Module 2 – lesson 03: Document Elements

Script

For this lesson we’re going to continue working with the same R markdown document “module2\_rmd1.Rmd”. So, If you need to, go ahead and log in to your Github account, open RStudio, and open the RStudio project for “Module2\_rmd1”.

So far you’ve explored the YAML header and tried out several R markdown syntax markings to change the formatting of text in your document. Next we’re going to see how to insert other objects and elements within your document. In this lesson you’ll learn how to insert:

* + - Figures; images; photos; or pictures
    - Tables
    - Videos or animations
    - Equations
    - and footnotes

Figures

Take a look at the bottom of your current R markdown file. The last section of this current document contains a chunk of R code that makes a plot of the pressure dataset which is built into the base R software.

Let’s take a quick look at this built-in dataset. Go to the Console windows (bottom left) and type in

data1 <- pressure

which creates an object called “data1” that is assigned a copy of the built-in pressure dataset. Then click on the Environment TAB in the upper right window – click on the little table icon on the right – this opens the dataset in a viewer window at the top left. As you can see there are 2 columns of data and 19 rows for 19 data points of temperature and pressure. You can get more detailed information on this built-in dataset by looking it up in the HELP window. This brings up a help page on the pressure dataset which says that the pressure datasets is “Data on the relation between temperature in degrees Celsius and vapor pressure of mercury in millimeters (of mercury).”

So, in the R chunk shown here, there is only 1 line of code between backticks ```{r} ending with 3 more backticks ```

This line of code makes a scatterplot of the pressure (along the Y vertical axis) against temperature (along the X horizontal axis).

plot(pressure)

By running this R code chunk, a scatterplot is created and then inserted in the document where this code was specified.

By the way, we can control the size of the figure using code “chunk options”. You can learn more about code chunk options at Yihui Xie’s website for the knitr package at <https://yihui.name/knitr/options/>

Let’s change the figure width and height using the fig.width and fig.height chunk options. These options come after the r in the {r} curly brackets. The current R code chunk has the following

```{r pressure, echo=FALSE}

Let’s explore each piece.

The curly brackets indicate that this is a code chunk and the r indicates that the code will be R code.

The word pressure is a label of the code chunk. This is helpful for debugging later. When you get an error the R Markdown log will show in which code chunk the error occurred. Giving them descriptive names will help you keep track.

Let’s look at the R Markdown TAB at the bottom left and review the log of when we last rendered or compiled the document – when we last ran KNIT.

After the chunk label and a comma, there is already 1 chunk option saying echo=FALSE. This tells R markdown to hide this code chunk and not “echo” it or show it in the final document. So, if you look back at the documents you’ve made so far – all you see is the plot of the pressure dataset – you do not see the R code.

At the very end of the document, let’s add another code chunk and alter this next figure slightly. Click the button in the editor window with a little green C with a + plus, click the down arrow for insert an R code chunk. When you click this, it automatically enters the 3 backticks with the r in curly brackets ```{r} followed by a blank line and then 3 more backticks ``` ending the code chunk.

Go to the blank line and type in

plot(pressure)

Now go back to the 1st line, let’s add the following options. We’ll add a new label “pressure2” since each code chunk has to have a unique chunk name. We’ll also add fig.width=5 and fig.height=5 which does 2 things – it will make the horizontal and vertical dimensions of the figure to be the same and should render a figure approximately 5 inches by 5 inches – see the chunk options described on Yihui Xie’s website.

```{r pressure2, fig.width=5, fig.height=5}

Save the RMD file and KNIT to HTML as see the differences between the 2 plots. Not only will the plot sizes be different, in the 2nd one we left out the echo=FALSE so the R code was shown right before the 2nd plot. If you want to hide this code, go back and add echo=FALSE to the chunk options.

```{r pressure2, fig.width=5, fig.height=5, echo=FALSE}

In RStudio 1.1.x, you’ll also notice that in the editor window, within the code chunk at the far right there are a couple of little icons. Click on the one that looks like a gear and another window pops up that shows various code chunk options available including

* The name of the code chunk
* What kind of output you want to show (with or without the code – setting echo-TRUE or FALSE)
* You can also turn on or off warnings and messages
* At the bottom you can also set the figure width and height – try changing this to 4 inches and watch the changes in your R markdown file.

There is also a green arrow > that you can click which will “run” the R code as if you were running R from the command line or from a standalone R script. If you click this, the plot(pressure) command will be executed and the plot will be generated in the Plots window.

Images

We do not have to use R code to add figures in your document. You can also bring in external images and pictures. Let’s use the picture called sunstar.png – the read ahead materials had instructions on how to download this picture. For now, let’s put this picture in the same directory as your R markdown document. If you want to see this in your document, you can use the simple R markdown syntax ![alt text](filename)

Let’s create a new header in our document

## Insert Images

Here is an image inserted

![sunstar](sunstar.png)

You can also insert images off the web by linking directly to them via their web-address URL

Here is the R logo

![Rlogo](https://www.r-project.org/logo/Rlogo.svg)

NOTE!! The image with the web-address URL will NOT compile correctly for the KNIT to WORD and KNIT to PDF since the image is not stored locally. If you want this image for a DOC or PDF formatted file, you will need to download the image and store it locally when you “KNIT” the final document. Then use the same syntax as you just did for sunstar.png. So, delete this section before you KNIT to WORD or KNIT to PDF.

Tables

Doing tables in with R markdown or rather using Pandoc’s markdown can be done but is a tedious process for even simple tables. There are numerous examples provided at <http://rmarkdown.rstudio.com/authoring_pandoc_markdown.html#tables>

Typically, it is easier to use R code to generate a table. The best function for making tables using R markdown is the kable function from the knitr package, see <https://yihui.name/knitr/> It may also help to install and learn more about the printr package also which improves the formatting of knitr output, see <https://yihui.name/printr/>

Let’s try a simple table of another built-in dataset, cars. Look at the help pages for cars. The cars dataset has 50 observations or rows and 2 columns with the 1st column having data on the speed of the cars and the 2nd column having data on the stopping distance. We can use the head function from base R to look at the top 6 rows of the cars dataset. The cars dataset is in an R object called a “data.frame” which the kable function handles like a table. So the following R chunk will make a table of the top 6 rows of the cars dataset.

Let’s create another new header in our document

## Insert Tables

```{r}

knitr::kable(head(cars))

```

Let me explain the R code a little bit. Since the kable function comes from the knitr R package, it is good practice to list the package followed by 2 colons followed by the function so it is easy for you or anyone else reading your R code to know which package the function came from. There are literally tens of thousands of R packages and some have the same function name which do entirely different things. So, to avoid confusion it is always a good idea to list use the package::function() when using a function in R.

So this code says to extract the “head” or top 6 rows of the cars dataset and use the kable function from the knitr package to print out a table in the final document.

Go ahead and save your document and click KNIT to HTML to see the result. Feel free to also try KNIT to WORD and KNIT to PDF to see how the table appears in each of those formats.

We can add other options to the kable function – like adding a table caption. Let’s add the option caption = "Top 6 Rows of Cars Dataset"

knitr::kable(head(cars),

caption = "Top 6 Rows of Cars Dataset")

You’ll notice this added a caption or title at the top of the table in each of the formats (HTML, DOC and PDF).

Videos and animations

Suppose you have a video you want to include in your document. This doesn’t make sense for a printed final document, but these will work for a HTML document, you can view an embedded video. Let’s work again with the sunstar graphic as an animated GIF and as a MP4 video. The read ahead materials had instructions on how to download both the animated GIF “sunstar.gif” and video “sunstar.mp4” into a subfolder in your project called “sunstar”. Since these are in a directory called “sunstar” in your project folder, you need to include the folder name when you list the filename. The syntax for embedding videos and animated GIFs is similar to how images are inserted.

## Insert an Animated GIF and Video

![sunstar](sunstar/sunstar.gif)

![sunstar](sunstar/sunstar.mp4)

KNIT to HTML to see the results. You may need to open the saved html file in a separate browser window to see the embedded MP4 video. These videos will not work if you KNIT to WORD – the document will compile but these sections will be blank. If you try running KNIT to PDF you will get an error. So, Videos and Animated GIFs only work in HTML format – at least using this approach. New functions and methods are created daily, so there is probably a way to embed videos and animated GIFs in other formats if you look for it.

In the final HTML document, you’ll notice that the animated GIF plays over and over again. But the MP4 file is embedded with a video viewer that you can click play to see the video. This is only a simple introduction to videos. This is only scratching the surface. There is a R package called vebmedr which allows embedding of videos like YouTube in your HTML documents. Learn more at <http://ijlyttle.github.io/vembedr/>

Equations

For those of you not interested in typing math equations in your documents, you can skip this section. But if you are interested, know that you can embed equations using LaTeX syntax. For example, suppose we wanted to write out the equation for a simple linear regression model. We would embed the LaTeX formatting inside a block beginning and ending with 2 dollar signs $$

## Insert an equation

$$ Y = \beta\_0 + \beta\_1x $$

There are hundreds of websites with information, tutorials and help online for formatting equations using LaTeX. One example is <https://www.sharelatex.com/learn/Mathematical_expressions> if you want to learn more.

Footnotes

Finally, you might want to include a footnote in your document. The syntax for inserting a footnote is square brackets with an up arrow ^ inserted. You can add footnotes in one of two ways.

First – you can simply add the notation where you want to add a footnote in the place in the text where you want the index number. And then at the END of the document you have to provide the content you want displayed with the footnote index.

Second, you can use what is called an inline note. With an inline note you don’t have to remember to add the footnote references at the end. Let’s add the following to your document using both methods.

## Insert text with some footnotes

Here is a footnote reference,[^1] and another.[^longnote]

Here is an inline note.^[Inlines notes are easier to write, since you don't have to pick an identifier and move down to type the note.]

[^1]: Here is the footnote.

[^longnote]: Here's one with multiple blocks.

Save your document and KNIT to HTML to view your document.

Finally, let’s make sure to back everything up and save your changes to your Github repository.

Open Git Bash and change to the directory for your Github repository created for “Module2\_rmd1” – so go to:

C:\RepTemplates\Module2\_rmd1

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 “inserting multiple elements to my RMD file”

git push