Module 4 – lesson 01: Examples and Demonstrations of Existing Templates

Script

In Module 4 we’re going to explore several existing R markdown templates that are available through R packages you can download and use right away. You’ve already seen a few of these packages in earlier modules. In Module 2, we explored the revealjs package to use the revealjs template for making HTML slide presentations. Then in Module 3, we explored two packages for customizing HTML documents through the prettydoc and rmdformats packages. So, you’ve already got some experience with the process of installing R packages to access the R markdown templates included with these packages.

Let’s explore a few more R packages that have helpful R markdown templates included with them. The next package we’ll take a look at is the rticles package

<https://cran.r-project.org/web/packages/rticles/>

<https://github.com/rstudio/rticles>

However, if you want to test out these templates, you will need to install LaTeX for your operating system. I will also demonstrate a few of the rticles templates here also so you are welcome to simply watch the demonstrations.

[COMPUTER DEMO]

If you would like to save your templates and files created during this lesson, go ahead and create a new Github repository called “Module4\_templateExamples”. Log into your Github account and create this new repository with a Readme file. After you have your Github repository created, create a new R project in RStudio. Start RStudio and click File/New Project/Using Version Control/Git. Then go to your new Github repository and copy the URL to the clipboard and then paste the URL into your RStudio window to create the new R project. Make sure you save it with your other R projects for this course at “C:/RepTemplates”

Now that you’ve got your new RStudio project created, let’s install the rticles package. Go to Tools/Install Packages and type in rticles – click Install. [Remember to utilize the functionality of the rticles package, you will also need to have installed LaTeX on your computer.] After the rticles package installs, you will have access to the many R markdown templates.

In RStudio, click on File/New File/R Markdown and click the Templates tab. In this tab you will see all of the R markdown templates available from the rticles package (and any other R packages you might have installed on your local drive that also have templates). Let’s pick the template for “PLOS Journal Article”. This template was developed to produce articles in the appearance and style of the PLOS journals (see <https://www.plos.org/publications>). The R markdown template shown here follows the guidelines provided by PLOS Computational Biology using their LaTeX template at <http://journals.plos.org/ploscompbiol/s/latex>

Type in a filename like “PLOSexample” for your new R markdown PLOS Journal Article and save it in your RStudio project directory for “Module4\_templateExamples”.

This will open an R Markdown template file in the editor window at the top left. It should be entitled “PLOSexample.Rmd”. Take a few minutes and scroll through the template. You’ll notice that the YAML header is more extensive than the ones you’ve seen so far in this course. The YAML header for the PLOS Journal Article template has details for multiple authors and their contact information and affiliations. There is also a place in the YAML header for the abstract. This is an example in which the document content – in this case the abstract – is actually defined inside the YAML header. There is also a section for a brief author summary as well.

As you scroll through the rest of the document, you will notice much of the same R markdown syntax for section headers and other R markdown syntax you’ve explored already in earlier lessons. There are also examples of including reference citations using BibTeX bibliographies. To learn more about citations and bibliographies, see

<http://rmarkdown.rstudio.com/authoring_bibliographies_and_citations.html>

I should also point out that when we created the R Markdown file “PLOSexample.Rmd”, we didn’t create just a single file. In fact, when we created the R markdown file based on the PLOS Article Template, a new folder/directory in your project directory called PLOSexample was created. If we open this folder, there is a file called PLOSexample.Rmd but there are several other files that were added as well. The file “PLOS-submission.eps” is the graphic for the PLOS Logo in EPS (encapsulated postscript format). The other 2 files are template files for setting up the citations for the article. The file “mybibfile.bib” is an example BibTeX file for these citations and the file “plos.csl” is the citation style definition file for the citations to be formatted for PLOS articles (CSL stands for Citation Style Language), see <https://tex.stackexchange.com/questions/69267/citation-style-language-csl>

If you have LaTeX installed, click on KNIT to plos\_article to see the resulting PDF demonstrated here. Let’s also take a quick look at the results. You’ll notice that the PDF has a nice header layout with the PLOS logo and that there are custom margins and text layout with more white space on the left than on the right. Towards the bottom of the first page there are little numbers on the right hand side – these are often used for DRAFT article submissions allowing for reviewers to provide comments and feedback at specific line numbers in the document. There is also a footer with horizontal line, another PLOS logo and page numbers.

In addition to the PLOSexample.PDF file created, the PLOS logo in EPS format was also converted to a PDF format. And another file “PLOSexample.text” was also created – this is the LaTeX file created from the R markdown file. When you clicked KNIT to plos\_article, the first step taken by Pandoc was to conver the R markdown file to a LaTeX TEX file which was then converted to a PDF file. Sometimes if there are problems or errors it often occurs in the process of converting the TEX file to PDF. So, if you have the TEX file, the errors can be corrected using a TEX editor and then the PDF can be created. So, having the TEX file is helpful.

Let’s try a couple more. In RStudio click on File/New File/R Markdown – choose template and find the rticles template for “Statistics in Medicine Article”. Type in a filename like “StatsMedExample” and click OK. This template is designed for the resulting article to be in the format used by the Statistics in Medicine journal (<http://onlinelibrary.wiley.com/journal/10.1002/(ISSN)1097-0258> )

This template also creates a new director/folder. Open this folder and review the files. The “Statistics in Medicine” template includes the R markdown file, a BibTeX example bibliography; a CLS Class file which defines formatting and style for the “Statistics in Medicine” journal; and the last file “wiley.bst” is another type of file for bibliographic style definition – this one is for wiley publications.

Let’s take a closer look at the “StatsMedExample.Rmd” R markdown file you just created in RStudio. At the top of the R markdown template you just created, you’ll notice that the YAML header for this template is similar but not exactly the same. The layout and approach for entering multiple authors and their contact and affiliation details is laid out somewhat differently that what was in the PLOS example R markdown file. In this YAML header there is also a place to include the abstract as well as the text for an acknowledgements section in the final article.

You’ll notice in the body of the document some of the R markdown syntax we’ve seen before. However, some of the syntax used here also assumes the document will be compiled using LaTeX so there is also some LaTeX syntax mixed in here as well. For example the \LaTeX syntax inserts a graphic image for the word LaTeX with unique formatting. If you have LaTeX installed, to see this graphic, go ahead and click KNIT to sim\_article. You’ll notice here in the PDF the unique formatting of the word LaTeX – this is a result of the \LaTeX syntax used in the R markdown document. Take a few minutes and review the rest of this R markdown template and compare it to the resulting PDF. You’ll notice that the resulting PDF document has even more formatting applied to the header and footers and in other sections of the document. Much of this formatting is applied and controlled using LaTeX.

There are several more R markdown templates available in the rticles R packages. I encourage you to try out more of these to get some ideas on what is possible when creating document templates using RStudio and R markdown.

Let’s try one more that does not require LaTeX. This time we’re going to install the tufte R package. In RStudio, go to Tools/Install Packages and type in tufte. The tufte package was designed to create documents in the style of Edward Tufte. The read ahead supplementary materials provided the links to some websites that explain the tufte style further.

<http://rstudio.github.io/tufte/>

<http://rmarkdown.rstudio.com/tufte_handout_format.html>

So, who is Edward Tufte? He is a professor emeritus of political science, statistics and computer science at Yale University. He is well known for his books on data visualization and information design. His books and handouts have a distinct style and format that makes extensive use of sidenotes and tightly integrated graphics and text. We can use the tufte package and the associated R markdown templates to create documents in the style of Edward Tufte.

Now that you have the tufte package installed, in RStudio, click on File/New File/R Markdown – Templates and find the “Tufte Handout (HTML or PDF)”. Click OK. Go ahead and save the R markdown file as “TufteExample.Rmd”. You’ll notice that there are several additional parameters and options defined in the tufte YAML header for the different output options. There are 3 KNIT options – HTML, Handout (which makes a PDF document), and a Book option (which uses the YAML header to create a title page for the book and each level 1 header indicated by a single hashtag # to begin a new chapter). Both the Handout and Book options which create PDF documents require LaTeX. So, let’s KNIT to tufte-html and take a look at the results.

You can see the style somewhat in the RStudio preview Viewer, but let’s open it in a full Internet browser.

When the TufteExample.html file opens, if you view in the browser open in a full window, you’ll easily see the sidenotes and plots and such in the right side margins, which is classic tufte style. The other nice option of the tufte format for HTML files is that the resulting webpage is “responsive.” This means that the style and formatting adapts as needed based on the width of the browser window. Let’s make the browser window smaller and watch what happens with the document.

For example, at the top the first footnote (1) references Github repositories. When the window is wide enough you simply see the footnote in the side margin next to the paragraph. However, if we make the browser window narrower, the sidenote goes away and the footnote number 1 now has a hyperlink. If you click on the number 1 the footnote toogles on and is viewable in the text below and then when you click it again it toggles back off. This is very helpful when viewing on mobile devices with smaller viewer screens like smart phones. A similar hyperlink is also create for the Allaire et al. (2017) reference. This time the hyperlink is connected to a circle symbol with a plus inside it. Scroll through and also notice that the graphics in the side right margins merge under the main text when you narrow the browser window.

I encourage you to explore the wide variety of formatting and layout options available using the tufte R markdown template. This R markdown document provides several different options for displaying table and figures either in the body of the text or in the right side margin or as full width of the document. The template also illustrates providing indented quotes, equations, footnotes, margin notes which are slightly different than footnotes, and citations.

Take some time to compare the results in the HTML file with those sections in the Rmarkdown document. You’ll notice that there are places where the R code chunks have specific options directing where the resulting code, table or figure will be placed in the final document. For example, in the 3rd section on Figures, the first R code chunk here has an option specified for fig.margin=TRUE to place the resulting figure in the right side margin as opposed to in the main body of the text. Compare this to the figure under the “Full Width Figure” section that has an R code chunk option fig.fullwidth=TRUE. And the plot in the “Main Column Figures” section simply does not specify a figure placement option so the figure is placed within the main body of the document. However, it is interesting to notice that for this figure in the “Main Column Figures” section, the caption defined by the fig.cap= option shows up in the right side margin but the figure is in the main body of the document.

Now let’s go ahead and back everything up to your Github account.

Open Git Bash and make sure you are in the correct directory:

C:\RepTemplates\Module4\_templateExamples

Once in that directory, type in the following 4 Git commands to check the status of your local files compared to your Github cloud repository; add or stage the modified files; commit your changes; and then push the changes to your Github cloud repository.

git status

git add .

git commit –m “add template example documents”

git push

Now go to your Github repository, refresh if necessary and see your newly committed files.