

More R Markdown

Andrew Muehleisen

More R Markdown!

At this point, you're familiar with the basics of Markdown formatting. But – there's much more you can do!

You can add:

- **Inline code**
- **Images**
- **Tables**
- **Equations**
- **HTML tags**
- **Interactive features**
- **Bibliographies**

More Features: inline code

R code chunks aren't the only way to include R code in our document. We can also use 'inline' code with a single backtick

```
## Inline Code
```

```
We can embed formatted code into our text with a single  
backtick. For example, `mean(iris$sepal.length, na.rm =  
TRUE)`.
```

```
If we want to run the code, we just include `r` in the  
inline statement. This is helpful if you want to insert  
a value in your document that always comes directly from  
the data: `r mean(iris$Sepal.Length, na.rm = TRUE)`.
```



Inline Code

We can embed formatted code into our text with a single backtick. For example, `mean(iris$sepal.length, na.rm = TRUE)`.

If we want to run the code, we just include `r` in the inline statement. This is helpful if you want to insert a value in your document that always comes directly from the data: 5.8433333.

More Features: images

There are a couple ways of embedding images into our document. There's the easy Markdown way:

`## Including images`

We can include images via the following format.
Note that this is the same as creating a hyperlink,
but with an "!" in front. |

`![LTER Logo] (https://lternet.edu/wp-content/uploads/2018/02/LTER-network-horizontal.png)`



Including images

We can include images via the following format. Note that this is the same as creating a hyperlink, but with an "!" in front.



LTER Logo

NATIONAL SCIENCE FOUNDATION

LTER NETWORK

LONG TERM ECOLOGICAL RESEARCH

More Features: images

We can also embed images with R code chunks, which gives us more control over size and orientation:

```
## Embedding an image using R chunks

We can scale the image size with `out.width` or
`out.height`, and manage its location with
`fig.align`.

```{r, LTER logo chunk, out.width='50%',
fig.align="center", fig.cap="LTER logo",
echo=FALSE}
knitr::include_graphics("../images/LTER-logo.png")
```
```

Embedding an image using R chunks

We can scale the image size with `out.width` or `out.height`, and manage its location with `fig.align`.



More Features: **tables**

Creating a formatted table from a Data Frame is as easy as a single function:

```
## Creating a table from a Data Frame

We can easily create a formatted table from a
data frame using `kable`.

```{r, echo = FALSE, warning = FALSE}
iris_avg <- aggregate(iris, list(iris$Species),
mean)[-6]

knitr::kable(iris_avg, caption = "Iris means")
```
```

Creating a table from a Data Frame

We can easily create a formatted table from a data frame using `kable`.

Iris means

| Group.1 | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width |
|------------|--------------|-------------|--------------|-------------|
| setosa | 5.006 | 3.428 | 1.462 | 0.246 |
| versicolor | 5.936 | 2.770 | 4.260 | 1.326 |
| virginica | 6.588 | 2.974 | 5.552 | 2.026 |

More Features: interactive tables

We might have a table that's normally too large to be useful.

```
## Interactive tables

This table is way too large for our document!

```{r, echo = FALSE}
knitr::kable(iris, caption = "Iris dataset")
```
```

Interactive tables

This table is way too large for our document!

Iris means



| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |

More Features: interactive tables

We can use external packages to create an interactive table instead.

Interactive tables

This table is way too large for our document!

Instead, lets use the `datatable` function from the "DT" package.

```
## Interactive tables

This table is way too large for our document!

Instead, lets use the `datatable` function from the "DT"
package.

```{r interactive table, echo = FALSE}
library(DT)
DT::datatable(iris)
```
```



Show entries

Search:

| | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---|--------------|-------------|--------------|-------------|---------|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 2 | 4.9 | 3 | 1.4 | 0.2 | setosa |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5 | 5 | 3.6 | 1.4 | 0.2 | setosa |
| 6 | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 7 | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 8 | 5 | 3.4 | 1.5 | 0.2 | setosa |

More Features: **equations**

Using a pair of dollar signs '\$\$', we can create equations. Note that they follow LaTeX syntax.

```
## LaTeX equations
```

```
We bracket a LaTeX expression with '$$' to  
create a formatted equation.
```

```
$$\lambda = 1 - m + m\frac{E}{C}$$
```

$$\lambda = 1 - m + m\frac{E}{C}$$



LaTeX equations

We bracket a LaTeX expression with `$$` to create a formatted equation.

$$\lambda = 1 - m + m\frac{E}{C}$$

More Features: HTML tags

When rendering an HTML document, we can use HTML tags to customize elements. For example, you can change your font size.

```
## HTML tags

Here is some regular Markdown text

<font size = "1"> Here is some text made smaller
with HTML tags </font>

<font size = "20"> Or much larger text!</font>
```



Here is some regular Markdown text

Here is some text made smaller with HTML tags

Or much larger text!

The Header

The YAML header configures the output of your document. We've been using it for HTML output. However, you can also create:

- **PDF documents**
- **Word documents**
- **Slide shows**
- **Bibliographies**
- **And more...**

```
---  
title: "Problem Set 2"  
output:  
  html_document: default  
  html_notebook: default  
  pdf_document: default  
  word_document: default  
---
```

Each type of output has various configuration options

The Header: **table of contents**

HTML outputs can have interactive features. For example, a table of contents.

```
---  
title: "Problem Set 2"  
output:  
  html_document:  
    toc: true  
    toc_float: true  
---
```



| Name: Andrew Muehleisen |
|---------------------------|
| I. CO2 Trends |
| II. Temperature trends |
| III. Ice sheets |
| IV. Sea level |
| V. Longer term CO2 trends |

Problem Set 2

Name: Andrew Muehleisen

About this format: This problem set is saved as an [R Markdown Notebook](#). When you execute code within the notebook, the results appear beneath the code. When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file). To add a new R chunk click the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

To submit: Please rename this file **LASTNAME_ProblemSet2** and upload both the **.Rmd** and the final **.html** file to the assignments folder on Canvas

To submit: Please rename this file **LASTNAME_ProblemSet2** and upload both the **.Rmd** and the final **.html** file to the assignments folder on Canvas

The Header: shiny apps

You can use “R Shiny” to create interactive figures or applications.

```
---
title: "shiny_demo"
output: html_document
runtime: shiny
---

```{r, echo = FALSE}
sliderInput("bins", "Number of Bins", min = 1, max = 50,
value = 25)

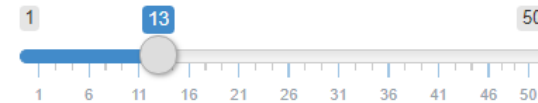
renderPlot({
 x <- iris[, "Sepal.Length"]
 bins <- seq(min(x), max(x), length.out = input$bins +
1)

 hist(x, breaks = bins, col = "black", border = "white")
})
```
```

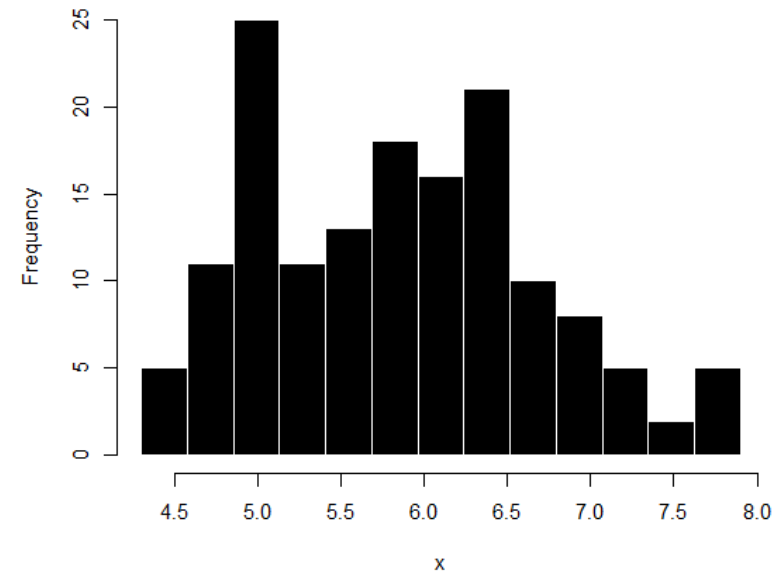


shiny_demo

Number of Bins



Histogram of x



The Header: **bibliography**

If you have a bibliography formatted with BibTeX, you can automatically include it in your document from the header.

```
---  
title: "Problem Set 2"  
output:  
  html_document: default  
  html_notebook: default  
  pdf_document: default  
  word_document: default  
bibliography: biblio.bib  
---
```



References

Cleland, E.E., Collins, S.L., Dickson, T.L., Farrer, E.C., Gross, K.L., Gherardi, L.A., Hallett, L.M., Hobbs, R.J., Hsu, J.S., Turnbull, L. & Suding, K.N. (2013). Sensitivity of grassland plant community composition to spatial vs. Temporal variation in precipitation. *Ecology*, **94**, 1687–1696.

Collins, S.L., Micheli, F. & Hartt, L. (2000). A method to determine rates and patterns of variability in ecological communities. *Oikos*, **91**, 285–293.

Collins, S.L., Suding, K.N., Cleland, E.E., Batty, M., Pennings, S.C., Gross, K.L., Grace, J.B., Gough, L., Fargione, J.E. & Clark, C.M. (2008). Rank clocks and plant community dynamics. *Ecology*, **89**, 3534–3541.

Grman, E., Lau, J.A., Schoolmaster, D.R. & Gross, K.L. (2010). Mechanisms contributing to stability in ecosystem function depend on the environmental context. *Ecology letters*, **13**, 1400–10.