Dynamic Documents 1

Intro to R Markdown

Daniel Anderson
Week 4, Class 1

UNIVERSITY OF OREGON

Slides available at: http://www.datalorax.com/vita/ds/ds1-slides/w4p1/

Agenda

- Questions
- R Markdown
- Lab

Agenda

- Questions
- R Markdown
- Lab

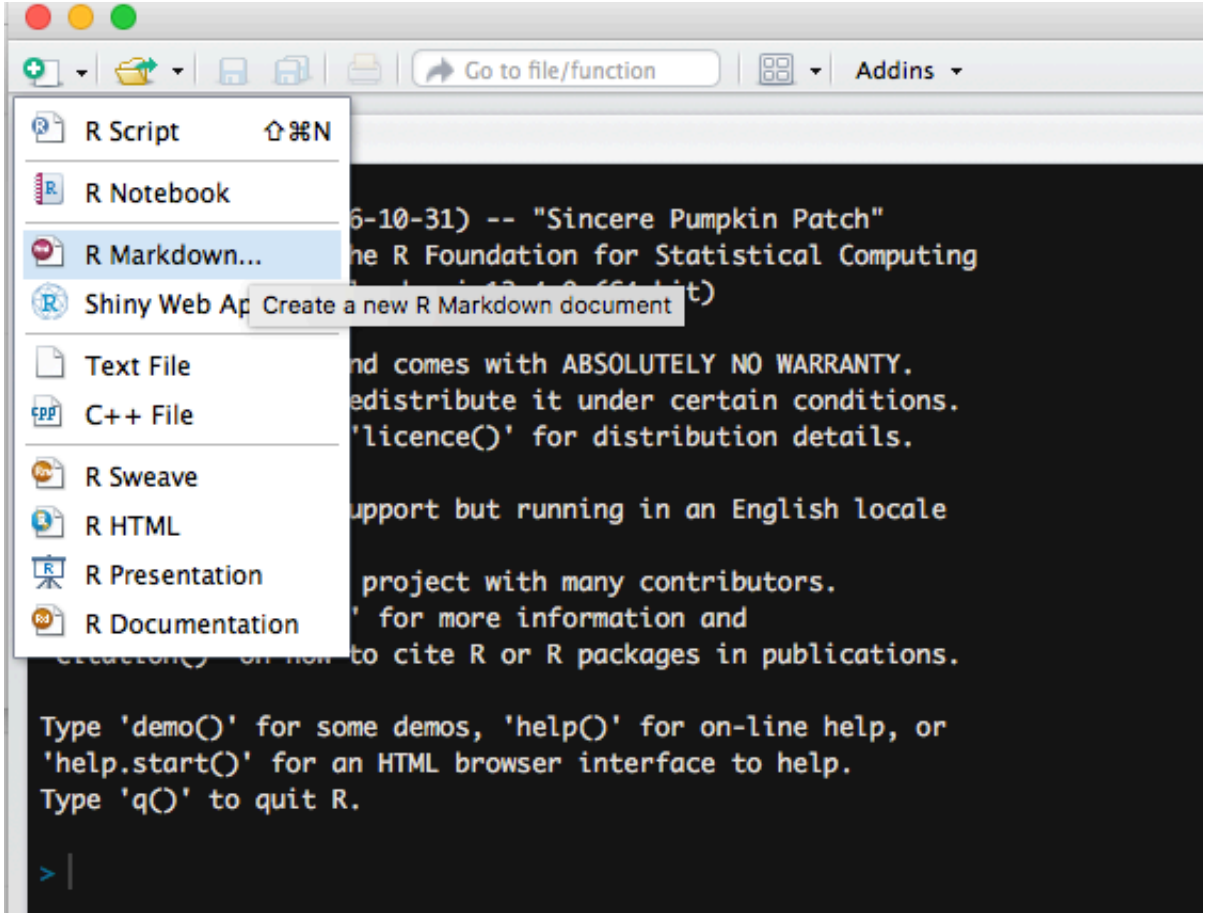
Learning objectives for today

- Understand how to render R Markdown documents & mix code with text
- Understand different chunk options
- Understand inline code evaluation

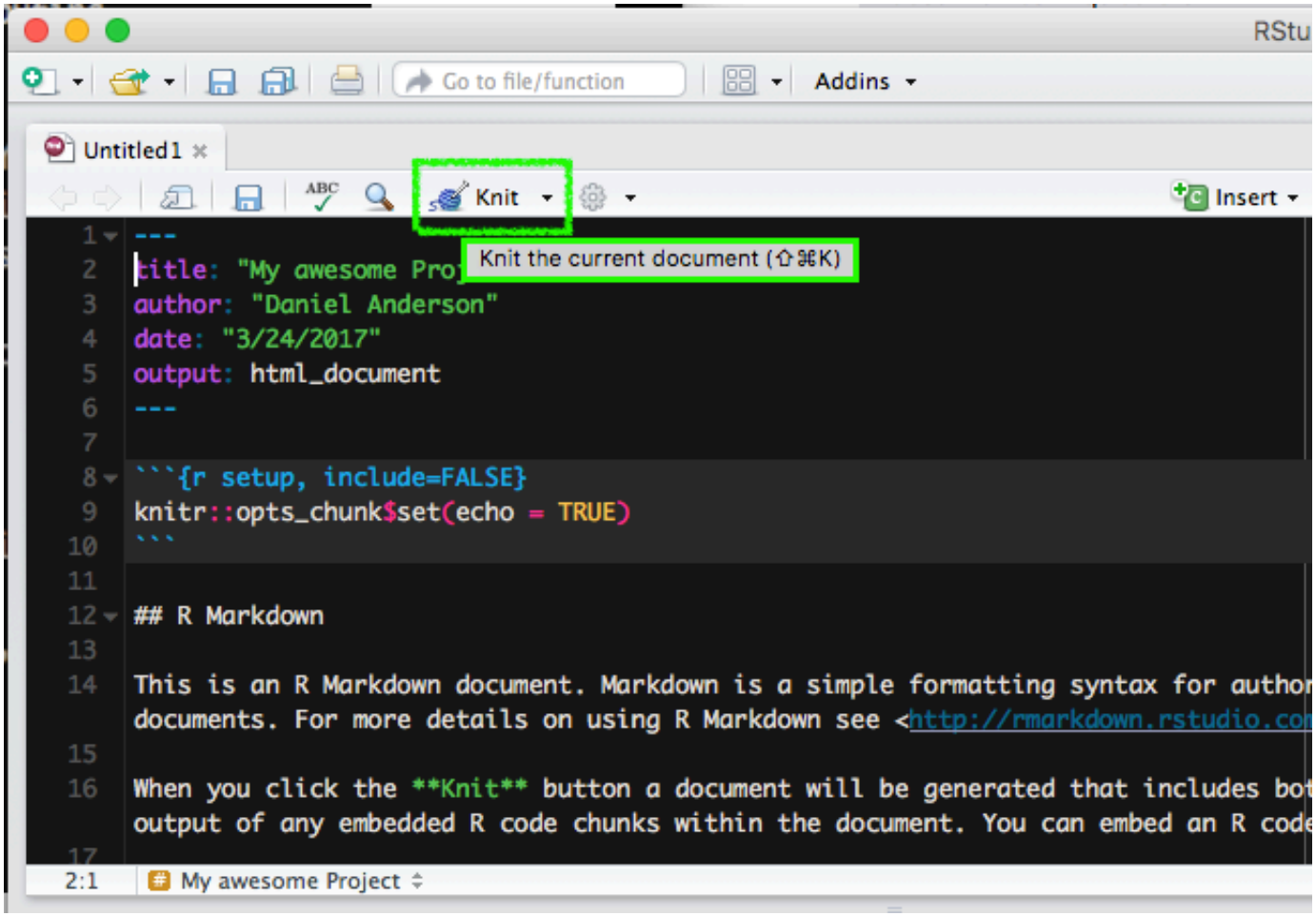
What questions do you have?

R Markdown

From within your R Studio Project:



First thing: Render!



YAML Front Matter

```
---
title: Example Markdown document
author: Daniel Anderson
date: "2015-09-17"
---
```

- Three dashes before and after the YAML fields
- Case sensitive
- Many other fields are possible.
    - For example, you may want to include an `output:` argument (`pdf_document`, `html_document`, `word_document`). Must be specified as it is rendered, if not supplied.

Code chunks versus text

```
---
title: "My awesome Project"
author: "Daniel Anderson"
date: "3/24/2017"
output: html_document
---

## R Markdown

Text here

```{r cars}
# Code here
summary(cars)
```

More text here

```{r pressure, echo=FALSE}
# More code here
plot(pressure)
```

Yet more text...
```

Code chunks

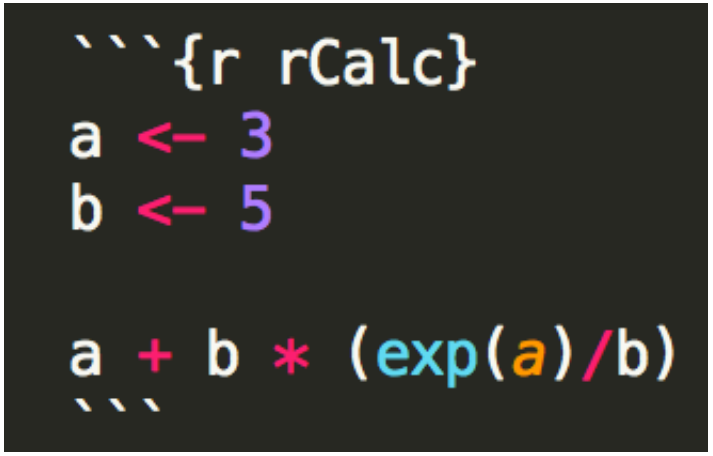
Start a code chunk with ```` ```{r} ````, then produce some r code, then close the chunk with three additional back ticks ```` ``` ````.

````
```{r rCalc}
a <- 3
b <- 5

a + b * (exp(a)/b)
```
````

Code chunks

Start a code chunk with ```` ```{r} ````, then produce some r code, then close the chunk with three additional back ticks ```` ``` ````.




```
a <- 3
b <- 5

a + b * (exp(a)/b)
```

```
## [1] 23.08554
```

Headings and Lists

Not R-lists

```
# Level 1
## Level 2
### Level 3 (etc.)
```

Level 1

Level 2

Level 3 (etc.)

```
 * Unordered list
  - inset
    + inset more
  - etc.

1. Ordered list
  a. blah blah
2. More stuff
```

- Unordered list
  - inset
    - inset more
  - etc.

1. Ordered list
   a. blah blah
2. More stuff

Final Product

ExampleRMarkdown.Rmd

```
---
title: Example Markdown document
author: Daniel Anderson
date: "`r Sys.Date()`"
output: html_document
---

```{r setup, include = FALSE}
library(knitr)

# Set global chunk options
opts_chunk$set(cache = TRUE, cache.comments = FALSE, autodep = TRUE)

# Determine caching dependencies automatically
dep_auto()
```

Level 1

Level 2

Level 3 (etc.)

* Unordered list
  - inset
    + inset more
  - etc.

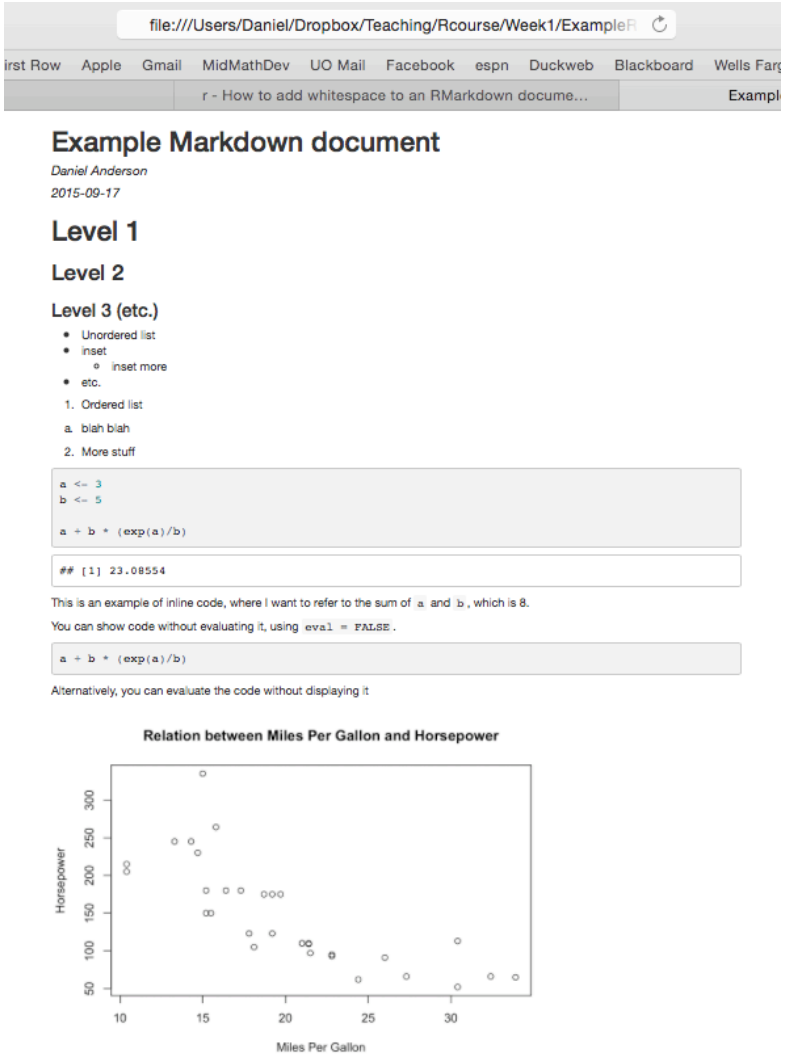
1. Ordered list
   a. blah blah
2. More stuff

```{r ex_rCalc1}
a <- 3
b <- 5

a + b * (exp(a)/b)
```

This is an example of inline code, where I want to refer to the sum of `a` and
  `b`, which is `r a + b`.
```

117 Words, git branch: master, index: 1?, working: 1?, Line 1, Column 1 | 12 misspelled words | Tab Size: 4 | R Markdown



More advanced options

- Chunk options
- Setting global options
- Inline code evaluation

A few select chunk options

Options	Arguments	Default	Result
eval	logical	TRUE	Evaluate the code?
echo	logical	TRUE	Show the code?
results	markup, asis, hold, hide	markup	Render the results
warning	logical	TRUE	Print warnings?
error	logical	TRUE	Preserve errors? (if FALSE, quit)
message	logical	TRUE	Print any messages?
include	logical	TRUE	Include any of the code or output or code?
tidy	logical	FALSE	Tidy code? (see formatR package)

(and a few more)

	Options	Arguments	Default	Result
9	cache	logical, 0:3	FALSE	Cache code chunks?
10	cache.comments	logical	NULL	Cache invalidated by comment changes?
11	dependson	char, num	NULL	Current chunk depend on prior cached chunks?
12	autodep	logical	FALSE	Depends determined automatically?
13	fig.height/fig.width	numeric	7, 7	Height and width of figure
14	fig.show	asis, hold, animate, hide	asis	How the figure should be displayed
15	interval	numeric	1	Animate speed

For complete documentation, see http://yihui.name/knitr/options/

echo and eval

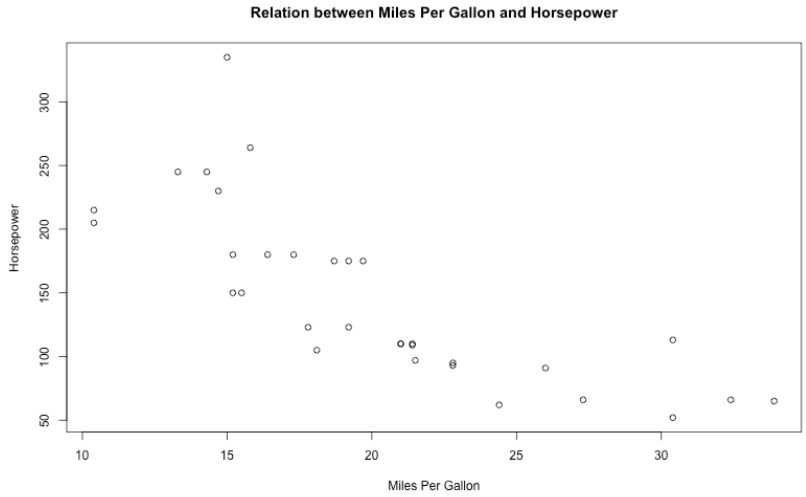
You can show code without evaluating it, using `eval = FALSE`.

````
```{r ex_rCalc2, eval = FALSE}
a + b * (exp(a)/b)
```
````

```
a + b * (exp(a)/b)
```

Alternatively, you can evaluate the code without displaying it, using `echo = FALSE`.

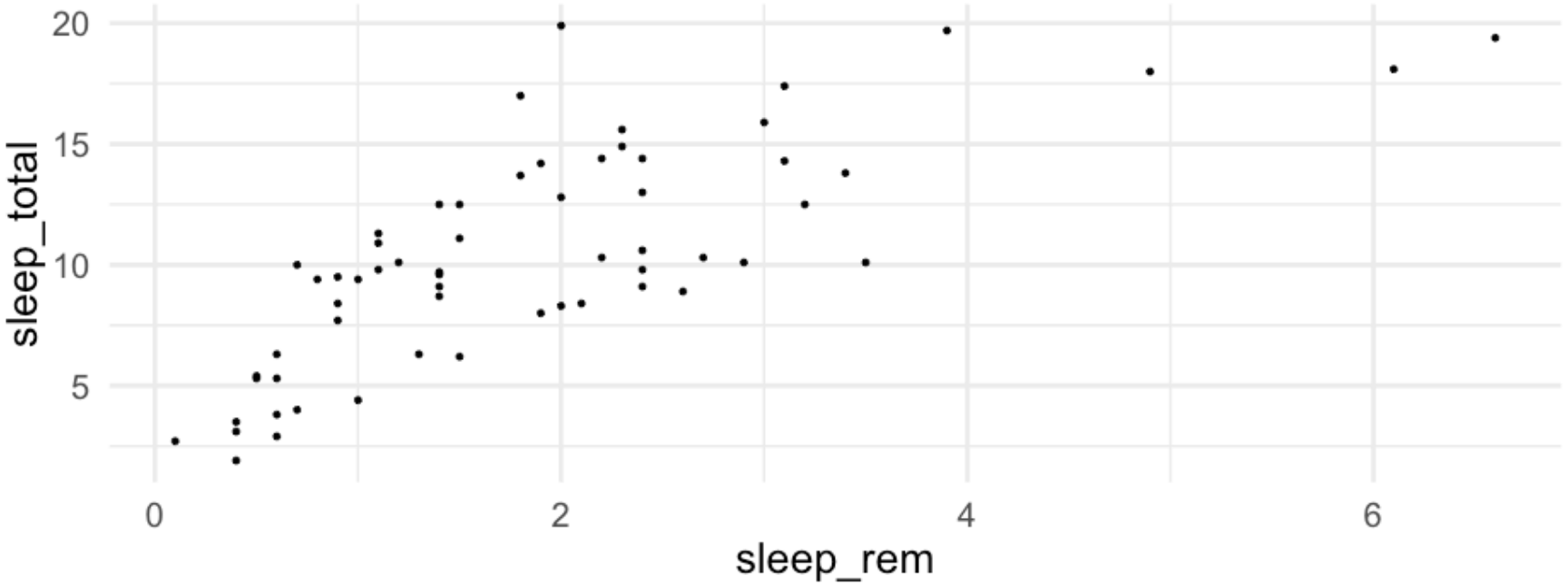
````
```{r plotExample, echo = FALSE, fig.width = 6, fig.height = 3.8}
data(mtcars)
with(mtcars, plot(mpg, hp,
  xlab = "Miles Per Gallon",
  ylab = "Horsepower",
  main = "Relation between Miles Per Gallon and Horsepower"))
```
````



warning

Warning = FALSE

```
ggplot(msleep, aes(sleep_rem, sleep_total)) +
  geom_point()
```

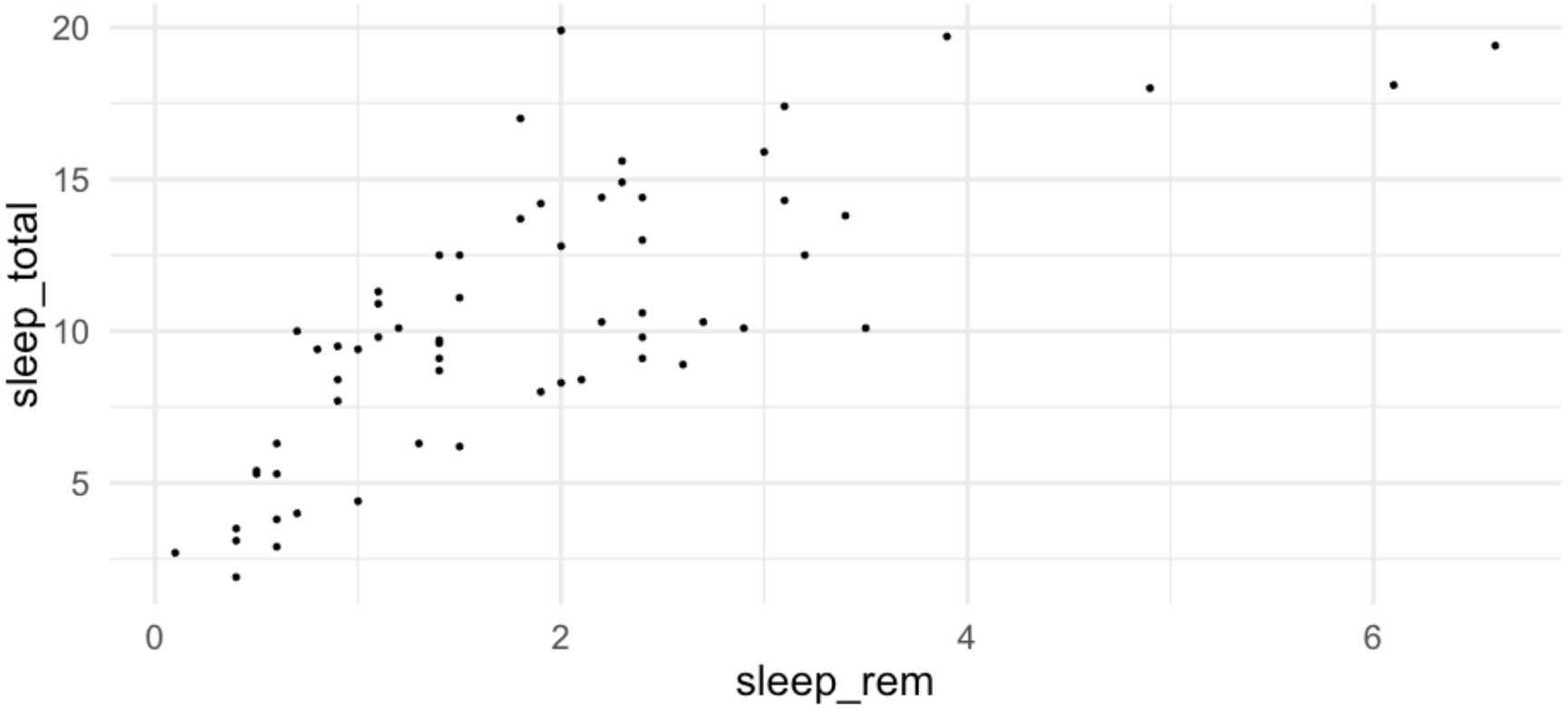


Warning is printed to the console when rendering.

Warning = TRUE

```
ggplot(msleep, aes(sleep_rem, sleep_total)) +
  geom_point()
```

```
## Warning: Removed 22 rows containing missing values (geom_point).
```



Show errors

`error = TRUE`

```
ggplot(msleep, aes(sleep, sleep_total)) +
  geom_point()
```

```
## Error: Aesthetics must be either length 1 or the same as the data (83): x
```

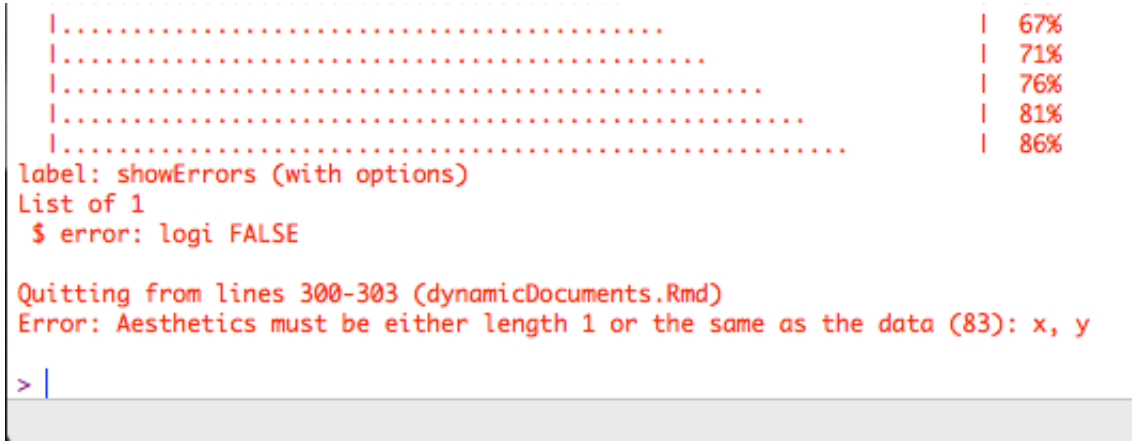
Show errors

`error = TRUE`

```
ggplot(msleep, aes(sleep, sleep_total)) +
  geom_point()
```

```
## Error: Aesthetics must be either length 1 or the same as the data (83): x
```

If `error = FALSE`, the document won't render if it encounters an error.

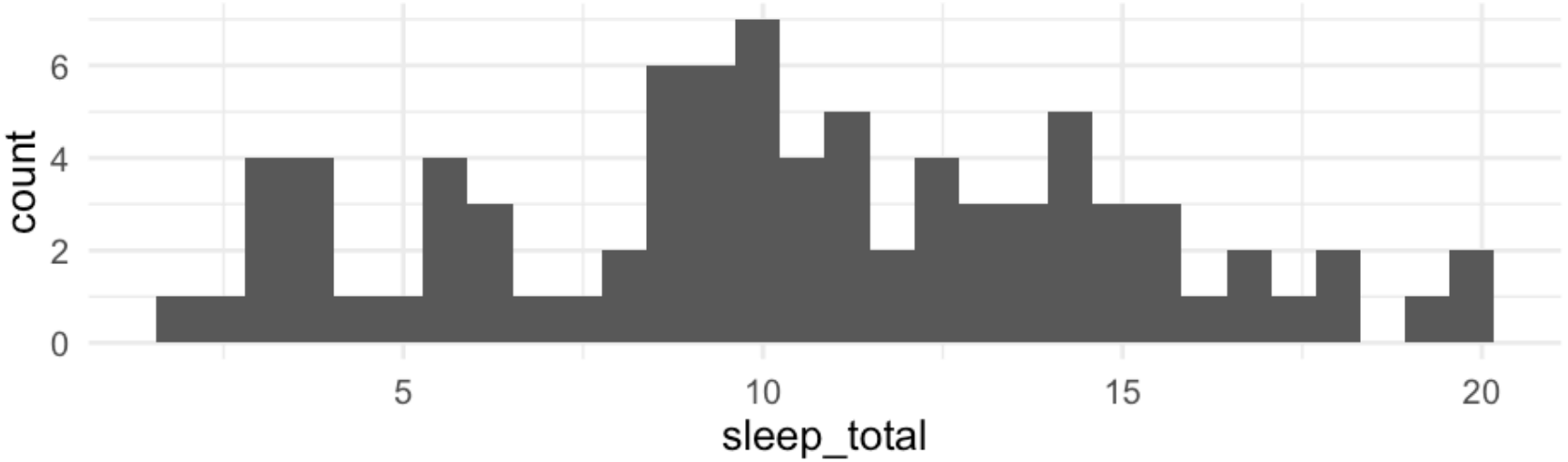


Message

Some functions will return messages. You may want to suppress these.

message = FALSE

```
ggplot(msleep, aes(sleep_total)) +
  geom_histogram()
```

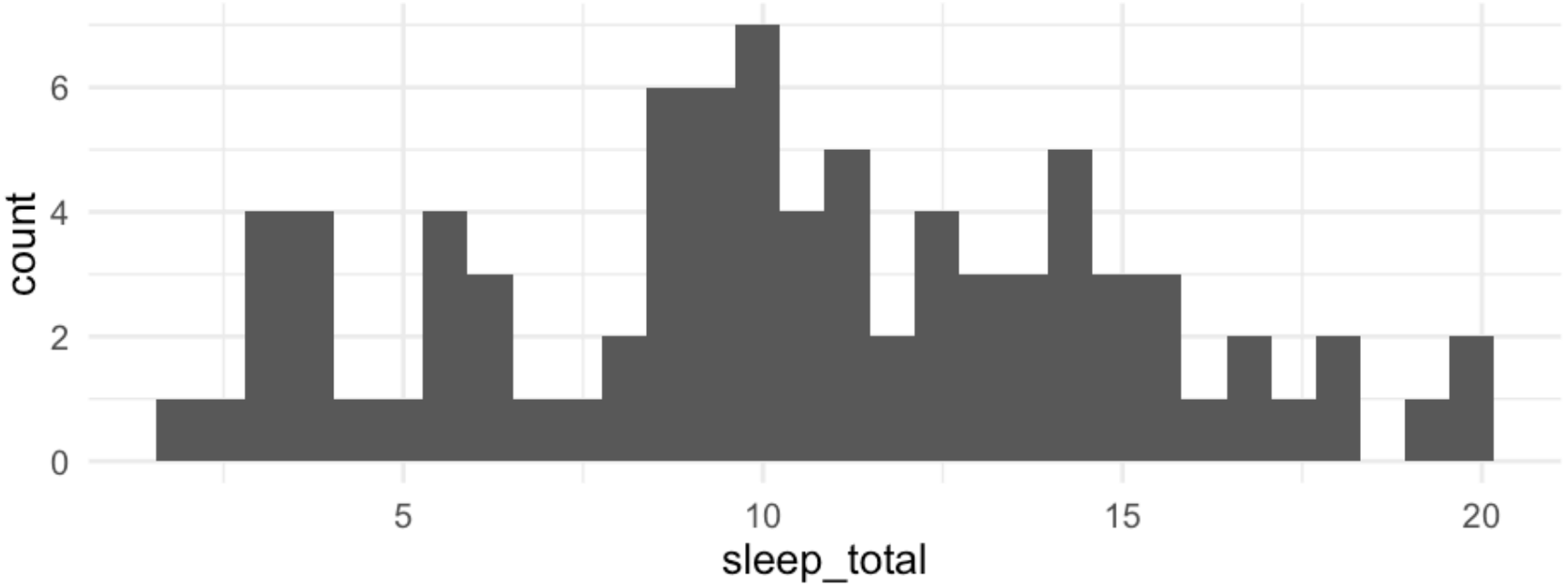


Message

message = TRUE

```
ggplot(msleep, aes(sleep_total)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



include

```
```{r setup, include = FALSE}
library(knitr)

# Set global chunk options
opts_chunk$set(cache = TRUE, cache.comments = FALSE, autodep = TRUE)

dep_auto()
```
```

The `include` argument is used to evaluate code that is not included in the document at all. For example, when setting up your global options.

Setting global options

Change the default behavior

```
opts_chunk$set(options)
```

For example, you can set `echo = FALSE` and `fig.width = 6.5` and `fig.height = 8` with the following code.

```
opts_chunk$set(echo = FALSE, fig.width = 6.5, fig.height = 8)
```

This is most useful when producing a report for somebody who doesn't use R and has no use or knowledge of the code.

You can always override the global options within a particular chunk, e.g.

````
```{r, chunkName, echo = TRUE}

```
````

Other things to consider setting globally:

- `warnings = FALSE`
- `message = FALSE`
- `errors = TRUE`
- `echo = FALSE`
- Caching options (next slides)

Inline code

A single back tick followed by `r` produces inline code to be evaluated.

```
This is an example of inline code, where I want to refer to the sum of `a` and
  `b`, which is `r a + b`.
```

This is an example of inline code, where I want to refer to the sum of `a` and `b`, which is 8.

This is *extremely* useful in writing reports. Never have to update any numbers in text, regardless of changes to your models or data (if you are careful about it).

Real example

```
```{r ell_grp_means}
ell_means <- with(d, tapply(reading, ell, mean, na.rm = TRUE))
```

Figure 2 represents a example of why the PP plot is such a powerful
visualization of group differences. As previously stated, the *Monitor* group
had the highest average achievement, scoring `r round(ell_means[2] - ell_means[
  3], 2)` points higher than *Non-ELL* students and `r round(ell_means[2] -
  ell_means[1], 2)` points higher than *Active* students, on average. However,
the data visualization provides a more nuanced picture of the achievement
differences. The *Monitor* curve is below the reference line on the lower end
of the scale, but above the reference line at the upper end of the scale. In
other words, students in the *Monitor* group only scored higher than students
in the *Non-ELL* group at the bottom of the scale. At the upper end of the
scale the effect was reversed, and students in the *Non-ELL* group had the
higher achievement. In this case, the line crosses at essentially the 50th
percentile of achievement for non-ELL students. Observing these achievement
differences may help lead us to more refined research questions and research
hypotheses. For example, a reasonable hypothesis stemming from Figure 2 may be
that low-performing students are in need of additional attention, and that
providing additional attention is beneficial to their academic achievement
even if the intervention is not strictly focused on academics. Interestingly,
although not reported here, when the equivalent plot is produced with the
mathematics outcome, the overall group differences appear very similar at the
bottom of the scale, but the groups are essentially indistinguishable above
the 50th percentile.
```

Want to customize further?

The YAML will control a lot of how a document looks. For example, if you wanted to render with a different syntax highlighter:

Standard Rmd

```
---
title: "Doc Title"
output: pdf_document
---
```

kate

```
---
title: "Doc Title"
output:
  pdf_document:
    highlight: kate
---
```

Last bit

Formats

- R Markdown will render to HTML and PDF really well
  - Word kinda-sorta supported
  - You can also create websites, slides (like these), books, etc.

Last bit

Formats

- R Markdown will render to HTML and PDF really well
  - Word kinda-sorta supported
  - You can also create websites, slides (like these), books, etc.
- Need a tex (pronounced tek) distribution
  - My recommendation for this class, and probs everything you'll ever need: {tinytex}

Last bit

Formats

- R Markdown will render to HTML and PDF really well
  - Word kinda-sorta supported
  - You can also create websites, slides (like these), books, etc.
- Need a tex (pronounced tek) distribution
  - My recommendation for this class, and probs everything you'll ever need: {tinytex}

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

This is another amazing package by Yihui Xie. See more about it here

Full distributions

- MacTex: http://www.tug.org/mactex/
- MikTex: https://miktex.org/download
- TexLive: http://www.tug.org/texlive/

Full distributions

- MacTex: http://www.tug.org/mactex/
- MikTex: https://miktex.org/download
- TexLive: http://www.tug.org/texlive/

Modify the YAML

Get the same document to render to different formats by modifying the YAML

```
output:
  pdf_document:
    highlight: kate
  html_document:
    highlight: tango
  word_document: default
```

Lab