

PSYC3361 internship

Jenny Richmond

2024-05-18

Table of contents

Let's learn to code in R	4
how to use this book	4
1 Week 1	5
2 Welcome to coding in R!	6
2.1 Resources	7
2.2 Learning outcomes	7
2.3 Getting started in RMarkdown	7
2.3.1 The beginning	7
2.3.2 Stating the problem	7
2.3.3 Starting markdown	8
2.3.4 More markdown	8
2.3.5 RMarkdown	8
2.3.6 Custom documents	8
2.3.7 Inserting equations & sharing your document	8
2.4 Say hello to data	9
2.4.1 Read and glimpse your data	9
2.4.2 What is the “pipe”?	9
2.4.3 How to write data	9
3 Week 2	10
4 Data visualisation	11
4.1 Resources	12
4.2 Learning outcomes	12
4.3 ggplot	12
4.3.1 Setting up RStudio Cloud	12
4.3.2 Make a scatterplot	12
4.3.3 Writing a script	12
4.3.4 What are aesthetics?	13
4.3.5 Aesthetics vs parameters	13
4.3.6 Global and local	13
4.3.7 Named and unnamed argument	13
4.3.8 Facets and prettifying	13

5	Week 3	14
6	Data wrangling	15
6.1	Resources	15
6.2	Learning outcomes	16
6.3	dplyr, dance with data	16
6.3.1	dealing with variable names	16
6.3.2	subset using <code>filter()</code>	16
6.3.3	sort using <code>arrange()</code>	16
6.3.4	make your data smaller using <code>select()</code>	16
6.3.5	make new variables with <code>mutate()</code>	17
6.3.6	join data using <code>bind_rows()</code>	17
6.3.7	make wide data long with pivot functions	17
6.3.8	joining data with <code>left_join()</code>	17
7	Week 4	18
8	Reproducible projects	19
8.1	Resources	19
8.2	Project Structure	19
8.2.1	naming things for machines	19
8.2.2	naming things for humans	19
8.2.3	naming things (make them sortable/searchable)	20
8.2.4	filepaths	20
8.2.5	templates	20
8.2.6	where do things live?	20
9	Week 5	21
10	Drama-free installation	22
10.1	Resources	22
10.2	Installing R and RStudio	22
10.2.1	... on a Mac	22
10.2.2	... on Windows	22
10.2.3	... on Ubuntu	22
10.3	Installing and loading R packages	23
10.3.1	Extra info re packages on Linux	23
10.3.2	Installing R packages from Github	23

Let's learn to code in R

This book contains the coding modules (slides, videos, RStudio cloud links, self test challenges) for PSYC3361 Research Internship.

how to use this book

This book lives online so you might want to bookmark in your browser for easy access.

There is one chapter dedicated to Weeks 1-5; you can navigate the chapters using the side bar on the left.

In each chapter, there are links to...

 Slides

 Videos

 RStudio Cloud projects

 Self test resources

Each week we recommend that you...

- watch the videos
- do the exercises that Danielle recommends within them
- AND test yourself by trying the self test challenges

Learning to code is an active process. You cannot learn to code by watching someone else do it

The exercises and self test challenges are KEY LEARNING EXPERIENCES.

Please try and complete the WATCH part of your learning before you attend your lab. Then you can spend your lab time DOING; lab time should be spent trying the exercises and self test challenges because your tutor will be there to help when you get stuck.

1 Week 1

2 Welcome to coding in R!



Figure 2.1: art by Allison Horst

This week we are going to start by getting familiar with RMarkdown and end up reading some data into RStudio. There are a total of 10 videos to watch and each video is 10-15 min long so you can dip in and out across the week.

The idea is for you to code along with the videos, stopping and starting and troubleshooting as you go. Try the exercises Danielle recommends AND to see whether you are really getting

it, have a go at the SELF TEST challenge.

For this reason, you should allocate ~ 3 hours to your coding this week.

NOTE: Your first learning log should include a knitted .Rmd document published to Rpubs. Don't know what that means? no problem! Danielle covers how to do that in the sharing your document video (video 7) below.

2.1 Resources

 [Slides](#)

 [YouTube playlist](#)

2.2 Learning outcomes

By the end of this module you will be able to...

- Find your way around RStudio and RMarkdown
- Read data into RStudio
- Use the pipe (%>%) to string together group_by and summarise commands
- Write data to a .csv file

2.3 Getting started in RMarkdown

2.3.1 The beginning

In this video, Danielle shows you how to set up an RStudio Cloud account and gives a brief intro of RMarkdown.

[Video 1](#)

2.3.2 Stating the problem

In this video, Danielle talks about why word documents can be problematic and why Markdown is a good solution if you want to write about science reproducibly. She also shows you how to create a new RStudio Cloud project.

[Video 2](#)

2.3.3 Starting markdown

In this video, Danielle shows you how to create a plain Markdown document and create some text. She shows you how to format your text with bold and italics, make lists, and add headings.

[Video 3](#)

2.3.4 More markdown

In this video, Danielle shows you how to insert hyperlinks, block quotes, and nested lists into your plain Markdown document.

[Video 4](#)

2.3.5 RMarkdown

In this video, Danielle shows you how to use RMarkdown. She shows you how to insert both text and R “chunks” and “knit” your document.

[Video 5](#)

2.3.6 Custom documents

In this video, Danielle shows you how to edit the yaml to change the kind of document that your Rmd knits to. She shows you how to knit to pdf, change the theme of your document and include a table of contents.

[Video 6](#)

2.3.7 Inserting equations & sharing your document

In this video, Danielle shows you how to use LaTeX to insert equations into your Rmd . She also shows you how to share your knitted document by publishing to RPubS.

[Video 7](#)

2.4 Say hello to data

 [Slides](#)

 [YouTube playlist](#)

 [RStudio Cloud project](#)

 [Self Test challenge](#)

Now that you have your head around RMarkdown, lets get some data into R.

2.4.1 Read and glimpse your data

In this video, Danielle introduces the “reasoning” data set that she will go on to use in upcoming videos, talks about how to read data into R using `read_csv()` and view it with `print()` and `glimpse()`

[Video 8](#)

2.4.2 What is the “pipe”?

The pipe operator is a key component of the tidyverse. Here Danielle introduces the pipe along with some examples using `group_by()` and `summarise()`.

[Video 9](#)

2.4.3 How to write data

A quick wrap up of the section, an example of using `write_csv()`, and pointers to more resources.

[Video 10](#)

3 Week 2

4 Data visualisation



Figure 4.1: art by Allison Horst

This week we are going to explore one of the most powerful functions of R, that is data visualisation. The `ggplot` package has revolutionised how we visualise data, making it really easy to plot raw data (by participant or trial) and summarised data, in a way that is reproducible.

This week, there are a total of 8 videos to watch and each video is 10-15 min long so you can dip in and out across the week.

The idea is for you to code along with the videos, stopping and starting and troubleshooting as you go. Try the exercises Danielle recommends AND to see whether you are really getting it, have a go at the SELF TEST challenge.

For this reason, you should allocate ~ 3 hours to your coding this week.

4.1 Resources

 [Slides](#)

 [RStudio Cloud project](#)

 [YouTube playlist](#)

 [Self Test challenge](#)

4.2 Learning outcomes

By the end of this module you will be able to...

- Use the ggplot package to make informative plots
- Use colour to make your plots pretty (and more informative)
- Add layers to your plot
- Use jitter, boxplot, and violins to illustrate data distributions
- Use facets to plot data by group

4.3 ggplot

4.3.1 Setting up RStudio Cloud

In this video, Danielle shows you how to set up an RStudio Cloud account.

[Video 1](#)

4.3.2 Make a scatterplot

In this video, covers basic mechanics of typing R commands, and takes the learner through the process of constructing a scatterplot.

[Video 2](#)

4.3.3 Writing a script

In this video, Danielle recreates the exact same plot as last time, but uses a script, and explains a few weird concepts like “sourcing a script” and “printing a plot object”.

[Video 3](#)

4.3.4 What are aesthetics?

In this video, Danielle explains aesthetics in the context of ggplot.

[Video 4](#)

4.3.5 Aesthetics vs parameters

In this video, Danielle talks about aesthetics, the difference between aesthetics and parameters, and introduces concepts behind plot layers.

[Video 5](#)

4.3.6 Global and local

In this video, Danielle shows you that even R experts suffer from technical failures! She also talks about global and local mappings in ggplot.

[Video 6](#)

4.3.7 Named and unnamed argument

In this video, Danielle discusses named and unnamed arguments in R.

[Video 7](#)

4.3.8 Facets and prettifying

In this video, Danielle shows you how to split plots using facets, and gives you some tips to make plots pretty.

[Video 8](#)

5 Week 3

6 Data wrangling



Figure 6.1: art by Allison Horst

R and RStudio make data cleaning fast and reproducible. The module this week is all about the dplyr package, which has lots of functions that make data wrangling fun! This week, there are a total of 8 videos to watch and each video is 10-15 min long so you can dip in and out across the week. The idea is for you to code along with the videos, stopping and starting and troubleshooting as you go. Try the exercises Danielle recommends AND to see whether you are really getting it, have a go at the SELF TEST challenge.

For this reason, you should allocate ~ 3 hours to your coding this week.

6.1 Resources

 [Slides](#)

 [RStudio Cloud project](#)

 [YouTube playlist](#)

 [Self Test challenge](#)

6.2 Learning outcomes

By the end of this module you will be able to...

- Quickly clean and `rename` variables
- Use `filter` to look subsets of your dataframe and arrange to order by a given variable
- Use `select` to make your dataframe smaller
- Use `mutate` to create new variables
- Join dataframes by row and common columns
- Convert data from wide to long and back again using pivot functions

6.3 dplyr, dance with data

6.3.1 dealing with variable names

In this video, Danielle introduces the small world of words dataset and shows you how to deal with variable names.

[Video 1](#)

6.3.2 subset using `filter()`

In this video, Danielle introduces how to subset your dataset using the `filter()` function.

[Video 2](#)

6.3.3 sort using `arrange()`

In this video, Danielle shows you how to sort your data using the `arrange()` function.

[Video 3](#)

6.3.4 make your data smaller using `select()`

In this video, Danielle explains the `select()` function for selecting columns in a data set.

[Video 4](#)

6.3.5 make new variables with `mutate()`

In this video, Danielle talks about how to make new variables using the `mutate()` function, and also a weird digression into measurement issues in psychology.

[Video 5](#)

6.3.6 join data using `bind_rows()`

In this video, Danielle provides an introduction to the `bind_rows()` function in dplyr.

[Video 6](#)

6.3.7 make wide data long with pivot functions

In this video, Danielle's unnecessarily long ramble about `pivot_longer()` and `pivot_wider()`, wherein it is obvious that the lecturer is very tired, but also manages to talk a little more about the pipe, order of operations, and other neat things.

[Video 7](#)

6.3.8 joining data with `left_join()`

In this video, Danielle gives a brief wrap up to the dplyr series, in which `left_join()` is briefly discussed. She also gets grumpy about sexism in language, and provides pointers to other resources.

[Video 8](#)

7 Week 4

8 Reproducible projects

Now that you have some RMarkdown, ggplot, and dplyr functions under your belt, let's think about how to put together a reproducible project. In this module, Danielle talks about structuring directories, naming things, file paths, templates, and README documentation. This week, there are a total of 6 videos to watch and each video is 10-15 min long so you can dip in and out across the week.

The idea is for you to code along with the videos, stopping and starting and troubleshooting as you go. Try the exercises Danielle recommends AND to see whether you are really getting it, have a go at the SELF TEST challenge.

For this reason, you should allocate ~ 3 hours to your coding this week.

8.1 Resources

 [Slides](#)

 [YouTube playlist](#)

8.2 Project Structure

8.2.1 naming things for machines

In this video, Danielle talks about how to name files in a way that the computer likes.

[Video 1](#)

8.2.2 naming things for humans

In this video, Danielle talks about how you can be strategic about naming files to make your life easier.

[Video 2](#)

8.2.3 naming things (make them sortable/searchable)

In this video, Danielle explains why you might want to think (just a little bit) more about how you name things

[Video 3](#)

8.2.4 filepaths

In this video, Danielle explains how file paths work and how you can use them to your advantage.

[Video 4](#)

8.2.5 templates

In this video, Danielle talks about how you can automate your project structure using a template and talk to your future self using a README file.

[Video 5](#)

8.2.6 where do things live?

In this video, Danielle asks the question of where your projects should live so you can find them at a later date

[Video 6](#)

9 Week 5

10 Drama-free installation

You have probably run out of RStudio Cloud hours now and it is time to work out how to install R and RStudio on your own machine. R is open source software so it is free to download and use, but sometimes the installation process can be a little bit painful. That is why we start this course with you working in the cloud and gradually transition to working on your machine once you have habituated to RStudio error messages a little bit.

In this module, Danielle walks through the process of installing R and RStudio on a PC, Mac and Ubuntu. You can pick and choose from the videos below, depending on which setup you use.

10.1 Resources

 [Slides](#)

 [YouTube playlist](#)

10.2 Installing R and RStudio ...

10.2.1 ... on a Mac

[Video 1](#)

10.2.2 ... on Windows

[Video 2](#)

10.2.3 ... on Ubuntu

[Video 3](#)

10.3 Installing and loading R packages

[Video 4](#)

10.3.1 Extra info re packages on Linux

[Video 5](#)

10.3.2 Installing R packages from Github

[Video 6](#)