

**Understanding and creating code notebooks**

# Extensive data science with R scripts become challenging quickly

```
RStudio Source Editor
E4_R_ggplots.R
Source on Save Run Source
1 #####
2 # Copyright (c) 2017. All rights reserved. See the file LICENSE for
3 # license terms.
4 #####
5
6 # File: E4_R_ggplots.R
7 # Proj: R Workshop
8 # Desc: A non-technical introduction to R,
9 #       Exercises Part 4
10 # Auth: ML, Jvo
11 # Date: 2017
12 #####
13
14 # Set working directory ###
15 # setwd("")
16 setwd("~/Documents/Rintro2018/Code_LE4")
17
18
19 # Install and load packages ###
20 #install.packages("data.table")
21 library(ggplot2)
22 library(data.table)
23
24 # Read in data ###
25 myData <- fread("transactions.csv", sep = ",", nrow=10000)
26
27 # Take a sample if needed
28 myData <- myData[sample(nrow(myData), 10000), ]
29
30 # Part 1: ggplot basics ###
31 # -----
32
33 # Task 1: Create a histogram of PurchAmount with ggplot
34 ggplot(myData, aes(PurchAmount)) + geom_histogram() +
35 ggtitle("Histogram of Purchase Amount") +
36 xlab("Purchase Amount")
37
38 # Avoid warning using binwidth, which specifies bin range
39 ggplot(myData, aes(PurchAmount)) + geom_histogram(binwidth = 40) +
40 ggtitle("Histogram of Purchase Amount") +
41 xlab("Purchase Amount")
42
43 # Task 2: Create a scatter plot with smooth curve for PurchAmount (x) and Cost (y) ###
44 ggplot(myData, aes(PurchAmount, Cost)) + geom_point() +
45 xlab("Purchase Amount") + ylab("Cost") + ggtitle("Cost by Purchase Amount")
46
47 # Add plots next to each other
48 # Install additional package gridExtra to adjust the layout
49
50
```

1  
Comments in your script tend to be terse or non-existent

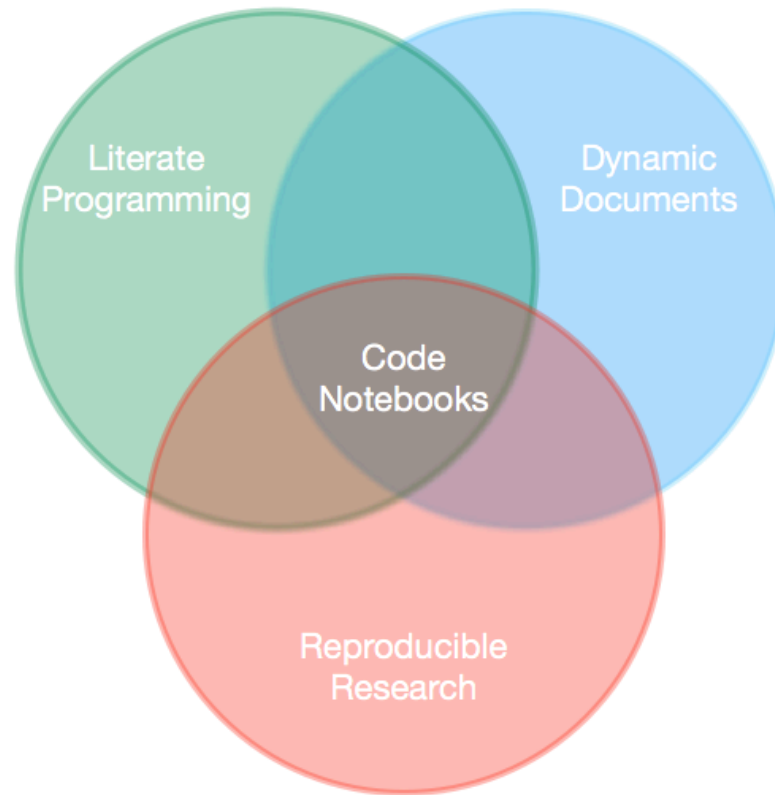
3  
Coworkers spend a lot of time reading your comments

2  
Intermediate results are not immediately accessible

4  
You struggle to find the right plot in your output folder, because there are so many

Scripts for data science can become messy quickly.

# Code notebooks are a tool to report your code and results and make them publicly available



# Code notebooks help you to:

- Divide your code into manageable chunks.
  - Execute small chunks of code one by one.
  - Evaluate and store preliminary results.
- Take notes along the process in rich text format (paragraphs, equations, tables, ... ).
- Present and discuss your results and code chunks with colleagues.



# What do you get out of code notebooks?

## Example: R Notebook

The screenshot displays an R Notebook interface with the following components:

- Code Editor (Left):** Contains R code for document metadata, Shiny integration, and a histogram plot. The code includes comments and function calls like `selectInput` and `sliderInput`.
- Output Panel (Right):** Displays the rendered HTML output, including a title, date, and a histogram titled "Geyser eruption duration" with a density curve.
- Interactive Elements:** A "Number of bins" dropdown menu and a "Bandwidth adjustment" slider are visible above the histogram.
- Annotations:** Six yellow callout boxes highlight key features:
  - Different export formats, e.g. HTML or PDF
  - Code can be included in the output document
  - Integration of other languages, e.g. SQL and Python
  - Exporting your document will produce immediate results of your calculations
  - Publish your work on rpubs.com or Rstudio Connect
  - Easy interaction of (interactive) graphs, images and tables

# Notebooks versus regular programming scripts

Activity	Scripts	Notebooks	
Building a narrative	<b>Comments</b> are available to describe code.	<b>Rich text</b> is available to describe the process and conclusion	<div>1 Allows very descriptive and reproducible documentation</div>
Manage output	Output is available in the console or environment.	<b>Output is embedded</b> in a single document. Code and output may be divided into separate <b>code chunks</b> .	<div>2 Immediate update of output and report</div>
Creating a final report	<b>Creating a report is a separate</b> and time-consuming <b>step</b> .	<b>Instant report possible.</b> Reports are publishable as HTML, PDF, ...)	<div>3 Easy sharing possible</div>

# Notebooks versus regular programming scripts

Activity	Scripts	Notebooks	
Building a narrative	<b>Comments</b> are available to describe code.	<b>Rich text</b> is available to describe the process and conclusion	Allows very descriptive and reproducible documentation <sup>1</sup>
Manage output	Output is available in the console or environment.	<b>Output is embedded</b> in a single document. Code and output may be divided into separate <b>code chunks</b> .	Immediate update of output and report <sup>2</sup>
Creating a final report	<b>Creating a report is a separate</b> and time-consuming <b>step</b> .	<b>Instant report possible.</b> Reports are publishable as HTML, PDF, ...)	Easy sharing possible <sup>3</sup>

# Notebooks versus regular programming scripts

Activity	Scripts	Notebooks
Building a narrative	<b>Comments</b> are available to describe code.	<b>Rich text</b> is available to describe the process and conclusion
Manage output	Output is available in the console or environment.	<b>Output is embedded</b> in a single document. Code and output may be divided into separate <b>code chunks</b> .
Creating a final report	<b>Creating a report is a separate</b> and time-consuming <b>step</b> .	<b>Instant report possible.</b> Reports are publishable as HTML, PDF, ...)

Allows very descriptive and reproducible documentation <sup>1</sup>

Immediate update of output and report <sup>2</sup>

Easy sharing possible <sup>3</sup>



# Notebooks versus regular programming scripts

Activity	Scripts	Notebooks
Building a narrative	<b>Comments</b> are available to describe code.	<b>Rich text</b> is available to describe the process and conclusion
Manage output	Output is available in the console or environment.	<b>Output is embedded</b> in a single document. Code and output may be divided into separate <b>code chunks</b> .
Creating a final report	Creating a report is a <b>separate</b> and time-consuming <b>step</b> .	<b>Instant report possible.</b> Reports are publishable as HTML, PDF, ...)

Allows very descriptive and reproducible documentation <sup>1</sup>

Immediate update of output and report <sup>2</sup>

Easy sharing possible <sup>3</sup>

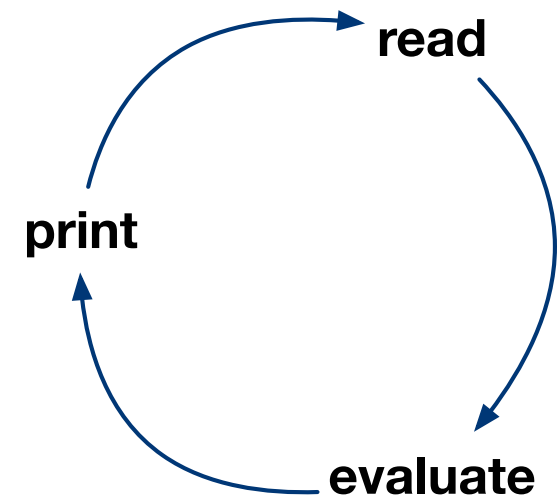
# Notebooks are based on the read-eval-print loop (REPL) technique

The notebook user-interface is developed on the **concept of interactive computing**:

1. User input
2. Input is evaluated
3. Result is printed and notebook is ready for next input

There are **two main features**:

- Store the last output for reuse
- Ability to save input and later reload and re-evaluate



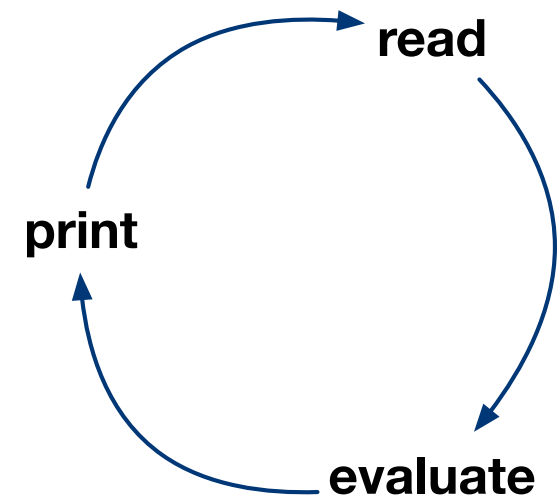
# Notebooks are based on the read-eval-print loop (REPL) technique

The notebook user-interface is developed on the **concept of interactive computing**:

1. User input
2. Input is evaluated
3. Result is printed and notebook is ready for next input

There are **two main features**:

- Store the last output for reuse
- Ability to save input and later reload and re-evaluate



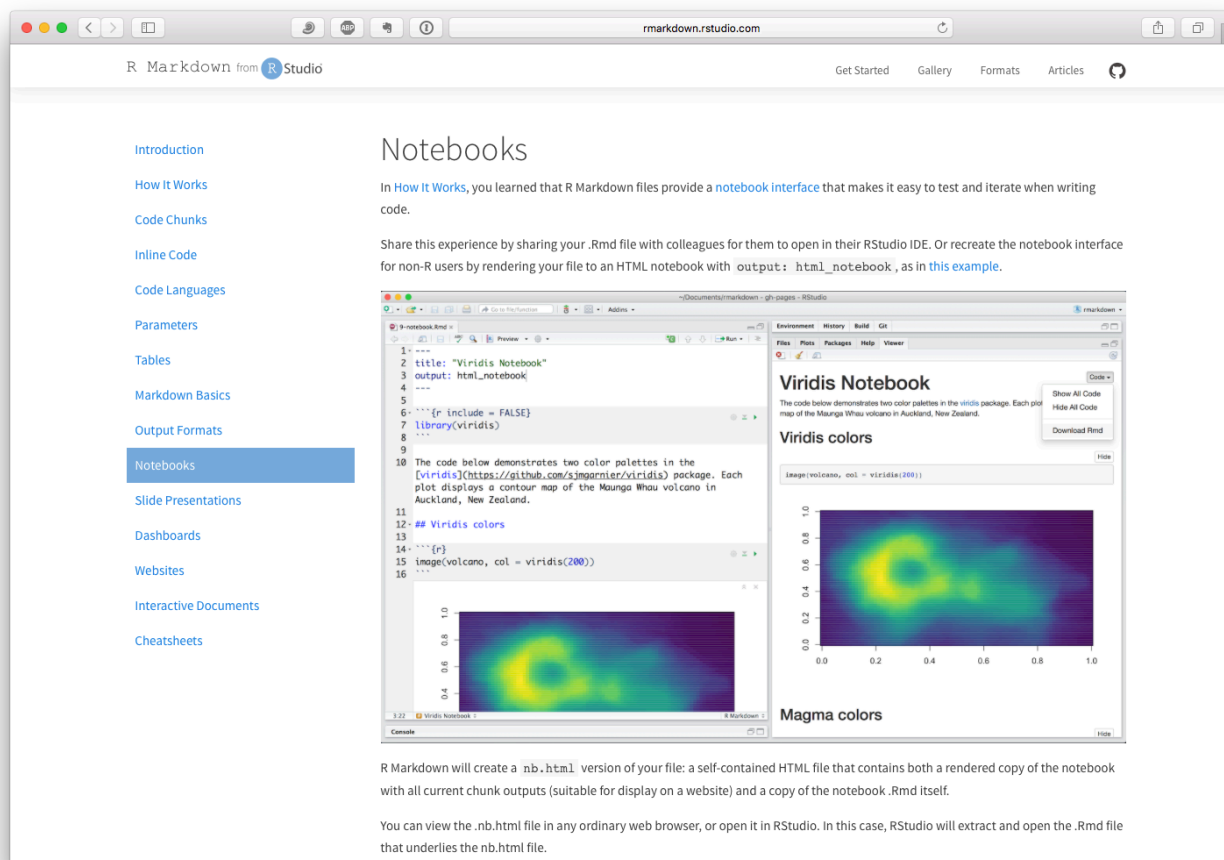
# There are many different notebooks, which mostly support multiple programming languages

- Jupyter (e.g. used by Google Colab; R, Python; around 40 different languages)
- R Notebooks (R, Python; around 8 different languages)
- Beaker (R, Python; around 17 different languages)
- Zeppelin (R, Python, around 20 different languages)



- Which one to pick depends on the requirements of the programming project and your personal preference. We pick to present R notebooks due to its integration into R Studio.

# Using R Notebooks in R Studio is very convenient



The screenshot shows the R Markdown website with the 'Notebooks' section highlighted in the sidebar. The main content area is titled 'Notebooks' and contains the following text:

In [How It Works](#), you learned that R Markdown files provide a [notebook interface](#) that makes it easy to test and iterate when writing code.

Share this experience by sharing your .Rmd file with colleagues for them to open in their RStudio IDE. Or recreate the notebook interface for non-R users by rendering your file to an HTML notebook with output: `html_notebook`, as in [this example](#).

The code chunk example shows the following R code:

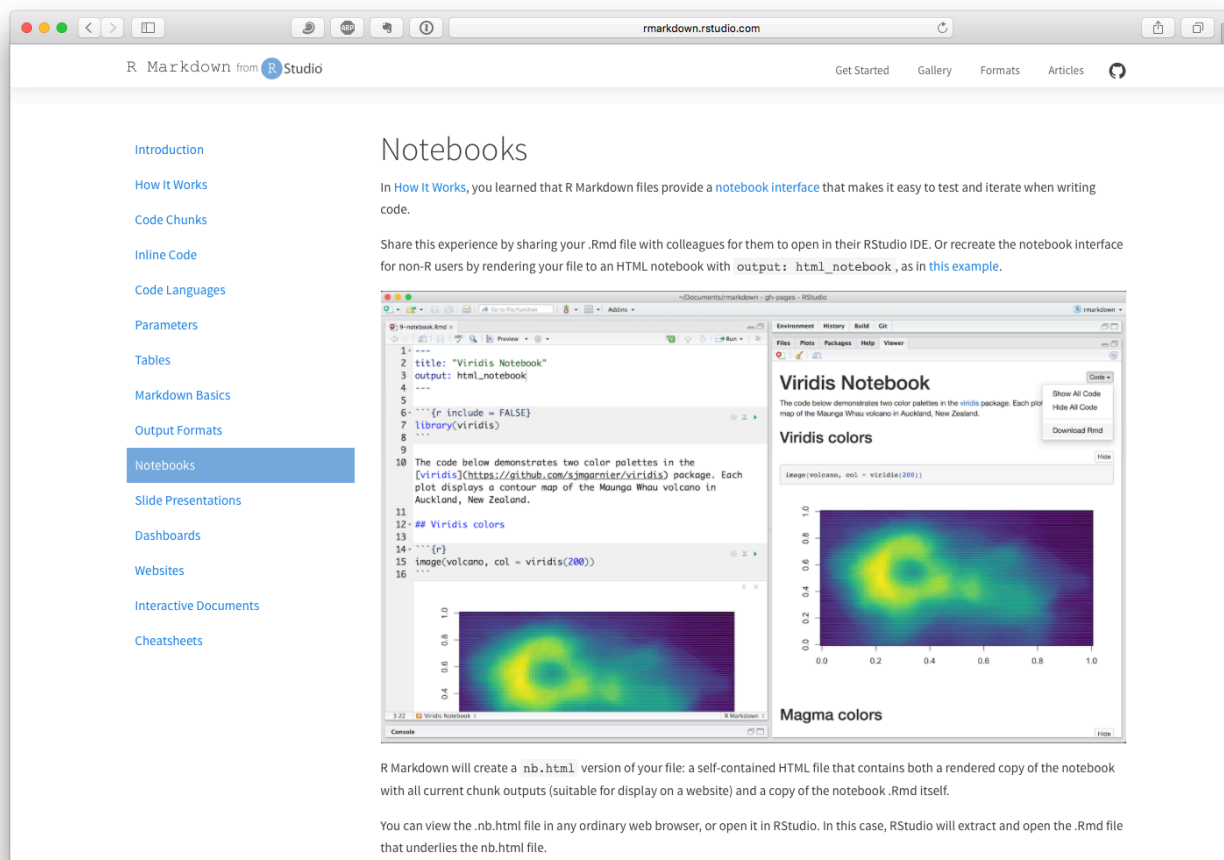
```
1 ---
2 title: "Viridis Notebook"
3 output: html_notebook
4 ---
5
6 {r include = FALSE}
7 library(viridis)
8
9
10 The code below demonstrates two color palettes in the
11 [viridis](https://github.com/jrnold/viridis) package. Each
12 plot displays a contour map of the Maunga Whau volcano in
13 Auckland, New Zealand.
14
15 ## Viridis colors
16
17 {r}
18 image(volcano, col = viridis(200))
19
20
```

The output of the code chunk shows two contour maps of the Maunga Whau volcano. The first map is titled 'Viridis Notebook' and the second map is titled 'Viridis colors'. Both maps use the viridis color palette.

R Markdown will create a `.nb.html` version of your file: a self-contained HTML file that contains both a rendered copy of the notebook with all current chunk outputs (suitable for display on a website) and a copy of the notebook .Rmd itself.

You can view the `.nb.html` file in any ordinary web browser, or open it in RStudio. In this case, RStudio will extract and open the .Rmd file that underlies the `.nb.html` file.

# Using R Notebooks in R Studio is very convenient



R Markdown from R Studio

Get Started Gallery Formats Articles

Introduction  
How It Works  
Code Chunks  
Inline Code  
Code Languages  
Parameters  
Tables  
Markdown Basics  
Output Formats  
**Notebooks**  
Slide Presentations  
Dashboards  
Websites  
Interactive Documents  
Cheatsheets

## Notebooks

In [How It Works](#), you learned that R Markdown files provide a [notebook interface](#) that makes it easy to test and iterate when writing code.

Share this experience by sharing your .Rmd file with colleagues for them to open in their RStudio IDE. Or recreate the notebook interface for non-R users by rendering your file to an HTML notebook with `output: html_notebook`, as in [this example](#).

```
1 ---
2 title: "Viridis Notebook"
3 output: html_notebook
4 ---
5
6 {r include = FALSE}
7 library(viridis)
8
9
10 The code below demonstrates two color palettes in the
11 [viridis](https://github.com/sjmgarnier/viridis) package. Each
12 plot displays a contour map of the Maunga Whau volcano in
13 Auckland, New Zealand.
14
15 image(volcano, col = viridis(200))
16
```

Viridis Notebook

The code below demonstrates two color palettes in the `viridis` package. Each plot displays a contour map of the Maunga Whau volcano in Auckland, New Zealand.

Viridis colors

`image(volcano, col = viridis(200))`

Magma colors

R Markdown will create a `.nb.html` version of your file: a self-contained HTML file that contains both a rendered copy of the notebook with all current chunk outputs (suitable for display on a website) and a copy of the notebook .Rmd itself.

You can view the `.nb.html` file in any ordinary web browser, or open it in RStudio. In this case, RStudio will extract and open the .Rmd file that underlies the `nb.html` file.

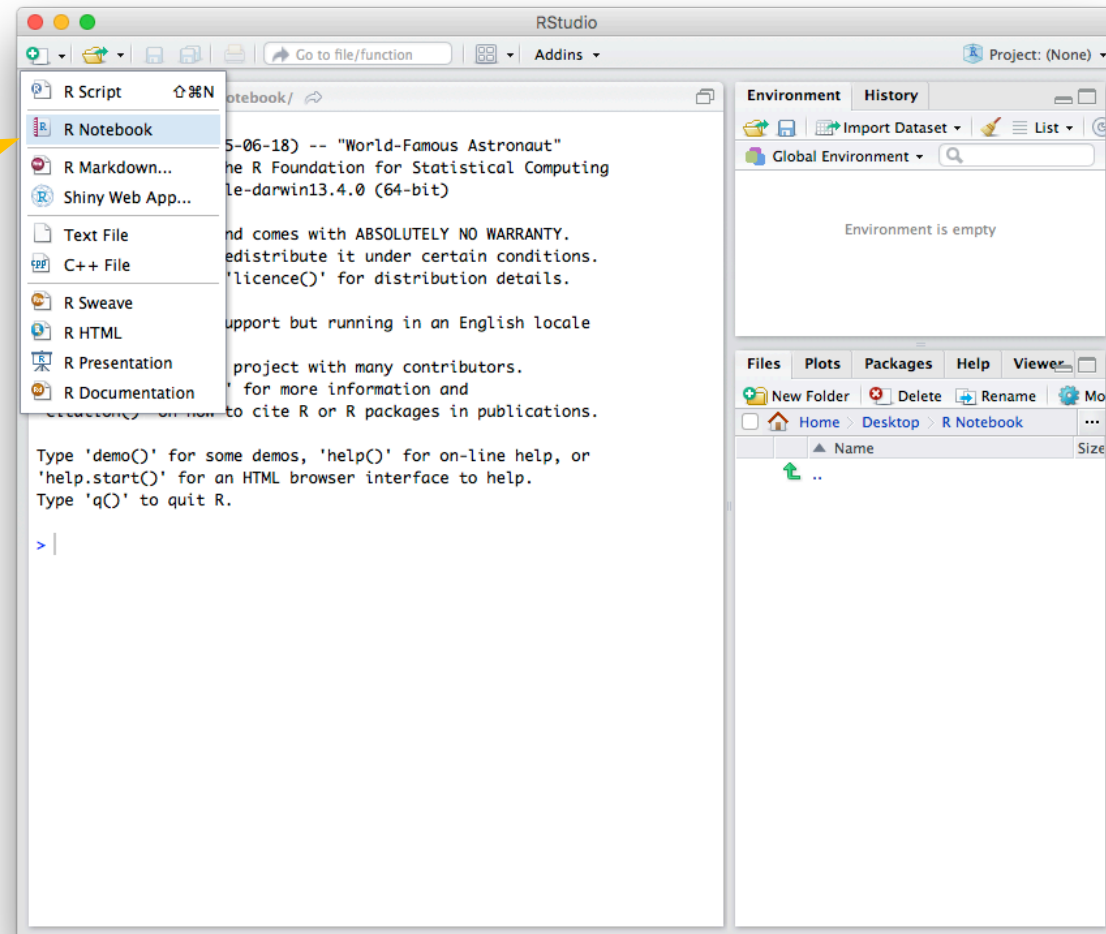
# Create your first R Notebook

## Steps

1. **Create a new R Notebook**
2. Create Content
  - Add text elements as Markdown syntax
  - Add code elements in any supported program language
  - Use LaTeX in your R Notebook

# Create your first R Notebook

Click on "New" and select  
"R Notebook"





# Main components of an R Notebook

The image shows the RStudio interface with an R Notebook open. The notebook is titled "R Notebook" and contains R code and text. The interface is divided into several panes: Source (left), Environment (top right), Files (bottom right), and Console (bottom left). The notebook content is displayed in the center pane.

Numbered callouts highlight the following components:

- 1** View the notebook or export it (pdf, html, Word)
- 2** Notebook settings
- 3** Insert code chunks
- 4** Execute code chunks
- 5** Notebook

The notebook content includes the following text and code:

```

1 title: "R Notebook"
2
3 This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.
4
5 Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing Cmd+Shift+Enter.
6
7
8
9 ```{r}
10 plot(cars)
11 ```
12
13 Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing Cmd+Option+I.
14
15 When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press Cmd+Shift+K to preview the HTML file).
16
17
  
```

The notebook output shows the plot of the 'cars' dataset.

# Elements of an R Notebook

There are three types of content in an R Notebook:

- **YAML header  
(basic information)**

```
1 ---  
2 title: "R Notebook"  
3 output: html_notebook  
4 ---
```

1 The YAML ("Yet another Markup Language") header is optional and surrounded by ---

- **Inline text**

```
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax  
15 for authoring HTML, PDF, and MS Word documents. For more details on using  
R Markdown see <http://rmarkdown.rstudio.com>.
```

2 Inline text helps you structuring your document

- **R code chunks**

```
18 ```{r cars}  
19 summary(cars)  
20 ```
```

3 Code chunks include your R calculations and are surrounded by ```