

Report Reproducibly with



Navigate to the 03-Report folder.

Open 03-Report-Exercises.Rmd

R Markdown

How it works

```
---
title: "Test markdown"
author: "Vendula"
date: "07/03/2020"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

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. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

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

Test markdown

Vendula
07/03/2020

R Markdown

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. 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
```

Including Plots

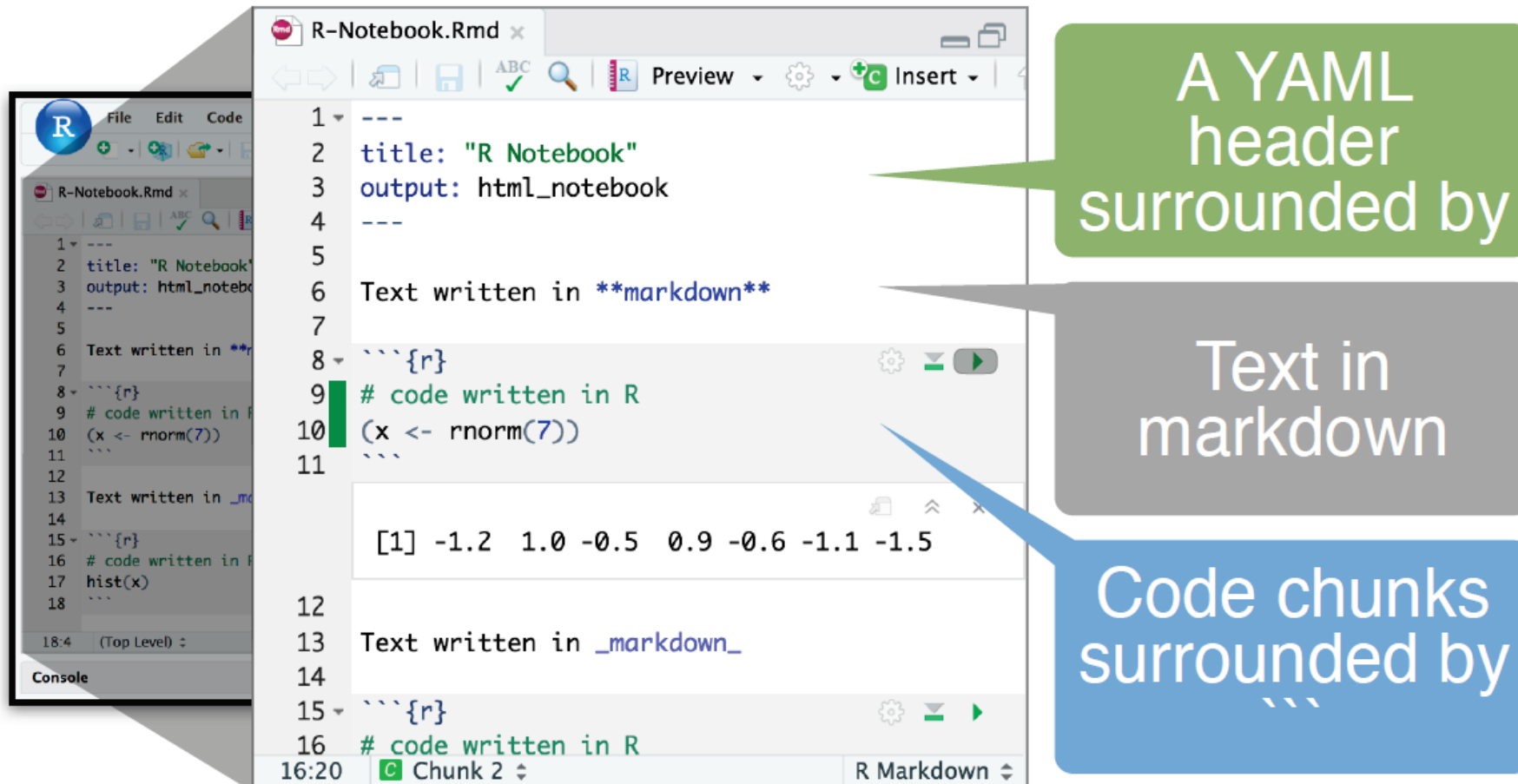
You can also embed plots, for example:



Write some code about your data, click on the button and get the report

R Markdown

Plain text file with 3 types of content:



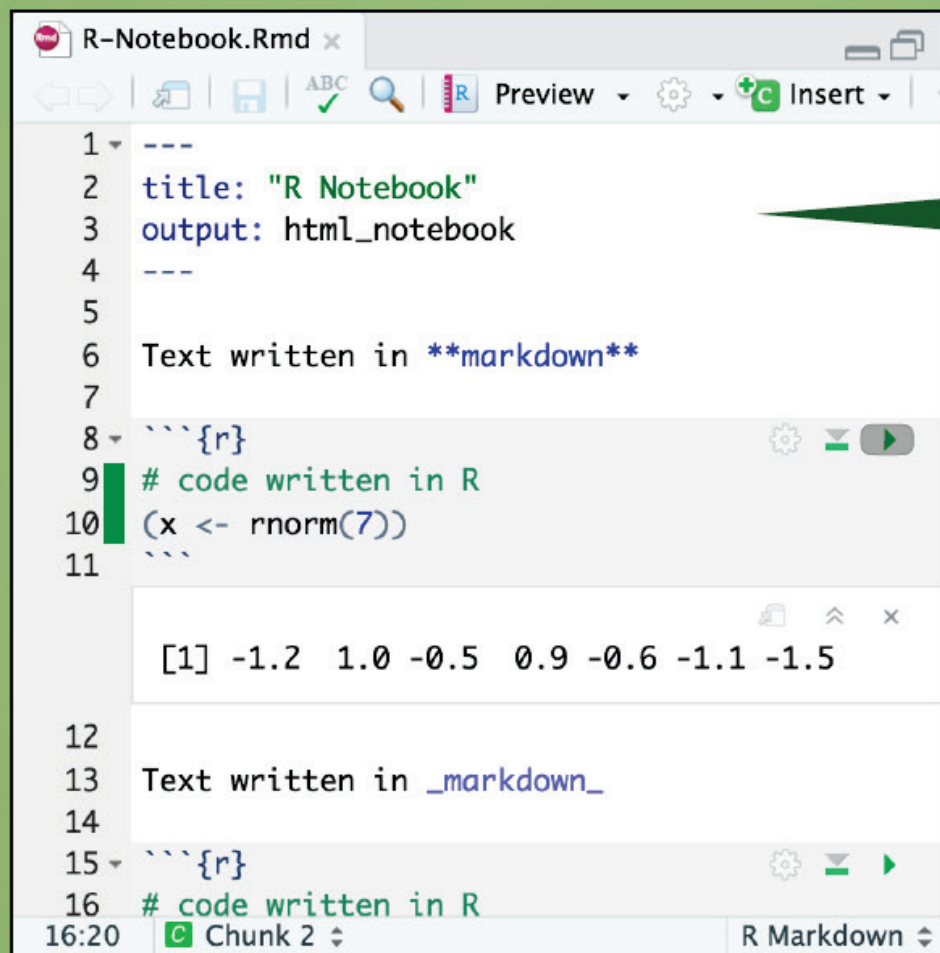
The screenshot displays the RStudio interface with an R Markdown file open. The editor shows the following content:

```
1 ---
2 title: "R Notebook"
3 output: html_notebook
4 ---
5
6 Text written in markdown
7
8 ```{r}
9 # code written in R
10 (x <- rnorm(7))
11 ```
12
13 [1] -1.2  1.0 -0.5  0.9 -0.6 -1.1 -1.5
14
15 Text written in markdown
16
17 ```{r}
18 # code written in R
19 hist(x)
20 ```
```

Three callouts highlight key features:

- A YAML header surrounded by** (green callout pointing to lines 1-4)
- Text in markdown** (grey callout pointing to line 6)
- Code chunks surrounded by** (blue callout pointing to lines 8-11)

YAML



The screenshot shows an R Notebook editor window titled "R-Notebook.Rmd". The editor displays a YAML header at the top, followed by a text block, an R code chunk, and another text block. The R code chunk contains a single line of code: `(x <- rnorm(7))`. Below the code, the output is displayed as a vector of seven random numbers. The editor interface includes a toolbar with icons for navigation, saving, and previewing, as well as a status bar at the bottom showing the current chunk and time.

```
1 ---  
2 title: "R Notebook"  
3 output: html_notebook  
4 ---  
5  
6 Text written in markdown  
7  
8 ```{r}  
9 # code written in R  
10 (x <- rnorm(7))  
11 ```  
12  
13 Text written in markdown  
14  
15 ```{r}  
16 # code written in R
```

[1] -1.2 1.0 -0.5 0.9 -0.6 -1.1 -1.5

16:20 [C] Chunk 2 R Markdown

A YAML
header
surrounded by

YAML

A section of key:value pairs
separated by dashed lines — — —

```
---  
title: "Untitled"  
author: "RStudio"  
date: "February 4, 2015"  
output: html\_document  
---  
Text of document
```



Untitled

RStudio

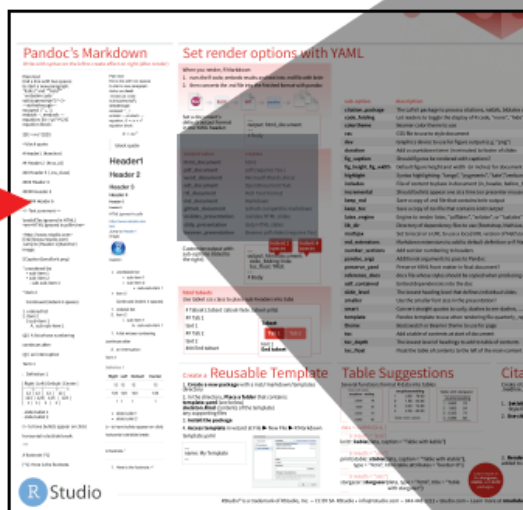
February 4, 2015

Text of document

output

The output: field sets the format of the final report

on back



output value

html_document
pdf_document
word_document
odt_document
rtf_document
md_document
github_document
ioslides_presentation
slidy_presentation
beamer_presentation

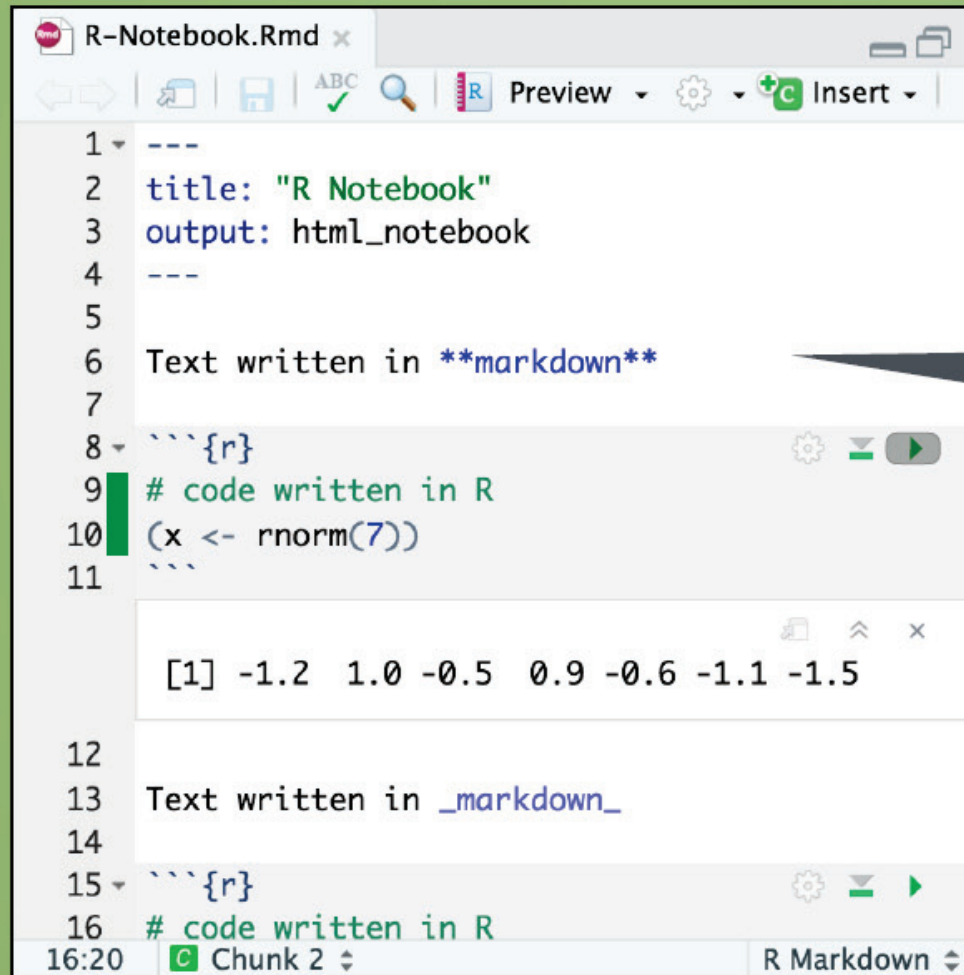
creates

html
pdf (requires Tex)
Microsoft Word (.docx)
OpenDocument Text
Rich Text Format
Markdown
Github compatible markdown
ioslides HTML slides
slidy HTML slides
Beamer pdf slides (requires Tex)

More at rmarkdown.rstudio.com/formats.html



Markdown



The screenshot shows an R Notebook editor window titled "R-Notebook.Rmd". The editor displays a document with the following content:

```
1 ---  
2 title: "R Notebook"  
3 output: html_notebook  
4 ---  
5  
6 Text written in markdown  
7  
8 ```{r}  
9 # code written in R  
10 (x <- rnorm(7))  
11 ```  
[1] -1.2  1.0 -0.5  0.9 -0.6 -1.1 -1.5  
12  
13 Text written in markdown  
14  
15 ```{r}  
16 # code written in R
```

The output of the first code chunk is displayed below the code:

```
[1] -1.2  1.0 -0.5  0.9 -0.6 -1.1 -1.5
```

The status bar at the bottom indicates "16:20" and "Chunk 2".

Text in
markdown

Headers

Use # to create headers.

Multiple #'s create lower level

```
# Header 1
## Header 2
### Header 3
#### Header 4
##### Header 5
##### Header 6
```



Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

Text

Add two spaces at the end of a line to start a new line

Text is rendered as plain text.
Surround text with `_`, `**`, or ``` to

Text ○
italics
bold
`code`



Text
italics
bold
`code`

Lists

Use asterisks to make bullet points.

Use numbers to make numbered lists.

Bullets

- * bullet 1
- * bullet 2

Numbered list

1. item 1
2. item 2



Bullets

- bullet 1
- bullet 2

Numbered list

1. item 1
2. item 2

Hyperlinks

Use brackets to denote a link.
Place the URL in parentheses.

This is a
[link](www.git.com).



This is a [link](#).

Images

Use a link preceded by an ! to insert an image.

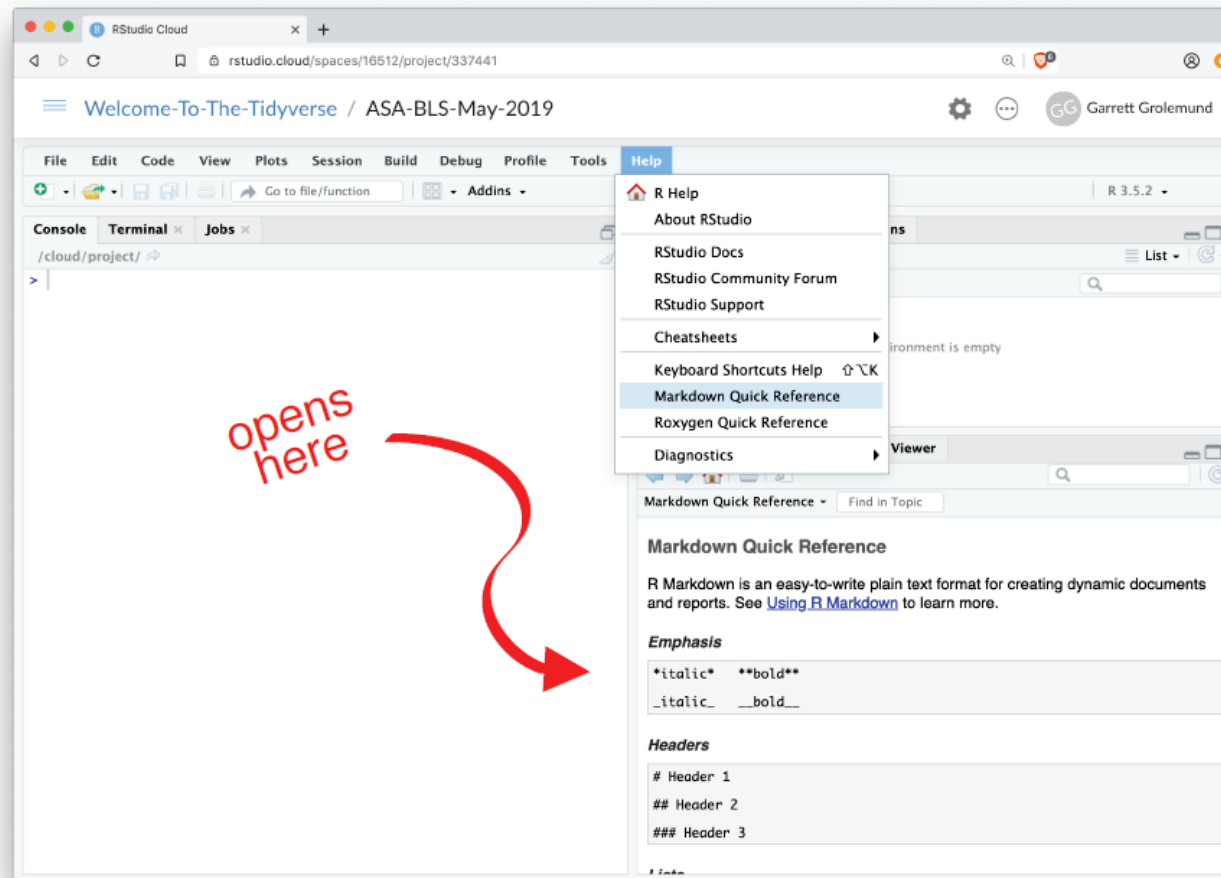
The link text should be a URL (if the image is hosted online), or a file path (if the image is saved as a file)

```
  
The RStudio logo.
```

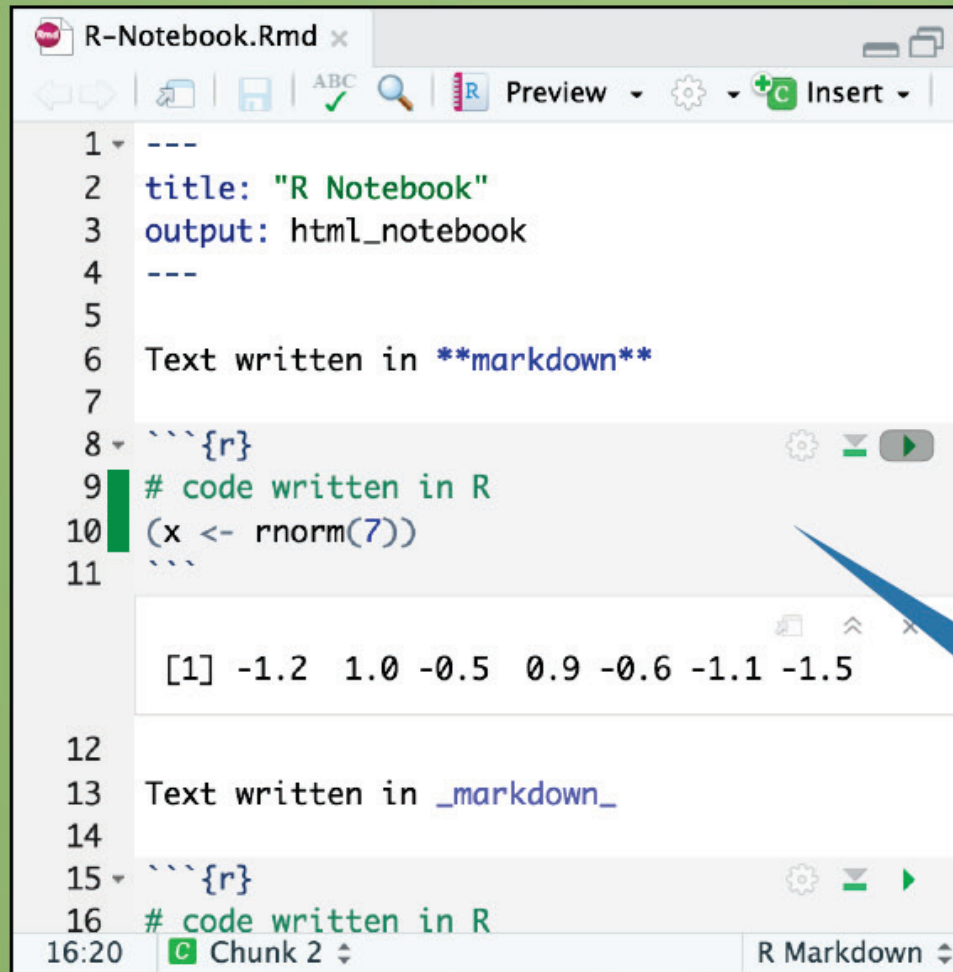


IDE Reference

Go to Help > Markdown Quick Reference



Code



The screenshot shows an R Notebook editor window titled "R-Notebook.Rmd". The editor has a toolbar with icons for navigation, saving, and previewing. The code is organized into numbered lines. Lines 1-4 define the notebook title and output format. Line 6 contains a text block. Lines 8-11 show an R code chunk enclosed in triple backticks, with a comment and a function call. Below the code, the output of the function is displayed in a separate box. Lines 12-14 contain another text block. Lines 15-16 show another R code chunk, partially visible. The status bar at the bottom indicates the current chunk is "Chunk 2" and the editor is in "R Markdown" mode.

```
1 ---  
2 title: "R Notebook"  
3 output: html_notebook  
4 ---  
5  
6 Text written in markdown  
7  
8 ```{r}  
9 # code written in R  
10 (x <- rnorm(7))  
11 ```  
  
[1] -1.2  1.0 -0.5  0.9 -0.6 -1.1 -1.5  
  
12  
13 Text written in markdown  
14  
15 ```{r}  
16 # code written in R
```

16:20 [C] Chunk 2 R Markdown

Code chunks
surrounded by
```

# Code chunks

Insert a chunk of R code with

```
```{r}  
# some code  
```
```

When you render the report, R Markdown will run the code and include its results. R Markdown will also remove the ````{r}` and `````.

# Code chunks

Insert a chunk of R code with

```
`` `{r}
some code
`` `
```

 +  +  (Mac)

 +  +  (PC)

# chunk options

By default, R Markdown includes both the code and its results

```
Here's some code
```${r}  
dim(iris)  
```
```



Here's some code

```
dim(iris)
```

```
[1] 150 5
```

# chunk options

- write in the brackets after letter "r" or
- as general statement e.g. `knitr::opts_chunk$set(echo = TRUE)` on the beginning

```
```{r setup, include=FALSE}  
knitr::opts_chunk$set(  
  echo = TRUE,  
  fig.height = 15,  
  fig.width = 14,  
  message = TRUE,  
  warning = TRUE,  
  comment = "",  
  results = "asis",  
  suppressMessage = FALSE  
)  
  
library(tidyverse)  
```
```

# echo

Add options in the brackets after r.  
`echo = FALSE` hides the code.

```
Here's some code
```${r echo=FALSE}  
dim(iris)  
```
```



Here's some code

```
[1] 150 5
```

Very useful  
for plots





# eval

`eval = FALSE` prevents the code from being run. As a result, no results will be displayed with the code.

```
Here's some code
```${r eval=FALSE}  
dim(iris)  
```
```



```
Here's some code

dim(iris)
```

# include

`include = FALSE` runs the code, but prevents both the code and the results from appearing (e.g. to setup).

```
Here's some code
```${r include=FALSE}  
dim(iris)  
```
```



Here's some code

# fig.height, fig.width

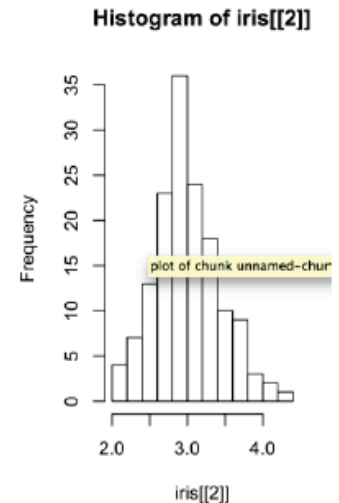
Specify the dimension of plots (in inches) with `fig.width` and `fig.height`. Separate multiple arguments with commas.

Here's a plot

```
```{r echo=FALSE, fig.width=3, fig.height=5}  
hist(iris[[2]])  
```
```



Here's a plot



# Your Turn

- INSERT code for boxplot graph into for loop
- Write a sentence below graphs containing words with bold, italics, and underlined formatting
- Change size a plot to 13 inches wide and 15 inches high
- Test different Chunk options (echo, message, warnigs for final document

05:00

# Report Reproducibly with



# Thank you