

My Presentation

You R. Name

v. 2021-10-03 02:55

R Markdown Basics

- Lists

- Line breaks

- R chunks

- Inline code

Mathematical equations

- A quick example of *some* of the package's shortcuts

- Statistical notation

- Calculus notation

- Matrices

Additional resources

References

R Markdown Basics

Here is a brief introduction into using *R Markdown*. *Markdown* is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. *R Markdown* provides the flexibility of *Markdown* with the implementation of **R** input and output. For more details on using *R Markdown* see <https://rmarkdown.rstudio.com>.

Be careful with your spacing in *Markdown* documents. While whitespace largely is ignored, it does at times give *Markdown* signals as to how to proceed. As a habit, try to keep everything left aligned whenever possible, especially as you type a new paragraph. In other words, there is no need to indent basic text in the Rmd document (in fact, it might cause your text to do funny things if you do).

It's easy to create a list. It can be unordered like

- Item 1
- Item 2

or it can be ordered like

1. Item 1
2. Item 2

Notice that I intentionally mislabeled Item 2 as number 4.

Markdown automatically figures this out! You can put any numbers in the list and it will create the list. Check it out below.

To create a sublist, just indent the values a bit (at least four spaces or a tab). (Here's one case where indentation is key!)

1. Item 1
2. Item 2
3. Item 3
 - Item 3a
 - Item 3b

Make sure to add white space between lines if you'd like to start a new paragraph. Look at what happens below in the outputted document if you don't:

Here is the first sentence. Here is another sentence. Here is the last sentence to end the paragraph.

This should be a new paragraph.

Now for the correct way:

Here is the first sentence. Here is another sentence. Here is the last sentence to end the paragraph.

This should be a new paragraph.

When you click the **Knit** button above a document will be generated that includes both content as well as the output of any embedded **R** code chunks within the document.

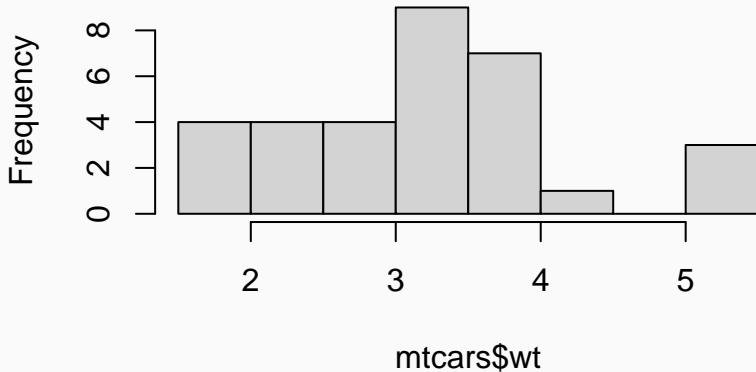
You can embed an **R** code chunk like this (mtcars is a built-in **R** dataset):

```
summary(mtcars)
```

##	mpg	cyl	disp	hp
##	Min. :10.40	Min. :4.000	Min. : 71.1	Min. :
##	1st Qu.:15.43	1st Qu.:4.000	1st Qu.:120.8	1st Qu.:
##	Median :19.20	Median :6.000	Median :196.3	Median :
##	Mean :20.09	Mean :6.188	Mean :230.7	Mean :
##	3rd Qu.:22.80	3rd Qu.:8.000	3rd Qu.:326.0	3rd Qu.:
##	Max. :33.90	Max. :8.000	Max. :472.0	Max. :
##	drat	wt	qsec	vs
##	Min. :2.760	Min. :1.513	Min. :14.50	Min. :
##	1st Qu.:3.080	1st Qu.:2.581	1st Qu.:16.89	1st Qu.:
##	Median :3.695	Median :3.325	Median :17.71	Median :
##	Mean :3.597	Mean :3.217	Mean :17.85	Mean ₁₁ :
##	3rd Qu.:3.920	3rd Qu.:3.610	3rd Qu.:18.90	3rd Qu.:

```
hist(mtcars$wt)
```

Histogram of mtcars\$wt



If you'd like to put the results of your analysis directly into your discussion, add inline code like this:

The cos of 2π is 1.

Another example would be the direct calculation of the standard deviation:

The standard deviation of speed in cars is 5.2876444.

One last neat feature is the use of the `ifelse` conditional statement which can be used to output text depending on the result of an **R** calculation:

The standard deviation is less than 6.

Note the use of `>` here, which signifies a quotation environment that will be indented.

As you see with 2π above, mathematics can be added by surrounding the mathematical text with dollar signs. More examples of this are in Mathematical equations.

Mathematical equations

$\text{T}_{\text{E}}\text{X}$ is the best way to typeset mathematics. Donald Knuth designed $\text{T}_{\text{E}}\text{X}$ when he got frustrated at how long it was taking the typesetters to finish his book, which contained a lot of mathematics.

One nice feature of *R Markdown* is its ability to read LaTeX code directly.

A quick example of *some* of the package's shortcuts

Let K be a field of **scalars**—usually either the real numbers \mathbb{R} or the complex numbers \mathbb{C} , or occasionally the rationals \mathbb{Q} . A **vector space** over K is a set V of **vectors** equipped with two operations, vector addition $(x, y) \mapsto x + y$, and scalar multiplication $(\alpha, x) \mapsto \alpha x$, where $x, y \in V$ and $\alpha \in K$.

The operations satisfy:

V.1 $x + y = y + x$

V.2 $(x + y) + z = x + (y + z)$

V.3 There is a vector 0 , satisfying $x + 0 = x$ for every vector x .

V.4 $x + (-1)x = 0$

V.5 $\alpha(\beta x) = (\alpha\beta)x$

V.6 $1x = x$

V.7 $\alpha(x + y) = (\alpha x) + (\alpha y)$

V.8 $(\alpha + \beta)x = (\alpha x) + (\beta x)$

- Algebra and semi-algebra: \mathcal{A}, S
- Sigma field: \mathcal{F}
- Set of probability measures: $\mathcal{P}(), \mathcal{P}(X), \mathcal{P}(A)$, etc
- Probability of x : $\mathbb{P}(X = x)$
- Different thetas: $\hat{\theta}, \tilde{\theta}$
- Convergence in probability: $\hat{\theta} \xrightarrow{p} \theta$
- Union and intersection: $\bigcup_{i=1}^{\infty}, \bigcap_{i=1}^{\infty}, \bigcup_a^b, \bigcap_a^b$
- Normal distributions: $\mathcal{N}(0, 1), \mathcal{N}(\mu, \sigma^2)$
- Measurable and probability space: $(\Omega, \mathcal{F}), (\Omega, \mathcal{F}, \mathbb{P})$

Calculus notation

Many shortcuts for derivatives. Let $f : A \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, then

$$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial x_i}, \frac{\partial f}{\partial x_j}$$

$$\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial x_i}, \frac{\partial F}{\partial x_j}$$

Same for $g()$ which is $g : A \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$

$$\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}, \frac{\partial g}{\partial x_i}, \frac{\partial g}{\partial x_j}$$

Now let f be a C^2 function

$$\frac{\partial^2 f}{\partial y \partial x}, \frac{\partial^2 f}{\partial x \partial y}, \frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial y^2}, \frac{\partial^2 f}{\partial x_j \partial x_i}, \frac{\partial^2 f}{\partial x_i \partial x_j}, \dots, \frac{\partial f_i}{\partial x_j}$$

You can also have $f, g : A \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ and $f, g : A \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$.

In general, you can write

$$\frac{\partial f}{\partial x}$$

or more general

$$\frac{\partial L}{\partial \beta} \text{ or } \frac{\partial L(\hat{\beta}_\lambda)}{\partial \beta}$$

which is more flexible because it allows

$$\frac{\partial L^2(\hat{\beta}_\lambda)}{\partial \beta_3}$$

Matrices

You can also type matrices with relative efficiency. Here are some simple examples of use, and of course you can use \LaTeX default commands.

Start by replacing `\left[\begin{array}{...}` by `\begin{matrix}`. Note that there is no column specifier. You can make as many columns as you like, but they will all be centered. To finish, instead of `\end{array}\right]`, just type `\end{matrix}`. Like this:

$$\begin{matrix} \text{indices} & 1 & 2 & 3 & 4 \\ 1 & \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} \end{bmatrix} \\ 2 & \begin{bmatrix} M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} \end{bmatrix} \end{matrix}$$

You can also add a vertical rule and if you don't like all the horizontal space around the vertical rule, you can get rid of it using plain TEX's `\omit` command.

$$\begin{array}{cc}
 \text{indices} & \begin{array}{cc} 1 & 2 \end{array} \left| \begin{array}{cc} 3 & 4 \end{array} \right. \\
 \begin{array}{c} 1 \\ 2 \end{array} & \left[\begin{array}{cc|cc} M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} \\ M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} \end{array} \right]
 \end{array}$$

If you think the vertical rule in the first row is too tall, you can shorten it using plain TEX's `height` command.

$$\begin{array}{cc}
 \text{indices} & \begin{array}{cc} 1 & 2 \end{array} \left| \begin{array}{cc} 3 & 4 \end{array} \right. \\
 \begin{array}{c} 1 \\ 2 \end{array} & \left[\begin{array}{cc|cc} M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} \\ M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} \end{array} \right]
 \end{array}$$

Horizontal lines works as you might expect.

indices	1	2	3	4
1	$M_{1,1}$	$M_{1,2}$	$M_{1,3}$	$M_{1,4}$
2	$M_{2,1}$	$M_{2,2}$	$M_{2,3}$	$M_{2,4}$

Notice how the `\hline` cuts across all the columns, but it doesn't connect to the closing bracket. I am not sure I like this behavior, but you can use `\cline`.

indices	1	2	3	4
1	$M_{1,1}$	$M_{1,2}$	$M_{1,3}$	$M_{1,4}$
2	$M_{2,1}$	$M_{2,2}$	$M_{2,3}$	$M_{2,4}$

indices	1	2	3	4
1	$M_{1,1}$	$M_{1,2}$	$M_{1,3}$	$M_{1,4}$
2	$M_{2,1}$	$M_{2,2}$	$M_{2,3}$	$M_{2,4}$

Here's a more complex example that summarises the previous steps.

indices	(1·11)	(1·12)	(1·22)	(2·11)	(2·12)	(2·22)
(1)	$\lambda(1)^2$	$2\lambda(1)\lambda(2)$	$\lambda(2)^2$	0	0	0
3(2)	0	0	0	$\lambda(1)^2$	$2\lambda(1)\lambda(2)$	$\lambda(2)^2$
(111)	3	0	0	0	0	0
(112)	0	2	0	1	0	0
(122)	0	0	1	0	2	0
(222)	0	0	0	0	0	3

where `\text{}` is defined in the `amstext` \LaTeX package.

Changing delimiters

The `varsitybluesmatrix` package defines four style parameters that can be used to change the appearance of the array. The first two are `\vblldelim` and `\vbrdelim`, the left and right delimiters. By default they are `[` and `]` but you can change them like this:

$$\begin{array}{cc} \text{indices} & \begin{array}{cc} 1 & 2 \end{array} & \begin{array}{cc} 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \left\{ \begin{array}{cc} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{array} \right. & \left| \begin{array}{cc} M_{1,3} & M_{1,4} \\ M_{2,3} & M_{2,4} \end{array} \right. \end{array}$$

Another option:

indices	1	2	3	4
1	$M_{1,1}$	$M_{1,2}$	$M_{1,3}$	$M_{1,4}$
2	$M_{2,1}$	$M_{2,2}$	$M_{2,3}$	$M_{2,4}$

Yet another option:

$$\begin{array}{cc} \text{indices} & \begin{array}{cc|cc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \left(\begin{array}{cc|cc} M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} \\ M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} \end{array} \right) \end{array}$$

Changing the border row and column style

You can change the style of the border row and column entries by redefining `\vbrowsestyle` and `\vbcolstyle`, which are by default set to `\scriptstyle`. You could, for instance, say `\renewcommand{\vbrowsestyle}{\relax}` to typeset the first row as usual. By the way, the upper left corner is governed by the column style. You can always use an `\mbox{}` to change the style of any particular entry.

Changing spacing

Besides changing the delimiters, you can also change the space inserted after the first row and after the first column. These are governed by the lengths `\vbrowsep` and `\vbrowsep`. By default they are 0pt and `.5\arraycolsep`, respectively.

indices	1	2		3	4
1	$M_{1,1}$	$M_{1,2}$		$M_{1,3}$	$M_{1,4}$
2	$M_{2,1}$	$M_{2,2}$		$M_{2,3}$	$M_{2,4}$

Additional resources

Additional resources

- *Markdown* Cheatsheet
- *R Markdown* Reference Guide
- *R Markdown* Cheatsheet
- *RStudio IDE* Cheatsheet
- *RStudio IDE* Official website
- Introduction to `dplyr`
- `ggplot2` Documentation
- `ggplot2` Cheatsheet

References

Angel, Edward. *Batch-File Computer Graphics : A Bottom-up Approach with QuickTime*. Boston, MA: Wesley Addison Longman, 2001.

———. *Interactive Computer Graphics : A Top-down Approach with OpenGL*. Boston, MA: Addison Wesley Longman, 2000.

———. *Test Second Book by Angel*. Boston, MA: Wesley Addison Longman, 2001.