Support

Premium support

Sign in

RStudio Support > Documentation > R Presentations

Search

Authoring R Presentations

Josh Paulson
February 11, 2014 20:14

Authoring R Presentations

NOTE: R Presentations require RStudio v0.98 or later versions.

Overview

R Presentations are a feature of RStudio that enable easy authoring of HTML5 presentations. R Presentations are based on R Markdown, and include the following features:

- Very straightforward authoring syntax (Markdown)
- Easy incorporation of R code and it's output (including plots)
- Support for LaTeX equations using MathJax
- Flexible two column layouts
- Many options for slide transitions and slide navigation
- Ability to customize the appearance of slides using CSS
- Extensive support for authoring and previewing presentations within the RStudio IDE
- Can be played back either inside the RStudio IDE or as standalone HTML5 presentations in a web browser.
- Can be easily published as either a standalone HTML file or to RPubs

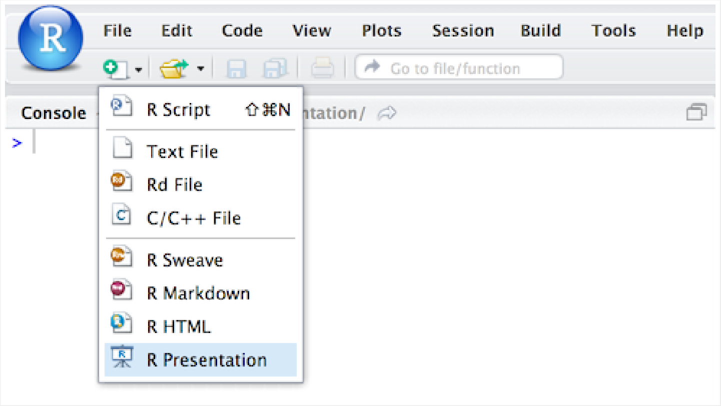
The goal of R Presentations is to make authoring slides that make use of R code and LaTeX equations as straightforward as possible. Several options for customizing the appearance of presentations are provided, however tools like PowerPoint and Keynote will always be superior when fine grained control over apperance or sophsiticated visual effects are of primary importance.

Getting Started

To create a new R Presentation you execute the **New File -> R Presentation** command:

Related articles

- Slide Transitions and Navigation
- Displaying and Distributing Presentations
- Customizing Fonts and Appearance
- Using R Markdown
- Getting Started



After specifying the location to save the presentation, a new presentation will be created and a preview will show within the Presentation tab in the upper right corner of the IDE.

Slide Basics

Slides are composed using markdown and delimited by section headings that use `=============`.

The very first slide in a deck is the title slide. It will automatically be displayed with a larger heading (H1) and a special background color. It can also include special `author` and `date` fields. For example:

```
Title Slide
====================================
author: John Doe
date: February 7th, 2013
```

To create a slide with bullets you simply use standard Markdown bullets:

```
Slide 1
====================================
- Bullet 1
- Bullet 2
- Bullet 3
```

Slides automatically display their titles unless `title: false` is specified. For example:

```
Slide 1
====================================
title: false
```

Editing Presentations

There are a number of tools within the RStudio IDE designed to make editing presentations more productive:

1. Every time you save your presentation the preview is refreshed and navigated to whatever slide you were editing.
2. You can use code folding within the source code editor to expand and collapse slides.
3. You can use the navigation menu at the bottom of the source code editor to quickly switch between slides.
4. If you are looking at a slide within the preview pane you can press the **Edit** button on the toolbar to jump immediately to it's location in the source file.

Authoring Content

Formatted Text

Content within R Presentations is formatted using standard Markdown syntax (the bullets example above is one example of this syntax).

If you aren't familiar with Markdown, the following resources might be helpful:

- Using Markdown with RStudio
- The **Markdown Quick Reference** available from the help menu on the source editor toolbar

R Code Chunks

Since they are R Markdown documents, R Presentations can include R code chunks. Code chunks can be used as a means of displaying code for illustration or to actually render output into slides. Here is a simple R code chunk that will result in both the code and it's output being included in the slide:

~~~
Slide With Code
====================================
```{r}
summary(cars)
```
~~~

Displaying Code Only

To display R code without evaluating it, you specify the `eval=FALSE` option:

~~~
Code Only Slide
====================================
```{r, eval=FALSE}
summary(cars)
```
~~~

Displaying Output Only

To display the output of a code chunk but not the underlying R code, you specify the `echo=FALSE` option:

~~~
Output Only Slide
====================================
```{r, echo=FALSE}
summary(cars)
```
~~~

Note that R code chunks can also be used to include plots within Presentations. To display a plot while omitting the code used to generate the plot you'd do this:

~~~
Plot Slide
====================================
```{r, echo=FALSE}
plot(cars)
```
~~~

Caching

Every time a presentation is saved it's preview is automatically updated by re-running knitr for the presentation. If doing this becomes time consuming due to long computations or plots that are expensive to render you can use knitr caching to improve performance. The documentation on knitr chunk and package options describe how caching works and the cache examples provide additional details.

If you want to enable caching globally for your presentation you can include a code chunk like this at the top of the document:

~~~
```{r setup, include=FALSE}
opts_chunk$set(cache=TRUE)
```
~~~

If you run into problems with cached output you can always use the **Clear Knitr Cache** command on the **More** menu to rebuild your presentation without previously cached output.

Images

You can embed images within slides using standard markdown syntax. For example:

```
![alt text](myimage.png)
```

If you want to float an image to the left or right, simply include it within a column (two-column layouts are discussed in more detail below). For example:

```
Slide With Image Left
====================================
![alt text](myimage.png)
***
This text will appear to the right
```

There are a few special rules regarding how images within presentations are rendered:

1. When contained within a column, images fill all available space within the column.
2. When the only content on a slide is an image, then the image will fill all available space on the slide
3. Standalone images (images that appear in their own paragraph) that appear along with text are limited to 50% of vertical space.

Note that these rules also govern how plot images generated from within R code chunks are laid out.

Equations with MathJax

You can embed LaTeX or MathML equations in R Presentations using the following syntax:

- `$equation$` for inline equations (note there must not be whitespace adjacent to the $ delimiters)
- `$$ equation $$` for display equations
- `<math ...> </math>` for MathML equations.

Here are examples of the syntax used for inline and display equations respectively:

```

The Arithmetic mean is equal to $\frac{1}{n} \sum_{i=i}^{n} x_{i}$, or the summation of n numbers divided by n.

```

```

$$
  \begin{aligned}
  \dot{x} & = \sigma(y-x) \\
  \dot{y} & = \rho x - y - xz \\
  \dot{z} & = -\beta z + xy
  \end{aligned}
$$

```

The article on Equations in R Markdown includes more detailed information on embedded equations.

Two Column Layouts

You can create a two-column slide layout by using a markdown horizontal rule (`***`).

For example:

```
Two-Column Slide
====================================
First column
***
Second column
```

By default the columns are split equally across the slide. If you want to specify that one column gets proportionally more space you can use the `left` or `right` field. For example

```
Two-Column Slide
====================================
left: 70%
```

Learning More

These articles include documentation on more advanced features and customization options of R presentations, as well as details on displaying and distributing presentations:

- Slide Transitions and Navigation
- Customizing Fonts and Appearance
- Displaying and Distributing Presentations

Was this article helpful?
1 out of 1 found this helpful

Have more questions? Submit a request

Comments

Article is closed for comments.