# Introduction to R Markdown

Xuemao Zhang
Department of Mathematics
East Stroudsburg University

June 27, 2019

# 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?

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,
  - ▶ Creat 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: "7/18/2019"
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: "7/18/2019"
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: "7/18/2019"
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](www.github.com).

- **Plain code blocks**: Plain code blocks are used to show R code without runing 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: "7/18/2019"
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 *itlatics* and **bold**.
```

# Markdown text

```
Create lists with a dash

- Item 1

    - item 1.1

    - item 1.2 [Github](www.github.com).

- Item 2

- Item 3

```
Use back ticks to create a block of code
```
```

# 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 backtrick 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 whthout 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) .

- ▶ 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 r dim(iris) .

# 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");
````

Questions?

?