

Introduction to `ggplot2`

Data Science for Biologists

Rhino jawn from Naman Srivastava

Using R libraries/packages

1. Install the library/package one time.

```
install.packages("nameoflib")
```

1. Load the library/package for *each R session or script* where you want to use it.

```
library(nameoflib)
```

All packages you *need* have been installed for you in RStudio Cloud Projects inside the Class Workspace. You have to install them yourself in your Personal Workspace.

We are using the tidyverse packages

<https://www.tidyverse.org/>

Loading the core tidyverse once installed:

```
#library(tidyverse)    -- without quotes also works here
library("tidyverse")

## — Attaching packages ——————
## ✓ tibble  3.0.4      ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2      ✓ stringr 1.4.0
## ✓ readr   1.4.0      ✓forcats 0.5.0
## ✓ purrr   0.3.4

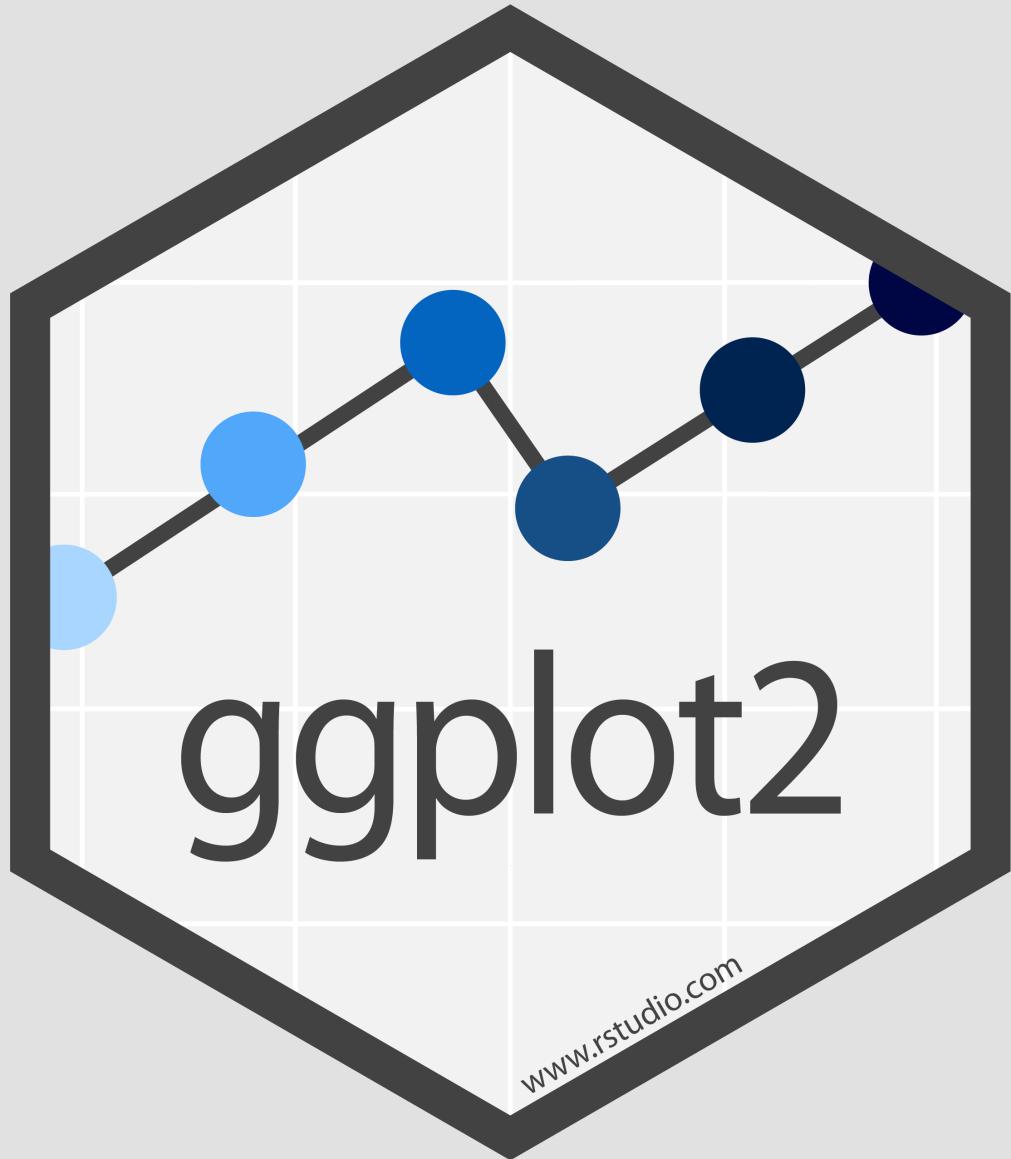
## — Conflicts ——————
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
```

If you don't load your package, you can't use its functions

Ok, only kind of. We'll talk more about this later!

Example of an error:

```
Error in ggplot() : could not find function "ggplot"
```



The dataset

```
msleep_smol
```

```
## # A tibble: 20 x 7
##   name  vore  order
##   <chr> <fct> <fct>
## 1 Greas... omni  Sori...
## 2 Dog    carni Carn...
## 3 Roe   ... herbi Arti...
## 4 Guin... herbi Rode...
## 5 Chin... herbi Rode...
## 6 Star... omni  Sori...
## 7 Less... omni  Sori...
## 8 Long... carni Cing...
## 9 Tree... herbi Hyra...
## 10 Nort... omni  Dide...
## 11 Euro... omni  Erin...
## 12 Pata... omni  Prim...
## 13 Dome... carni Carn...
## 14 Gray... herbi Hyra...
## 15 Gold... herbi Rode...
## 16 Hous... herbi Rode...
## 17 Rabb... herbi Lago...
## 18 Labo... herbi Rode...
## 19 Arct... herbi Rode...
## 20 Thir... herbi Rode...
## # ... with 4 more variables:
## #   conservation <fct>,
## #   awake <dbl>,
```

The dataset

```
summary(msleep_smol)
```

```
##      name          vore
##  Length:20      carni: 3
##  Class :character  herbi:11
##  Mode  :character   omni : 6
##
##           order
##  Rodentia    :7
##  Soricomorpha:3
##  Carnivora   :2
##  Hyracoidea  :2
##  Artiodactyla:1
##  Cingulata   :1
##  (Other)     :4
##           conservation
##  domesticated: 5
##  en         : 1
##  lc         :13
##  nt         : 1
##  vu         : 0
##
##           awake
##  Min.    : 6.00
```

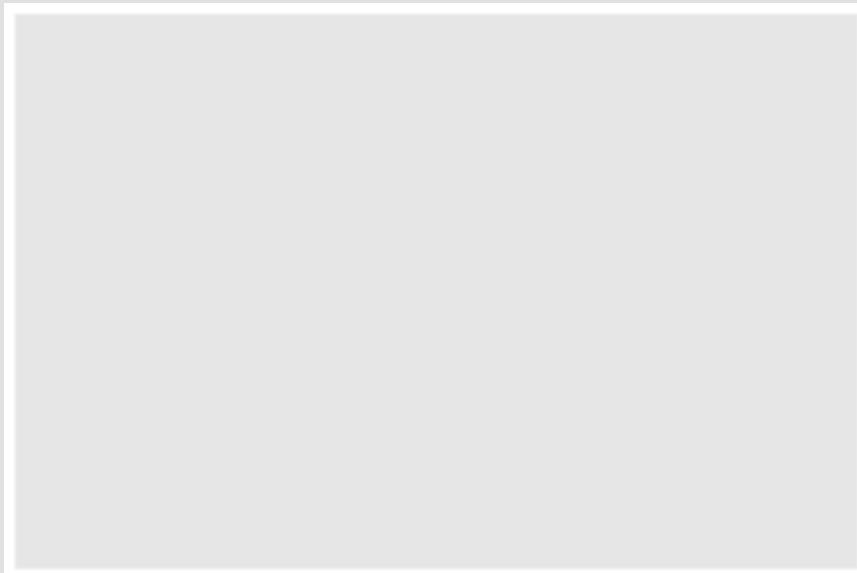
Let's dive in: Scatterplots in ggplot2

Goal: Visualize the relationship between body weight (`bodywt`) and brain weight `brainwt`) of all smol mammals, where `bodywt` is across `brainwt`

Step 1: *Roughly draw/plan the plot by hand. I am completely serious*

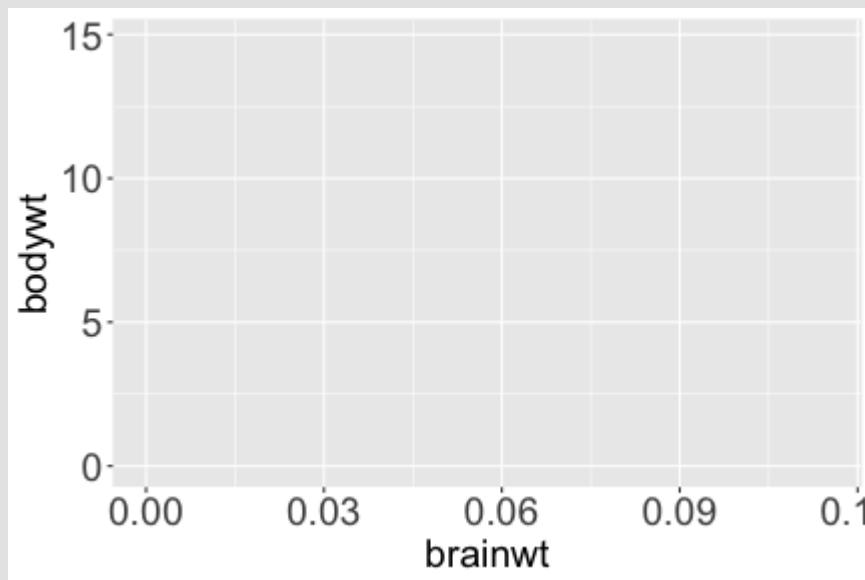
Scatterplots in ggplot2

```
ggplot(msleep_smol)
```



Scatterplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = brainwt, y = bodywt)
```



`aes()` MAPS columns (variables!) onto the plot.

Scatterplots in ggplot2

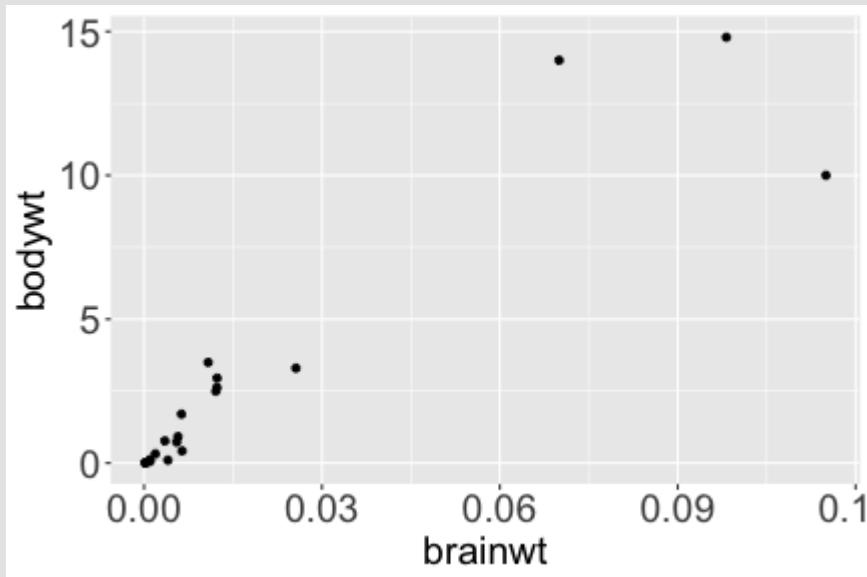
```
ggplot(msleep_smol) +  
  (x = brainwt, y = bodywt)
```

```
## Error: <text>:2:15: unexpected ',',  
## 1: ggplot(msleep_smol) +  
## 2:   (x = brainwt,  
##
```

Don't forget `aes()`!

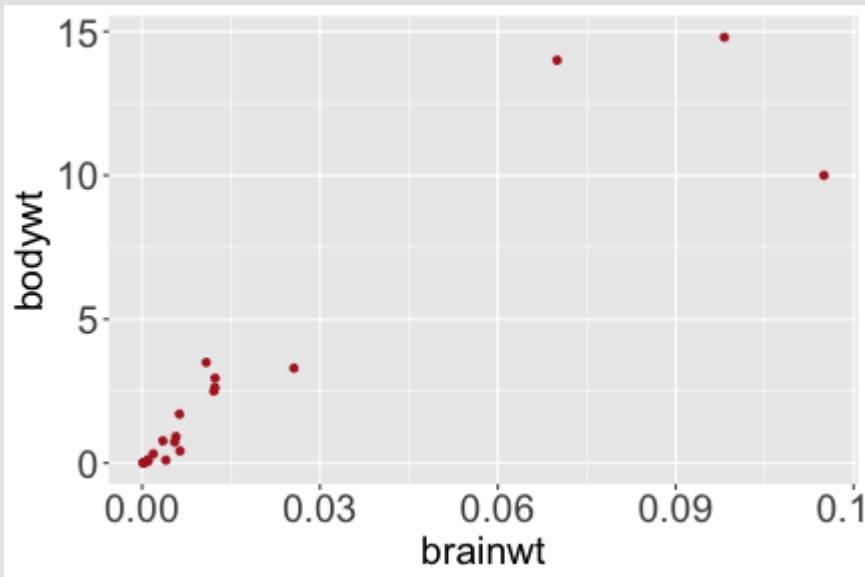
Scatterplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = brainwt, y = bodywt) +  
  geom_point()
```



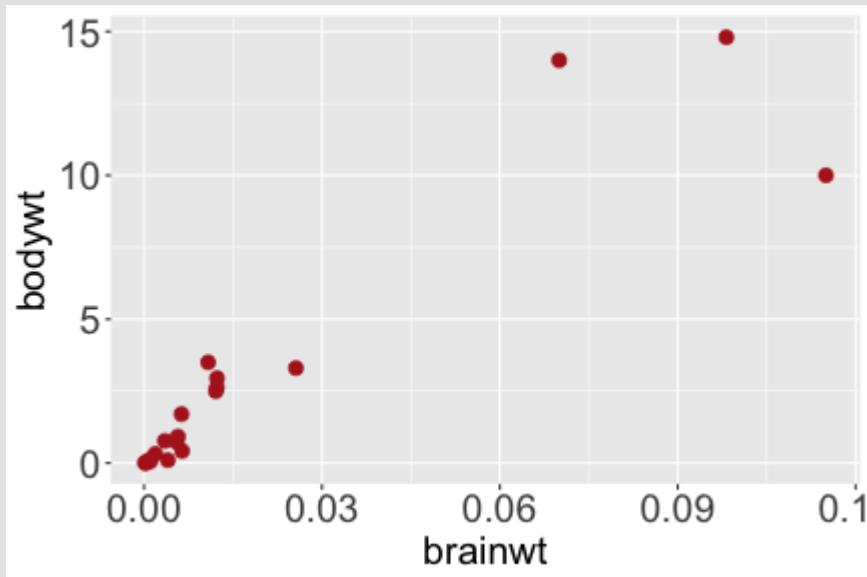
Scatterplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = brainwt, y = bodywt) +  
  geom_point(color = "firebrick")
```



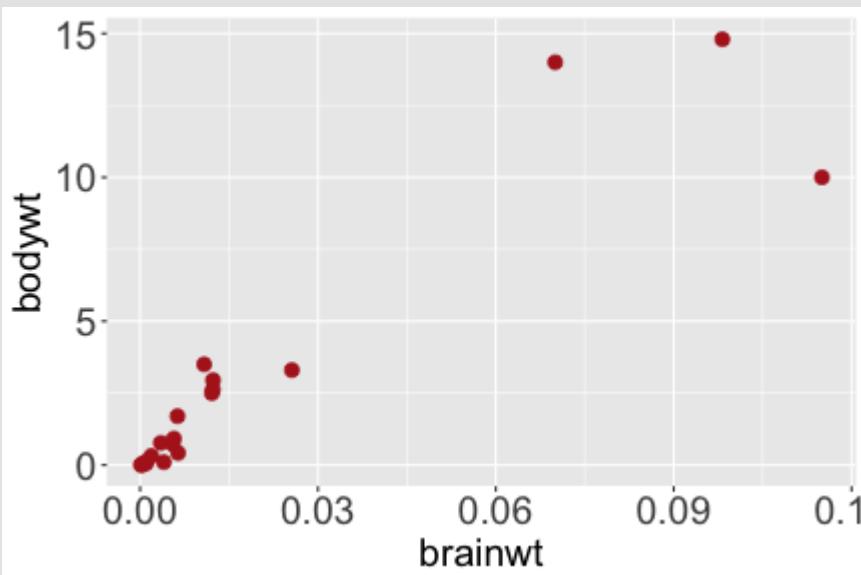
Scatterplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = brainwt, y = bodywt) +  
  geom_point(color = "firebrick", size = 3)
```



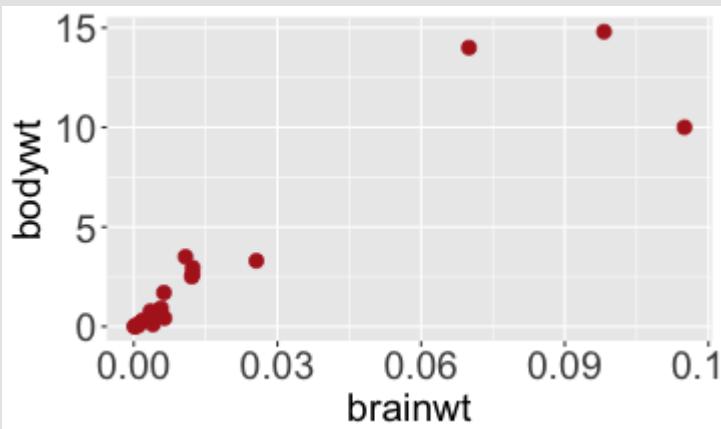
Plots can be saved just like variables

```
sleep_scatter <- ggplot(msleep_smol) +  
  aes(x = brainwt, y = bodywt) +  
  geom_point(color = "firebrick", size = 3)  
  
print(sleep_scatter)
```



Don't forget the *forward assignment* operator

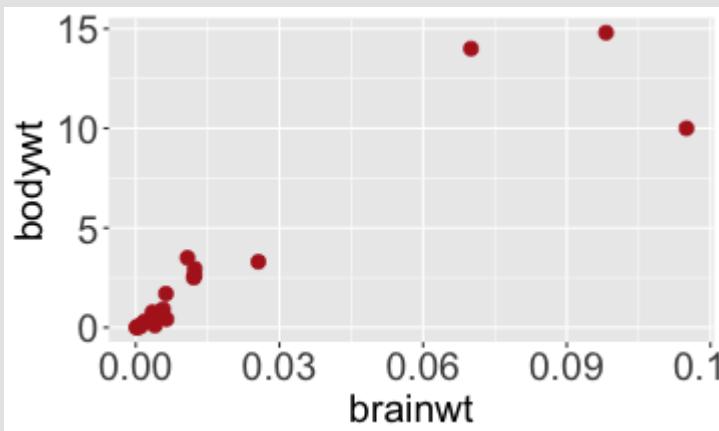
```
ggplot(msleep_smol) +  
  aes(x = brainwt, y = bodywt) +  
  geom_point(color = "firebrick", size = 3) -> sleep_scatter  
  
print(sleep_scatter)
```



Adding onto an existing plot

```
ggplot(msleep_smol) +  
  aes(x = brainwt, y = bodywt) +  
  geom_point(color = "firebrick", size = 3)
```

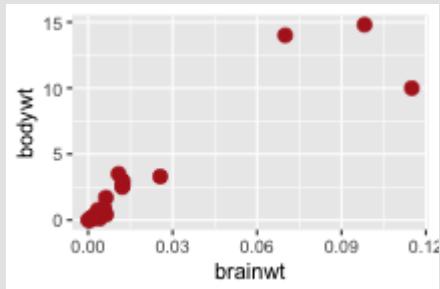
```
ggplot(msleep_smol) +  
  aes(x = brainwt, y = bodywt) -> sleep_xy  
  
sleep_xy +  
  geom_point(color = "firebrick", size = 3)
```



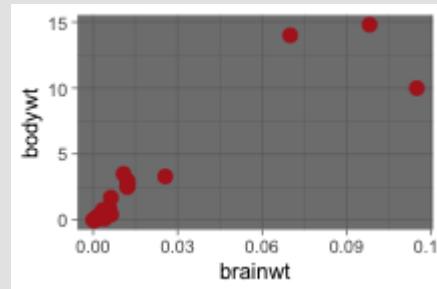
Quick tangent: The plot theme

The default theme is `theme_gray()` (`theme_grey()`)

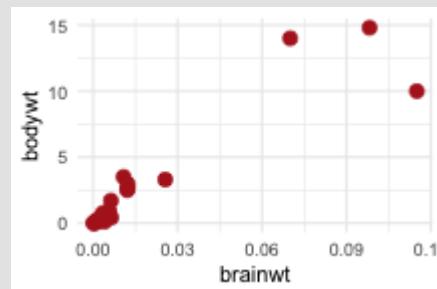
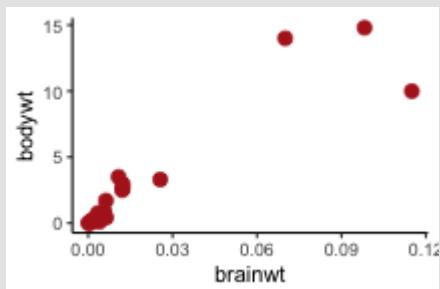
`sleep_scatter`



`sleep_scatter + theme_dark()`

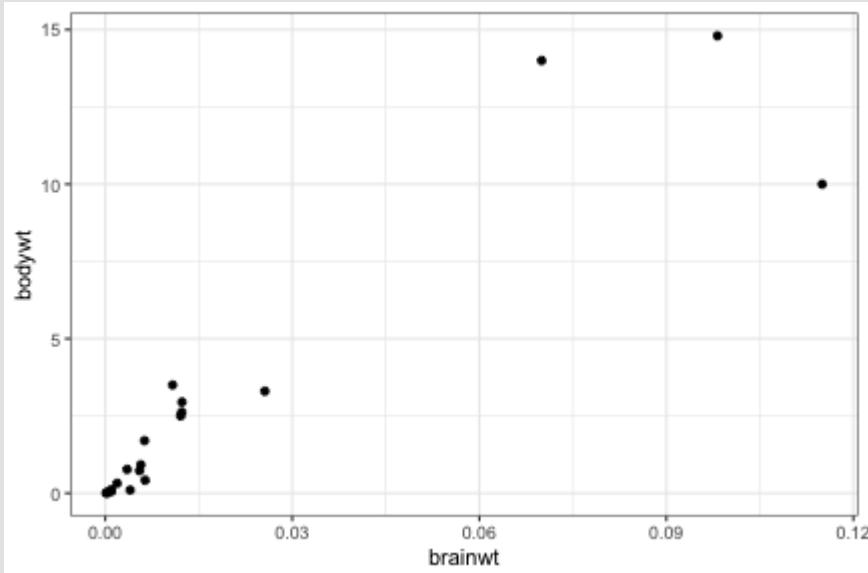


`sleep_scatter + theme_classic()` `sleep_scatter + theme_minimal()`



Adding on a theme

```
ggplot(msleep_smol) +  
  aes(x = brainwt, y = bodywt) +  
  geom_point() +  
  theme_bw()
```



Setting a theme for *all plots*

All plots created *after* this line is run will use `theme_classic()`. All remaining plots in the slides will now use this theme as the new default.

```
theme_set(theme_classic())
```

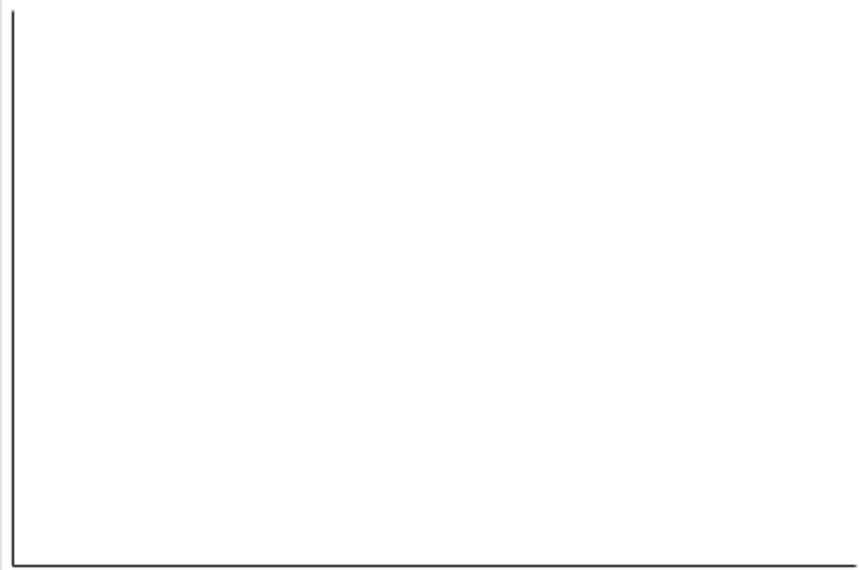
Histograms in ggplot2

Goal: Visualize the distribution of all mammal times spent awake (awake)

Step 1: Plan the plot *by hand*

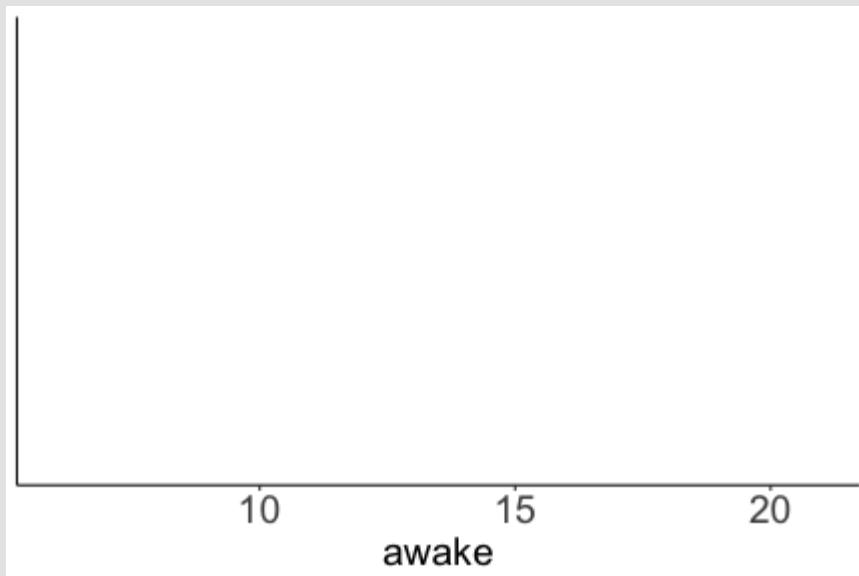
Histograms in ggplot2

```
ggplot(msleep_smol)
```



Histograms in ggplot2

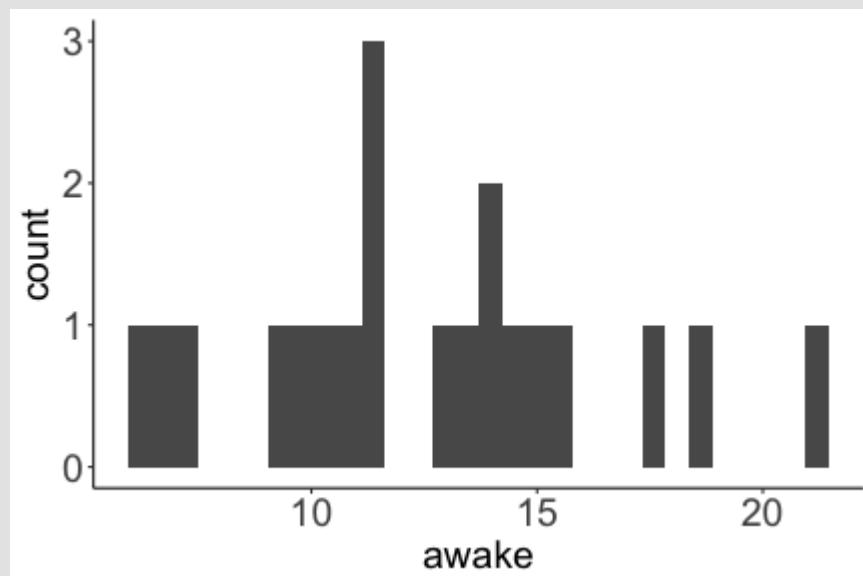
```
ggplot(msleep_smol) +  
  aes(x = awake)
```



Histograms in ggplot2

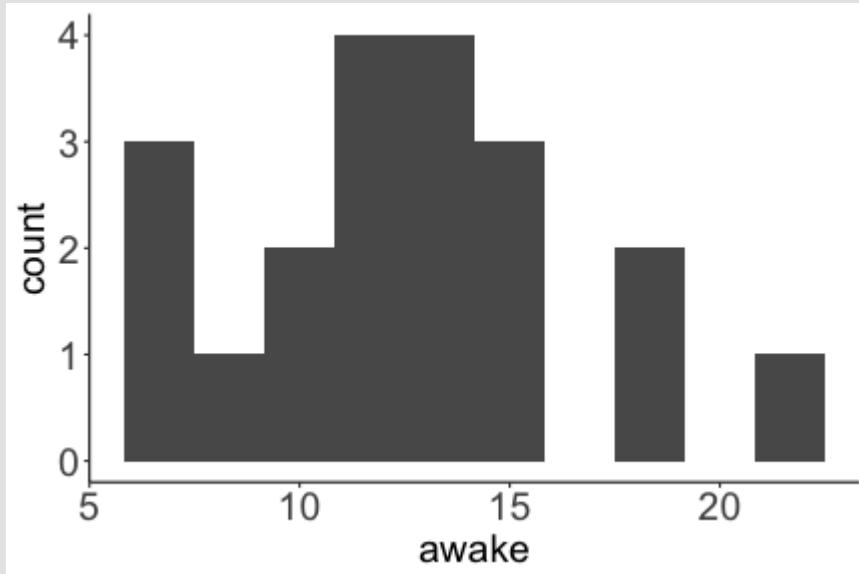
```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_histogram()
```

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



Histograms in ggplot2

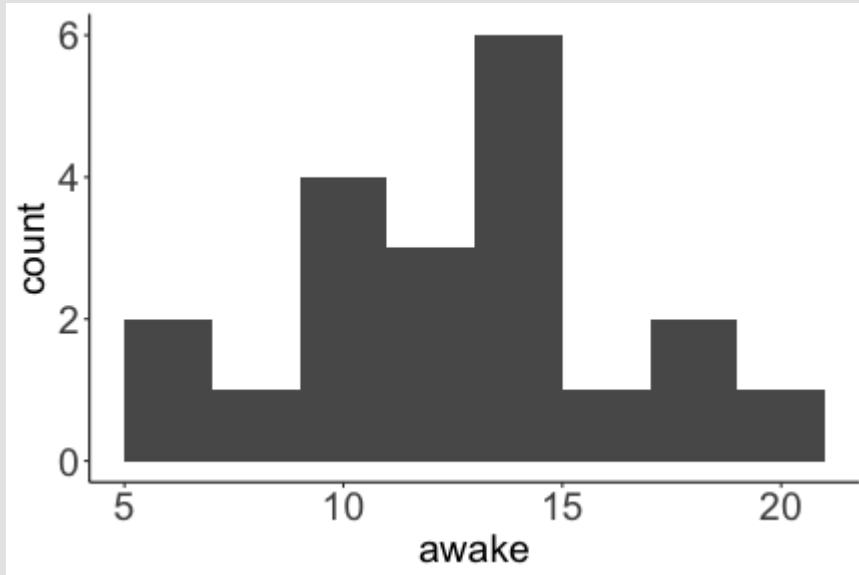
```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_histogram(bins = 10)
```



Use the argument `bins` to specify literally how many bins there should be

Histograms in ggplot2

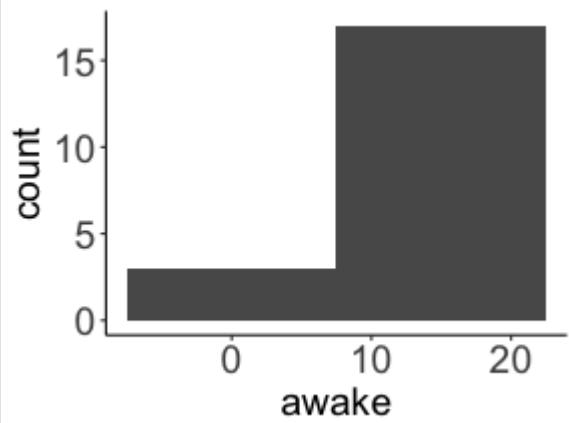
```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_histogram(binwidth = 2)
```



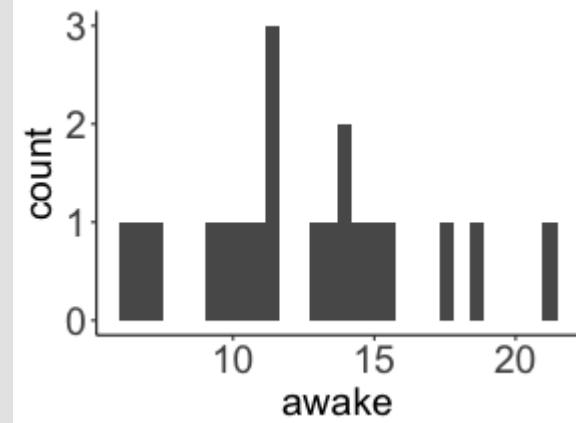
OR, use the argument **binwidth** to specify how wide the bins should be along the X-axis

"Trial and error" to find the "right" binning

```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_histogram(bins = 2)
```

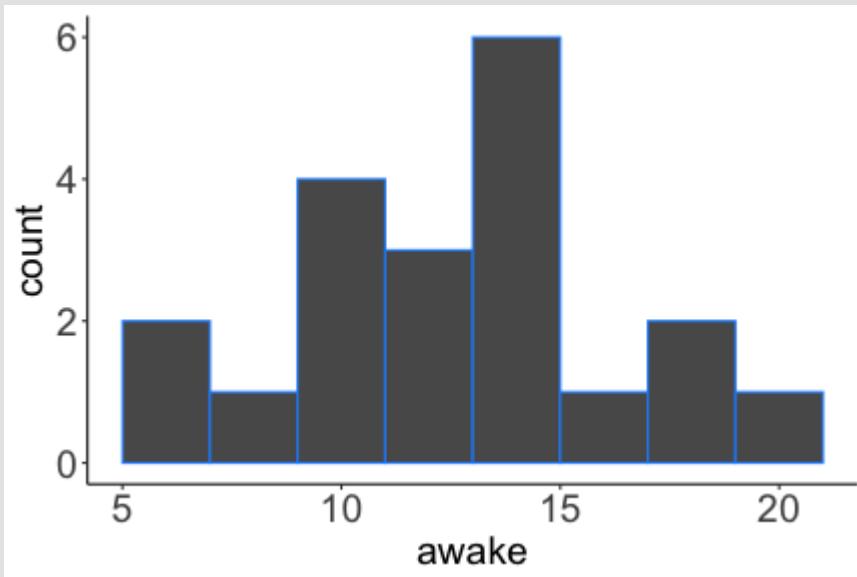


```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_histogram(bins = 30)
```



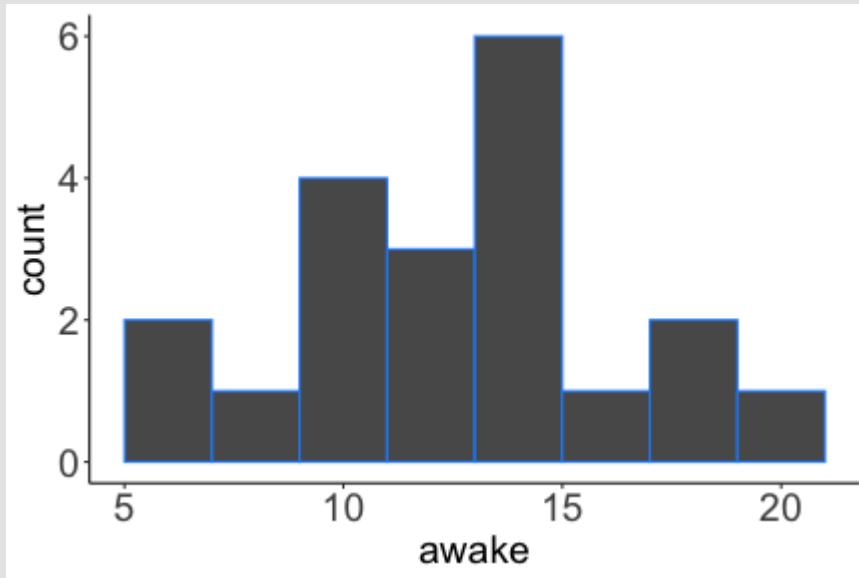
Histograms in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_histogram(binwidth = 2, color = "dodgerblue")
```



Histograms in ggplot2

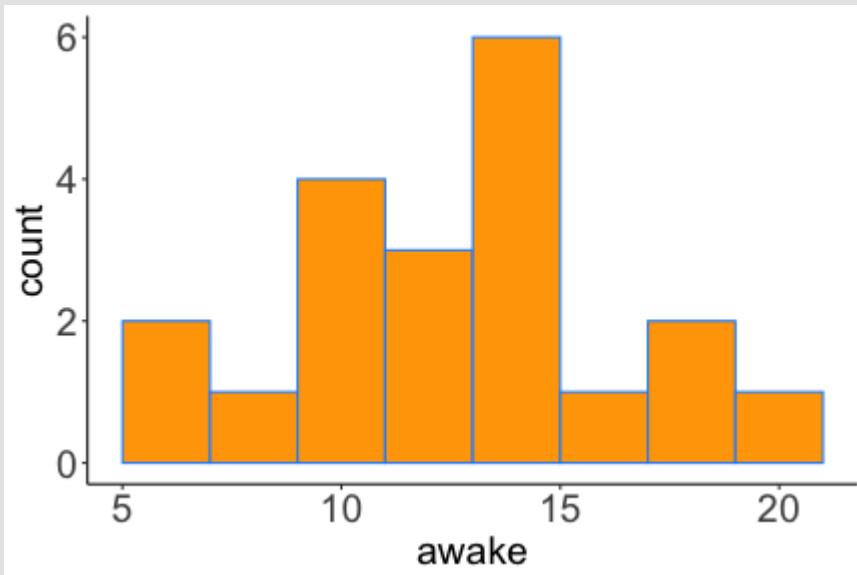
```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_histogram(binwidth = 2,  
                 color = "dodgerblue")
```



When code gets long, separate onto multiple lines for clarity. *YOU WILL BE GRADED ON THIS!!!!*

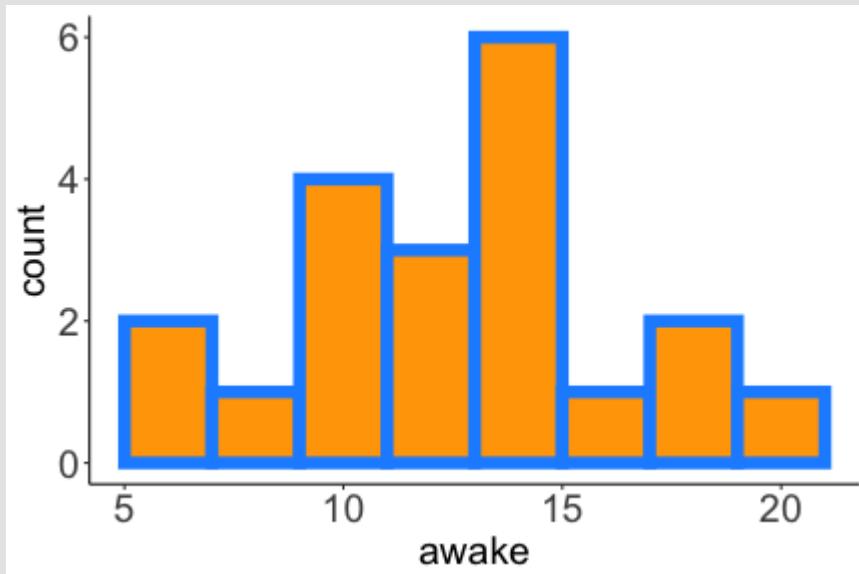
Histograms in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_histogram(binwidth = 2,  
                 color = "dodgerblue",  
                 fill = "orange")
```



Histograms in ggplot2

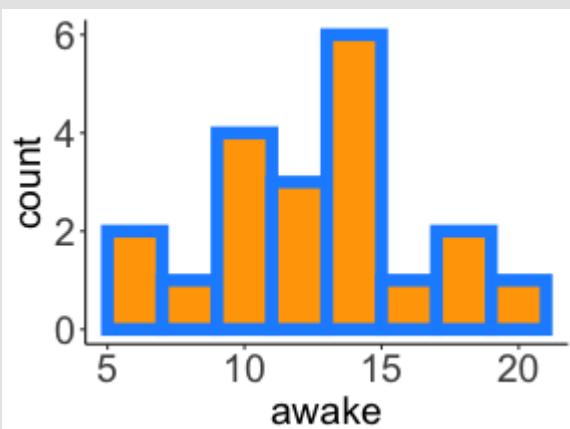
```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_histogram(binwidth = 2,  
                 color = "dodgerblue",  
                 fill = "orange",  
                 size = 3)
```



This is *pretty ugly*, but now you've learned about `size`!

COMMENT COMMENT COMMENT COMMENT

```
ggplot(msleep_smol) +  
  # specify x on the x-axis  
  aes(x = awake) +  
  geom_histogram(binwidth = 2, # use binwidth of 2  
                 # outline the histogram in blue  
                 color = "dodgerblue",  
                 # fill the bars with orange  
                 fill = "orange",  
                 # increase width of histogram lines  
                 size = 3)
```



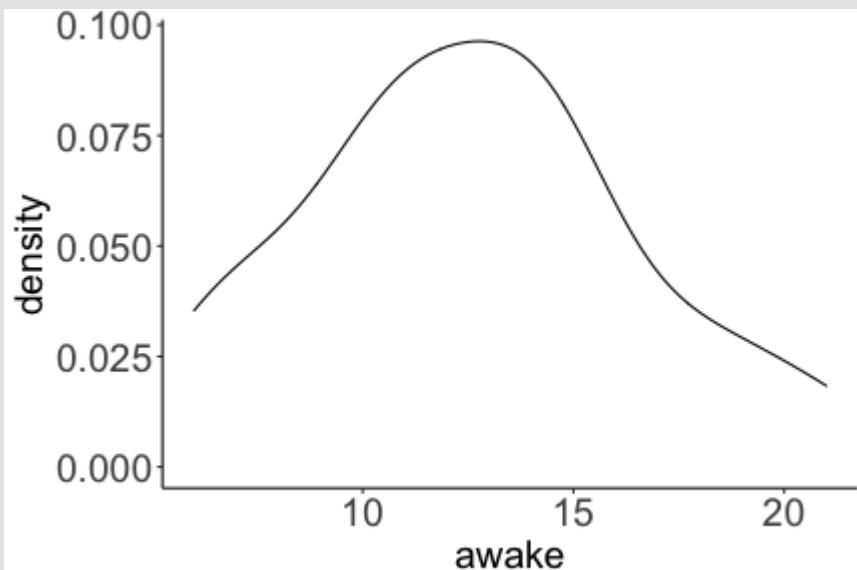
Density plots in ggplot2

Goal: Visualize the distribution of all mammal times spent awake (awake) for each vore

Step 1: Draw it by hand

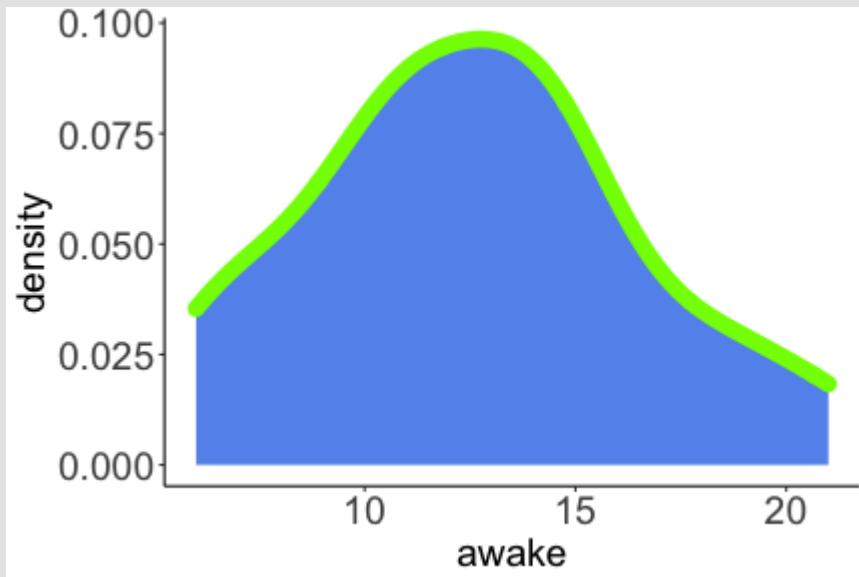
Density plots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_density()
```



Density plots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = awake) +  
  geom_density(fill = "cornflowerblue",  
              color = "chartreuse",  
              size = 4)
```



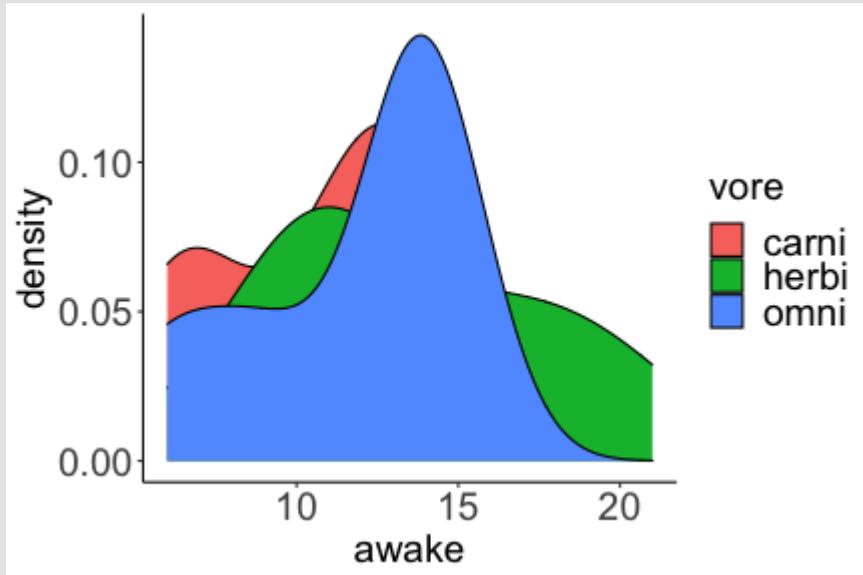
Again, kinda ugly, but look how easy that was to make!!

What if we wanted to show *all* vores separately?

Conceptualize by hand:

Density plots in ggplot2

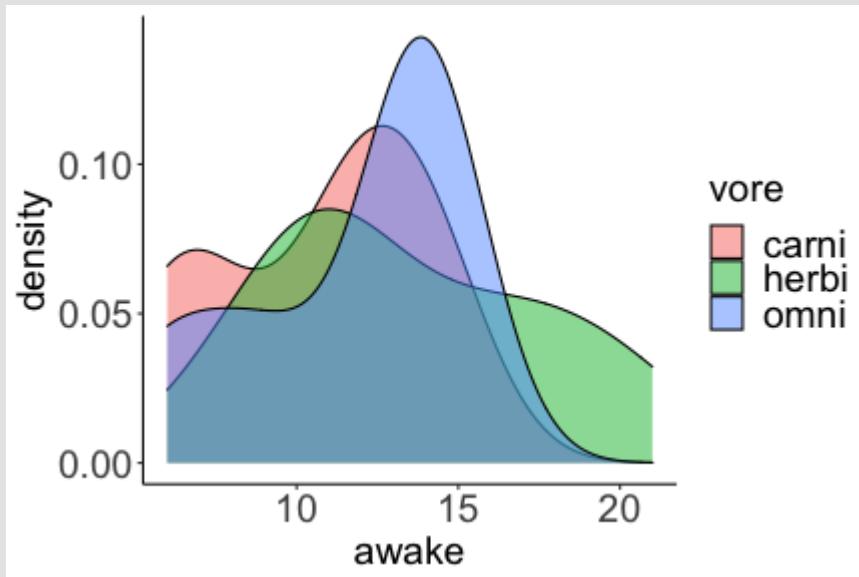
```
ggplot(msleep_smol) +  
  aes(x = awake, fill = vore) +  
  geom_density()
```



We need to make the density plots *transparent*.

Density plots in ggplot2

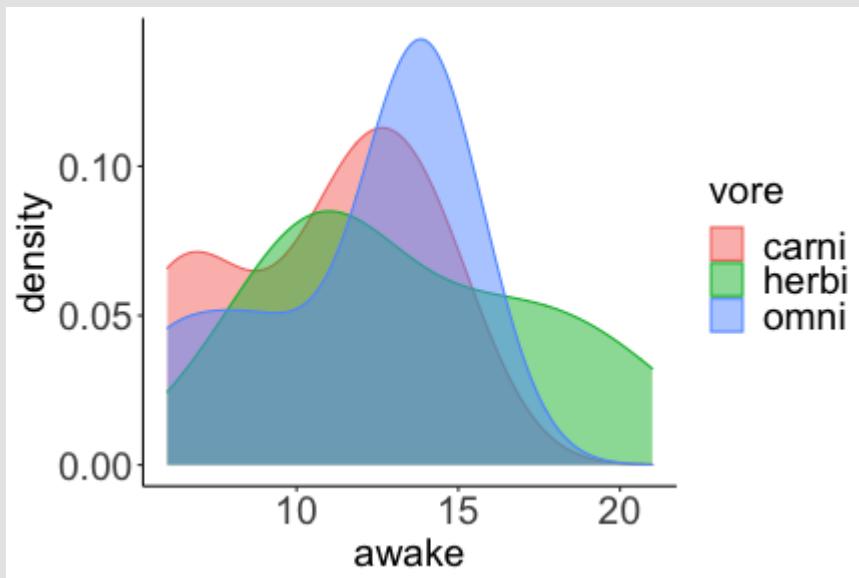
```
ggplot(msleep_smol) +  
  aes(x = awake, fill = vore) +  
  geom_density(alpha = 0.5)
```



- alpha is NOT a variable, so it does NOT belong in `aes()`. It is a visual aspect of the *density plot*, so we provide the argument to `geom_density()`
 - alpha = 1: Completely transparent.
 - alpha = 0: Completely opaque

Density plots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = awake,  
      fill = vore,  
      color = vore) +  
  geom_density(alpha = 0.5)
```



We will learn how to customize these colors next week

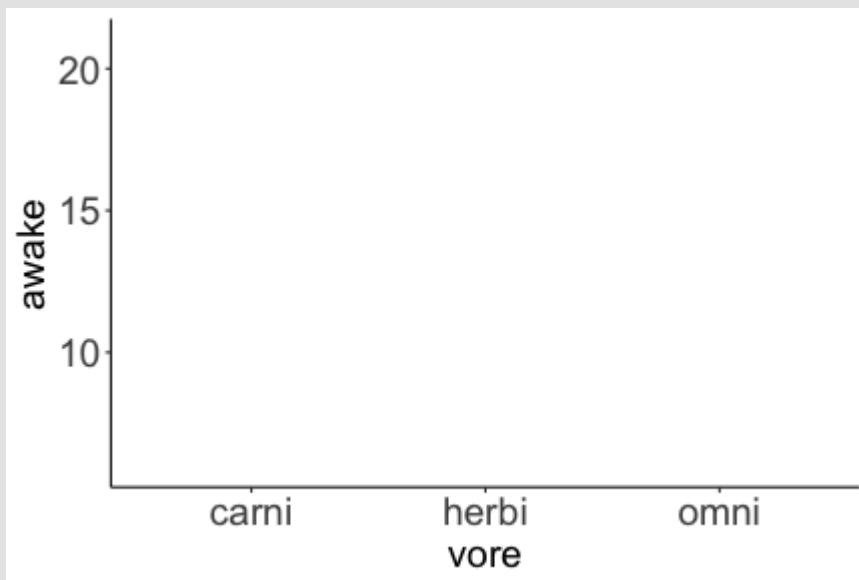
Boxplots in ggplot2

Goal: Visualize the distribution of all mammal times spent awake (awake) for each vore

Step 1:

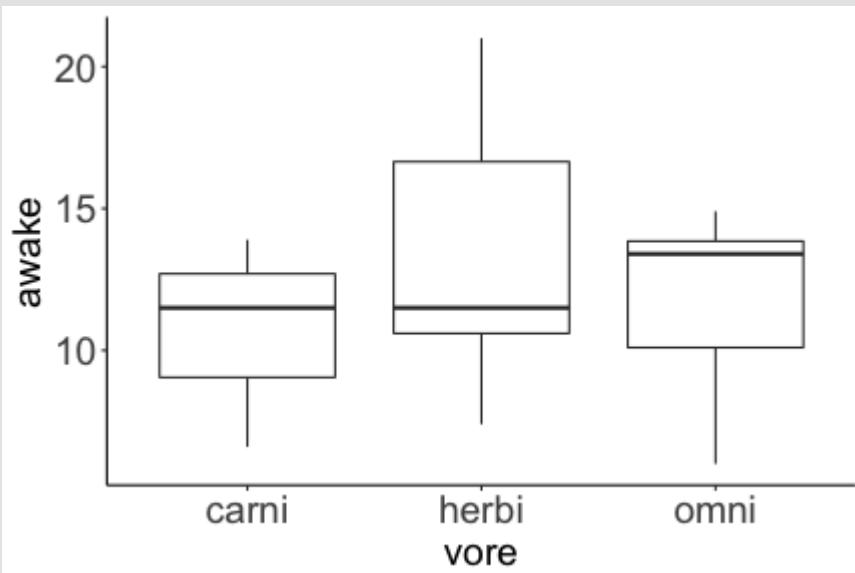
Boxplots in ggplot2

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



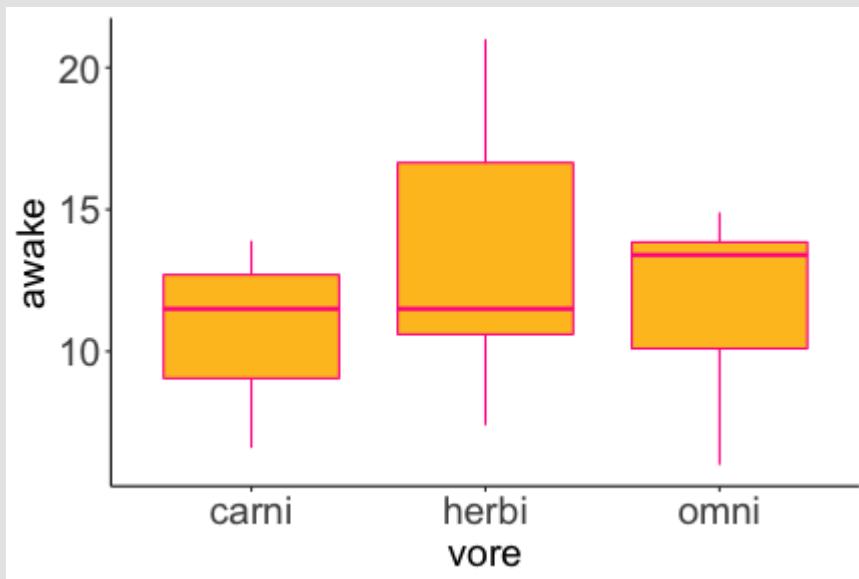
Boxplots in ggplot2

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



Boxplots in ggplot2

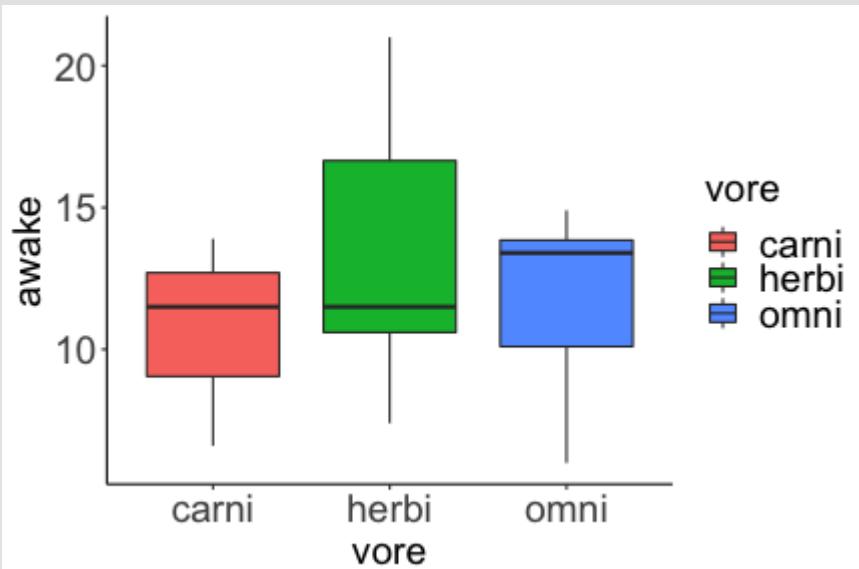
```
ggplot(msleep_smol) +  
  aes(x = vore, y = awake) +  
  geom_boxplot(color = "deeppink",  
               fill   = "goldenrod1")
```



What if we want a separate fill *for each vore?*

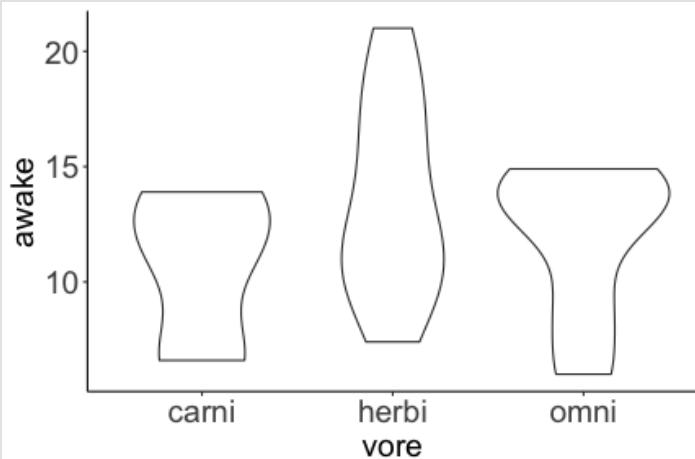
Boxplots in ggplot2

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

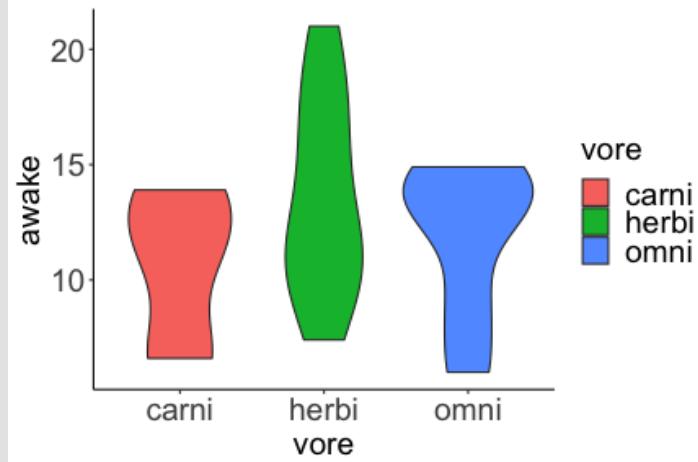


Violin plots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore, y = awake) +  
  geom_violin()
```

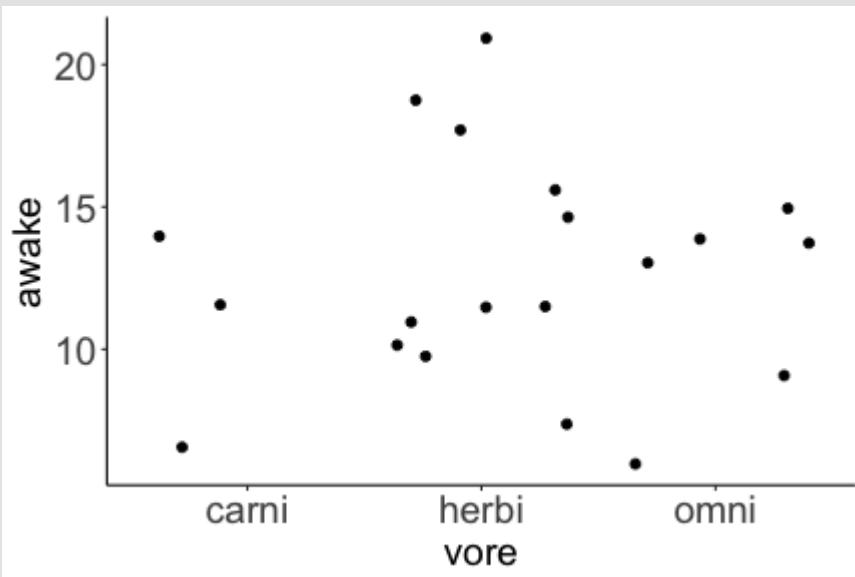


```
ggplot(msleep_smol) +  
  aes(x = vore,  
      y = awake,  
      fill = vore) +  
  geom_violin()
```



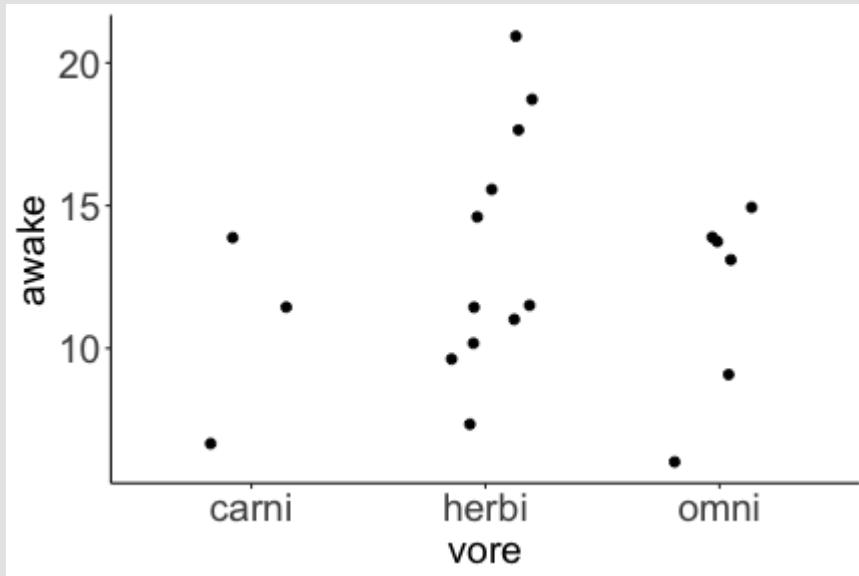
Strip/jitter plots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore, y = awake) +  
  # size = 2 to see more easily in slides  
  geom_jitter(size = 2)
```



Strip/jitter plots in ggplot2

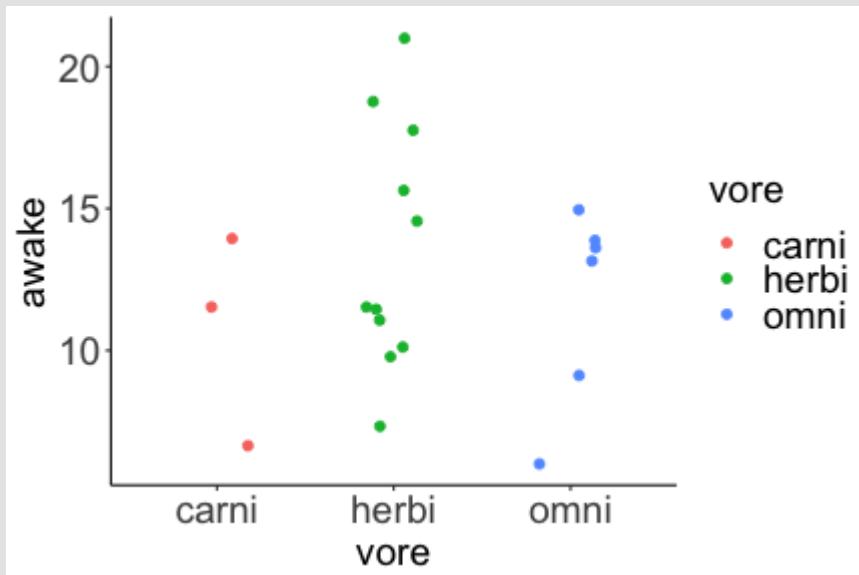
```
ggplot(msleep_smol) +  
  aes(x = vore, y = awake) +  
  geom_jitter(width = 0.2,  
              size   = 2)
```



Usually need to pick a width between 0.1 - 0.3, *in my opinion*

Strip/jitter plots in ggplot2

```
ggplot(msleep_smol) +  
  # COLOR, NOT FILL!! These are points  
  aes(x = vore,  
      y = awake,  
      color = vore) +  
  geom_jitter(width = 0.2, size = 2)
```



Did you notice the points were placed little differently each plot version?

Barplots in ggplot2

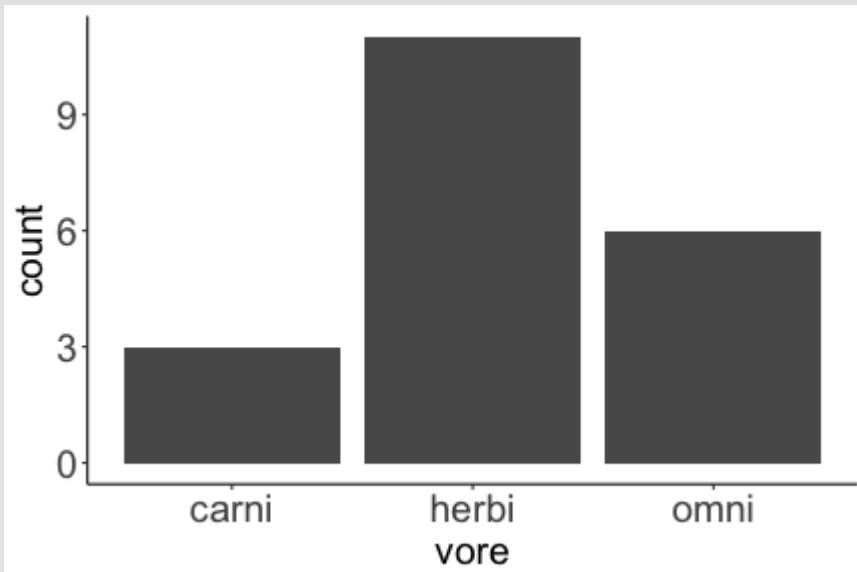
Goal: How many of each vote are there?

Note: There are many ways to make barplots. Today we will learn how to make barplots specifically to show *amount of a categorical variable*.

Step 1:

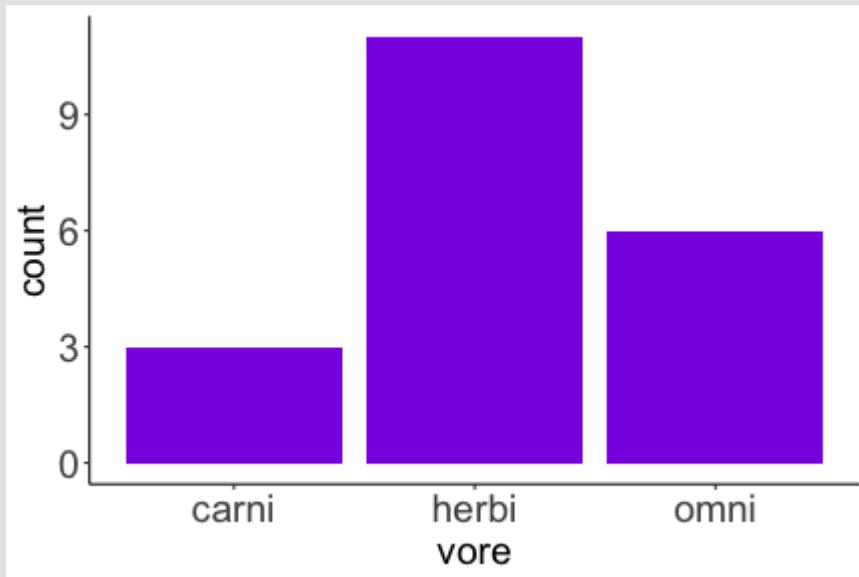
Barplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore) +  
  geom_bar()
```



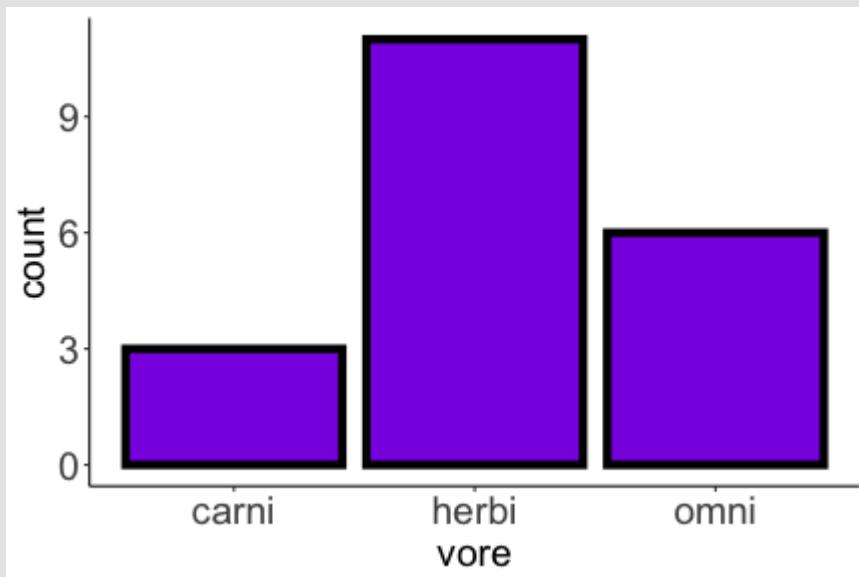
Barplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore) +  
  geom_bar(fill = "blueviolet")
```



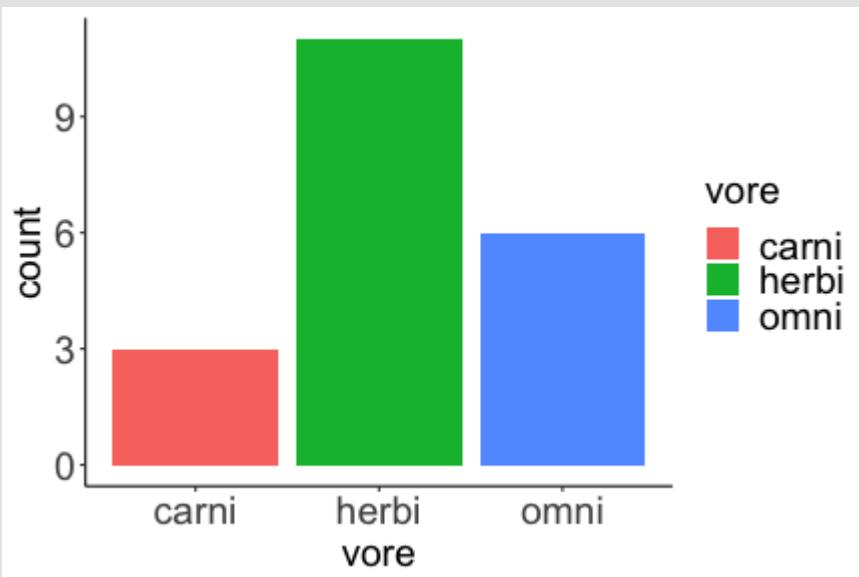
Barplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore) +  
  geom_bar(fill = "blueviolet",  
           color = "black", size = 2)
```



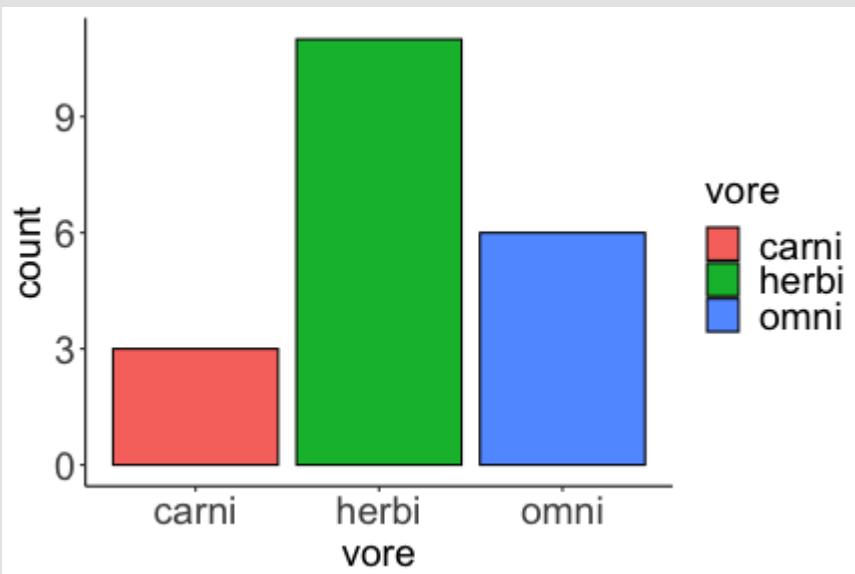
Barplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore,  
      fill = vore) +  
  geom_bar()
```



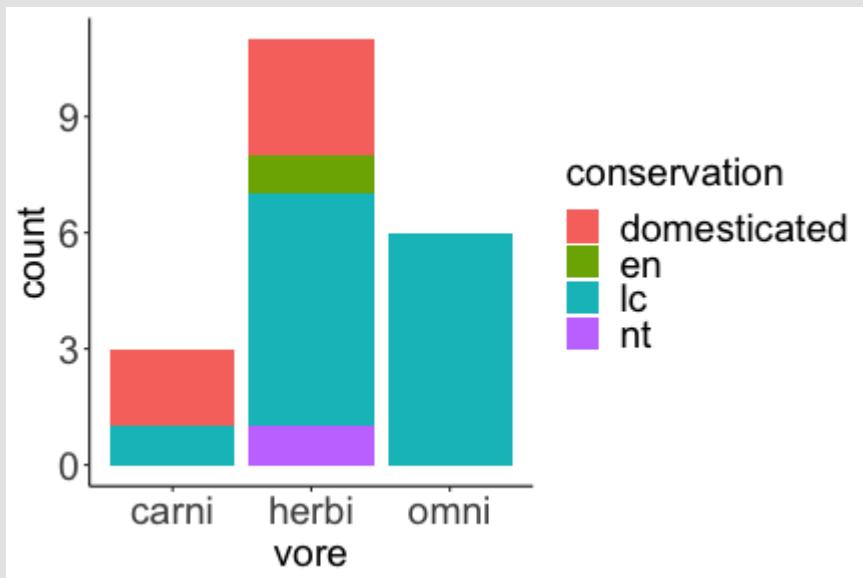
Barplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore,  
      fill = vore) +  
  geom_bar(color = "black")
```



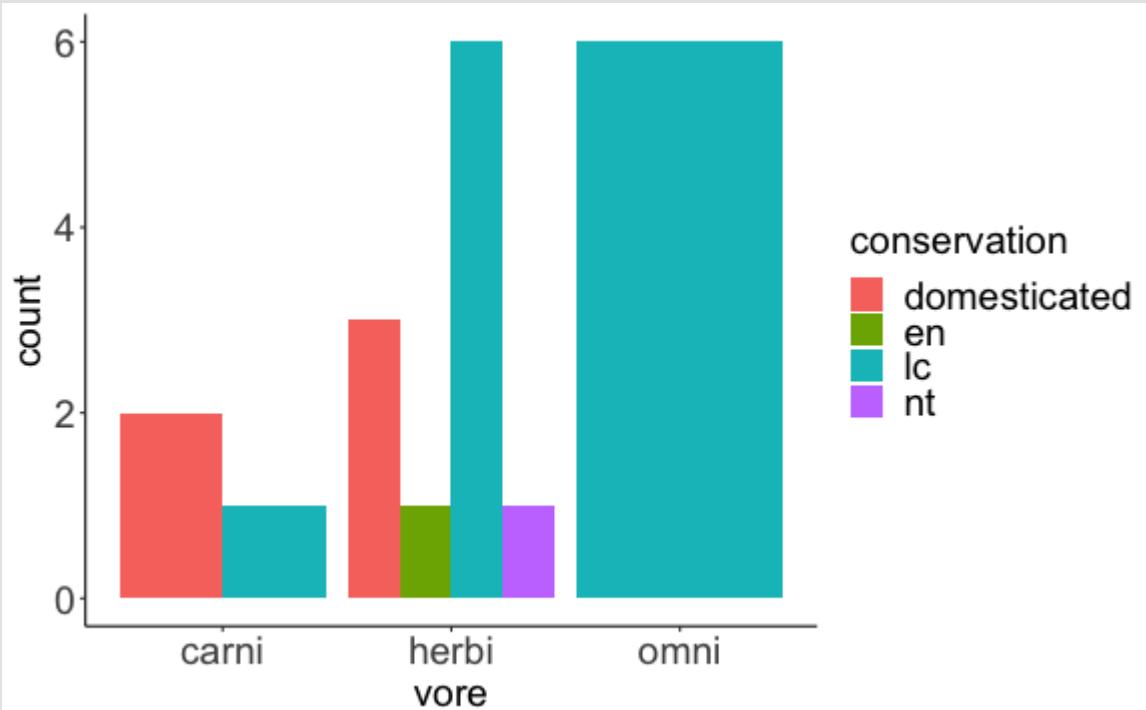
Barplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore,  
      fill = conservation) +  
  geom_bar()
```



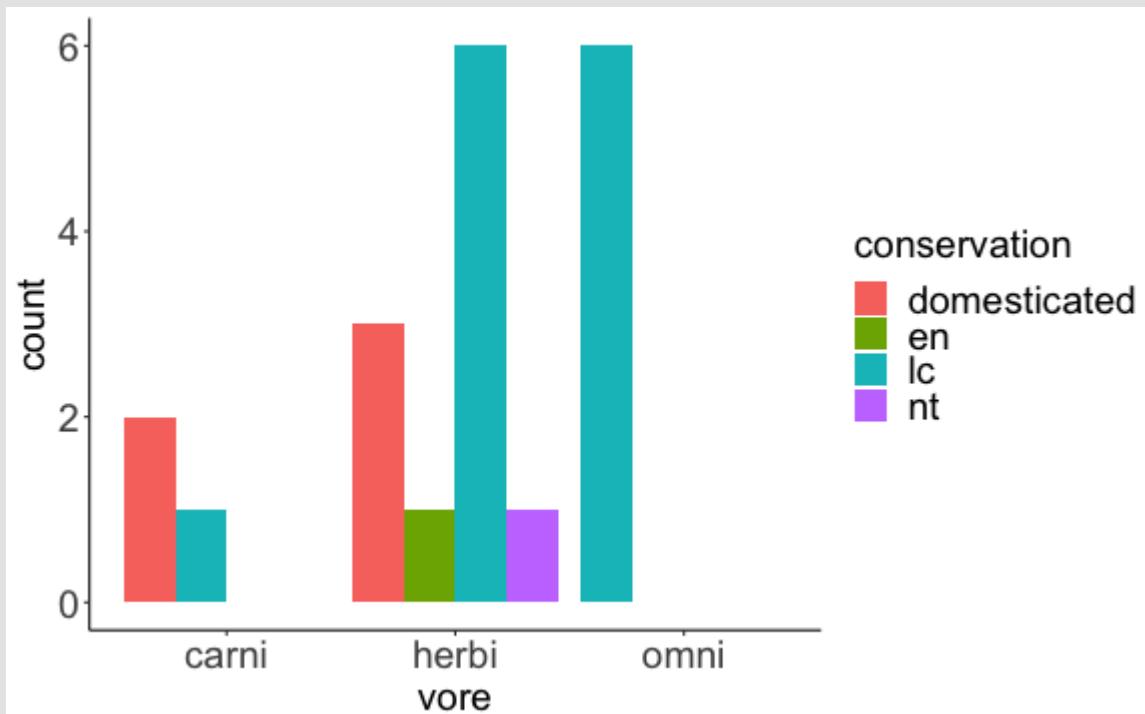
Barplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore,  
      fill = conservation) +  
  geom_bar(position = position_dodge())
```



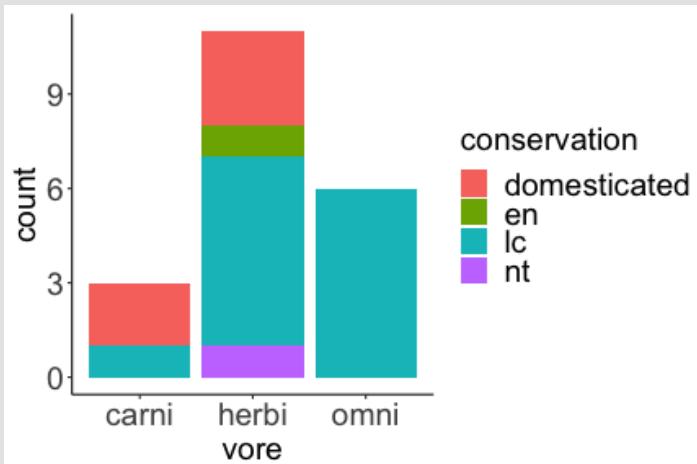
Barplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore,  
      fill = conservation) +  
  geom_bar(position = position_dodge(preserve="single"))
```

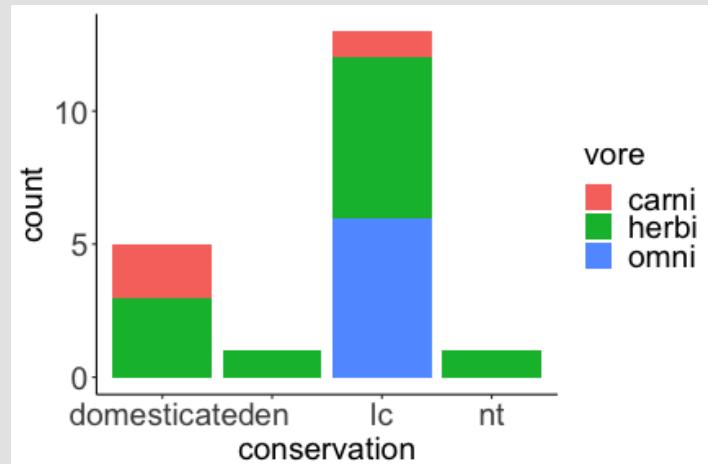


Barplots in ggplot2

```
ggplot(msleep_smol) +  
  aes(x = vore,  
      fill = conservation) +  
  geom_bar()
```

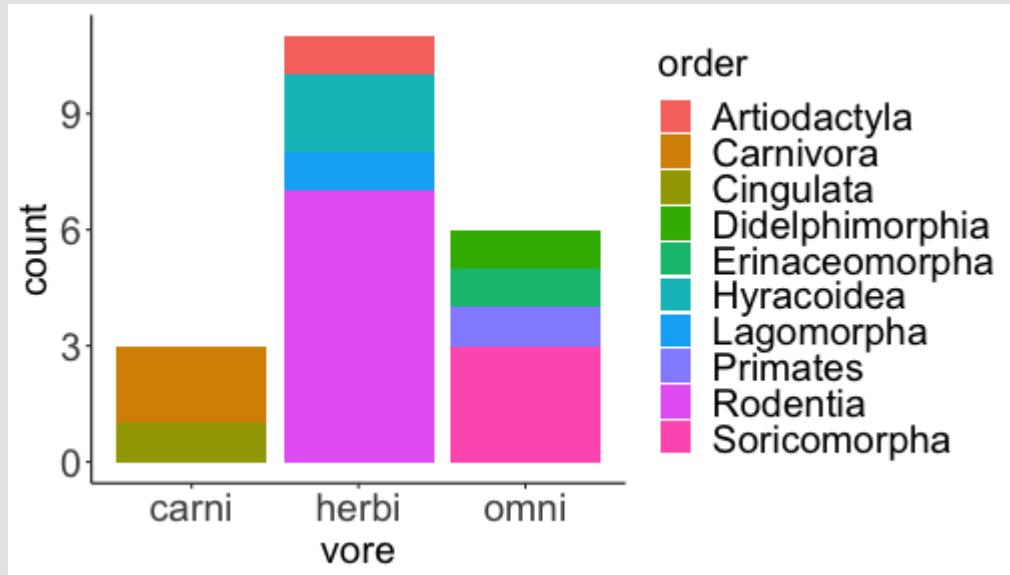


```
ggplot(msleep_smol) +  
  aes(x = conservation,  
      fill = vore) +  
  geom_bar()
```



Barplots in ggplot2

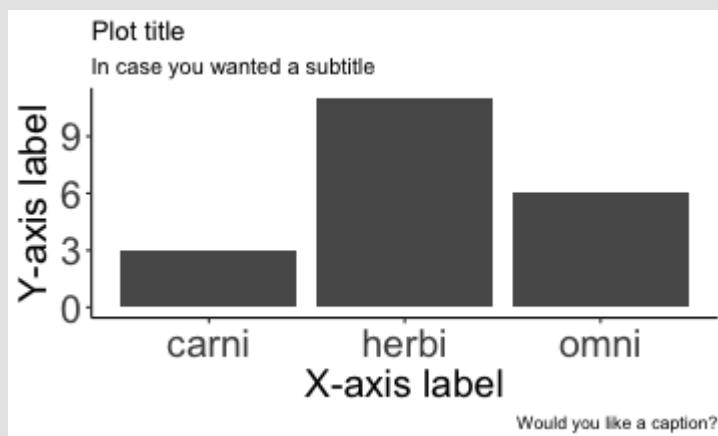
```
ggplot(msleep_smol) +  
  aes(x = vore,  
      fill = order) +  
  geom_bar()
```



My first impression of this plot is *feeling overwhelmed*.

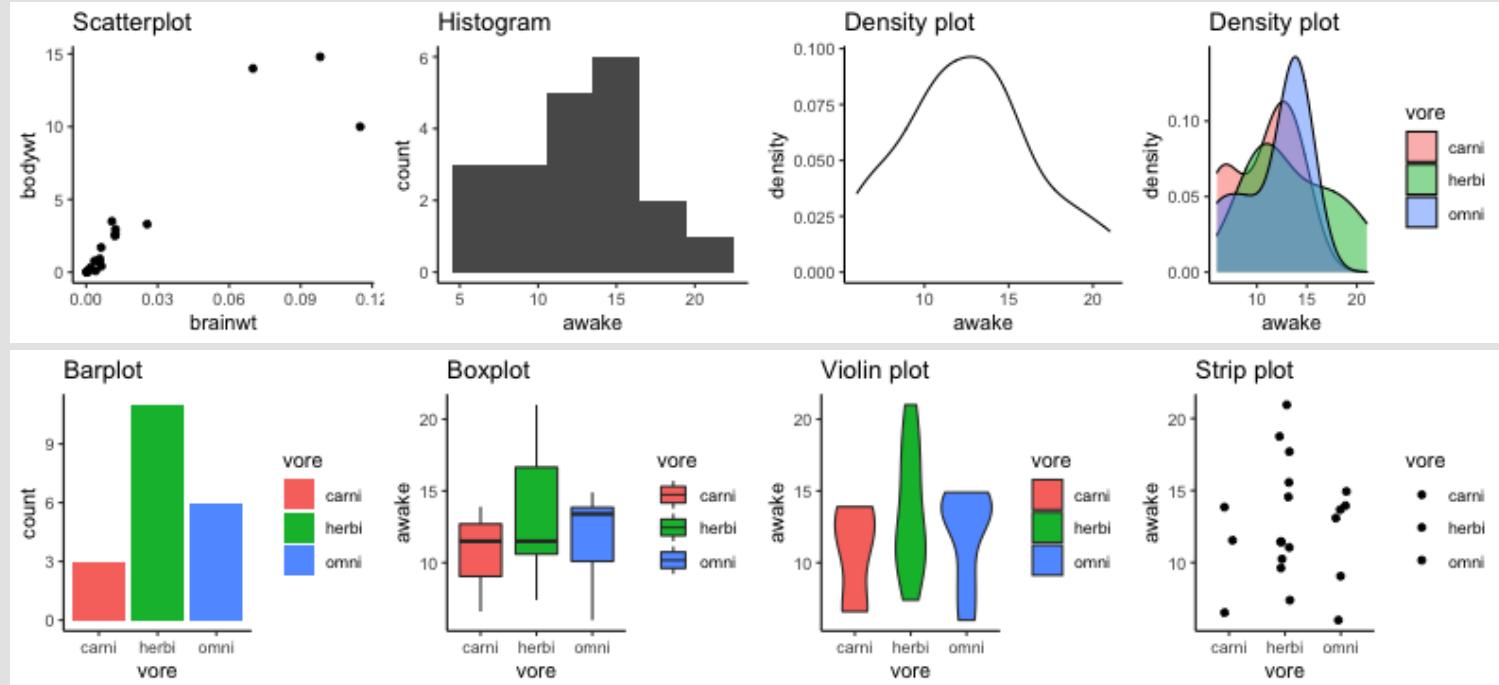
Labeling your plots

```
ggplot(msleep_smol) +  
  aes(x = vore) +  
  geom_bar() +  
  labs(x = "X-axis label",  
       y = "Y-axis label",  
       title = "Plot title",  
       subtitle = "In case you wanted a subtitle",  
       caption = "Would you like a caption?")
```



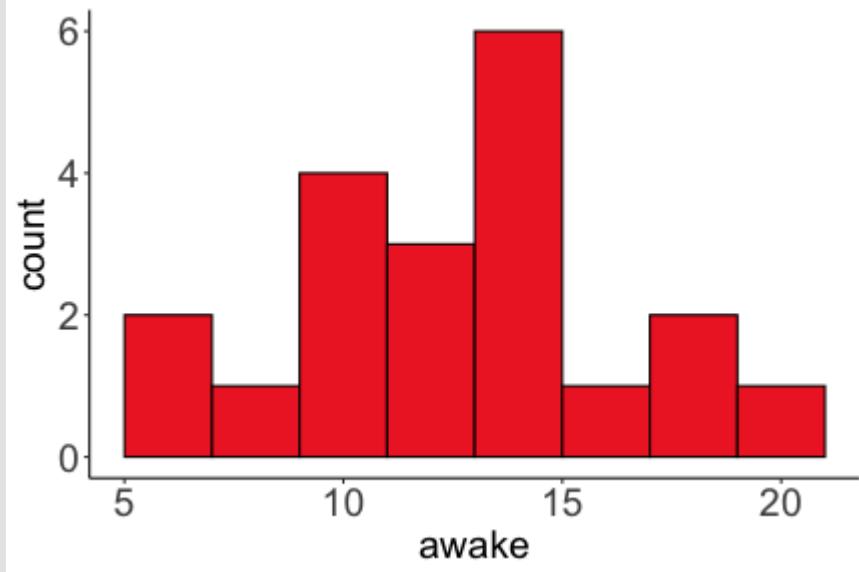
Always make sure your x and y axes are professionally labeled (*no underscores!*). Title, subtitle, caption are optional to include.

Plots we can make!



Adding panels ("facets") to plots

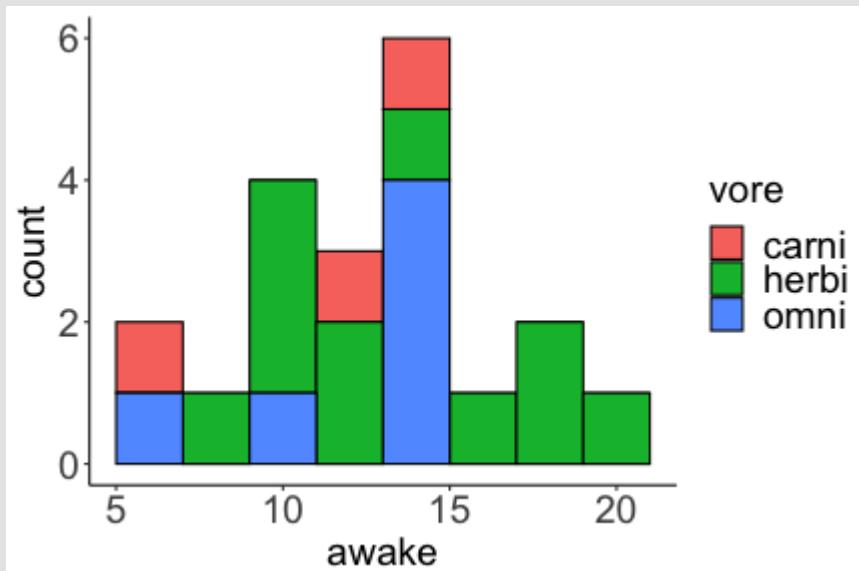
```
ggplot(msleep_smol, aes(x = awake)) +  
  geom_histogram(binwidth = 2,  
                 color = "black",  
                 fill = "firebrick2")
```



What if we want a separate histogram for each **vore**?

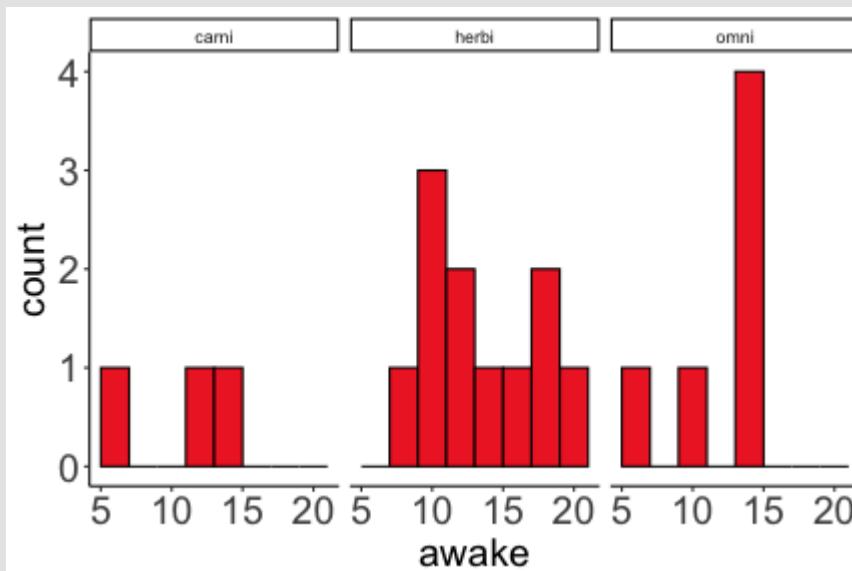
This is a bad idea.

```
ggplot(msleep_smol, aes(x = awake, fill = vore)) +  
  geom_histogram(binwidth = 2,  
                 color = "black")
```



Adding panels ("facets") to plots

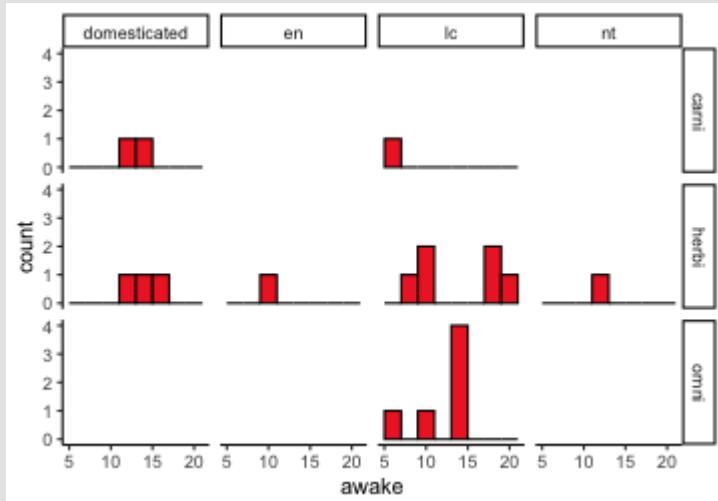
```
ggplot(msleep_smol, aes(x = awake)) +  
  geom_histogram(binwidth = 2, # may need to tweak again!  
                 color = "black",  
                 fill = "firebrick2") +  
  facet_wrap( vars(vore) )
```



Faceting is the *ONLY TIME* we do not use `aes()` to refer to a column!!! We use `vars()` when facetting.

Adding panels ("facets") to plots

```
ggplot(msleep_smol, aes(x = awake)) +  
  geom_histogram(binwidth = 2, # may need to tweak again!  
                 color = "black",  
                 fill = "firebrick2") +  
  # you get to choose which is row and which is column  
  facet_grid( rows = vars(vore),  
              cols = vars(conservation) )
```



Use `facet_grid()` to make a panel grid across 2 variables. This plot is *pretty bad*, but it teaches `facet_grid()`!