

Getting Help with R

After this workshop

 [steffilazerte](#)
 @steffilazerte@fosstodon.org
 [@steffilazerte](#)
 [steffilazerte.ca](#)

Dr. Steffi LaZerte





Analysis and Data Tools for Science



Compiled: 2024-02-13

First things first

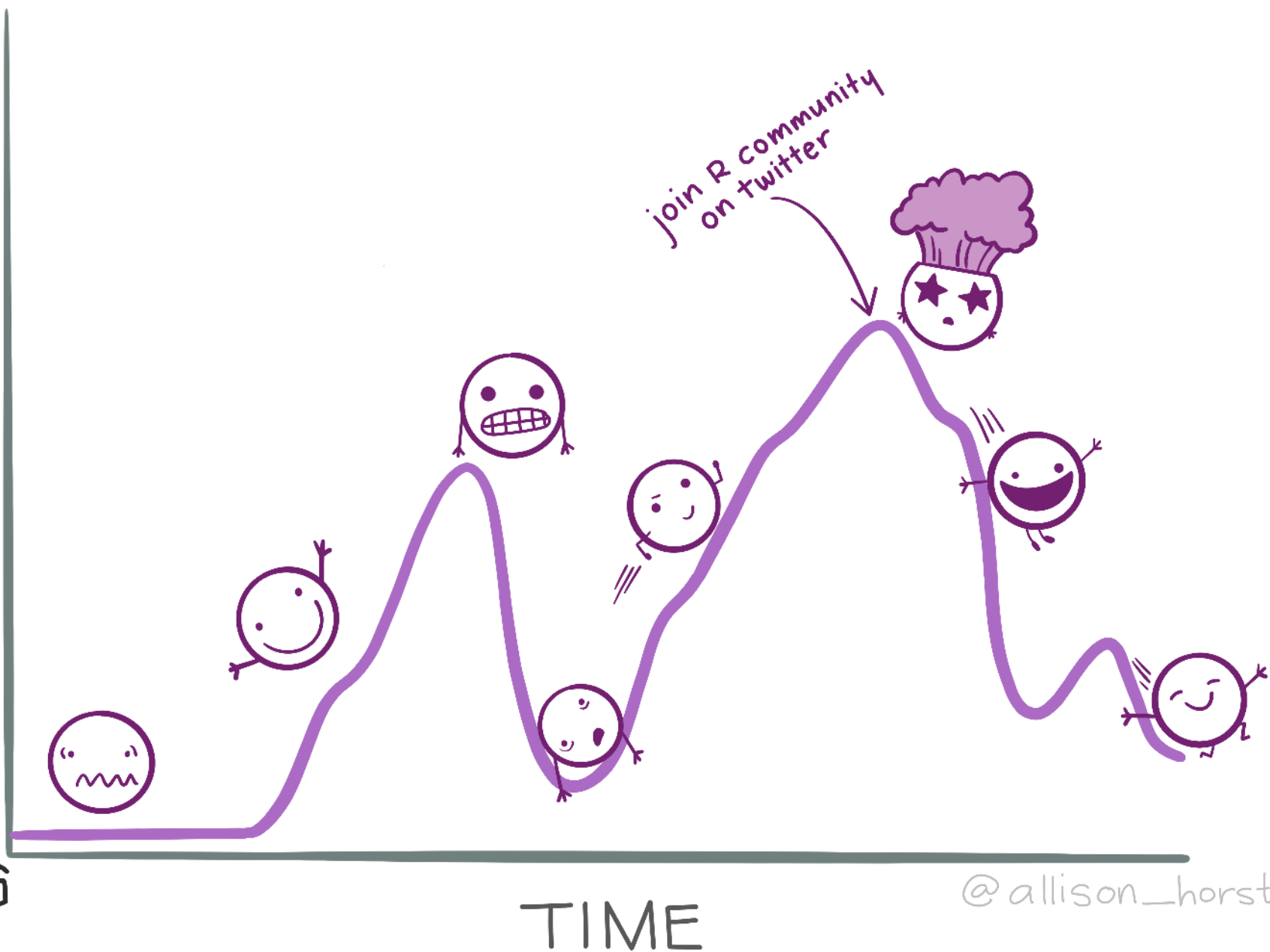
-  Save previous script
-  Consider taking notes during this section

I KNOW_
LOTS!

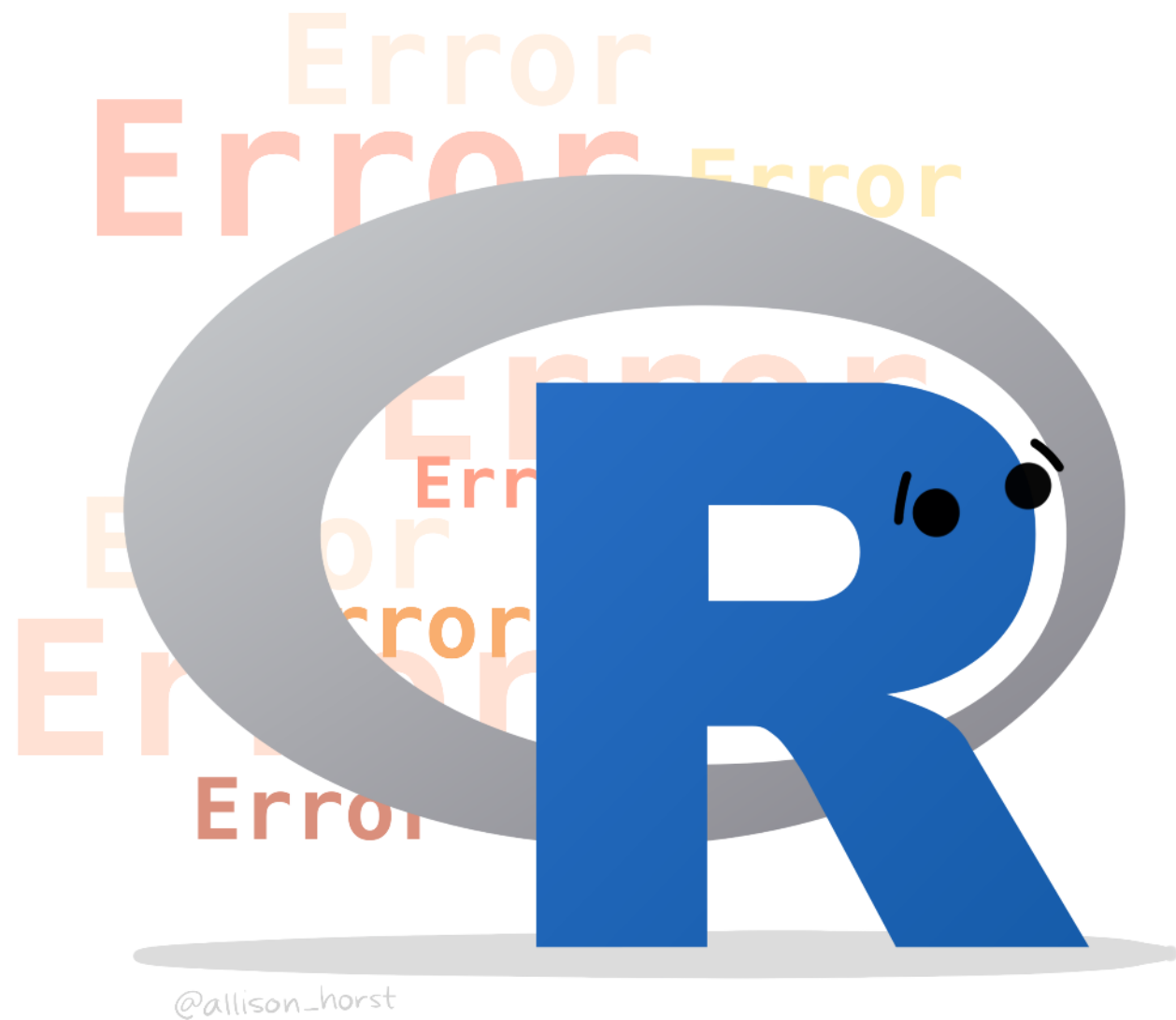
HOW
MUCH
I THINK
I KNOW
ABOUT **R**

I KNOW_
NOTHING

join R community
on twitter



Troubleshooting



Bit by bit

Line by line

- R is sequential
- If you skip lines, you're not running that part (and R has no idea)

```
1 #a <- 1  
2 b <- 2  
3 a + b
```

```
Error in eval(expr, envir, enclos): object 'a' not found
```

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

```
1 a <- 1  
2 b <- 2  
3 a + b
```

```
[1] 3
```

Bit by bit

Line by line

```
1 library(tidyverse)
2
3 # Load Data
4 size <- read_csv("../data/grain_size2.csv")
5
6 # First modification
7 size <- mutate(size,
8               total_sand = coarse_sand + medium_sand + fine_sand,
9               total_silt = coarse_silt + medium_silt + fine_silt)
10
11 # Second modification
12 size <- size |>
13   group_by(plot) |>
14   summarize(n = n(),
15             total_sand = sum(total_sand),
16             mean_sand = mean(total_sand),
17             sd_sand = sd(total_sand),
18             se_sand = sd_sand / sqrt(n))
```

Especially important if
loading and modifying
data

Can't run 1st modification
after 2nd modification

Bit by bit

Section by section

```
1 library(tidyverse)
2
3 size <- read_csv("../data/grain_size2.csv") |>
4   mutate(total_sand = coarse_sand + medium_sand + fine_sand,
5          total_silt = coarse_silt + medium_silt + fine_silt) |>
6   group_by(plot) |>
7   summarize(n = n(),
8             total_sand = sum(total_sand),
9             mean_sand = mean(total_sand),
10            sd_sand = sd(total_sand),
11            se_sand = sd_sand / sqrt(n))
```

Error in `summarize()`:

i In argument: `mean_sand = mean(total_sand)`.

i In group 1: `plot = "CSP01"`.

Caused by error:

! object 'total_sand' not found

Bit by bit

Section by section

```
1 library(tidyverse)
2
3 size <- read_csv("./data/grain_size2.csv")
```

No error

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

No error

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

No error

Bit by bit

Section by section

```
1 size <- read_csv("../data/grain_size2.csv") |>
2   mutate(total_sand = coarse_sand + medium_sand + fine_sand,
3           total_silt = coarse_silt + medium_silt + fine_silt) |>
4   group_by(plot) |>
5   summarize(n = n(),
6             total_sand = sum(total_sand),
7             mean_sand = mean(total_sand),
8             sd_sand = sd(total_sand),
9             se_sand = sd_sand / sqrt(n))
```

Error in `summarize()`:

i In argument: `mean_sand = mean(total_sand)`.

i In group 1: `plot = "CSP01"`.

Caused by error:

! object 'total_sand' not found

Ah ha!

Bit by bit

Applies to error messages too

- First, don't panic!
- Look at the error bit by bit

```
Error: Problem with `summarise()` column `mean_sand`.  
i `mean_sand = mean(totall_sand)`.  
x object 'totall_sand' not found  
i The error occurred in group 1: plot = "CSP01".
```

Bit by bit

Applies to error messages too

```
Error: Problem with 'summarise()' column 'mean_sand`
```

Okay, we know the problem is in the `summarize()` part and then `mean_sand` part of that

```
i 'mean_sand = mean(totall_sand)'  
x object 'totall_sand' not found
```

Looks like this is the line with the problem.

And the problem is `object 'totall_sand' not found`.

Ooops! Typo!

```
i The error occurred in group 1: plot = "CSP01".
```

Lastly, it's telling us that the problem was when working with this group of data.

(This can be useful when troubleshooting, because you can `filter()` your data and take a look)

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

Just sometimes unhelpful!

Getting Help

Cheat Sheets

RStudio Menu

- Help
 - Cheatsheets

Take a look yourself

Data Visualization with ggplot2 : : CHEAT SHEET

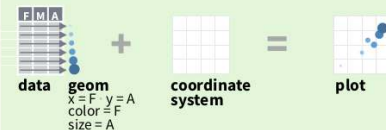


Basics

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> +  
  <FACET_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.

aesthetic mappings data geom

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, unemployment))  
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, y, 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 = unemployment - 900, ymax = unemployment + 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 = 1:1155, 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(fl))
```

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, unemployment))
```

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 (tutorials)

List Vignettes

```
1 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

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

Try it!

Tutorials

Vignettes are also online

- e.g., [ggplot2](#)
- e.g., [tidyverse](#)

Organizations/Websites

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



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](#)

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)

Communities!

- [rOpenSci](#)
- [#RStats Twitter](#)
- [#RStats Mastodon](#)
(e.g., [Fosstodon.org](#) or [Hachyderm.io](#))
- [R4DS Online learning community on Slack](#)
(ask any question, they're really nice!)



Specific Groups

- [rLadies](#)
- [MiR](#)
- [AfricaR](#)
- [AsiaR](#)



Specific help

Examples

In R

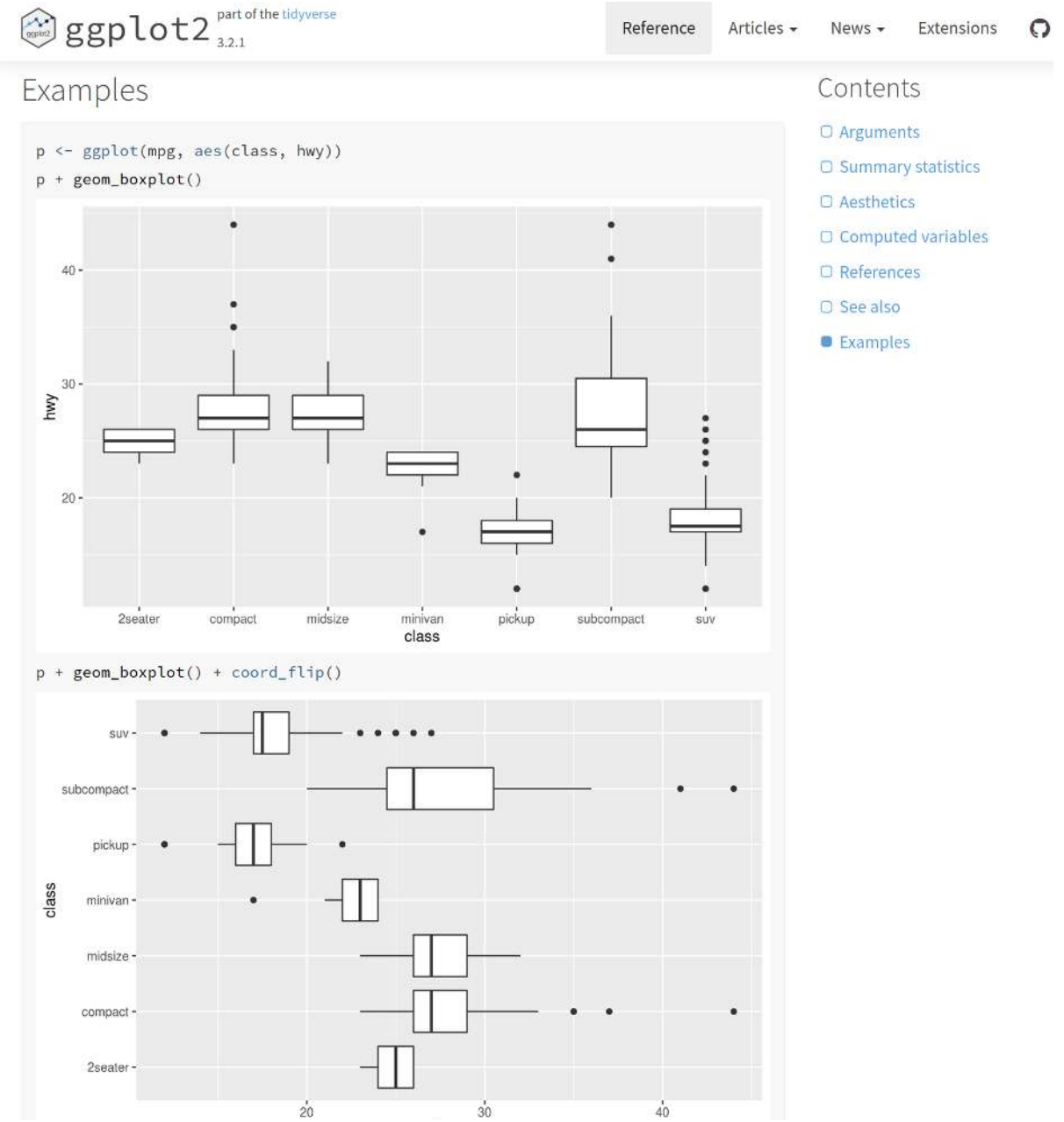
```
1 ?geom_boxplot
```

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



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”

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

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

Error in eval(expr, envir, enclos): object 'm' not found

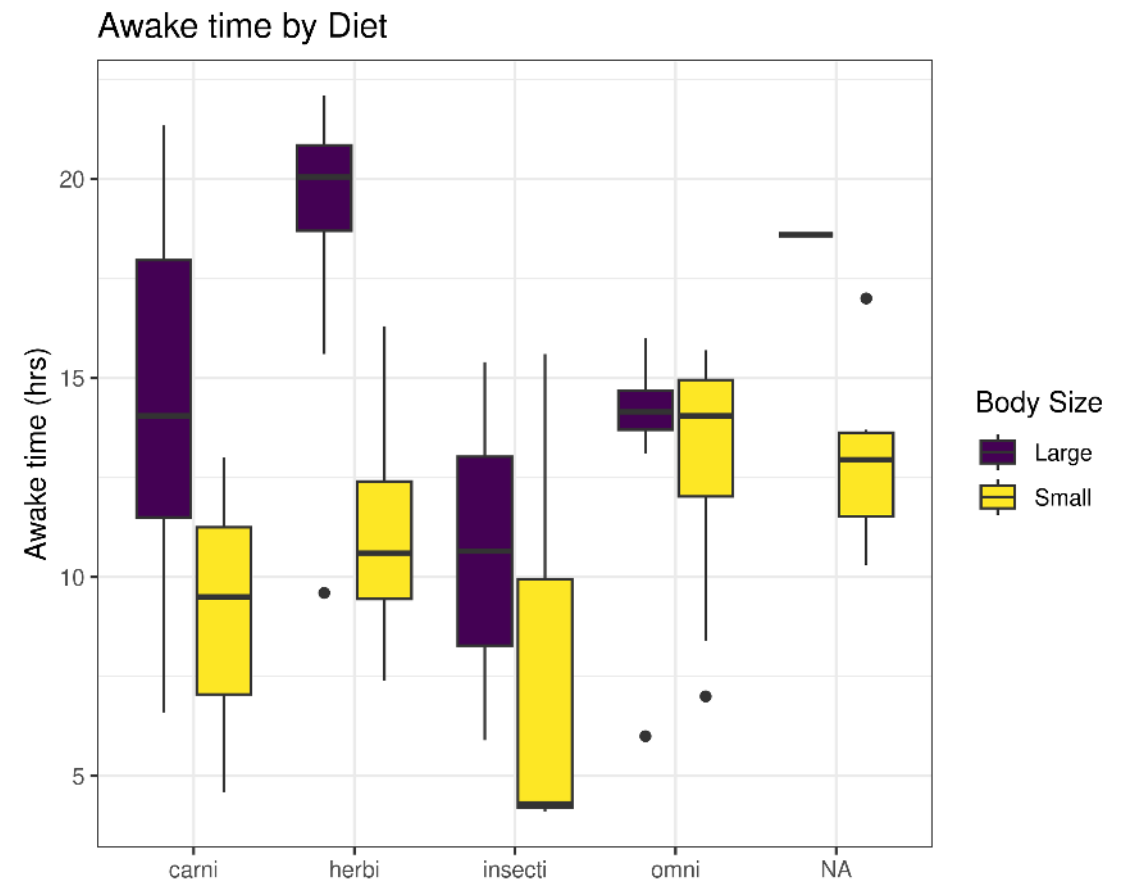
- No indication of packages
- No indication of what **m** is

Reproducible Examples

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

Reproducible, but not minimal

```
1 library(ggplot2)
2
3 m <- msleep |>
4   mutate(`Body Size` = if_else(bodywt > median(bodywt),
5                                "Large", "Small"))
6
7 ggplot(m, aes(x = vore, y = awake, fill = `Body Size`)) +
8   theme_bw() +
9   theme(axis.title.x = element_blank()) +
10  geom_boxplot() +
11  scale_fill_viridis_d() +
12  labs(y = "Awake time (hrs)",
13       title = "Awake time by Diet")
```

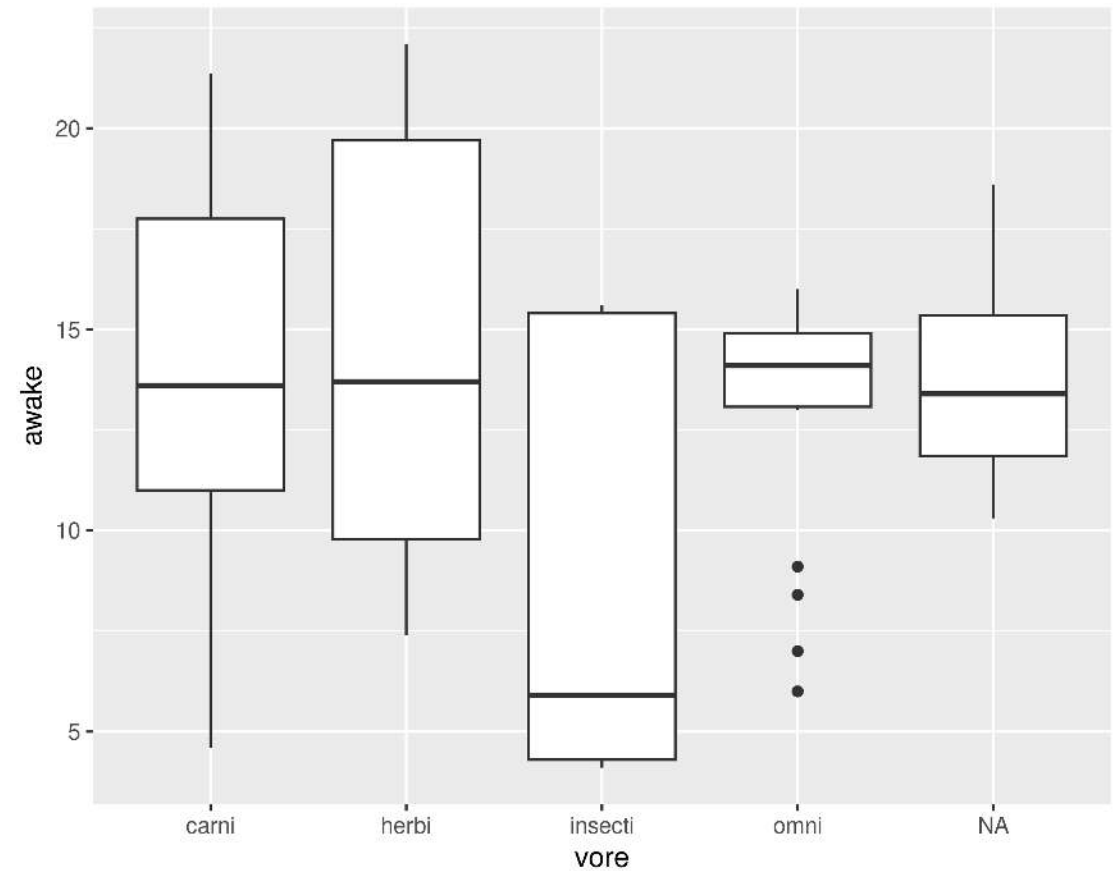


Reproducible Examples

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

Reproducible AND Minimal

```
1 library(ggplot2)
2
3 ggplot(msleep, aes(x = vore, y = awake)) +
4   geom_boxplot()
```



Paying it forward

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 (v4.3.2, R Core Team 2023).”

Bibliography

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

Citing R

Version information

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

```
[1] "R version 4.3.2 (2023-10-31)"
```

Citation information

```
1 citation()
```

To cite R in publications use:

```
R Core Team (2023). _R: A Language and Environment for Statistical  
Computing_. R Foundation for Statistical Computing, Vienna, Austria.  
<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 2019).”

Bibliography

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

Citing R Packages

Version information

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

```
[1] '3.1.2'
```

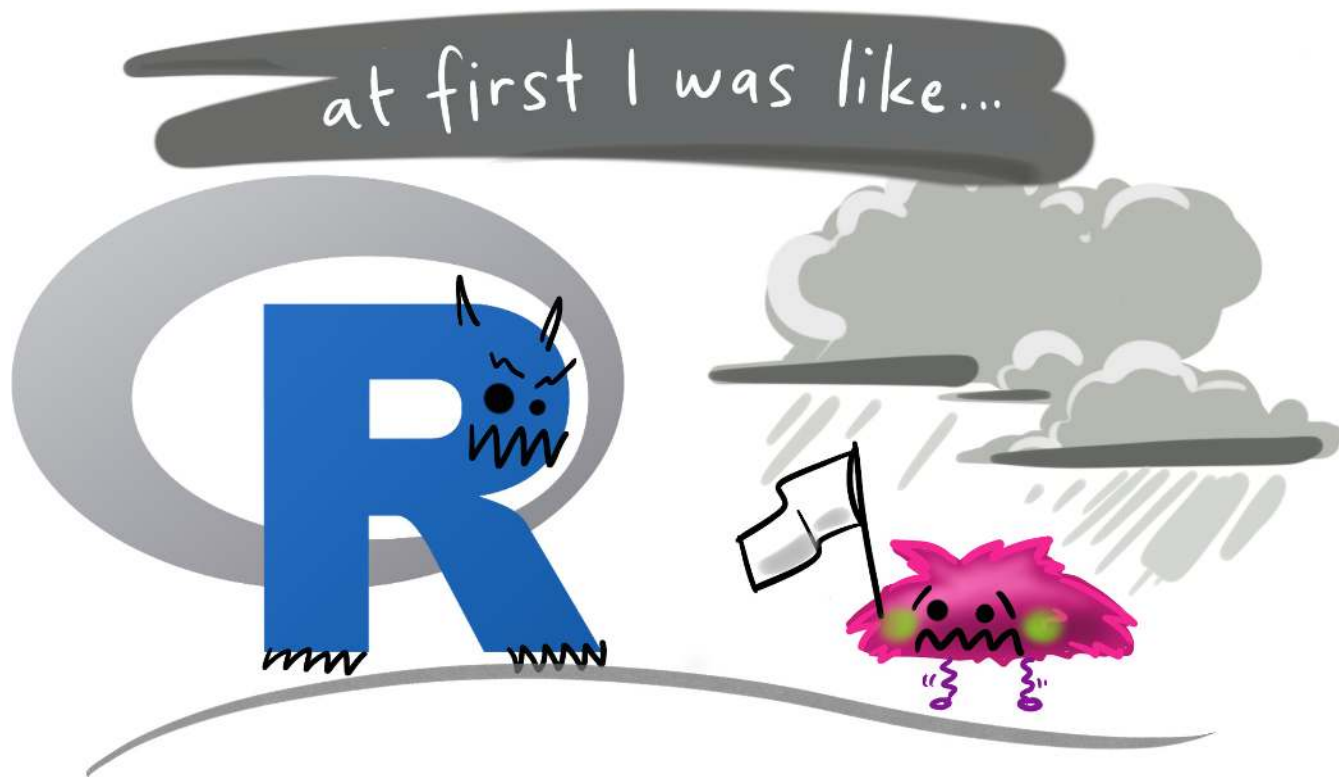
Citation information

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

To cite the car package in publications use:

```
Fox J, Weisberg S (2019). _An R Companion to Applied Regression_,  
Third edition. Sage, Thousand Oaks CA.  
<https://socialsciences.mcmaster.ca/jfox/Books/Companion/>.
```

See more about citing packages in my rOpenSci blog post: [How to Cite R and R packages](#)



You made it!

Thank you!

(Feedback!)