Module 2 – lesson 04: Presentation Formats

Script

For this lesson we’re going to create some presentation formats. We’ll make slides using either ioslides or slidy formats which make HTML based slides. If you have LaTeX installed you can also try making PDF slides in Beamer format. We’ll also check out a newer slide template format called revealjs which can be accessed by installing a new R package.

For now, let’s keep using the same Github repository and the same RStudio project for “Module2\_rmd1”. Go ahead and open this project in RStudio. We will create your new R markdown files for the different presentation formats in this repository. This repository should be saved on your local drive at

C:\RepTemplates\Module2\_rmd1

In RStudio, click File/Open Project and find the .Rproj file and open it.

Once you have your project open in RStudio, go ahead and create a new R Markdown file. But this time instead of creating a new document, select Presentation and the ioslides format. Type in a title like “Module 2 – ioslides” and click OK to create the new ioslides document.

Take a quick look at the YAML header. The keywords title, author and date are all very similar to the YAML header for the HTML document. But notice that the output has changed to ioslides\_presentation. This is the line that tells R markdown that this document will be rendered as an ioslides HTML formatted presentation.

You’ll notice that this template for the HTML ioslides format has many similar elements and formatting as the basic R markdown HTML document we created in an earlier lesson.

However, this time in ioslides the 2 hashtags ## for level 2 headers are the titles for each new slide. So, this basic ioslides template contains 4 slides. The 1st slide has some basic text with 2 paragraphs. The 2nd slide has a list with 3 bullets. The 3rd slide shows a simple summary of the cars dataset. The 4th slide shows a slide containing the plot of the pressure dataset.

Go ahead and KNIT to HTML (ioslides) to see the final slides. Save the file to your local drive with a filename like “module2\_ioslides.Rmd”

You’ll notice that there are actually 5 total slides. The 1st slide is your title slide and was generated using the information contained in the YAML header – the other 4 slides created using the level 2 headers with the 2 hashtags ## come after this title slide. In your YAML header, try changing your title, author or date and KNIT to HTML (ioslides) the slides again to see how it changes your title slide.

Open the saved HTML slides in a browser so you can view these outside of the RStudio viewer. A nice feature of the ioslides format is the ability to view the slides in a normal width/height ratio or in a wide format. While viewing the slides in a browser, type the letter w to switch from a normal to a wide view and back again.

You can also view the slides in a full screen mode by typing the letter f to toggle from full screen to browser window view.

Let’s also try making these slides in the Slidy HTML format as well. We can do this one of two ways. We can create a new R Markdown file and choose Presentations and the Slidy format. Or since we have the ioslides formatted file open already, we can simply KNIT to HTML (Slidy) and the output format will update automatically in the YAML header and your Slidy formatted slides will be created also. BEFORE we do this, go ahead and save the file as another filename to keep them separate from the ioslides format. Click File/SaveAs and save as “Module2\_slidy.Rmd”.

After you save your file as “Module2\_slidy”, click KNIT to HTML (Slidy) and see these slides in the different format for Slidy HTML slides.

Take a few moments now and choose one of these 2 R Markdown files – either for ioslides or Slidy – and spend some time adding a few new slides inserting an image or video or an equation and view the results. You can use the text and R Markdown syntax we created earlier for the R markdown HTML document. Everything should work as expected in HTML except for the footnotes which are not supported in ioslides or Slidy but do work in the Beamer PDF format. You always have to check the documentation for each format as some options are not supported across formats – like animations or videos not working in PDF or DOC formats.

It is a good idea to read through the documentation for each R Markdown format at <http://rmarkdown.rstudio.com/formats.html>

For now, let’s try adding 3 slides to our ioslides (or Slidy) presentation here based on what we did earlier in the R Markdown HTML document. ***[NOTE TO SELF – demonstrate this on the video in both ioslides and Slidy format for the viewers]***

## A slide with an inserted image

Here is an image inserted

![sunstar](sunstar.png)

## A slide with a table

```{r}

knitr::kable(head(cars),

caption = "Top 6 Rows of Cars Dataset")

```

## A slide with an equation

A simple linear regression equation

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

Notice that the slide with the sunstar image shows the word sunstar below the image. This is because we included AltText in the first set of text between the square brackets []. To remove this word, we simply need to remove the AltText. Change this slide to be

## A slide with an inserted image

Here is an image inserted

![](sunstar.png)

KNIT to HTML (ioslides) or (Slidy) whichever you prefer and view the changes. Save your file.

Suppose we want to center the image on this slide. To the header (title) for this slide we can add the following layout option as follows.

## A slide with an inserted image {.flexbox .vcenter}

You can learn more about these options for the ioslides format at <http://rmarkdown.rstudio.com/ioslides_presentation_format.html>

You can also make 2 columns using an ioslide option {.columns-2} in the slide header. Try adding this slide to your ioslides R markdown file.

## A slide with 2 columns an image and a bulleted list {.columns-2}

![](sunstar.png)

- bullet 1

- bullet 2

- bullet 3

KNIT to HTML (ioslides) to see the results. NOTE: This 2-column option shown here will NOT work for Slidy NOR for the Beamer PDF formats.

Optionally, if you have LaTeX installed, save your file as “Module2\_beamer” and try KNIT to PDF (Beamer) to see the Beamer PDF slide format. Again some options and syntax may work differently in the Beamer format. Read through the Beamer documentation to learn more about what is possible. <http://rmarkdown.rstudio.com/beamer_presentation_format.html>

For the most part you can switch between the different slide formats without too much trouble but you’ll notice that the 2-columns option only worked for the ioslides format.

Another HTML presentation format gaining popularity is the revealjs format. To use this format, we first have to install the R package for this slide presentation format.

See the details at <http://rmarkdown.rstudio.com/revealjs_presentation_format.html>

Also learn more at

<http://lab.hakim.se/reveal-js/#/> and

<https://github.com/hakimel/reveal.js/>

In RStudio, click on Tools/Install Package, make sure the CRAN repository is selected and type in revealjs. Alternatively, in the Console window you can type in the R command install.packages with the name of the package revealjs put in quotes between the parentheses

install.packages("revealjs")

Once the revealjs package is installed, we can now access a revealjs slide template by going to File/New File/R Markdown file – choose From Templates and look for “Reveal.js Presentation (HTML)”.

By the way, any other R packages you have installed that have R markdown templates available will be listed here in the Templates window. Later in this course you will learn about more R packages and templates that are available and you will learn how to create an R package with your own template!

For now, let’s go ahead and create a new R markdown file using the revealjs presentation template. This will create a set of slides similar to those we just did using ioslides or slidy.

You will notice that when the new R markdown file was created, this time the YAML header is slightly different – only the title and output were defined. Let’s go ahead and add back author and date to your YAML header.

Click KNIT to revealjs\_presentation to see the resulting HTML slides. These slides are produced with a default slide transition animation.

Let’s try changing the default slide transition to “zoom” (see list at <http://rmarkdown.rstudio.com/revealjs_presentation_format.html#slide_transitions>)

To change a parameter in the YAML header we will move the revealjs::revealjs\_presentation after output: to a new line and indent it 2 spaces. Then we add a colon: and on a 3rd new line also indented another 2 spaces (4 space total) we add the transition option we want as follows.

output:

revealjs::revealjs\_presentation:

transition: zoom

Save your file and KNIT to revealjs\_presentation to see the transition changes.

The revealjs format also utilizes themes to apply some styling to the slides using color. See the example provided at <http://rmarkdown.rstudio.com/revealjs_presentation_format.html#appearance_and_style>

Let’s add a few more lines to your YAML header and give your slides some color. We’ll set the theme to solarized, the highlight to kate and the center option to true to give us this YAML header

output:

revealjs::revealjs\_presentation:

transition: zoom

theme: solarized

highlight: kate

center: true

All of the keywords after revealjs::revealjs\_presentation: are specific parameters or options available in the revealjs\_presentation format from the revealjs package.

Save your file again. KNIT to revealjs\_presentation to see your updated slides with a new color and format.

Take a few moments and try different options and settings by modifying the YAML header to experiment with making changes that affect your final presentation. Refer back to the revealjs format to see what options are available, <http://rmarkdown.rstudio.com/revealjs_presentation_format.html>

Finally, be 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” – 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 “created multiple slide presentation formats”

git push