Reproducible research #2

R Notebooks

R Notebooks

From RStudio's article on R Notebooks:

> *"An R Notebook is an R Markdown document with chunks that can be executed independently and interactively, with output visible immediately beneath the input."*

R Notebooks are a new feature. Right now, if you want to use them, you need to update to RStudio's Preview version. You can get that here.

R Notebooks

You can render an R Notebook document to a final, static version (e.g., pdf, Word, HTML) just like an R Markdown file.

Therefore, you can use R Notebooks as an alternative to R Markdown, with the ability to try out and change chunks interactively as you write the document.

R Notebooks

You can open a new R Notebook file by going in RStudio to "File" -> "New File". In the Preview version of RStudio, there's an option there for "R Notebook".

As with R Markdown files, when you choose to create a new R Notebook file, RStudio opens a skeleton file with some example text and formatting already in the file.

The syntax is very similar to an R Markdown file, but the YAML now specifies:

```
output: html_notebook
```

Templates

Templates

R Markdown **templates** can be used to change multiple elements of the style of a rendered document. You can think of these as being the document-level analog to the themes we've used with `ggplot` objects.

To do this, some kind of style file is applied when rendering document. For HTML documents, Cascading Style Sheets (CSS) (`.css`) can be used to change the style of different elements. For pdf files, LaTeX package (style) files (`.sty`) are used.

Templates

To open a new R Markdown file that uses a template, in RStudio, go to "File" -> "New File" -> "R Markdown" -> "From Template".

Different templates come with different R packages. A couple of templates come with the `rmarkdown` package, which you likely already have.

Templates

Many of these templates will only render to pdf.

To render a pdf from R Markdown, you need to have a version of TeX installed on your computer. Like R, TeX is open source software. RStudio recommends the following installations by system:

- For Macs: MacTeX
- For PCs: MiKTeX

Links for installing both can be found at `http://www.latex-project.org/ftp.html`

Current version of TeX: 3.14159265.

Templates

The `tufte` package has templates for creating handouts typeset like Edward Tufte's books.

This package includes templates for creating both pdf and HTML documents in this style.

The package includes special functions like `newthought`, special chunk options like `fig.fullwidth`, and special knitr engines like `marginfigure`. Special features available in the tufte template include:

- Margin references
- Margin figures
- Side notes
- Full width figures

Templates

The `rticles` package has templates for several journals:

- *Journal of Statistical Software*
- *The R Journal*
- *Association for Computing Machinery*
- ACS publications (*Journal of the American Chemical Society*, *Environmental Science & Technology*)
- Elsevier publications

TEMPLATES

Some of these templates create a whole directory, with several files besides the .Rmd file. For example, the template for *The R Journal* includes:

- The R Markdown file in which you write your article
- “RJournal.sty”: A LaTeX package (style) file specific to *The R Journal*. This file tells LaTeX how to render all the elements in your article in the style desired by this journal.
- “RJreferences.bib”: A BibTeX file, where you can save citation information for all references in your article.
- “Rlogo.png”: An example figure (the R logo).

Templates

Once you render the R Markdown document from this template, you'll end up with some new files in the directory:

- "[your file name].tex": A TeX file with the content from your R Markdown file. This will be "wrapped" with some additional formatting input to create "RJwrapper.tex".
- "RJwrapper.tex": A TeX file that includes both the content from your R Markdown file and additional formatting input. Typically, you will submit this file (along with the BibTeX, any figure and image files, and possibly the style file) to the journal.
- "RJwrapper.pdf": The rendered pdf file (what the published article would look like)

Templates

This template files will often require some syntax that looks more like LaTeX than Markdown.

For example, for the template for *The R Journal*, you need to use `\citep{}` and `\citet{}` to include citations. These details will depend on the style file of the template.

As a note, you can always use raw LaTeX in R Markdown documents, not just in documents you're creating with a template. You just need to be careful not to mix the two. For example, if you use a LaTeX environment to begin an itemized list (e.g., with `begin{itemize}`), you must start each item with `item`, not `-`.

Templates

You can create your own template. You create it as part of a custom R package, and then will have access to the template once you've installed the package. This can be useful if you often write documents in a certain style, or if you ever work somewhere with certain formatting requirements for reports.

RStudio has full instructions for creating your own template: `http://rmarkdown.rstudio.com/developer_document_templates.html`

R Projects

Organization

So far, you have run much of your analysis within a single R script or R Markdown file. Often, any associated data are within the same working directory as your script or R Markdown file, but the files for one project are not separated from files for other projects.

As you move to larger projects, this kind of set-up won't work as well. Instead, you'll want to start keeping all materials for a project in a single and exclusive directory.

Organization

Often, it helps to organize the files in a project directory into subdirectories. Common subdirectories include:

- `data-raw`: Raw data and R scripts to clean the raw data.
- `data`: Cleaned data, often saved as `.RData` after being generated by a script in `data-raw`.
- `R`: Code for any functions used in analysis.
- `reports`: Any final products rendered from R Markdown and their original R Markdown files (e.g., paper drafts, reports, presentations).

R Projects

RStudio allows you to create “Projects” to organize code, data, and results within a directory. When you create a project, RStudio adds a file with the extension “.Rproj” to the directory.

Advantages of setting a directory to be an R Project are:

- Automatically uses the directory as your current working directory when you open the project.
- Coordinates well with git version control and GitHub repository system.
- Opens a “Files” window for navigating project files in an RStudio pane when you open the project.

R Projects

You can create a new project from scratch or from an existing directory.

To create an R project from a working directory, in RStudio go to “File” -> “New Project” -> “New Directory”. You can then choose where you want to save the new project directory.

GitHub

GIT

[What is git]

GIT

[Downloading git]

GIT

[Setting up git]

GitHub

[Creating GitHub account]

GitHub

[GitHub repositories]

GitHub

[local and GitHub setup]

GitHub

[working with others– collaborating vs. forks]

Syncing RStudio and GitHub

[Setting up key]

Syncing RStudio and GitHub

[Pushing local project to GitHub]

Syncing RStudio and GitHub

[Pulling a GitHub repo to local]

Syncing RStudio and GitHub

[workflow for RStudio and GitHub]

GitHub

[merge conflicts]