, include=FALSE}  
19 options(htmltools.dir.version = FALSE)  
20 ``  
21  
22 ???  
23  
24  Presentation Ninja  
1:1 # Presentation Ninja
```

Environment is empty

Files Plots Packages Help

New Folder Delete Rename

Home

Name

.Renviron .Rhistory

2020_2021

Dowersoft

Console Terminal Jobs

R 4.0.5 ~



Don't panic



Start at the top

```
---
```

```
title: "Presentation Ninja"
subtitle: "x<br/>with xaringan"
author: "Yihui Xie"
institute: "RStudio, PBC"
date: "2016/12/12 (updated: `r Sys.Date()`)"
output:
  xaringan::moon_reader: # output format
    lib_dir: libs #used for dependent libraries
    nature:
      highlightStyle: # how the code displays
      highlightLines: true
      countIncrementalSlides: false
      #ratio "16:9" # or 4:3
---
```

- **Highlight options:** arta, ascetic, dark, default, far, github, googlecode, idea, ir-black, magula, monokai, rainbow, solarized-dark, solarized-light, sunburst, tomorrow, tomorrow-night-blue, tomorrow-night-bright, tomorrow-night, tomorrow-night-eighties, vs, zenburn.



Writing slides

There are two "types" of slide.

Each slide is separated by a ---, this denotes a new slide

- A normal slide (like this one) is just...

```
---
```

```
# Writing slides
```

There are two "types" of slide.

- An inverse slide (the ones with just words on and the blue background)...example on next slide

```
---
```

```
class: inverse, center, middle
```

```
# Writing slides
```

There are two "types" of slide.



Writing slides

There are two "types" of slide.



Inverse slides are useful for breaking up slides with text and making a specific point



If you don't use inverse you just get a white (or other, depending on your colourscheme) background...

```
class: center, middle
```

```
# If you don't use inverse you just get a white (or other, depending on your colourscheme) background...
```



Slide tools

```
# Slide tools
```

Sub heading

```
## Sub heading
```

sub sub heading

```
### sub sub heading
```

- Bullet point
 - Sub bullet point

- * Bullet point
 - * Sub bullet point



Slide tools 2

1. List
2. List 2

```
1. List  
1. List 2 # yes, it's 1 for both, R knows it's a list.
```

To add a break in a slide, meaning you have to click to advance **on the specific slide...**

--

- To have things on the left

```
.pull-left[  
  To have things on the left  
]
```

- To have things on the right

```
.pull-right[  
  To have things on the right  
]
```

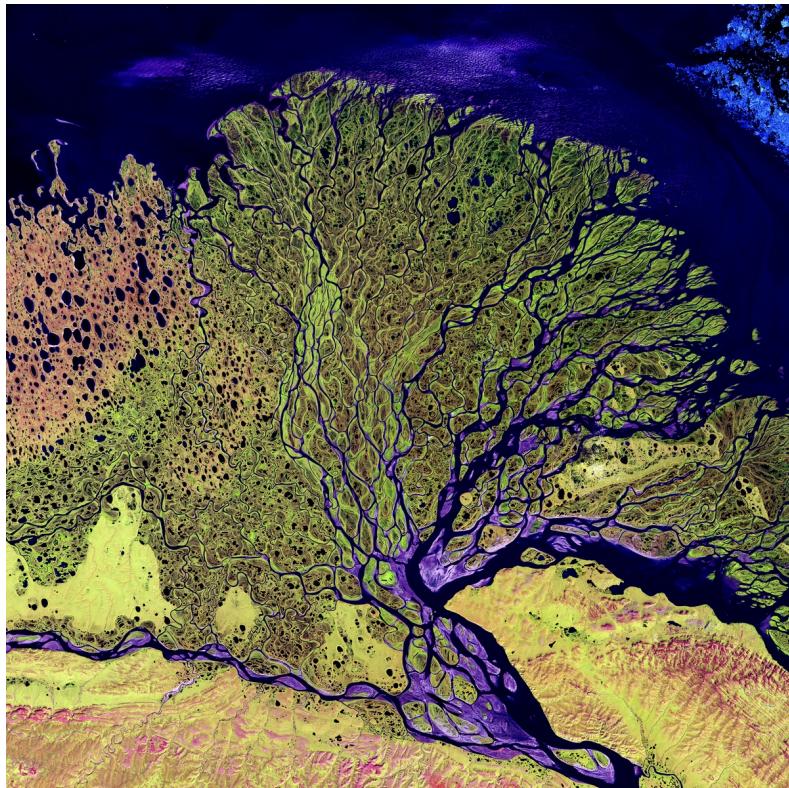
Remember then a new slide is starts with ...



Images

...it's massive!...we need to remember code chunk options....

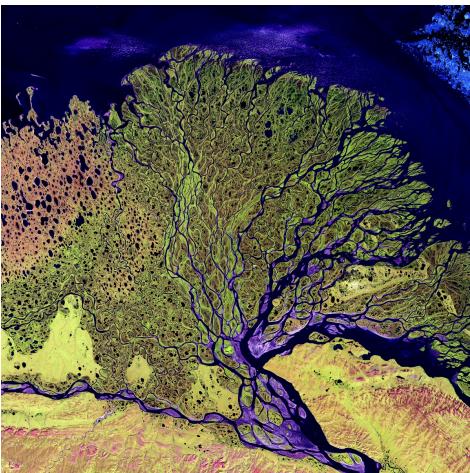
```
knitr::include_graphics('img/Lena-river.jpg')
```



Images 2

Common code chunk options

- Echo = TRUE, shows code or FALSE = doesn't
- fig.align='center'
- out.width= 'x%'



The Lena River, some 2,800 miles long, is one of the largest rivers in the world. Source: [Original from NASA](#). Digitally enhanced by rawpixel



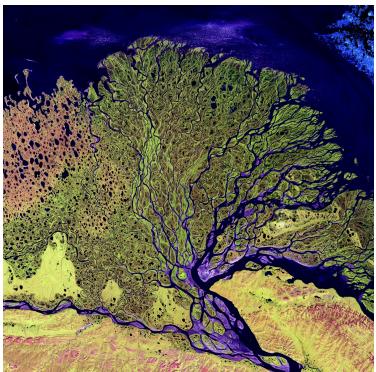
```
{r echo=FALSE, out.width='60%', fig.align='center'}  
knitr::include_graphics('img/Lena-river.jpg')
```

Images 3

- Creating hyperlinks [*the linking text*] (*the URL*)

[make a link] (<https://www.rawpixel.com/image/440230/free-photo-image-nasa-russia-earth>)

- You can use `.pull-left[]` and `.pull-right[]` with images too.



- Some text

The Lena River, some 2,800 miles long, is one of the largest rivers in the world.

Source: Original from NASA. Digitally enhanced by rawpixel



Tables

```
library(tidyverse) #where the mpg data is stored  
  
## Warning: package 'tidyverse' was built under R version 4.2.3  
  
## Warning: package 'tibble' was built under R version 4.2.3  
  
knitr::kable(head(mpg), format = 'html')
```

manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact



Tables 2

If you need to create a table....

- write a data set in excel or Google sheets
- load that into R (not showing the code to load it)
- wrangle anything you don't want
- use the code on the previous slide to display it



Citing

A bit difficult in Xaringan, but Zotero can now continuously update your `.bib` file.

To do so:

1. Download the better-bibtex latest release --- the `.xpi` file: <https://github.com/retorquere/zotero-better-bibtex/releases>
2. In Zotero > Tools > Add ons > Extensions
3. Select the cog > Install add on from file
4. Select the `.xip` > restart Zotero.

On the restart select the default naming convention.

To export the library or collection

1. File > export or right click the collection and export
2. Select Better BibLaTex
3. Click keep updated
4. Select the file to save into your project



Citing 2

- Install RefMangerR

```
remotes::install_github("ropensci/RefManageR")
```

- Add the following code near the top of your slides

```
library(RefManageR)
BibOptions(check.entries = FALSE,
           bib.style = "authoryear",
           cite.style = "authoryear",
           style = "markdown",
           hyperlink = TRUE,
           dashed = FALSE,
           no.print.fields=c("doi", "url", "urldate", "issn"))
myBib <- ReadBib("./Bib.bib", check = FALSE)

## Loading required namespace: bibtex
```



Citing 3

- load `knitcitations`

```
library(knitcitations)
```

- Then use

```
`r Citet(myBib, "wulderOpeningArchiveHow2012")` or  
`r Citep(myBib, "lovelandLandsatBuildingStrong2012")` or  
`r Citep(myBib, "lovelandLandsatBuildingStrong2012", "wulderOpeningArchiveHow2012")`
```

To create:

- Wulder, Masek, Cohen, Loveland, and Woodcock (2012)
- (Loveland and Dwyer, 2012)
- (Loveland and Dwyer, 2012; Wulder, Masek, Cohen, et al., 2012)



Citing 4

Finally, to add a reference list use the following with the chunk with options `results='asis'`, `echo=FALSE`

```
PrintBibliography(myBib, start = 1, end = 7)
```

Loveland, T. R. and J. L. Dwyer (2012). "Landsat: Building a Strong Future". In: *Remote Sensing of Environment* 122, pp. 22-29.

Wulder, M. A., J. G. Masek, W. B. Cohen, et al. (2012). "Opening the Archive: How Free Data Has Enabled the Science and Monitoring Promise of Landsat". In: *Remote Sensing of Environment* 122, pp. 2-10.



Live preview of slides

It can be annoying to have to click Knit every time.

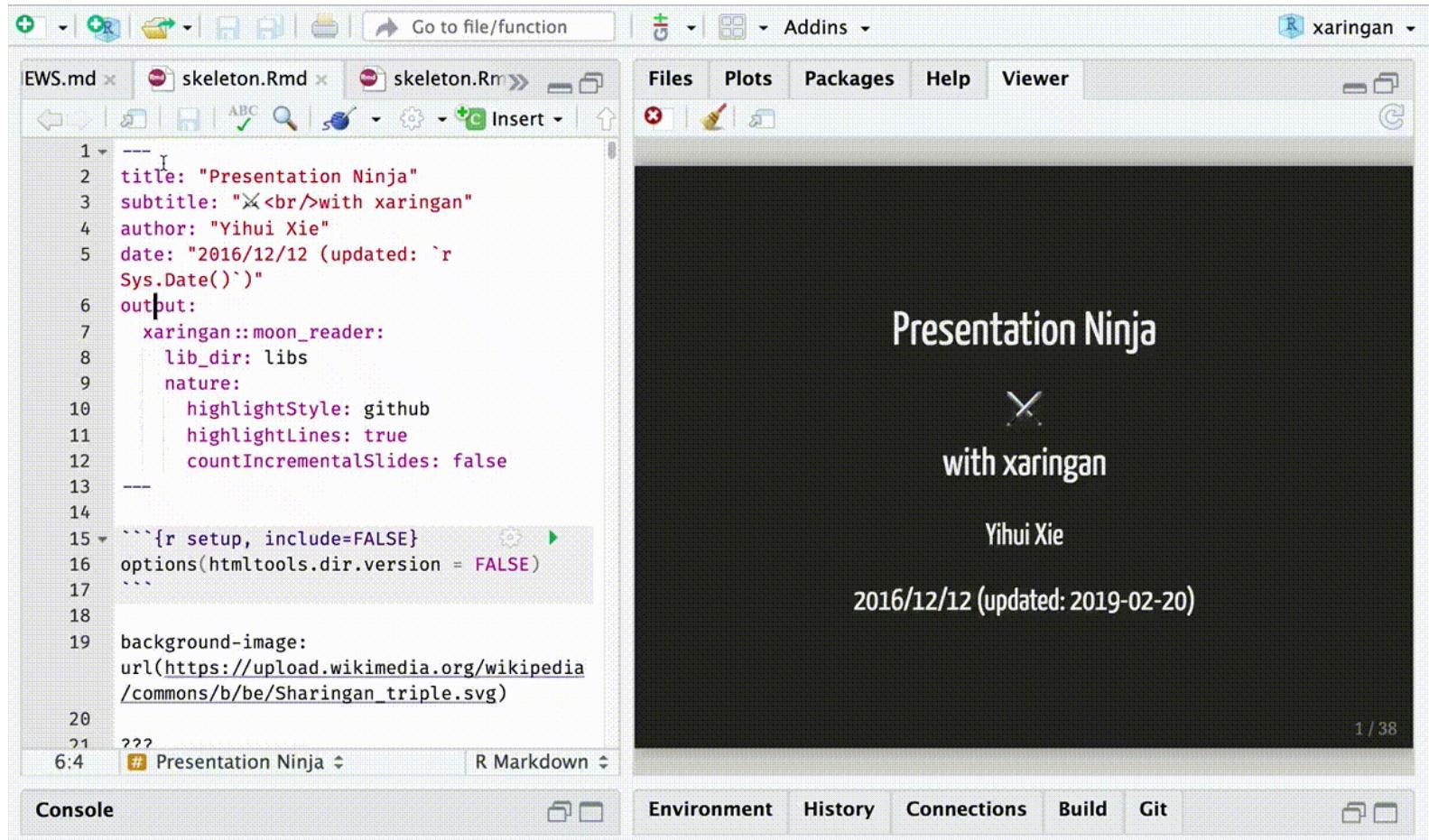
The Add In "Infinite Moon Reader" from Yihui Xie generates the slides every time you save the .Rmd.

Generates the slides on the viewer tab --- bottom right of RStudio

As you scroll through the slides the viewer tab updates with the slide you are on...



Live preview of slides



The screenshot shows the RStudio interface with a live preview of an xaringan presentation slide. The left pane displays the R Markdown code for the slide, and the right pane shows the rendered presentation.

```

1 ---  

2 title: "Presentation Ninja"  

3 subtitle: "with xaringan"  

4 author: "Yihui Xie"  

5 date: "2016/12/12 (updated: `r  

Sys.Date()`)"  

6 output:  

7   xaringan::moon_reader:  

8     lib_dir: libs  

9     nature:  

10       highlightStyle: github  

11       highlightLines: true  

12       countIncrementalSlides: false  

13 ---  

14  

15 ```{r setup, include=FALSE}  

16 options(htmltools.dir.version = FALSE)  

17 ```  

18  

19 background-image:  

url(https://upload.wikimedia.org/wikipedia/commons/b/be/Sharingan\_triple.svg)  

20  

21 ???

```

The rendered slide content includes:

- Title: Presentation Ninja
- Subtitle: with xaringan
- Author: Yihui Xie
- Date: 2016/12/12 (updated: 2019-02-20)
- Background image: A Sharingan eye pattern (blue, red, white) from the anime Naruto.

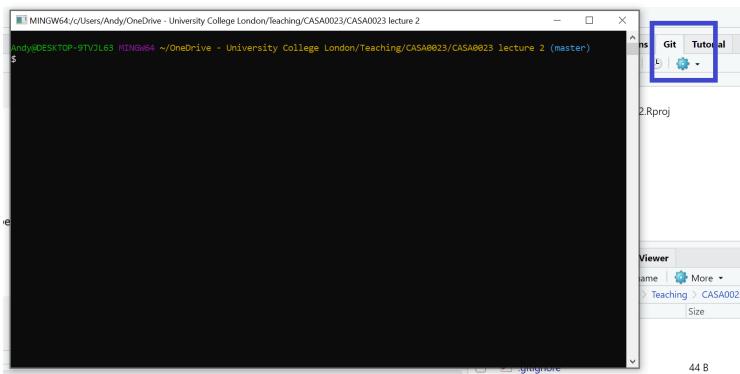
At the bottom of the slide, it says "1 / 38".

The Ultimate Infinite Moon Reader for xaringan Slides. Source: [Yihui Xie](#)

Use Git

- Same as we saw in CASA0005
- Store your images in a folder, i use one called img
- Any additional libraries you use that are automatically loaded go into the libs folder (should be automatically created)
- It will become quicker to use the shell as opposed to the RStudio git GUI...
- Commands:

```
git add .  
  
git commit -m "my commit message"  
  
git push
```



Use Git 2

Remember to ignore any large files you might have...

When you initialize git a .gitignore is created...



GitHub maximum file upload of 50mb. Source:[reddit r/ProgrammerHumor](#)



Hosting on GitHub

- Few ways we can do this...like CASA0005
 1. `use_github()` from the `usethis` package will automatically create a repo of the same name as your RProject.
 2. Or create a repository on GitHub
 - Then use the code provided in the shell for your project that has git enabled
 - "...or push an existing repository from the command line"
 - Let's you call the repo something other than your RProject name....
 - to do this you must **first** have committed some files to your git already.
 3. To "deploy" the presentation on GitHub pages
 - `usethis::use_github_pages(branch = "main")` OR
 - On GitHub repo > settings > Pages > select source as main > Save



For the presentation it's expected all group members will contribute on Git

Remember pushing and pulling?



Themes

- Xaringan can be styled using Cascading Style Sheets (CSS)
- But there are many themes within the Xaringan itself

```
names(xaringan:::list_css())
```

```
## [1] "chocolate-fonts"   "chocolate"      "default-fonts"
## [4] "default"           "duke-blue"     "fc-fonts"
## [7] "fc"                 "glasgow_template" "hygge-duke"
## [10] "hygge"              "ki-fonts"       "ki"
## [13] "kunoichi"          "lucy-fonts"    "lucy"
## [16] "metropolis-fonts"  "metropolis"    "middlebury-fonts"
## [19] "middlebury"         "nhsr-fonts"    "nhsr"
## [22] "ninjutsu"          "rladies-fonts" "rladies"
## [25] "robot-fonts"        "robot"         "rutgers-fonts"
## [28] "rutgers"            "shinobi"       "tamu-fonts"
## [31] "tamu"               "uiو-fonts"     "uiو"
## [34] "uo-fonts"           "uo"             "uol-fonts"
## [37] "uol"                "useR-fonts"    "useR"
## [40] "uwm-fonts"          "uwm"           "wic-fonts"
## [43] "wic"
```



Themes 2

```
---
```

```
title: "Presentation Ninja"
subtitle: "X<br/>with xaringan"
author: "Yihui Xie"
institute: "RStudio, PBC"
date: "2016/12/12 (updated: `r Sys.Date()`)"
output:
  xaringan::moon_reader:
    css: [default, metropolis, metropolis-fonts]
    lib_dir: libs
  nature:
    highlightStyle: github
    highlightLines: true
    countIncrementalSlides: false
---
```



Add ons / useful resources

Gerrick Aden-Buie has many Xaringan extensions

They are refreshingly easy to use:

xaringanExtra

- Search
- Clipboard
- Tile view
- Hover over code

xaringanthemer



Presentation Ninja

✗ xaringan +
☕ xaringanthemer

Yihui Xie

Gerrick Aden-Buie

2020-04-18

Hello World

Install the `xaringan` package from Github:
`devtools::install_github("yihui/xaringan")`

You are recommended to use the RStudio IDE, but you do not have to.

- Create a new R Markdown document from the menu `File -> New File -> R Markdown -> From Template -> Ninja Presentation!`
- Click the Knit button to compile it;
- or use the RStudio Addin? "Infinite Moon Reader" to live preview the slides (every time you update and save the Rmd document, the slides will be automatically reloaded in RStudio Viewer).

2 / 2

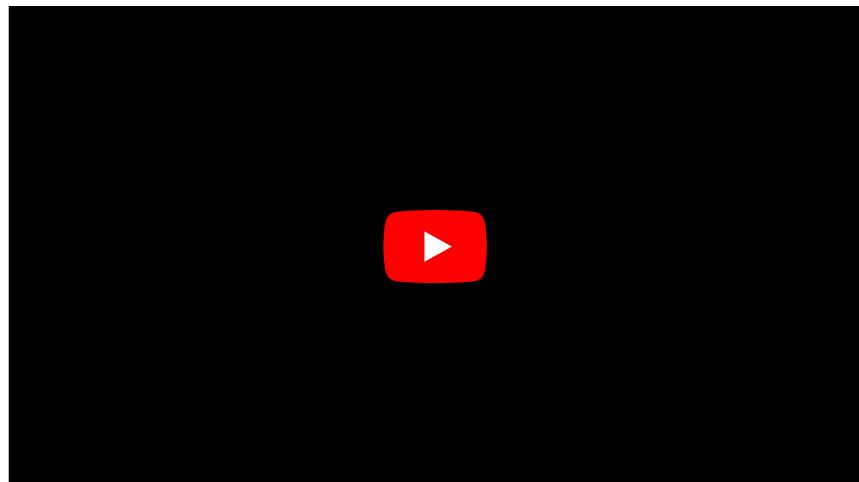
Source: [xaringanthemer](#)

Resources:

- Sydney Xaringan theme
- Alison Hill's Making slides in R Markdown



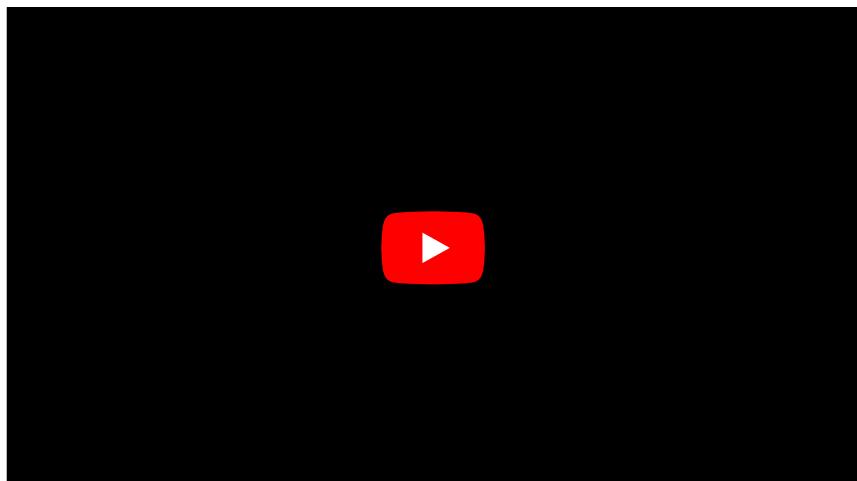
Add ons / useful resources 2



Extra info

The slides are styled with **CSS** = Cascading Style Sheets, used to style web pages

I almost exclusively use templates, but this video is helpful and will assist you with understanding how the slides are made.



Keep it simple

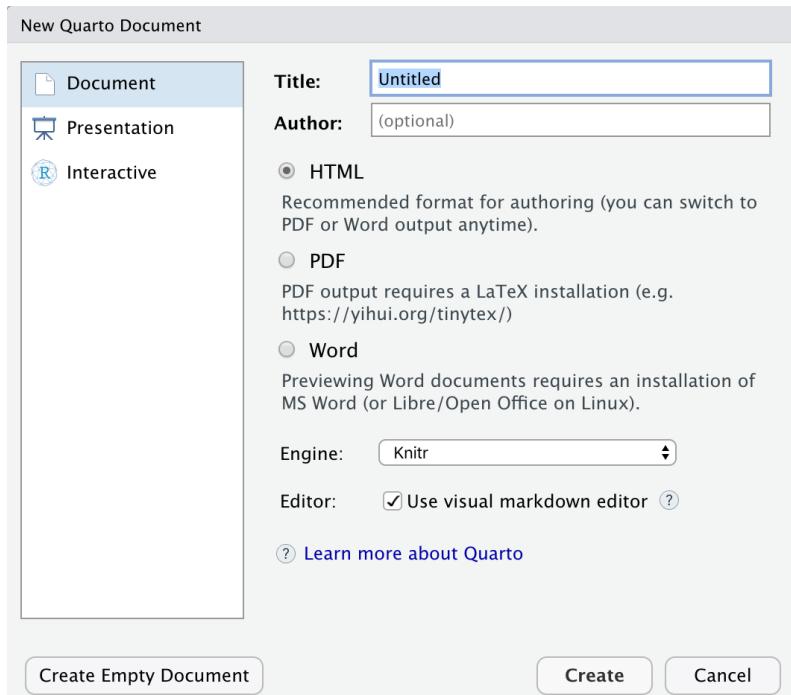
Short example in R of slides, formatting and citing...



Quarto presentations

- Very similar **but less templates**
- Much easier to reference as it is built in
- Basic intro:
<https://quarto.org/docs/tools/rstudio.html>
- Some slides from Andy:
 - [Code](#)
 - [Slides](#)

.pull-right[



Source: [R Quarto](#)

Task

30 minutes now + homework

Create a small 5 slide presentation and host it on GitHub - place the link in your Quarto portfolio

- Select a sensor of your choice (any)
- Create a short (maximum 9 slide, not including reference slide) presentation on the sensor in xaringan

The presentation will **be marked in the same manner set out in mark scheme** for the learning diary. Specifically:

- The **summary criterion** will refer to the summary of the sensor you have selected.
- The **application criterion** will refer to examples of studies that have used the data from the sensor and their purpose
- The **reflection criterion** will refer to what you have learnt in relation to the sensor, its use and how the data might be used in future work.



Task 2 (after the next session this week)

On Thursday

- There will be another short lecture
- We will make our individual Quarto workbooks

Quarto workbook

- The production of the Quarto learning diary is not marked, but it is required for component 2.
- Once you have made your Quarto document update it with your learning diary entry from week 1, and the Xaringan presentation from week 2.
- It is recommended you use chapters for each week and that you **do not** run the practical code in the learning diary
- Have a separate project for your code and a separate project for your learning diary





End of session 1



Start of session 2



bookdown.. > Quarto

Bookdown

Pros:

- Makes a portfolio
- One set of markdown files for multiple formats
- Auto updates figures as the file name is the same
- Set up the style then never worry about it again

Cons:

- Longer time to learn
- Can be tempting to try and change formatting
- Doesn't work with other programming languages (easily)



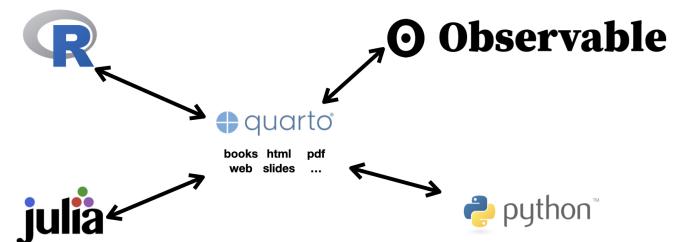
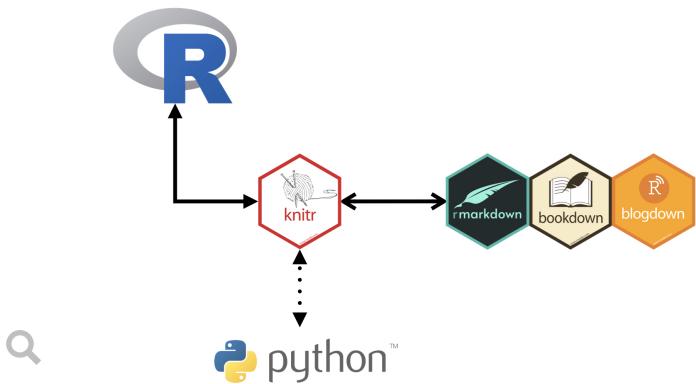


Quarto

Quarto was officially (?) released in 2022, just after I wrote all this in Bookdown

Quarto:

- let's you use whatever language you want (Python, R, Observable)
- you don't need the other installed, e.g. just python not R
- it works with the VS code IDE
- it is a separate piece of software
- removes some of the complexity of bookdown...
- output to anything - website, presentation, blog, book



Nick Tierney. Source:[Notes on Changing from Rmarkdown/Bookdown to Quarto](#)

Less complex

Bookdown

```
directory
├── .Rproj.user
├── book.Rproj
├── 00-preamble.Rmd
├── 01-intro.Rmd
├── 02-literature.Rmd
├── 03-method.Rmd
├── 04-results.Rmd
├── 05-discussion.Rmd
├── 06-conclusion.Rmd
├── 07-references.Rmd
├── 08-appendix.Rmd
└── README.md
    └── bookdown.yml #how it generates
    └── output.yml # output options like pdf or gitbook
        └── book.bib
        └── index.Rmd # headers for the book
        └── preamble.tex # style for LaTeX
        └── style.css # style for website
```



source: <https://arm.rbind.io/slides/bookdown.html#12>

Important files (bookdown)

`_output.yml`

```
bookdown::gitbook:
  css: style.css
  config:
    toc:
      collapse: section
      scroll_highlight: yes
      before: |
        <li class="toc-logo"><a href="https://www.ucl.ac.uk/casa/thesis"></a></li>
      after: |
        <li><a href="" target="blank"></a></li>
  edit: https://github.com/andrewmaclachlan/CASA-MS
  download: ["pdf", "rmd"]
  sharing:
    github: yes
    facebook: no
    twitter: no
    all: no
bookdown::pdf_book:
  toc: false
  includes:
    in_header: preamble.tex
  latex_engine: xelatex
  citation_package: biblatex
  keep_tex: yes
#bookdown::epub_book: default
```



`_bookdown.yml`

```
output_dir: docs
book_filename: "CASA-Thesis"
rmd_files:
- index.Rmd
- 00-preamble.Rmd
- 01-intro.Rmd
- 02-literature.Rmd
- 03-method.Rmd
- 04-results.Rmd
- 05-discussion.Rmd
- 06-conclusion.Rmd
- 07-references.Rmd
- 08-appendix.Rmd
language:
  ui:
    chapter_name: "Chapter "
  delete_merged_file: true
```

Quarto

- Combined the `_output.yml` and `_bookdown.yml` into a single file
- Easily change the output, there are a few ways to do this:
 - website - change the `_quarto.yml` to something like this
 - slides - need no for `_quarto.yml`

`_quarto.yml`

```
project:
  type: book

book:
  title: "Quarto_example"
  author: "Jane Doe"
  date: "17/05/2022"
  chapters:
    - index.qmd
    - intro.qmd
    - summary.qmd
    - references.qmd

  bibliography: references.bib

format:
  html:
    theme: cosmo
  pdf:
    documentclass: scrreprt

editor: visual
```



Slides in Quarto

- Don't need a `_quarto.yml`
- Just add the header to the top of one `.qmd`
- Similar commands to Xaringan **but some small differences**
- Not as many templates as Xaringan yet

`slides.qmd`

```
---
```

```
title: "Quarto Presentations"
subtitle: "Create beautiful interactive slide decks w
format:
revealjs:
  slide-number: true
  chalkboard:
    buttons: false
  preview-links: auto
  logo: images/quarto.png
  css: styles.css
  footer: <https://quarto.org>
```

```
---
```



Tables

- For the example ones, I've made an excel document that is then read into R and formatted with the `kable` package.
- A massive benefit of using Quarto is now you can edit your RMarkdown documents with the visual editor
- This also includes the ability to insert your own manual tables
- But could you export a `.csv` from your analysis that is then read straight into R?



Tables 2

Packages

Code

Table

Code 2

Table 2

```
library(tidyverse)
library(knitr)
library(kableExtra)
library(readxl)
library(fs)
library(here)
```



Tables 2

Packages **Code** Table Code 2 Table 2

```
#read in data
read_excel(here("tables", "abbreviations.xlsx"))%>%
  arrange(Term) %>% # i.e. alphabetical order by Term
  # booktab = T gives us a pretty APA-ish table
  knitr::kable(booktabs = TRUE)%>%
  kable_styling(position = "center")%>%
  # any specific row changes you want
  row_spec(.,
  row=0,
  bold = TRUE)
```



Tables 2

[Packages](#)[Code](#)[Table](#)[Code 2](#)[Table 2](#)

Term	Abbreviation
Digital Elevation Model	DEM
Digital Surface Model	DSM
Digital Terrain Model	DTM



Tables 2

Packages Code Table **Code 2** Table 2

```
read_excel(here("tables", "policy.xlsx"))%>%  
  mutate_all(~ replace_na(.x, "")) %>%  
  # booktab = T gives us a pretty APA-ish table  
  knitr::kable(longtable = T, booktabs = T,  
               caption = 'Relevant influential international, metropolitan and local UHI and urban expansion policies')  
  kable_styling(position = "center", full_width = T)%>%  
  # any specific row changes you want  
  row_spec(.,  
           row=c(0,1,8, 18),  
           bold = TRUE)%>%  
  column_spec(1, width = "14cm")%>%  
  row_spec(c(1, 8, 18), hline_after = T)
```



Tables 2

[Packages](#)[Code](#)[Table](#)[Code 2](#)[Table 2](#)

Relevant influential international, metropolitan and local UHI and urban expansion policies, strategies and assessments (with publication date) referred to in this paper. * Denotes documents that lack specific UHI related policy but recognise the value of maintaining vegetation.

Policy

International

United Nations The World Cities in 2016 (2016)

United Nations New Urban Agenda (2017)

ARUP City Resilience Framework (2015)

United Nations International Strategy for Disaster Reduction Sendai Framework (2015)

Universal Sustainable Development Goals (2015)

Biological Diversity, Cities and Biodiversity Outlook (2012)

Metropolitan

AECOM Australia, Economic Assessment of the Urban Heat Island Effect, Melbourne (2012)

The Spatial Development Strategy For Greater London (2017)

Western Australia Planning Commission, Perth and Peel @3.5 million (2015)

City of Johannesburg Metropolitan Municipality, Spatial



Cross referencing

Now the difficult files are out of the way.

- Each chapter is stored in its own `.qmd`.
- You should recall the heading types `#` and `##`
- So we can have: `# Introduction {#sec-introduction}`
- To reference the section it's `@sec-introduction`
- A figure caption is set in the code chunk ...`nice-fig, fig.cap='Here is a nice figure!'`
- Before the code use `#| label: fig-example`
- Then to reference figure in the text it's `@fig-example`
- It's the same for tables but `#tbl-`
- See [Quarto cross referencing](#) for more info



Example

```
![Elephant](elephant.png){#fig-elephant}
```

See @fig-elephant **for** an illustration.



Equations

```
p= h\frac{c}{\varrho}{#eq-example}
```

$$p = h \frac{c}{\varrho}$$

- Then to reference figure in the text it's `@eq-example`

You can also use \$ to add a quick equation e.g. `ρ` gives ρ

For more equation guidance consult [Mathematics in R Markdown](#)



Citing in Quarto

It's much easier:

- Quarto should recognise your Zotero library and folders!
- Change to visual model
- Click Insert > citation! It will add the citation to a local .bib

This just adds the following...

[@xie2015] giving (Xie 2015).

If you remove the citation from the [] it will give Xie (2015).

When citing multiple authors use a ;[@xie2015; @maclachlan2017urban] = (Xie 2015; MacLachlan et al. 2017)

To remove the author add a minus sign [-@maclachlan2017urban] = (2017)



Citing example

Screenshot of a Quarto editor interface showing the citation insertion process.

The main window displays a document titled "Preface". The left sidebar contains a "Chunk 1" section with R code and output, and a "Console" tab showing R session details.

A modal dialog titled "Insert Citation" is open, listing various sources for citations:

- My Sources
 - Bibliography
 - Zotero
 - My Library
 - From DOI
 - Crossref
 - DataCite
 - PubMed
 - R Package

The "Search for citation" field is empty. Below it, a list of citation entries is shown:

- @abramov2004 Abramov, O, and McEwen, A 2004 Technical note: An evaluation of interpolation methods for Mars Orbiter Laser Altimeter...
- @adams2004 Adams, J, and White, M 2004 Is the inverse care law no longer operating?
- @adams2013 Adams, M 2013 Mega-fires, tipping points and ecosystem services: Managing forests and woodlands in...
- @adebiyi2022 Adebiyi, B, Donga, G, Omukunyi, et al. 2022 How South African Families Protected Themselves during the COVID-19 Pandemic: A Q...
- @adeel2002 Adeel, Z, and Pomeroy, R 2002 Assessment and management of mangrove ecosystems in developing countries
- @adhikari2014 Adhikari, S, Dixit, A, Basyal, et al. 2014 Urban Risk Atlas of Nepal: A Tool for Disaster Risk Reduction Planning at Local Govern...
- @ajebon2013 Ajebon, M, and Asikhia, M 2013 Geographical Analysis of Smoking in Leeds: A GIS-Based Location-Allocation Technique...

At the bottom of the modal, there are buttons for "Selected Citation Keys", "Add to bibliography: references.bib", "Use in-text citation", "Insert", and "Cancel".

**Do not do your analysis and thesis in the same R project. It's a
bad idea**



Hosting

- Don't mess around with the file structure
1. Go to the `_quarto.yml` file and make sue that that you have this line of code: `output-dir: docs` (it should be there)
 2. Build your book locally, close the preview window
 3. Save, stage changes, commit and then push to GitHub
 4. On your GitHub repository > settings > GitHub pages > select the source as main and the folder as docs
 5. Make sure you build your `gitbook`, if you don't and just save and commit it won't render online!



Word and PDF

Typically to export to word you need a word template...

In terminal (in R)

```
$ quarto pandoc -o custom-reference-doc.docx \ --print-default-data-file reference.docx
```

Then in the `_quarto.yml` add:

See [Word Templates for customisation \(e.g. headings and font\)](#)

```
format:  
  html:  
    theme: cosmo  
  pdf:  
    documentclass: scrreprt  
  docx:  
    reference-doc: custom-reference-doc.docx
```

To export to PDF you need to run `quarto install tinytex` in the terminal in R..



Final tips

- If you are trying to make a complicated table or format a page a certain way....don't. It's much easier to accept the defaults and work around them to include the required information
- In Quarto you can do all this in the visual editor! It makes it much easier.
- Xaringan doesn't work with visual editor



Task

30 minutes + homework

For the individual assignment of the module you are required to produce a online portfolio of independent study and response questions. In the practical sessions this week you should have created your online portfolio and added your response from week 1 to it. For week 2 you must add the URL of your small 9 slide presentation.





End of session 2