

# Exploratory Data Analysis with R

## Introduction to Rmarkdown

Xuemao Zhang  
East Stroudsburg University

August 31, 2022

# What's covered in this lecture?

- What is R Markdown?
- Header
- Markdown text
- Code chunks
- Some LaTeX Basics

# What is R Markdown?

- Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
- An R Markdown document is written in markdown (an easy-to-write plain text format) and contains **chunks of embedded R code**.
- An R Markdown file has name extension `.Rmd`.
- When you click the **Knit** button, a document will be generated that includes both content as well as the output of any embedded **R code chunks** within the document.
- Installation: First, install R and the RStudio IDE; Then in the R console, type `install.packages('rmarkdown')`.

# What is R Markdown?

- Detailed information about RMarkdown can be found in the book [R Markdown: The Definitive Guide](#) by Yihui Xie.
- An R Markdown file generally contains three things.
  - ▶ A header at the top of the document.
  - ▶ Markdown text.
  - ▶ Code chunks.
    - ★ Code chunks are used to render R (and code from other programming languages!) output into a document.

# Header

- To create an R Markdown file,
  - ▶ Create a plain text file and save it with the extension `.Rmd`.
  - ▶ Or you can click File > New File > R Markdown... in the RStudio toolbar.
- There are two types of output formats in the rmarkdown package: **documents**, and **presentations**.
- You can specify the output format in the YAML (originally meant Yet Another Markup Language now stands for YAML Ain't Markup Language) **header** at the top of the document.

# Header

- The following is a header of a Markdown file. The header is enclosed by two sets of three dashes ---. This block allows you to fine-tune the output of your document.

```
---  
title: "Writing documents with R Markdown"  
author: John  
date: "09/09/2022"  
output: html_document  
---
```

- Create an Rmarkdown file with this header only and click Knit, you can see how the output looks like.

# Header

- The following is a list of some common output formats
  - ▶ beamer\_presentation
  - ▶ powerpoint\_presentation
  - ▶ html\_document
  - ▶ pdf\_document
- You can see the full list of YAML header options for a HTML document in the book [R Markdown: The Definitive Guide](#) by Yihui Xie.

# Header

- You can add a table of contents (TOC) using the `toc` option and specify the depth of headers that it applies to using the `toc_depth` option. For example:

```
---  
title: "Writing documents with R Markdown"  
author: John  
date: "09/09/2022"  
output:  
  html_document:  
    toc: true  
    toc_depth: 2  
---
```



# Header

- You can specify the `toc_float` option to float the table of contents to the left of the main document content. The floating table of contents will always be visible even when the document is scrolled.

```
---  
title: "Writing documents with R Markdown"  
author: John  
date: "09/09/2022"  
output:  
  html_document:  
    toc: true  
    toc_float: true  
    toc_depth: 3  
---
```

# Markdown text

- **Headers:** Place one or more hashtags at the start of a line that will be a header (or sub-header). For example,
  - ▶ # Say Hello to markdown. A single hashtag creates a first level header.
  - ▶ Two hashtags, ##, creates a second level header, and so on.
- **Italicized and bold text:**
  - ▶ Surround italicized text with asterisks, like this *\*italicized text\**.
  - ▶ Surround bold text with two asterisks, like this **\*\*bold text\*\***.
- **Lists:** Group lines into bullet points that begin with asterisks, dashes - or plus signs +. Leave a blank line before the first bullet, like this

This is a list

```
* item 1
* item 2
* item 3
```

# Markdown text

- **Hyperlinks:** Surround links with brackets, and then provide the link target in parentheses, like this [Github] (<https://github.com/>).
- **Plain code blocks:** Plain code blocks are used to show R code without running it. They can be written after three or more backticks, and ended with three or more backticks.

```
```
```

```
install.packages('ggplot2');
```

```
library(ggplot2);
```

```
help(ggplot);
```

```
```
```

# Markdown text

- The following is an example of R Markdown file

```
---  
title: "Writing documents with R Markdown"  
author: John  
date: "09/09/2022"  
output:  
  html_document:  
    toc: true  
    toc_float: true  
    toc_depth: 3  
---
```

# Header 1

This is an R Markdown document.

## Header 2

Use an asterisk mark to provide emphasis,  
such as *italics* and **bold**.

# Markdown text

Create lists with a dash

- Item 1
  - item 1.1
  - item 1.2 [Github](https://github.com/).
- Item 2
- Item 3

...

Use back ticks to create a block of code

...

# Markdown text

## Formatting:

- Italic. *\*italic\**.

This is *italic*.

- Bold. **\*\*bold\*\***.

This is **bold**.

- Superscripts.  $y^2$ .

This is  $y^2$ .

# Code chunks

- The `knitr` package extends the basic markdown syntax to include chunks of executable code. When you render the report, knitr will run the code and add the results to the output file.
- Code chunks are used to render R (and code from other programming languages!) output into a document.

A code chunk delimiter looks like:

```
```{r}
```

```
```
```

- All code falls between the triple back tick marks, e.g:

```
```{r}
```

```
sin(3.1416/2);
```

```
```
```

# Code chunks

- To omit the results from your final report (and not run the code) add the argument `eval = FALSE` inside the brackets and after `r`. This will place a copy of your code into the report.

```
```{r eval = FALSE}
```

```
# An example without running the code
```

```
sin(3.1416/2);
```

```
```
```



# Code chunks

- To omit the code from the final report (while including the results) add the argument `echo = FALSE`. This will place a copy of the results into your report without showing the code.

```
```{r echo = FALSE}  
  
# The dimensions of iris data are  
  
dim(iris);  
  
```
```

- For more other code chunk options, see section 2.6 R code chunks and inline R code of the book [R Markdown: The Definitive Guide](#) by Yihui Xie.

# Code chunks

## Inline code:

- To embed R code in a line of text, surround the code with a pair of backticks and the letter `r`, like this.

The dimensions of iris data are `r dim(iris)` . (sorry, I could not type the back ticks)

- `knitr` will replace the inline code with its result in your final document (inline code is always replaced by its result). The result will appear as if it were part of the original text.

# Code chunks

- Add the following code trunks to the previous R Markdown file, knit and see the results.

```
```{r}

sin(3.1416/2);

...

```{r eval = FALSE}

# An example without running the code

sin(3.1416/2);

...

```{r echo = FALSE}

# The dimensions of iris data are

dim(iris);

...

```

The dimensions of iris data are `dim(iris)` . (sorry, I could not type the back ticks)

# Code chunks

- We can also create plots.
- By default, figures produced by R code will be placed immediately after the code chunk they were generated from. For example

```
```{r fig.align="center", out.width = '60%', echo=TRUE}  
library(ggplot2);  
qplot(data = mpg, displ, cty, geom = "point");  
```
```

# Code chunks

- We can use figure options to customise the output of the plot, e.g:
  - ▶ `fig.align='center'` to set the alignment to the middle of the document
  - ▶ `fig.height=8` to set the height of the figure
  - ▶ `fig.width=8` to set the width of the figure
  - ▶ `fig.cap="Fig 1."` to add a caption describing the plot
- Again, for more information read the book [R Markdown: The Definitive Guide](#) by Yihui Xie.

# Code chunks

- Add the following code to the RMarkdown file and knit

```
```{r fig.align="center", out.width = '60%', echo=TRUE}  
library(ggplot2);  
qplot(data = mpg, displ, cty, geom = "point");  
```
```

# Some LaTeX Basics

Latex equations can be written in a Markdown file as well. If you want to generate PDF output, you will need to use PDF LaTeX. For R Markdown users who have not installed LaTeX before, installing TinyTeX (<https://yihui.name/tinytex/>) is recommended.

```
install.packages('tinytex')
tinytex::install_tinytex()
```

## Subscripts and Superscripts

To indicate a subscript, use the underscore `_` character. To indicate a superscript, use a single caret character `^`. **Note:** this can be confusing, because the R Markdown language delimits superscripts with two carets. In LaTeX equations, a single caret indicates the superscript.

If the subscript or superscript has just one character, there is no need to delimit with braces. However, if there is more than one character, braces must be used.

The following examples illustrate:

```
$$X_i$$
```

```
$$X_{i}$$
```

$$X_i$$
$$X_i$$

# Some LaTeX Basics

## Subscripts and Superscripts

Notice that in the above case, braces were not actually needed.

In this next example, however, failure to use braces creates an error, as LaTeX sets only the first character as a subscript

```
$$X_{i,j}$$
```

```
$$X_i,j$$
```

$$X_{i,j}$$

$$X_i,j$$

Here is an expression that uses both subscripts and superscripts

```
$$X^2_{i,j}$$
```

$$X^2_{i,j}$$



# Some LaTeX Basics

## Square Roots

We indicate a square root using the `\sqrt` operator.

`$$\sqrt{b^2 - 4ac}$$`

$$\sqrt{b^2 - 4ac}$$

## Fractions

Displayed fractions are typeset using the `\frac` operator.

`$$\frac{4z^3}{16}$$`

$$\frac{4z^3}{16}$$

# Some LaTeX Basics

## Summation Expressions

These are indicated with the `\sum` operator, followed by a subscript for the material appearing below the summation sign, and a superscript for any material appearing above the summation sign.

Here is an example.

`$$\sum_{i=1}^n X^3_i$$`

$$\sum_{i=1}^n X_i^3$$

# Some LaTeX Basics

## Self-Sizing Parentheses

In LaTeX, you can create parentheses, brackets, and braces which size themselves automatically to contain large expressions. You do this using the `\left` and `\right` operators. Here is an example

`$$\sum_{i=1}^n\left(\frac{X_i}{Y_i}\right)$$`

$$\sum_{i=1}^n \left( \frac{X_i}{Y_i} \right)$$

# Some LaTeX Basics

## Greek Letters

Many statistical expressions use Greek letters. Much of the Greek alphabet is implemented in LaTeX, as indicated in the LaTeX cheat sheet available at the course website. There are both upper and lower case versions available for some letters.

`$$\alpha, \beta, \gamma, \Gamma$$`

$\alpha, \beta, \gamma, \Gamma$

# Some LaTeX Basics

## Special Symbols

All common mathematical symbols are implemented, and you can find a listing on the LaTeX cheat sheet. Some examples. (Notice that, in the third example, I use the tilde character for a forced space. Generally LaTeX does spacing for you automatically, and unless you use the tilde character, R will ignore your attempts to add spaces.)

`$$a \pm b$$`

`$$x \geq 15$$`

`$$a_i \geq 0 \quad \forall i$$`

$$a \pm b$$

$$x \geq 15$$

$$a_i \geq 0 \quad \forall i$$

# Some LaTeX Basics

## Special Functions

LaTeX typesets special functions in a different font from mathematical variables. These functions, such as  $\sin$ ,  $\cos$ , etc. are indicated in LaTeX with a backslash. Here is an example that also illustrates how to typeset an integral.

```
$$\int_0^{2\pi} \sin x dx$$
```

$$\int_0^{2\pi} \sin x \, dx$$

# Some LaTeX Basics

## Matrices

Matrices are presented in the `array` environment. One begins with the statement `\begin{array}` and ends with the statement `\end{array}`. Following the opening statement, a format code is used to indicate the formatting of each column. In the example below, we use the code `{rrr}` to indicate that each column is right justified.

Each row is then entered, with cells separated by the `&` symbol, and each line (except the last) terminated by `\\`.

```
$$\begin{array}
```

```
{rrr}
```

```
1 & 2 & 3 \\\
```

```
4 & 5 & 6 \\\
```

```
7 & 8 & 9
```

```
\end{array}
```

```
$$
```

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

# Some LaTeX Basics

## Matrices

In math textbooks, matrices are often surrounded by brackets, and are assigned to a boldface letter. Here is an example

```
$$\mathbf{X} = \left[\begin{array}{rrr}1 & 2 & 3 \\4 & 5 & 6 \\7 & 8 & 9\end{array}\right]$$
```

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$



# Some LaTeX Basics

## Inline vs. Display Material

As can be seen in the above, equations can be formatted *inline* or as *displayed formulas*. In the latter case, they are centered and set off from the main text. In the former case, the mathematical material occurs smoothly in the line of text.

In order to fit neatly in a line, summation expressions (and similar constructs) are formatted slightly differently in their inline and display versions.

Inline mathematical material is set off by the use of single dollar-sign characters. Consequently, if you wish to use a dollar sign (for example, to indicate currency), you need to preface it with a back-slash. The following examples, followed by their typeset versions, should make this clear.

This summation expression  `$\sum_{i=1}^n X_i$`  appears inline.

This summation expression  $\sum_{i=1}^n X_i$  appears inline.

`$$\sum_{i=1}^n X_i$$`

This summation expression is in display form.

$$\sum_{i=1}^n X_i$$

# License



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).