Work Practices in R

```
$ echo "Data Sciences Institute"
```

Resources for this lesson

- Section 3.5.1: Dealing with Errors in *Telling Stories with Data*
- Section 3.5.2: Reproducible examples in *Telling Stories with Data*

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

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.

Tips for dealing with errors

- isolate the issue by running code line-by-line or in small chunks
- try to understand the error message, or do a web search of your error message or task
    - include "in R" or "tidyverse" in the search
- look for help in the function documentation e.g. `?mean`
- restart R session
- ask for help
    - create a reproducible example (reprex)

Making reproducible examples

What is a reprex?

A **reprex**, or **reproducible example** is a minimal snippet of code that 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 code that recreates the error

The `reprex` package

An R package that helps prepare a reproducible example to share and ask for help

1. Highlight all the relevant code and copy to clipboard (Ctrl/Cmd + C)
2. Run `reprex()`
3. The code and output will be copied to your clipboard. Paste (Ctrl/Cmd + V) and share.

Demo: Creating a reprex

The following code doesn't work!

```r
library(tidyverse)
#library(palmerpenguins)

# Choose a dataset
# ?penguins
# data(penguins)
data(trees)

View(penguins)

rename(trees,Girth=tree_girth)
```

Demo: Creating a reprex

Checklist:

1. Are all the relevant libraries and data included?
2. Does the reprex produce the intended error?
3. Does the reprex include **only** the relevant code?

Questions?