Gov 50: 2. R, RStudio, and Rmarkdown

Matthew Blackwell

Harvard University

Roadmap

1. Working in Plain Text
2. Let’s take a touR
3. Using Rmarkdown
4. Getting R bearings
5. Our first visualizations

1/ Working in Plain Text

The two computer revolutions

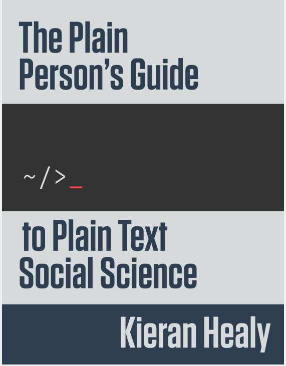
The frontier of computing

- Touch-based interfaces
- Single app at a time
- Little multitasking between apps
- Hides the file system

Where statistical computing lives

- Windows and pointers
- Multi-tasking, multiple windows
- Works heavily with the file system
- Underneath it’s UNIX and the command line

Plain-text tools for data analysis



The Plain
Person's Guide

~/>_

to Plain Text
Social Science

Kieran Healy

- Often free, open-sourced, and powerful.
- Large, friendly communities around them.
- Tons of resources
- But... far from the touch-based paradigm of modern computing
- So why use them?

The process of data science is instrinsically messy

Office vs engineering model of computing

What's real in the project? How are changes managed?

In the Office model

- Formatted documents are real.
- Intermediate ouptuts copy/pasted into documents.
- Changes are tracked inside files.
- Final output is the file you are working on (e.g., Word file or maybe converted to a PDF).

In the Engineering model

- Plain-text files are real.
- Intermediate outputs are produced via code, often inside documents.
- Changes are tracked outside files.
- Final outputs are assembled programatically and converted to desired output format.

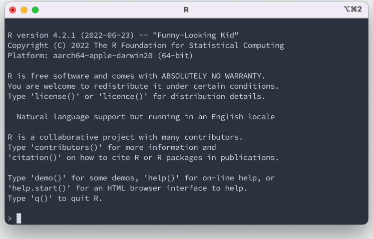
Pros and cons to each approach

- Office model:
    - Everyone knows Word, Excel, Google Docs.
    - “Track changes” is powerful and easy.
    - Wait, how did I make this figure?
    - Which version of my code made this table?
    - `Blackwell_report_final_submitted_edits_FINAL_v2.docx`
- Engineering model:
    - Plain text is universally portable.
    - Push button, recreate analysis.
    - Why won’t R just do what I want!
    - Version control is a pain.
    - `Object of type 'closure' is not subsettable`

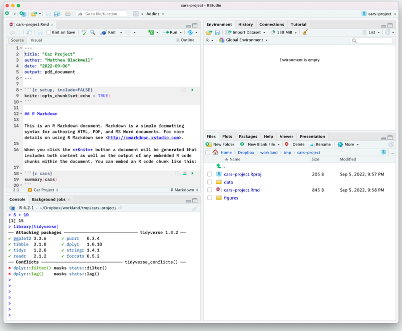
We’ll tend toward the Engineering model because it’s better suited to keep the mess in check.

2/ Let's take a touR

R versus RStudio



```
R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (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.

  Natural language support but running in an English locale

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.

> 
```



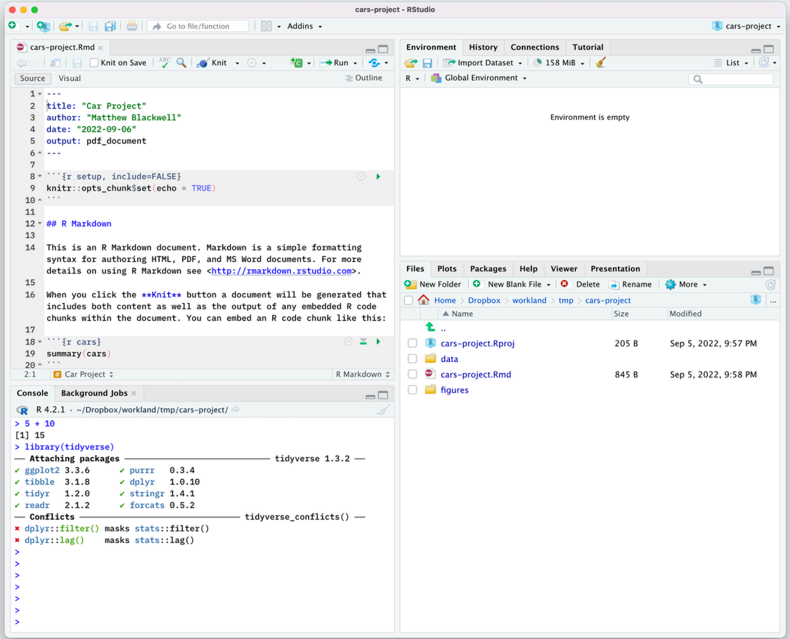
~~~
---
title: "Car Project"
author: "Matthew Blackwell"
date: "2022-09-06"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
~~~

R Markdown

This is an R Markdown document. Markdown is a simple formatting
syntax for authoring HTML, PDF, and MS Word documents. For more
details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that
includes both content as well as the output of any embedded R code
chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```
```

```
> 5 + 10
[1] 15
> library(tidyverse)
── Attaching packages ─────────────────────── tidyverse 1.3.2 ──
✔ ggplot2 3.3.6     ✔ purrr   0.3.4
✔ tibble  3.1.8     ✔ dplyr   1.0.10
✔ tidyr   1.2.0     ✔ stringr 1.4.1
✔ readr   2.1.2     ✔ forcats 0.5.2
── Conflicts ────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
>
>
>
>
>
>
>
```

Environment is empty

Name	Size	Modified
..		
cars-project.Rproj	205 B	Sep 5, 2022, 9:57 PM
data		
cars-project.Rmd	845 B	Sep 5, 2022, 9:58 PM
figures		

```

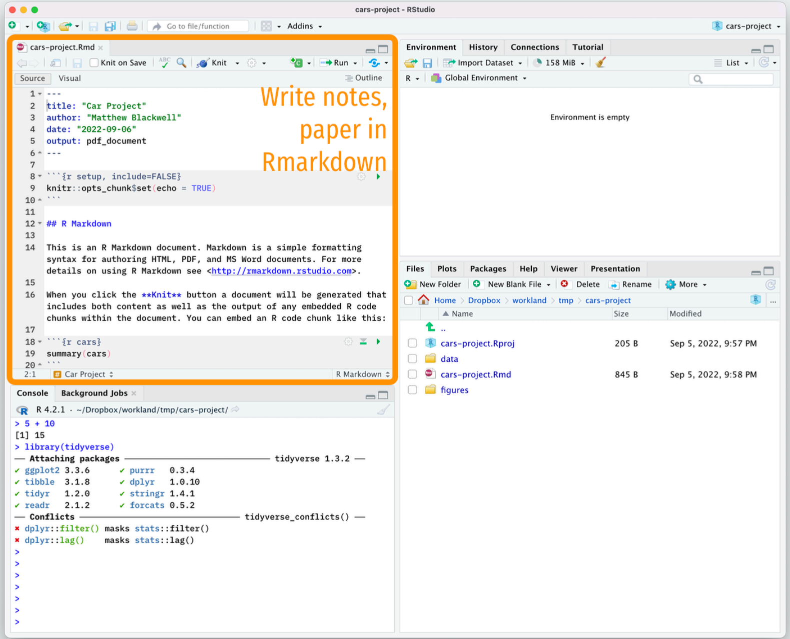


~~~
---
title: "Car Project"
author: "Matthew Blackwell"
date: "2022-09-06"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting
syntax for authoring HTML, PDF, and MS Word documents. For more
details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that
includes both content as well as the output of any embedded R code
chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```
~~~

Environment is empty

Name	Size	Modified
..		
cars-project.Rproj	205 B	Sep 5, 2022, 9:57 PM
data		
cars-project.Rmd	845 B	Sep 5, 2022, 9:58 PM
figures		

```
R 4.2.1 · ~/Dropbox/workland/tmp/cars-project/
> 5 + 10
[1] 15
> library(tidyverse)
── Attaching packages ─────────────────────────────── tidyverse 1.3.2 ──
✔ ggplot2 3.3.6     ✔ purrr   0.3.4
✔ tibble  3.1.8     ✔ dplyr   1.0.10
✔ tidyr   1.2.0     ✔ stringr 1.4.1
✔ readr   2.1.2     ✔ forcats 0.5.2
── Conflicts ────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
>
>
>
>
>
>
>
```



Write notes, paper in Rmarkdown

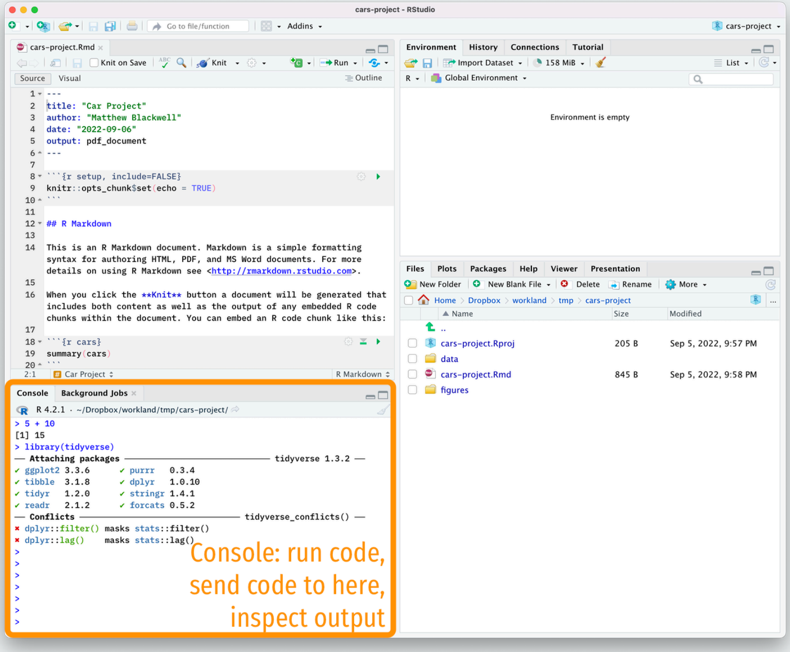
~~~
---
title: "Car Project"
author: "Matthew Blackwell"
date: "2022-09-06"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
~~~

R Markdown

This is an R Markdown document. Markdown is a simple formatting
syntax for authoring HTML, PDF, and MS Word documents. For more
details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that
includes both content as well as the output of any embedded R code
chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```
```

Environment is empty

| | Name | Size | Modified |
|---|---|---|---|
| | .. | | |
| □ | cars-project.Rproj | 205 B | Sep 5, 2022, 9:57 PM |
| □ | data | | |
| □ | cars-project.Rmd | 845 B | Sep 5, 2022, 9:58 PM |
| □ | figures | | |

```
R 4.2.1 · ~/Dropbox/workland/tmp/cars-project/
> 5 + 10
[1] 15
> library(tidyverse)
── Attaching packages ─────────────────────────── tidyverse 1.3.2 ──
✔ ggplot2 3.3.6   ✔ purrr   0.3.4
✔ tibble 3.1.8   ✔ dplyr   1.0.10
✔ tidyr   1.2.0   ✔ stringr 1.4.1
✔ readr   2.1.2   ✔ forcats 0.5.2
── Conflicts ────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()   masks stats::lag()
>
>
>
>
>
>
>
```
```



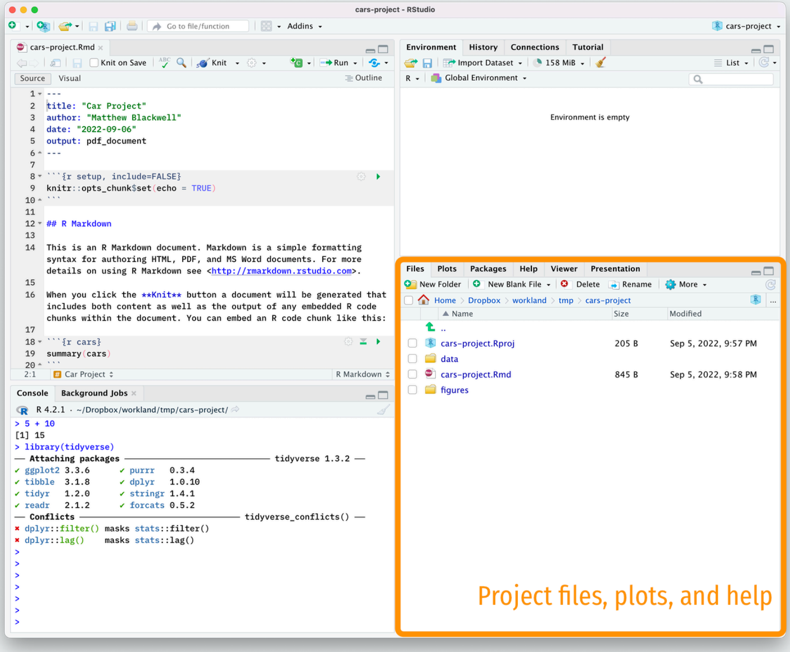
~~~
---
title: "Car Project"
author: "Matthew Blackwell"
date: "2022-09-06"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
~~~

R Markdown

This is an R Markdown document. Markdown is a simple formatting
syntax for authoring HTML, PDF, and MS Word documents. For more
details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that
includes both content as well as the output of any embedded R code
chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```
```

Name	Size	Modified
..		
cars-project.Rproj	205 B	Sep 5, 2022, 9:57 PM
data		
cars-project.Rmd	845 B	Sep 5, 2022, 9:58 PM
figures		

```
> 5 + 10
[1] 15
> library(tidyverse)
── Attaching packages ─────────────────────────── tidyverse 1.3.2 ──
✔ ggplot2 3.3.6     ✔ purrr   0.3.4
✔ tibble  3.1.8     ✔ dplyr   1.0.10
✔ tidyr   1.2.0     ✔ stringr 1.4.1
✔ readr   2.1.2     ✔ forcats 0.5.2
── Conflicts ────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
>
>
>
>
>
>
>
```

Console: run code, send code to here, inspect output
```

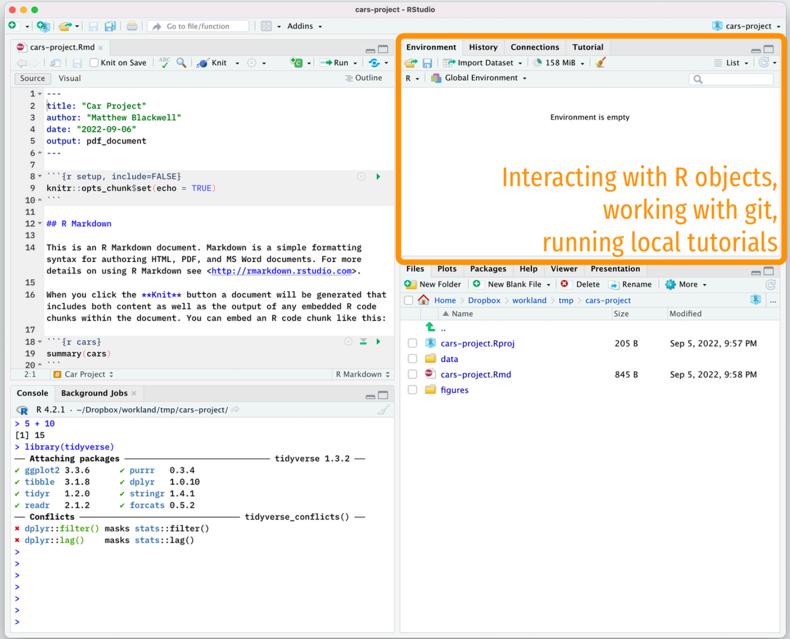


~~~
---
title: "Car Project"
author: "Matthew Blackwell"
date: "2022-09-06"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting
syntax for authoring HTML, PDF, and MS Word documents. For more
details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that
includes both content as well as the output of any embedded R code
chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```
~~~

```
R 4.2.1 · ~/Dropbox/workland/tmp/cars-project/
> 5 + 10
[1] 15
> library(tidyverse)
── Attaching packages ─────────────────────────── tidyverse 1.3.2 ──
✔ ggplot2 3.3.6      ✔ purrr   0.3.4
✔ tibble  3.1.8      ✔ dplyr   1.0.10
✔ tidyr   1.2.0      ✔ stringr 1.4.1
✔ readr   2.1.2      ✔ forcats 0.5.2
── Conflicts ────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
>
>
>
>
>
>
>
```

Name	Size	Modified
..		
cars-project.Rproj	205 B	Sep 5, 2022, 9:57 PM
data		
cars-project.Rmd	845 B	Sep 5, 2022, 9:58 PM
figures		

Project files, plots, and help



~~~
---
title: "Car Project"
author: "Matthew Blackwell"
date: "2022-09-06"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
~~~

R Markdown

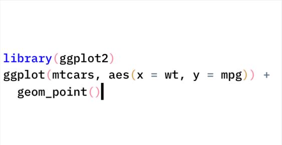
This is an R Markdown document. Markdown is a simple formatting
syntax for authoring HTML, PDF, and MS Word documents. For more
details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that
includes both content as well as the output of any embedded R code
chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```
```

```
> 5 + 10
[1] 15
> library(tidyverse)
── Attaching packages ─────────────────────────── tidyverse 1.3.2 ──
✔ ggplot2 3.3.6      ✔ purrr   0.3.4
✔ tibble  3.1.8      ✔ dplyr   1.0.10
✔ tidyr   1.2.0      ✔ stringr 1.4.1
✔ readr   2.1.2      ✔ forcats 0.5.2
── Conflicts ────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
>
>
>
>
>
>
>
```

Name	Size	Modified
..		
cars-project.Rproj	205 B	Sep 5, 2022, 9:57 PM
data		
cars-project.Rmd	845 B	Sep 5, 2022, 9:58 PM
figures		

Interacting with R objects,
working with git,
running local tutorials
```

3/ Using Rmarkdown

The acts of coding



```r
library(ggplot2)
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point()
```

Figure: 1. Writing code

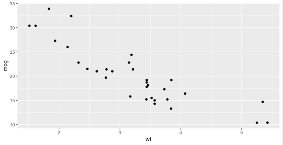
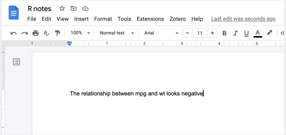


Figure: 2. Looking at output

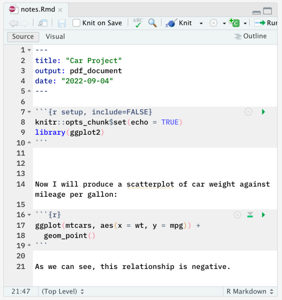


The relationship between mpg and wt looks negative

Figure: 3. Taking notes

How to do all of these efficiently?

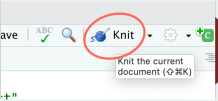
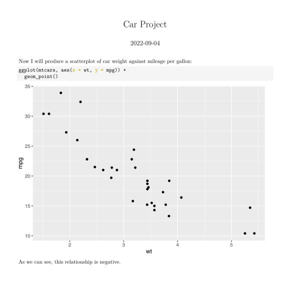
Rmarkdown files to the rescue



~~~
---
title: "Car Project"
output: pdf_document
date: "2022-09-04"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
~~~

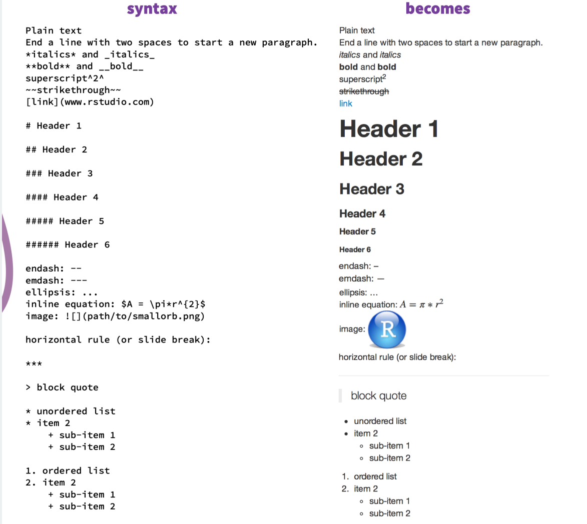
Now I will produce a scatterplot of car weight against
mileage per gallon:

```{r}
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point()
```

As we can see, this relationship is negative.
```

Figure: Rmarkdown file

Keep code and notes together in plain text


ave
Knit
Knit the current document (⇧⌘K)

Figure: Knit in R



Car Project

2022-09-04

Now I will produce a scatterplot of car weight against mileage per gallon:

```
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point()
```

As we can see, this relationship is negative.

Figure: PDF output

Produce nice-looking outputs in different formats
```

# Markdown: formatting in plain text

Non-code text in Rmd files is plain text with formatting instructions



**syntax**

```
Plain text
End a line with two spaces to start a new paragraph.
*italics* and _italics_
**bold** and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6

endash: --
emdash: ---
ellipsis: ...
inline equation: $A = \pi*r^{2}$
image: ![](path/to/smallorb.png)

horizontal rule (or slide break):

***

> block quote

* unordered list
* item 2
    + sub-item 1
    + sub-item 2

1. ordered list
2. item 2
    + sub-item 1
    + sub-item 2
```

becomes

Plain text
End a line with two spaces to start a new paragraph.
italics and *italics*
bold and **bold**
superscript^2
~~strikethrough~~
link

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

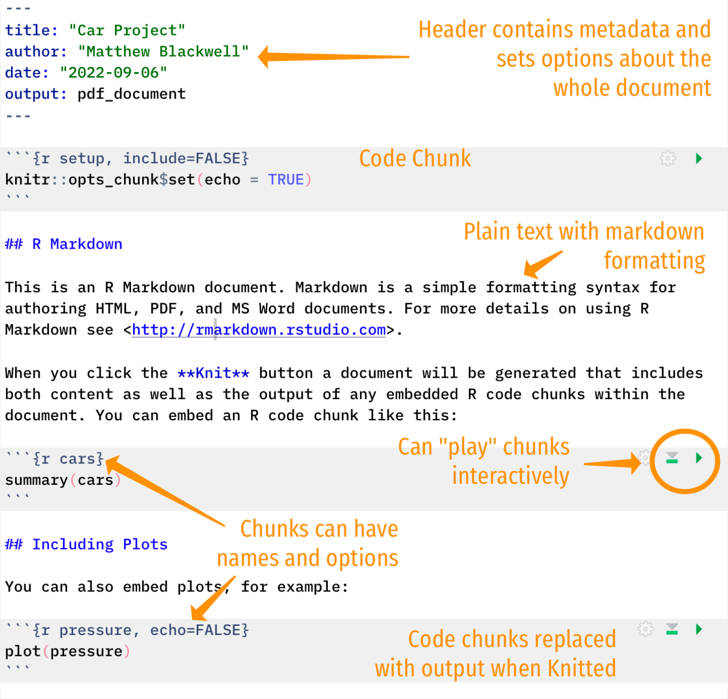
endash: –
emdash: —
ellipsis: …
inline equation: $A = \pi * r^2$
image:

horizontal rule (or slide break):

> block quote

- unordered list
- item 2
  - sub-item 1
  - sub-item 2

1. ordered list
2. item 2
   - sub-item 1
   - sub-item 2



~~~
---
title: "Car Project"
author: "Matthew Blackwell"
date: "2022-09-06"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for
authoring HTML, PDF, and MS Word documents. For more details on using R
Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes
both content as well as the output of any embedded R code chunks within the
document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```
~~~

Header contains metadata and sets options about the whole document

Code Chunk

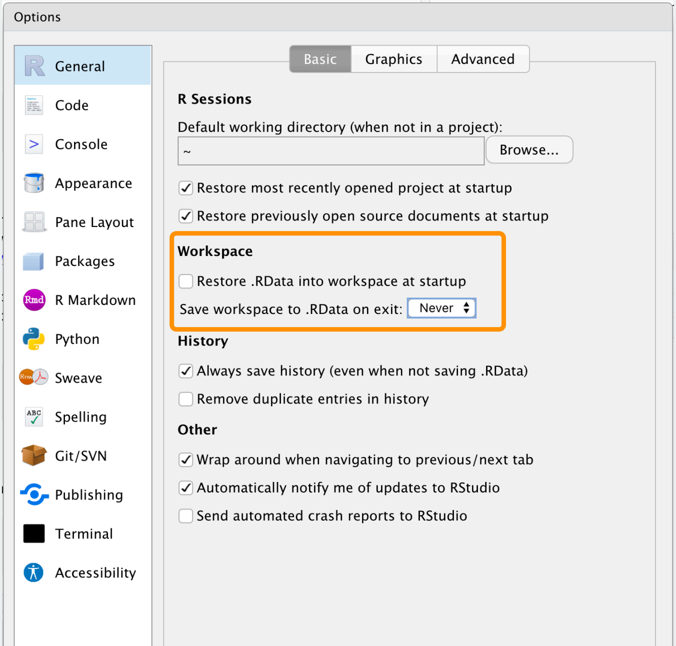
Plain text with markdown formatting

Can "play" chunks interactively

Chunks can have names and options

Code chunks replaced with output when Knitted

Remember what’s real



R Sessions

Default working directory (when not in a project): ~

☑ Restore most recently opened project at startup

☑ Restore previously open source documents at startup

Workspace

□ Restore .RData into workspace at startup

Save workspace to .RData on exit: Never

History

☑ Always save history (even when not saving .RData)

□ Remove duplicate entries in history

Other

☑ Wrap around when navigating to previous/next tab

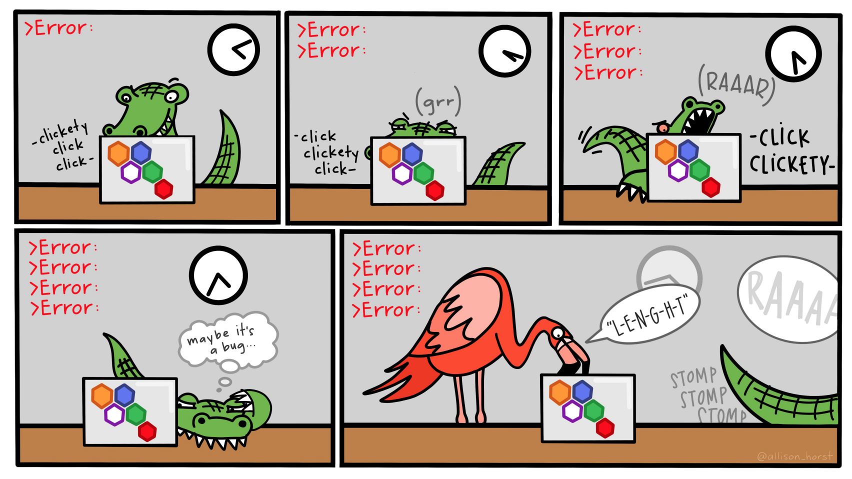
☑ Automatically notify me of updates to RStudio

□ Send automated crash reports to RStudio

4/ Getting R bearings

Try to type your code by hand

Typing speeds up the try-fail cycle



Physically typing the code is best way to familiarize yourself with R and the try-fail-try-fail-try-succeed cycle

Credit: Allison Horst

What R looks like

Code that you can type and run:

```
## Any R code that begins with the # character is a comment
## Comments are ignored by R

my_numbers <- c(4, 8, 15, 16, 23, 42) # Anything after # is also a comment
```

Output from code prefixed by ## by convention:

```
my_numbers
```

```
## [1]  4  8 15 16 23 42
```

Output also has a counter in brackets when over one line:

```
letters
```

```
##  [1] ”a” ”b” ”c” ”d” ”e” ”f” ”g” ”h” ”i” ”j” ”k” ”l”
## [13] ”m” ”n” ”o” ”p” ”q” ”r” ”s” ”t” ”u” ”v” ”w” ”x”
## [25] ”y” ”z”
```

Everything in R has a name

```r
my_numbers # just created this
```

```
## [1]  4  8 15 16 23 42
```

```r
letters # this is built into R
```

```
##  [1] ”a” ”b” ”c” ”d” ”e” ”f” ”g” ”h” ”i” ”j” ”k” ”l”
## [13] ”m” ”n” ”o” ”p” ”q” ”r” ”s” ”t” ”u” ”v” ”w” ”x”
## [25] ”y” ”z”
```

```r
pi # also built in
```

```
## [1] 3.14
```

Some names are forbidden (NA, TRUE, FALSE, etc) or strongly not recommended (c, mean, table)

We do things in R with functions

Functions take in objects, perform actions, and return outputs:

```
mean(x = my_numbers)
```

```
## [1] 18
```

- `x` is the argument name,
- `my_numbers` is what we're passing to the that argument

If you omit the argument name, R will assume the default order:

```
mean(my_numbers)
```

```
## [1] 18
```

Getting help with R

How do we know the default argument order? Look to help files:

```
help(mean)
?mean # shorter
```

- Sometimes inscrutable, so look elsewhere:
  - Google, StackOverflow, Twitter, RStudio Community.
  - Ask on Ed or on class Slack.
  - Come to section, office hours, study hall.
- Get help **early** before becoming too frustrated!
  - Easy to overlook small issues like missing commas, etc.

Functions live in packages

Packages are bundles of functions written by other users that we can use.

Install packages using `install.packages()` to have them on your machine:

```r
install.packages(”ggplot2”)
```

Load them into your R session with `library()`:

```r
library(ggplot2)
```

Now we can use any function provided by `ggplot2`.

Functions live in packages

We can also use the `mypackage::` prefix to access package functions without loading:

```r
knitr::kable(head(mtcars))
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.62	16.5	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.88	17.0	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.6	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.21	19.4	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.0	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.46	20.2	1	0	3	1

5/ Our first visualizations

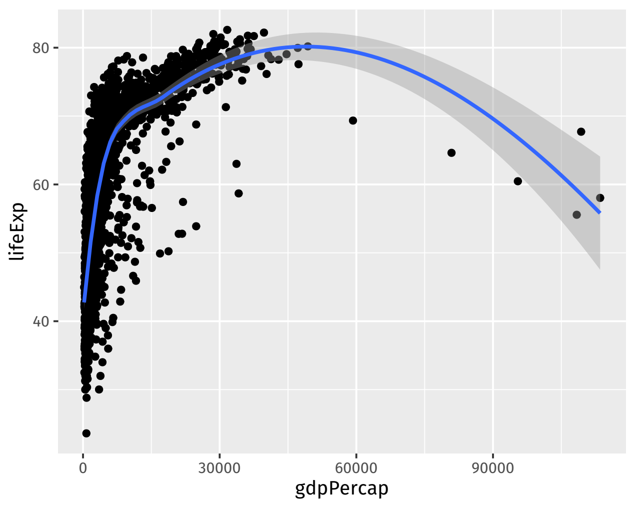
Gapminder data

```
library(gapminder)
gapminder
```

```
## # A tibble: 1,704 x 6
##    country     continent  year lifeExp      pop gdpPe~1
##    <fct>       <fct>     <int>   <dbl>    <int>   <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333    779.
##  2 Afghanistan Asia       1957    30.3  9240934    821.
##  3 Afghanistan Asia       1962    32.0 10267083    853.
##  4 Afghanistan Asia       1967    34.0 11537966    836.
##  5 Afghanistan Asia       1972    36.1 13079460    740.
##  6 Afghanistan Asia       1977    38.4 14880372    786.
##  7 Afghanistan Asia       1982    39.9 12881816    978.
##  8 Afghanistan Asia       1987    40.8 13867957    852.
##  9 Afghanistan Asia       1992    41.7 16317921    649.
## 10 Afghanistan Asia       1997    41.8 22227415    635.
## # ... with 1,694 more rows, and abbreviated variable
## #   name 1: gdpPercap
```

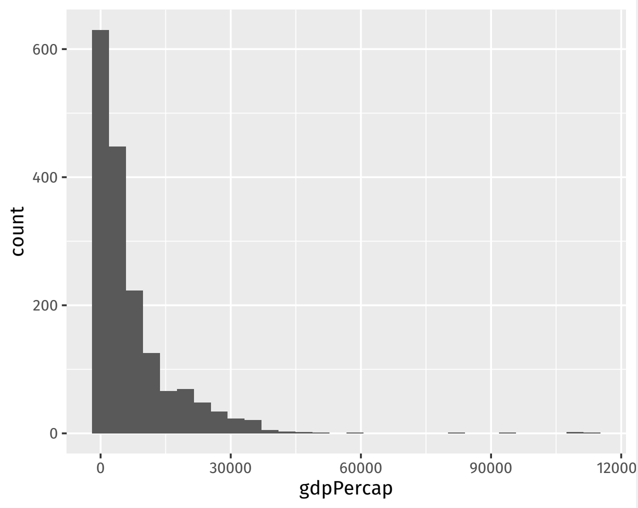
Plotting life expectancy over time

```
ggplot(gapminder, mapping = aes(x = gdpPercap, y = lifeExp)) +
  geom_point() + geom_smooth(method = "loess")
```



A histogram of GDP per capita

```
ggplot(gapminder, mapping = aes(x = gdpPercap)) +
  geom_histogram()
```


count
gdpPercap