

Working with RStudio

Michael Friendly

Psych 6136

<http://friendly.github.io/psy6136>

Getting started: Tools

- To profit best, you need to install both R and R Studio on your computer



The basic R system: R console (GUI) & packages

Download: <http://cran.us.r-project.org/>

Add my recommended packages:

source(["https://friendly.github.io/psy6136//R/install-pkgs.R"](https://friendly.github.io/psy6136//R/install-pkgs.R))



The R Studio IDE: analyze, write, publish

Download:

<https://www.rstudio.com/products/rstudio/download/>

Add: R Studio-related packages, as useful



R package tools



Data prep: Tidy data makes analysis and graphing much easier.

Packages: [tidyverse](#), comprised of: [tidyr](#), [dplyr](#), [lubridate](#), ...

The tidyverse

Components



R graphics: general frameworks for making standard and custom graphics

Graphics frameworks: base graphics, [lattice](#), [ggplot2](#), [rgl](#) (3D)

Application packages: [car](#) (linear models), [vcd](#) (categorical data analysis), [heplots](#) (multivariate linear models)



Publish: A variety of R packages make it easy to write and publish research reports and slide presentations in various formats (HTML, Word, LaTeX, ...), all within R Studio



Web apps: R now has several powerful connections to preparing dynamic, web-based data display and analysis applications.

Getting started: R Studio

The screenshot shows the RStudio interface with several panels and annotations:

- Console:** A large empty white area on the left side, annotated with "R console (just like Rterm)".
- Workspace/History:** A panel on the top right with a toolbar containing "Load", "Save", "Import Dataset", and "Clear All". An arrow points to the "Save" button with the text "command history workspace: your variables".
- Files:** A panel on the bottom right showing a file explorer view of the user's home directory. It contains a list of files and folders with columns for Name, Size, and Modified. Annotations include "files" pointing to the file list, "plots" pointing to the "Plots" menu, "packages" pointing to the "Packages" menu, and "help" pointing to the "Help" menu.

Name	Size	Modified
.Rhistory	1.6 KB	Jun 10, 2011, 1:59 PM
20070724_data.xls	13.5 KB	Jan 7, 2008, 11:51 PM
AutoHotkey.ahk	11.3 KB	Feb 28, 2011, 12:04 PM
code	10.3 KB	Apr 21, 2009, 10:00 AM
counts.xls	23.5 KB	Jan 3, 2008, 5:35 PM
cuznsim_ss.xls	29.5 KB	Aug 5, 2008, 4:32 PM
Default.rdp	2 KB	Dec 20, 2010, 8:30 AM
Digsby Logs		
docs		
docs-archive		
Downloads		
eagle		
ExpressPCB		
facs-log.xls	13.5 KB	Oct 22, 2007, 11:29 AM
FlowJo75.prefs	6.4 KB	Mar 9, 2011, 5:01 PM
funding_ops_deadearly2007.txt	918 bytes	Dec 28, 2006, 11:47 AM
Geneious Backup 2011-04-27.zip	3.8 MB	Apr 27, 2011, 1:36 PM
Geneious Backup 2011-05-16.zip	6.5 MB	May 16, 2011, 5:55 PM

R Studio navigation

R folder navigation commands:

- Where am I?

```
> getwd()
[1] "C:/Dropbox/Documents/6136"
```

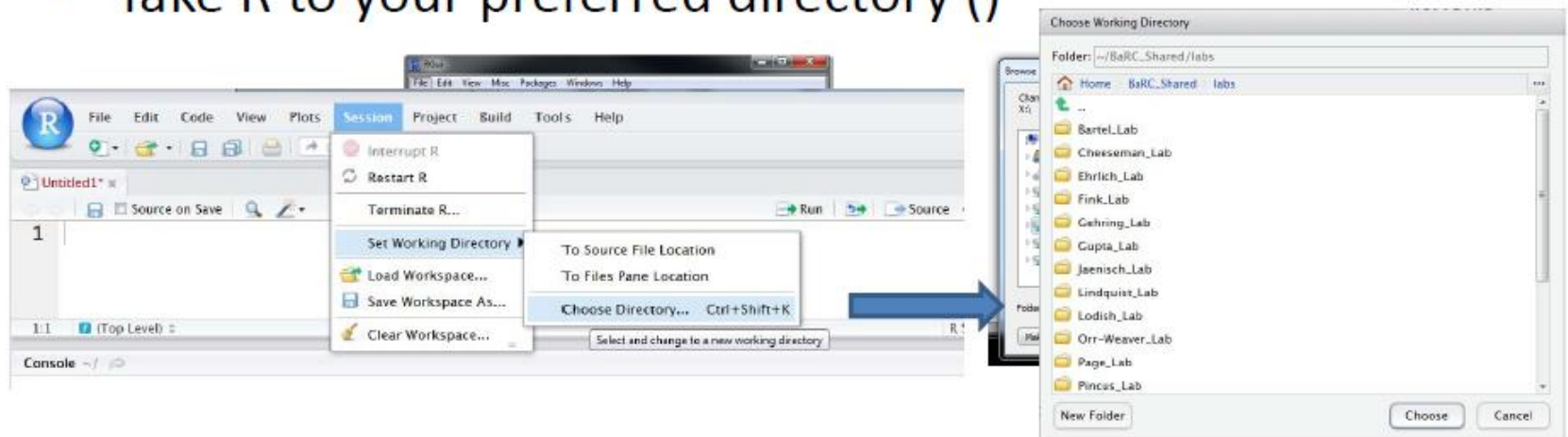
Better yet: create an R project!

- Go somewhere:

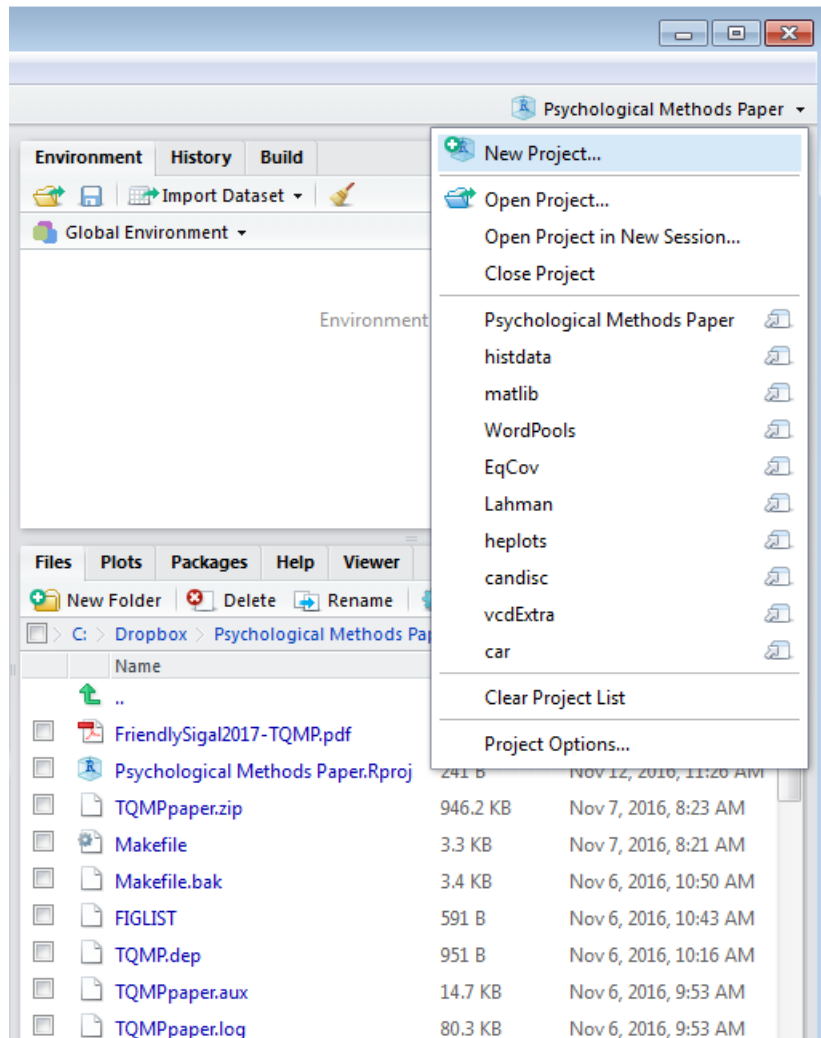
```
> setwd("C:/Dropbox")
> setwd(file.choose())
```

R Studio GUI

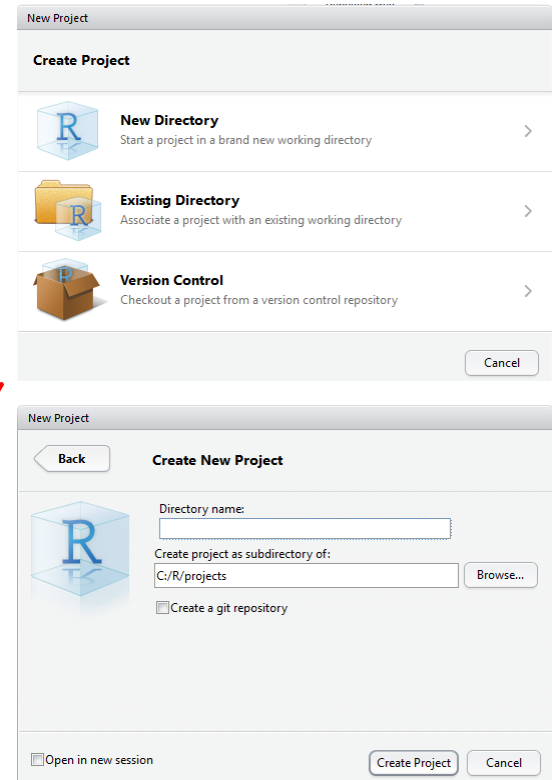
- Take R to your preferred directory ()



R Studio projects



R Studio projects are a handy way to organize your work



Lahman.Rproj

The .Rproj item opens the project in R Studio

R Studio projects

An R Studio project for a **research paper**: R files (scripts), Rmd files (text, R “chunks”)

The screenshot displays the R Studio interface for a project titled "Psychological Methods Paper". The main editor shows an R Markdown file with the following content:

```
1 ---
2 title: "Graphical Methods for Multivariate Linear Models in
3 Psychological Research: An R Tutorial"
4 shorttitle: "Graphical Methods for MLMs"
5 author:
6   - name: Michael Friendly
7     affiliation: 1
8   - name: Matthew Sigal
9     affiliation: 1
10  address: Psychology Department, York University, Toronto, Ontario,
11  Canada, M3J1P3
12  email: friendly@yorku.ca
13  - name: Matthew Sigal
14  affiliation: 1
15  affiliation:
16  - id: 1
17  institution: York University
18 abstract: |
19  This paper is designed as a tutorial to highlight some
20  recent developments for visualizing the relationships among
21  response and predictor variables in multivariate linear
```

The console shows the R startup message:

```
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

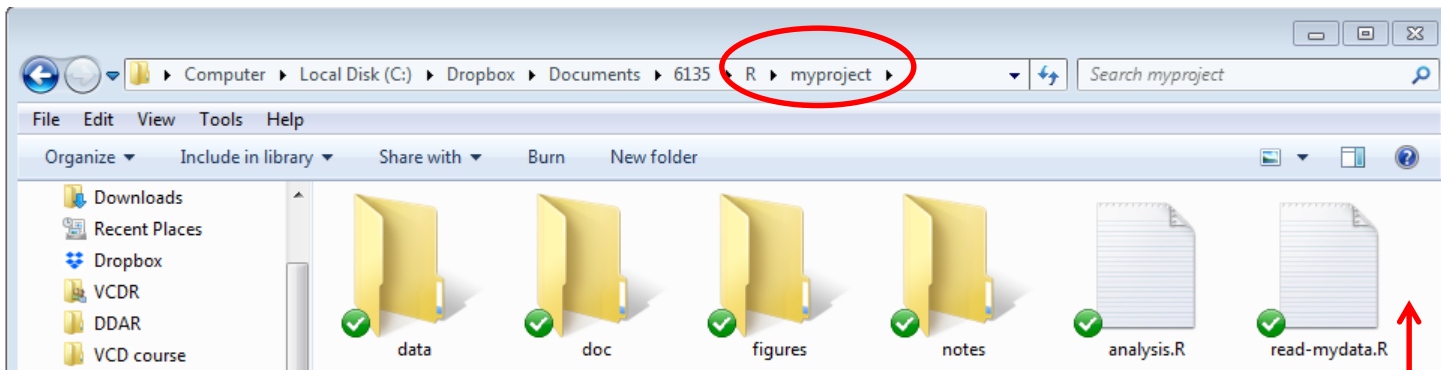
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

The Environment pane is empty, and the Files pane shows the project structure:

Name	Size	Modified
..		
FriendlySigal2017-TQMP.pdf		
Psychological Methods Paper.Rproj	241 B	Nov 12, 2016, 11:20 AM
TQMPpaper.zip	946.2 KB	Nov 7, 2016, 8:23 AM
Makefile	3.3 KB	Nov 7, 2016, 8:21 AM
Makefile.bak	3.4 KB	Nov 6, 2016, 10:50 AM
FIGLIST	591 B	Nov 6, 2016, 10:43 AM
TQMP.dep	951 B	Nov 6, 2016, 10:16 AM
TQMPpaper.aux	14.7 KB	Nov 6, 2016, 9:53 AM
TQMPpaper.log	80.3 KB	Nov 6, 2016, 9:53 AM
TQMPpaper.out	6 KB	Nov 6, 2016, 9:53 AM
TQMPpaper.pdf	819.8 KB	Nov 6, 2016, 9:53 AM
TQMPpaper.synctex.gz	286.1 KB	Nov 6, 2016, 9:53 AM
.Rhistory	13.5 KB	Nov 3, 2016, 9:35 AM

Organizing an R project

- Use a **separate** folder for each **project**
- Use **sub-folders** for various **parts**



data files:

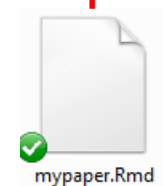
- raw data (.csv)
- saved R data (.Rdata)

figures:

- diagrams
- analysis plots

R files:

- data import
- analysis



Write up files will go here (.Rmd, .docx, .pdf)

This project, saved in a **Dropbox** folder automatically syncs with all my computers & collaborators. I use Git & **GitHub** for more serious work.

Organizing an R project

- Use separate R files for different steps:
 - Data import, data cleaning, ... → save as an RData file
 - Analysis: load RData, ...

read-mydata.R

```
# read the data; better yet: use RStudio File -> Import Dataset ...  
mydata <- read.csv("data/mydata.csv")
```

```
# data cleaning:  
#   filter missing, make factors, transform variables, ....
```

```
# save the current state  
save("data/mydata.RData")
```

Organizing an R project

- Use separate R files for different steps:
 - Data import, data cleaning, ... → save as an RData file
 - Analysis: load RData, ...

analyse.R

#' ## load the data

```
load("data/mydata.RData")
```

#' ## do the analysis – exploratory plots

```
plot(mydata)
```

#' ## fit models

```
mymod.1 <- lm(y ~ X1 + X2 + X3, data=mydata)
```

#' ## plot models, extract model summaries

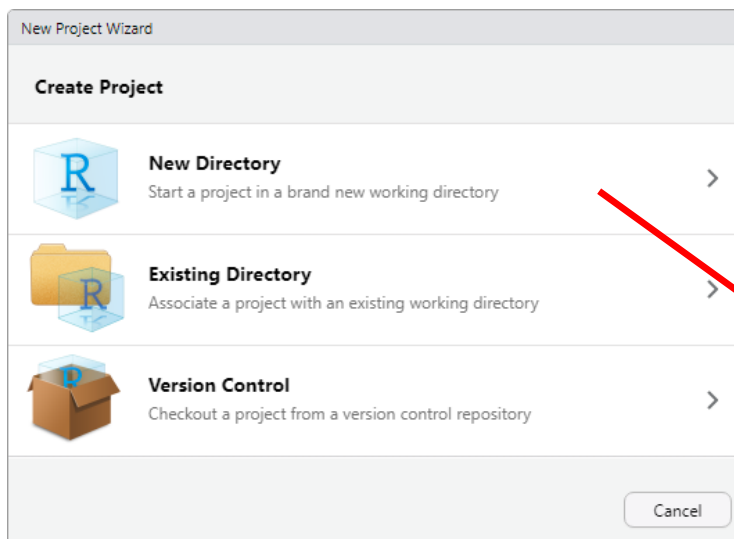
```
plot(mymod.1)
```

```
summary(mymod.1)
```

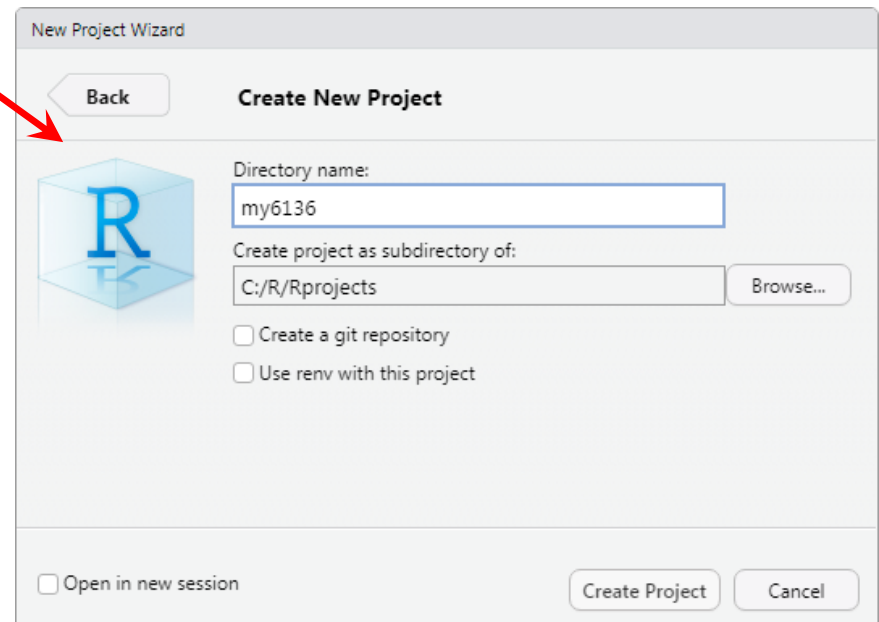
NB: #' ## is a special R comment for a H2 heading in an R “notebook” script

Your psy6136 project

I recommend that you create your own RStudio project for work in this course
This will enable you to keep all your work together, in a structured way



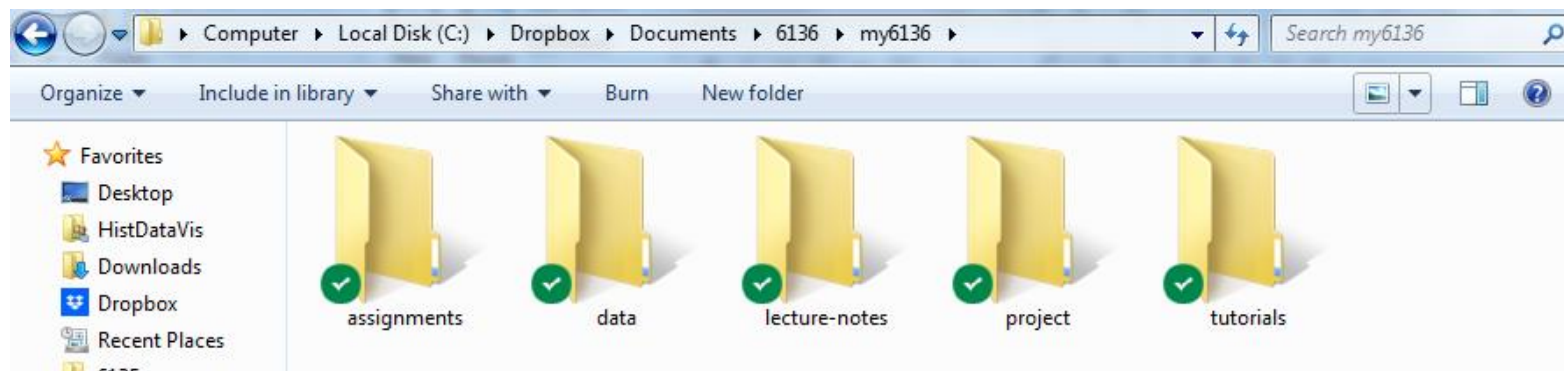
This will create a file, my6136.Rproj
Double-click on this to launch it in
RStudio



Your psy6136 project

I also recommend that you create [sub-folders](#) there to organize your work

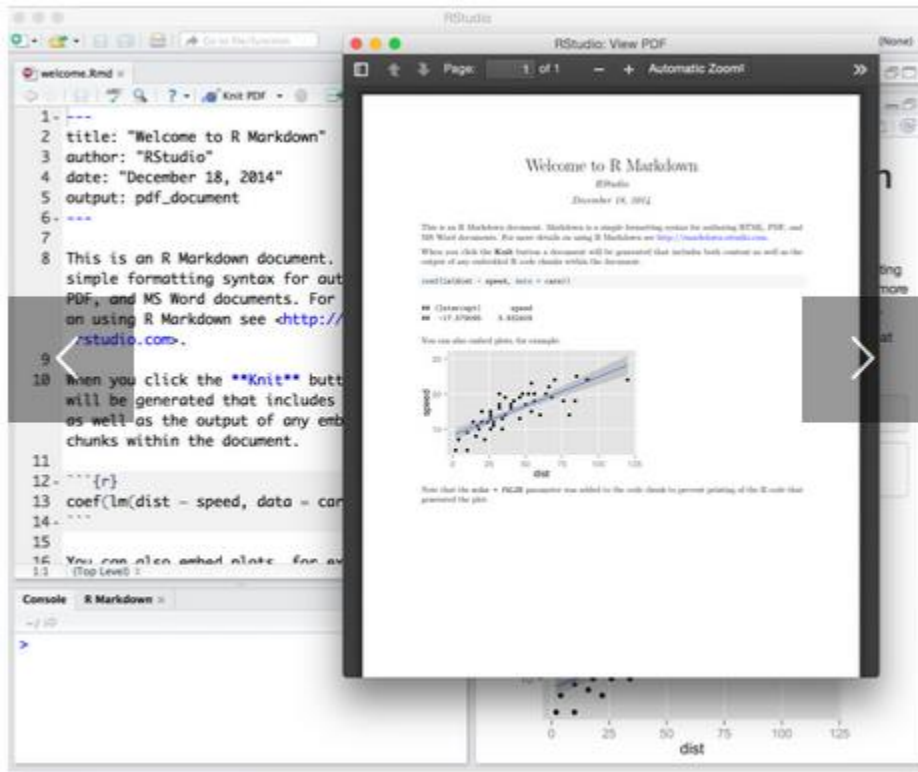
What you name them is up to you, but the main thing is for you to keep things organized by topics.



R scripts, data,
outputs for
assignments

work on
tutorials, R
examples

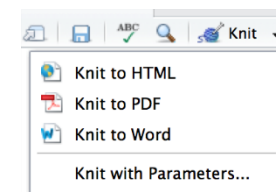
Reproducible analysis & reporting



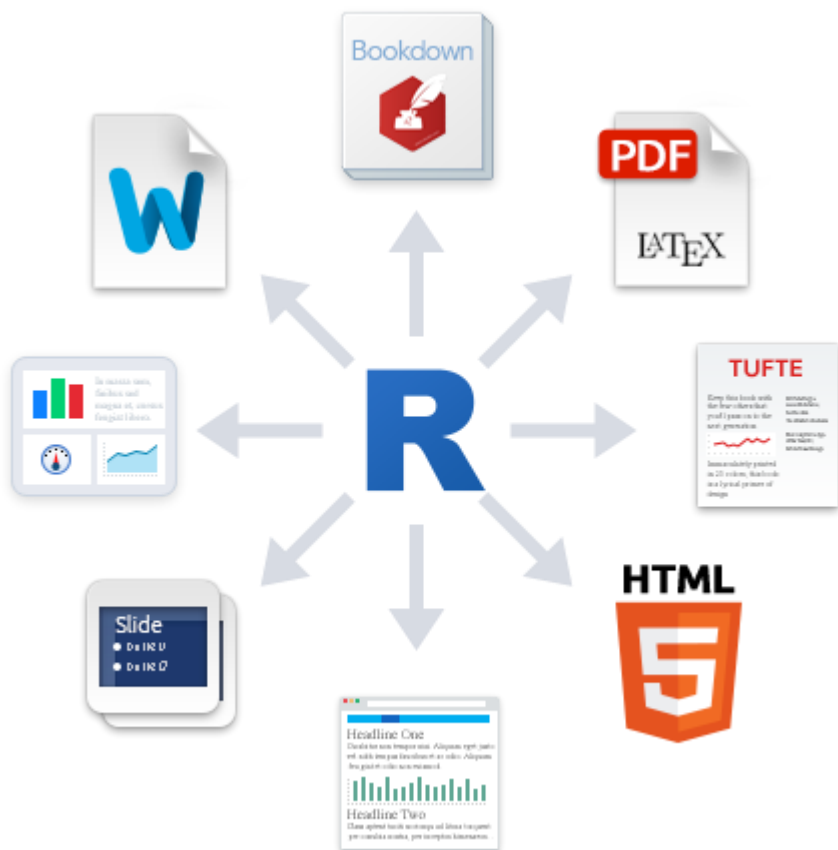
R Studio, together with the knitr and rmarkdown packages provide an easy way to combine writing, analysis, and R output into complete documents

.Rmd files are just text files, using rmarkdown markup and knitr to run R on “code chunks”

A given document can be rendered in different output formats:

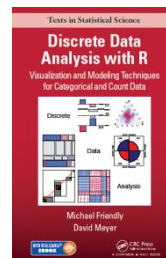


Output formats and templates



The integration of R, R Studio, knitr, rmarkdown and other tools is now highly advanced.

My last book was written entirely in R Studio, using .Rnw syntax → LaTeX → PDF → camera ready copy



The ggplot2 book was written using .Rmd format.



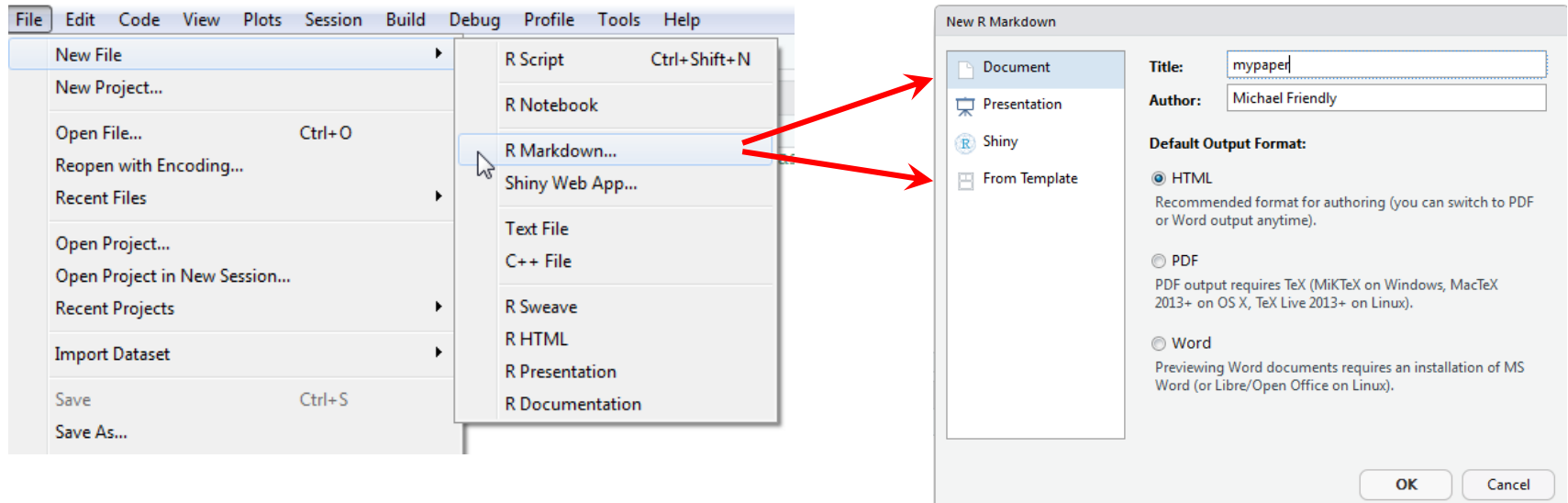
The [bookdown](#) package makes it easier to manage a book-length project – TOC, fig/table #s, cross-references, etc.

Also: [blogdown](#), [posterdown](#), ...

Templates are available for APA papers, slides, handouts, entire web sites, etc.

Writing it up

- In R Studio, create a `.Rmd` file to use R Markdown for your write-up
 - lots of options: HTML, Word, PDF (needs LaTeX)
 - templates for various pub types



Writing it up

- Use simple Markdown to write text
- Include **code chunks** for analysis & graphs

mypaper.Rmd, created from a template

```
1 ---
2 title: "mypaper"
3 author: "Michael Friendly"
4 date: "January 29, 2018"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting
15 details on using R Markdown see <http://rmarkdown.rstudio.com>.
16
17 when you click the Knit button a document will be generated
18 R code chunks within the document. You can embed an R code chunk
19
20 ```{r cars}
21 summary(cars)
22 ```
23
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 ```{r pressure, echo=FALSE}
29 plot(pressure)
30 ```
```

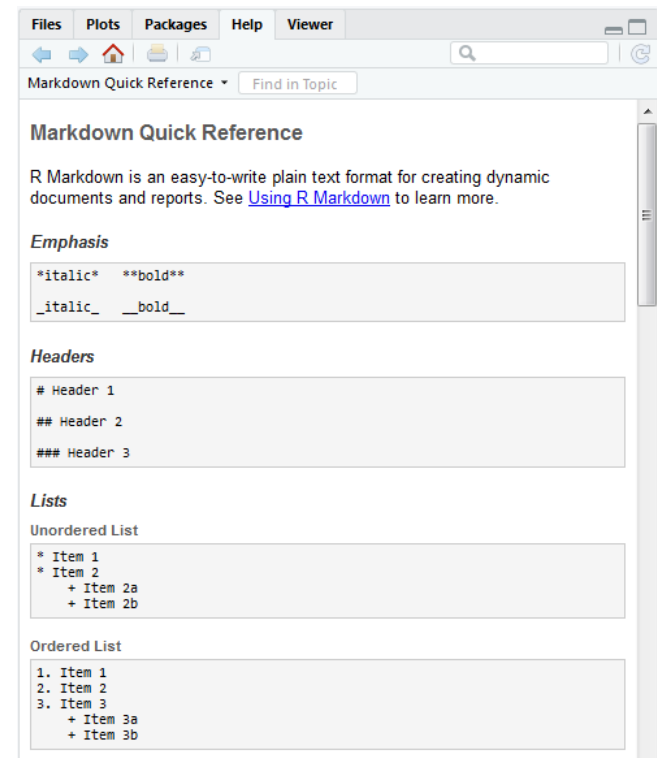
yaml header

Header 2

output code chunk

plot code chunk

Help -> Markdown quick reference



rmarkdown basics

rmarkdown uses simple formatting for all standard document elements

The image shows a side-by-side comparison of R Markdown source code and its rendered HTML output. Red arrows point from the source code to the rendered elements.

Source Code (example.Rmd):

```
1 # Header 1
2
3 This is an R Markdown document. Markdown is a
4 simple formatting syntax for authoring webpages.
5 Use an asterisk mark to provide emphasis, such
6 as italics or bold.
7 Create lists with a dash:
8
9 - Item 1
10 - Item 2
11 - Item 3
12
13 ```
14 Use back ticks to
15 create a block of code
16 ```
17
18 Embed LaTeX or MathML equations,
19  $\frac{1}{n} \sum_{i=1}^n x_i$ 
20
21 Or even footnotes, citations, and a
22 bibliography. [^1]
23
24 [^1]: Markdown is great.
```

Rendered Output (example.html):

Header 1

This is an R Markdown document. Markdown is a simple formatting syntax for authoring web pages.

Use an asterisk mark to provide emphasis, such as *italics* or **bold**.

Create lists with a dash:

- Item 1
- Item 2
- Item 3

```
Use back ticks to
create a block of code
```

Embed LaTeX or MathML equations, $\frac{1}{n} \sum_{i=1}^n x_i$

Or even footnotes, citations, and a bibliography. ¹

1. Markdown is great. ↩

R code chunks

R code chunks are run by `knitr`, and the results are inserted in the output document

The screenshot shows the RStudio interface. On the left, the source editor displays an R Markdown file named 'chunks.Rmd'. The code includes a title 'R Code Chunks', a paragraph about R Markdown, and an R chunk for a quick summary and plot. The R code in the chunk is:

```
library(ggplot2)
summary(cars)
qplot(speed, dist, data=cars) +
  geom_smooth()
```

On the right, the 'Preview HTML' window shows the rendered output. It includes the title 'R Code Chunks', the introductory paragraph, the R code from the chunk, the summary output for the 'cars' dataset, and the resulting scatter plot with a smoothed regression line. The summary output is as follows:

##	speed	dist
##	Min. : 4.0	Min. : 2
##	1st Qu.:12.0	1st Qu.: 26
##	Median :15.0	Median : 36
##	Mean :15.4	Mean : 43
##	3rd Qu.:19.0	3rd Qu.: 56
##	Max. :25.0	Max. :120

Red arrows point from the R code in the source editor to the corresponding rendered output in the preview window. A red box highlights the R chunk syntax in the source editor.

An R chunk:
```\${r name, options}``  
# R code here  
```\n

There are many options for controlling the details of chunk output – numbers, tables, graphs

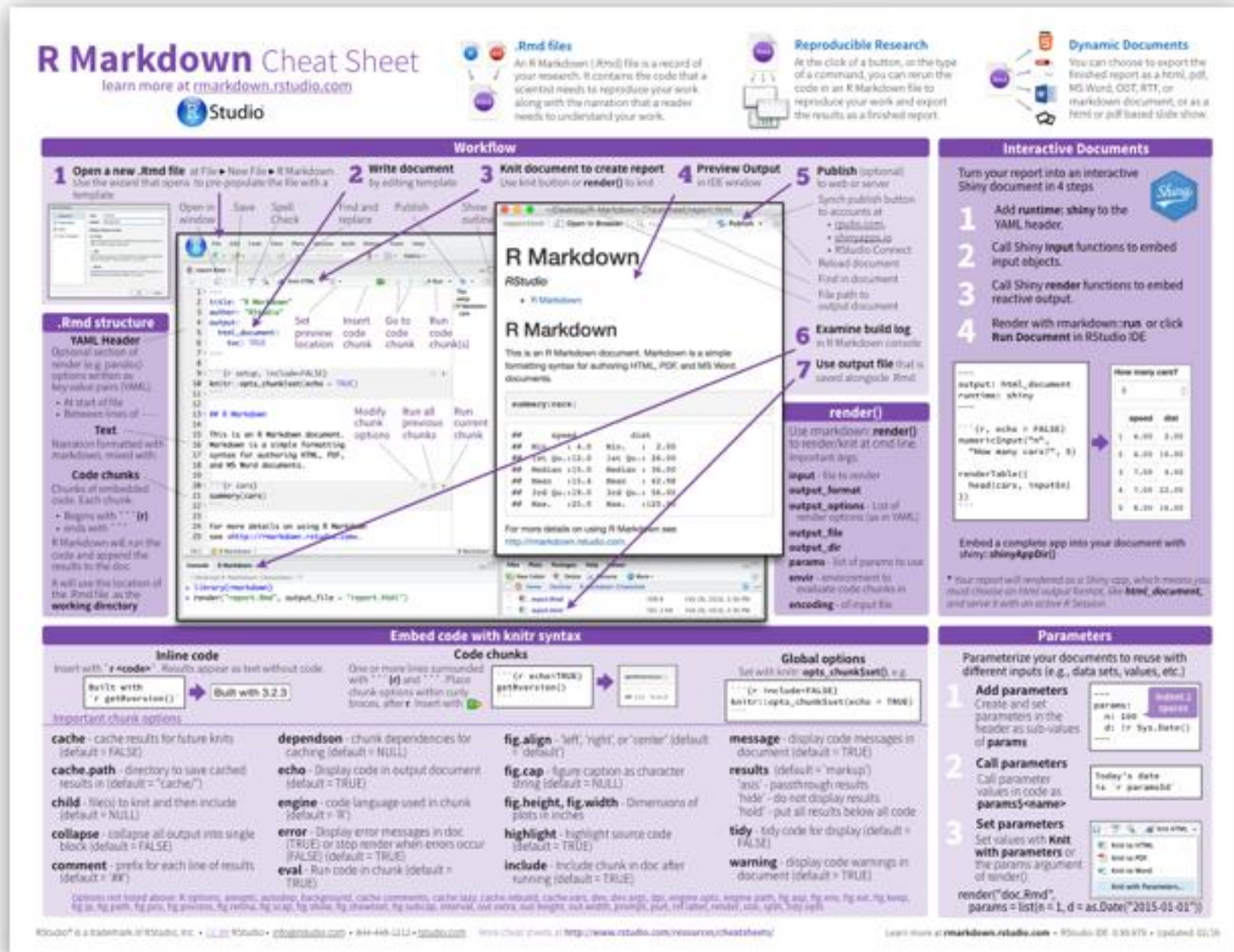
Choose the output format:

The screenshot shows the 'Knit' menu in RStudio, which offers the following options:

- Knit to HTML
- Knit to PDF
- Knit to Word
- Knit with Parameters...

The R Markdown Cheat Sheet provides most of the details

<https://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf>



R Markdown Cheat Sheet
learn more at rmarkdown.rstudio.com
Studio

Workflow

- 1 Open a new .Rmd file:** At File > New File > R Markdown. Use the wizard that opens to pre-populate the file with a template.
- 2 Write document:** by editing templates.
- 3 Knit document to create report:** Use knit button or `render()` to knit.
- 4 Preview Output:** in IDE window.
- 5 Publish (optional):** to web or server. Sync publish button to accounts at + GitHub, + shinyapps.io, + RStudio Connect. Reload document. Find in document. File path to output document.

Reproducible Research
At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

Dynamic Documents
You can choose to export the finished report as a html, pdf, MS Word, ODT, RTT, or markdown document, or as a html or pdf based slide show.

Interactive Documents
Turn your report into an interactive Shiny document in 4 steps:

- 1 Add runtime; shiny to the YAML header.
- 2 Call Shiny input functions to embed input objects.
- 3 Call Shiny render functions to embed reactive output.
- 4 Render with `markdown::run` or click Run Document in RStudio IDE.

.Rmd structure

YAML Header
Optional section of header in e.g. pandoc options written as key-value pairs (YAML).
- At start of file
- Remove lines of

Text
Narration formatted with markdown, mixed with

Code chunks
Chunks of embedded code. Each chunk
- Begins with ````{R}`
- ends with `````
R Markdown will run the code and append the results to the doc.
It will use the location of the .Rmd file as the working directory.

Embed code with knitr syntax

Inline code
Insert with `<code>`. Results appear as text without code.
Built with `r.getVersion()` → Built with 3.2.3

Code chunks
One or more lines surrounded with ````{R}` and `````. Place chunk options within curly braces, after ````{R}`.

Global options
Set with knitr: `opts_chunk$set()`, e.g.
(`r.getVersion()`)
knitr::opts_chunk\$set(cache = TRUE)

cache - cache results for future knits (default = FALSE)
cache.path - directory to save cached results in (default = "cache")
child - files() to knit and then include (default = NULL)
collapse - collapse all output into single block (default = FALSE)
comment - prefix for each line of results (default = "#")

dependson - chunk dependencies for caching (default = NULL)
echo - Display code in output document (default = TRUE)
engine - code language used in chunk (default = "R")
error - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = TRUE)
eval - Run code in chunk (default = TRUE)

fig.align - 'left', 'right', or 'center' (default = 'center')
fig.cap - figure caption as character string (default = NULL)
fig.height, **fig.width** - Dimensions of plots in inches
highlight - highlight source code (default = TRUE)
include - include chunk in doc after running (default = TRUE)

message - display code messages in document (default = TRUE)
results - display code results in document ('markup', 'asis' - pass through results, 'hide' - do not display results, 'hold' - put all results below all code)
tidy - tidy code for display (default = FALSE)
warning - display code warnings in document (default = TRUE)

Parameters
Parameterize your documents to reuse with different inputs (e.g., data sets, values, etc.)

- 1 Add parameters. Create and set parameters in the header as sub-values of `params`.
- 2 Call parameters. Call parameter values in code as `params$name`.
- 3 Set parameters. Set values with `knit` with parameters or the `params` argument of `render()`.

`render("doc.Rmd", params = list(a = 1, d = as.Date("2015-01-01")))`

render()
Use `markdown::render()` to render/knit at cmd line. Supports logs.
`input` - file to render
`output_format` - list of render options (see in YAML)
`output_file`
`output_dir`
`params` - list of params to use `envir` environment to evaluate code chunks in `encoding` - of input file

Embed a complete app into your document with `shiny::shinyAppDir()`

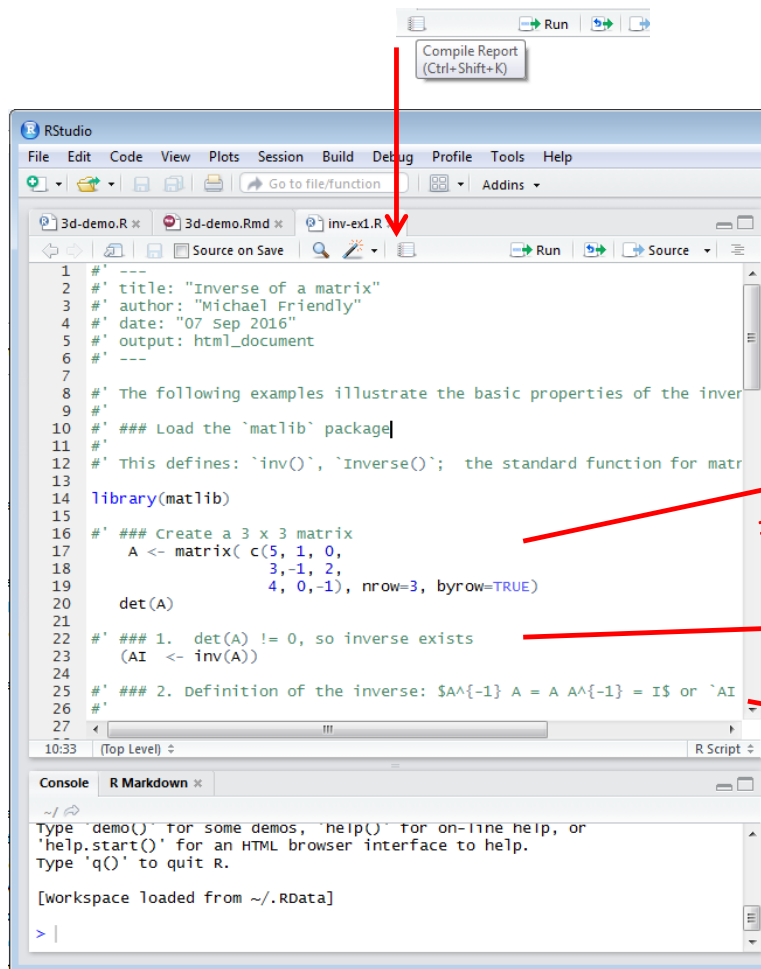
* Your report will be rendered as a Shiny app, which means you must check on file output format, like `knit_document`, and serve it with an active R Session.

RStudio® is a trademark of RStudio, Inc. • © 2016 RStudio • info@rstudio.com • 844-449-4111 • www.rstudio.com • <http://www.rstudio.com/resources/cheatsheets/>

Learn more at rmarkdown.rstudio.com • RStudio IDE © 2015-01-01 • updated 02-16

R notebooks

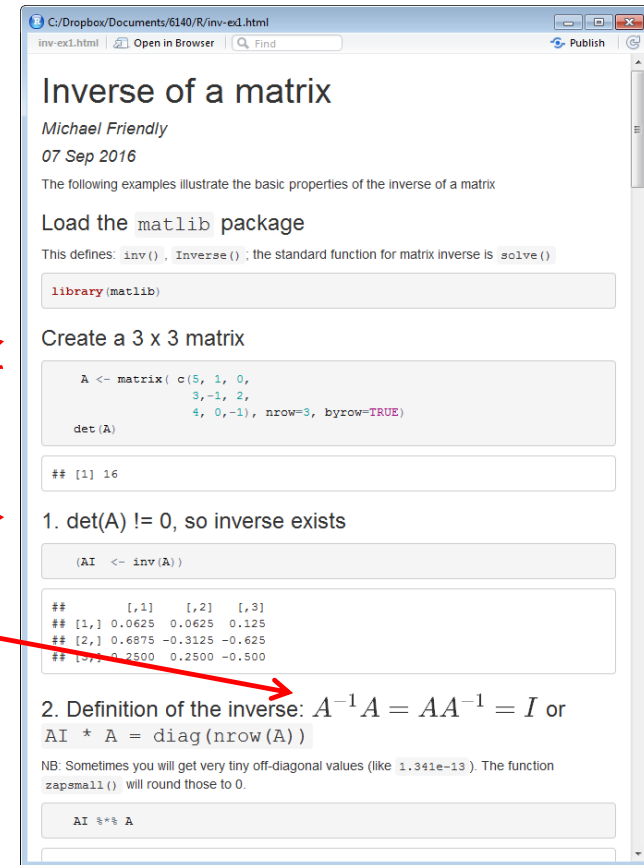
Often, you just want to “compile” an R script, and get the output embedded in the result, in HTML, Word, or PDF. Just type Ctrl-Shift-K or tap the **Compile Report** button



```
1 #' ---
2 #' title: "Inverse of a matrix"
3 #' author: "Michael Friendly"
4 #' date: "07 Sep 2016"
5 #' output: html_document
6 #' ---
7
8 #' The following examples illustrate the basic properties of the inverse
9 #'
10 #' ### Load the `matlib` package
11 #' This defines: `inv()`, `Inverse()`; the standard function for matrix inverse is `solve()`
12
13 library(matlib)
14
15 #' ### Create a 3 x 3 matrix
16 A <- matrix(c(5, 1, 0,
17             3, -1, 2,
18             4, 0, -1), nrow=3, byrow=TRUE)
19
20 det(A)
21
22 #' ### 1. det(A) != 0, so inverse exists
23 (AI <- inv(A))
24
25 #' ### 2. Definition of the inverse:  $AA^{-1} = A A^{-1} = I$  or  $AI$ 
26 #'
27
```

header

use math



Inverse of a matrix

Michael Friendly
07 Sep 2016

The following examples illustrate the basic properties of the inverse of a matrix

Load the `matlib` package

This defines: `inv()`, `Inverse()`; the standard function for matrix inverse is `solve()`

```
library(matlib)
```

Create a 3 x 3 matrix

```
A <- matrix(c(5, 1, 0,
              3, -1, 2,
              4, 0, -1), nrow=3, byrow=TRUE)
det(A)
```

```
## [1] 16
```

1. $\det(A) \neq 0$, so inverse exists

```
(AI <- inv(A))
```

```
##           [,1] [,2] [,3]
## [1,] 0.0625 0.0625 0.125
## [2,] 0.6875 -0.3125 -0.625
## [3,] 0.2500 0.2500 -0.500
```

2. Definition of the inverse: $A^{-1}A = AA^{-1} = I$ or $AI * A = \text{diag}(nrow(A))$

NB: Sometimes you will get very tiny off-diagonal values (like `1.341e-13`). The function `zapsmall()` will round those to 0.

```
AI %*% A
```