



R Markdown

2 / 20



Dynamic documents

- Combine code, rendered output (such as figures), and prose
- Reproduce your analyses
- Collaborate and share code with others
- Communicate your results with others

3 / 20

Markdown .md

- Created by [John Gruber](#) & [Aaron Swartz](#) in 2004
- A lightweight [markup language](#) to add formatting elements to plain-text documents, contrasting to WYSIWYG
- Portable, platform independent, everywhere (e.g. Piazza, but not Canvas 😊)

Rich text editor Plain text editor Markdown editor

PREVIEW: Report any editor feedback to bugs@piazza.com

Markdown syntax

I reply to student's questions in markdown on Piazza.

* **bold**
* *italic*

```r  
library(tidyverse)  
...  
```

We support [CommonMark](#) Markdown specification. Use `$\$` for LaTeX formulas.

4 / 20

M Basic syntax

Markdown	Rendered Output
## Heading level 2	Heading level 2
bold	bold
italic	<i>italic</i>
'inline code'	inline code
* apple	• apple
* orange	• orange
1. first	1. first
2. second	2. second
<http://stats220.earo.me>	http://stats220.earo.me
[stats220](http://stats220.earo.me)	stats220

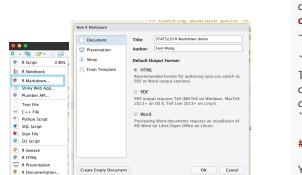
More on [Markdown syntax](#)

5 / 20



6 / 20

Create R Markdown .Rmd



See <<http://rmarkdown.rstudio.com>>. 7 / 20

YAML header

YAML is commonly used for configuration files, and starts and ends with ---

```
---  
title: "STATS220 R Markdown demo"  
author: "Earo Wang"  
date: "`r lubridate::today()`"  
output: html_document  
---
```

inline executable R code

➤ one markdown file, many formats

```
► pdf_document  
► xaringan::moon_reader
```

To compile to PDF,

```
install.packages('tinytex')  
tinytex::install_tinytex()
```

```
---  
title: "STATS220 R Markdown demo"  
author: "Earo Wang"  
date: "`r lubridate::today()`"  
output: html_document  
---
```

```
---  
title: "STATS220 R Markdown demo"  
author: "Earo Wang"  
date: "`r lubridate::today()`"  
output: html_document  
---
```

inline executable R code

```
► one markdown file, many formats  
► pdf_document  
► xaringan::moon_reader
```

R Markdown

You can embed an R code chunk like this:

```
```{r gapminder, message = FALSE}  
library(tidyverse)
gp <- read_rds("data/gapminder.rds")
gp
```

See <<http://rmarkdown.rstudio.com>>. 8 / 20

## R chunk for global settings

```
```{r setup, include = FALSE}
library(knitr)
opts_knit$set(root.dir = here::here())
opts_chunk$set(echo = TRUE)
````
```

### Executable R chunk

- r: language engine
- setup: unique chunk name
- include: include chunk output or not
- echo: display source code or not

```

title: "STATS220 R Markdown demo"
author: "Earo Wang"
date: "`r lubridate::today()`"
output: html_document
````
```

```
```{r setup, include = FALSE}
library(knitr)
opts_knit$set(root.dir = here::here())
opts_chunk$set(echo = TRUE)
````
```

R Markdown

You can embed an R code chunk like this:

```
```{r gapminder, message = FALSE}
library(tidyverse)
gp <- read_rds("data/gapminder.rds")
gp
````
```

See <http://rmarkdown.rstudio.com>. 9 / 20

Write in markdown

Weave together narrative text and code



```
---
title: "STATS220 R Markdown demo"
author: "Earo Wang"
date: "`r lubridate::today()`"
output: html_document
````
```

### ## R Markdown

You can embed an R code chunk like this:

```
```{r gapminder, message = FALSE}
library(tidyverse)
gp <- read_rds("data/gapminder.rds")
gp
````
```

See <http://rmarkdown.rstudio.com>. 10 / 20

## R chunk

- mpg: unique chunk name
- message: suppress message or not

```
library(tidyverse)
```

#> — Attaching packages

```
#> #> ✓ ggplot2 3.3.3 ✓ purrr 0.3.4
#> ✓ tibble 3.1.1 ✓ dplyr 1.0.5
#> ✓ tidyverse 1.1.3 ✓ stringr 1.4.0
#> ✓ readr 1.4.0 ✓ forcats 0.5.1
#> — Conflicts
#> #> ✘ dplyr::filter() masks stats::filter()
#> #> ✘ dplyr::lag() masks stats::lag()
```

```

title: "STATS220 R Markdown demo"
author: "Earo Wang"
date: "`r lubridate::today()`"
output: html_document
````
```

```
```{r setup, include = FALSE}
library(knitr)
opts_knit$set(root.dir = here::here())
opts_chunk$set(echo = TRUE)
````
```

R Markdown

You can embed an R code chunk like this:

```
```{r gapminder, message = FALSE}
library(tidyverse)
gp <- read_rds("data/gapminder.rds")
gp
````
```

See <http://rmarkdown.rstudio.com>. 11 / 20

Render

3 ways to render an .Rmd

1. click Knit
2. shortcut: Ctrl/Cmd + Shift + K
3. rmarkdown:::render("demo.Rmd")

STATS220 R Markdown demo

Earo Wang
2021-04-18

R Markdown

You can embed an R code chunk like this:

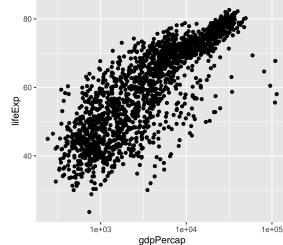
```
library(tidyverse)
gp <- read_rds("data/gapminder.rds")
gp
```

```
## # A tibble: 1,704 x 6
##   country continent year lifeExp   pop gdpPercap
##   <fct>    <fct>     <dbl>   <dbl>   <dbl>      <dbl>
## 1 Afghanistan Asia      1952 28.8  8425333    779.
## 2 Afghanistan Asia      1957 30.3  9249934   821.
## 3 Afghanistan Asia      1962 31.9  10745342   851.
## 4 Afghanistan Asia      1967 34.0 11537966   836.
## 5 Afghanistan Asia      1972 34.1 13079460   740.
## 6 Afghanistan Asia      1977 34.5 15237517   786.
## 7 Afghanistan Asia      1982 34.9 12881816   971.
## 8 Afghanistan Asia      1987 40.8 13867957   852.
## 9 Afghanistan Asia      1992 41.7 16317921   649.
## 10 Afghanistan Asia     1997 42.9 18121115   655.
## # ... with 1,694 more rows
```

See <http://rmarkdown.rstudio.com>.

12 / 20

```
```{r scatterplot, fig.align = "center", fig.width = 4.5, fig.height = 4}
ggplot(gp, aes(gdpPerCap, lifeExp)) +
 geom_point() + scale_x_log10()
```



13 / 20

```
```{r scatterplot, fig.align = "center", fig.width = 4.5, fig.height = 4}
ggplot(gp, aes(gdpPerCap, lifeExp)) +
  geom_point() + scale_x_log10()
```

Chunk options

- scatterplot gives the chunk a name.
- fig.align: alignment of figures
- fig.width/fig.height
- other figure options

Good practice: naming every single chunk!



image credit: Maëlle Salmon

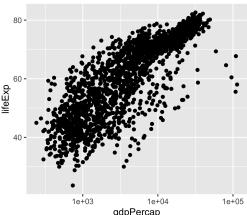
14 / 20

```
```{r show-code, ref.label = "scatterplot", eval = FALSE}
ggplot(gp, aes(gdpPerCap, lifeExp)) +
 geom_point() + scale_x_log10()
```

## Reuse chunks by reference

- ref.label: labels of the chunks from which the code of the current chunk is inherited
- eval: evaluate the code chunk

**Good practice:** naming every single chunk!



15 / 20

```
```{r lm, echo = FALSE, cache = TRUE}
fit <- lm(lifeExp ~ log10(gdpPerCap), data = gp)
kable(coef(summary(fit)))
```
```

## Chunk options

- echo: overwrites the global setting
- cache: evaluate once and skip for the future

kable() generates a very simple HTML table. ↗

|                  | Estimate  | Std. Error | t value   | Pr(> t ) |
|------------------|-----------|------------|-----------|----------|
| (Intercept)      | -9.100889 | 1.2276738  | -7.413117 | 0        |
| log10(gdpPerCap) | 19.353423 | 0.3425372  | 56.500206 | 0        |

16 / 20

## Beyond R: language engines

```
```{python hello}
# remotes::install_github("rstudio/reticulate")
x = "Hello, python world!"
print(x.split(" "))
```
#> ['hello,', 'python', 'world!']
```

```
```{sh head}
head -4 data/pisa/pisa-student.csv
```

```

```
#> year,country,school_id,student_id,mother_educ,father_educ,gender,computer,internet,math,
#> 2000,ALB,1001,1,NA,NA,female,NA,no,324.35,397.87,345.66,2.16,yes,no,no,1,3*,1,11-50,-0.6,
#> 2000,ALB,1001,3,NA,NA,female,NA,no,NA,368.41,385.83,2.16,yes,yes,no,2,0,0,1-10,-1.84,-1.46,
#> 2000,ALB,1001,6,NA,NA,male,NA,no,NA,294.17,327.94,2.16,yes,yes,no,2,0,0,1-10,-1.46,-1.306
```

17 / 20

## Beyond R: language engines

```
```{r db}
library(RSQLite)
db <- dbConnect(SQLite(), dbname = "data/pisa/pisa-student.db")
```
```{sql_query, connection = db, output.var = "pisa18"}
SELECT * FROM pisa WHERE year = 2018
```

```

```
#> # A tibble: 612,004 x 22
#> year country school_id student_id mother_educ father_educ
#> <dbl> <chr> <dbl> <chr> <chr> <chr>
#> 1 2018 ALB 800002 800251 ISCED 3A ISCED 3A
#> 2 2018 ALB 800002 800492 ISCED 2 ISCED 2
#> 3 2018 ALB 800002 801902 ISCED 2 ISCED 2
#> 4 2018 ALB 800002 803546 ISCED 2 ISCED 2
#> 5 2018 ALB 800002 804776 ISCED 2 ISCED 3A
#> 6 2018 ALB 800002 804825 ISCED 2 ISCED 2
#> # ... with 611,998 more rows, and 16 more variables:
#> # gender <chr>, computer <chr>, internet <chr>,
#> # math <dbl>, read <dbl>, science <dbl>, stu_wgt <dbl>,
```

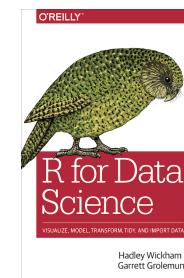
18 / 20

## Beyond R: language engines

```
names(knitr::knit_engines$get())
#> [1] "awk" "bash" "coffee" "gawk" "groovy" "haskell"
#> [7] "lein" "mysql" "node" "octave" "perl" "pgsql"
#> [13] "Rscript" "ruby" "sas" "scala" "sed" "sh"
#> [19] "stata" "zsh" "highlight" "Rcpp" "tikz" "dot"
#> [25] "c" "cc" "fortran" "fortran95" "asy" "cat"
#> [31] "asis" "stan" "block" "block2" "js" "css"
#> [37] "sql" "go" "python" "julia" "sass" "scss"
#> [43] "R" "bslib"
```

19 / 20

## Reading



- [R Markdown](#)
- [R Markdown formats](#)
- [R Markdown cheat sheet](#)
- [R Markdown reference guide](#)

20 / 20

