

# Getting Help with R

After this workshop

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

Dr. Steffi LaZerte



Analysis and Data Tools for Science



Compiled: 2023-04-19

# First things first

 Save previous script

 Open New File

(make sure you're in the RStudio Project)

 Write `library(tidyverse)` at the top

 Save this new script

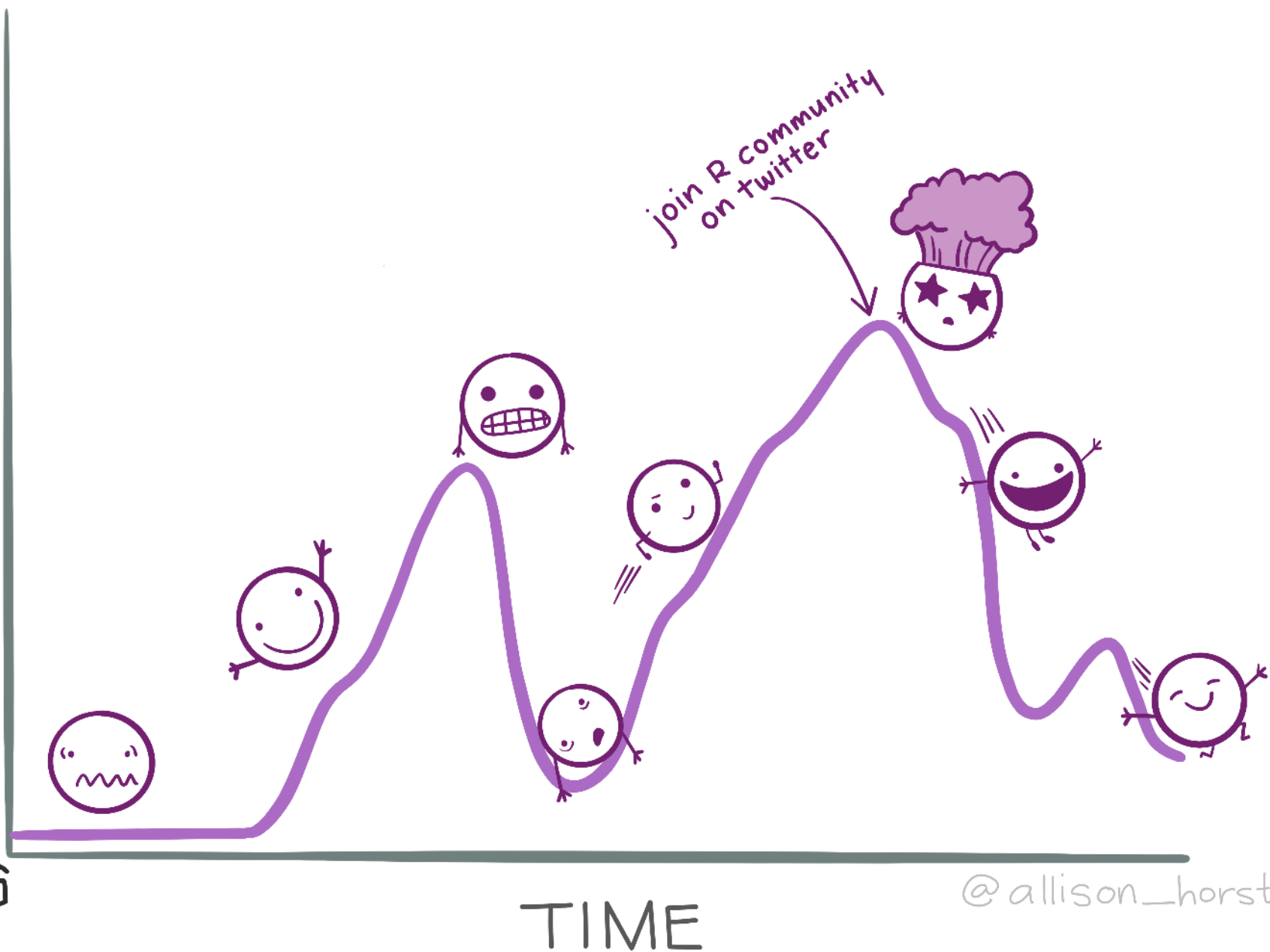
(consider names like `troubleshooting.R` or `5_getting_help.R`)

I KNOW\_  
LOTS!

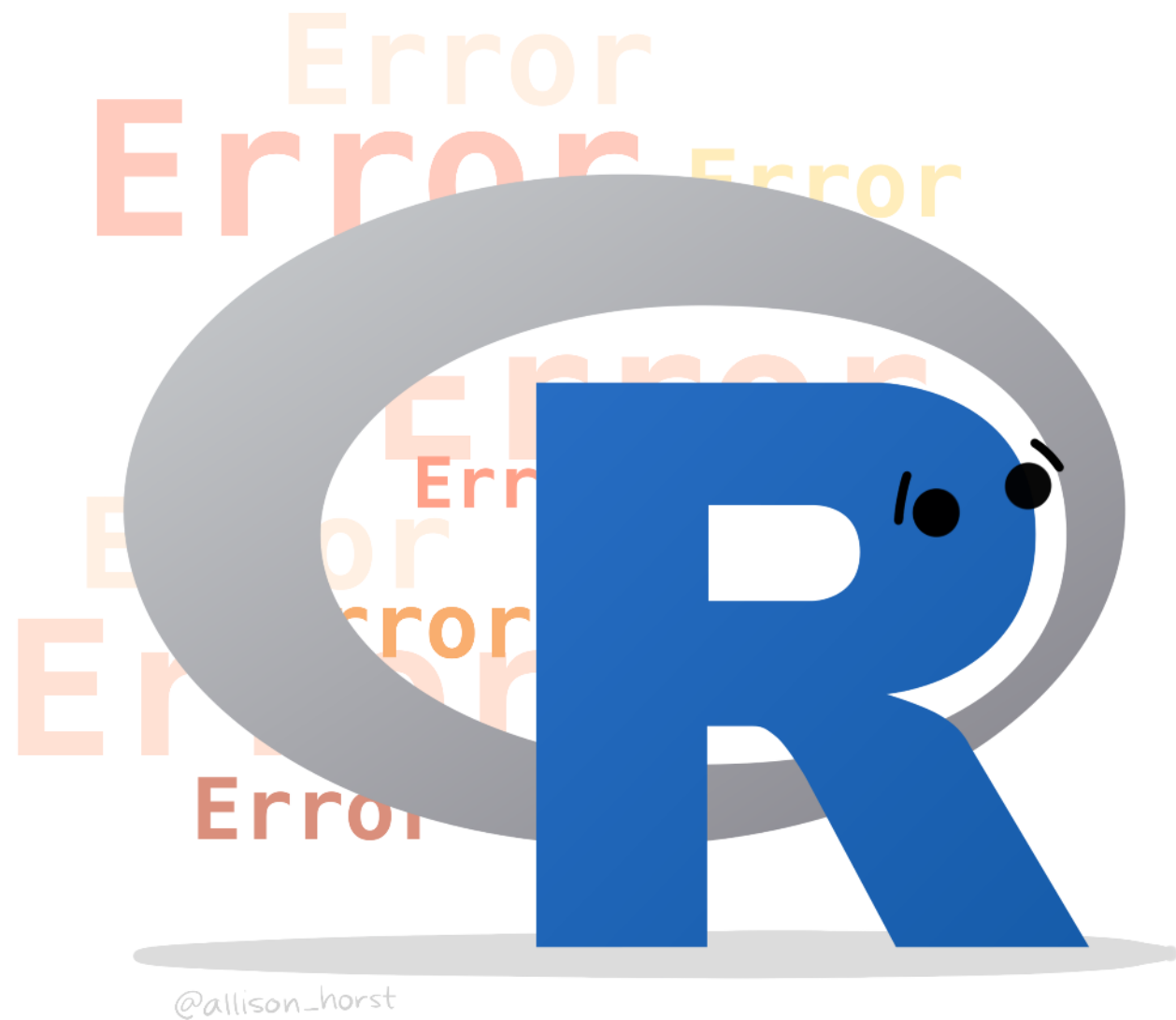
HOW  
MUCH  
I THINK  
I KNOW  
ABOUT **R**

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

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

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

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

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

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

# Bit by bit

## Line by line

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

! object 'total\_sand' not found



# Bit by bit

## Section by section

```
1 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 in `mean()`:

! 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

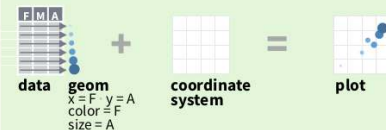


## Basics

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 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](#)
- [rLadies](#)
- [#RStats Twitter](#)
- [#RStats Mastodon](#)  
(e.g., [Fosstodon.org](#) or [Hachyderm.io](#))
- [R4DS Online learning community on Slack](#)



# Specific help

# Examples

## In R

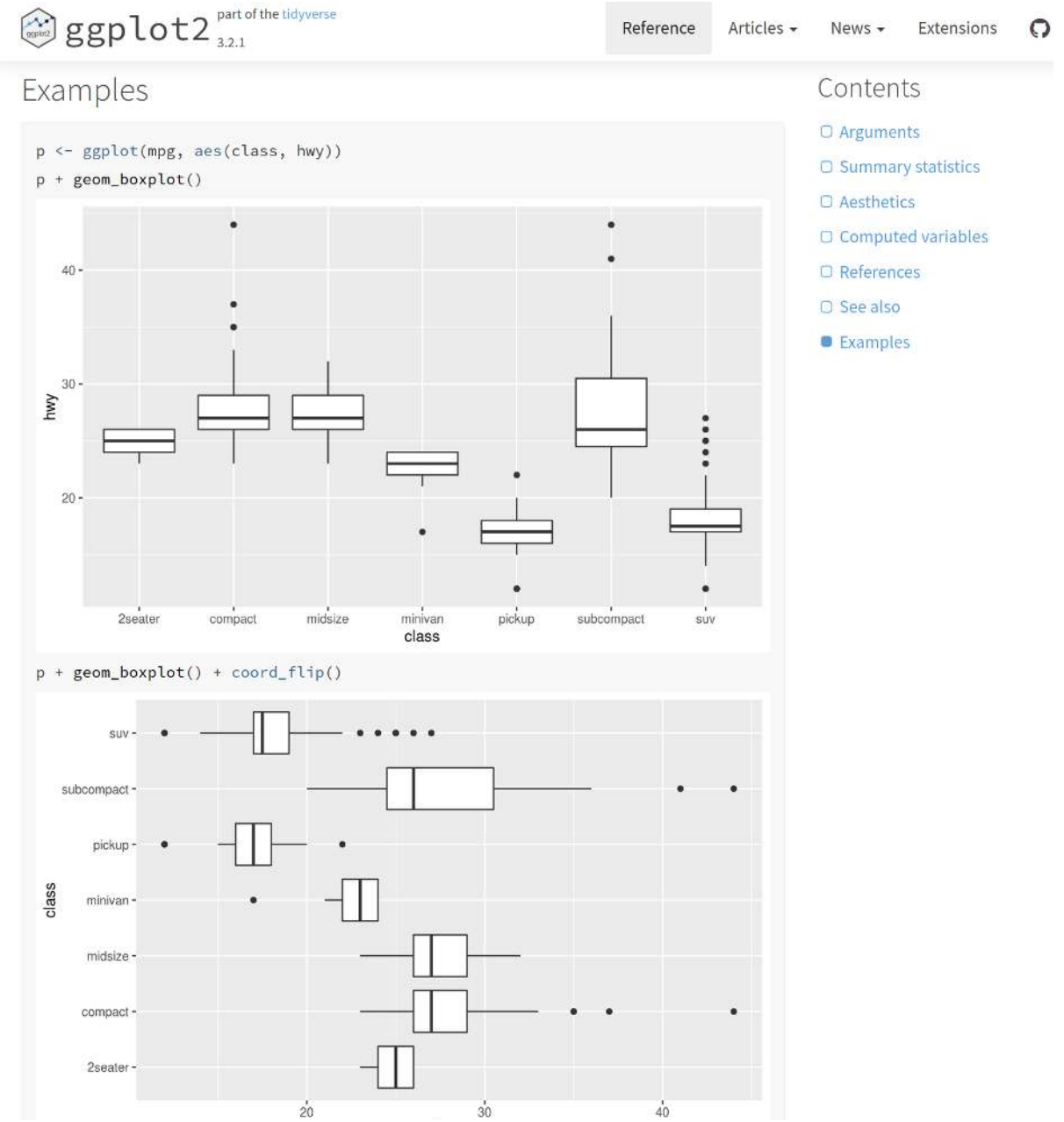
```
1 ?geom_boxplot
```

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 ggplot(data = m, aes(x = vore, y = awake, fill = `Body Size`)): object 'm' not found
```

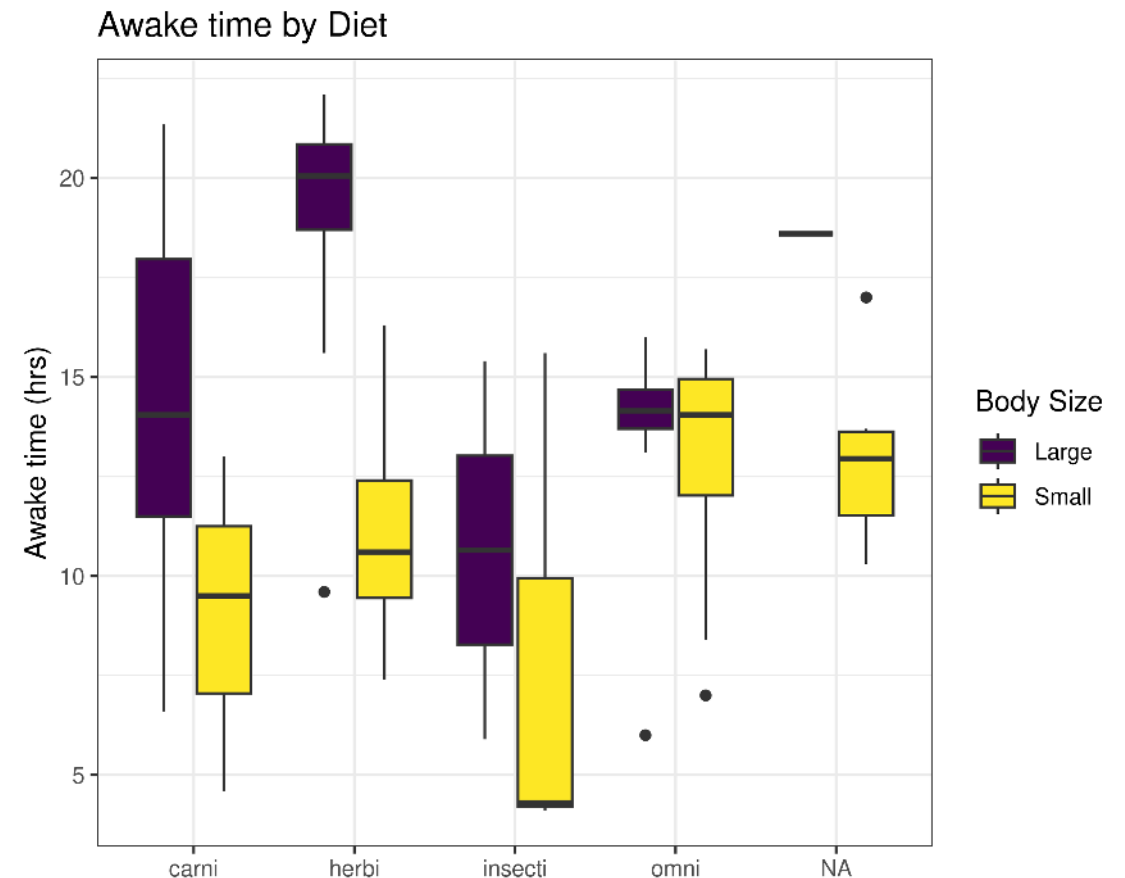
- No indication of packages
- No indication of what **m** is

# 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")
```



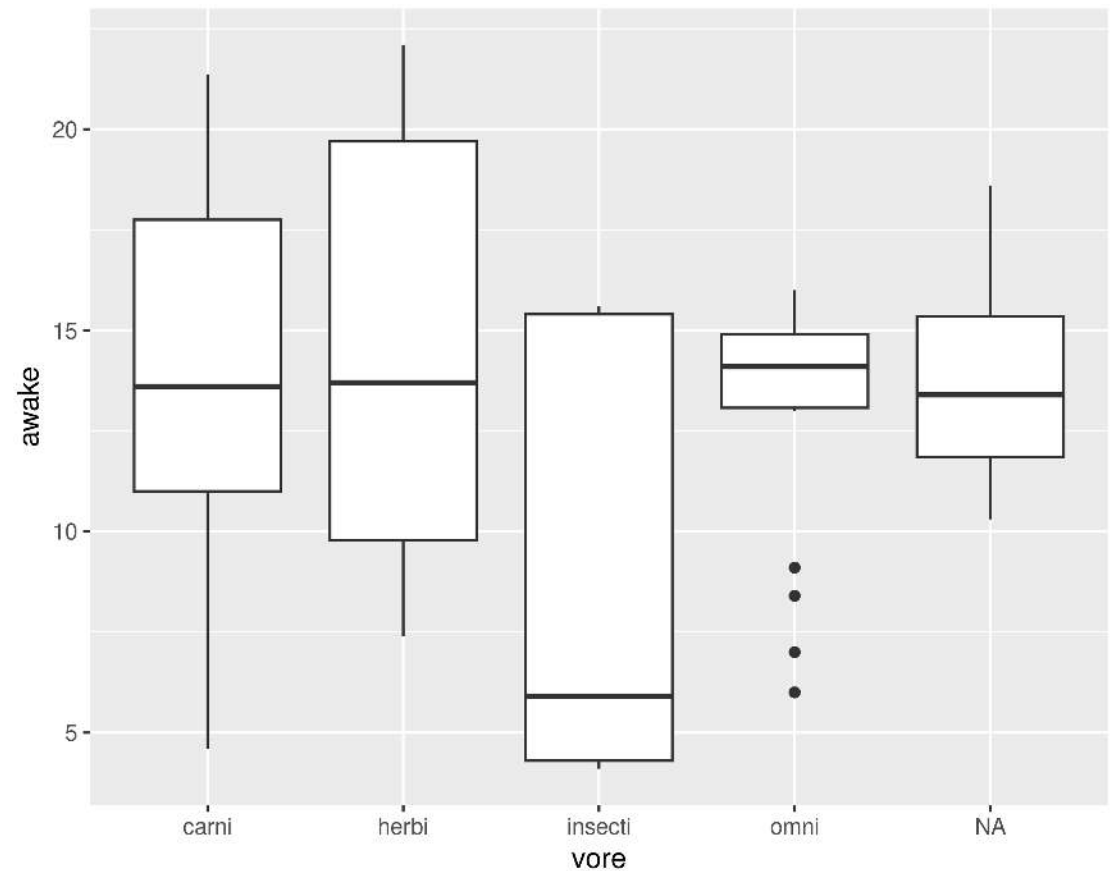


# Reproducible Examples

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 (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

```
1 R.Version()$version.string  
[1] "R version 4.2.3 (2023-03-15)"
```

## 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.  
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

```
1 packageVersion("car")  
[1] '3.1.1'
```

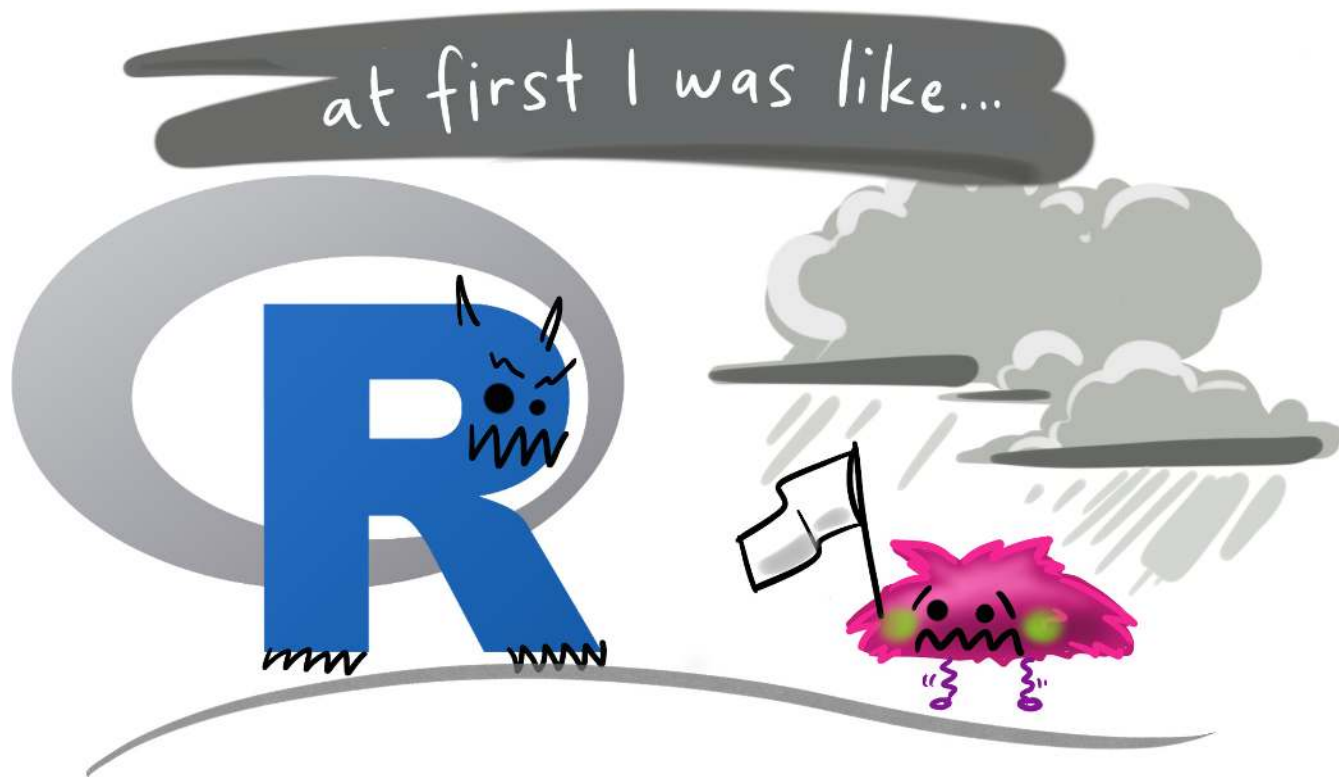
## Citation information

```
1 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/
```

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



**You made it!**

**Thank you!**

(Feedback!)



