Rmarkdown

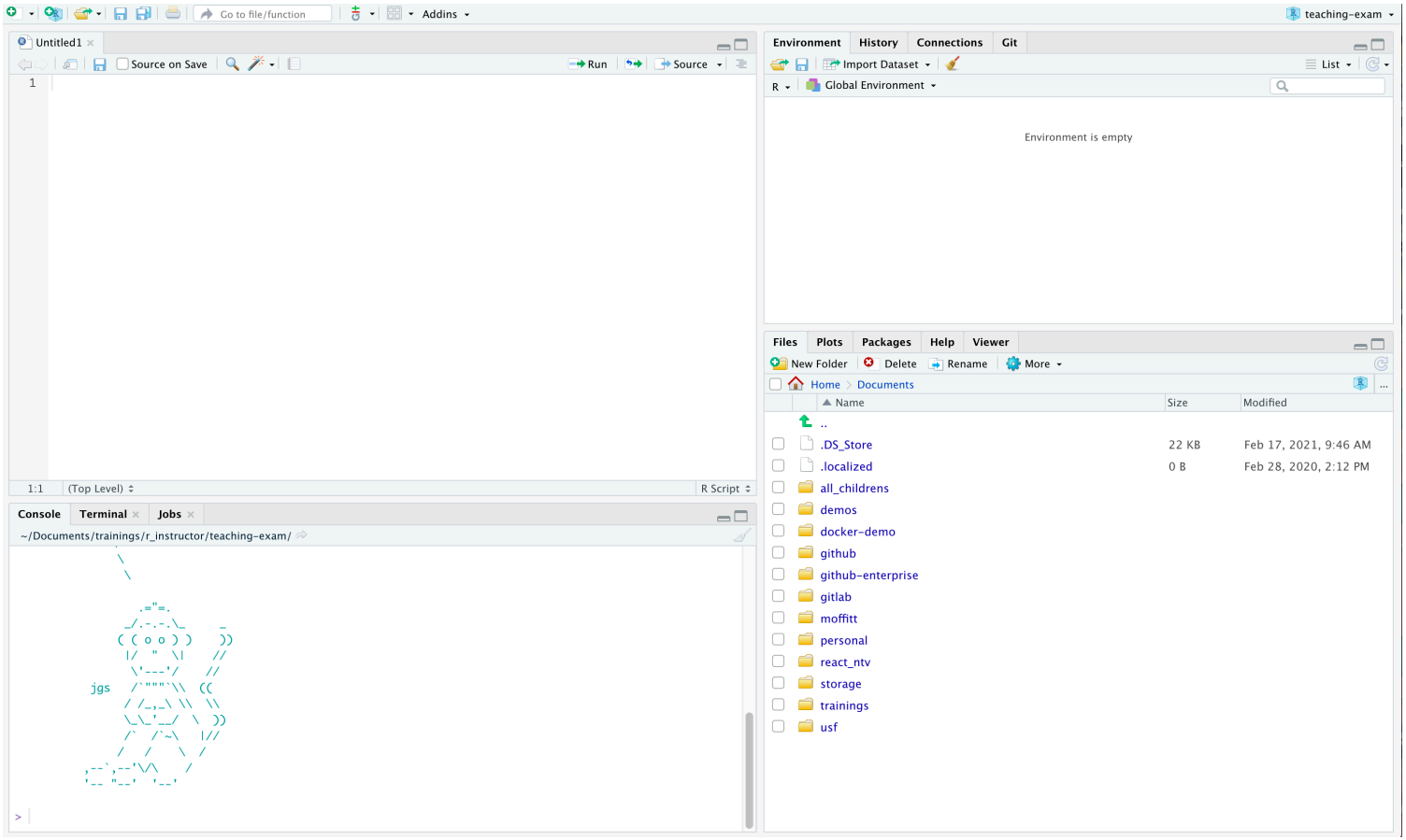
Creating interactive documents

Jordan Creed
Moffitt Cancer Center

June 09, 2021

R & RStudio

RStudio


Go to file/function
Addins
teaching-exam
Untitled1
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
Console
Terminal
Jobs
~/Documents/trainings/r_instructor/teaching-exam/
jgs
Environment
History
Connections
Git
Import Dataset
List
R
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Documents
Name
Size
Modified
..
.DS_Store
22 KB
Feb 17, 2021, 9:46 AM
.localized
0 B
Feb 28, 2020, 2:12 PM
all_childrens
demos
docker-demo
github
github-enterprise
gitlab
moffitt
personal
react_ntv
storage
trainings
usf

Packges

A `package` is a collection of functions, data, and documentation that extend the functionality of R.

```
install.packages("foo")
```

vs

```
library("foo")
```

The tidyverse

R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

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

Data types and structures

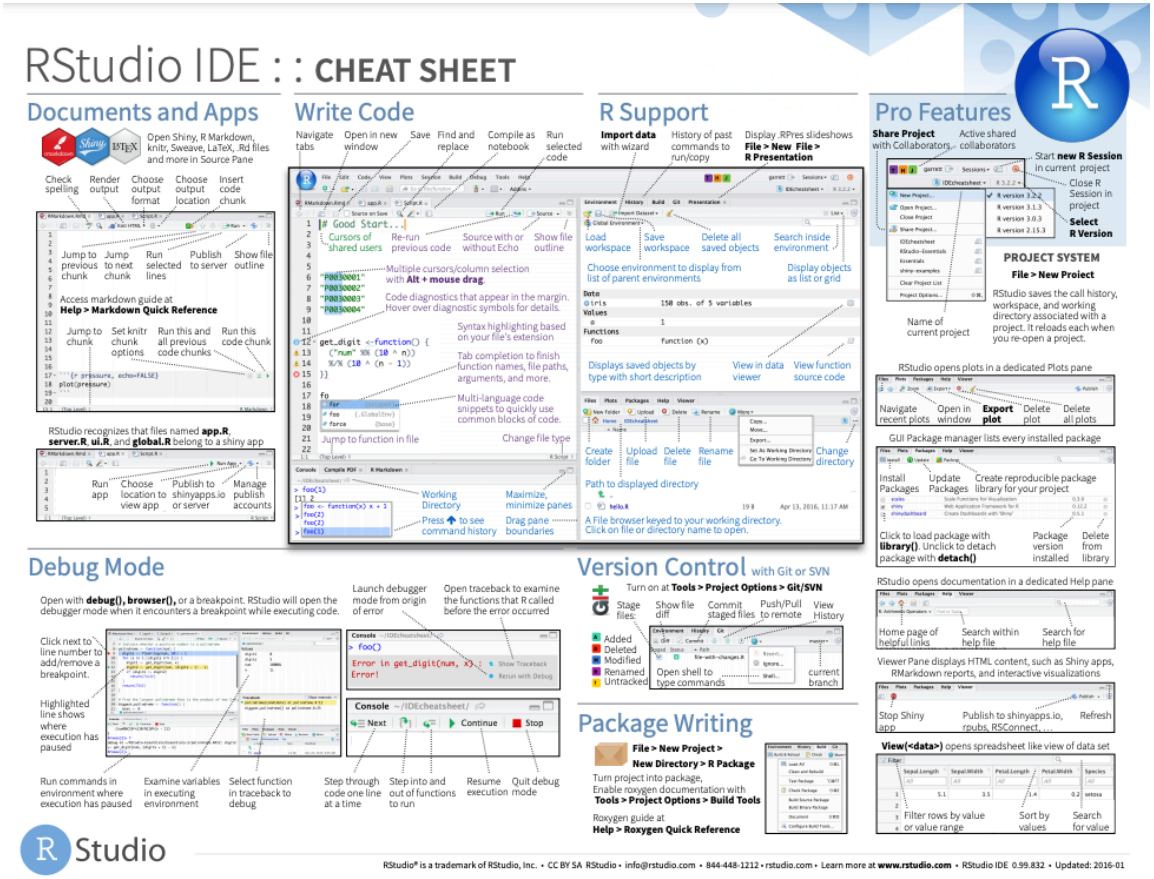
Data types:

- logical - `TRUE` or `FALSE`
- numeric or integer - `22.2`
- character - `"character example"`

Data structures:

- vector - `c(1:4, 5, 6)`
- list - `list(c(1:4), 12, "banana")`
- matrix
- dataframe/tibble

`tibble` and `dataframe` - look and perform virtually the same with the biggest difference being that `dataframes` have rows and `tibbles` do not



r cheatsheets

R Projects & `here()`

Working directory = where R looks for your files

Setting absolute paths can be problematic - especially if you are working on a project that uses Windows/Mac/Linux

R Projects provide a structure for storing all files, data, scripts and output for a project

To create a new Project: `File` > `New Directory` > `New Project` and fill out the information

☁️ In RStudio Cloud we will be working inside a project so you will not have the option to create projects and switch between them.

Rmarkdown

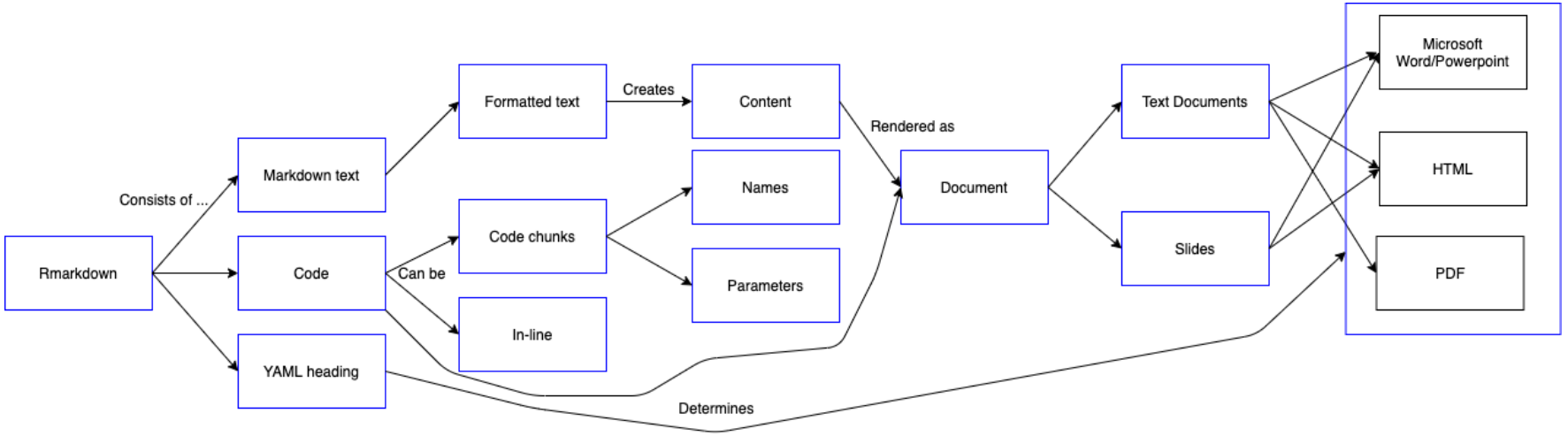
Rmarkdown

Allows you to share code, comments and outputs easily

☆ REPRODUCIBLE! ☆

3 elements: YAML header, formatted text, code

Rmarkdown


Rmarkdown
Consists of ...
Markdown text
Code
YAML heading
Formatted text
Code chunks
In-line
Can be
Creates
Content
Names
Parameters
Rendered as
Document
Text Documents
Slides
Microsoft Word/Powerpoint
HTML
PDF
Determines

YAML

Located at the top of your file and enclosed by `---`

Controls "whole document" parameters and settings

```
---
title: "My super cool document"
author:
  - Jordan Creed
output: html_document
---
```

Formatted Text

Rmarkdown uses regular markdown rules/syntax.

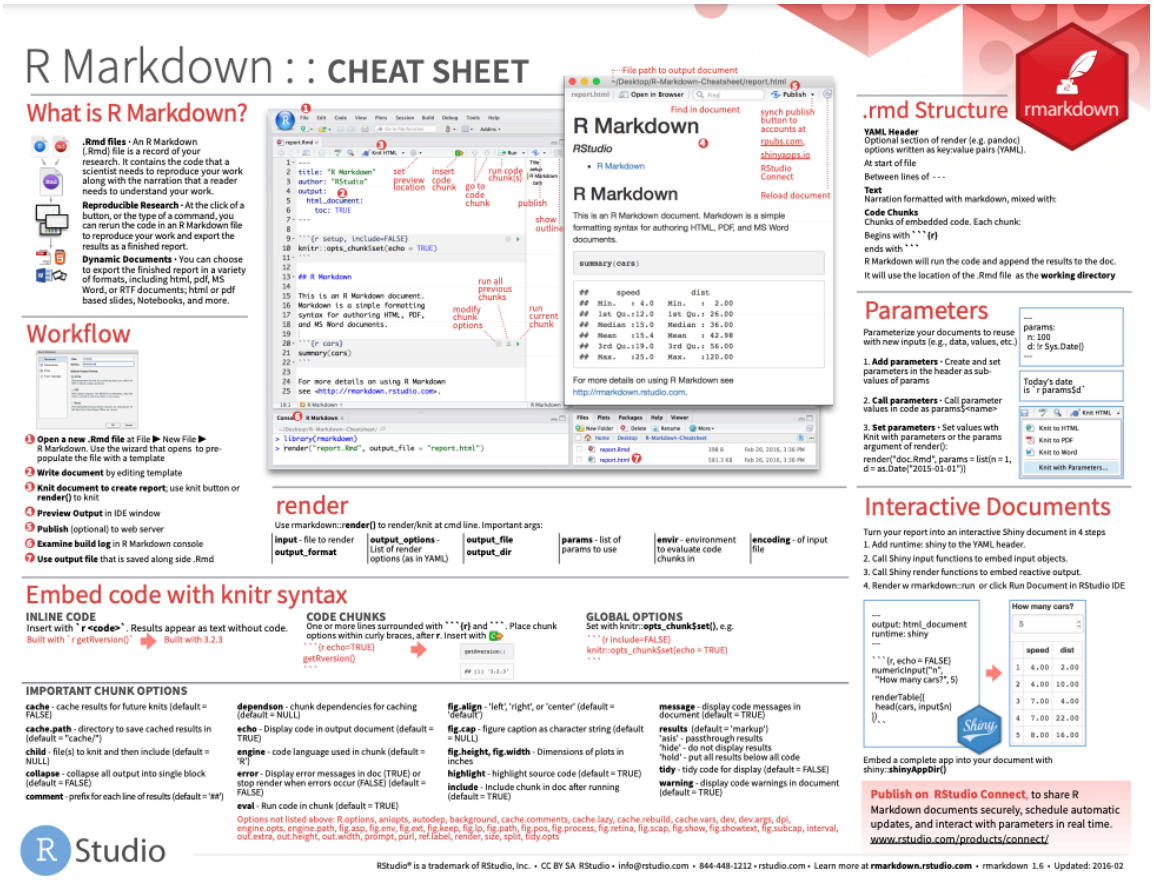
When you `knit` your document the final output will display according to the formatting rules on your `.Rmd`

Code

Code can be written as either code chunks or inline code

Output can be modified by chunk options placed inside `{}`

- `eval = FALSE` keeps code from being evaluated
- `echo = FALSE` keeps code from being displayed
- `message = FALSE` and `warning = FALSE` keeps messages/warnings from being displayed


R Markdown : : CHEAT SHEET
What is R Markdown?
.Rmd files - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.
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 in a variety of formats, including html, pdf, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.
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 template
3 Knit document to create report; use knit button or render() to knit
4 Preview Output in IDE window
5 Publish (optional) to web server
6 Examine build log in R Markdown console
7 Use output file that is saved along side .Rmd
File path to output document
~/Desktop/R-Markdown-Cheatsheet/report.html
Find in document
synch publish button to accounts at rpubs.com, shinyapps.io
RStudio Connect
Reload document
set preview location
insert code chunk
run code chunk(s)
go to code chunk
publish
show outline
run all previous chunks
modify chunk options
run current chunk
R Markdown
RStudio
R Markdown
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.
summary(cars)
For more details on using R Markdown see http://rmarkdown.rstudio.com.
render
Use rmarkdown::render() to render/knit at cmd line. Important args:
input - file to render
output_format
output_options - List of render options (as in YAML)
output_file
output_dir
params - list of params to use
envir - environment to evaluate code chunks in
encoding - of input file
Embed code with knitr syntax
INLINE CODE
Insert with `r <code>`. Results appear as text without code.
Built with `r getRversion()` Built with 3.2.3
CODE CHUNKS
One or more lines surrounded with ```{r} and ```. Place chunk options within curly braces, after r. Insert with
```{r echo=TRUE}
getRversion()
```
GLOBAL OPTIONS
Set with knitr::opts_chunk$set(), e.g.
```{r include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
IMPORTANT CHUNK OPTIONS
cache - cache results for future knits (default = FALSE)
cache.path - directory to save cached results in (default = "cache/")
child - file(s) 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 = FALSE)
eval - Run code in chunk (default = TRUE)
fig.align - 'left', 'right', or 'center' (default = 'default')
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 (default = 'markup') 'asis' - passthrough 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)
Options not listed above: R.options, aniopts, autodep, background, cache.comments, cache.lazy, cache.rebuild, cache.vars, dev, dev.args, dpi, engine.opts, engine.path, fig.asp, fig.env, fig.ext, fig.keep, fig.lp, fig.path, fig.pos, fig.process, fig.retina, fig.scap, fig.show, fig.showtext, fig.subcap, interval, out.extra, out.height, out.width, prompt, purl, ref.label, render, size, split, tidy.opts
.rmd Structure
rmarkdown
YAML Header
Optional section of render (e.g. pandoc) options written as key:value pairs (YAML).
At start of file
Between 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
Parameters
Parameterize your documents to reuse with new inputs (e.g., data, 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(n = 1, d = as.Date("2015-01-01")))
Knit to HTML
Knit to PDF
Knit to Word
Knit with Parameters...
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 w rmarkdown::run or click Run Document in RStudio IDE
How many cars?
speed dist
Embed a complete app into your document with shiny::shinyAppDir()
Publish on RStudio Connect, to share R Markdown documents securely, schedule automatic updates, and interact with parameters in real time.
www.rstudio.com/products/connect/
RStudio
RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at rmarkdown.rstudio.com • rmarkdown 1.6 • Updated: 2016-02

GitHub

- GitHub is a code hosting platform used for collaborating and code sharing
- Materials and information for the course can be found on the class GitHub page
- Class GitHub: https://github.com/FridleyLab/Introduction-to-R

RStudio Cloud

To get started:

- Create an account with RStudio Cloud
- Log in
- To your right you should see "Introduction To R Programming Class (2021)" - click on it
- Under "Projects" press "Start" beside "Lab One"