Intro to Markdown

Karl Dunkle Werner

June 29, 2021

Slides:
github.com/karldw/markdown_intro_2021-06-29

Thank you, Pamela!

- #econ_prosem is a great service

What is Markdown?

- Plain text
  - Which is a good thing!
    (See The Plain Person's Guide to Plain Text Social Science)
- Readable by itself
- That gets converted into HTML

Why?

- Lighter weight than LaTeX
  - Easier to jot things down
  - Less fiddly
  - Failure to compile is very rare
- Usable in places Word and LaTeX aren't
  - e.g. every Github page you've seen
  - Your own website (come back July 13!)

Why?

- Easy to use with version control (e.g. Git)
  - Track changes to your code *and your words* over time
  - Easily search through history
- Prettier than just writing in a `.txt` file

How do you write it?

- Recommended: Find a general-purpose text editor you like
  - VS Code, Atom, Emacs, Nano, Sublime, Vi, …
- Or: edit in-website
  - Github
  - Today: Hedgedoc

Syntax Basics

- Headings
- Items
- Enumerations
- Emphasis
- Math
- Code
- Links
- Images

Headings

Markdown	Rendered Output
`# Heading level 1`	Heading level 1
`## Heading level 2`	Heading level 2
`### Heading level 3`	Heading level 3
`#### Heading level 4`	Heading level 4
`##### Heading level 5`	Heading level 5
`###### Heading level 6`	Heading level 6

Items

```
Groceries:

- Yeast
- Flour
- Basil


Or with checkboxes (not universal):

- [x] Mozzarella
- [ ] Chili flakes
```

Groceries:

- Yeast
- Flour
- Basil

Or with checkboxes (not universal):

- ☑ Mozzarella
- ☐ Chili flakes

Enumerations

```
Colors I know:

1. Magenta
1. Cyan
1. Black

Colors I don't know:

4. Taupe
8. Chartreuse
7. Puce
```

Colors I know:

1. Magenta
2. Cyan
3. Black

Colors I don’t know:

4. Taupe
5. Chartreuse
6. Puce

Emphasis

```
*Italic* or _italic_

**Bold** or __bold__

***Bold italic*** or ___bold italic__

~~strikethrough~~
```

Italic or *italic*

Bold or **bold**

Bold italic or ***bold italic***

~~strikethrough~~

Math (inline)

- In: `$a^2 + b^2 = c^2$`
- Out: $a^2 + b^2 = c^2$

Math (display)

```
\[
\begin{align}
\int_0^\infty f(\cos^2(\psi)) &= z_0\\
&= \pi r^2
\end{align}
\]
```

$$\begin{aligned}\int_0^\infty f(\cos^2(\psi)) &= z_0\\ &= \pi r^2\end{aligned}$$

Caveats:

1. Depends on Mathjax, which is in some, but not all markdown renderers
2. The list of supported commands is very long, but it's not a LaTeX substitute

Code (inline)

- Inline code with single backticks:
  - In: `y = x1 + x2`
  - Out: `y = x1 + x2`

Code (blocks)

- Code blocks with triple backticks
  - Optionally add language syntax highlighting (e.g. r)

````
```r
x = lm(mpg ~ wt + cyl, data=mtcars)
broom::tidy(x)
#> # A tibble: 3 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)    39.7      1.71      23.1  3.04e-20
#> 2 wt             -3.19     0.757     -4.22 2.22e- 4
#> 3 cyl            -1.51     0.415     -3.64 1.06e- 3
```
````

```r
x = lm(mpg ~ wt + cyl, data=mtcars)
broom::tidy(x)
#> # A tibble: 3 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)    39.7      1.71      23.1  3.04e-20
#> 2 wt             -3.19     0.757     -4.22 2.22e- 4
#> 3 cyl            -1.51     0.415     -3.64 1.06e- 3
```

Comments and Escapes

```
<!--
Block comments

escape everything inside (except in a code block)

This is the same as comments in HTML code
-->
```

- Escape special characters with \

Links

- Plain URLs and emails get auto-linked (usually)
  - https://twitter.com/hashtag/econ_prosem
  - karldw@berkeley.edu
- Text links have the form `[words to display](URL)`
  - In: `[Twitter](https://twitter.com)`
  - Out: Twitter
- We can also link within the document:
  - In: `[Last section](#/comments-and-escapes)`
  - Out: Last section

Images

- Images are almost identical, but with `!` before `[`
  - In:

    ```
    ![penguin logo](palmerpenguins_logo.png)
    ```

  - Out:

Tables

- Sometimes easy to read
- Always a pain to write
- Have a computer do it for you (e.g. knitr in R)

```
| Reg. A       | Reg. B            |
| ------------ | ----------------- |
| 3.14         | 0.01              |
| [1.0, 10.0] | [-0.001, 0.015] |
```

→

Reg. A	**Reg. B**
3.14	0.01
[1.0, 10.0]	[-0.001, 0.015]

Tables

```
x = lm(mpg ~ wt + cyl, data=mtcars)
y = broom::tidy(x)
knitr::kable(y)

#> |term        |  estimate| std.error| statistic|   p.value|
#> |:-----------|---------:|---------:|---------:|---------:|
#> |(Intercept) | 39.686262| 1.7149840| 23.140893| 0.0000000|
#> |wt          | -3.190972| 0.7569065| -4.215808| 0.0002220|
#> |cyl         | -1.507795| 0.4146883| -3.635972| 0.0010643|
```

↓

term	estimate	std.error	statistic	p.value
(Intercept)	39.686262	1.7149840	23.140893	0.0000000
wt	-3.190972	0.7569065	-4.215808	0.0002220
cyl	-1.507795	0.4146883	-3.635972	0.0010643

More fancy markdown

- References
- R Markdown (`.Rmd`)
  - Works with Stata too, with some setup
- Direct HTML/CSS
- Footnotes (support varies)
- Pandoc conversion to other formats

References

- Citations (via Pandoc & Citeproc)
  - Input: `@Pigouvian_taxes:1932` (from my bib file)
  - Output: Pigou (1932)

```
<!-- Code to ask pandoc to print refs: -->
::: {#refs}
:::
```

Pigou, Arthur Cecil. 1932. *The Economics of Welfare*. 4th ed. Vol. 2. Macmillan; Co.

R Markdown

- Combines code and text
- Never have to copy updated tables, figures, or numbers into your text
- Works with Markdown or LaTeX
- Use Knitr to covert `.Rmd` to `.md`
  - Older tool: Sweave
- Use Pandoc to convert `.md` to anything

What does R Markdown look like?

What does R Markdown look like?

https://github.com/allisonhorst/palmerpenguins

README.md

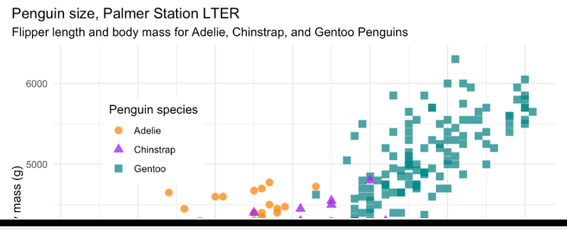
Examples

You can find these and more code examples for exploring palmerpenguins in `vignette("examples")`.

Penguins are fun to summarize! For example:

```
library(tidyverse)
penguins %>%
  count(species)
#> # A tibble: 3 x 2
#>   species       n
#>   <fct>     <int>
#> 1 Adelie      152
#> 2 Chinstrap    68
#> 3 Gentoo      124
penguins %>%
  group_by(species) %>%
  summarize(across(where(is.numeric), mean, na.rm = TRUE))
#> # A tibble: 3 x 6
#>   species   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g  year
#>   <fct>              <dbl>         <dbl>             <dbl>       <dbl> <dbl>
#> 1 Adelie              38.8          18.3               190.       3701. 2008.
#> 2 Chinstrap           48.8          18.4               196.       3733. 2008.
#> 3 Gentoo              47.5          15.0               217.       5076. 2008.
```

Penguins are fun to visualize! For example:



→

https://github.com/allisonhorst/palmerpenguins/blob/master/README.Rmd

```
## Examples

You can find these and more code examples for exploring palmerpenguins in `vignette("examples")`.

Penguins are fun to summarize! For example:

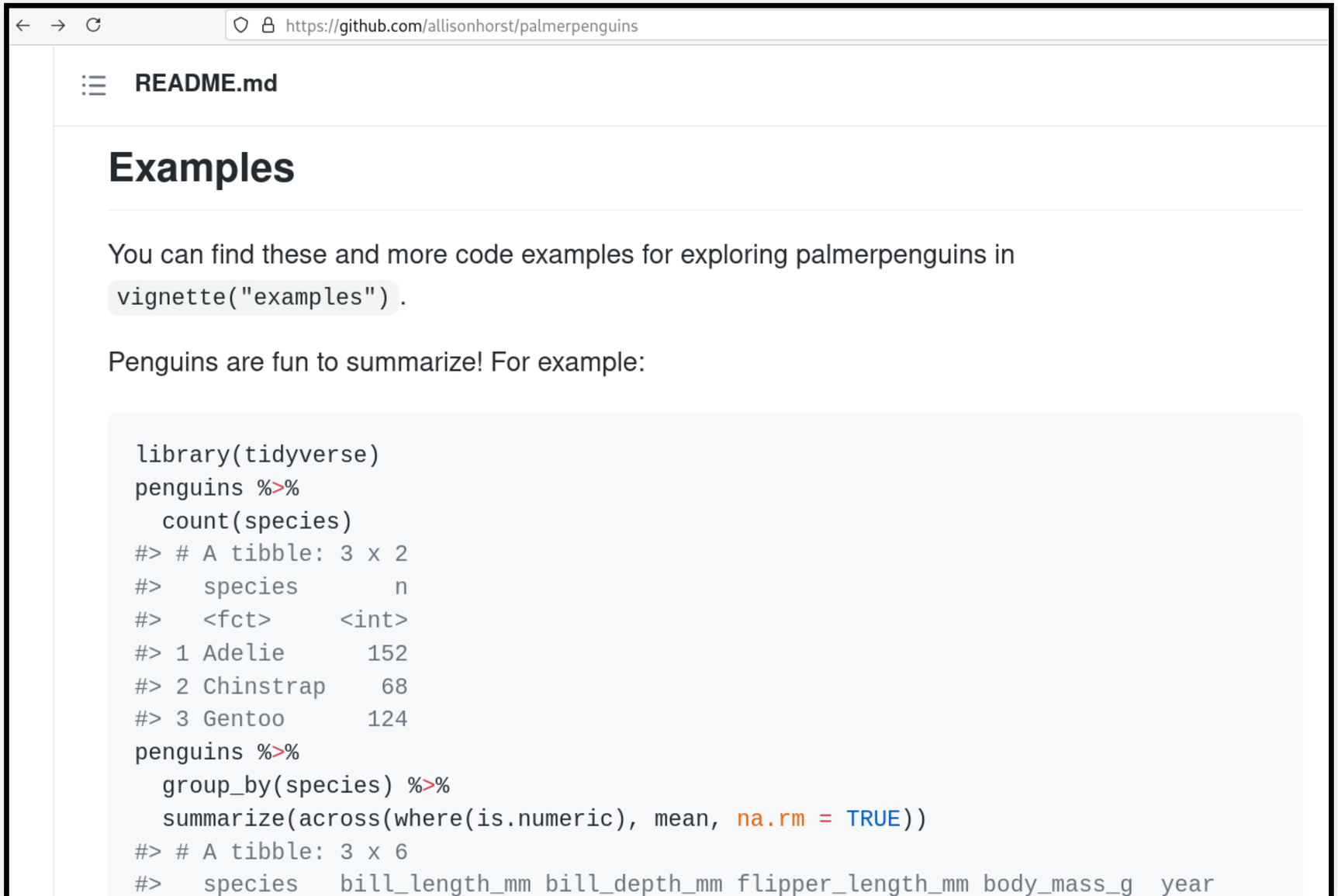
```{r example, warning=FALSE, message=FALSE}
library(tidyverse)
penguins %>%
  count(species)
penguins %>%
  group_by(species) %>%
  summarize(across(where(is.numeric), mean, na.rm = TRUE))
```

Penguins are fun to visualize! For example:

```{r mass-flipper, warning = FALSE, message = FALSE, echo = FALSE, out.width='75%', fig.retina=2}
mass_flipper <- ggplot(data = penguins,
                       aes(x = flipper_length_mm,
                           y = body_mass_g)) +
  geom_point(aes(color = species,
                 shape = species),
             size = 3,
             alpha = 0.8) +
  theme_minimal() +
  scale_color_manual(values = c("darkorange","purple","cyan4")) +
  labs(title = "Penguin size, Palmer Station LTER",
       subtitle = "Flipper length and body mass for Adelie, Chinstrap, and Gentoo Penguins",
       x = "Flipper length (mm)",
       y = "Body mass (g)",
       color = "Penguin species",
       shape = "Penguin species") +
  theme(legend.position = c(0.2, 0.7),
        legend.background = element_rect(fill = "white", color = NA),
        plot.title.position = "plot",
        plot.caption = element_text(hjust = 0, face= "italic"),
        plot.caption.position = "plot")

mass_flipper
```
```

Zooming in


https://github.com/allisonhorst/palmerpenguins
README.md
Examples
You can find these and more code examples for exploring palmerpenguins in vignette("examples").
Penguins are fun to summarize! For example:
library(tidyverse)
penguins %>%
count(species)
#> # A tibble: 3 x 2
#> species n
#> <fct> <int>
#> 1 Adelie 152
#> 2 Chinstrap 68
#> 3 Gentoo 124
penguins %>%
group_by(species) %>%
summarize(across(where(is.numeric), mean, na.rm = TRUE))
#> # A tibble: 3 x 6
#> species bill_length_mm bill_depth_mm flipper_length_mm body_mass_g year

Zooming in

https://github.com/allisonhorst/palmerpenguins/blob/master/README.Rmd

````
## Examples

You can find these and more code examples for exploring palmerpenguins in `vignette("examples")`.

Penguins are fun to summarize! For example:

```{r example, warning=FALSE, message=FALSE}
library(tidyverse)
penguins %>%
  count(species)
penguins %>%
  group_by(species) %>%
  summarize(across(where(is.numeric), mean, na.rm = TRUE))
```

Penguins are fun to visualize! For example:

```{r mass-flipper, warning = FALSE, message = FALSE, echo = FALSE, out.width='75%', fig.retina=2}
mass_flipper <- ggplot(data = penguins,
                       aes(x = flipper_length_mm,
                           y = body_mass_g)) +
  geom_point(aes(color = species,
                 shape = species),
             size = 3,
             alpha = 0.8) +
  theme_minimal() +
````
````

R markdown code blocks

- Almost identical to markdown code blocks
- Start with ```{r ...} instead of ``` or ```r
- Optionally add chunk name and options:

````
```{r mass-flipper, warning = FALSE, message = FALSE,
    echo = FALSE, out.width='75%', fig.retina=2}
```
````

Questions?

- Let’s try out markdown in Hedgedoc (link in the chat)

Links

Markdown

- Basic syntax
- Extended syntax
- Github-flavored markdown
- Mathjax

R Markdown

- RStudio's markdown guide
- Knitr
  - Not just R
  - Not just markdown
- Pandoc
  - Citeproc