Tutorial 4 - Git and RMarkdown

Rossi Abi-Rafeh

11/12/2017

R Markdown Basics

Create an R Markdown file

To create a new R Markdown file (.Rmd) in RStudio, select File -> New File -> R Markdown. . . , then choose the file type you want to create (Pdf, html, etc.) For this tutorial, choose html.

The .Rmd file you just created comes with default text that you can change to include you own text or R analysis.

R Markdown Syntax : The YAML Header

At the top of any R Markdown script is always the YAML header section enclosed by —

This is the minimum code chunk that should be put there. By default this includes a title, author, date, and the file type you want as output. You can change from PDF (pdf_document) to HTML (html_document) here if you wish. Whatever you set in the header section applies to the whole document.

This is an example of a YAML header at the top of an .Rmd script which creates an html document.

title: "R Markdown Example" author: Your Name date: 13/11/2017 output: html_document —

Inserting Code

Code that is included in your .Rmd document should be enclosed by three backwards apostrophes "‘ like the example below:

```
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
```

Inside an R code chunk, you can put whatever R code you want, including comments. 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
```

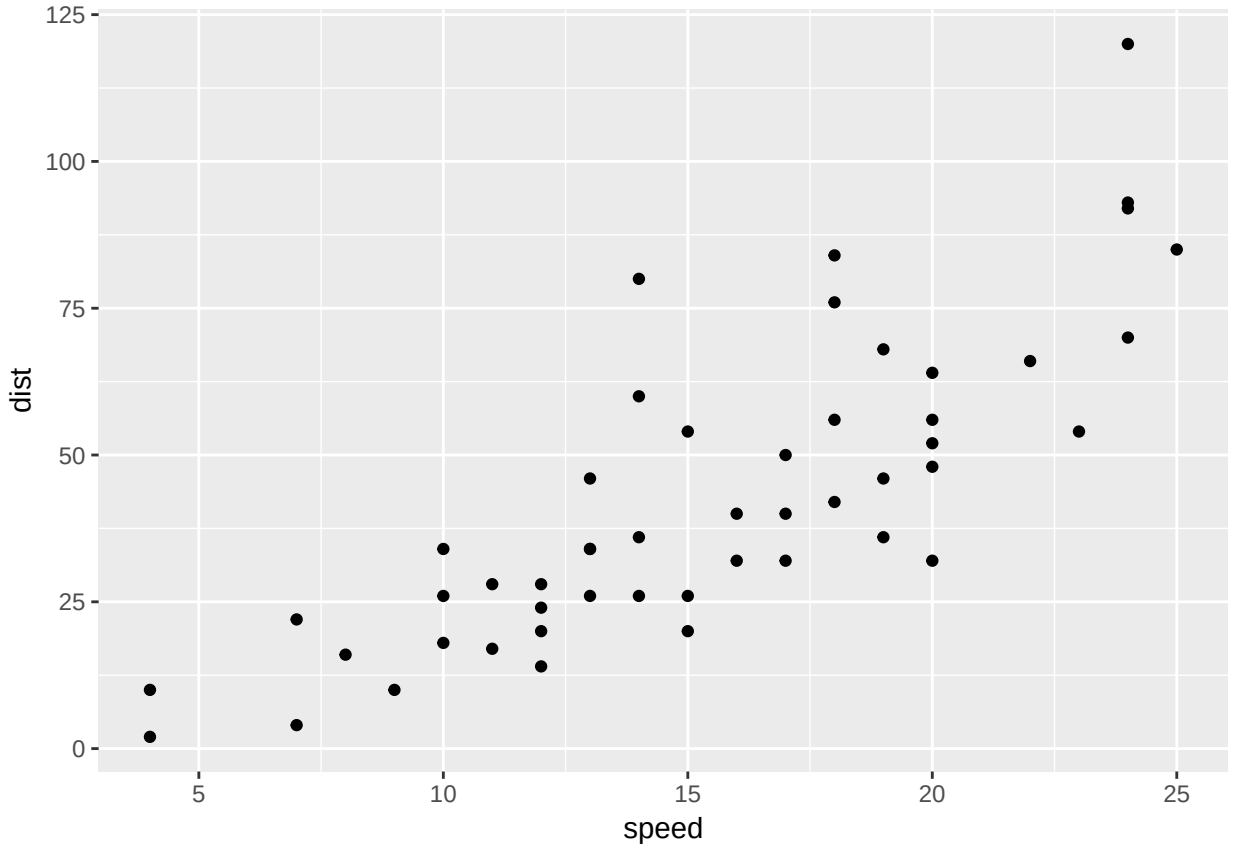
```
# cars is a dataset pre-loaded in R with information on speed and stopping
# distances of cars
```

The space inside the brackets is where you can assign rules for that code chunk. The code chunk above says that the code is R code.

Instructions for Code Chunks

You can generate a plot like this:

```
library(ggplot2)
ggplot(data = cars) + geom_point(mapping = aes(x = speed, y = dist))
```



You can create a dataframe:

```
library(tidyverse)
```

```
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
company <- c("Alphabet Inc", "Apple", "Facebook", "Amazon", "Ali Baba", "Snap Inc",
    "Criteo", "General Motors")
market_val <- c(714, 896, 518, 542, 477, 15, 2, 60)
# Market capitalization of company on the 13 Nov 2017 - from Yahoo! Finance
gafa_info <- data.frame(company, market_val)
head(gafa_info)
```

```
##        company market_val
## 1 Alphabet Inc        714
## 2        Apple        896
## 3     Facebook        518
## 4       Amazon        542
## 5     Ali Baba        477
## 6     Snap Inc         15
```

If you don't want a code to appear in the report, but just want display the result, then you should set echo=FALSE in the code chunk instructions.

```
##        company listed
## 1 Alphabet Inc NASDAQ
## 2        Apple NASDAQ
## 3     Facebook NASDAQ
## 4       Amazon NASDAQ
## 5     Ali Baba   NYSE
## 6     Snap Inc NASDAQ
```

If you want to load a package like knitr you must load the package in the .Rmd file. In case you do not want the entire code chunk and its output from appearing in the report, use include=FALSE. The code chunk will still be evaluated however.

If you don't want a code chunk to be evaluated at all, then you use : eval = FALSE.

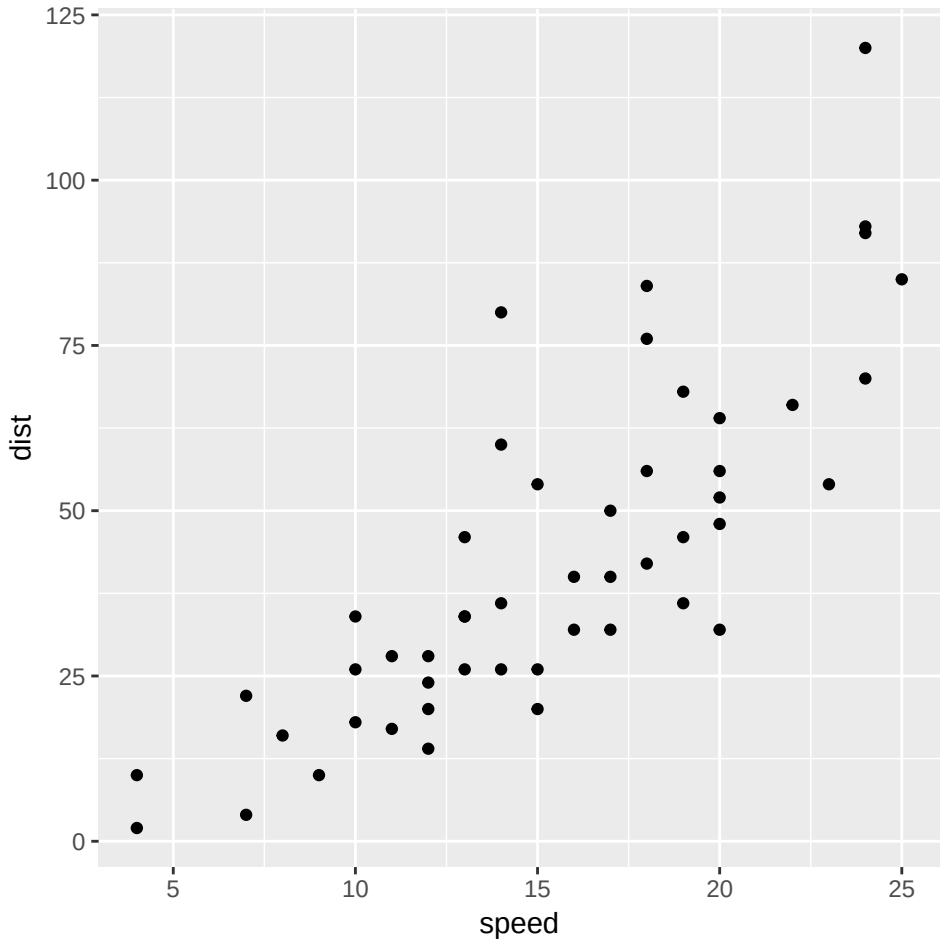
```
library(NOPACKAGE)  # there is no package called NOPACKAGE, but it does not give # an error message whe
```

More code instructions

- eval: eval=TRUE– Is the code run and the results included in the output? include: include=TRUE– Are the code and the results included in the output (the code is still run)?
- echo: echo=TRUE– Is the code displayed alongside the results?
- warning: warning=TRUE– Are warning messages displayed?
- error: error=FALSE– Are error messages displayed?
- message: message=TRUE– Are messages displayed?
- tidy: tidy=FALSE– Is the code reformatted to make it look "tidy"?
- results: results="markup"– How are results treated?
- cache: cache=FALSE– Are the results cached for future renders
- comment:comment="##"– What character are comments prefaced with?
- fig.width:, fig.width=7– What width (in inches) are the plots?
- fig.align: fig.align="left"– "left" "right" "center"

Inserting Figures

To manually set the figure dimensions, you can insert an instruction:



Inserting Tables

R Markdown can print the contents of a data frame easily by enclosing the name of the data frame in a code chunk

```
gafa_info
```

```
##          company market_val
## 1   Alphabet Inc        714
## 2          Apple        896
## 3       Facebook        518
## 4         Amazon        542
## 5       Ali Baba        477
## 6       Snap Inc         15
## 7         Criteo          2
## 8 General Motors         60
```

For nicer, better-looking tables, use the kable command from the knitr package.

```
library(knitr)
kable(gafa_info, digits = 1)
```

company	market_val
Alphabet Inc	714

company	market_val
Apple	896
Facebook	518
Amazon	542
Ali Baba	477
Snap Inc	15
Criteo	2
General Motors	60

Formatting Text

Markdown syntax can be used to change how text appears in your output file, here are a few common formatting commands

- header 1: # header 1
- header 2: ## header 2
- header 3: ### header 3
- header 4: #### header 4
- bold text: **bold** text
- italics text: *italics* text
- code text: `code` text
- LaTeX equation syntax: $A = \pi \times r^2$

You can also manually create small tables using markdown syntax.

`:----:`: Centre `:-----`: Left `-----:`: Right `------`: Auto

For example:

Teams	Wins	Loses
A	21	4
B	19	6
C	17	8

Compiling an R Markdown file

Select Knit -> Knit to HTML in RStudio.

A preview appears in the Viewer window in RStudio and the .html file is saved to your working directory. You can set your working directory location using: setwd("~/...")

You can find your working directory using: getwd()

Create your first R Markdown File

1. Create a new R Markdown file (.Rmd) in RStudio.
2. Insert a YAML Header with title, author and date of your choice at the top of your .Rmd script.
3. Display the summary of "iris" dataset in your report. (Remember iris is a dataset pre-loaded in R that has the petal and sepal length and width for 3 types of iris.) 3bis. Add a header called "Summary of Iris dataset" to the previous step.
4. Load the package tidyverse without showing the code or the warnings in the report
5. Plot the scatterplot of petal length and sepal length, differentiated by Specie.

6. Set fig.width and fig.height of your plot to 5.
7. Create a small test dataframe with variables of your choice and dipslay it in your report.
8. Hide the code of the previous question from your report.
9. Create a new repository in your Github account. Call it rprog2017-ps3-stud
10. Clone the repository in your rprog folder.
11. Create an Rproject tied to the clone on your local machine.
12. Move the Rmd file you created today into that new folder.
13. Commit the changes you made (adding .Rproj and the .Rmd file), and push the changes. Check these two files are now on your github repo.