

# R Markdown Primer

*Erico Cardelli Author*

## Contents

<b>References</b>	<b>2</b>
R markdown link to topic notebook . . . . .	2
<b>My First Section</b>	<b>2</b>
<b>New Section</b>	<b>2</b>
<b>Subsections</b>	<b>2</b>
This is a Subsection . . . . .	2
Another Subsection . . . . .	3
<b>Back to Sections</b>	<b>3</b>
<b>Formatting Text</b>	<b>3</b>
<b>Lists</b>	<b>3</b>
Unordered List . . . . .	3
Ordered List . . . . .	3
<b>Links</b>	<b>4</b>
<b>R Time!!!!</b>	<b>4</b>
<b>Text in between code? Such Heresy</b>	<b>4</b>
<b>Naming Chunks?</b>	<b>4</b>
Better Examples . . . . .	5
<b>Plots</b>	<b>5</b>
<b>Including Javascript</b>	<b>6</b>
<b>Plot Improvements</b>	<b>6</b>
A Caption . . . . .	6

# References

## R markdown link to topic notebook

Class OneNote

### Be Contextually aware

After your metadata section do a “metasetr” which will feed new metadata into new global settings

```
# Now a snippet to set things right
# its funny that outside of this block a comment is a giant Title

# This and by using autotab, I was able to piece out
# knitr::options for chunks then
# RMarkdown is rendering
# knitr is evaluating
# and the $ is for functions in the knitr opts_chunk function SET (whatever,otherstuff,etc)
knitr::opts_chunk$set(cache=TRUE, fig.align = 'center', echo = TRUE)
```

## My First Section

We start a new section with a single HASHTAG and this is the first paragraph in that section. Notice how the text wraps around to the next line in the editor but it still has the same line number and that's pretty cool. That's because all text on a single line is considered a paragraph.

Leaving a blank line starts a new paragraph. Remember Markdown is highly spaced and cased aware. If you don't leave a blank line you are still in the same paragraph

So it is very important to pay attention to spaces.

F7 for spell check.

## New Section

You start a new section by leaving a blank line and then starting the next line with a hash tag. No need to close the previous section, just start a new one. (Similar to La-Tex)

## Subsections

Creating a subsection is easy. Two Hash tags

### This is a Subsection

We start a section in a section to start it

You can use add decorator module

## Another Subsection

Same rules apply no need to close previous subsection just keep a blank line in between them  
the output type is actually an R function you are specifying and is in the R Markdown Package

## Back to Sections

## Formatting Text

One Underline is for *Emphasized*

*Emphasized Words Go Here*

Strong Text is with two underlines **Strong**

**Strong Text Goes Here**

And you can combine them with Three Underlines ***Bold and Strong***

And finally code snippets are the BACKTIC under the Esc

This is computer Code

## Lists

### Unordered List

Fruit

- Apple
- Orange
- Banana
- Mango
- Durian
- Watermellon
- Dragonfruit

### Ordered List

To create a ordered list use 1. at the beginning of your list items Hit Dice and a sublist with a. to make the sublist

### Sublists

I added subs in there too

1. Bug
  - a. gnat
  - b. fly
  - c. spider

- d. wasp
- 2. Fool
  - i. dope
  - ii. idiot
- 3. Knight
- 4. Troll
- 5. Dragon

## Links

You put the text you want displayed in square brackets and then the url into parenthesis

[Google] (<https://www.google.com>) = Google

Class OneNote

## R Time!!!!

*Alt+Ctrl+I for new code chunk*

**1+1**

```
## [1] 2
```

R Markdown showed you the code, executed cleanly, and displays results

```
# this is a comment inside of R code
# here I will set a variable that is only inside this code bubbles
x<-3
x
```

```
## [1] 3
```

## Text in between code? Such Heresy

x\*2

```
## [1] 6
```

Chunks are cumulative and serial but not in any way related to the console or any other ENVIRONMENT

## Naming Chunks?

So in the code chunk right after the r that defines it as R Code put a space and then a name followed by the other curlybrace See the Example here

```
x*7+3
```

```
## [1] 24
```

Lets show the results of a code chunk without showing the code

```
## [1] 19
```

by doing echo=false the code is safely hidden but results flow  
look to the left and you can see carrots that let you fold the stuffs

## Better Examples

Alt+Shift+K will show you keyboard Shortcuts

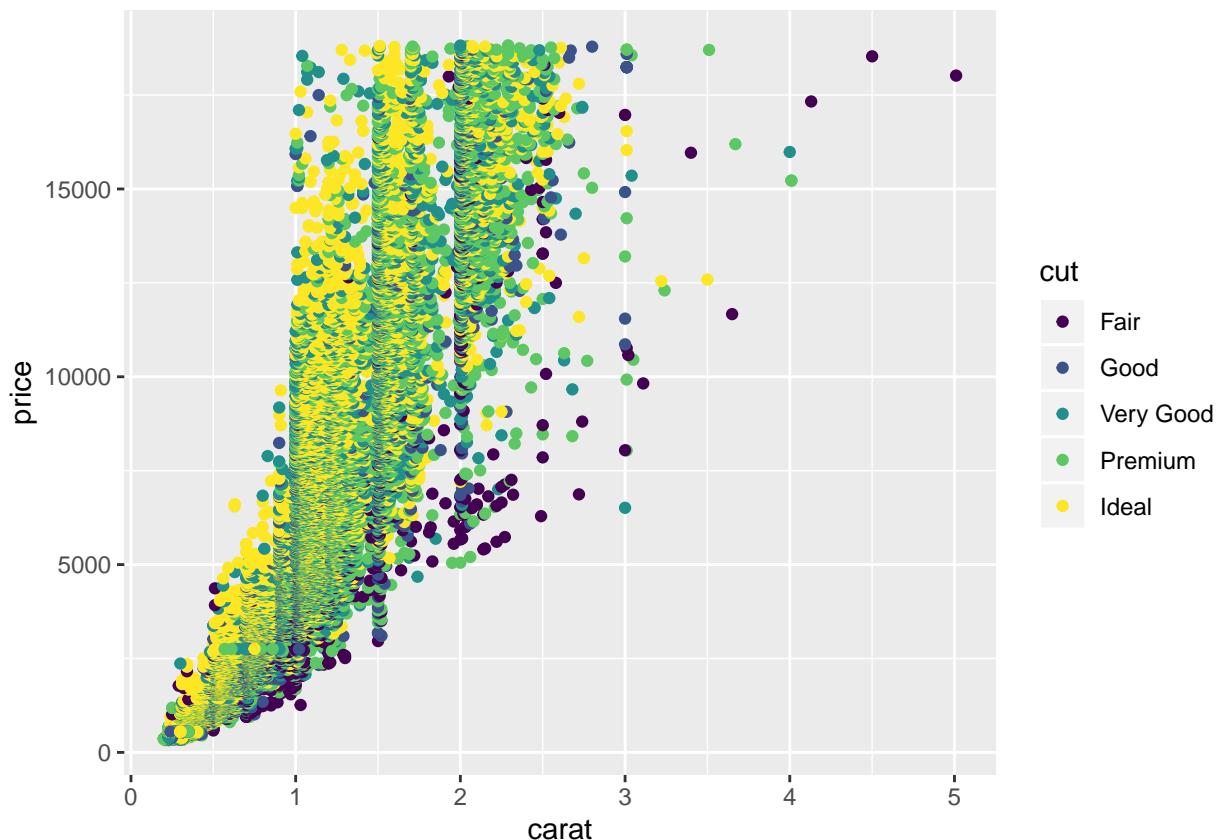
## Plots

Plots all make sense of data but first we have to prepare for the plots And loading more prerequisites

```
library(ggplot2)
```

Put the library code alone to enable smart reuse

```
ggplot(diamonds, aes(x=carat, y=price, color=cut)) + geom_point()
```



*What is amazing about this is that the picture is embeded as a graphic but its encoded so anyone can see it and all the images are embeded*

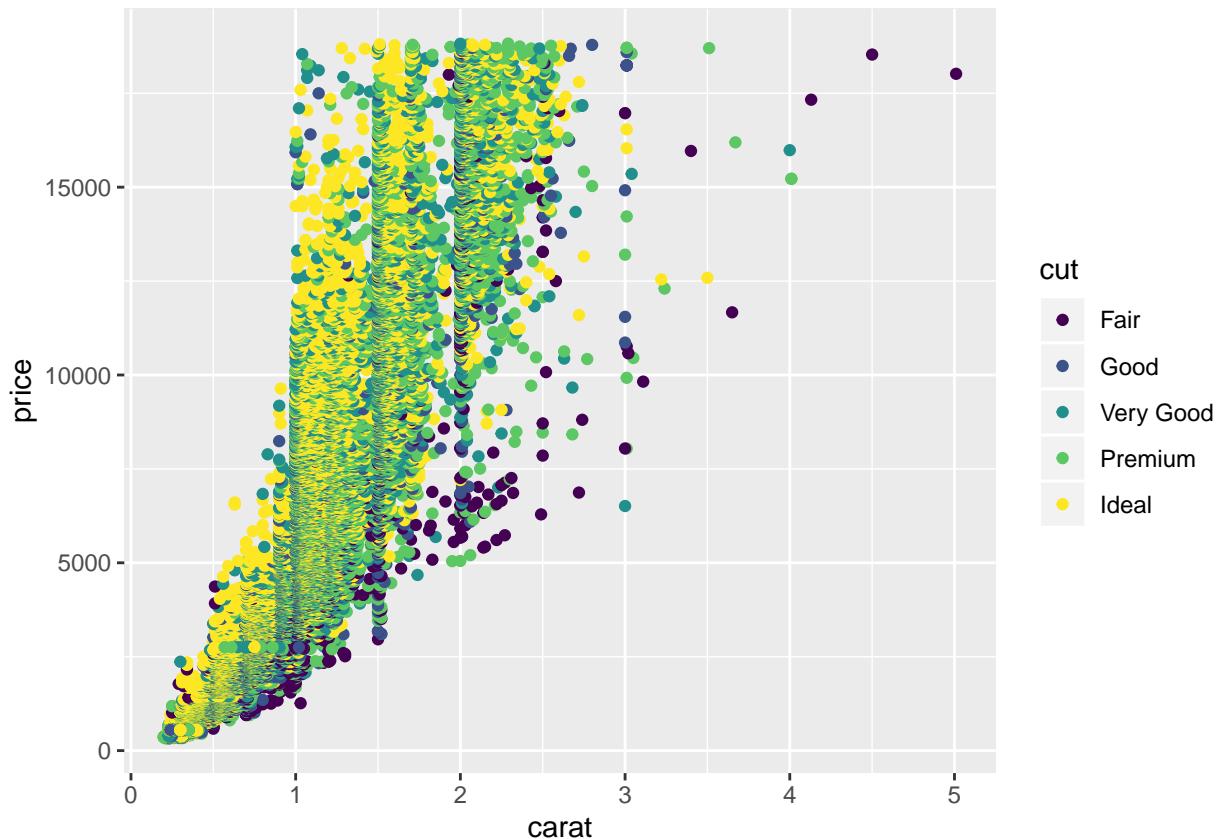
## Including Javascript

The web page actually embeds everything for your use

## Plot Improvements

I want to center the plot...

```
ggplot(diamonds, aes(x=carat, y=price, color=cut)) + geom_point()
```



The alignment directive in that code was easy to figure out with the comma intellisyntax

## A Caption

```
ggplot(diamonds, aes(x=carat, y=price, color=cut)) + geom_point()
```

Lots to unpack in this code... but where it says cache=TRUE, that means that if this code snippet hasn't changed use it as is. possibly saving a lot of time

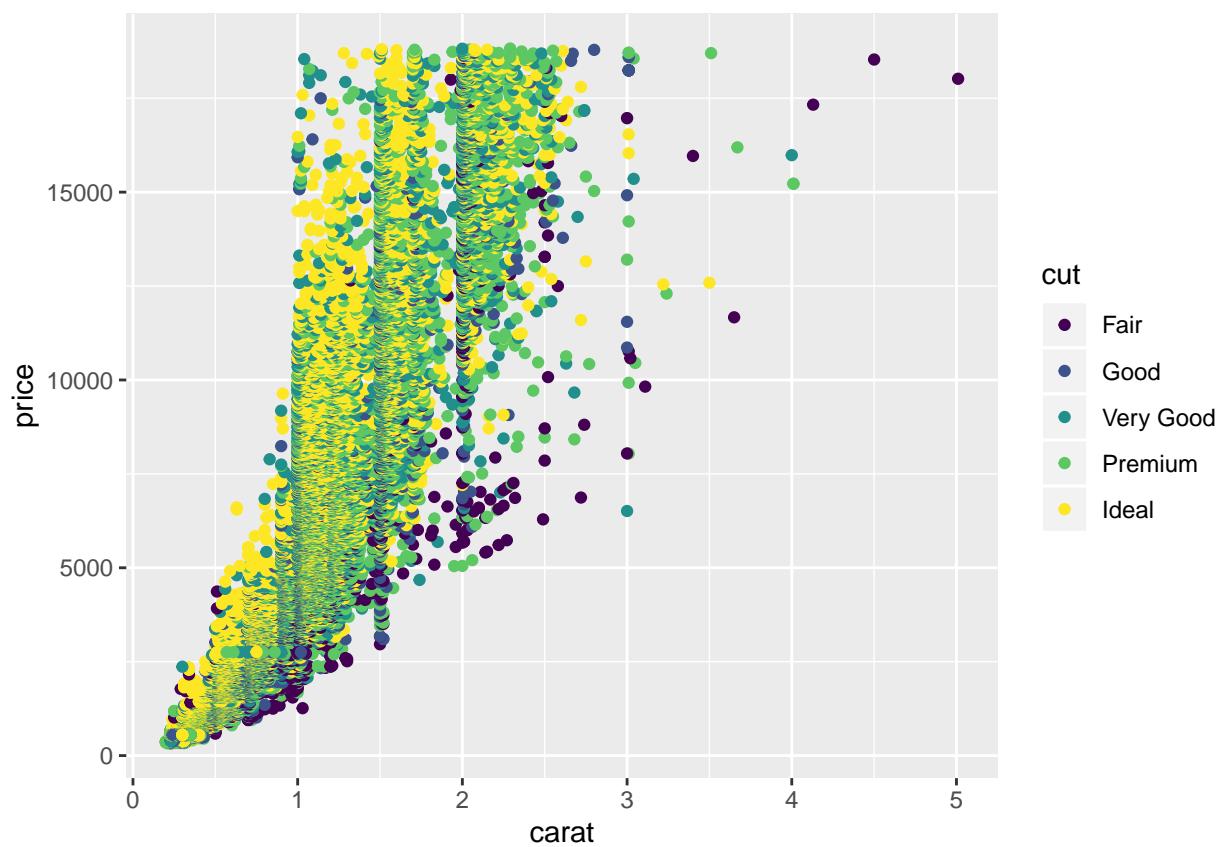


Figure 1: A scatterplot of diamond price vs. size, color coded according to diamond cut