01 R, RStudio and Rmarkdown

Soyoung Park

Pusan National University
Department of Statistics

Section 1

Getting started with R and RStudio

Why R?

R is was developed by statisticians as an interactive environment for data analysis. You can read the full history in the paper A Brief History of S[1].

1. R is free and open source[2].
2. You will soon learn, the ability to quickly explore data
3. Scripts and data objects can be shared seamlessly across platforms.
4. There is a large, growing, and active community of R users and, as a result, there are numerous resources for learning and asking questions[3] [4] [5].
5. It is easy for others to contribute add-ons which enables developers to share software implementations of new data science methodologies.

[1] https://pdfs.semanticscholar.org/9b48/46f192aa37ca122cfabb1ed1b59866d8bfda.pdf

[2] https://opensource.org/history

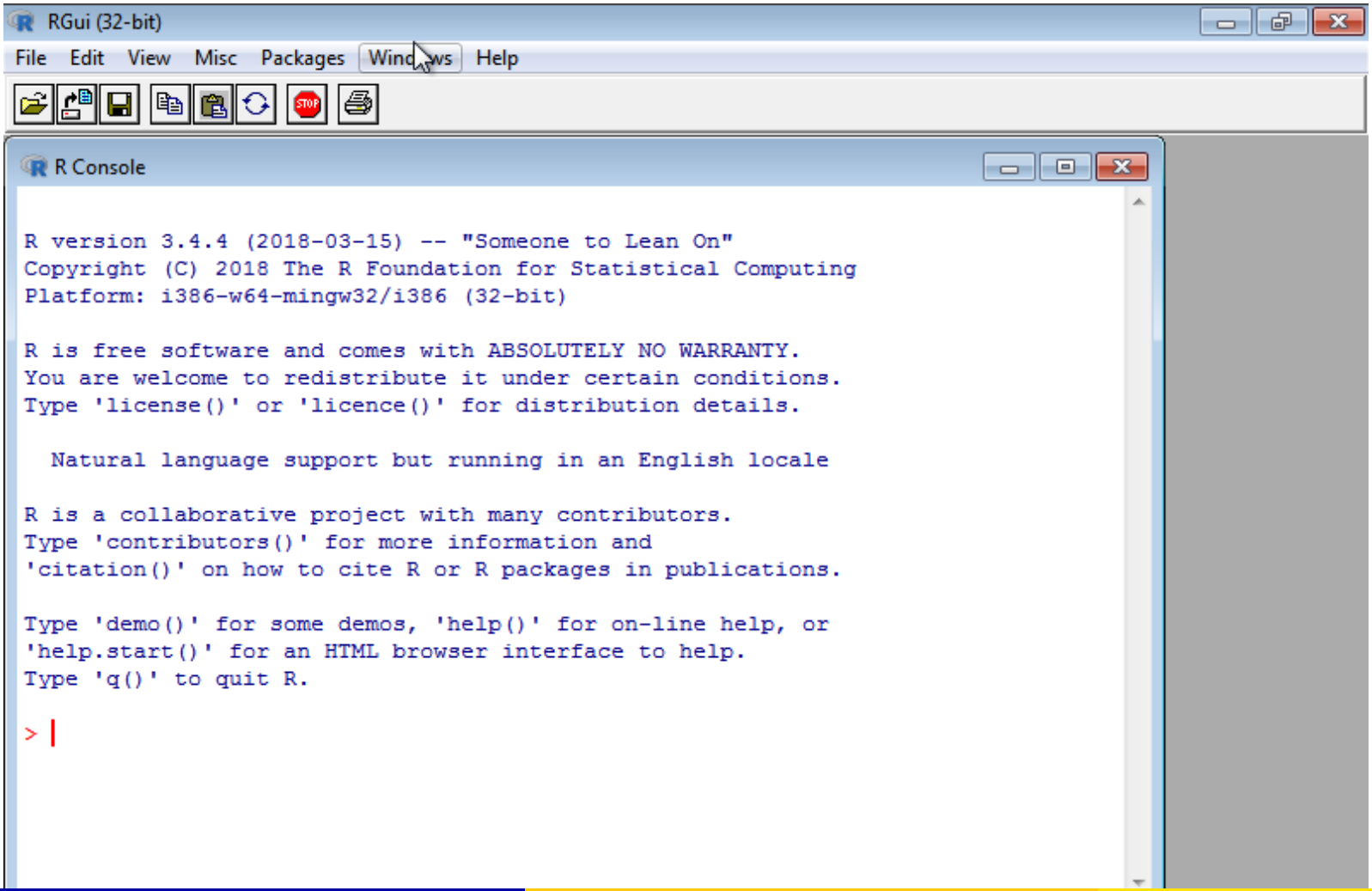
[3] https://stats.stackexchange.com/questions/138/free-resources-for-learning-r

[4] https://www.r-project.org/help.html

[5] https://stackoverflow.com/documentation/r/topics

The R console

Interactive data analysis usually occurs on the *R console* that executes commands as you type them. The console looks something like this:



The R console

As a quick example, try using the console to calculate a 15% tip on a meal that cost $19.71:

```
0.15 * 19.71
```

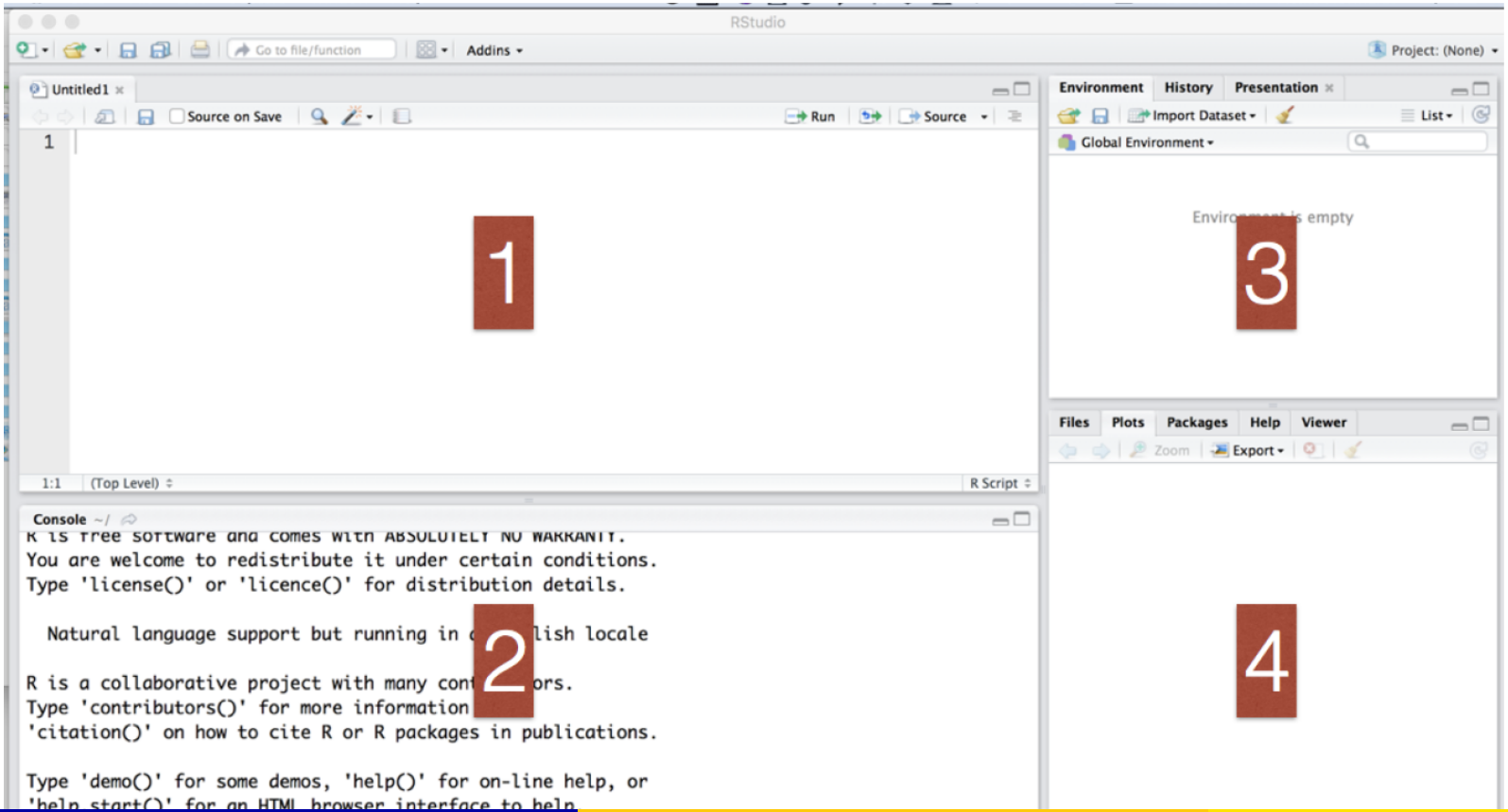
```
## [1] 2.9565
```

Note that in this book, grey boxes are used to show R code typed into the R console. The symbol #> is used to denote what the R console outputs.

Scripts

Save your work as scripts! You can edit and save these scripts using a text editor.

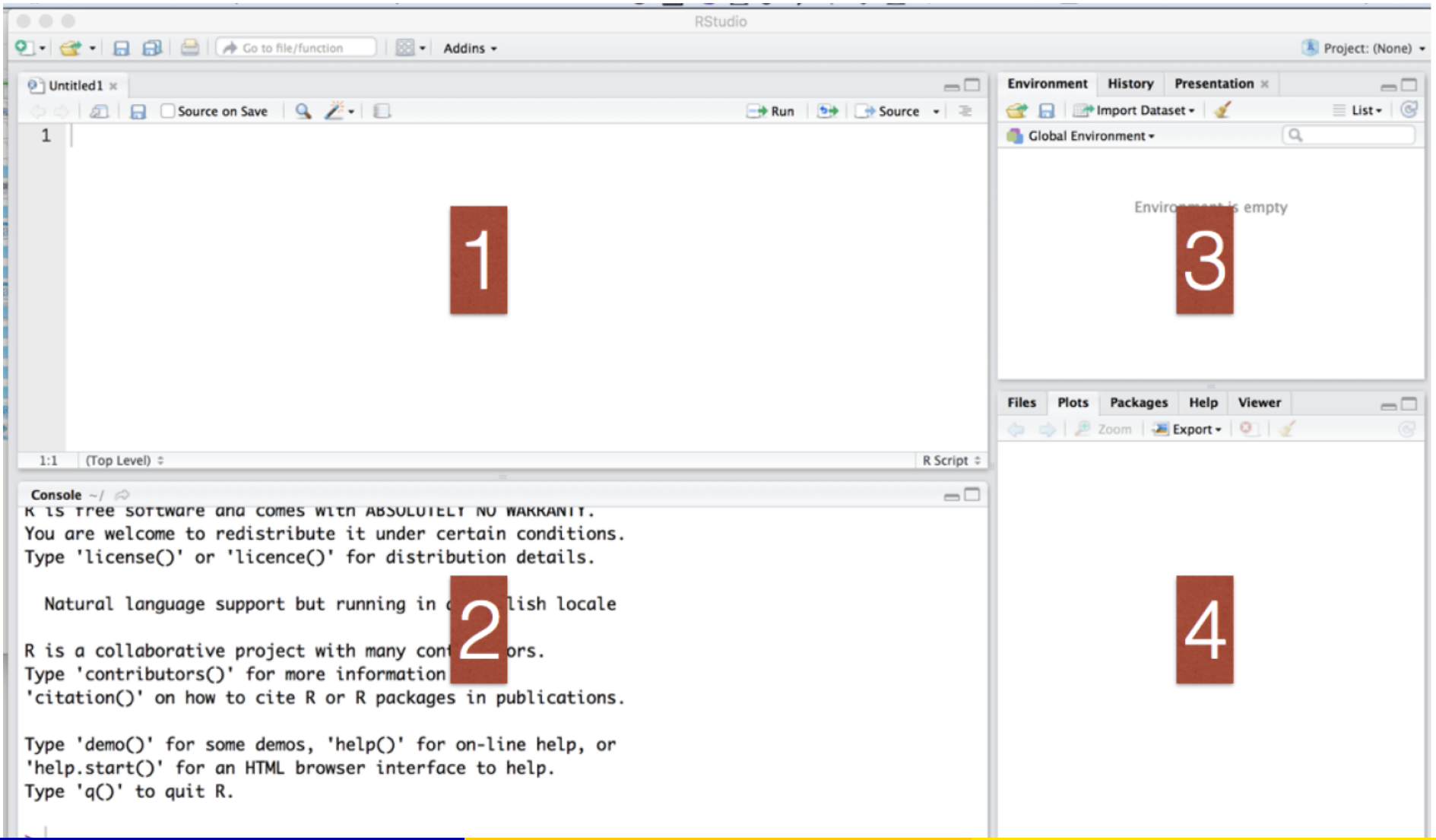
RStudio includes an editor with many R specific features, a console to execute your code, and other useful panes, including one to show figures.



R studio is. . .

- From Julie Lowndes:
  +If R were an airplane, RStudio would be the airport, providing many, many supporting services that make it easier for you, the pilot, to take off and go to awesome places. Sure, you can fly an airplane without an airport, but having those runways and supporting infrastructure is a game-changer.

The R studio IDE (Intergrated Development Environment)



The R studio IDE

1. Source editor: Docking station for multiple files, Useful shortcuts ("Knit"), Highlighting/Tab-completion, Code-checking (R, HTML, JS), Debugging features
2. Console window: Highlighting/Tab-completion, Search recent commands
3. Environment pane: Tools for package development, git, etc
4. Other tabs/panes: Graphics, R documentation, File system navigation/access

Protip: save yourself the headaches figuring out file paths and instead work in an RStudio R project (popup menu at the top left)

Installing R packages

R makes it very easy to install packages from within R. For example, to install the **tidyverse** package, which we use to share datasets and code related to this book, you would type:

```
install.packages("dslabs")
```

In RStudio, you can navigate to the *Tools* tab and select install packages. We can then load the package into our R sessions using the `library` function:

```
library(dslabs)
```

tidyverse

We can install more than one package at once by feeding a character vector to this function:

```
install.packages(c("tidyverse"))
installed.packages()
```

Note that installing **tidyverse** actually installs several packages.

This commonly occurs when a package has *dependencies*, or uses functions from other packages.

When you load a package using `library`, you also load its dependencies.

Markdown (.rmd)

- Markdown is a particular type of markup language.
- Markup languages are designed to produce documents from plain text.
- Some of you may be familiar with LaTeX. This is another (less human friendly) markup language for creating pdf documents.
- LaTeX gives you much greater control, but it is restricted to pdf and has a much greater learning curve.
- Markdown is becoming a standard. Many websites will generate HTML from Markdown (e.g. GitHub, Stack Overflow, reddit, . . .).

Markdown is easy

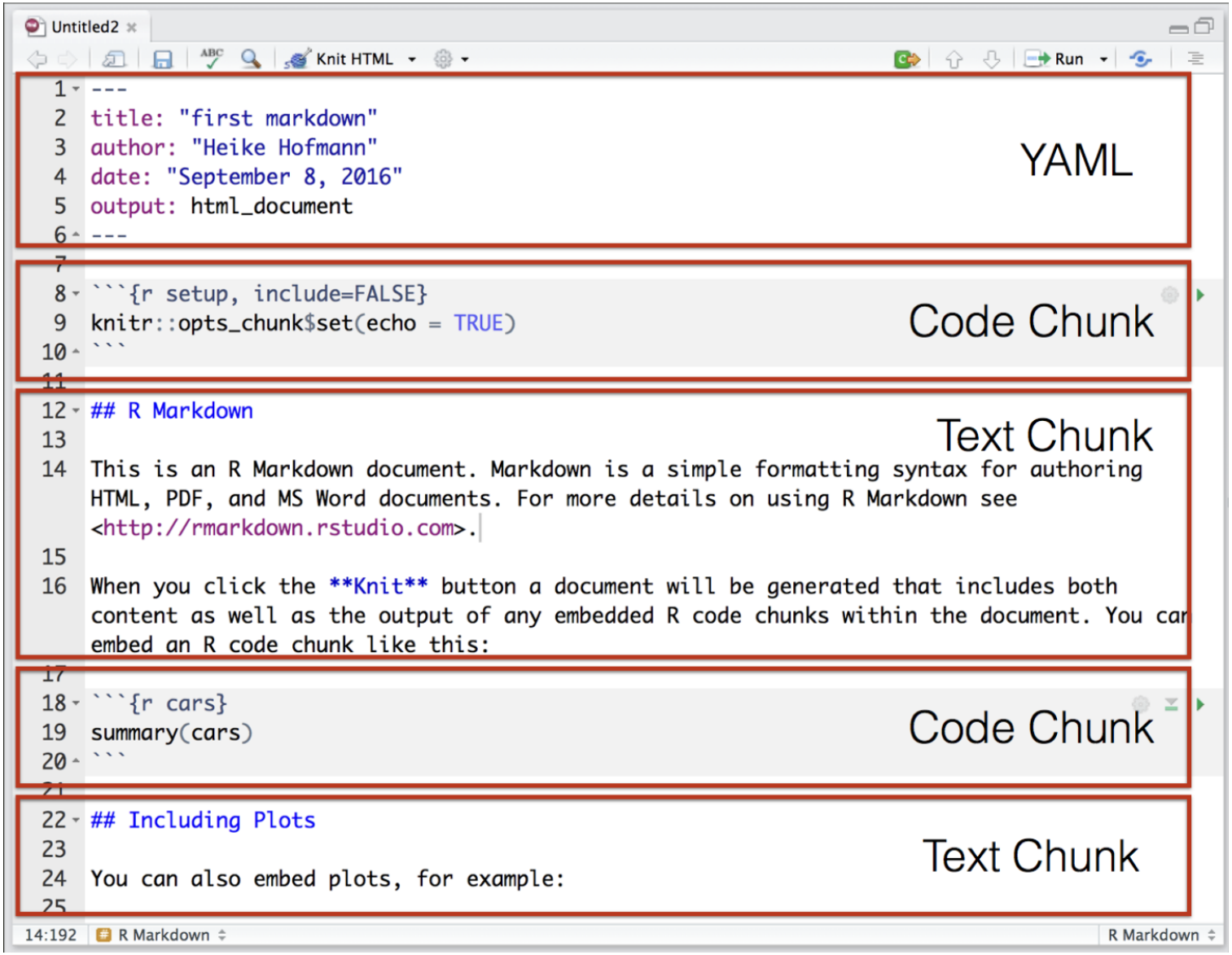
```
 *italic*
 **bold**

# Header 1
## Header 2
### Header 3

- List item 1
- List item 2
    - item 2a
    - item 2b

1. Numbered list item 1
1. Numbered list item 2
    - item 2a
    - item 2b
```

Have a look at RStudio's RMarkdown cheat sheet

What is RMarkdown?

- . . . an authoring format that enables easy creation of dynamic documents, presentations, and reports from R.
- it combines the core syntax of markdown with embedded R code chunks that are run so their output can be included in the final document.
- R Markdown documents are fully reproducible (they are automatically regenerated whenever underlying R code or data changes).

RMarkdown


Untitled2
Knit HTML
Run
---
title: "first markdown"
author: "Heike Hofmann"
date: "September 8, 2016"
output: html_document

YAML
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
Code Chunk
R Markdown
Text Chunk
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)
```
Code Chunk
Including Plots
Text Chunk
You can also embed plots, for example:
14:192
R Markdown
R Markdown

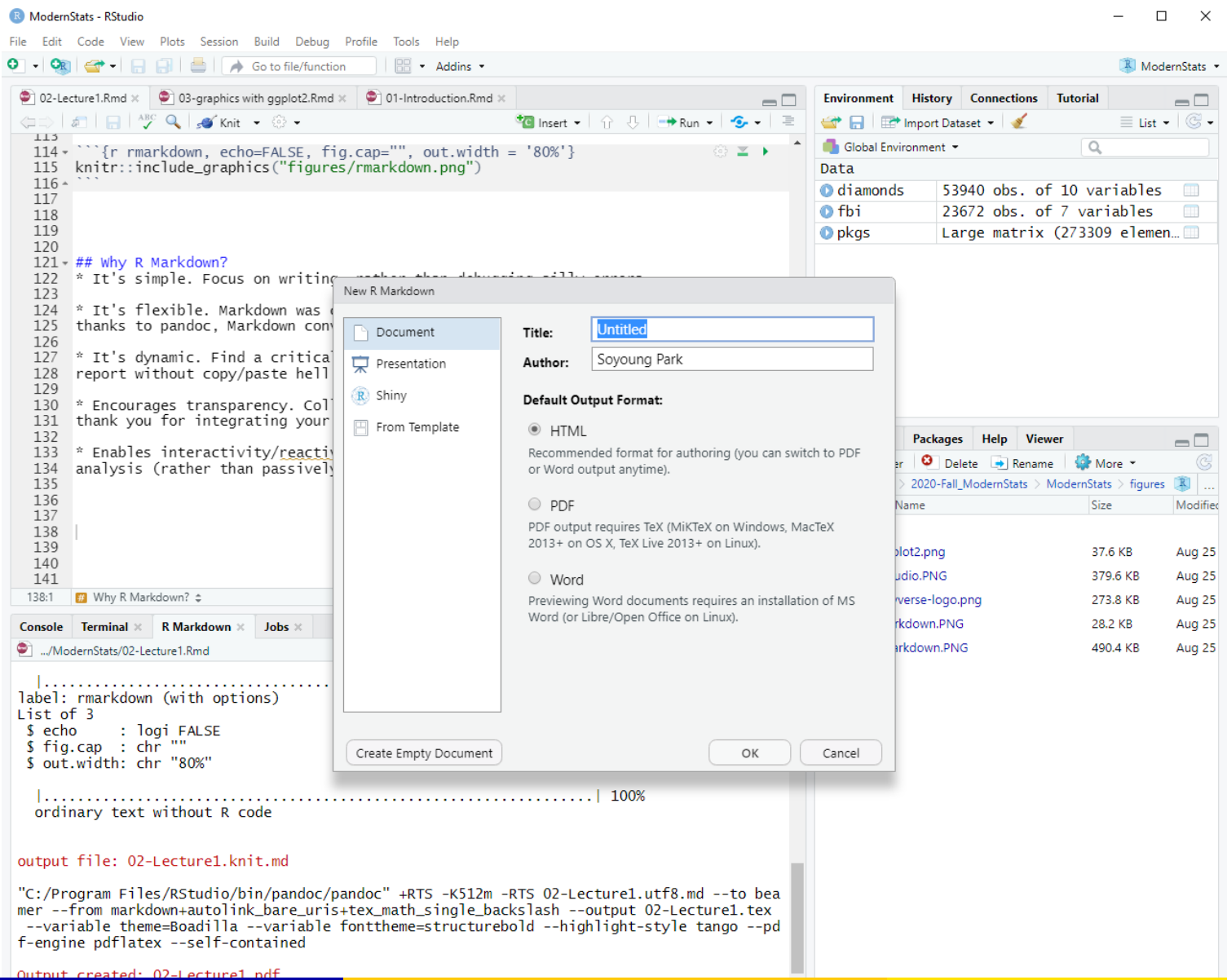
What is RMarkdown?

```
#install.packages("rmarkdown")
#install.packages("knitr")
library(rmarkdown)
library(knitr)
```

Why R Markdown?

- It's simple. Focus on writing, rather than debugging silly errors.
- It's flexible. Markdown was created to simplify writing HTML, but thanks to pandoc, Markdown converts to many different formats!
- It's dynamic. Find a critical error? Get a new dataset? Regenerate your report without copy/paste hell!
- Encourages transparency. Collaborators (including your future self) will thank you for integrating your analysis & report.
- Enables interactivity/reactivity. Allow your audience to explore the analysis (rather than passively read it).

RMarkdown


ModernStats - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
02-Lecture1.Rmd
03-graphics with ggplot2.Rmd
01-Introduction.Rmd
```{r rmarkdown, echo=FALSE, fig.cap="", out.width = '80%'}
knitr::include_graphics("figures/rmarkdown.png")
```
Why R Markdown?
* It's simple. Focus on writing
* It's flexible. Markdown was
thanks to pandoc, Markdown
* It's dynamic. Find a critica
report without copy/paste hell
* Encourages transparency. Col
thank you for integrating your
* Enables interactivity/reacti
analysis (rather than passively
Environment History Connections Tutorial
Import Dataset
Global Environment
Data
diamonds 53940 obs. of 10 variables
fbi 23672 obs. of 7 variables
pkgs Large matrix (273309 elemen...
New R Markdown
Document
Presentation
Shiny
From Template
Title: Untitled
Author: Soyoung Park
Default Output Format:
HTML
Recommended format for authoring (you can switch to PDF or Word output anytime).
PDF
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).
Word
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).
Create Empty Document
OK
Cancel
Packages Help Viewer
Delete Rename More
2020-Fall_ModernStats > ModernStats > figures
Name Size Modified
plot2.png 37.6 KB Aug 25
udio.PNG 379.6 KB Aug 25
verse-logo.png 273.8 KB Aug 25
rkdown.PNG 28.2 KB Aug 25
rkdown.PNG 490.4 KB Aug 25
Console Terminal R Markdown Jobs
.../ModernStats/02-Lecture1.Rmd
label: rmarkdown (with options)
List of 3
$ echo : logi FALSE
$ fig.cap : chr ""
$ out.width: chr "80%"
ordinary text without R code
output file: 02-Lecture1.knit.md
"C:/Program Files/RStudio/bin/pandoc/pandoc" +RTS -K512m -RTS 02-Lecture1.utf8.md --to beamer --from markdown+autolink_bare_uris+tex_math_single_backslash --output 02-Lecture1.tex --variable theme=Boadilla --variable fonttheme=structurebold --highlight-style tango --pdf-engine pdflatex --self-contained

Your turn

1. Open RStudio, create a new project.
2. Create a new RMarkdown file and knit it.
3. In a browser, navigate to the RMarkdown cheat sheet and download a copy (Google for the link!)
4. Use the cheat sheet to figure out how to make a change to the markdown formatting and knit again.
5. Export output to a different file format (try a word document). If you have Latex installed on your machine you can also try to export to a pdf file.
6. If you feel adventurous, change some of the R code and knit again.