

PSYC3361 internship

Jenny Richmond

2024-05-18

Table of contents

Let's learn to code in R	6
how to use this book	6
1 Week 1	7
2 Welcome to coding in R!	8
2.1 Resources	9
2.2 Learning outcomes	9
2.3 Getting started in RMarkdown	9
2.3.1 The beginning	9
2.3.2 Stating the problem	9
2.3.3 Starting markdown	10
2.3.4 More markdown	10
2.3.5 RMarkdown	10
2.3.6 Custom documents	10
2.3.7 Inserting equations & sharing your document	10
2.4 Say hello to data	11
2.4.1 Read and glimpse your data	11
2.4.2 What is the “pipe”?	11
2.4.3 How to write data	11
3 Week 2	12
4 Data visualisation	13
4.1 Resources	14
4.2 Learning outcomes	14
4.3 ggplot	14
4.3.1 Setting up RStudio Cloud	14
4.3.2 Make a scatterplot	14
4.3.3 Writing a script	14
4.3.4 What are aesthetics?	15
4.3.5 Aesthetics vs parameters	15
4.3.6 Global and local	15
4.3.7 Named and unnamed argument	15
4.3.8 Facets and prettifying	15

5 Week 3	16
6 Data wrangling	17
6.1 Resources	17
6.2 Learning outcomes	18
6.3 dplyr, dance with data	18
6.3.1 dealing with variable names	18
6.3.2 subset using filter()	18
6.3.3 sort using arrange()	18
6.3.4 make your data smaller using select()	18
6.3.5 make new variables with mutate()	19
6.3.6 join data using bind_rows()	19
6.3.7 make wide data long with pivot functions	19
6.3.8 joining data with left_join()	19
7 test prep	20
8 PSYC 3361 coding self test	21
8.0.1 Week 1 module	21
8.0.2 Week 1 self test	21
8.0.3 Week 2 module	22
8.0.4 Week 2 self test	22
8.0.5 Week 3 module	22
8.0.6 Week 3 self test	23
9 Week 4	24
10 Reproducible projects	25
10.1 Resources	25
10.2 Project Structure	25
10.2.1 naming things for machines	25
10.2.2 naming things for humans	25
10.2.3 naming things (make them sortable/searchable)	26
10.2.4 filepaths	26
10.2.5 templates	26
10.2.6 where do things live?	26
11 Week 5	27
12 Drama-free installation	28
12.1 Resources	28
12.2 Installing R and RStudio	28
12.2.1 ... on a Mac	28
12.2.2 ... on Windows	28

12.2.3 ... on Ubuntu	28
12.3 Installing and loading R packages	29
12.3.1 Extra info re packages on Linux	29
12.3.2 Installing R packages from Github	29
13 now what	30



14	31
-----------	-----------

15 Data skills	32
15.1 Websites	32
15.1.1 The R-Studio Education site	32
15.1.2 Teacup Giraffes	32

15.2 Books	32
15.2.1 R for Data Science book	32
15.3 Videos	32
15.3.1 R-Ladies Sydney #RYouWithMe	32
16 Statistics	33
16.1 Books	33
16.1.1 Learning Statistics with R	33
16.2 Videos	33
16.2.1 Statistics of DOOM	33
16.2.2 Kelly Bodwin	33
17 Practice	34
17.1 PsiChi challenge	34
17.2 TidyTuesday	34
18 documentation	35
18.1 Best practices	35
18.1.1 Use R Markdown	35
18.1.2 Follow a Clean Code Style Guide	35
18.1.3 Use Meaningful and Descriptive Names	35
18.1.4 Comment Liberally	35
18.1.5 Chunk Your Code and Explain Each Chunk	36
18.1.6 Use Inline Code Comments	36
18.1.7 Interpret Output and Visualizations	36
18.1.8 Document Data Transformation Steps	36
18.1.9 Provide an Overview and Narrative	36
18.1.10 Follow Best Practices for Reproducibility	36

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 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 test prep

8 PSYC 3361 coding self test

You can't learn to code by watching someone else do it. The best way to check whether you are really understanding the coding modules is to try out your skills on a new dataset.

Self test exercises related to the Week 1-3 coding modules are available in this [RStudio Cloud project](#)

8.0.1 Week 1 module

The RMarkdown + reading data module covers...

- RStudio and RMarkdown
- Reading data in
- %>% group_by + summarise
- Write data to csv

8.0.2 Week 1 self test

GestAge data

- Open a new Rmd file, customise
- Load packages
- Read the data frame into RStudio
- Calc mean separately by group1
- Calc min separately by group2
- Write notes with a URL link
- Insert a picture
- Write data to csv

8.0.3 Week 2 module

The ggplot module covers...

- Basic point plots*
- Colour*
- Layers*
- Jitter plot, Boxplot, Violins*
- Facets*
- Themes, scales, palettes*

8.0.4 Week 2 self test

dino data

- Reproduce the dino plot
- Make it pretty
- Challenge: get the descriptives to show that the M, variance, correlation the same

8.0.5 Week 3 module

The dplyr module covers...

- Clean names
- Rename*
- Filter*
- Arrange
- Select*
- Mutate*
- Bind_rows*
- Pivot
- Join*

*these concepts are assessed in Week 4 take home test

8.0.6 Week 3 self test

- Alone data
- Open new Rmd file, customize
- Load packages
- Read data
- Select and rename variables
- Arrange and slice
- Group_by summarise
- Point+line Plot w error bars
- Group by summarise
- Boxplot + jitter
- Mutate + casewhen
- Group_by summarise
- Filter, group_by, summarise
- Fct_relevel
- Column plot, position dodge

9 Week 4

10 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.

10.1 Resources

 [Slides](#)

 [YouTube playlist](#)

10.2 Project Structure

10.2.1 naming things for machines

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

[Video 1](#)

10.2.2 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](#)

10.2.3 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](#)

10.2.4 filepaths

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

[Video 4](#)

10.2.5 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](#)

10.2.6 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](#)

11 Week 5

12 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.

12.1 Resources

 [Slides](#)

 [YouTube playlist](#)

12.2 Installing R and RStudio ...

12.2.1 ... on a Mac

[Video 1](#)

12.2.2 ... on Windows

[Video 2](#)

12.2.3 ... on Ubuntu

[Video 3](#)

12.3 Installing and loading R packages

[Video 4](#)

12.3.1 Extra info re packages on Linux

[Video 5](#)

12.3.2 Installing R packages from Github

[Video 6](#)

13 now what

You have made your way through all of Dani's modules but probably still feel like you don't know enough. Here are some of my favourite R learning resources.



14

15 Data skills

15.1 Websites

15.1.1 The R-Studio Education site

Alison Hill and her team at RStudio have pulled together the best resources for beginners at the RStudio Education site. Cute art by Desiree de Leon make the site a joy to explore.

15.1.2 Teacup Giraffes

If you are looking for a gentle introduction to R and statistics and like a bit of cute, the statistics and teacup giraffe modules by Desiree de Leon and Hasse Walum are for you.

15.2 Books

15.2.1 R for Data Science book

If books are more your scene, the free online book by Hadley Wickham great. Its definitely worth checking out this tidyverse bible.

15.3 Videos

15.3.1 R-Ladies Sydney #RYouWithMe

The #RYouWithMe course by R-Ladies Sydney is a series of online modules that walk you through getting data into R, cleaning it up a bit, visualising it, and reporting using RMark-down. Lots of screencasts and practice exercises using enterococci data collected from Sydney beaches.

16 Statistics

16.1 Books

16.1.1 Learning Statistics with R

If you are keen to go beyond wrangling and visualising and to get some analysis done in R, Dani's book is a great place to start. It is genuinely fun to read (uncommon for a stats book) and covers basic statistical analysis. Also free and available online.

16.2 Videos

16.2.1 Statistics of DOOM

If you like to watch youtube and learn stats, rather than read and learn stats, try Dr Erin Buchanan's [Statistics of DOOM](#) youtube channel and associated [website](#). She is an A/Prof at Missouri State and teaches tons of stats courses and gives them all away.

16.2.2 Kelly Bodwin

Kelly Bodwin also teaches intro stats in R at CalStatePoly. Her youtube videos are also excellent.

17 Practice

17.1 PsiChi challenge

PsiChi, which is the US Hons psychology society, has a monthly R challenge on their OSF page. They use open data and set a few common data wrangling/visualisation/analysis tasks for you to complete. I've used the challenge in my lab meeting and have pulled some code templates into this [RStudio cloud project](#)

17.2 TidyTuesday

Each Tuesday Thomas Mock releases a dataset on the [Tidy Tuesday github](#), people all over the world plot the data to show something interesting and then post their plot to twitter using the hashtag #tidytuesday. If watching an expert is your kind of fun, check out [David Robinson's youtube channel](#). People have even timestamped his videos so you can find him demo almost anything.

18 documentation

18.1 Best practices

(produced with assistance from claude.ai)

By following these recommendations, your R code and analysis will be well-documented, easy to understand, and more likely to be reproducible by others (or even your future self). This not only aids in learning but also prepares you for collaborating on research projects and disseminating your work transparently.

18.1.1 Use R Markdown

R Markdown allows you to seamlessly integrate code, output, visualizations, and explanatory text into a single document. This makes it easy to document your code and tell a coherent story about your analysis.

18.1.2 Follow a Clean Code Style Guide

Adopt a consistent style guide for your R code, such as the tidyverse style guide (<https://style.tidyverse.org/>). This ensures that your code is readable, maintainable, and follows best practices.

18.1.3 Use Meaningful and Descriptive Names

Choose variable and function names that are clear and descriptive of their purpose. This makes your code self-documenting and easier to understand.

18.1.4 Comment Liberally

While good naming conventions help, you should still liberally comment your code. Use comments to explain the purpose of each code section, describe the logic behind your approach, and document any potential quirks or gotchas.

18.1.5 Chunk Your Code and Explain Each Chunk

In R Markdown, organize your code into logical chunks (e.g., data import, cleaning, analysis, visualization). For each chunk, provide a preceding comment that explains the purpose of that chunk and any necessary context.

18.1.6 Use Inline Code Comments

Within each code chunk, use inline (#) comments to explain complex or non-obvious operations. This helps future readers (including your future self) understand the rationale behind your code.

18.1.7 Interpret Output and Visualizations

After each code chunk that produces output or visualizations, include text that interprets the results and explains their significance in the context of your analysis.

18.1.8 Document Data Transformation Steps

When cleaning or transforming data, document each step in detail, including the rationale behind your choices and any potential implications or limitations.

18.1.9 Provide an Overview and Narrative

At the beginning of your R Markdown document, include an overview section that explains the research question, the dataset being used, and a high-level narrative of your analysis approach.

18.1.10 Follow Best Practices for Reproducibility

Document any external dependencies (e.g., packages, data sources), include session information (e.g., version numbers), and consider using techniques like relative file paths to enhance the reproducibility of your work.