Module 3: R

Work Practices

Instructor: Anjali Silva, PhD

Data Sciences Institute, University of Toronto

2022

Course Documents

- Visit: https://github.com/anjalisilva/IntroductionToR

- All course material will be available via IntroductionToR GitHub repository (https://github.com/anjalisilva/IntroductionToR). Folder structure is as follows:

  - Lessons - All files: This folder contains all files.
  - **Lessons - Data only**: This folder contains data only.
  - **Lessons - Lesson Plans only**: This folder contains lesson plans only.
  - **Lessons - PDF only**: This folder contains slide PDFs only.
  - README - README file
  - .gitignore - Files to ignore specified by instructor

Course Contacts

- Instructor: Anjali Silva Email: a.silva@utoronto.ca (Must use the subject line DSI-IntroR. E.g., DSI-IntroR: Inquiry about Lecture I.)

- TA: Tia Harrison Email: tia.harrison@mail.utoronto.ca

Overview

Getting help (Alexander (eds), 2021 Chapter 10)

Using Stack Overflow (Alexander (eds), 2021 Chapters 11-12)

Making reproducible examples (reprexes) (Alexander (eds), 2021 Chapters 13-14)

Debugging (Alexander (eds), 2021 Chapter 15)

Getting help

Errors are an inherent part of coding

Practicing and gaining familiarity with the specific tasks you have to perform will reduce the frequency with which you make errors when performing those tasks.

Still, even experienced programmers make errors frequently! Knowing how to resolve issues is an essential skill.

Avoiding errors in the first place

Use common packages and methods, especially when you are learning R

- If the functions you use are commonly recommended, there will be more troubleshooting materials and advice available.

Check in regularly

- Test small sections of code before moving on to writing more. Check that the results are what you expect.
- Restart R and rerun your code periodically.
- If you want to test your code, but it takes a significant amount of time to run, consider a toy dataset.

Isolating the issue

When you get an error, or an unexpected result from your code, you need to identify exactly which piece of code is creating the problem.

- To do this, you can run the smallest chunks possible. Typically this involves running the code line-by-line, but you may need to run parts of code within individual lines as well.

```
numbers <- seq(1, 100)
sum(numbers)/length(numbers)
```

```
## [1] 50.5
```

Starting debugging

Try, try, and try again.

- Make a change and run your code again. Repeat.
- Copy paste and edit if you want to keep track of what you are changing.

Check for typos. Did you misspell a variable name? Miss a comma somewhere?

Did you get the `could not find function` error? Check that you loaded the necessary libraries.

Look at function documentation. Are you giving the arguments to the function correctly?

```
?mean()
```

See if RStudio is indicating a syntax error. Read the error description.

Read the error message completely. Sometimes that's all the guidance you will need to fix the problem.

Using Google and Stack Overflow

Googling

Include "r" in your search terms. If you are using a specific package or function, include its name. Consider putting the name in quotes, as you require an exact match.

Copy paste error messages into the search bar and see what comes up. Often, many people have made the exact same error before you.

Check the dates of results. R packages and versions evolve quickly, so a solution from three years ago may no longer work.

Stack Overflow and Stack Exchange

If you search an error message, you may end up on stackoverflow.com.

Read the question to see how well it matches your concern.

When reading answers, note the answer scores and which answer has been accepted.

If you are not sure if a solution will work, you might as well try! Sometimes solutions will need to be edited or combined, but you will still be closer to solving your problem than before.

Making reproducible examples

What is a reprex?

Sometimes you will not be able to solve a problem yourself and you will need to ask for help. Whether you are sending the problem to a friend or posting on Stack Overflow, you can make it easier for others to help you. A reprex, or reproducible example, allows another person to recreate your error message on their own device.

Components of a reprex

1. Environment: calls for any necessary libraries and information about your R environment that might be relevant
2. Toy data set: a minimal data set that the code can be run on
3. Code: minimal and runnable code that recreates the error

Environment

Start a new R script for your reprex.

What libraries does your code use? Try not to load any extras. You can test your selection by restarting R, loading the selection, and running your error-generating code again to make sure it does not generate a `could not find function` error.

Note your R version, RStudio version, and operating system.

- Help > About RStudio will show your RStudio version.
- running `sessionInfo()` will show your R version and operating system information.

Data

You can select a subset of your real dataset and attach that with your reprex.

- From your dataset, select only the variables your code refers to.
- From that, select a chunk of rows.
- Save the dataset.

Data

You can create your own toy dataset in your code so your helpers do not have to load anything.

- What variables does your code refer to? Include those as columns.
- How many cases do you need to realistically test the code? Include that many rows.

```r
toy_data <- data.frame(ID = 1:4,
                       name = c("Abe", "Becca", "Calvin", "Danica"),
                       age = c(27, 32, 63, 55),
                       membership = c("yes", "no", "yes", "yes"))
toy_data
```

```
##   ID   name age membership
## 1  1    Abe  27        yes
## 2  2  Becca  32         no
## 3  3 Calvin  63        yes
## 4  4 Danica  55        yes
```

Code

The goal is for the code to be as stripped down as possible and still generate the exact error that is causing you trouble. Remove unnecessary lines, run the code again, and make sure that the error is still the same. Look for ways you can shorten necessary lines.

Ideally, the code should be formatted for easier reading and include comments that outline what the different lines are intended to do.

Code

The `reprex` package

After loading the `reprex` package, you can highlight your reprex code (including libraries and data), copy to your clipboard, and then run:

```
reprex()
```

The reprex will be automatically stored on your clipboard. You can paste it into messages asking others for help.

What if nothing works?

What if nothing works?

If you cannot fix the error, you can instead get rid of the error by trying a different method.

- Is there a different library or function that does the same thing?
- Could you find someone who successfully performed the task, look at their code, and use their methods (with citation)?
- Is there a workaround? Be cautious with these, as you may create more problems down the line for yourself.

Reproducibility

What is reproducibility?

- Reproducible research can be be recreated using the materials provided.
- Others should be able to completely follow and reproduce your workflow.
- This usually involves making your code, data, and environment available.
- Reproducibility is not all-or-nothing: it can be improved incrementally, as an ongoing process where each project is more reproducible.

Why make your work reproducible?

- Reproducibility increases accountability. Others can see and verify your work.
- Reproducibility adds to our collective knowledge by sharing processes instead of just results.
- Reproducibility encourages us to respect and reflect on the limitations of our data and processes.

Working directories

Your working directory is the folder path you can name files from. It will show in your "files" pane of RStudio.

```
getwd()
```

```
setwd()
```

If you set your working directory manually (by using setwd() or through the toolbar), the folder path will be unique to your computer. If you or someone else was attempting to run your code on another device, they would first need to edit the folder path to which the working directory is set. This reduces reproducibility.

RProjects

- RProjects are files with the extension .Rproj
- They designate a working directory that starts where the .Rproj file is located. You don't have to use setwd().
- You can also set up an RProject associated with a GitHub repository to share your work with the public or a team.
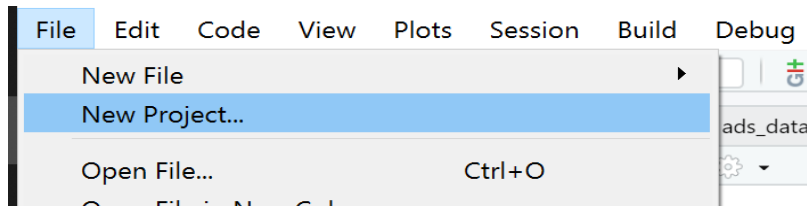
Working in RProjects

- Projects also keep all your work together: scripts, markdown files, data, results, figures, and more.
- You can create a file structure that organizes your content.

Name	Type
data	File folder
figures	File folder
papers	File folder
scripts	File folder
Project.Rproj	R Project

Coding practices

- Your process should be thoroughly documented. Each step should be explained.
- Comment your code! -- Use comments to explain what you are doing when it is not immediately apparent -- It will help you when you look at your work 6 months from now and can't remember anything -- It will help you work with collaborators -- It will increase reproducibility if you release your code

```
# you can insert a comment line like this
recorded_age <- 5 # or add a comment to a line like this
```

Coding practices

Name your variables well:

```
var <- 5 # not helpful for anyone reading your code
recorded_age <- 5 # better
```

Coding practices

Code for human readability

- Add spaces and lines. The code will work without them, but it's worse to read.

```
ads_data%>%gather(key="key",value="val")%>%mutate(is.missing=is.na(val))
%>%group_by(key,is.missing)%>%summarise(num.missing=n())
```

```
ads_data %>%
    gather(key = "key", value = "val") %>%
    mutate(is.missing = is.na(val)) %>%
    group_by(key, is.missing) %>%
    summarise(num.missing = n())
```

Discussion questions

What debugging techniques do you think will be the most helpful?

In your own work, what challenges do you anticipate in making processes reproducible? What about in the work of others? Brainstorm ways that these concerns could be addressed.

Any questions?