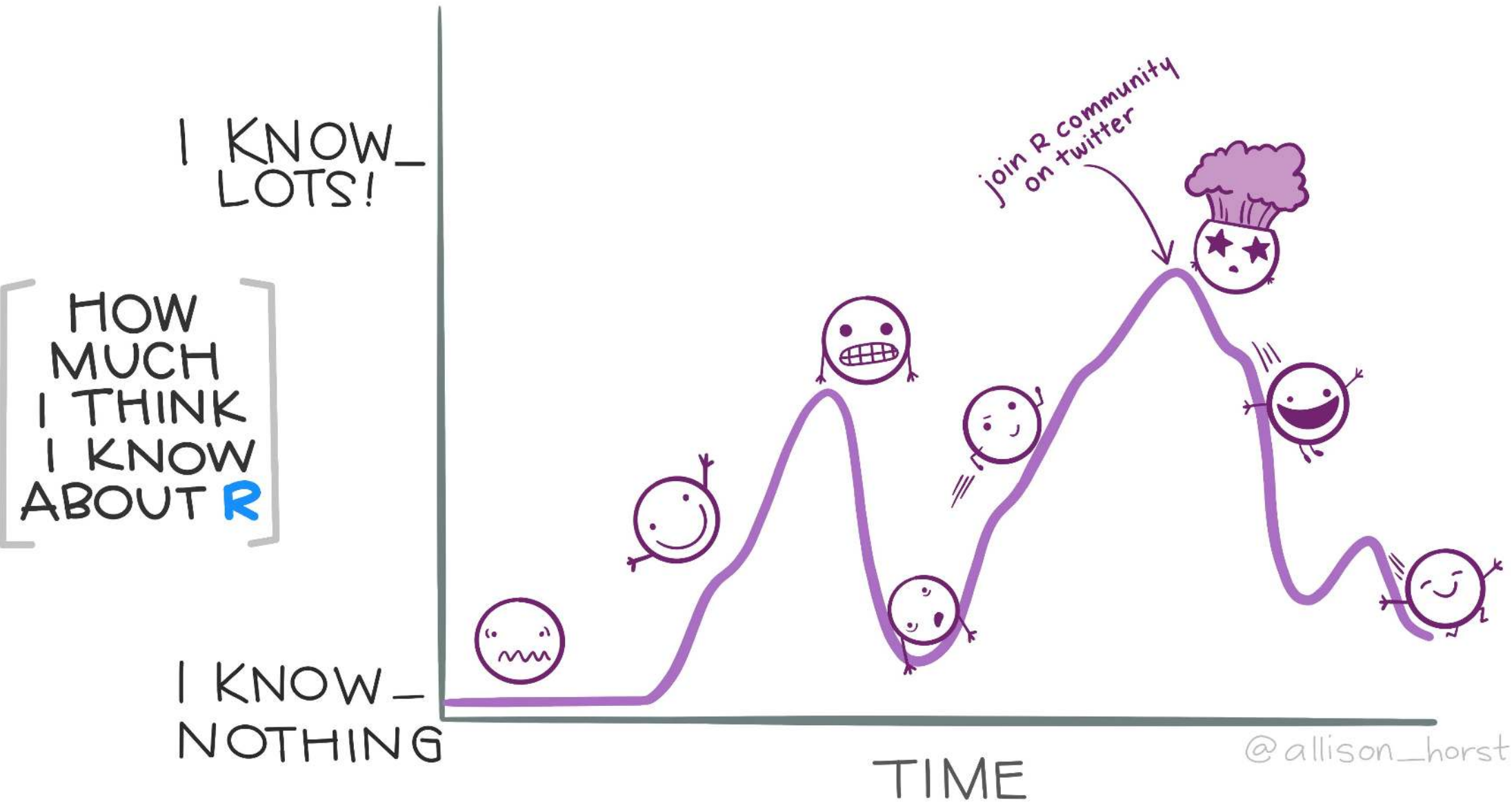


Getting help in R

After this workshop





Troubleshooting

Bit by bit

Line by line

- R is sequential
- If you skip lines, you're not running that part

```
#library(tidyverse)  
count(mtcars, am)
```

```
## Error in count(mtcars, am): could not find function "count"
```

Bit by bit

Line by line

- R is sequential
- If you skip lines, you're not running that part

```
#library(tidyverse)  
count(mtcars, am)
```

```
## Error in count(mtcars, am): could not find function "count"
```

- Error? Start at the beginning and go line by line

```
library(tidyverse)  
count(mtcars, am)
```

```
##      am  n  
## 1    0 19  
## 2    1 13
```

Bit by bit

Line by line

Especially important if loading and modifying data

```
# Load Data
size <- read_csv("./data/grain_size2.csv")

# First modification
size <- mutate(size,
               total_sand = coarse_sand + medium_sand + fine_sand,
               total_silt = coarse_silt + medium_silt + fine_silt)

# Second modification
size <- size %>%
  group_by(plot) %>%
  summarize(n = n(),
            total_sand = sum(total_sand),
            mean_sand = mean(total_sand),
            sd_sand = sd(total_sand),
            se_sand = sd_sand / sqrt(n))
```

Can't run 1st modification
after 2nd modification

Bit by bit

Section by section

```
size <- read_csv("./data/grain_size2.csv") %>%  
  mutate(total_sand = coarse_sand + medium_sand + fine_sand,  
         total_silt = coarse_silt + medium_silt + fine_silt) %>%  
  group_by(plot) %>%  
  summarize(n = n(),  
           total_sand = sum(total_sand),  
           mean_sand = mean(total_sand),  
           sd_sand = sd(total_sand),  
           se_sand = sd_sand / sqrt(n))
```

```
## Error: Problem with `summarise()` input `mean_sand`.  
## x object 'total_sand' not found  
## i Input `mean_sand` is `mean(total_sand)`.  
## i The error occurred in group 1: plot = "CSP01".
```

Bit by bit

Section by section

```
size <- read_csv("./data/grain_size2.csv")
```

No error

Bit by bit

Section by section

```
size <- read_csv("./data/grain_size2.csv")
```

No error

```
size <- read_csv("./data/grain_size2.csv") %>%  
  mutate(total_sand = coarse_sand + medium_sand + fine_sand,  
         total_silt = coarse_silt + medium_silt + fine_silt)
```

No error

Bit by bit

Section by section

```
size <- read_csv("./data/grain_size2.csv")
```

No error

```
size <- read_csv("./data/grain_size2.csv") %>%  
  mutate(total_sand = coarse_sand + medium_sand + fine_sand,  
         total_silt = coarse_silt + medium_silt + fine_silt)
```

No error

```
size <- read_csv("./data/grain_size2.csv") %>%  
  mutate(total_sand = coarse_sand + medium_sand + fine_sand,  
         total_silt = coarse_silt + medium_silt + fine_silt) %>%  
  group_by(plot)
```

No error

Bit by bit

Section by section

```
size <- read_csv("./data/grain_size2.csv") %>%  
  mutate(total_sand = coarse_sand + medium_sand + fine_sand,  
         total_silt = coarse_silt + medium_silt + fine_silt) %>%  
  group_by(plot) %>%  
  summarize(n = n(),  
           total_sand = sum(total_sand),  
           mean_sand = mean(total_sand),  
           sd_sand = sd(total_sand),  
           se_sand = sd_sand / sqrt(n))
```

```
## Error: Problem with `summarise()` input `mean_sand`.  
## x object 'total_sand' not found  
## i Input `mean_sand` is `mean(total_sand)`.  
## i The error occurred in group 1: plot = "CSP01".
```

Ah ha!

debugging



1.
I got this.



2.
Huh. Really
thought that
was it.



3.
(...)



4.
Fine. Restarting.



5.
OH WTF.



6..
Zombie
meltdown



7.



8.
A NEW HOPE!



9.
[insert awesome
theme song]



10.
I ♥ CODING!

R is never wrong

R is never wrong

Just sometimes unhelpful!

Getting Help

Cheat Sheets

RStudio Menu

- Help
 - Cheatsheets

Take a look yourself

Cheat Sheets

RStudio Menu

- Help
 - Cheatsheets

Take a look yourself

Data Visualization with ggplot2 : : CHEAT SHEET



Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and **geoms**—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),  
    stat = <STAT>, position = <POSITION>) +  
  <COORDINATE_FUNCTION> +  
  <SCALE_FUNCTION> +  
  <THEME_FUNCTION>
```

required
Not required, sensible defaults supplied

ggplot(data = mpg, aes(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

qplot(x = cty, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5" x 5" file named "plot.png" in working directory. Matches file type to file extension.

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank()
(Useful for expanding limits)

b + geom_curve(aes(yend = lat + 1, xend = long + 1, curvature = 1) - x, yend, yend, alpha, angle, color, curvature, linetype, size)

a + geom_path(lineend = "butt", linejoin = "round", linemitre = 1)
x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(group = group))
x, y, alpha, color, fill, group, linetype, size

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size)

a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900) - x, ymax, ymin, alpha, color, fill, group, linetype, size)

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:155, radius = 1))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); **c2** <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot
x, y, alpha, color, fill

c + geom_freqpoly(x, y, alpha, color, group, linetype, size)

c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

discrete

d <- ggplot(mpg, aes(f1))

d + geom_bar
x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

continuous x, continuous y

e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_jitter(height = 2, width = 2)
x, y, alpha, color, fill, shape, size

e + geom_point(x, y, alpha, color, fill, shape, size, stroke)

e + geom_quantile(x, y, alpha, color, group, linetype, size, weight)

e + geom_rug(sides = "bl", x, y, alpha, color, linetype, size)

e + geom_smooth(method = lm, x, y, alpha, color, fill, group, linetype, size, weight)

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

discrete x, continuous y

f <- ggplot(mpg, aes(class, hwy))

f + geom_col(x, y, alpha, color, fill, group, linetype, size)

f + geom_boxplot(x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight)

f + geom_dotplot(binaxis = "y", stackdir = "center", x, y, alpha, color, fill, group)

f + geom_violin(scale = "area", x, y, alpha, color, fill, group, linetype, size, weight)

discrete x, discrete y

g <- ggplot(diamonds, aes(cut, color))

g + geom_count(x, y, alpha, color, fill, shape, size, stroke)

THREE VARIABLES

sealsSz <- with(seals, sqrt(delta_long^2 + delta_lat^2)); **l** <- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z))
x, y, z, alpha, colour, group, linetype, size, weight

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500))
x, y, alpha, color, fill, linetype, size, weight

h + geom_density2d
x, y, alpha, colour, group, linetype, size

h + geom_hex
x, y, alpha, colour, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))

i + geom_area
x, y, alpha, color, fill, linetype, size

i + geom_line
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv")
x, y, alpha, color, group, linetype, size

visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

j + geom_crossbar(fatten = 2)
x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar(x, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh**()))

j + geom_linerange
x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange
x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

data <- data.frame(murder = USArrests\$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))

k + geom_map(aes(map_id = state), map = map) + **expand_limits**(x = map\$long, y = map\$lat), map_id, alpha, color, fill, linetype, size

Vignettes

Many packages come with vignettes (aka, R tutorials)

List Vignettes

```
vignette(package = "ggplot2")
```

Vignettes in package 'ggplot2':

ggplot2-specs	Aesthetic specifications (source, html)
extending-ggplot2	Extending ggplot2 (source, html)
profiling	Profiling Performance (source, html)

Vignettes

Many packages come with vignettes (aka, R tutorials)

List Vignettes

```
vignette(package = "ggplot2")
```

Vignettes in package 'ggplot2':

ggplot2-specs	Aesthetic specifications (source, html)
extending-ggplot2	Extending ggplot2 (source, html)
profiling	Profiling Performance (source, html)

Load Vignettes

```
vignette("ggplot2-specs", package = "ggplot2")
```

Try it!


Tutorials


Vignettes also online

- e.g., [tidyverse](#)

Organizations/Websites

- [Software Carpentry](#)
- [STHDA](#)

 **ggplot2** part of the [tidyverse](#)
3.2.1

Reference Articles ▾ News ▾ Extensions 

Overview

ggplot2 is a system for declaratively creating graphics, based on [The Grammar of Graphics](#). You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Installation

```
# The easiest way to get ggplot2 is to install the whole tidyverse:
install.packages("tidyverse")

# Alternatively, install just ggplot2:
install.packages("ggplot2")

# Or the the development version from GitHub:
# install.packages("devtools")
devtools::install_github("tidyverse/ggplot2")
```

Links

Download from CRAN at
<https://cloud.r-project.org/package=ggplot2>

Browse source code at
<https://github.com/tidyverse/ggplot2>

Report a bug at
<https://github.com/tidyverse/ggplot2/issues>

Learn more at
<http://r4ds.had.co.nz/data-visualisation.html>

Extensions at
<http://www.ggplot2-exts.org/gallery/>

License
[GPL-2](#) | file [LICENSE](#)

ggplot2 website

Books!

Free Online

- [R for Data Science](#) (read it!)
- [R Graphics Cookbook](#) (how to do X)
- [ggplot2](#) (next level)
- [Data Visualization: A practical introduction](#)
- [Geocomputation with R](#) (spatial, GIS, maps)
- [Statistical Inference via Data Science: A ModernDive into R and the tidyverse](#) (stats)

Specific help

Examples

In R

```
?geom_boxplot
```

Copy and paste the examples into your console

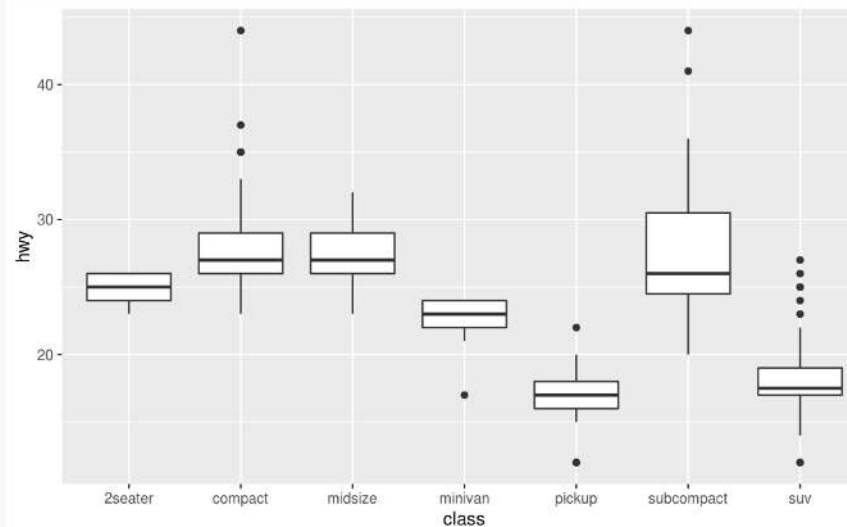
Examples

On the web

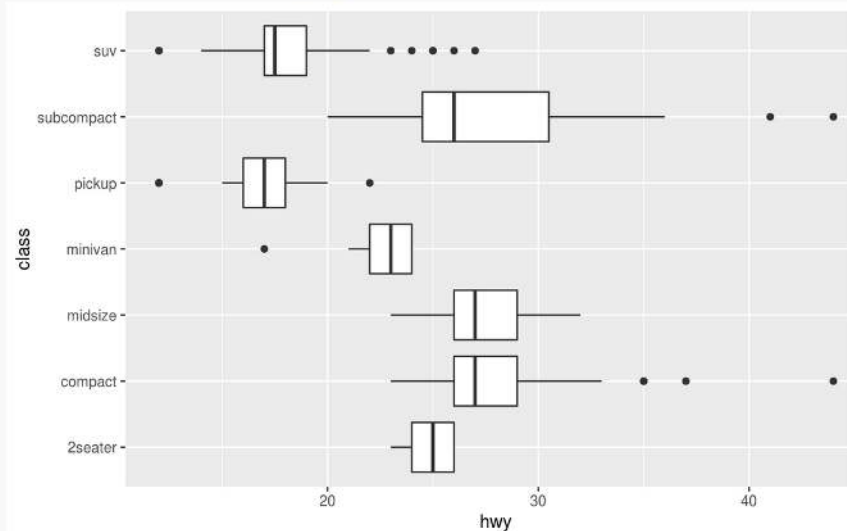
- Nice to see expected output
- Helps figure out if it's your system or your code

Examples

```
p <- ggplot(mpg, aes(class, hwy))  
p + geom_boxplot()
```



```
p + geom_boxplot() + coord_flip()
```



Contents

- [Arguments](#)
- [Summary statistics](#)
- [Aesthetics](#)
- [Computed variables](#)
- [References](#)
- [See also](#)
- [Examples](#)

Web searches

- **Always include "R" in the search**
- **Include the package name!**
- **Use keywords**
- **Some errors are very general**

Web searches

- **Always include "R" in the search**
- **Include the package name!**
 - Try "R boxplots" vs. "R boxplots ggplot2"
- **Use keywords**
 - Try "R boxplots ggplot2 notch"
- **Some errors are very general**
 - Try "R Error: object 'm' not found"

Stackoverflow etc.

"R how to remove duplicate rows"

Stackoverflow etc.

Things to consider

- Date (i.e., R version, Package Version)
- Packages used (**tidyverse**? R base? A mix?)
- What are the example data?
 - **mtcars** and **iris** are commonly used data sets built into R base
 - **msleep** and **diamonds** are commonly used data sets built into **ggplot2**
- What are the example columns?
- What is actually required to answer *your* question?

Asking for Help

Not useful

- "I got an error"
- "It didn't work"

Asking for Help

Not useful

- "I got an error"
- "It didn't work"

Better!

- "I got *this* error"
- "It didn't give me *this*"

Asking for Help

Not useful

- "I got an error"
- "It didn't work"

Better!

- "I got *this* error"
- "It didn't give me *this*"

Best!!

- "I did *this* and I got *this* error"
- "I expected it to do *this*, but in fact the output was *this*"

Asking for Help

Not useful

- "I got an error"
- "It didn't work"

Better!

- "I got *this* error"
- "It didn't give me *this*"

Best!!

- "I did *this* and I got *this* error"
- "I expected it to do *this*, but in fact the output was *this*"

Best of the Best!!!

- "I did *this* [small reproducible code, including data set] and I got *this* [exact error/output]"

Reproducible Examples

- Minimal code and data required to reproduce the error
- Often preparing this actually helps you solve the error!
- Includes
 - packages (**library()**)
 - data
 - runnable code

Reproducible Examples

How do I change the order of **vore**?

Not reproducible

```
ggplot(data = m, aes(x = vore, y = awake, fill = `Body Size`)) +  
  theme_bw() +  
  theme(axis.title.x = element_blank()) +  
  geom_boxplot() +  
  scale_fill_viridis_d() +  
  labs(y = "Awake time (hrs)",  
        title = "Awake time by Diet")
```

```
## Error in ggplot(data = m, aes(x = vore, y = awake, fill = `Body Size`)): could not find function  
"ggplot"
```

- No indication of packages
- No indication of what **m** is

Reproducible Examples

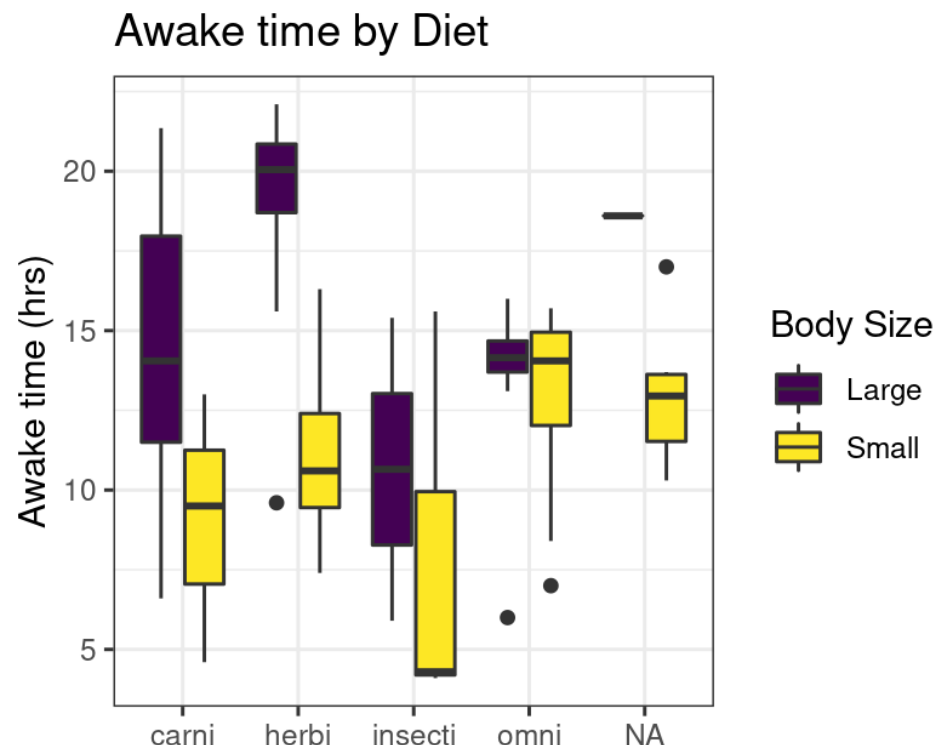
How do I change the order of **vore**?

Reproducible, but not minimal

```
library(ggplot2)
```

```
m <- msleep %>%  
  mutate(`Body Size` = if_else(bodywt > median(bodywt),  
                                "Large", "Small"))
```

```
ggplot(m, aes(x = vore, y = awake, fill = `Body Size`)) +  
  theme_bw() +  
  theme(axis.title.x = element_blank()) +  
  geom_boxplot() +  
  scale_fill_viridis_d() +  
  labs(y = "Awake time (hrs)",  
       title = "Awake time by Diet")
```



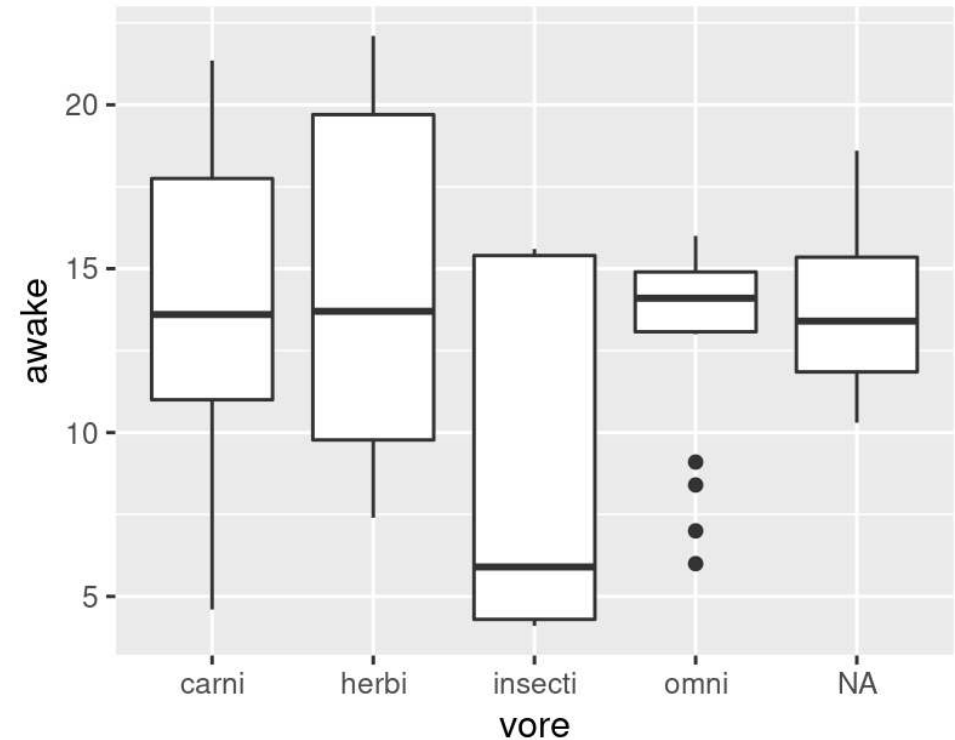
Reproducible Examples

How do I change the order of **vore**?

Reproducible AND Minimal

```
library(ggplot2)

ggplot(msleep, aes(x = vore, y = awake)) +
  geom_boxplot()
```



Talking about Reproducibility...

Citing Software

In-line Text

- Software name
- Version
- Programmers/authors OR Journal article releasing the software (if available)

Bibliography

- Journal article releasing the program **OR**
- Programmers/authors
- Year of release
- Program Name
- URL

Citing R

Inline

"All statistical analyses were performed with R statistical software (v3.6.2, R Core Team 2019)."

Bibliography

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Citing R

Version information

```
R.Version()$version.string
```

```
## [1] "R version 4.0.3 (2020-10-10)"
```

Citation information

```
citation()
```

```
##  
## To cite R in publications use:  
##  
## R Core Team (2020). R: A language and environment for statistical  
## computing. R Foundation for Statistical Computing, Vienna, Austria.  
## URL https://www.R-project.org/.
```


Citing R Packages

Inline

"All statistical analyses were performed with R statistical software (v4.0.3, R Core Team 2020). We performed Type III ANOVAs using the 'car' package for R (v3.0.10, Fox and Weisberg)."

Bibliography

John Fox and Sanford Weisberg (2019). An R Companion to Applied Regression, Third Edition. Thousand Oaks CA: Sage.

Citing R Packages

Version information

```
packageVersion("car")
```

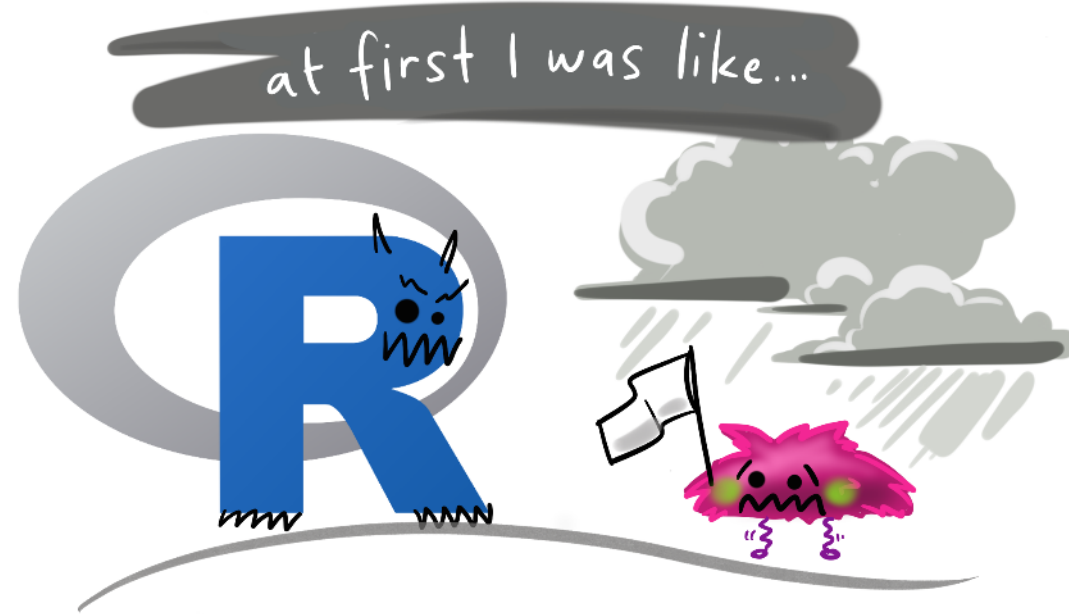
```
## [1] '3.0.10'
```

Citation information

```
citation("car")
```

```
##  
## To cite the car package in publications use:  
##  
## John Fox and Sanford Weisberg (2019). An {R} Companion to Applied  
## Regression, Third Edition. Thousand Oaks CA: Sage. URL:  
## https://socialsciences.mcmaster.ca/jfox/Books/Companion/
```

You made it!
Thank you!



...but now it's like...

