Jumpstart: Knitr & RMarkdown
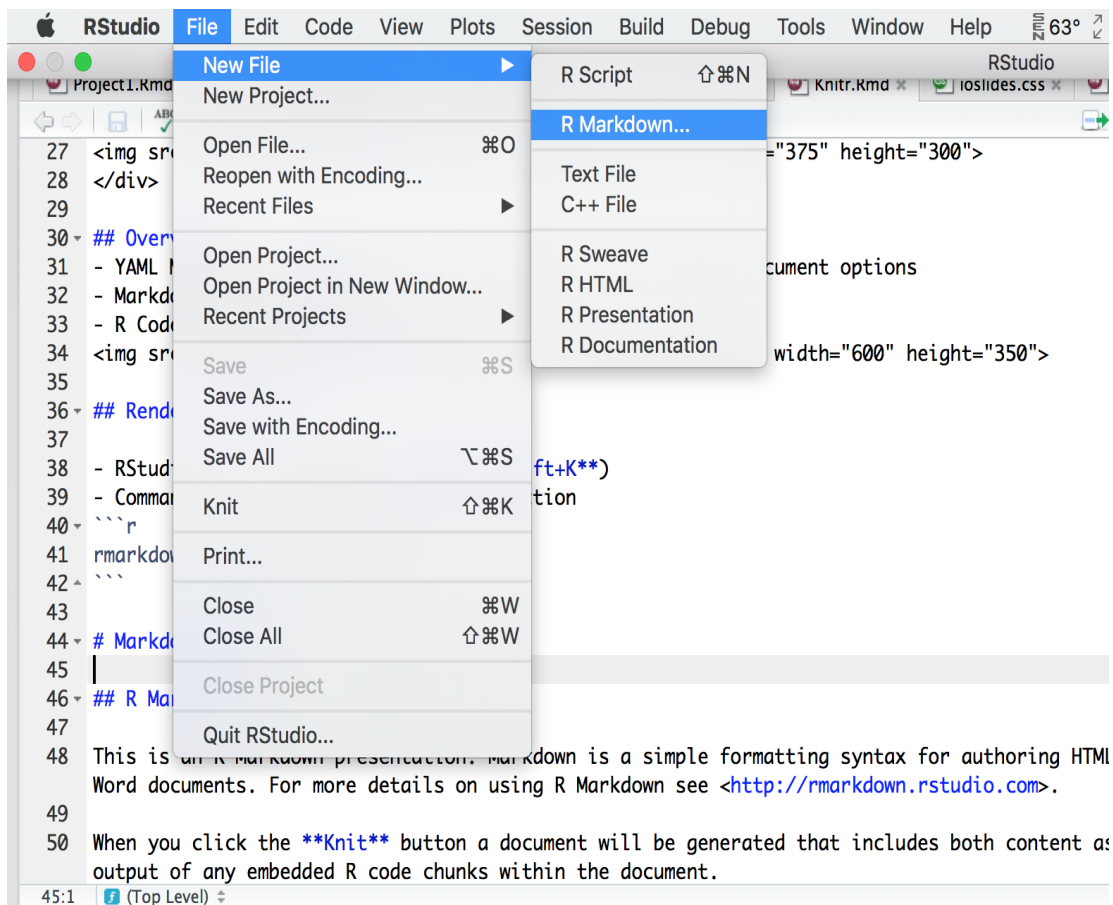
NYC Data Science Academy

2017-04-11

Outline

- HTML Documents

- Markdown Basics

- R Code Chunks

- Plots

- Example

Create a HTML Document

Create a HTML Document

Rendering Output

- RStudio: click the "Knit" button (**Command+Shift+K**)

- Command line: `rmarkdown::render` function

```
rmarkdown::render("input.Rmd")
```

- Aside: why is rmarkdown followed by the double colon sign?

- Try the following commands below:

```
library(plyr)
library(dplyr)
?summarise
```

Markdown Basics - Emphasis

- Empahsis could be either italic, bold or both.

- It also works on Slack!

```
*italic*   **bold**

_italic_   __bold__
```

I am italic **I am bold**

```
  ***italic and bold***
```

italic and bold

Headers & Lists

- In markdown, we have headers from 1 to 6 (largest to smallest), depending on the number of # signs you put in front of the text.

- Lists are straight forward but pay attention to the indentation.

```
# Header 1

## Header 2

### Header 3
```

```
Unordered List:
    * Item 1
    * Item 2
        + Item 2a
        + Item 2b

Ordered List:
    1. Item 1
    2. Item 2
    3. Item 3
        + Item 3a
        + Item 3b
```

Links

- A plain http address or add a link to a phrase.

- **Inline Link:**

```
[The link of your blog post](http://blog.nycdatascience.com/)
```

The link of your blog post

- **Reference Link:**

```
[homepage]: http://nycdatascience.com/
You can visit our [homepage]
```

You can visit our homepage

- For more details on using R Markdown see http://rmarkdown.rstudio.com.

R Code Chunks

- We wrapped regular R code within those backquotes and give it a name called cars. Then we can easily navigate through different code chunks.

```
```{r cars, echo=TRUE}
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
```

Basic Chunk Options

- `echo`(TRUE): whether to include R source code in the output file

- `eval`(TRUE): whether to evaluate the code chunk

- `include`(TRUE): if include=FALSE, nothing will be written into the output document, but the code is still evaluated

- `cache`(TRUE): if nothing changed in the code chunk, it will use the previous result as the output. Good for loading large dataset

- `warning`(TRUE): whether to preserve warnings in the output

- Set global chunk options:

```r setup, include=FALSE
knitr::opts_chunk$set()
```

Plots

- RMarkdown has its own standalone environment. So you have to load the library no matter what you have done in your current R session.
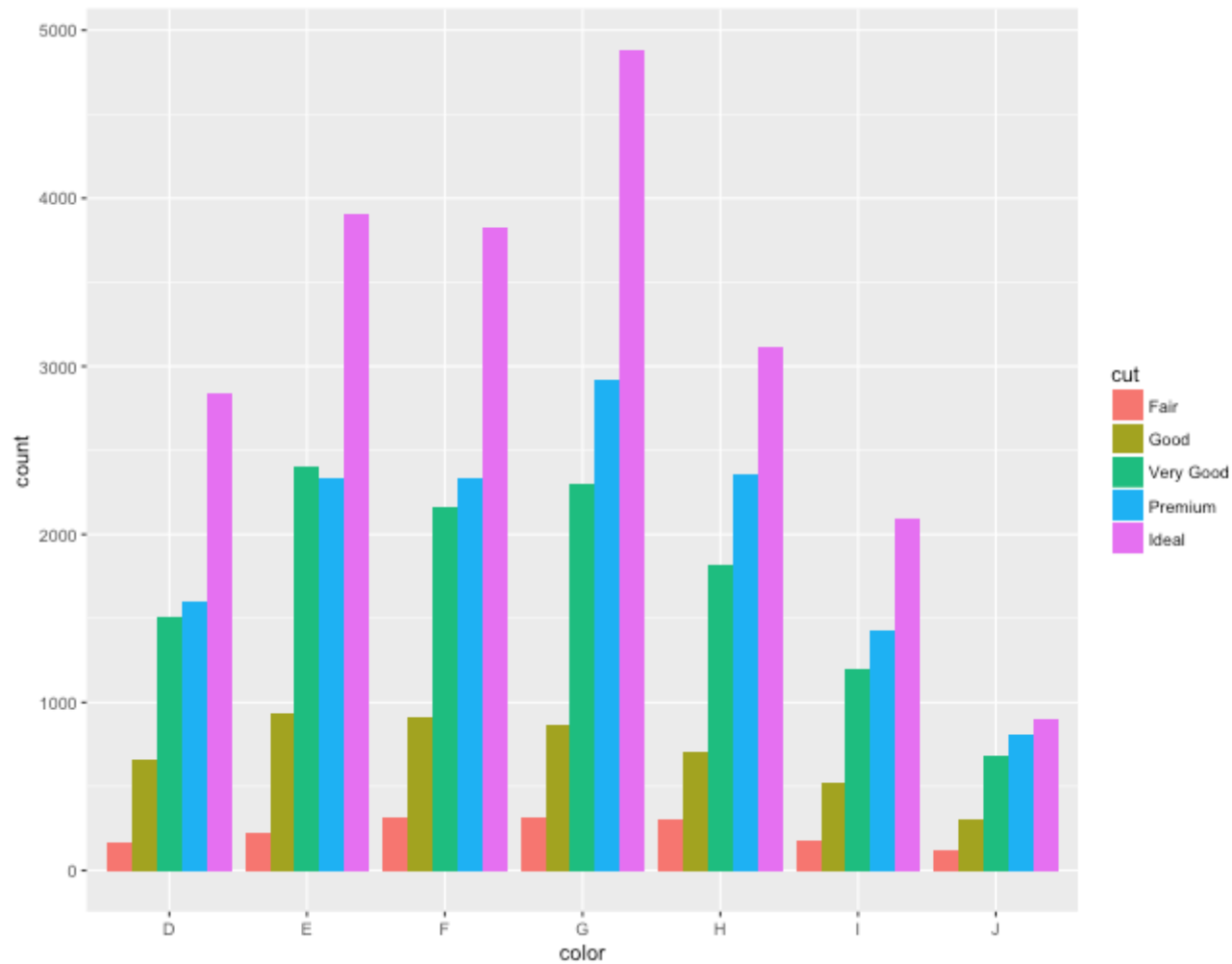
```r
```{r echo=TRUE, fig.height=7, fig.width=9}

library(ggplot2)

ggplot(data = diamonds, aes(x = color)) +

geom_bar(aes(fill = cut), position = "dodge")

```
```

Plots

Plots

- `dev`('png'): figure format(png, jpeg, tiff, svg, ...)

- `fig.path`('figure/'): figure path

- `fig.width`(7): figure width in inches

- `fig.height`(7): figure height in inches

Example

- NYC's Department of Health and Mental Hygiene (DOHMH) conducts unannounced inspections of restaurants at least once a year to check food handling, food temperature, personal hygiene and vermin control. Since 2010, NYC restaurants have to prominently post their Grade (e.g. A/B/C) which empowers diners with decision-making information and incentivizes establishments to improve their hygiene.

- Download the csv file from here and move it to your rmarkdown folder.

- Rename the file to `data.csv`.

Example

- Create a new R code chunk and read in the dataset.

- It is a good example to set `cache=TRUE` here since you don't want to read the whole csv file everytime you knit the rmarkdown file.

- Always use relative path to read your data instead of `setwd()` to make your project reproducible.

```
```{r load_data, cache=TRUE, warning=FALSE}
library(dplyr)
raw.df <- read.csv("./data.csv", stringsAsFactors = F)
raw.df <- tbl_df(raw.df)
```
```