Professor Bear - Intro to R Markdown

Nik Bear Brown

Lesson 1: R Markdown

In the first lesson, we start the lesson discussing R Markdown to create written reports with embedded working code. R Markdown is an authoring format that enables easy creation of dynamic documents, presentations, and reports from R. It combines the core syntax of markdown (an easy-to-write plain text format) with embedded R code chunks that are run so their output can be included in the final document. R Markdown documents can be regenerated whenever underlying R code or data changes.

This is an R Markdown document. 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.

In this lesson we will learn about R Markdown syntax. The following lessons and assingments all will use R Markdown.

R Markdown is an authoring format that enables easy creation of dynamic documents, presentations, and reports from R. It combines the core syntax of markdown (an easy-to-write plain text format) with embedded R code chunks that are run so their output can be included in the final document. R Markdown documents are fully reproducible (they can be automatically regenerated whenever underlying R code or data changes).

The RMarkdown website (http://rmarkdown.rstudio.com/) describes R Markdown v2, a next generation implementation of R Markdown based on knitr and pandoc. This implementation brings many enhancements to R Markdown, including:

Many available output formats including HTML, PDF, and MS Word. Support for creating Beamer, ioslides, and Slidy presentations. New markdown syntax including expanded support for tables and bibliographies.

- Hooks for customizing HTML and PDF output (include CSS, headers, and footers).
- Include raw LaTeX within markdown for advanced customization of PDF output.
- Compile HTML, PDF, or MS Word notebooks from R scripts. Extensiblity: create custom templates and even entirely new output formats.
- Create interactive R Markdown documents using Shiny.

Note that PDF output (including Beamer slides) requires a full installation of TeX.

See [RMarkdown] (http://rmarkdown.rstudio.com/)

Installing R, RStudio, & RMarkdown

- Install R
  - R for Mac (https://cran.r-project.org/bin/macosx/)
  - R for Windows (https://cran.r-project.org/bin/windows/base/)
  - R for Linux (https://cran.r-project.org/bin/linux/)

- Install the lastest version of RStudio
- Install the latest version of the `rmarkdown` package: `install.packages("rmarkdown")`

Additional packages needed

To run the code in this lesson you may need additional packages.

- If necessary install `ggplot2` and `lattice` packages:

`install.packages("ggplot2");`install.packages("lattice");

Opening R Markdown document

Open the .Rmd document in R studio and Click on "Knit HTML" or crtl-shift-K

R Markdown Cheat Sheets

R Markdown Cheat Sheet (PDF), a quick guide to the most commonly used markdown syntax, knitr options, and output formats. R Markdown Cheat Sheet (PDF)

R Markdown Reference Guide (PDF), a more comprehensive reference guide to markdown, knitr, and output format options. R Markdown Reference Guide (PDF)

Creating a new R Markdown document

To create a new R Markdown document:

- Open R Studio, and go to File - New - R Markdown
- Paste in the contents of this gist (which contains the R Markdown file used to produce this post) and save the file with an `.rmd` extension
- Click Knit HTML

Additional R Markdown document options

Click on gear to the right of "Knit HTML" to see the options.

Options such as table of contents, using a CSS style sheet, etc.

Convert Markdown files to Word, PDF

To switch between the output of HTML, PDF, and MS Word. jlick on gear to the right of "Knit HTML" to see the options.

Loading R libraries

```
set.seed(333)
library(ggplot2)
library(lattice)
```

Looading data

```
data(diamonds)
```

R Code Chunks (embedding R)

Within an R Markdown file, R Code Chunks can be embedded using the native Markdown syntax for fenced code regions. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

```
summary(diamonds)
```

```
##      carat               cut        color        clarity     
##  Min.   :0.2000   Fair      : 1610   D: 6775   SI1    :13065  
##  1st Qu.:0.4000   Good      : 4906   E: 9797   VS2    :12258  
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194  
##  Mean   :0.7979   Premium   :13791   G:11292   VS1    : 8171  
##  3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2   : 5066  
##  Max.   :5.0100                     I: 5422   VVS1   : 3655  
##                                     J: 2808   (Other): 2531  
##      depth           table           price             x         
##  Min.   :43.00   Min.   :43.00   Min.   :  326   Min.   : 0.000  
##  1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710  
##  Median :61.80   Median :57.00   Median : 2401   Median : 5.700  
##  Mean   :61.75   Mean   :57.46   Mean   : 3933   Mean   : 5.731  
##  3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540  
##  Max.   :79.00   Max.   :95.00   Max.   :18823   Max.   :10.740  
##                                                                  
##        y                z        
##  Min.   : 0.000   Min.   : 0.000  
##  1st Qu.: 4.720   1st Qu.: 2.910  
##  Median : 5.710   Median : 3.530
```

```
##  Mean   : 5.735   Mean   : 3.539
##  3rd Qu.: 6.540   3rd Qu.: 4.040
##  Max.   :58.900   Max.   :31.800
##
```

```
str(diamonds)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    53940 obs. of  10
variables:
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23
...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3
1 3 ...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6
5 2 5 ...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6
7 3 4 5 ...
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4
...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05
...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39
...
```

```
names(diamonds)
```

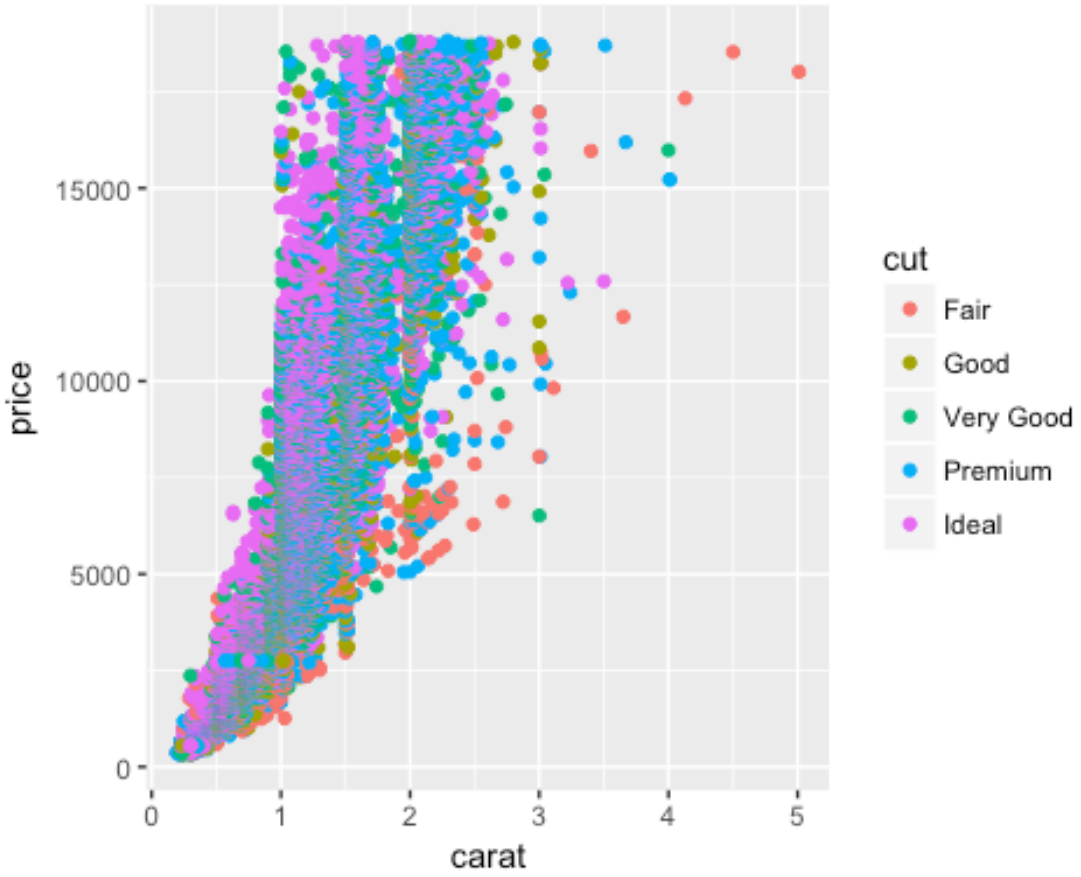
```
##  [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"
"price"
##  [8] "x"       "y"       "z"
```

```
head(diamonds)
```

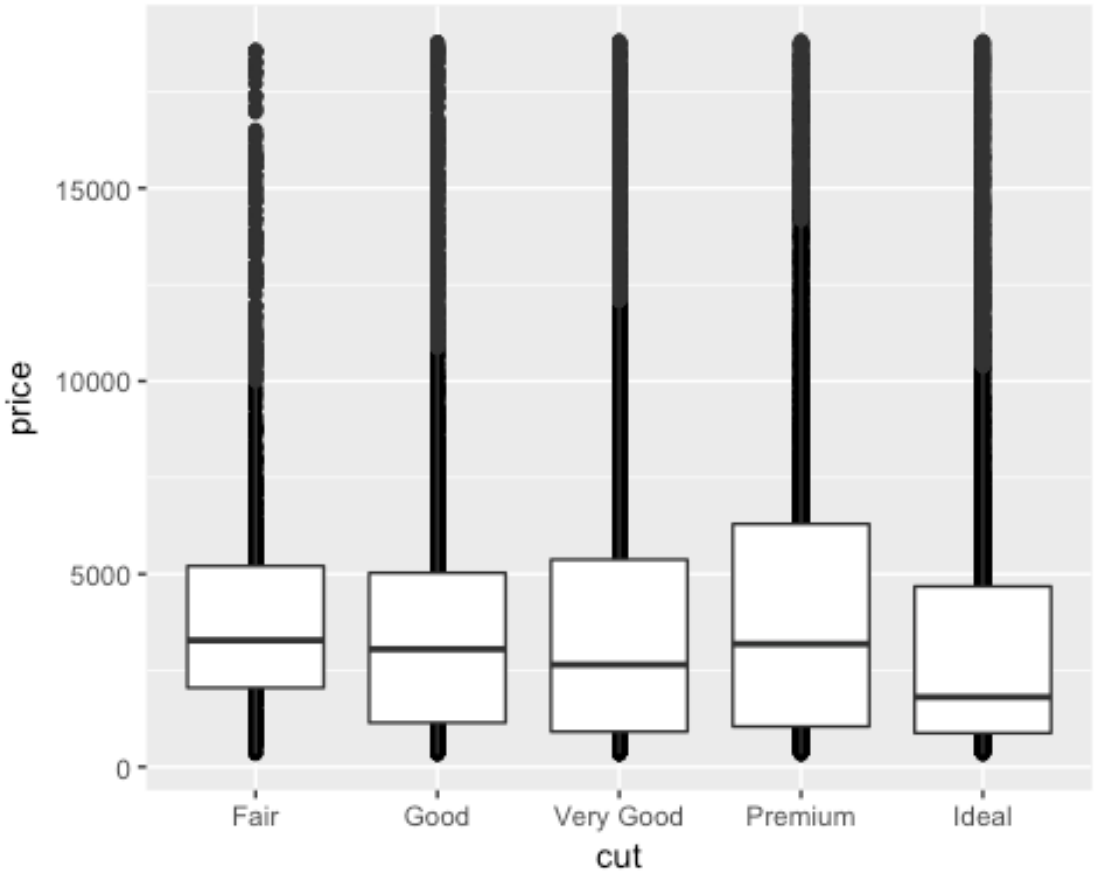
```
## # A tibble: 6 × 10
##   carat       cut color clarity depth table price     x     y     z
##   <dbl>     <ord> <ord>   <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23     Ideal     E     SI2  61.5    55   326  3.95  3.98  2.43
## 2  0.21   Premium     E     SI1  59.8    61   326  3.89  3.84  2.31
## 3  0.23      Good     E     VS1  56.9    65   327  4.05  4.07  2.31
## 4  0.29   Premium     I     VS2  62.4    58   334  4.20  4.23  2.63
## 5  0.31      Good     J     SI2  63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good     J    VVS2  62.8    57   336  3.94  3.96  2.48
```

You can plot:

```
qplot(carat, price, data=diamonds, color=cut)
```



You can also embed plots, for example ({r, echo=FALSE}):



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

For example, the following code

Inline R Code

You can also evaluate R expressions inline by enclosing the expression within a single back-tick qualified with 'r'. For example, the following code:

Markdown Inline

Results in this output: "I counted 2 red trucks on the highway."

Knitting

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.

Understand the basics of LaTeX

Understand the basics of LaTeX

Basic console output

To insert an R code chunk, you can type it manually or just press `Chunks - Insert chunks` or use the shortcut key. This will produce the following code chunk:

Pressing tab when inside the braces will bring up code chunk options.

The following R code chunk labelled `basicconsole` is as follows:

```
x <- 1:10
y <- round(rnorm(10, x, 1), 2)
df <- data.frame(x, y)
df
```

```
##     x    y
## 1   1 0.92
## 2   2 3.93
## 3   3 0.95
## 4   4 4.28
## 5   5 3.47
## 6   6 5.73
## 7   7 8.23
## 8   8 8.63
## 9   9 9.35
## 10 10 9.44
```

The code chunk input and output is then displayed as follows:

```
x <- 1:10
y <- round(rnorm(10, x, 1), 2)
df <- data.frame(x, y)
df
```

```
##     x     y
## 1   1 -0.12
## 2   2  1.13
## 3   3  3.04
## 4   4  3.42
## 5   5  4.18
## 6   6  6.11
## 7   7  7.36
## 8   8  8.05
## 9   9 10.37
## 10 10  9.62
```

Syntax

Plain text

End a line with two spaces to start a new paragraph.

```
Emphasis, aka italics, with *asterisks* or _underscores_.

> Blockquote

*italics* and _italics_

**bold** and __bold__

Strong emphasis, aka bold, with **asterisks** or __underscores__.

Combined emphasis with **asterisks and _underscores_**.

superscript^2^

Strikethrough uses two tildes. ~~Scratch this.~~

~~strikethrough~~
```

Emphasis, aka italics, with *asterisks* or *underscores*.

Blockquote

italics and *italics*

bold and **bold**

Strong emphasis, aka bold, with **asterisks** or **underscores**.

Combined emphasis with **asterisks and *underscores***.

superscript^{2}

Strikethrough uses two tildes. ~~Scratch this.~~

~~strikethrough~~

```
Inline `code` has `back-ticks around` it.

[link](www.rstudio.com)

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6
```

```
endash: --
emdash: ---
ellipsis: ...

inline equation: $A = \pi*r^{2}$

horizontal rule (or slide break):

***

> block quote

* unordered list

* item 2
    + sub-item 1
    + sub-item 2

1. ordered list
2. item 2
    + sub-item 1
    + sub-item 2

Simple dot points:

* Point 1
* Point 2
* Point 3

and numeric dot points:

1. Number 1
2. Number 2
3. Number 3

and nested dot points:

* A
    * A.1
    * A.2
* B
    * B.1
    * B.2


Table Header  | Second Header
------------- | -------------
```

```
Table Cell    | Cell 2
Cell 3        | Cell 4
```

Inline `code` has `back-ticks around` it.

link

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

endash: -- emdash: --- ellipsis: ...

inline equation: $A = \pi * r^2$

horizontal rule (or slide break):

---

block quote

- unordered list
- item 2
  - sub-item 1
  - sub-item 2

1. ordered list
2. item 2
   - sub-item 1
   - sub-item 2

Simple dot points:

- Point 1
- Point 2
- Point 3

and numeric dot points:

1. Number 1

2. Number 2
3. Number 3

and nested dot points:

- A
  - A.1
  - A.2
- B
  - B.1
  - B.2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

Plots

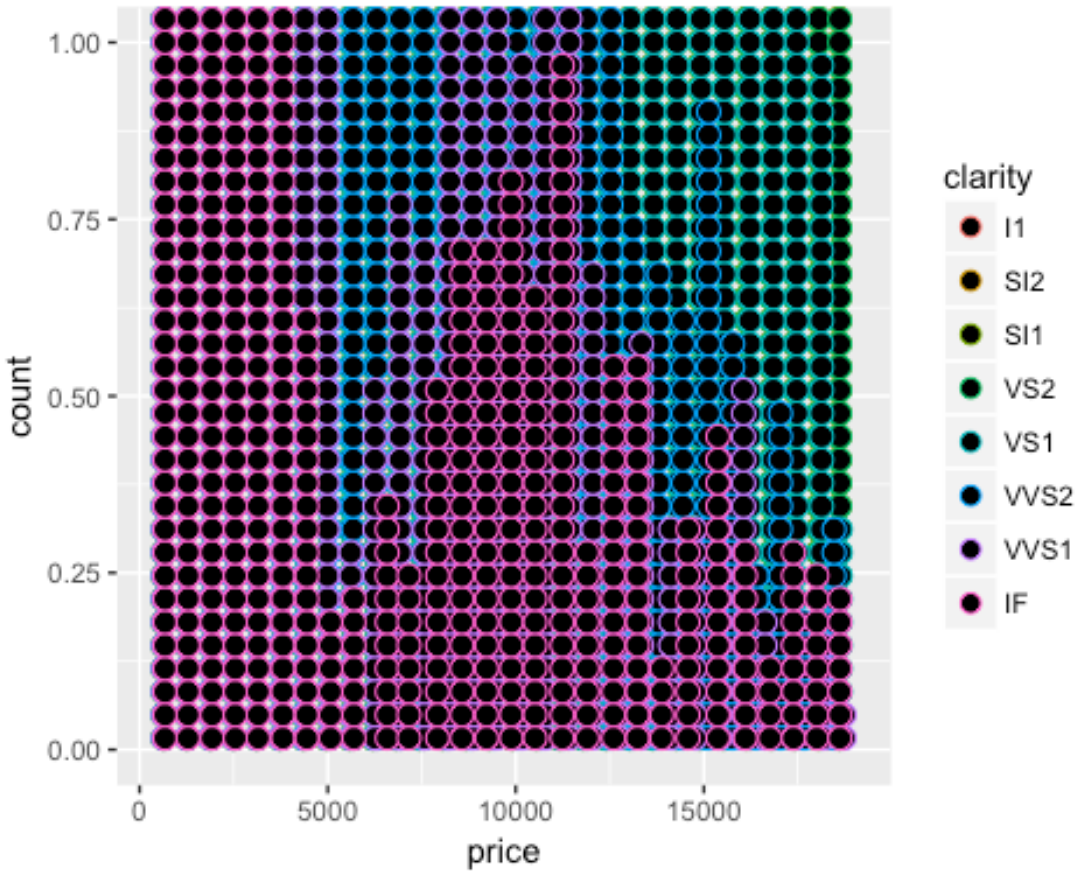
Images generated by `knitr` are saved in a figures folder. However, they also appear to be represented in the HTML output using a data URI scheme. This means that you can paste the HTML into a blog post or discussion forum and you don't have to worry about finding a place to store the images; they're embedded in the HTML.

Simple plot

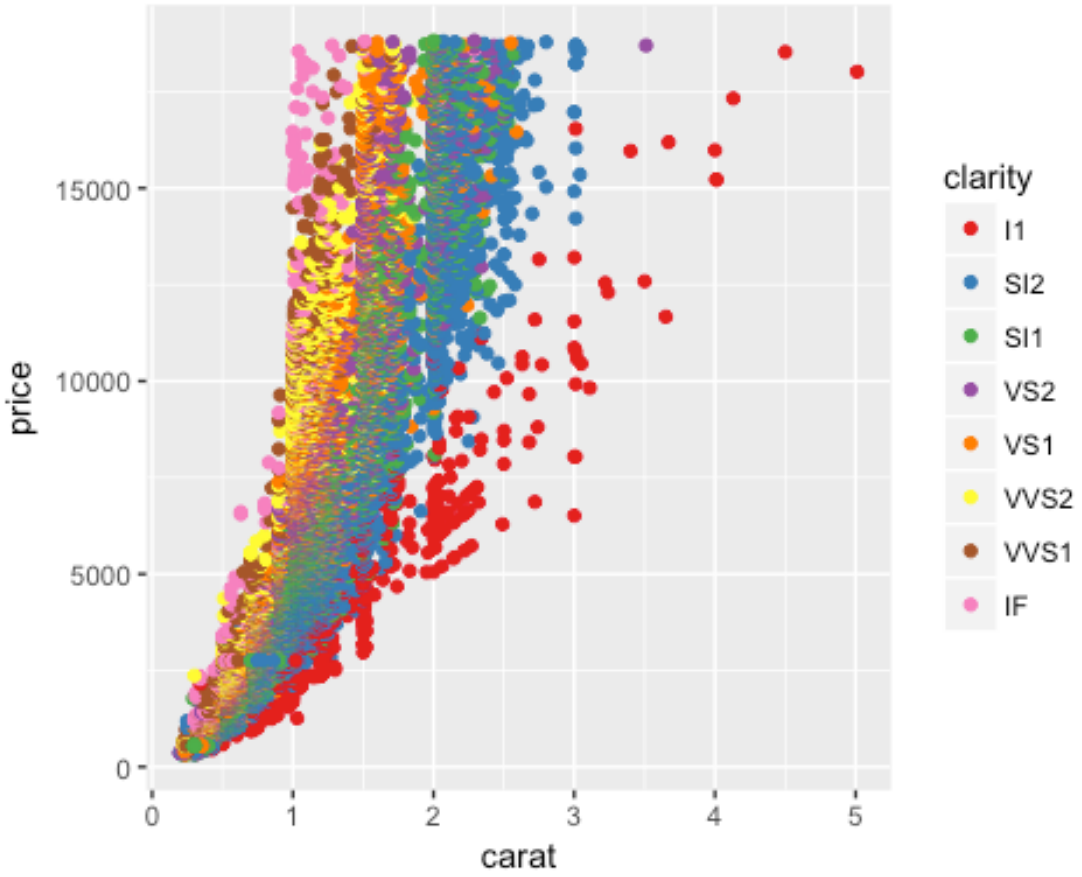
Here is a basic plot using base graphics:

```
qplot(price, data=diamonds, colour=clarity, geom="dotplot")

## `stat_bindot()` using `bins = 30`. Pick better value with
`binwidth`.
```




```
qplot(carat, price, data=diamonds, colour=clarity)+
scale_colour_brewer(palette="Set1")
```



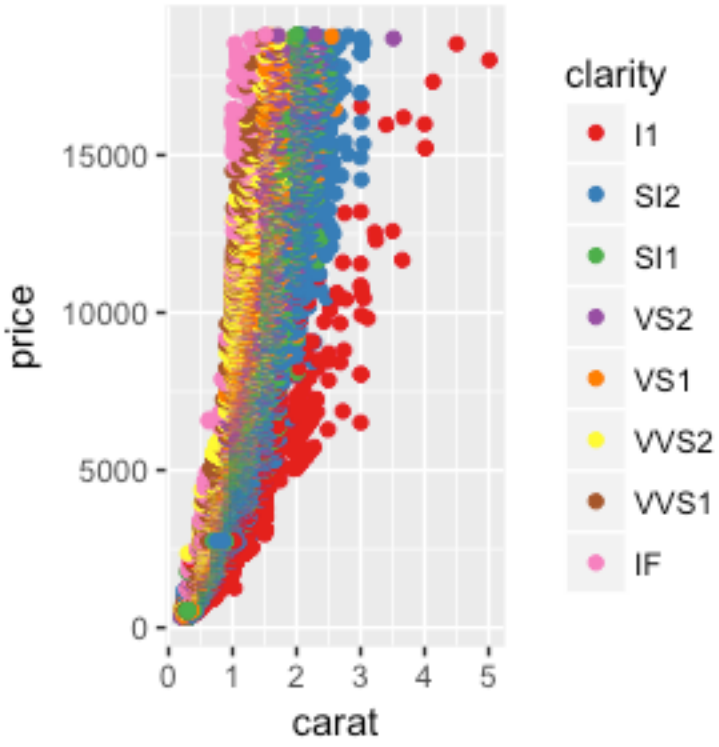
Note that unlike traditional Sweave, there is no need to write `fig=TRUE`.

Control figure size

The following is an example of a smaller figure using `fig.width` and `fig.height` options.

```
, fig.width=3, fig.height=3 qplot(carat, price, data=diamonds,
colour=clarity)+ scale_colour_brewer(palette="Set1")
```

```
qplot(carat, price, data=diamonds, colour=clarity)+
scale_colour_brewer(palette="Set1")
```



R Code chunk features

Create Markdown code from R

The following code hides the command input (i.e., `echo=FALSE`), and outputs the content directly as code (i.e., `results=asis`, which is similar to `results=tex` in Sweave).

Here are some dot points

- The value of y[1] is -0.12
- The value of y[2] is 1.13
- The value of y[3] is 3.04

Here are some dot points

- The value of y[1] is -0.12
- The value of y[2] is 1.13
- The value of y[3] is 3.04

Create Markdown table code from R

x	y
1	-0.12
2	1.13

3	3.04
4	3.42
5	4.18
6	6.11
7	7.36
8	8.05
9	10.37
10	9.62

x	y
1	-0.12
2	1.13
3	3.04
4	3.42
5	4.18
6	6.11
7	7.36
8	8.05
9	10.37
10	9.62

Control output display

The folllowing code supresses display of R input commands (i.e., `echo=FALSE`) and removes any preceding text from console output (`comment=""`; the default is `comment="##"`).

```
  x     y
1 1 -0.12
2 2  1.13
3 3  3.04
4 4  3.42
5 5  4.18
6 6  6.11

  x     y
1 1 -0.12
2 2  1.13
3 3  3.04
4 4  3.42
5 5  4.18
6 6  6.11
```

Cache analysis

Caching analyses is straightforward. Here's example code. On the first run on my computer, this took about 10 seconds. On subsequent runs, this code was not run.

If you want to rerun cached code chunks, just delete the contents of the cache folder

```
for (i in 1:5000) {
  lm((i+1)~i)
}
```

Equations

Multiple syntaxes exist:

dollars: $inline$ and

$$display$$

. (same as LaTeX) single backslash: $inline$ and

$$display$$

(same as LaTeX) double backslash: \(inline\) and \[display\] naked latex environment

.

Equations are included by using LaTeX notation and including them either between single dollar signs (inline equations) or double dollar signs (displayed equations). If you hang around the Q&A site CrossValidated you'll be familiar with this idea.

There are inline equations such as $y_i = \alpha + \beta x_i + e_i$.

And displayed formulas:

$$\frac{1}{1 + \exp(-x)}$$

Typesetting Equations

Inline vs. Display Material

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.

```
This summation expression is in display form.

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

This summation expression is in display form.

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

Some LaTeX Basics

In this section, we show you some rudiments of the LaTeX typesetting language.

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:

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

$$X_i$$

$$X_i$$

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

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

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

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.

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

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

Summation Expressions

These are indicated with the `' 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.

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

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

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

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

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

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.

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

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

Special Symbols

All common mathematical symbols are implemented, and you can find a listing on a LaTeX cheat sheet.

```
* Subscripts to get \( a_{b} \) write: $a_{b}$
* Superscripts write \( a^{b} \) write: $a^{b}$
* Greek letters like \( \alpha, \beta, \ldots \) write: $\alpha, \beta,
\ldots$
* Sums like \( \sum_{n=1}^N \) write: $\sum_{n=1}^N$
* Multiplication like \( \times \) write: $\times$
* Products like \( \prod_{n=1}^N \) write: $\prod_{n=1}^N$
* Inequalities like \( <, \leq, \geq \) write: $<, \leq, \geq$
* Distributed like \( \sim \) write: $\sim$
* Hats like \( \widehat{\alpha} \) write: $\widehat{\alpha}$
* Averages like \( \bar{x} \) write: $\bar{x}$
* Fractions like \( \frac{a}{b} \) write: $\frac{a}{b}$
```

- Subscripts to get (a_{b}) write: a_b
- Superscripts write (a^{b}) write: a^b
- Greek letters like (, ,) write: $\alpha, \beta, \ldots$
- Sums like (_{n=1}^N) write: $\sum_{n=1}^N$
- Multiplication like ( ) write: $\times$
- Products like (_{n=1}^N) write: $\prod_{n=1}^N$
- Inequalities like (<, ,) write: $<, \leq, \geq$
- Distributed like ( ) write: $\sim$
- Hats like ( ) write: $\hat{\alpha}$
- Averages like ({x}) write: $\bar{x}$
- Fractions like ( ) write: $\frac{a}{b}$

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.)

```
$$a \pm b$$
$$x \ge 15$$
$$a_i \ge 0~~~\forall i$$
```

$$a \pm b$$

$$x \geq 15$$

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

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.

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

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

Matrices

Matrics 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 `\\`.

```
$$\begin{array}
{rrr}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}
$$
```

$$\begin{array}{rrr} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

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

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

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

Mathjax script

knitr provides self-contained HTML code that calls a Mathjax script to display formulas. However, in order to include the script in my blog posts I took the script and incorporated it into my blogger template. If you are viewing this post through syndication or an RSS reader, this may not work. You may need to view this post on my website.

Tables

Tables can be included using the following notation

A	B	C
1	Male	Blue
2	Female	Pink

Hyperlinks

- Here is my blog nikbearbrown.com my blog nikbearbrown.com.

Images

Here's an example image:

image William Sealy Gosset, known as "Student", British statistician. Picture taken in 1908

William Sealy Gosset, known as "Student", British statistician. Picture taken in 1908

William Sealy Gosset (1876 - 1937) was a chemist and statistician, better known by his pen name Student. He worked in a beer brewery and his testing of very small patches led him to discover certain small-sample distributions.This led to the development of Student's t-Test. His communications with Fisher on the subject are legendary.

See

- William Sealy Gosset [William Sealy Gosset] (https://en.wikipedia.org/wiki/William_Sealy_Gosset)
- Famous Statisticians - Department of Statistics, GMU [Famous Statisticians - Department of Statistics, GMU] (http://statistics.gmu.edu/pages/famous.html)

Quote

Let's quote some stuff:

To be, or not to be, that is the question: Whether 'tis nobler in the mind to suffer The slings and arrows of outrageous fortune,

Conclusion

- R Markdown is awesome.
  - Analysis is much more readable and understandable when text, code, figures and equations can be shown together.
  - For journal articles, LaTeX will presumably still be required.
  - Code completion on R code chunk options is really helpful. See also chunk options documentation on the knitr website.

Learning More

- Tufte Handout
- Christopher Gandrud
- Markcus Gesmann
- Rstudio on R Markdown
- The website for the knitr package. Knitr is an extremely powerful tool for dynamic content generation and the website has a wealth of documentation and examples to help you utilize it to its full potential.
- Markdown Basics which describes the most commonly used markdown constructs.
- R Code Chunks, which goes into more depth on customizing the behavior of embedded R code.
- R Markdown Cheat Sheet (PDF) a quick guide to the most commonly used markdown syntax, knitr options, and output formats.
- R Markdown Reference Guide (PDF), a more comprehensive reference guide to markdown, knitr, and output format options.

- Bibliographies and Citations, which describes how to include references in R Markdown documents.
- Interactive Documents with Shiny, which describes how to make R Markdown documents interactive using Shiny.
- Compiling Notebooks which describes how to compile HTML, PDF, or MS Word notebooks from R scripts.

Evaluate (Test your knowledge)

- Try writing a simple R Markdown document with an equation, a web link, an image, and some running R code.