



Wrapup

Andrew Redd, PhD.

R Bootcamp 2021

Bringing it Together

Reproducible Research

“... the original data (and original computer code) can be analyzed (by an independent investigator) to obtain the same results of the original study. In essence, it is the notion that the data analysis can be successfully repeated.” - Roger Peng

Why Reproducible

[\(https://www.azquotes.com/quote/1194126\)](https://www.azquotes.com/quote/1194126)

Code Design Principles

- Document **EVERYTHING**
- Only use text files
- All input should be *human* readable.
- Organize your files
 - Use folders
 - Scripts for management
 - Markdown for reports
- *Never* modify raw data

Scripts

- Scripts end with `.r` or `.R`
- May be ran with
 - `R CMD BATCH <<script.R>>`
 - `Rscript <<script.R>>`

See `?Rscript`

(<https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/Rscript>)
or:

- `callr::rscript`
(<https://www.rdocumentation.org/packages/callr/versions/3.4.3/topics/rscript>)
- `xfun::Rscript`
(<https://www.rdocumentation.org/packages/xfun/versions/0.16/topics/Rscript>)

Parallelization and CHPC

The easiest way I have found to take advantage of High Performance Computing (HPC) Clusters like the University's [CHPC \(//chpc.utah.edu\)](http://chpc.utah.edu) is:

1. parallelize inside a script leveraging either,
 - [foreach \(//cran.r-project.org/package=foreach\)](http://cran.r-project.org/package=foreach) or
 - [futures \(//cran.r-project.org/package=futures\)](http://cran.r-project.org/package=futures), then
2. spawn multiple jobs that each run a version of the script.

though this may be easier to avoid with futures.

R Markdown

Rmarkdown

R Markdown allows for mixing code & results into documents to create reproducible analyses, reports and presentations.

- <https://rmarkdown.rstudio.com/>
(<https://rmarkdown.rstudio.com/>)
- <https://bookdown.org/yihui/rmarkdown/>
(<https://bookdown.org/yihui/rmarkdown/>)



Why Markdown?

- Easy to read
- Easy to write
- Can be converted to many outputs

Output Formats

See <https://rmarkdown.rstudio.com/formats.html>
(<https://rmarkdown.rstudio.com/formats.html>)

- HTML
- PDF, via LaTeX
- Word
- Markdown (change the flavor)
- Presentations
 - ioslides, this
 - reveal.js
 - Slidy
 - Beamer (LaTeX pdf)
 - even *PowerPoint*
- Journal Article Formats
- Books ([bookdown](https://bookdown.org/yihui/bookdown/) (<https://bookdown.org/yihui/bookdown/>))
- Blogs ([blogdown](https://bookdown.org/yihui/blogdown/) (<https://bookdown.org/yihui/blogdown/>))
- Etcetera.

Writing in Markdown

Code	Output
# h1	<h1>h1</h1> or equivalent
## h2	<h2>h1</h2> or equivalent
### h3	<h3>h1</h3> or equivalent
<i>*emph*</i>	<i>emph</i>
strong	strong
~~strike~~	strike

Lists

1. one
2. two
- * three
- + four

1. one
 2. two
- three
 - four

Math

Math can be included with LaTeX using dollar signs, \$ for inline and \$\$ for block statement

`$$f(k) = \{n \text{ \choose } k\} p^{\{k\}} (1-p)^{\{n-k\}}$$`

Produces:

$$f(k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Tables: Simple

Code	**Output**

`# h1`	`<h1>h1</h1>` or equivalent
`## h2`	`<h2>h1</h2>` or equivalent
`### h3`	`<h3>h1</h3>` or equivalent
`*emph*`	*emph*
`**strong**`	**strong**
`~~strike~~`	~~strike~~

Tables: Pandoc style

Petal.Length	Petal.Width	Species
1.4	0.2	setosa
1.4	0.2	setosa
1.3	0.2	setosa
1.5	0.2	setosa
1.4	0.2	setosa
1.7	0.4	setosa

Petal.Length	Petal.Width	Species
1.40	0.20	setosa
1.40	0.20	setosa
1.30	0.20	setosa
1.50	0.20	setosa
1.40	0.20	setosa
1.70	0.40	setosa

Table: Pipe Style

Tables the easy way.

- `library(printr)` in your setup chunk

```
library(printr)  
head(iris)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

Embedding R Code

embed R code with the tick/grave symbol (above tab)

- inline

```
* Inline `r code`
```

- block code

```
```${r name, options...}  
```
```

Example:

```
```${r setup, results='hide'}  
library(knitr)
```
```

Setting options

Each chunk can have options set. Specified after the name. complete list at <http://yihui.name/knitr/options> (<http://yihui.name/knitr/options>)

- `echo=TRUE`
 - controls if the input commands are repeated.
- `results='markup'`
 - 'markup' wraps the output in an appropriate environment.
 - 'asis' don't do anything with output.
 - 'hide' don't show any output.

Message Control

- `error=TRUE`
 - print errors
- `warning=TRUE`
 - print warnings
- `message=TRUE`
 - print control messages

Very useful when loading packages.

Plots

- `dev`
 - default depends on the format.
 - useful when going from markdown to pdf.
- `fig.width` & `fig.height`

What I Hope you learned

What I Hope you learned

- Know where to find help.
- Think in R, use the building blocks you have learned.
- A desire to learn more R.

Join the Salt Lake City R Users Group

<https://www.meetup.com/slc-rug> (<https://www.meetup.com/slc-rug>)

Questions?