

Markdown demo

Erin Shellman

June 8, 2015

Contents

Code chunks	1
Formatting headers	4
H1	4
H2	4
Formatting text	5
Tables	5

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>.

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.

Code chunks

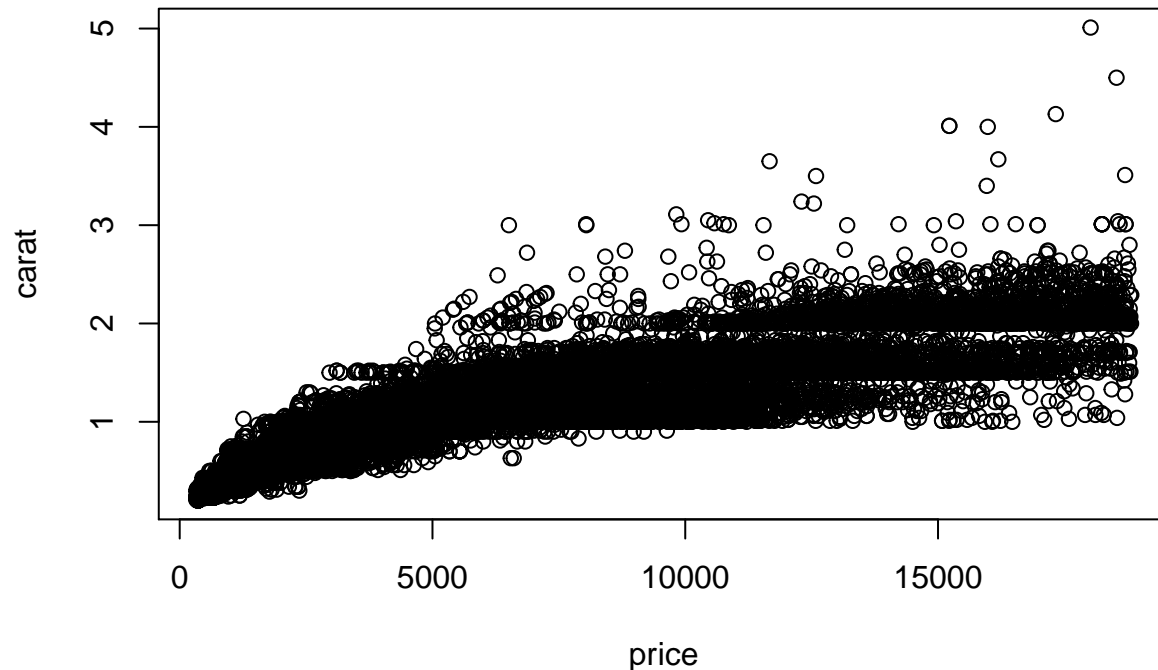
You can embed an R code chunk like this:

```
library(ggplot2)
data(diamonds)
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          J: 2808 VVS1     : 3655
##                      (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
##
```

You can also embed plots, for example:

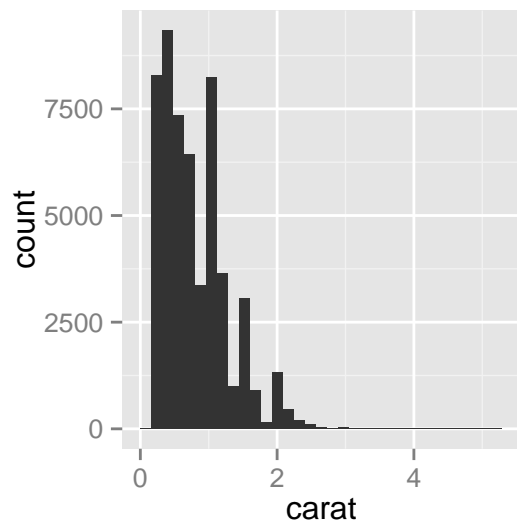


Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot. There are lots of different arguments we can give the code chunks that will change the behavior of how they render:

Table Header	Second Header	Second Header
<code>eval</code>	<code>TRUE</code>	Whether to evaluate the code and include its results
<code>echo</code>	<code>TRUE</code>	Whether to display code along with its results
<code>warning</code>	<code>TRUE</code>	Whether to display warnings
<code>error</code>	<code>FALSE</code>	Whether to display errors
<code>message</code>	<code>TRUE</code>	Whether to display messages
<code>tidy</code>	<code>FALSE</code>	Whether to reformat code in a tidy way when displaying it
<code>results</code>	<code>"markup"</code>	<code>"markup"</code> , <code>"asis"</code> , <code>"hold"</code> , or <code>"hide"</code>
<code>cache</code>	<code>FALSE</code>	Whether to cache results for future renders
<code>comment</code>	<code>"##"</code>	Comment character to preface results with
<code>fig.width</code>	<code>7</code>	Width in inches for plots created in chunk
<code>fig.height</code>	<code>7</code>	Height in inches for plots created in chunk

Let's try some and see what happens. Here's a code chunk with just plot size arguments:

```
ggplot(diamonds, aes(x = carat)) +  
  geom_histogram()
```

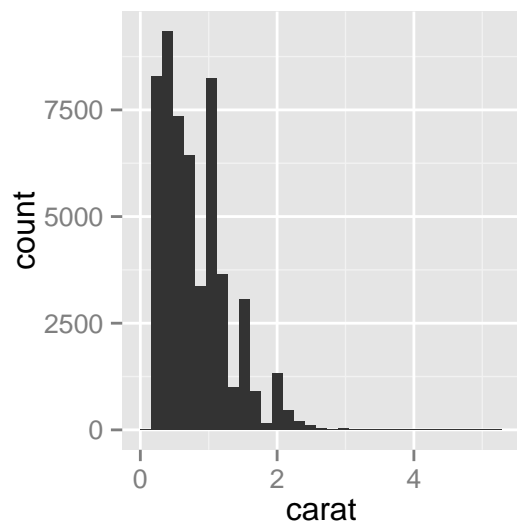


Don't run the code:

```
# we get nuthin'  
ggplot(diamonds, aes(x = carat)) +  
  geom_histogram()
```

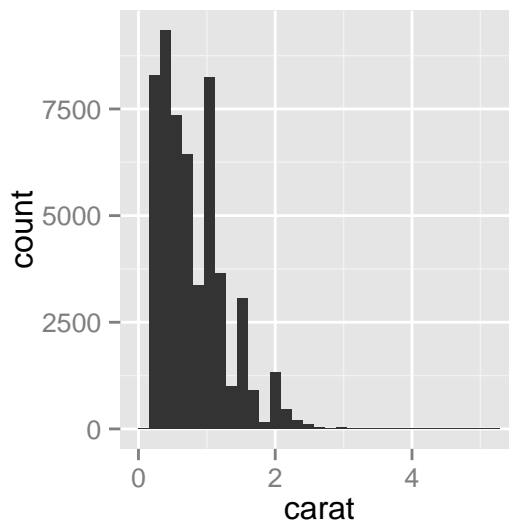
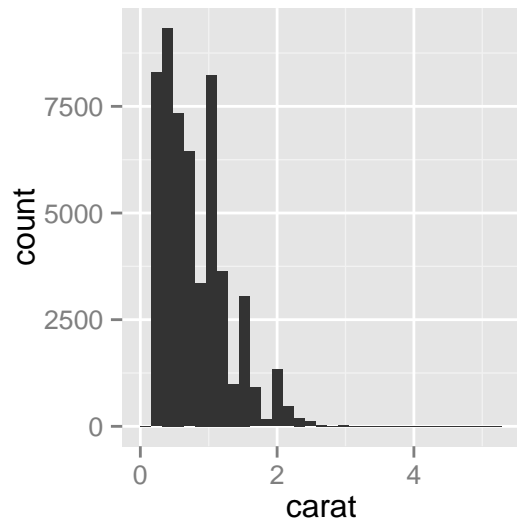
Don't print warnings:

```
ggplot(diamonds, aes(x = carat)) +  
  geom_histogram()
```



Automagically tidy up your code:

```
ggplot(diamonds, aes(x = carat)) + geom_histogram()
```



And if you want to skip the code...

Formatting headers

You can denote different header sizes with '#.'

H1

H2

H3

H4

H5 **H6**

Formatting text

Make lists with `'*'`.

- *this text is in italics*
- **this text is bold**
- `this text is in-line code`
- this text has an equation: $y = \beta_0 + \beta_1 x$

This text is a quoteable quote.

You can also make ordered lists naturally like this:

1. Here's an item
2. And the second one
3. The third item has some sub-points
 - great point
 - another great point
 - blah

Tables

You can create your own tables like this:

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

Or create them with code!

```
library(knitr)
agg = aggregate(price ~ clarity + color, diamonds, mean)
kable(xtabs(price ~ ., data = agg))
```

	D	E	F	G	H	I	J
I1	3863.024	3488.422	3342.182	3545.693	4453.414	4302.185	5254.060
SI2	3931.101	4173.826	4472.625	5021.684	6099.895	7002.649	6520.958
SI1	2976.146	3161.838	3714.226	3774.787	5032.415	5355.020	5186.048
VS2	2587.226	2750.942	3756.795	4416.256	4722.414	5690.506	5311.059
VS1	3030.159	2856.294	3796.718	4131.362	3780.689	4633.184	4884.461
VVS2	3351.128	2499.674	3475.513	3845.283	2649.067	2968.233	5142.397
VVS1	2947.913	2219.820	2804.277	2866.821	1845.658	2034.862	4034.176
IF	8307.370	3668.506	2750.836	2558.034	2287.870	1994.937	3363.882