PLOTTING ANYTHING WITH

GGPLOT2

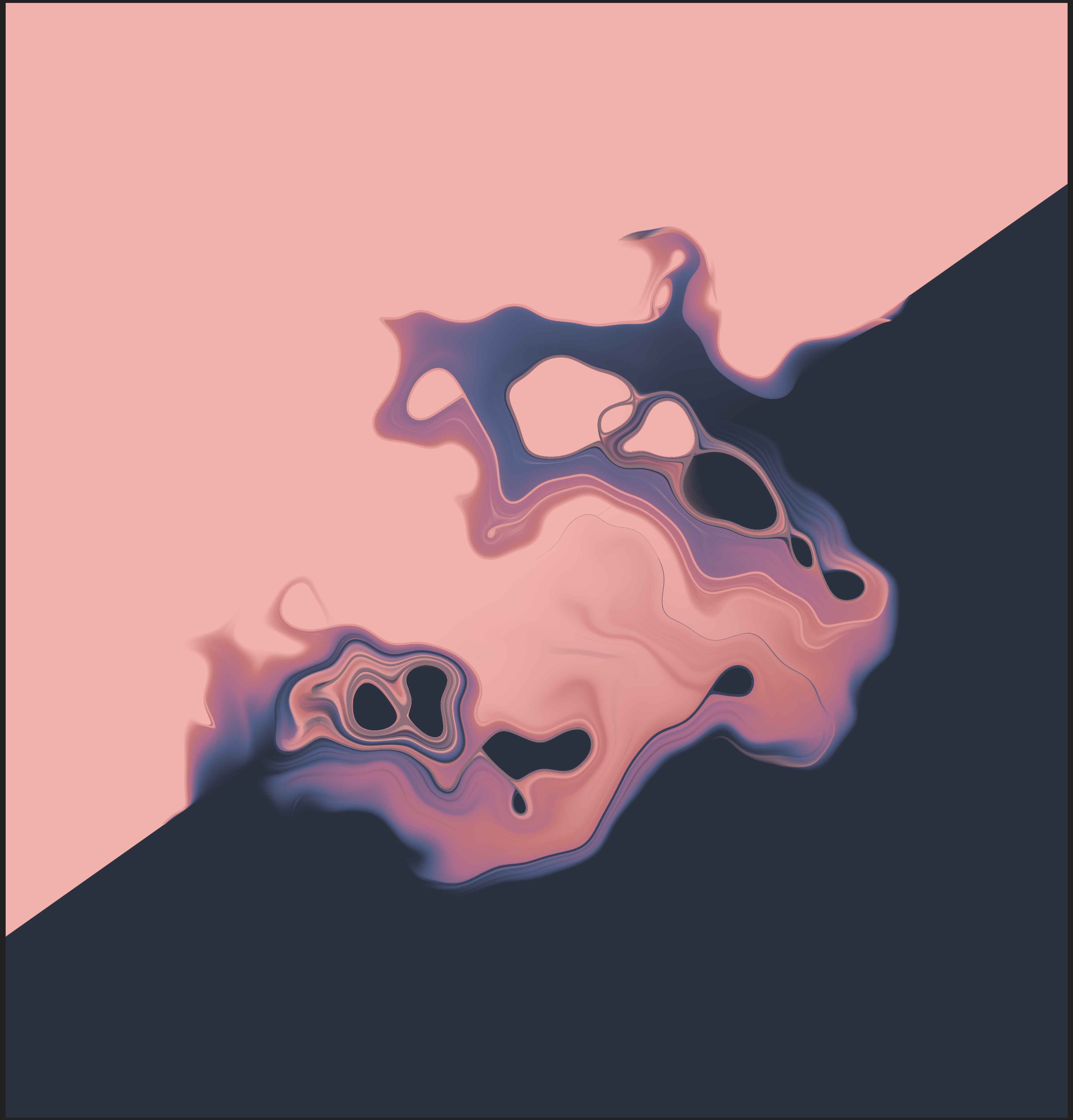
ABOUT ME

- Thomas Lin Pedersen
- Software Engineer at RStudio
- Once a bioinformatician
- Focus on graphics

🌐 data-imaginist.com

🐦 @thomasp85

@thomasp85



ABOUT TODAY

- The Grammar of Graphics

  … on why a theoretical foundation fixes everything

- The ggplot2 API

  … where we learn that every benefit has costs

- Beyond ggplot2

  … any glass ceiling can be shattered

- Drawing anything

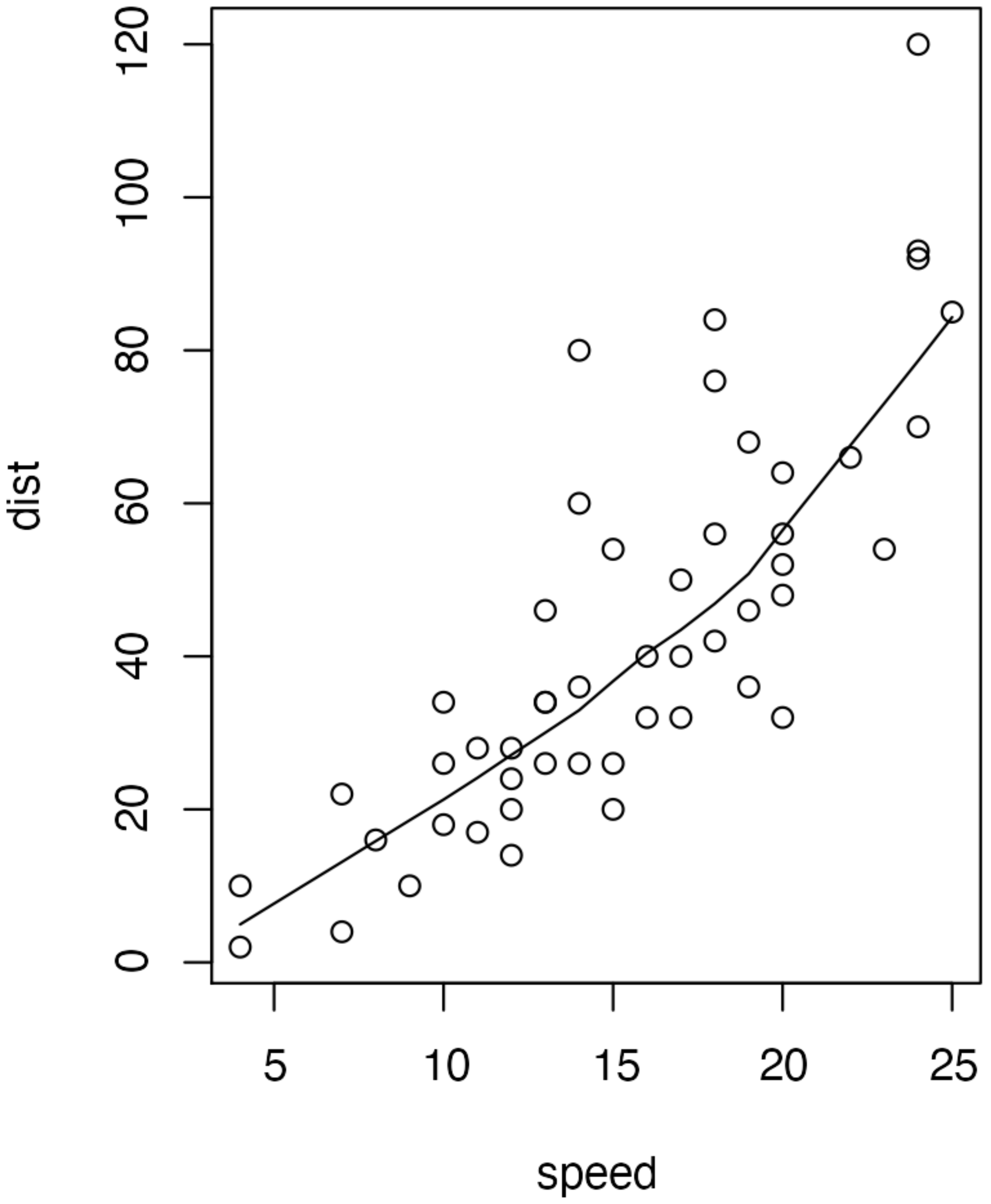
  … in which we see that the benefits far outweighs the cost

NOT ABOUT TODAY

- Data import
  - readr - readr.tidyverse.org
  - readxl - readxl.tidyverse.org
  - haven - haven.tidyverse.org
- Data manipulation
  - tidyr - tidyr.tidyverse.org
  - dplyr - dplyr.tidyverse.org
  - data.table - r-datatable.com
- R programming
  - R for Data Science - r4ds.had.co.nz
  - Advanced R Programming - adv-r.hadley.nz
  - R Packages - r-pkgs.org

INTRODUCTION

- RStudio Cloud
  - https://rstd.io/celebration_ggplot2
- Github Gist with code and exercises (only if you don't use RStudio Cloud)
  - https://rstd.io/celebration_code
- We will work with the next version of ggplot2 (3.3.0)

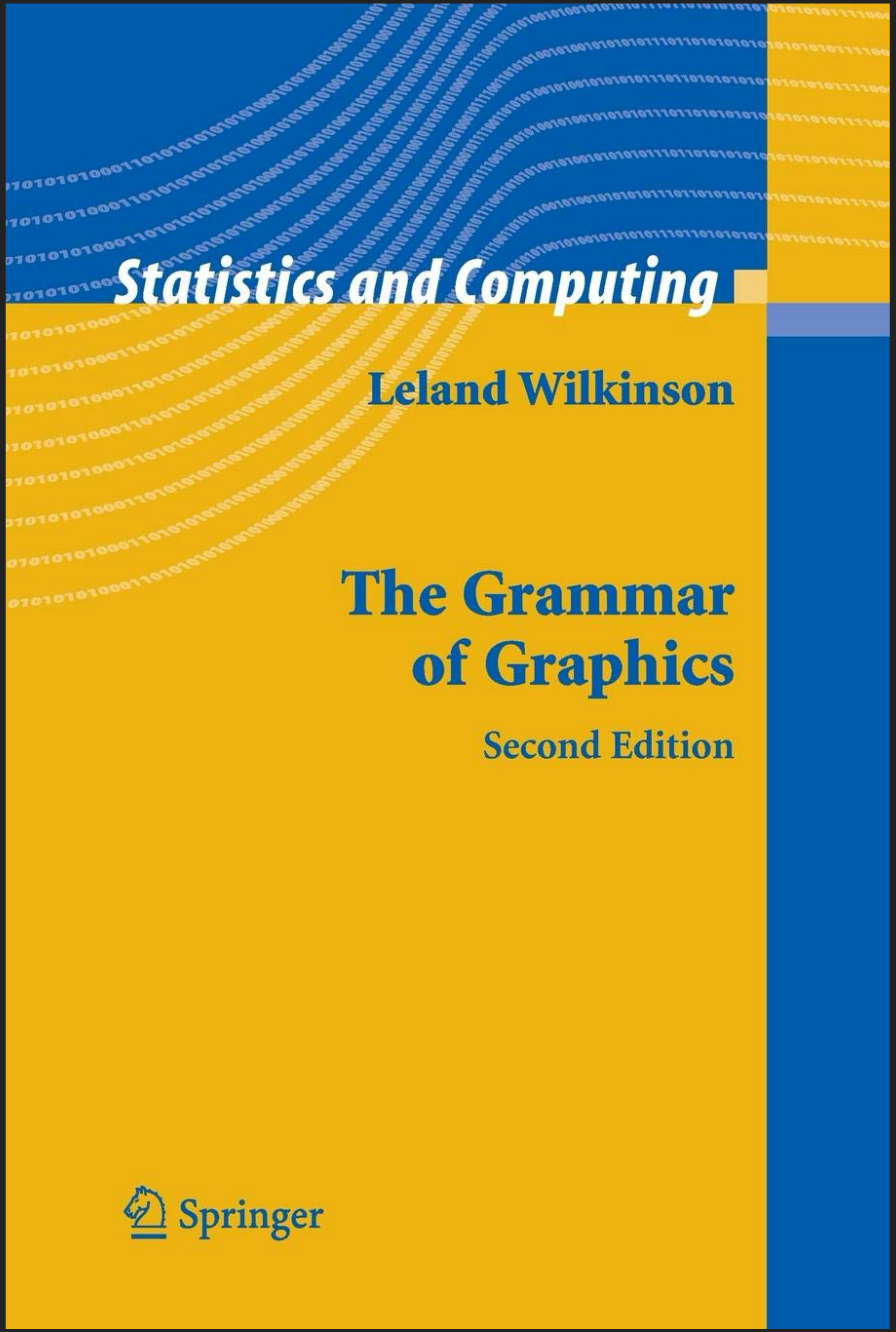


PART 1:

THE GRAMMAR OF GRAPHICS

THE BOOK

- 1st edition in 1999
- A theoretical deconstruction of data graphics
- Foundation for many graphic applications
  - ggplot2
  - Polaris (→ Tableau)
  - Vega-Lite



Statistics and Computing

Leland Wilkinson

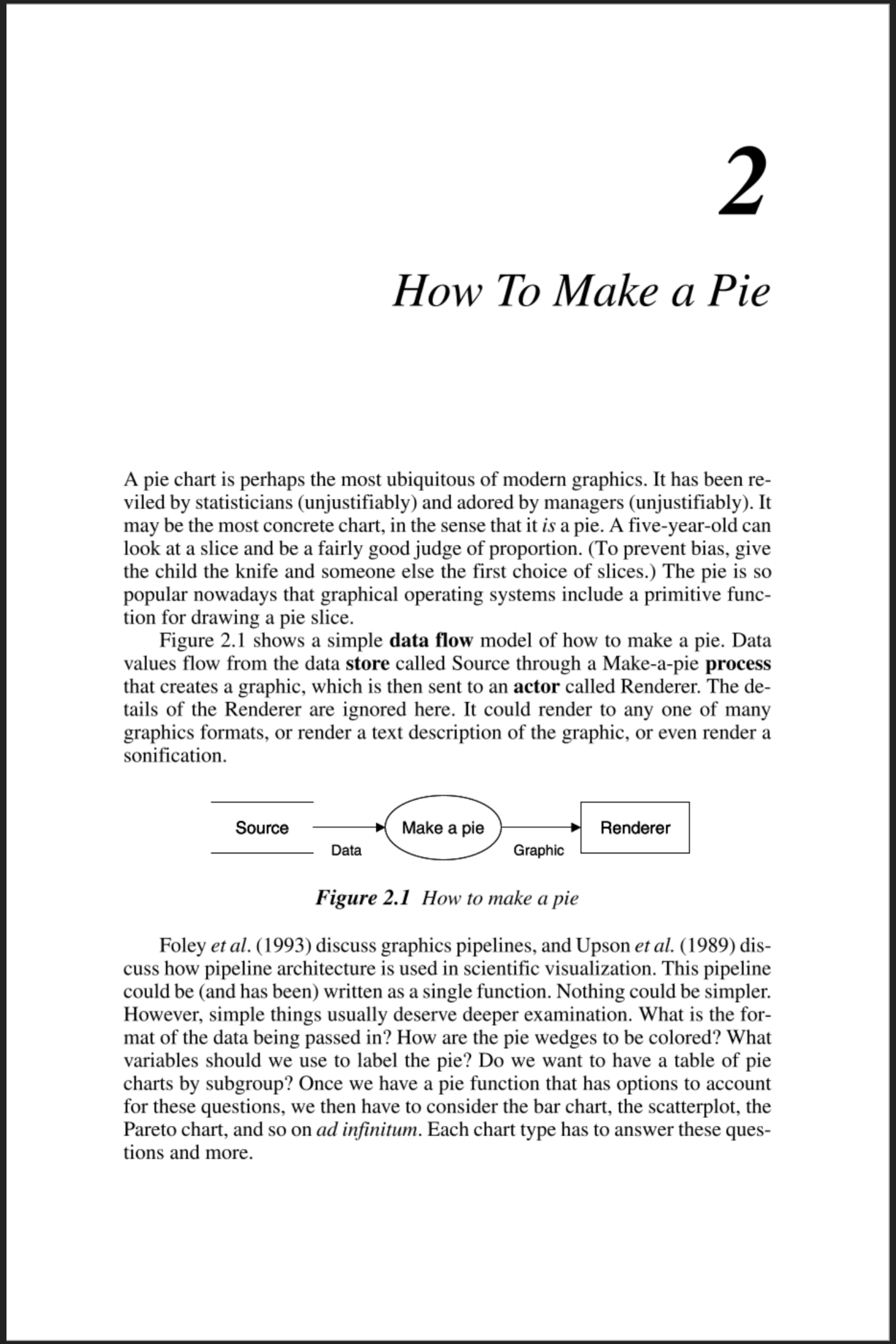
The Grammar of Graphics

Second Edition

Springer

THE BOOK

- Not interested in why
  - *why choose this or that chart type?*
- Not interested in beauty
  - *how do you make attractive charts?*
- Not interested in algorithms
  - *how do you calculate the correct positions of the data?*
- Very interested in how to design the system that allows all that



2

How To Make a Pie

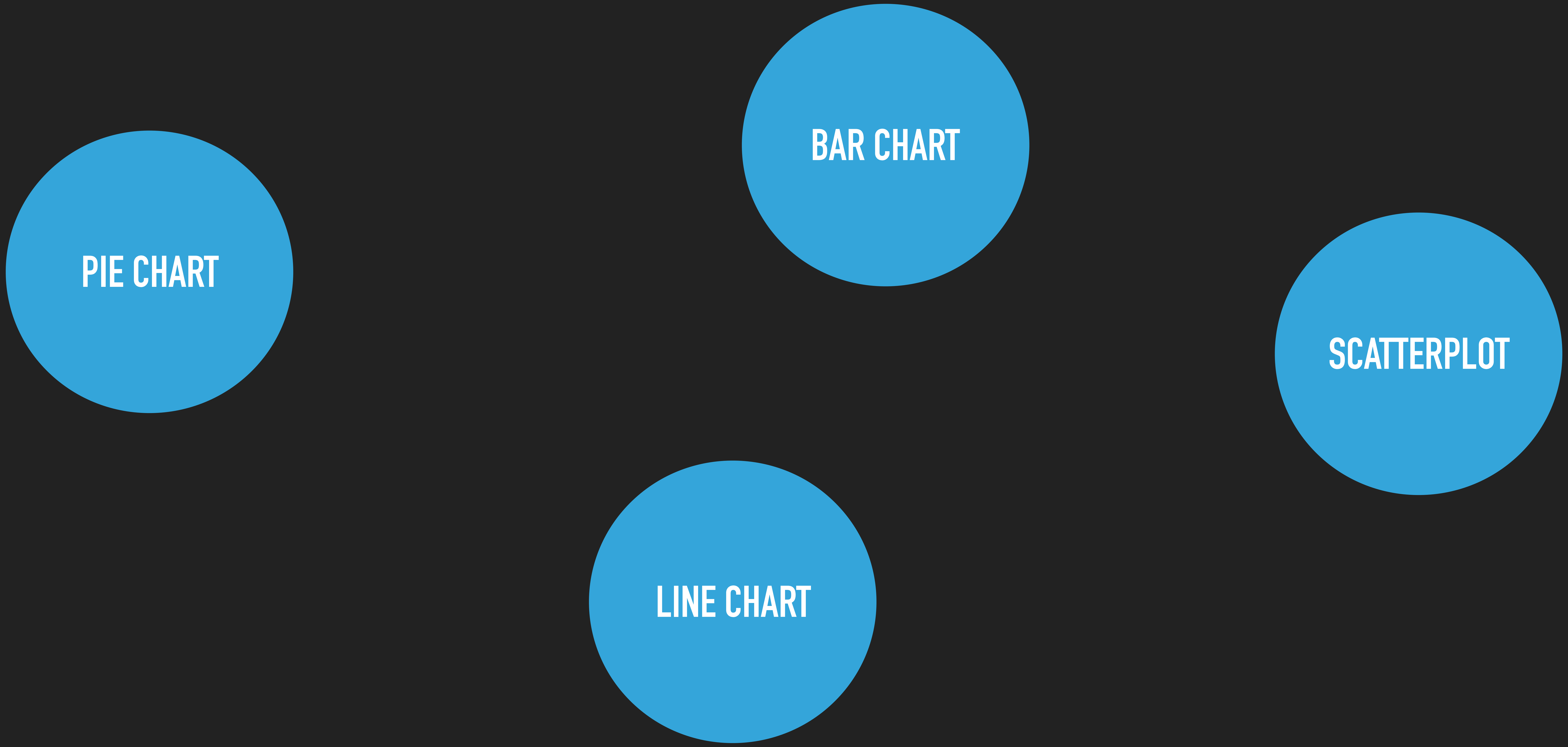
A pie chart is perhaps the most ubiquitous of modern graphics. It has been reviled by statisticians (unjustifiably) and adored by managers (unjustifiably). It may be the most concrete chart, in the sense that it *is* a pie. A five-year-old can look at a slice and be a fairly good judge of proportion. (To prevent bias, give the child the knife and someone else the first choice of slices.) The pie is so popular nowadays that graphical operating systems include a primitive function for drawing a pie slice.

Figure 2.1 shows a simple **data flow** model of how to make a pie. Data values flow from the data **store** called Source through a Make-a-pie **process** that creates a graphic, which is then sent to an **actor** called Renderer. The details of the Renderer are ignored here. It could render to any one of many graphics formats, or render a text description of the graphic, or even render a sonification.

Figure 2.1 *How to make a pie*

Foley *et al*. (1993) discuss graphics pipelines, and Upson *et al.* (1989) discuss how pipeline architecture is used in scientific visualization. This pipeline could be (and has been) written as a single function. Nothing could be simpler. However, simple things usually deserve deeper examination. What is the format of the data being passed in? How are the pie wedges to be colored? What variables should we use to label the pie? Do we want to have a table of pie charts by subgroup? Once we have a pie function that has options to account for these questions, we then have to consider the bar chart, the scatterplot, the Pareto chart, and so on *ad infinitum*. Each chart type has to answer these questions and more.

THE IDEA



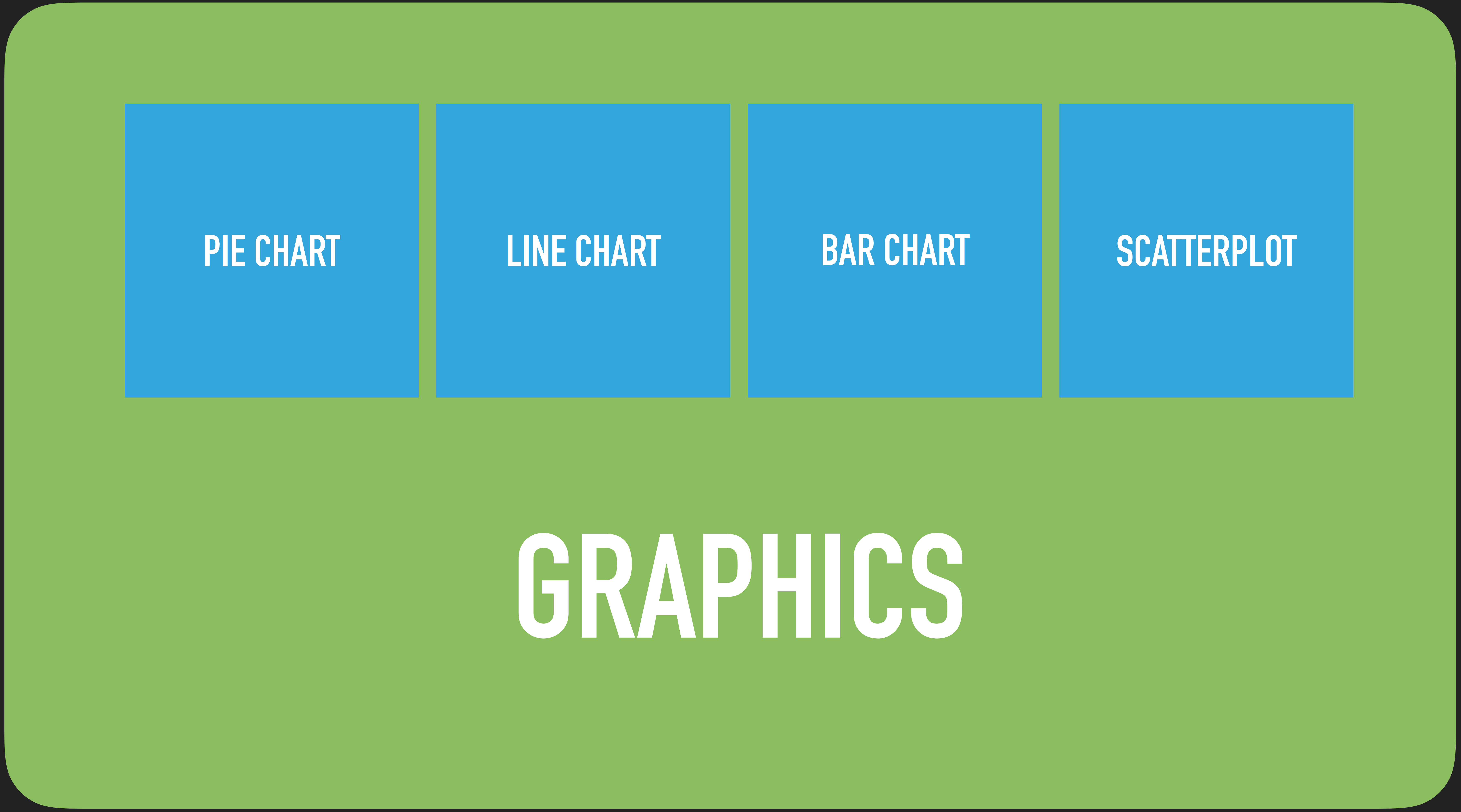
BAR CHART

PIE CHART

SCATTERPLOT

LINE CHART

THE IDEA



PIE CHART

LINE CHART

BAR CHART

SCATTERPLOT

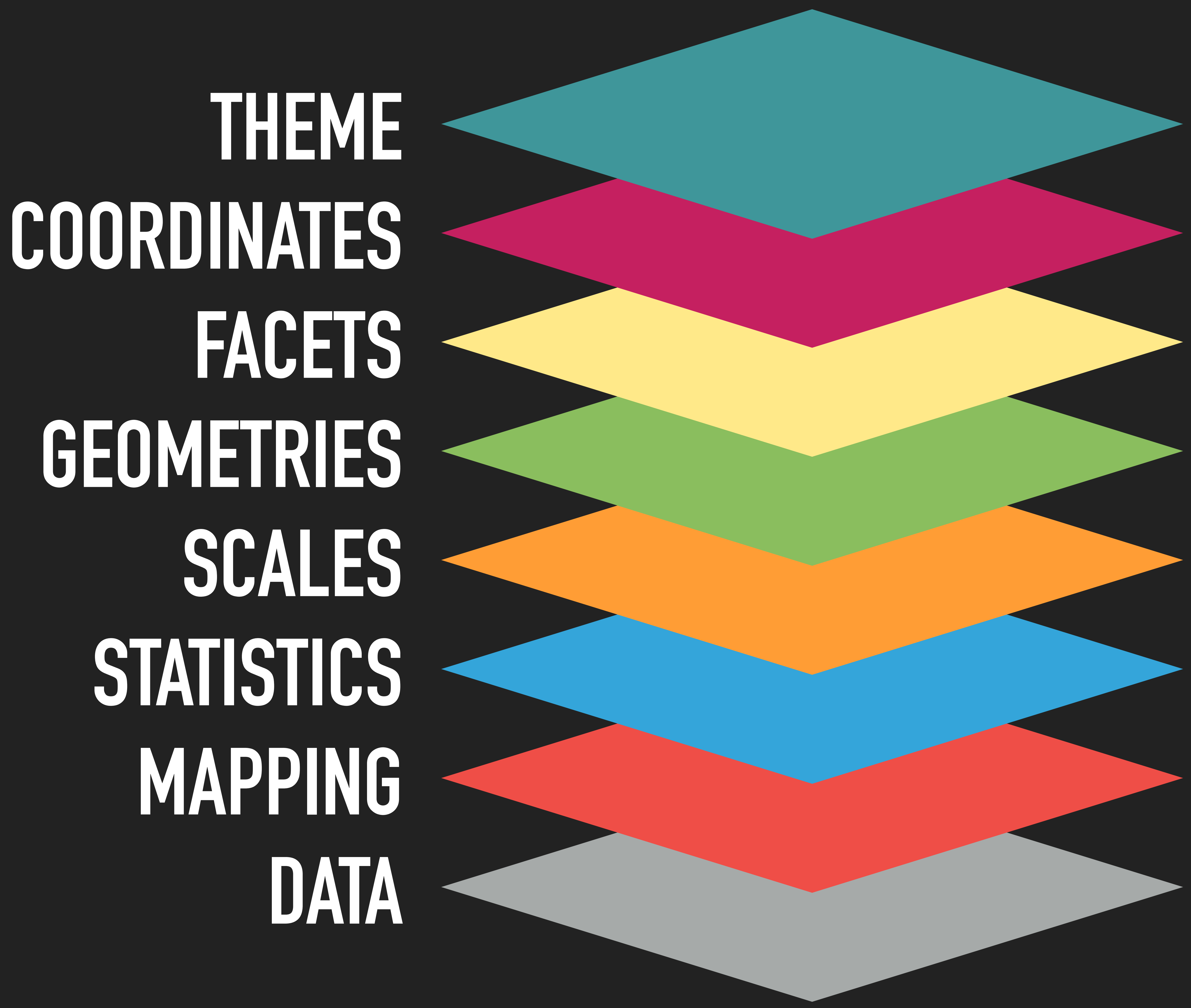
GRAPHICS

THE IDEA

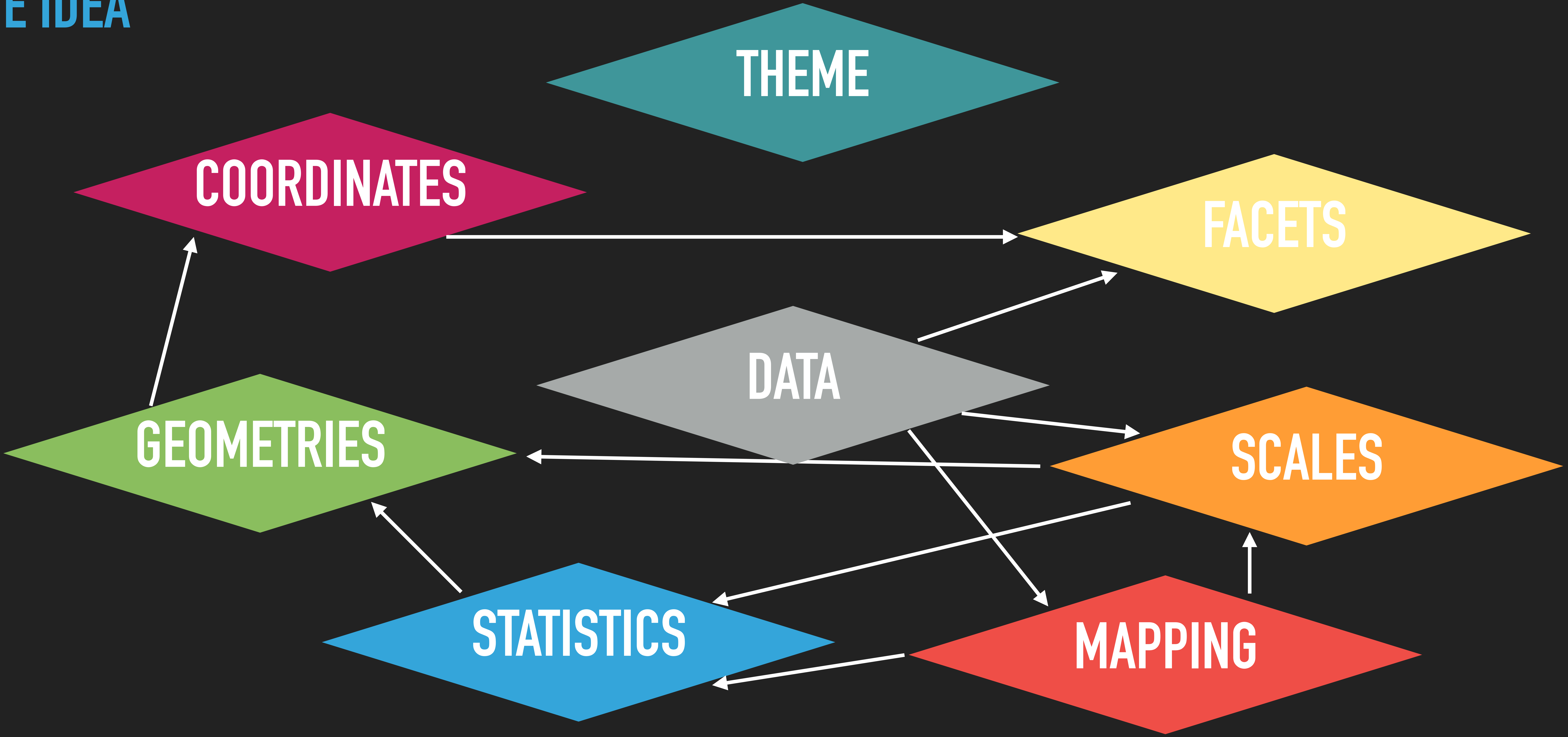
WHAT
IS
GRAPHICS?

THE IDEA

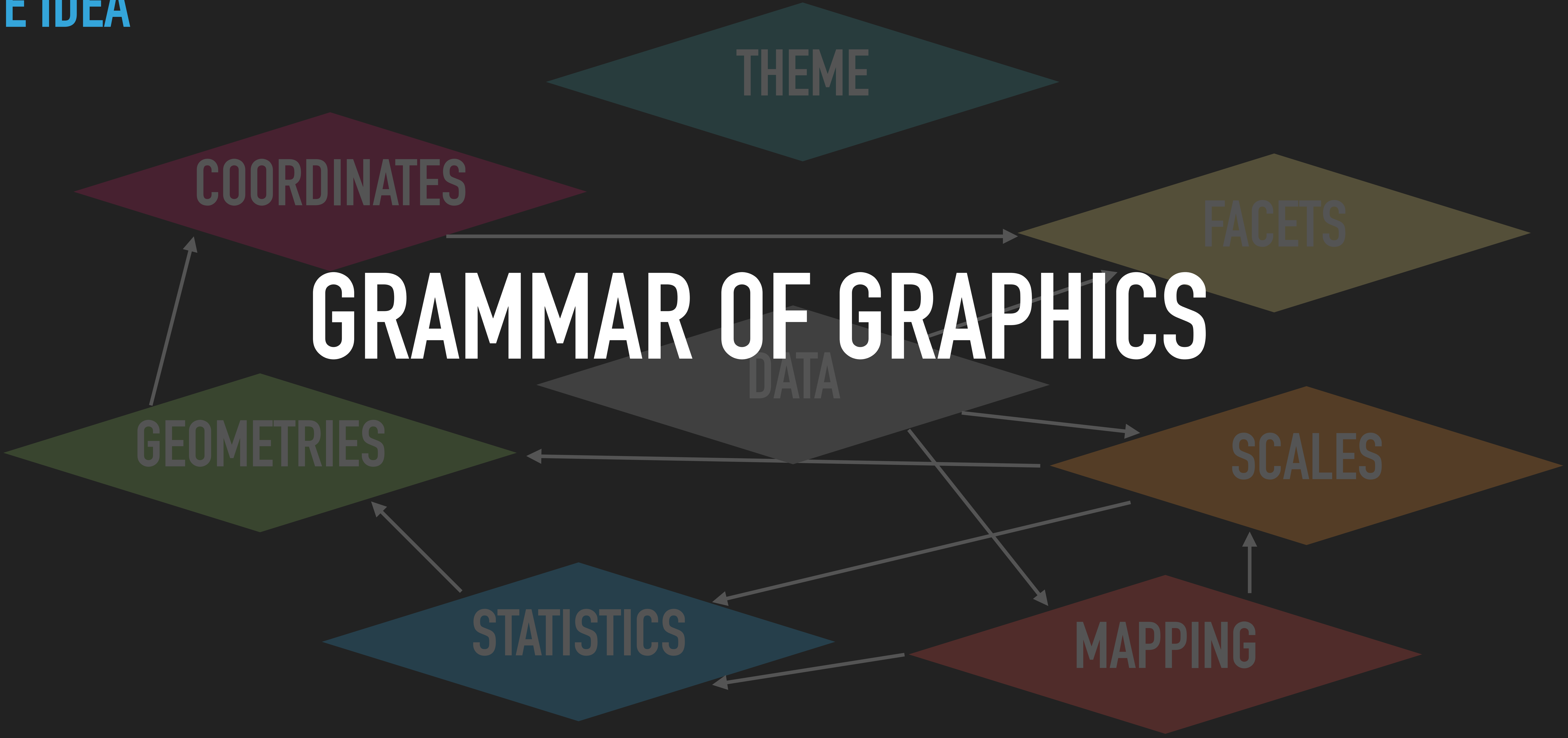
DECOMPOSE GRAPHICS INTO ITS CONSTITUENTS



THE IDEA

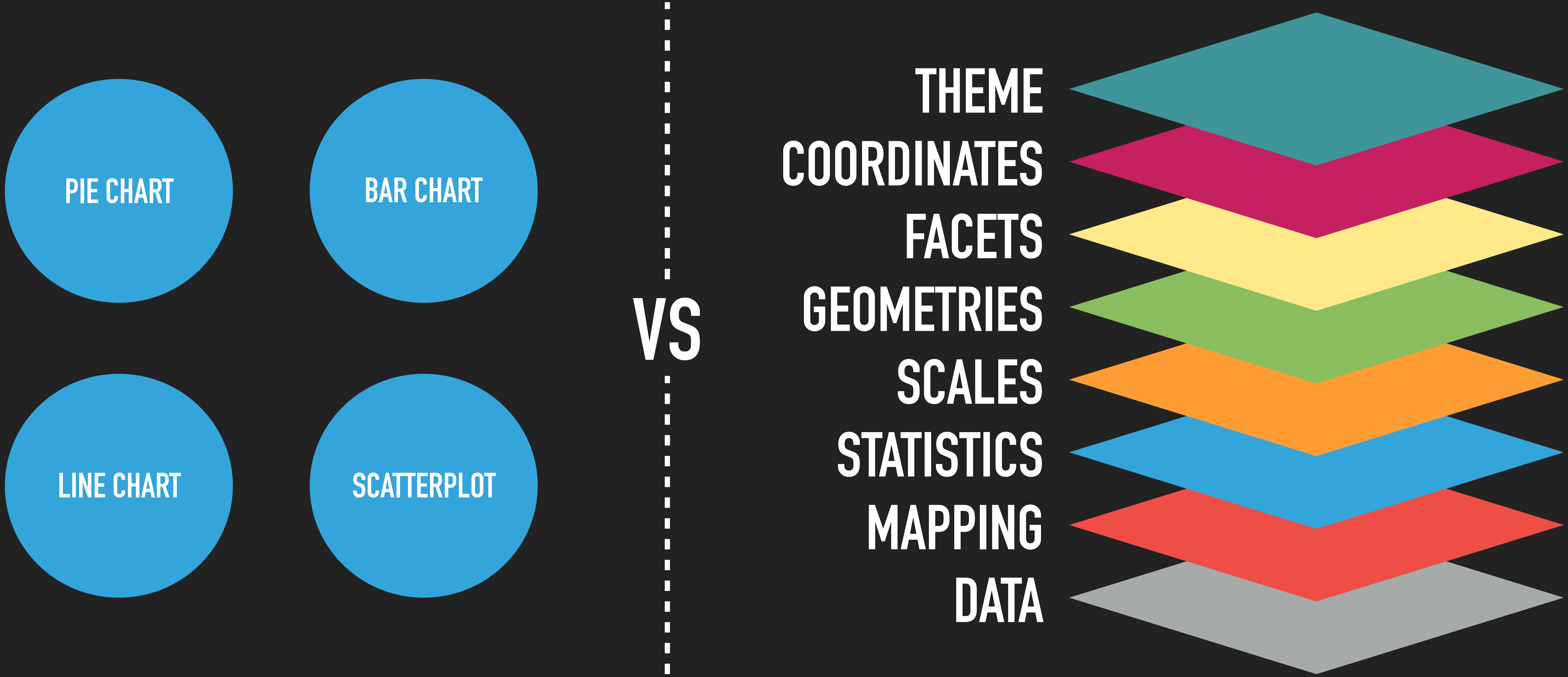

THEME
COORDINATES
FACETS
DATA
GEOMETRIES
SCALES
STATISTICS
MAPPING

THE IDEA



GRAMMAR OF GRAPHICS

THE IDEA



PIE CHART

BAR CHART

LINE CHART

SCATTERPLOT

VS

THEME

COORDINATES

FACETS

GEOMETRIES

SCALES

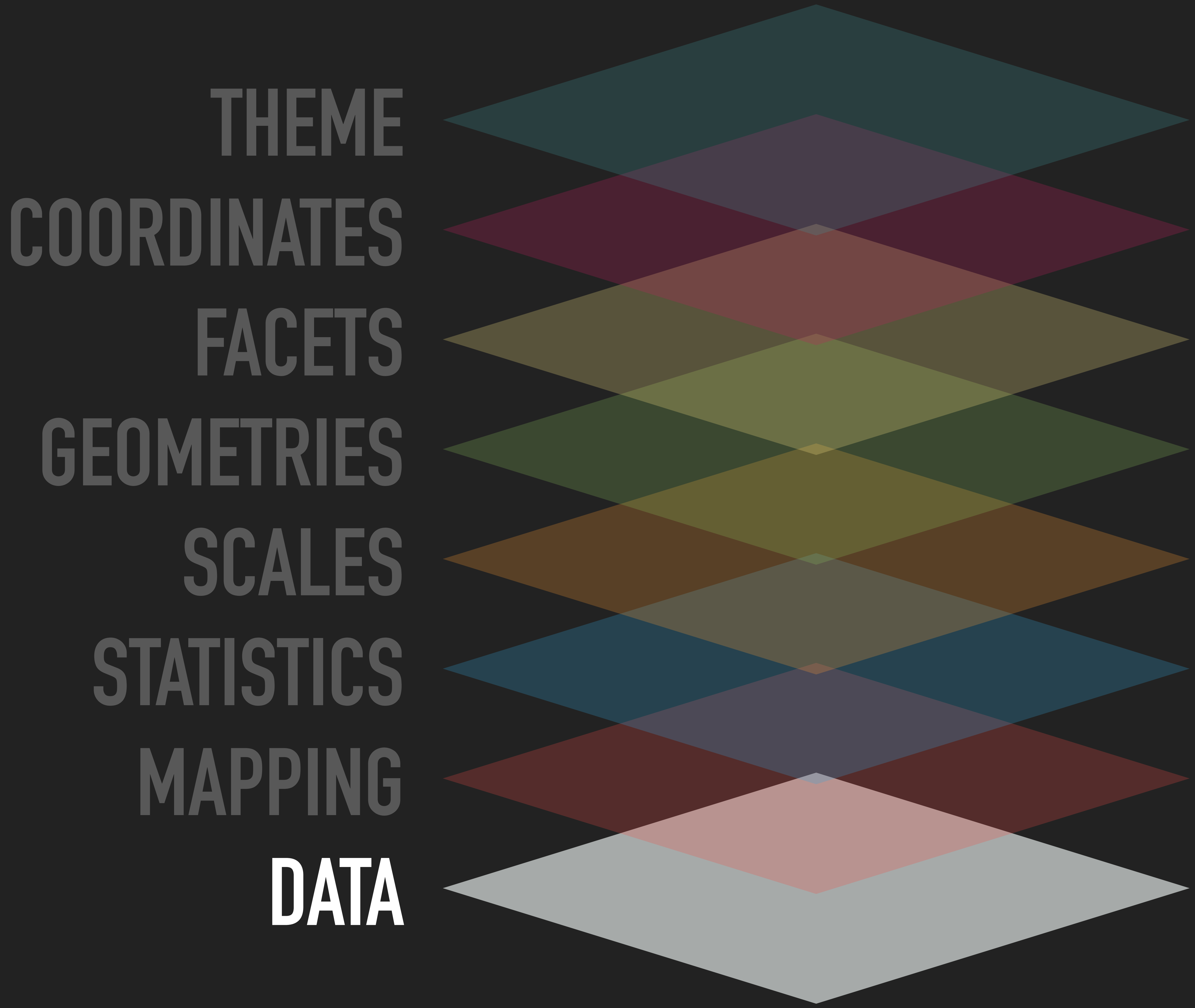
STATISTICS

MAPPING

DATA

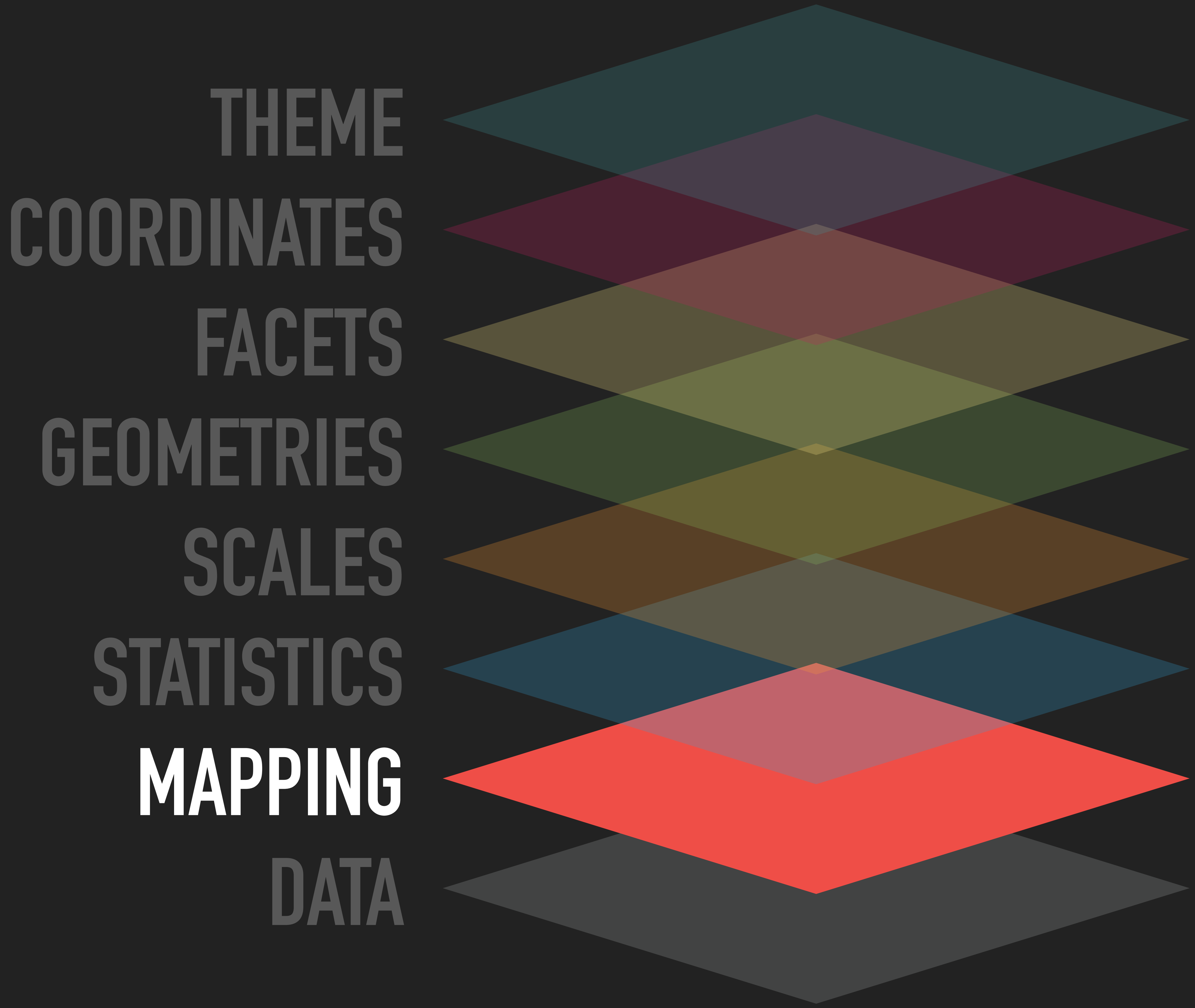
THE GRAMMAR

- Data is not just data
- Representation defines what can be done with it
- Grammar requires a tidy format (though it precedes the notion)
- We will not talk more about it but it is very important 😬



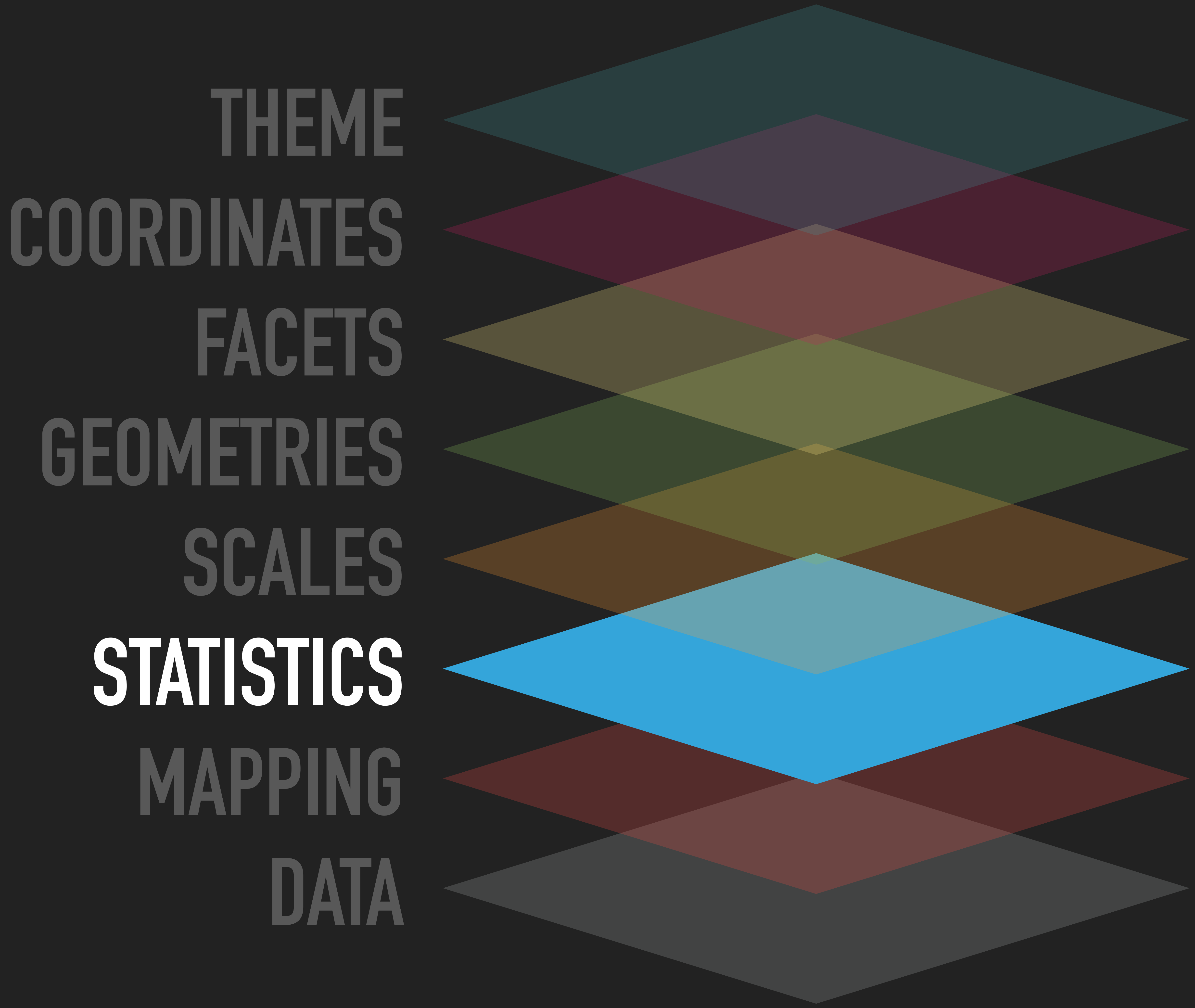
THE GRAMMAR

- Allow generic datasets to be understood by the graphic system.
- *Aesthetic* mapping: Link variables in data to graphical properties in the geometry.
- *Facet* mapping: Link variables in the data to panels in the facet layout.



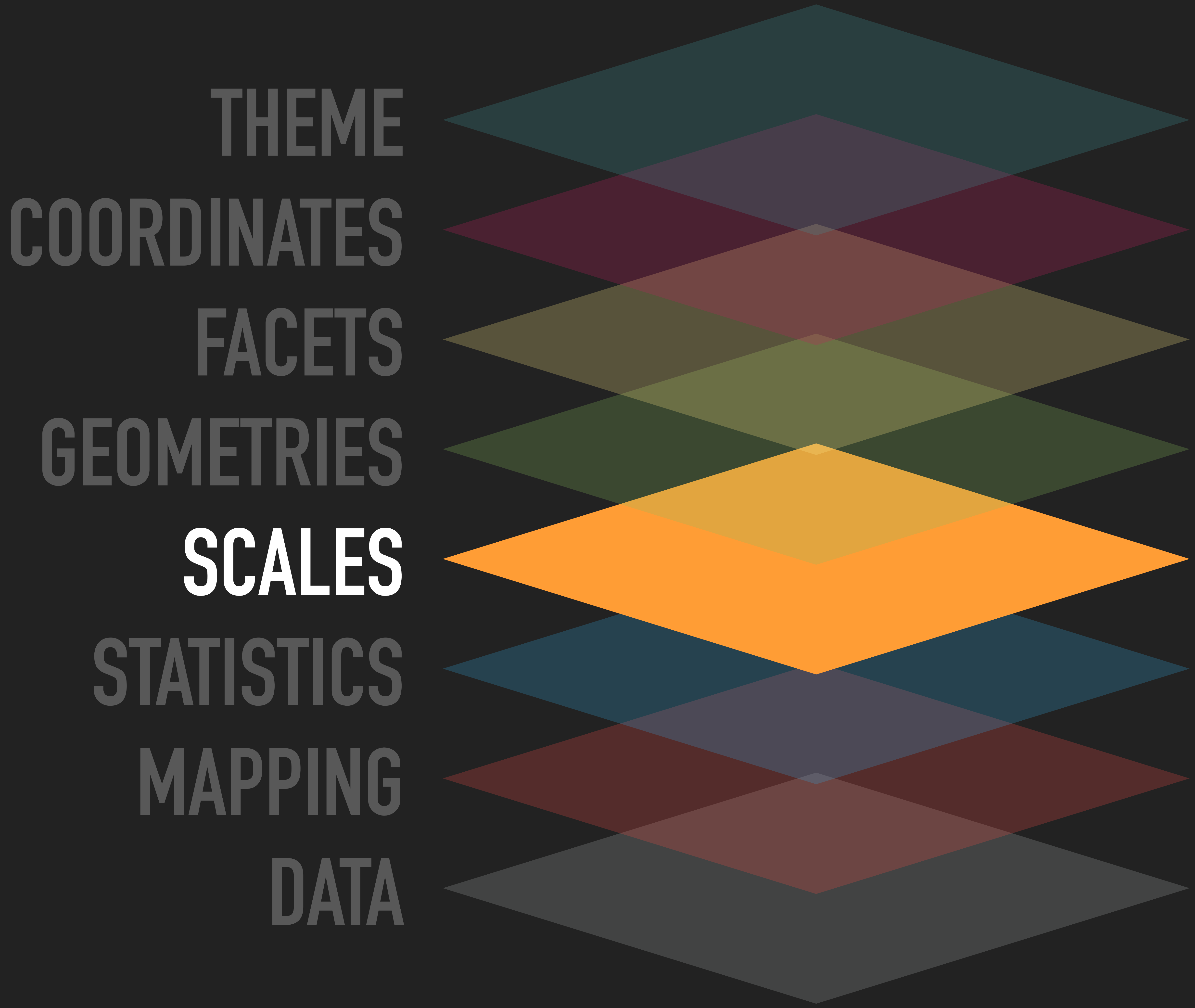
THE GRAMMAR

- Even though data is tidy it may not represent the displayed values
- Transform input variables to displayed values:
  - Count number of observations in each category for a bar chart
  - Calculate summary statistics for a boxplot.
- Is implicit in many plot-types but can often be done prior to plotting



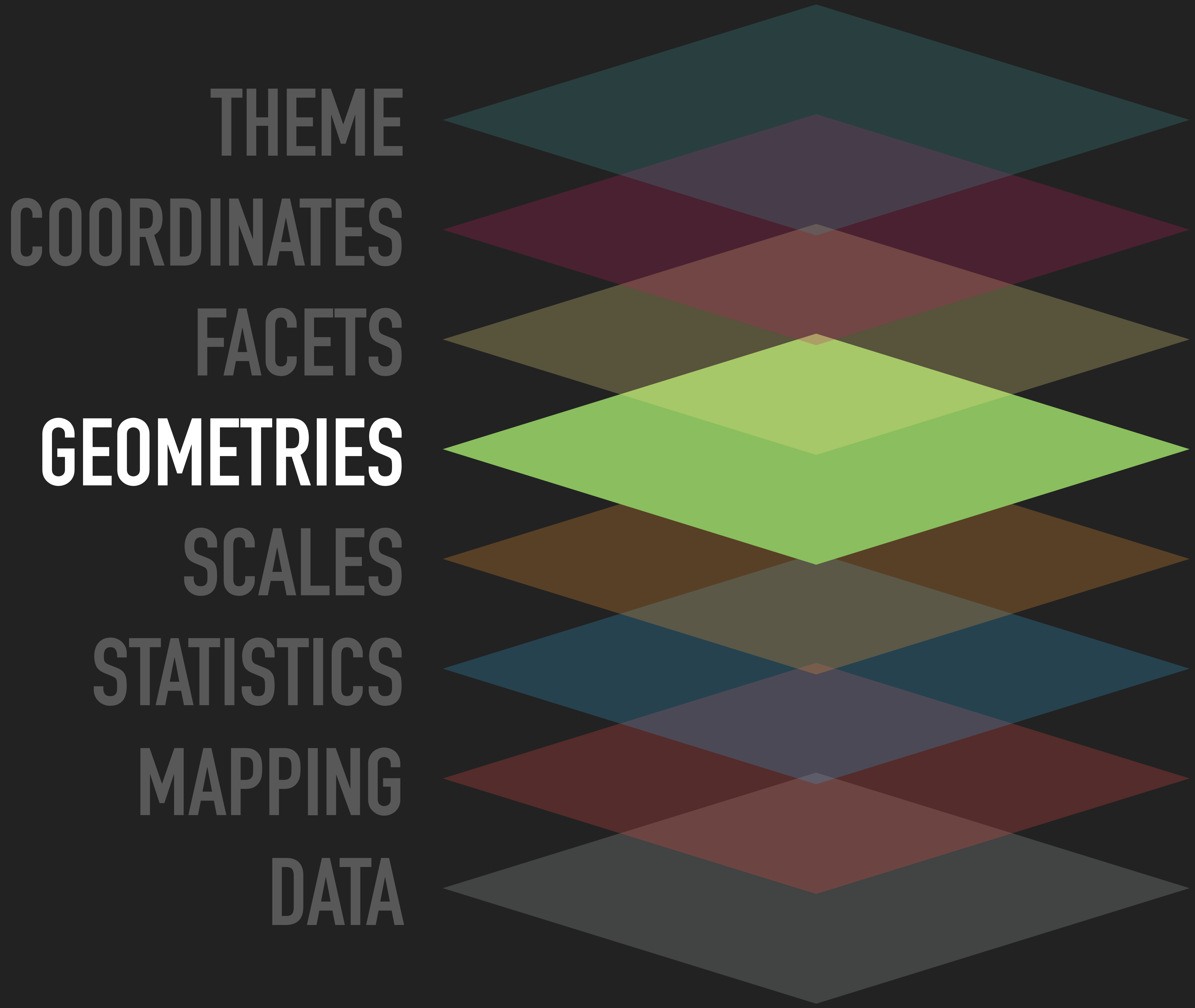
THE GRAMMAR

- Most data does not directly represent graphical properties.
- A scale translate back and forth between variable ranges and property ranges
  - Categories → Colour
  - Numbers → Position
  - …
- Imply a specific interpretation of values; discrete, continuous, etc.



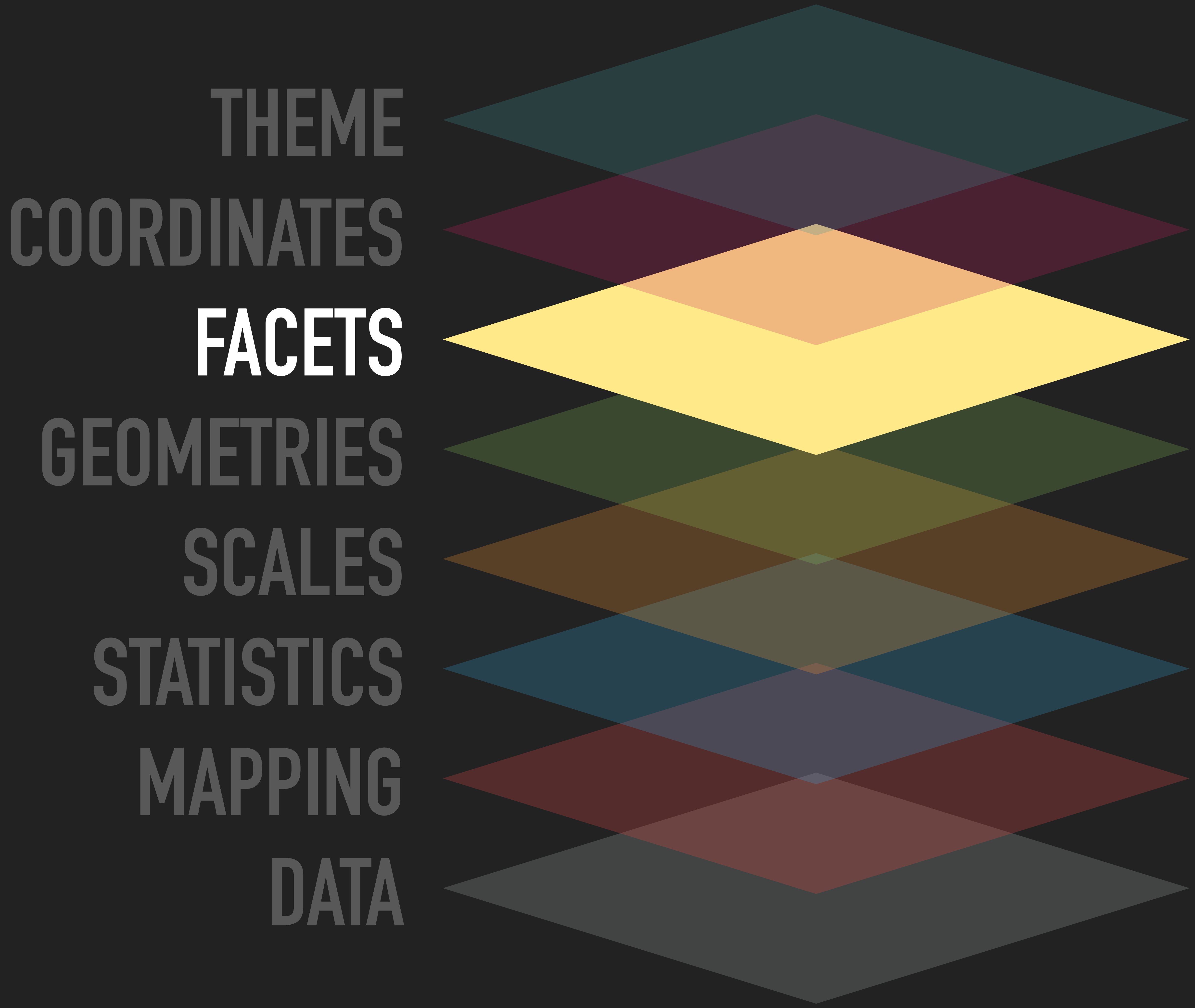
THE GRAMMAR

- How to interpret aesthetics as graphical representations
- Is a progression of positional aesthetics a number of points, a line, a single polygon, or something else entirely?
- To a high degree the determinator of your plot type



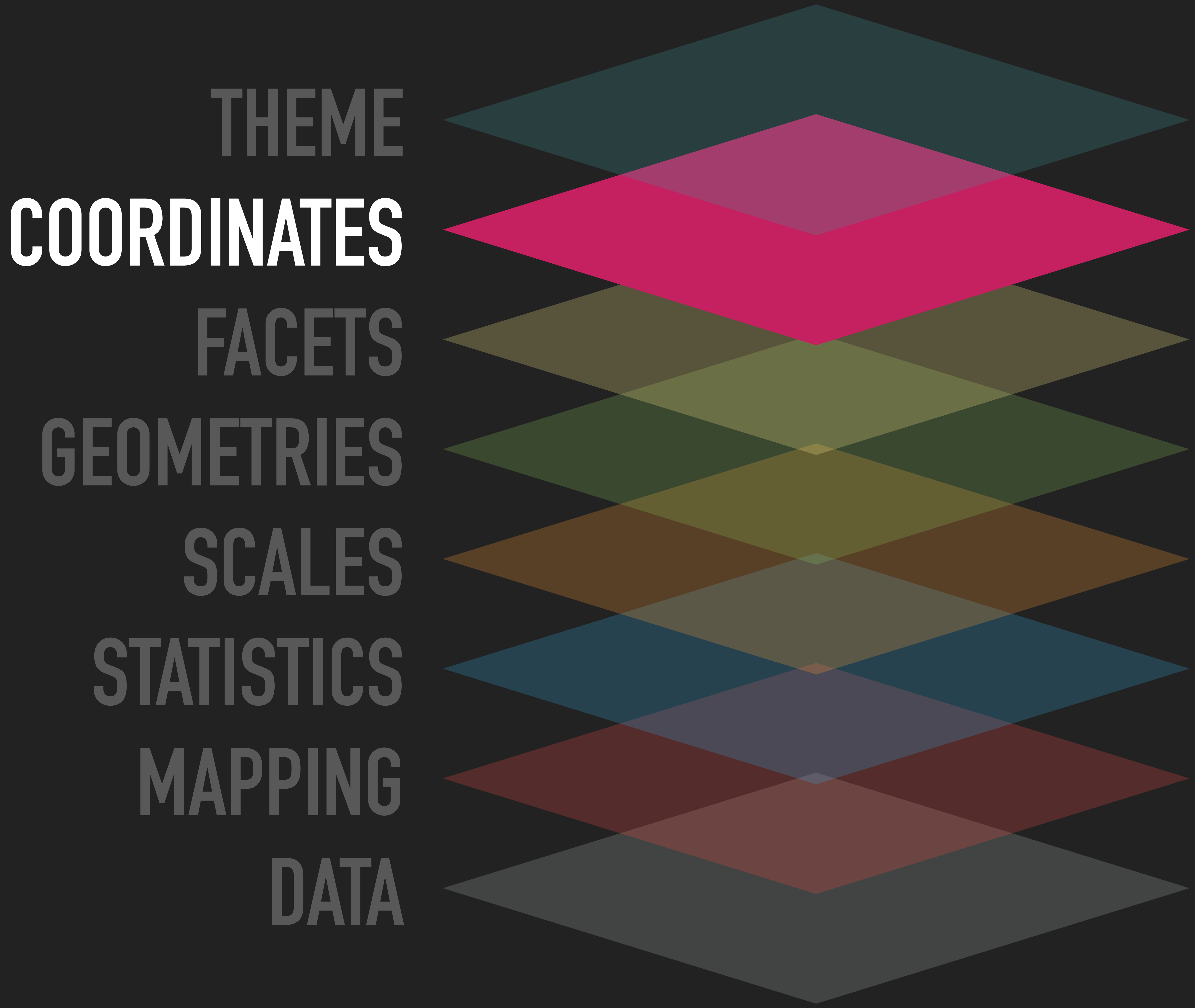
THE GRAMMAR

- Define the number of panels with equal logic and split data among them…
- Small multiples
  - Allows you to look at small subsets of your data in separate plots
- Panel layout may carry meaning



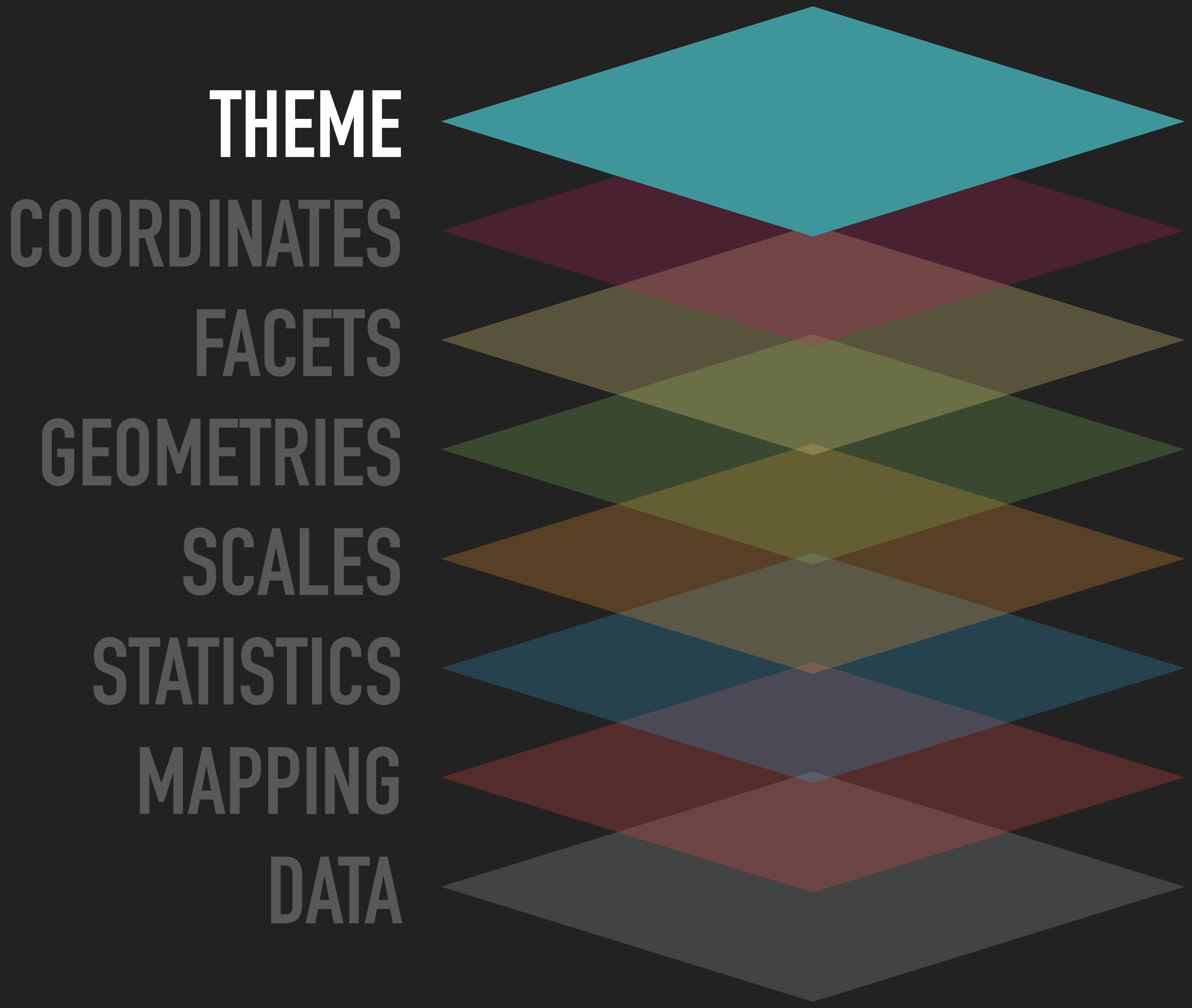
THE GRAMMAR

- Positional aesthetics are special.
  1. Variables are mapped, scaled, applied to a geometry
  2. But in the end, the position values are interpreted by a coordinate system
- Defines the physical mapping of the aesthetics to the paper
- Vaguely similar to colour profile mapping for colour aesthetics (though you don't have control over it)



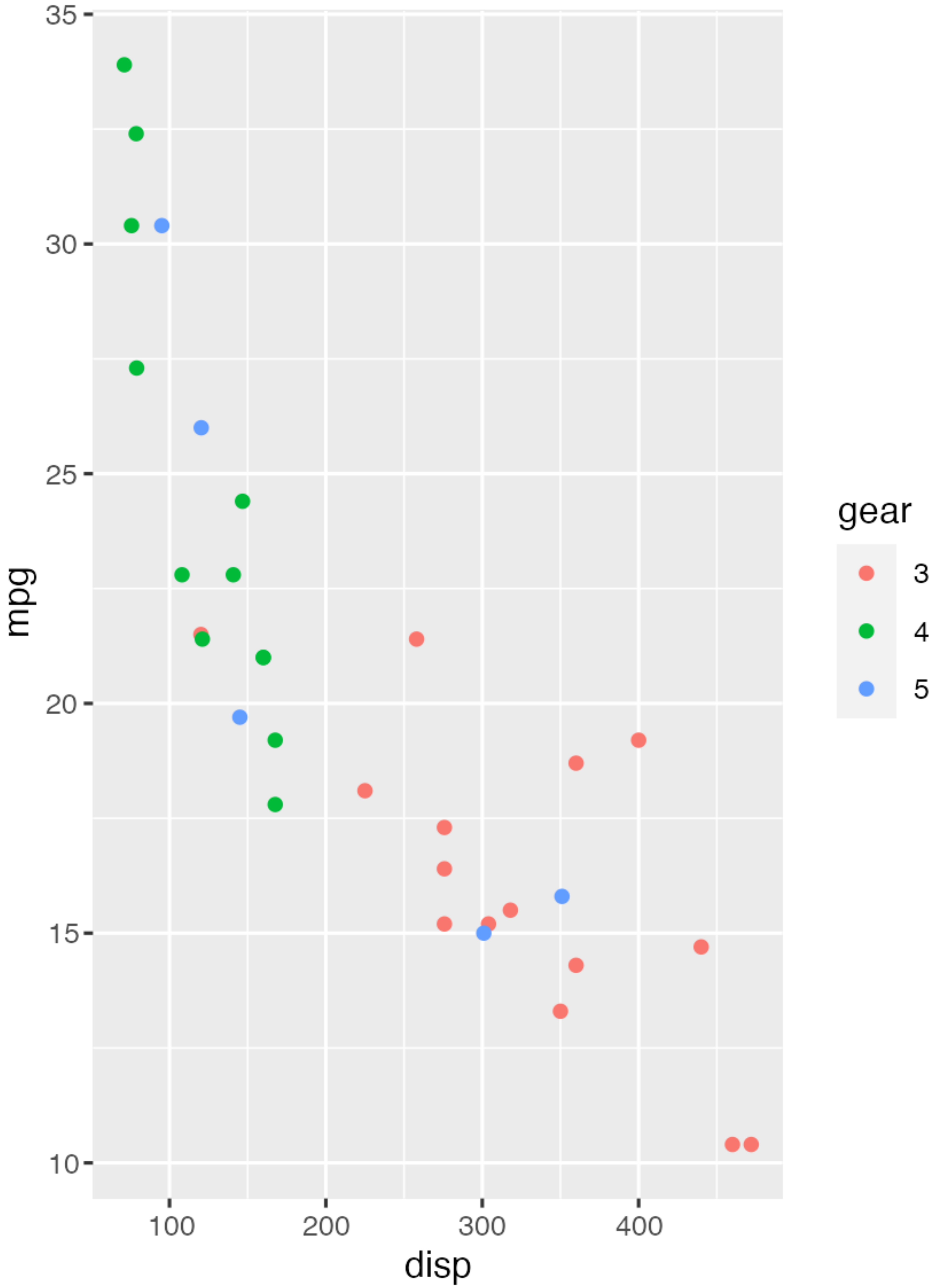
THE GRAMMAR

- None of the priors talked about the visual look of the plot.
- Theming spans every part of the graphic that is not linked to data



BUT I JUST WANTED TO MAKE A PIE CHART…

You

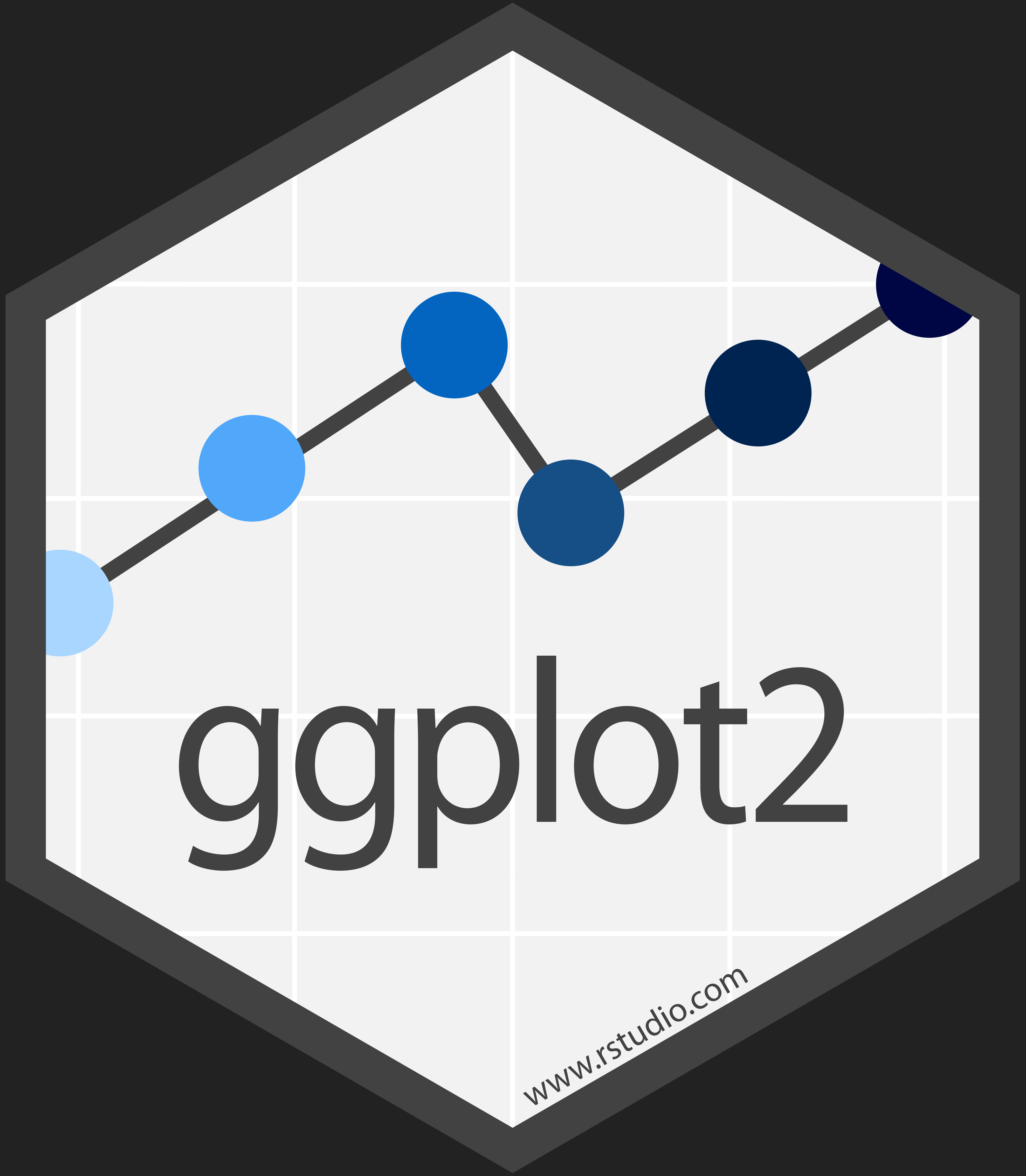


PART 2:

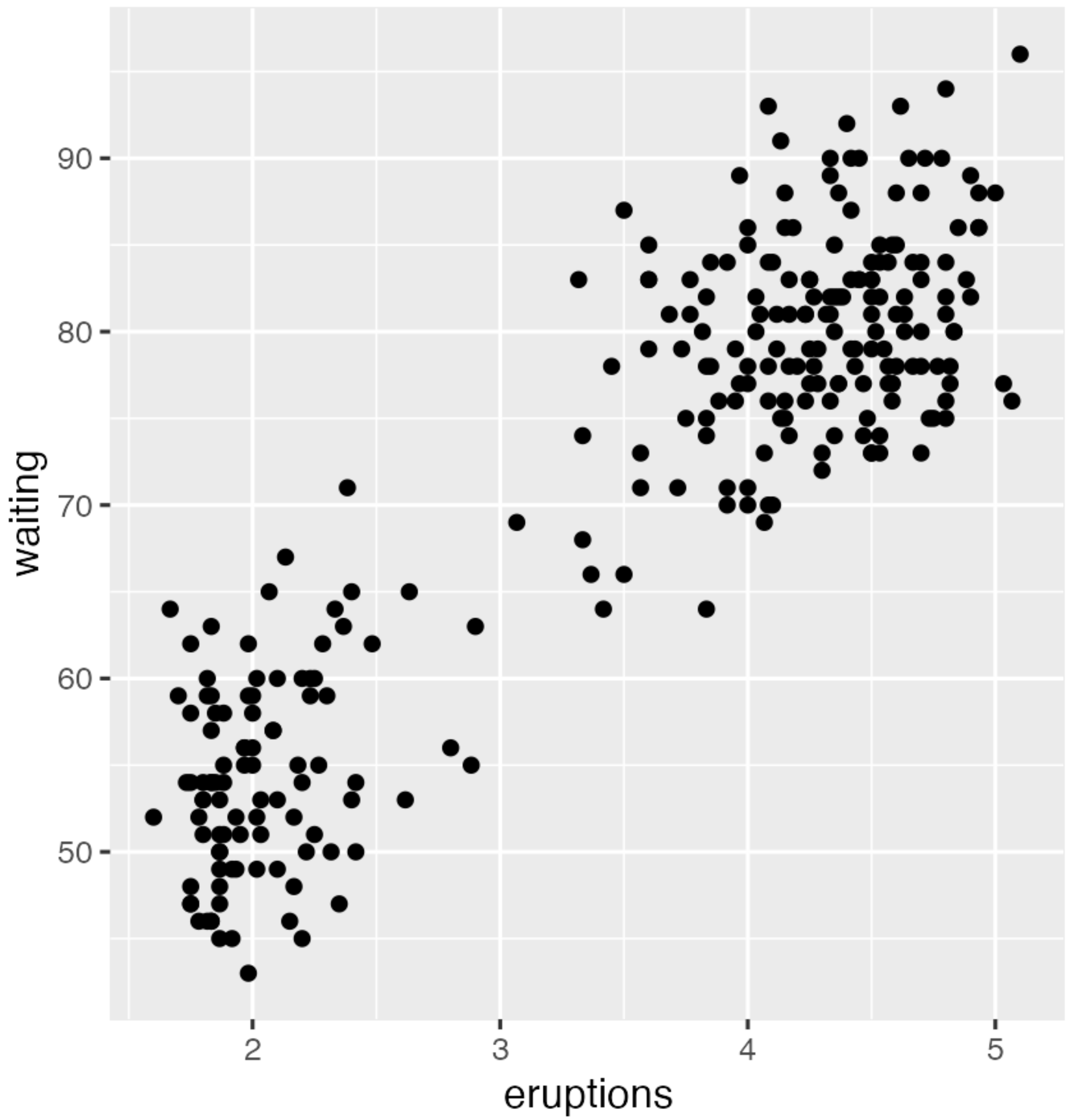
THE GGPLOT2 API

GGPLOT2

- Grammar of Graphics is a design system
- ggplot2 is an implementation
- Hadleys grammar != Lelands grammar
  - Sometimes real life gets in the way
- The spirit lives on
- Other implementations:
  - Tableau
  - Vega-Lite

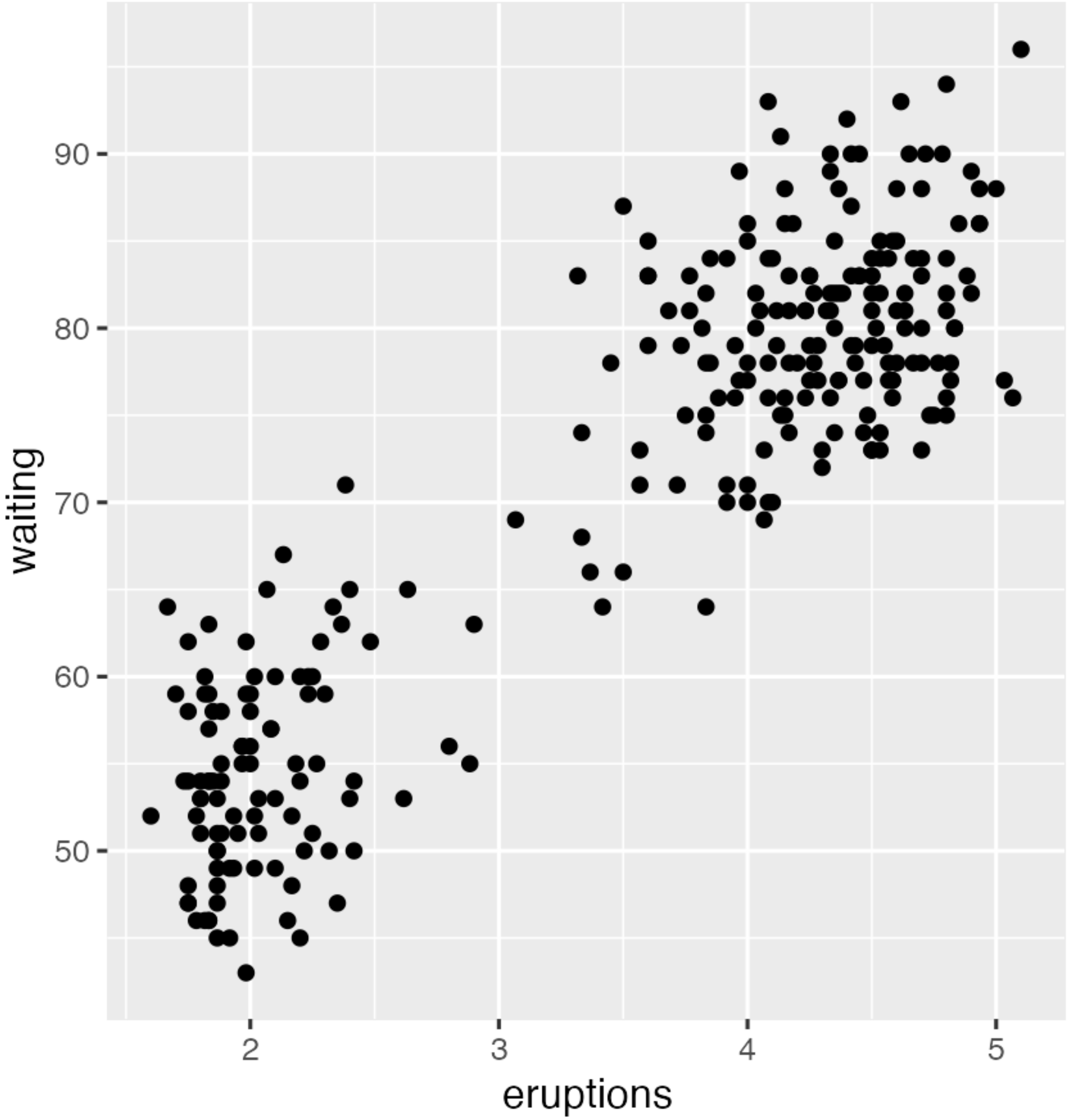


```
ggplot(data = faithful,
       mapping = aes(x = eruptions,
                     y = waiting)) +
  geom_point()
```



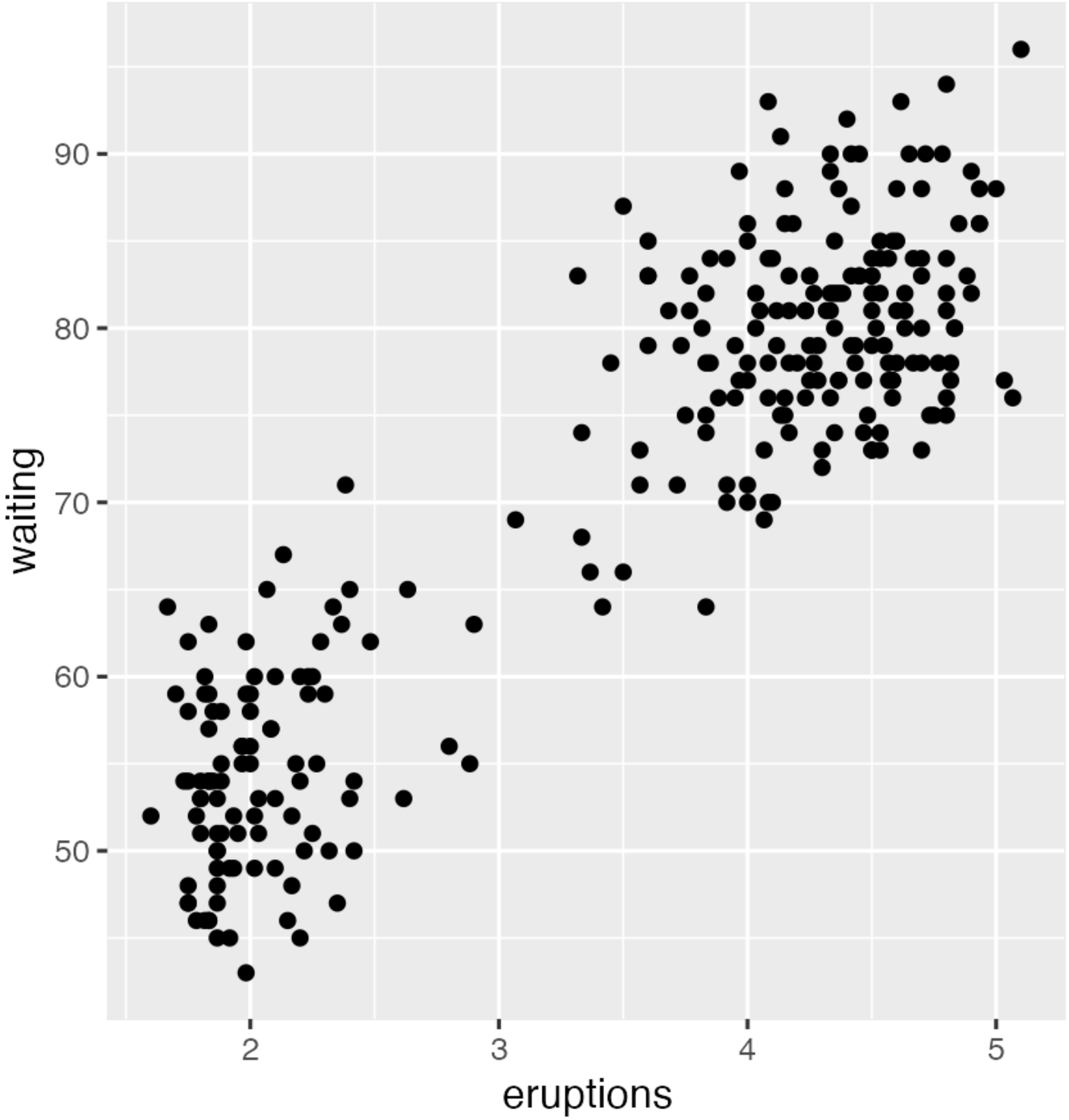
```
ggplot(data = faithful,
       mapping = aes(x = eruptions,
                     y = waiting)) +
  geom_point()
```

Which dataset to plot



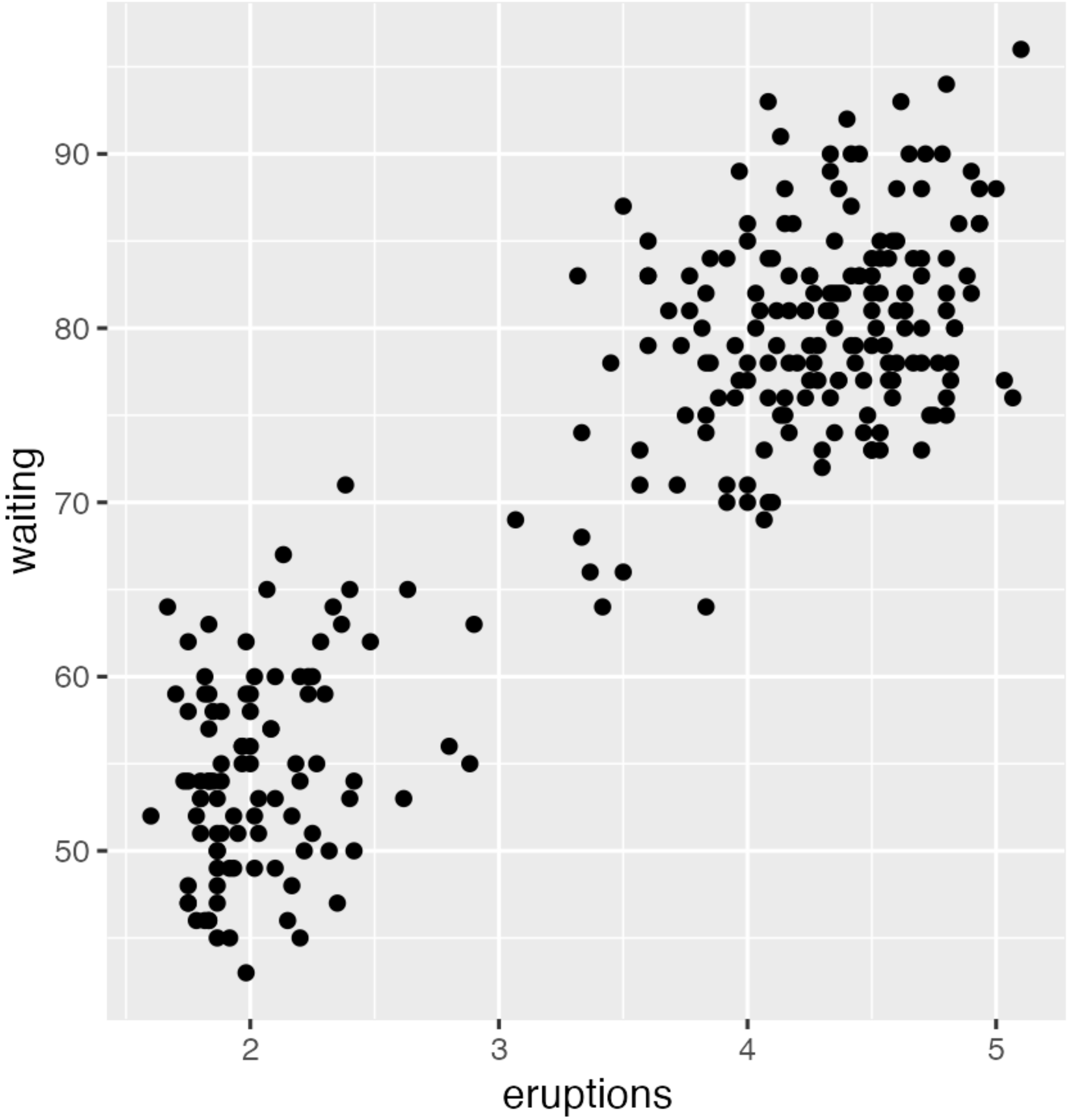
```
ggplot(data = faithful,
       mapping = aes(x = eruptions,
                     y = waiting)) +
  geom_point()
```

Which columns to use for x and y



```
ggplot(data = faithful,
       mapping = aes(x = eruptions,
                     y = waiting)) +
  geom_point()
```

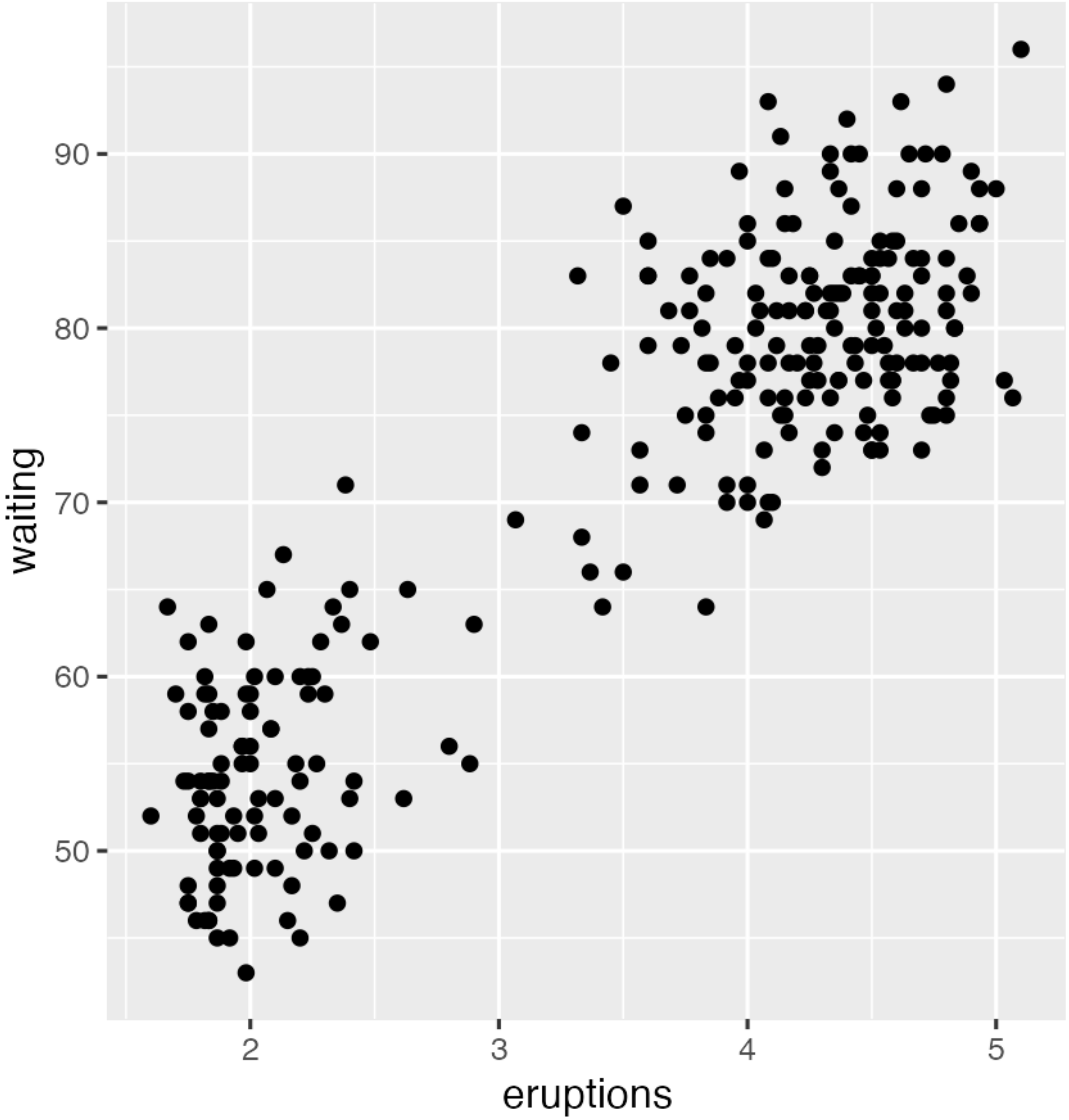
How to draw the plot



THE GGPLOT2 API

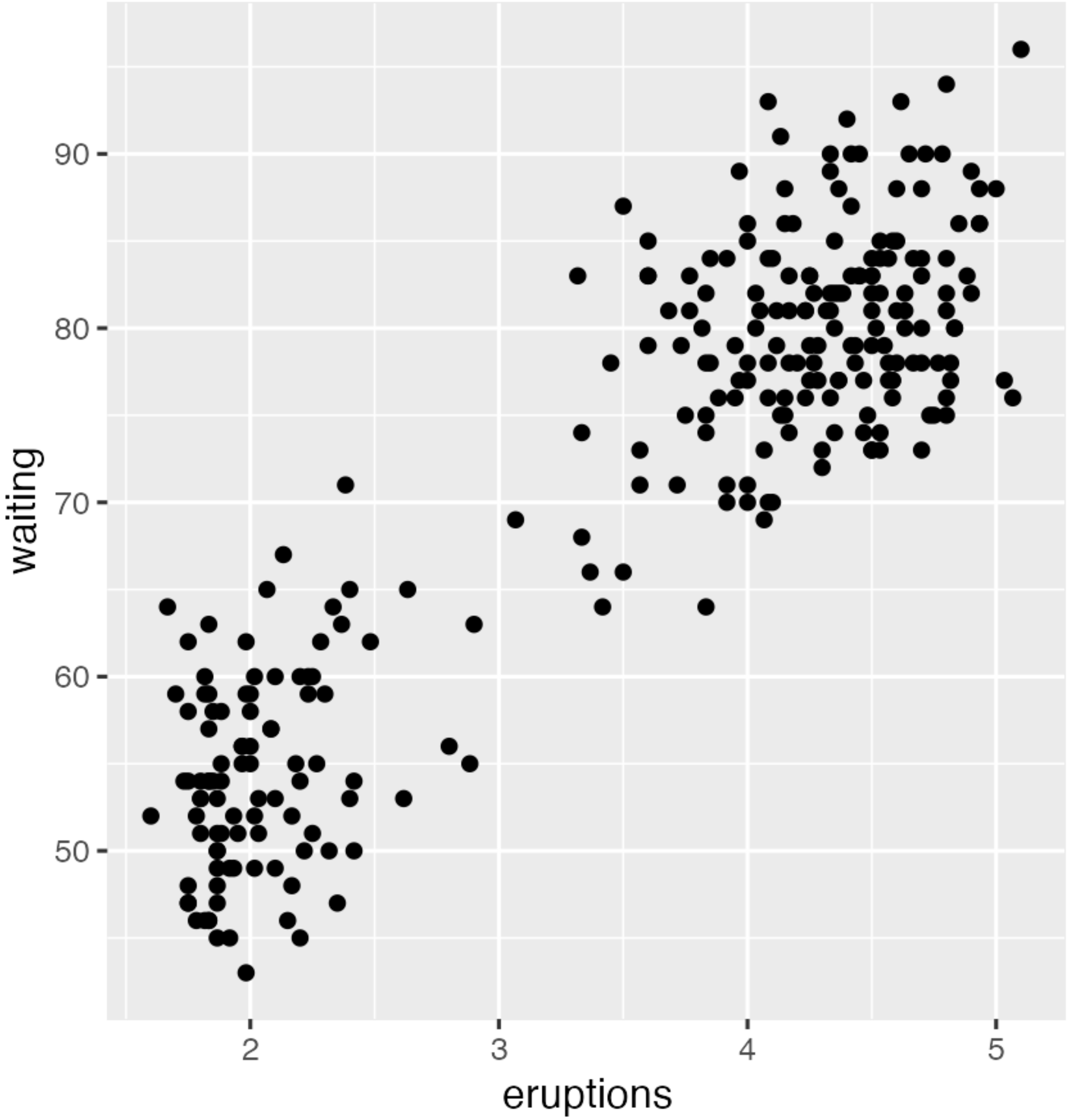
```
ggplot(data = faithful,
       mapping = aes(x = eruptions,
                     y = waiting)) +
  geom_point()
```

‘+’ is used to combine ggplot2 elements

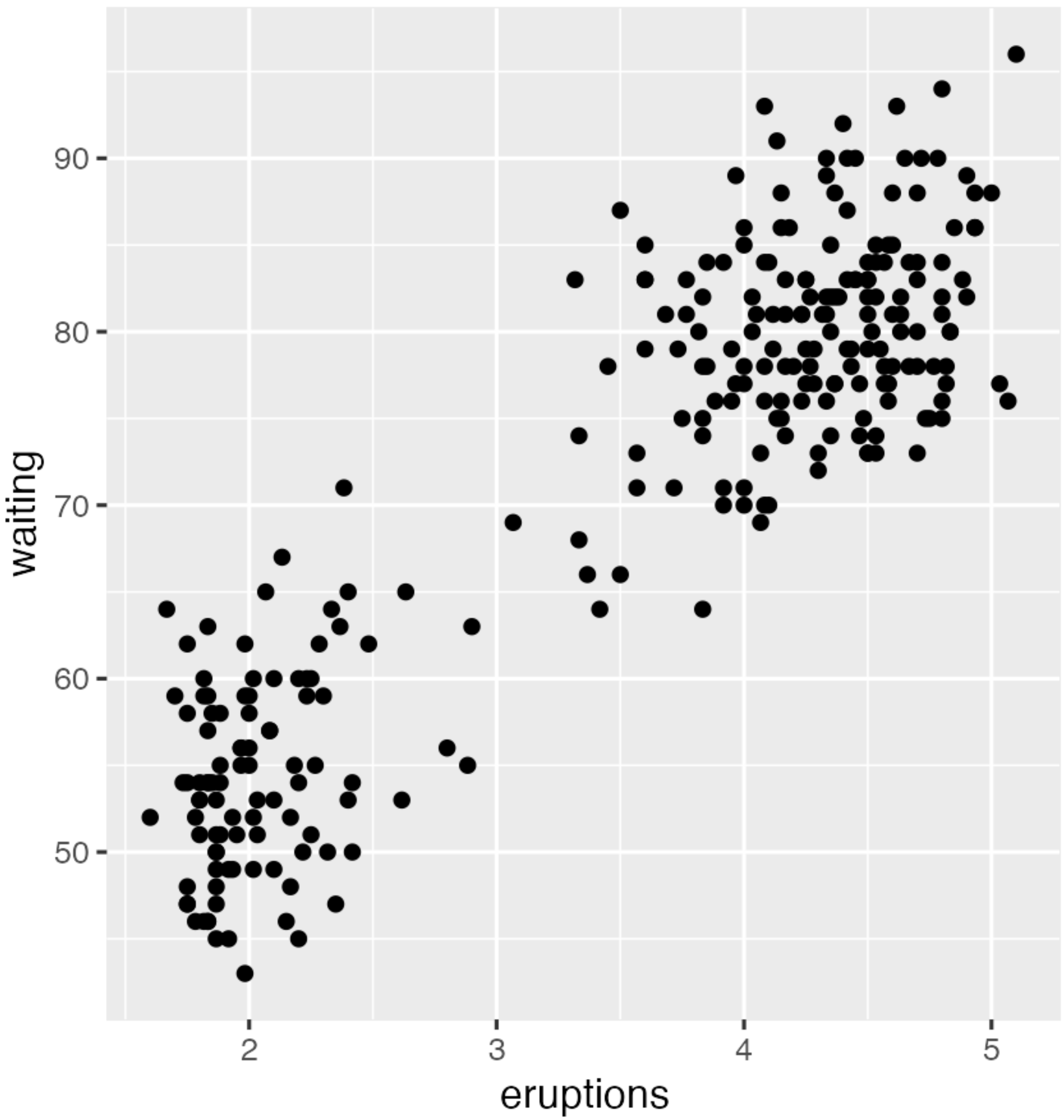


THE GGPLOT2 API

```
ggplot(data = faithful) +
  geom_point(mapping = aes(x = eruptions,
                           y = waiting))
```

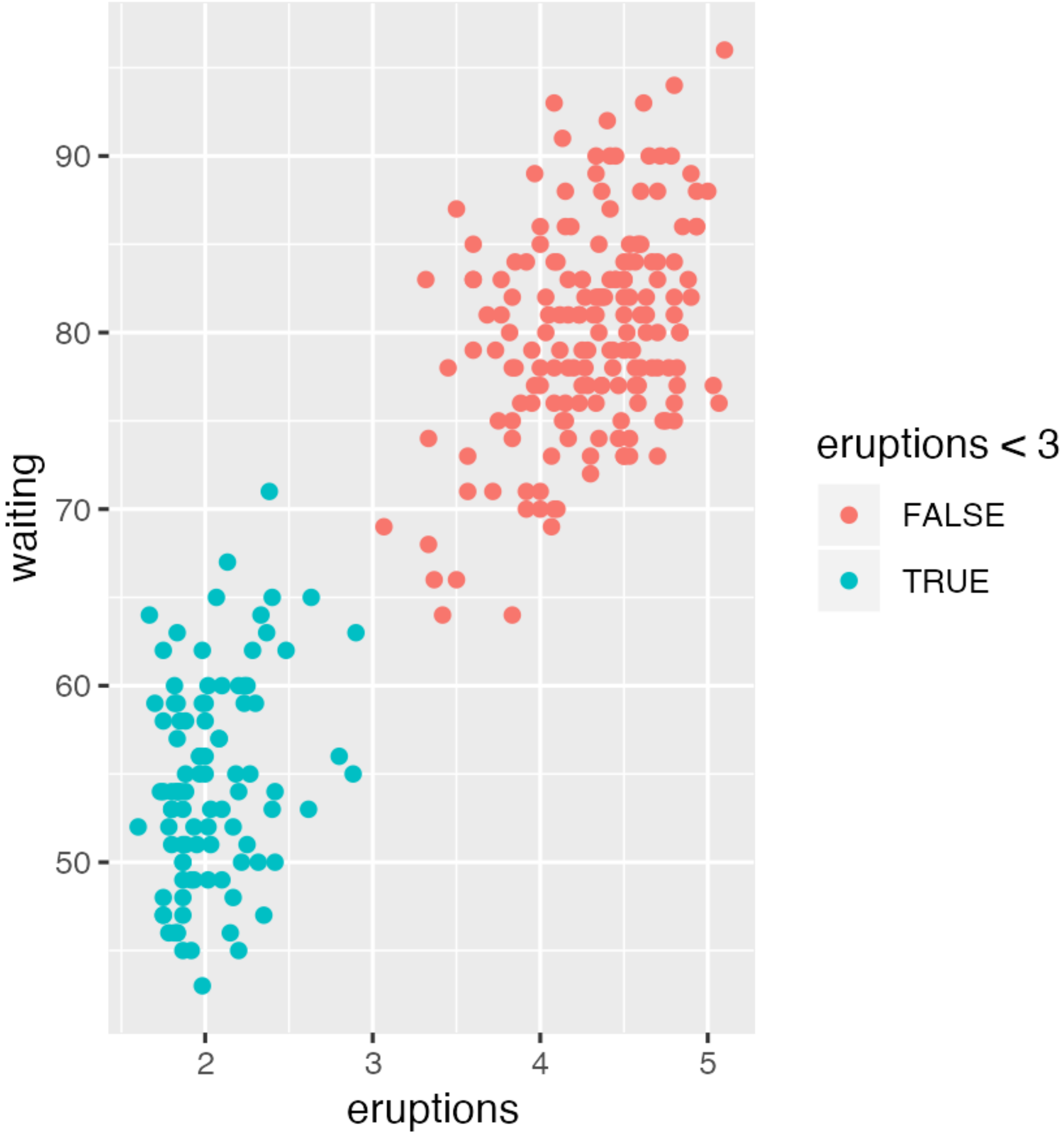


```
ggplot() +
  geom_point(mapping = aes(x = eruptions,
                           y = waiting),
             data = faithful)
```



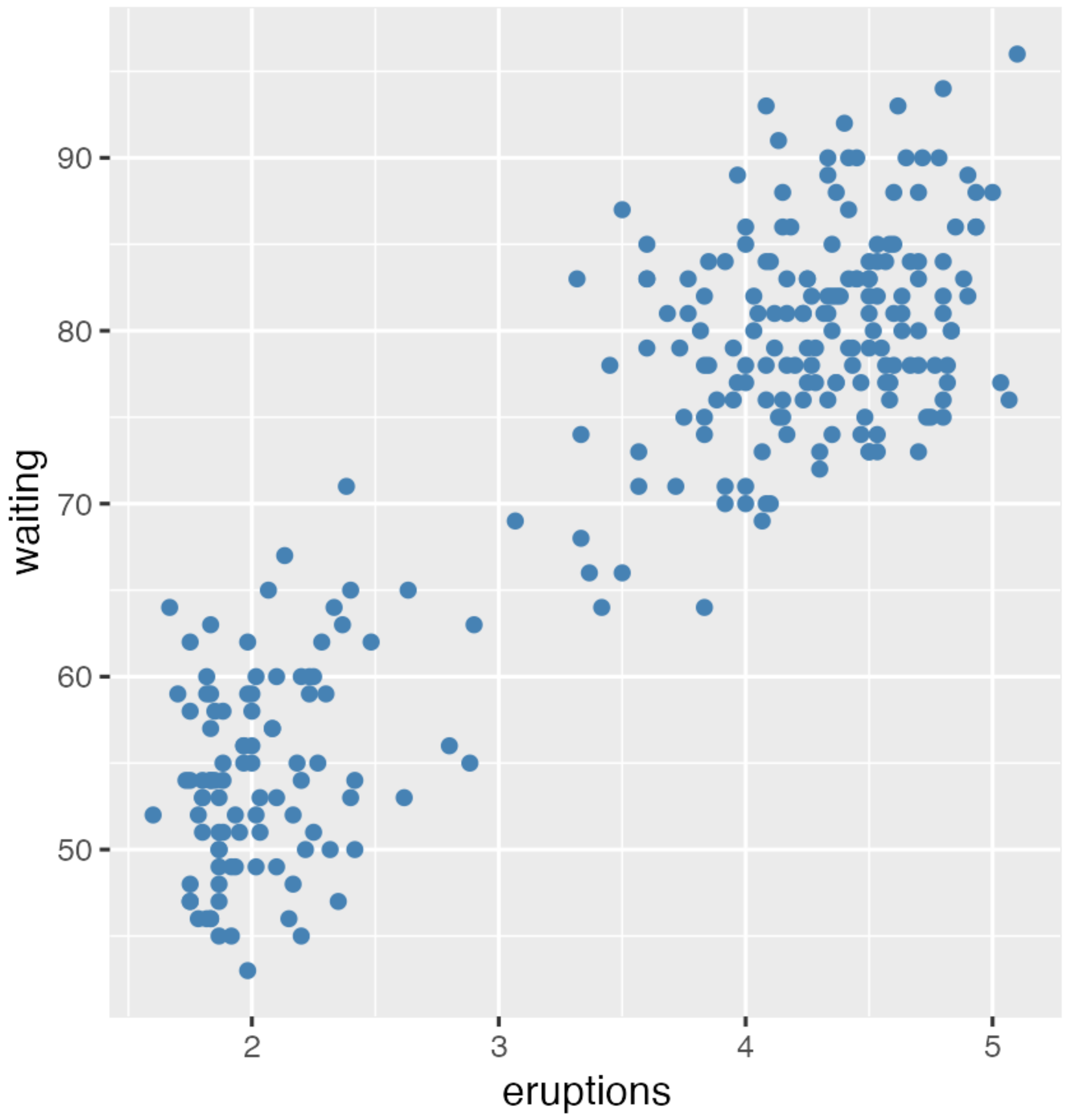
```
ggplot(faithful) +
  geom_point(aes(x = eruptions,
                 y = waiting,
                 colour = eruptions < 3))
```

Mapping colour

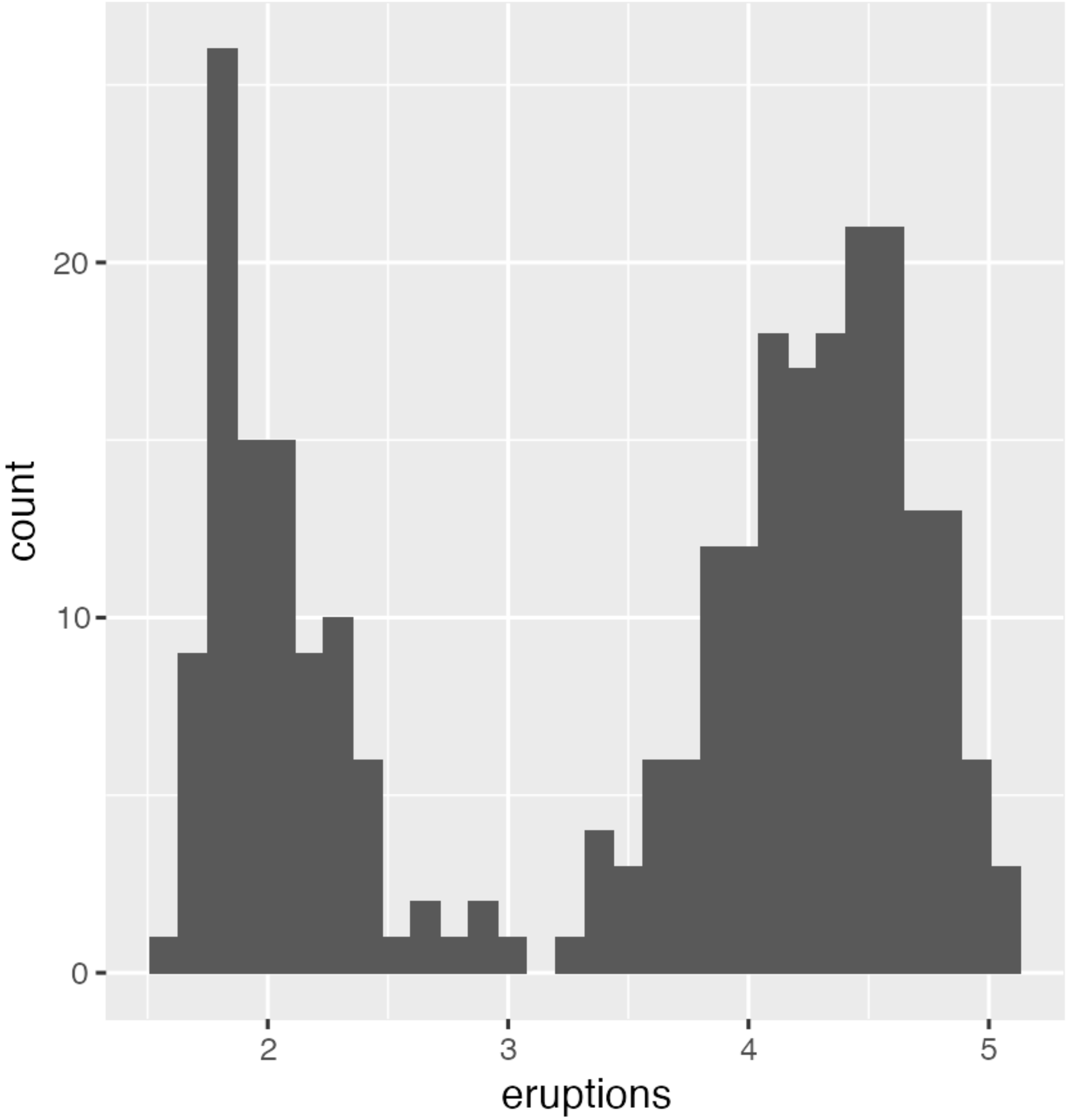


```
ggplot(faithful) +
  geom_point(aes(x = eruptions,
                 y = waiting),
             colour = 'steelblue')
```

Setting colour



```
ggplot(faithful) +
  geom_histogram(aes(x = eruptions))
```



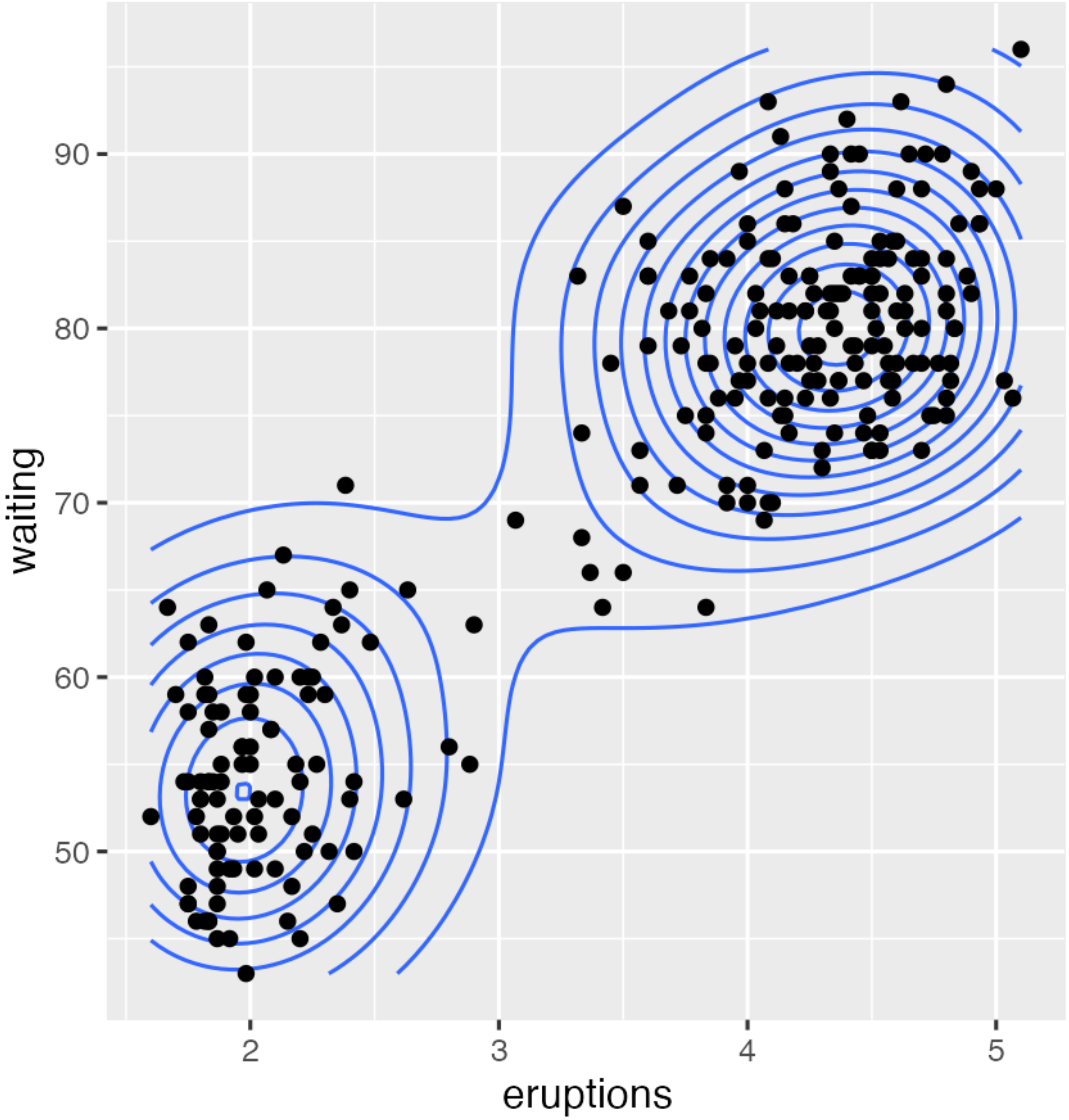
There are many types of geoms

Their mapping requirements differ

THE GGPLOT2 API

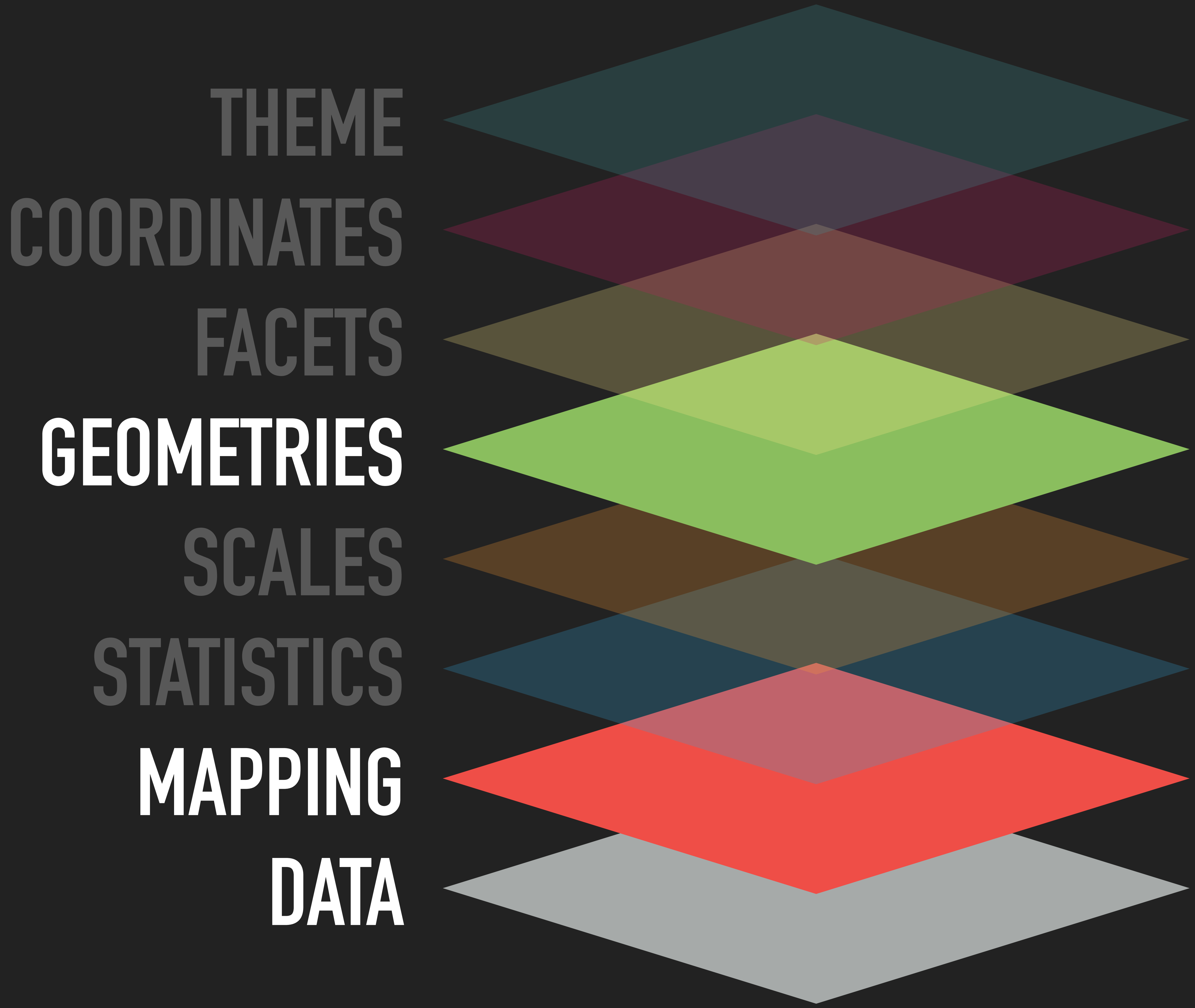
```
ggplot(faithful,
       aes(x = eruptions, y = waiting)) +
  geom_density_2d() +
  geom_point()
```

Layers are stacked in the order of code appearance



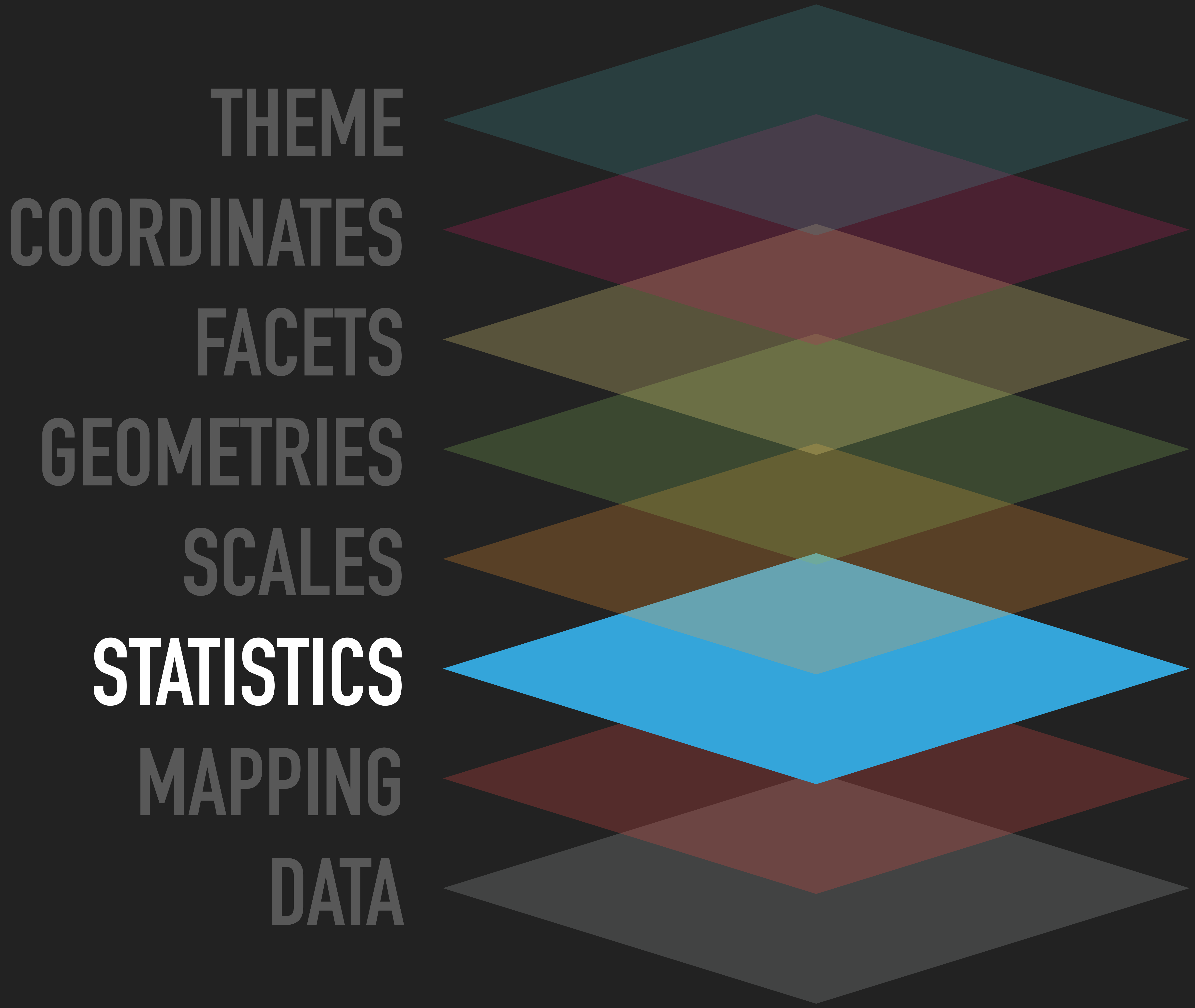
WHAT DID WE NEED?

EVERYTHING ELSE HAS SENSIBLE DEFAULTS



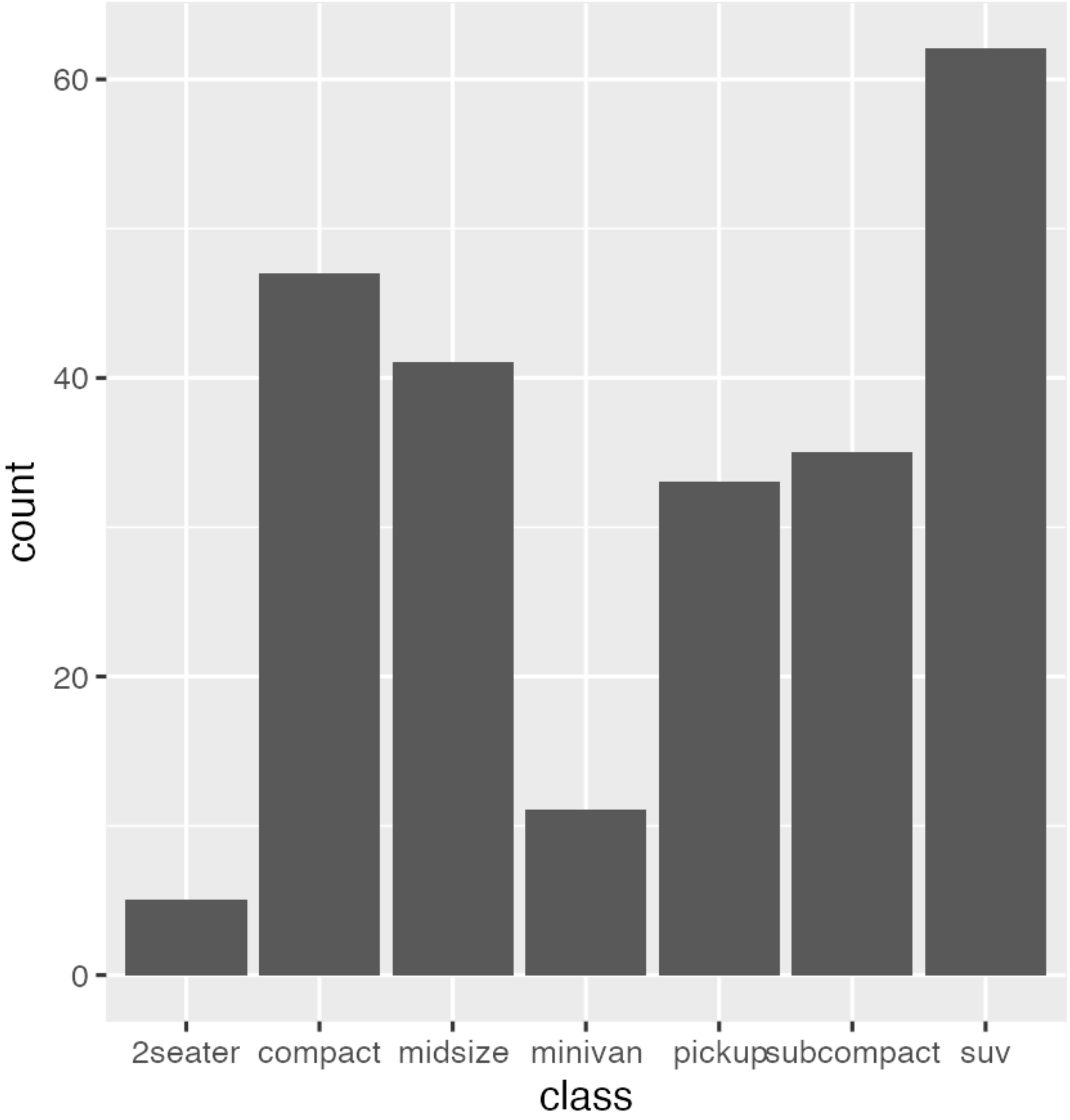
STATISTICS

- Linked to geometries
- Every geom has a default stat(istic)
- A layer can be created with a call to stat_*() or geom_*(), but community has coalesced around geom_*()



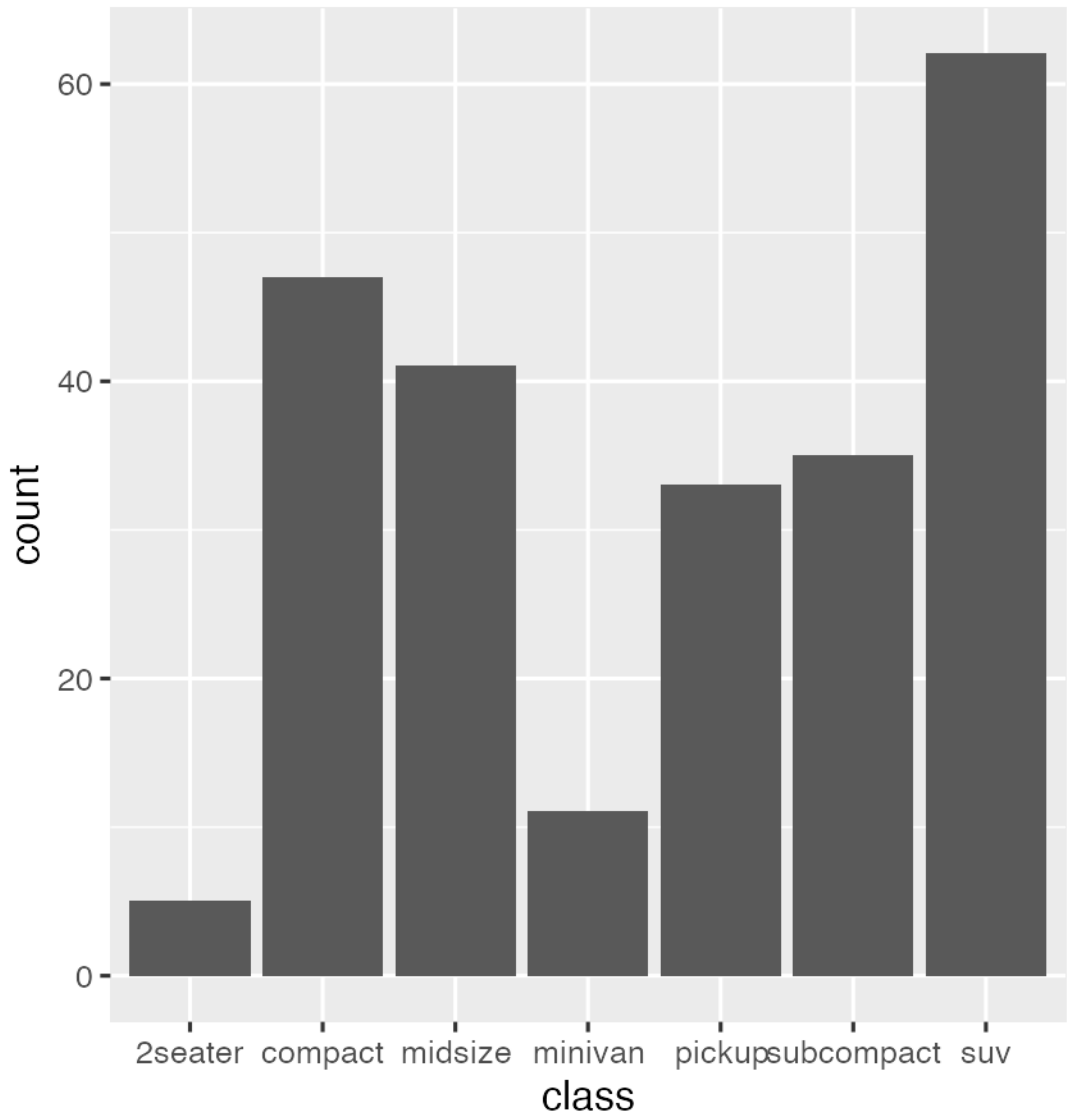
```
ggplot(mpg) +
  geom_bar(aes(x = class))
```

geom_bar() uses stat_count() by default



```
mpg_counted ← mpg %>%
  count(class, name = 'count')

ggplot(mpg_counted) +
  geom_bar(aes(x = class, y = count),
           stat = 'identity')
```

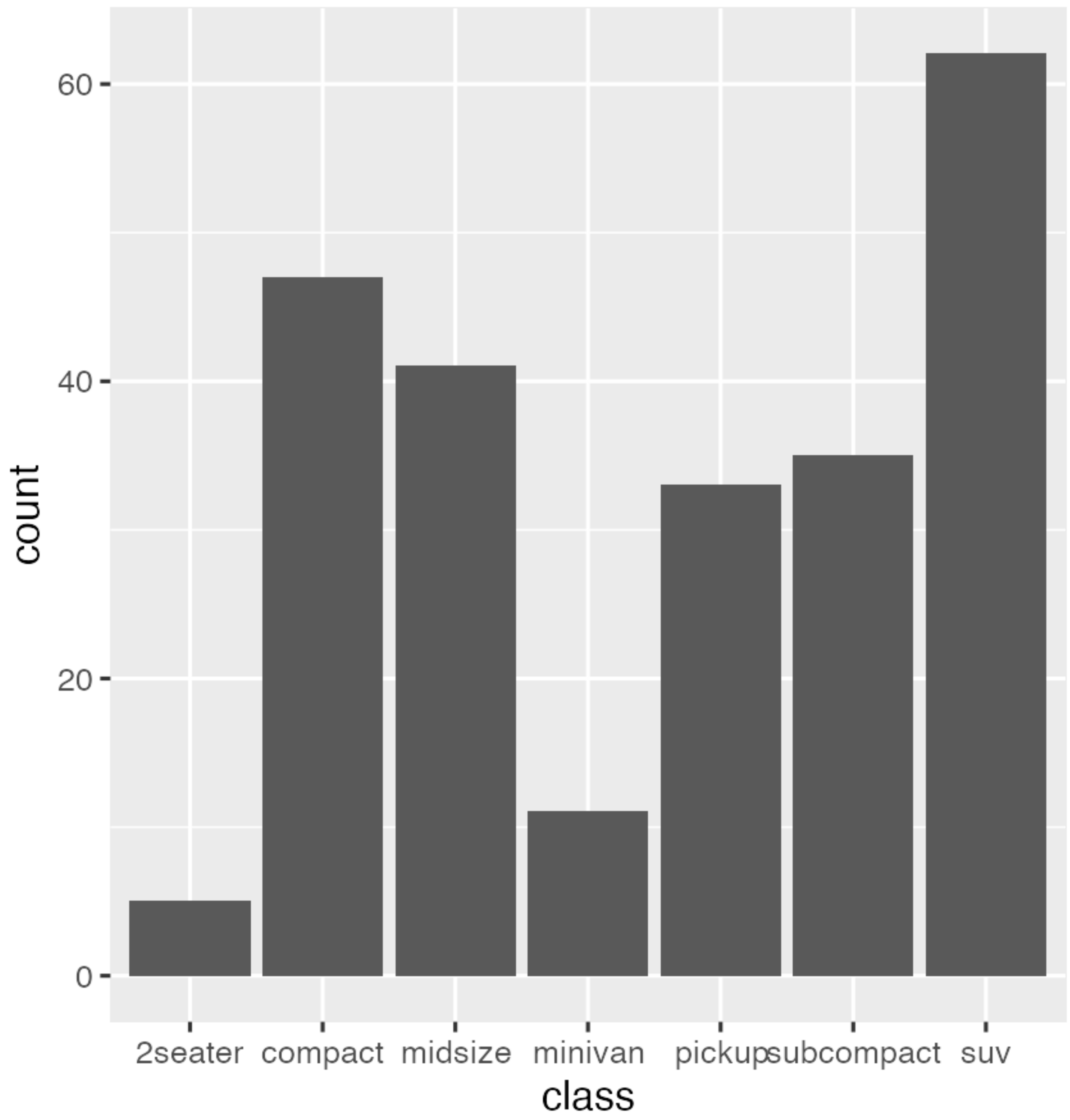
If you have precomputed data use identity stat



THE GGPLOT2 API

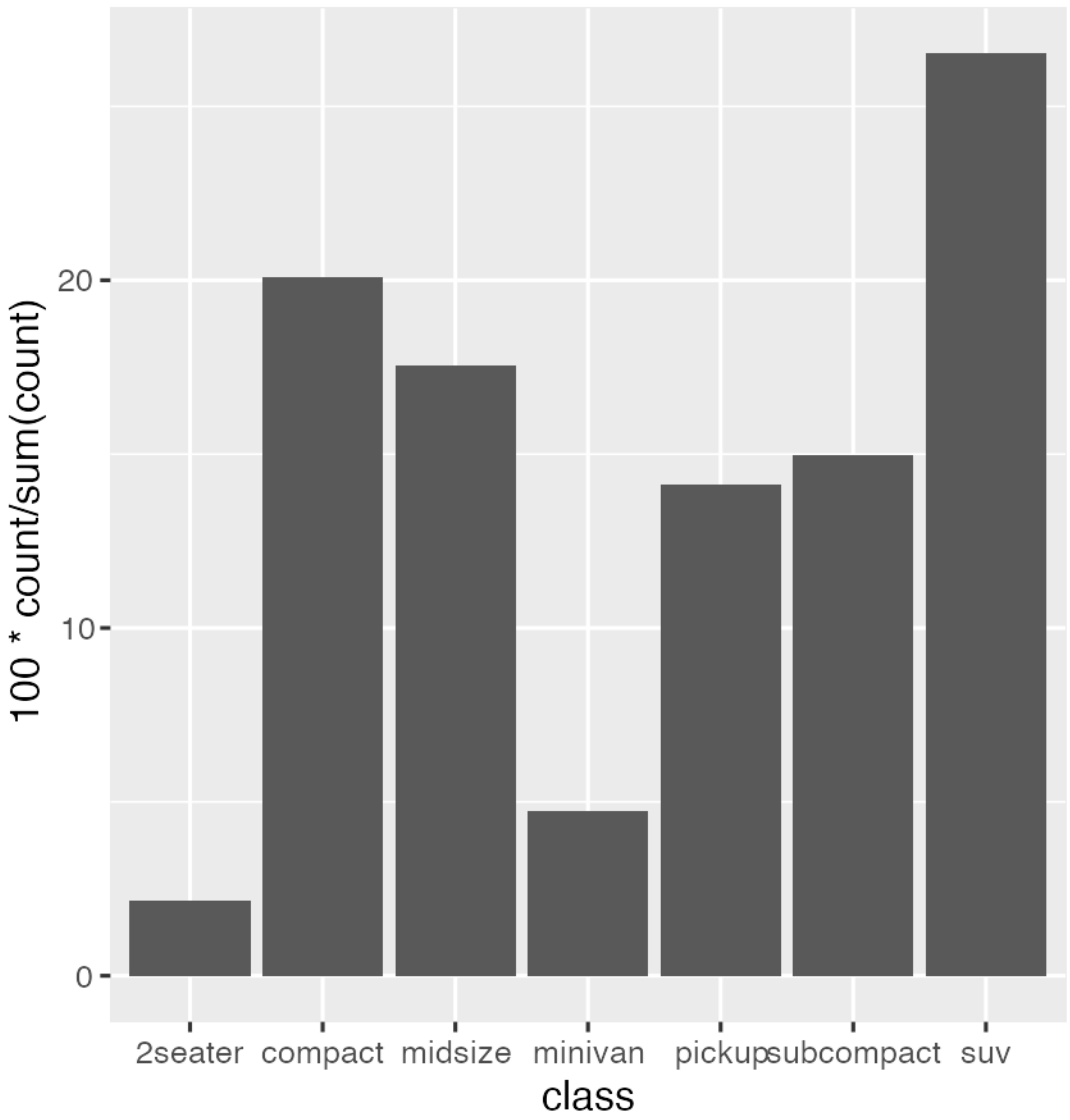
```
mpg_counted ← mpg %>%
  count(class, name = 'count')

ggplot(mpg_counted) +
  geom_col(aes(x = class, y = count))
```

… or use the geom_col() shortcut



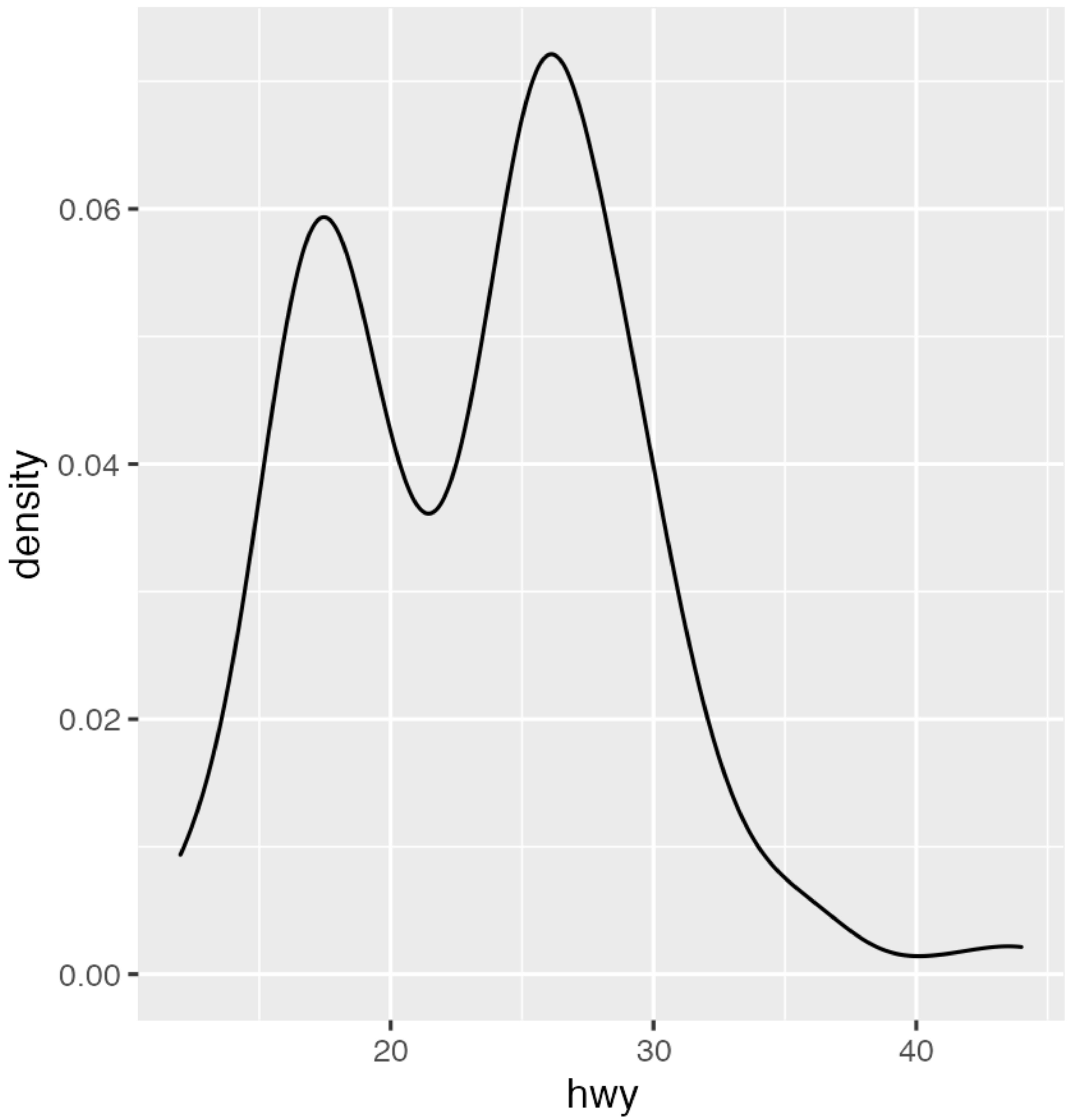
```
ggplot(mpg) +
  geom_bar(
    aes(
      x = class,
      y = after_stat(100 * count / sum(count))
    )
  )
```

Use after_stat() to modify mapping from stats (here we calculate %)



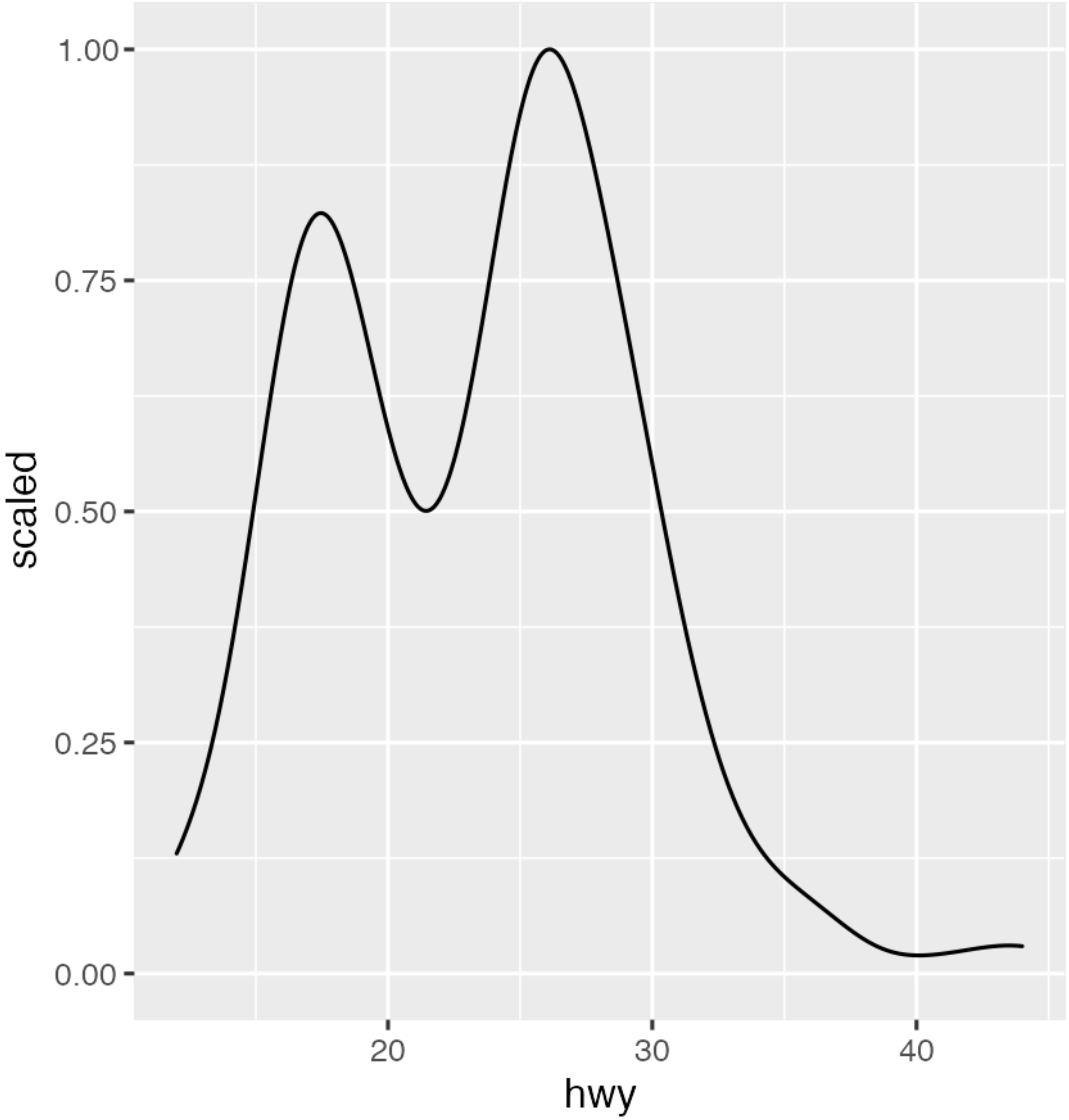
```
ggplot(mpg) +
  geom_density(aes(x = hwy))
```

Many stats provide multiple calculated values and use one by default (here ‘density’)



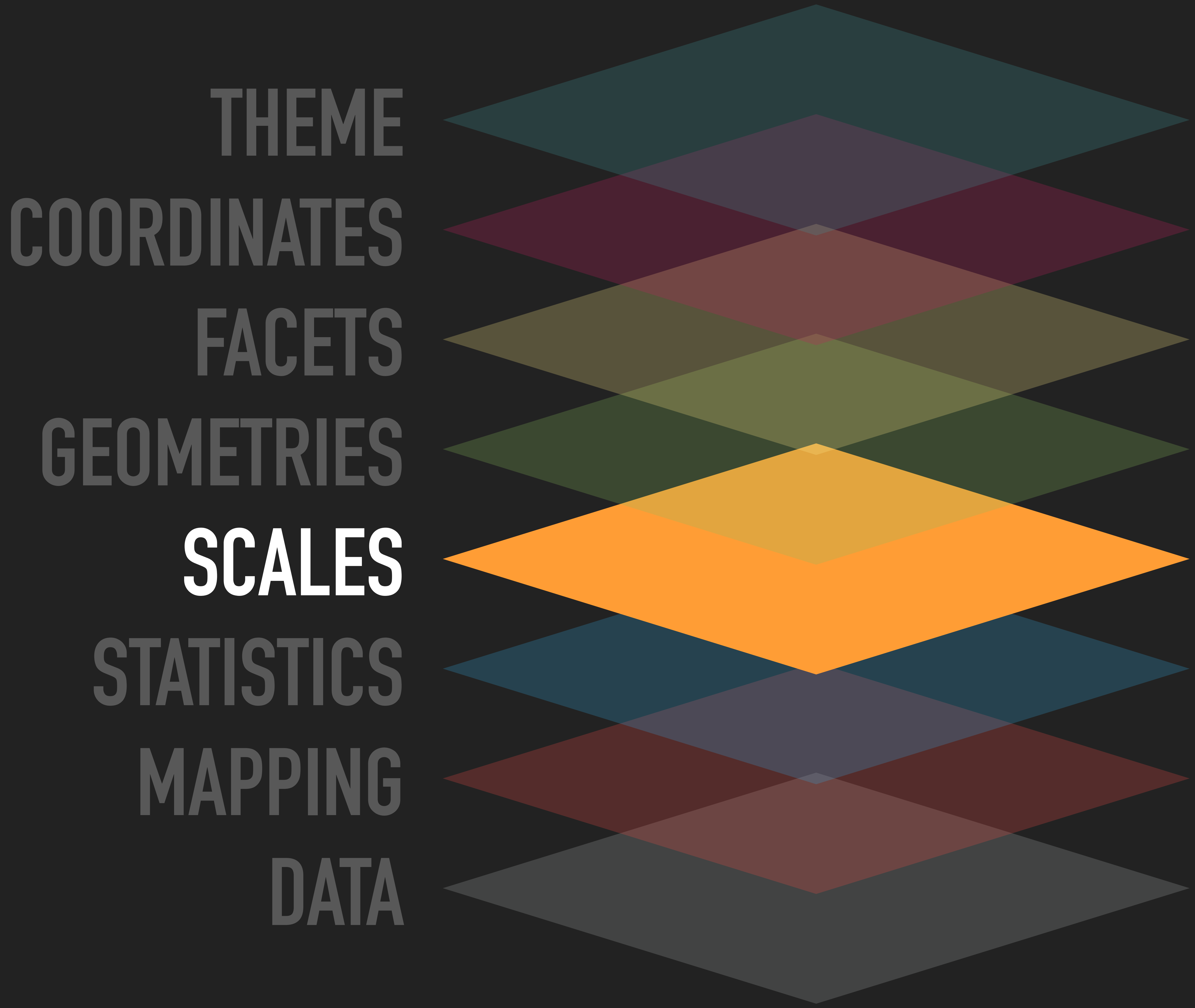
```
ggplot(mpg) +
  geom_density(aes(x = hwy,
                   y = after_stat(scaled)))
```

As before, these can be accessed with the after_stat() function



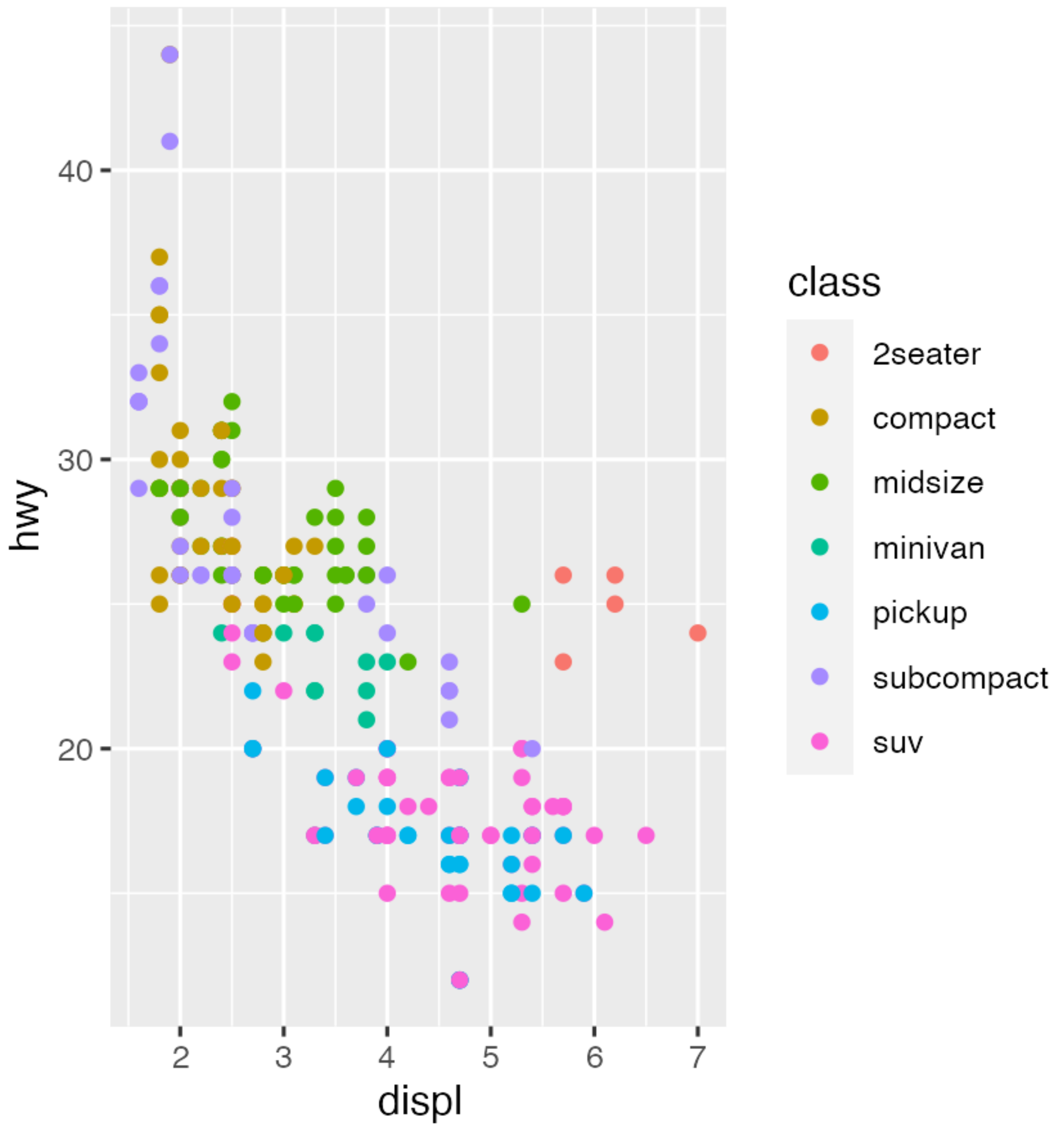
SCALES

- Everything inside aes() will have a scale
  - If none is provided it will get a default
- Scales follow a predictable naming scheme: scale_<aesthetic>_<type>()
- <type> can either be a generic (continuous, discrete, or binned) or specific (e.g. area, for scaling size to circle area)



```
ggplot(mpg) +
  geom_point(
    aes(x = displ, y = hwy, colour = class)
  )
```

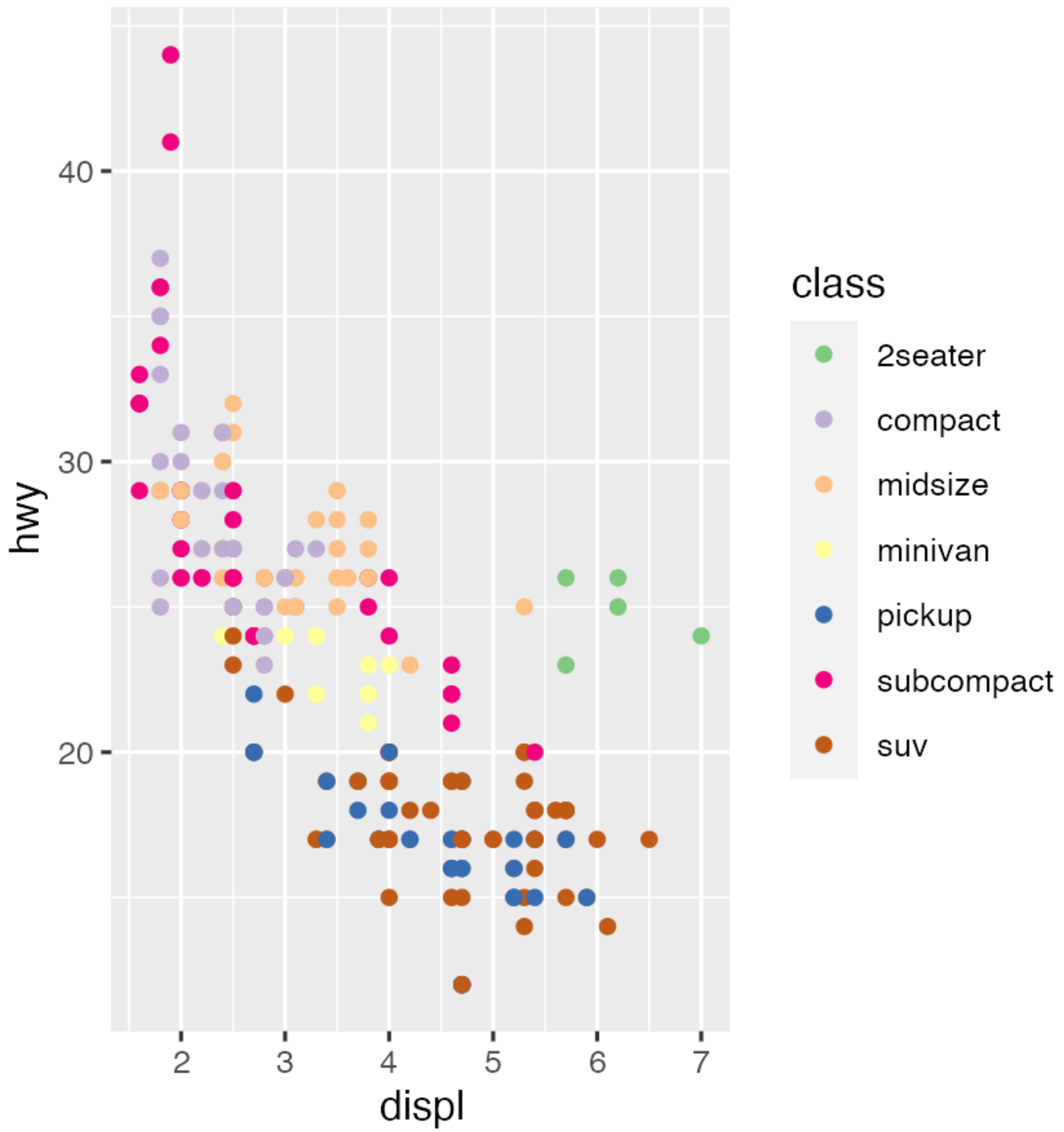
based on the vector type of class, a discrete colour scale is picked



```
ggplot(mpg) +
  geom_point(
    aes(x = displ, y = hwy, colour = class)
  ) +
  scale_colour_brewer(type = 'qual')
```

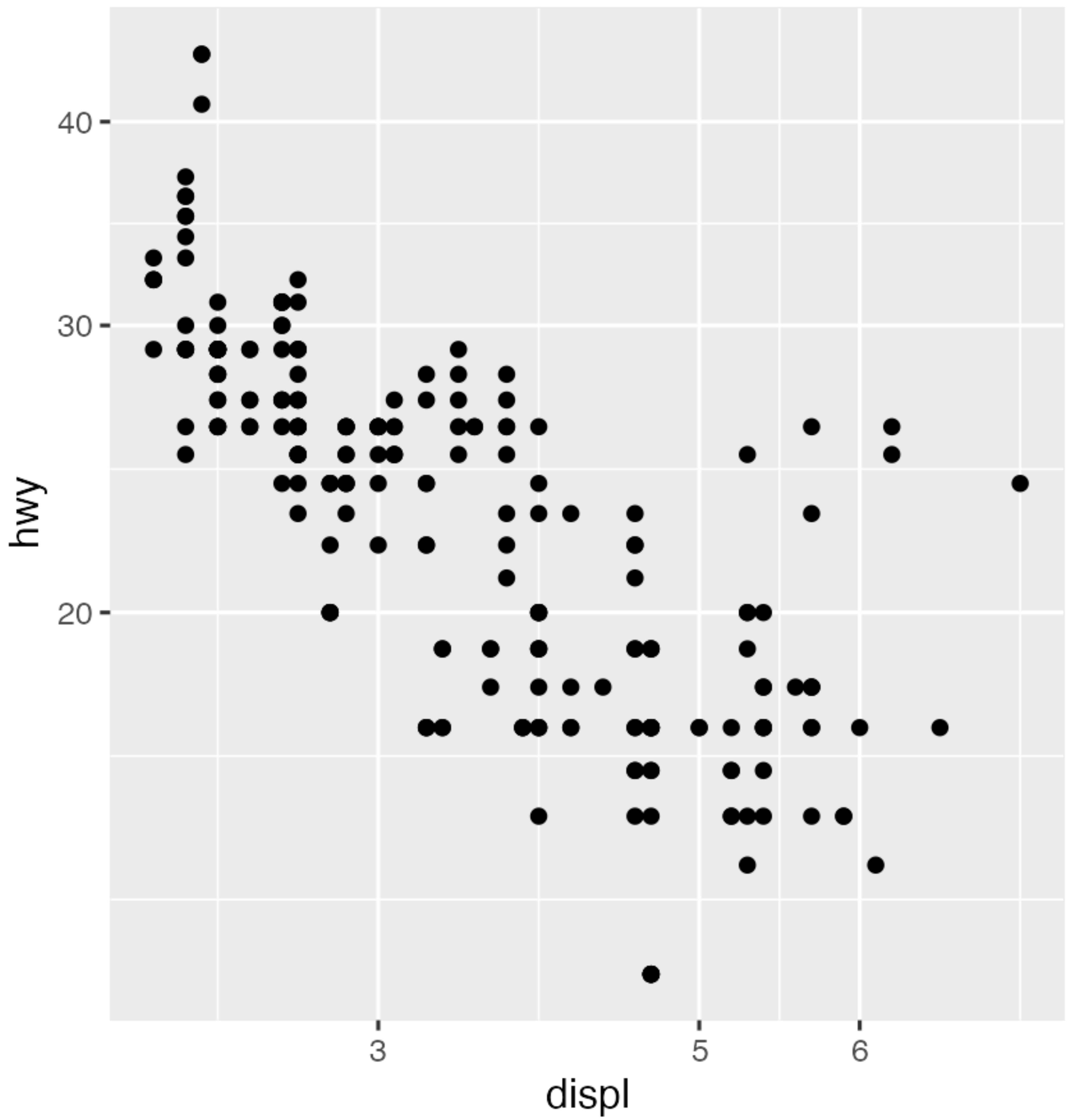
We take control by adding our own explicitly

While function name is predictable, arguments are not



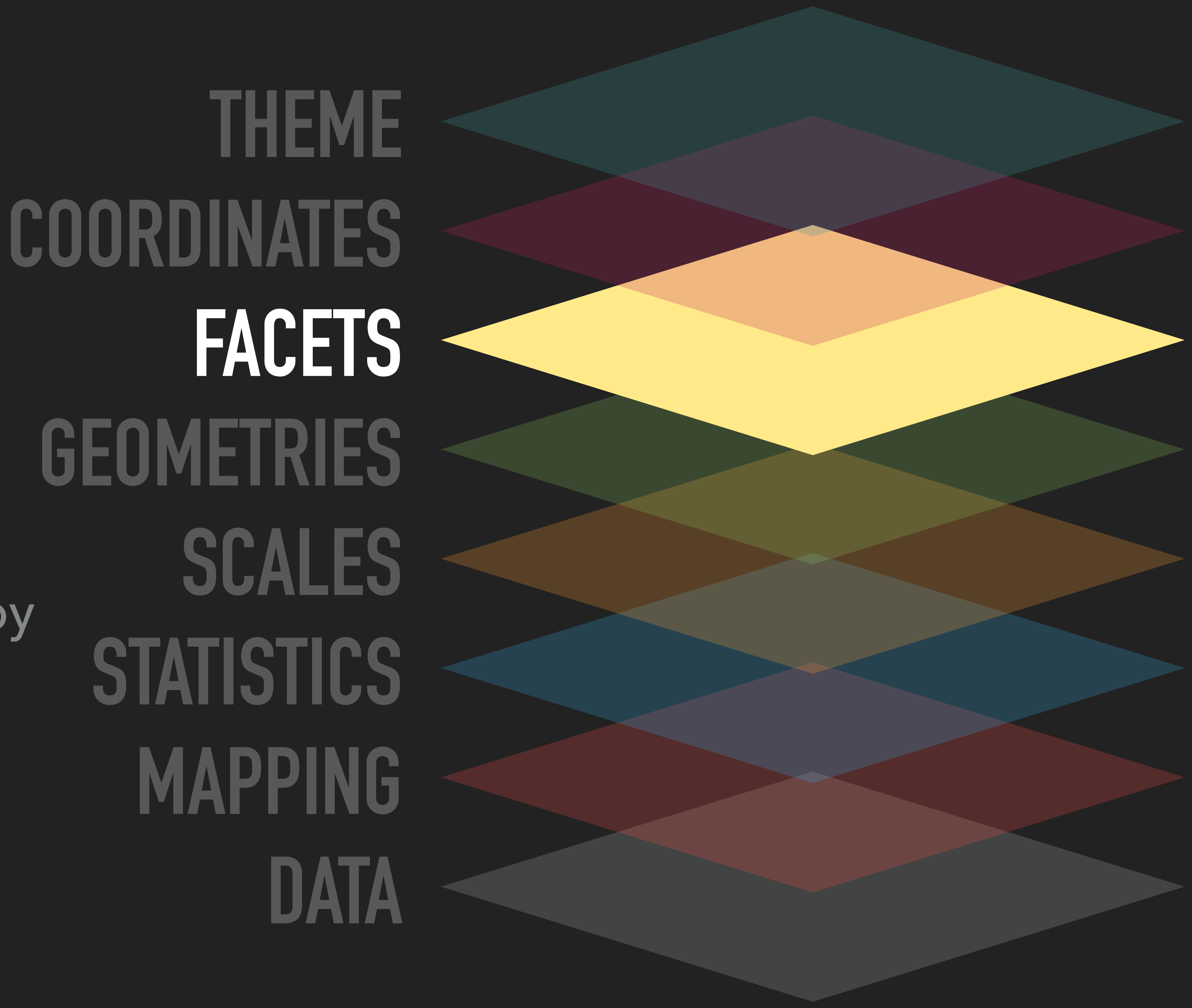
```r
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy)) +
  scale_x_continuous(breaks = c(3, 5, 6)) +
  scale_y_continuous(trans = 'log10')
```

x and y are also controlled with scales



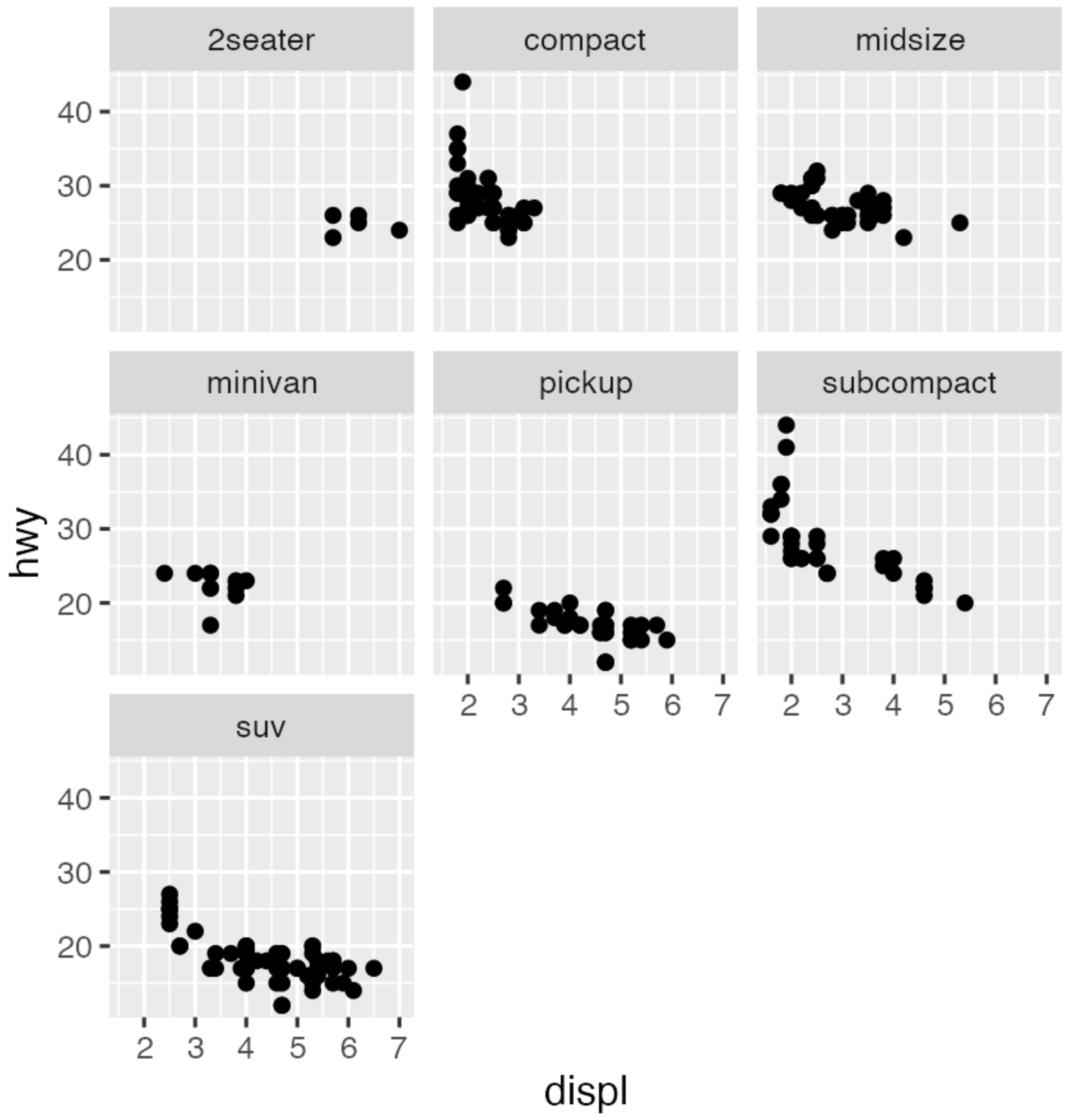
FACETS

- Split data into multiple panels
- Each panel is a representation of the same underlying logic
- Should not be used to combine multiple separate plots
- ggplot2 provide two facets for splitting data by categories

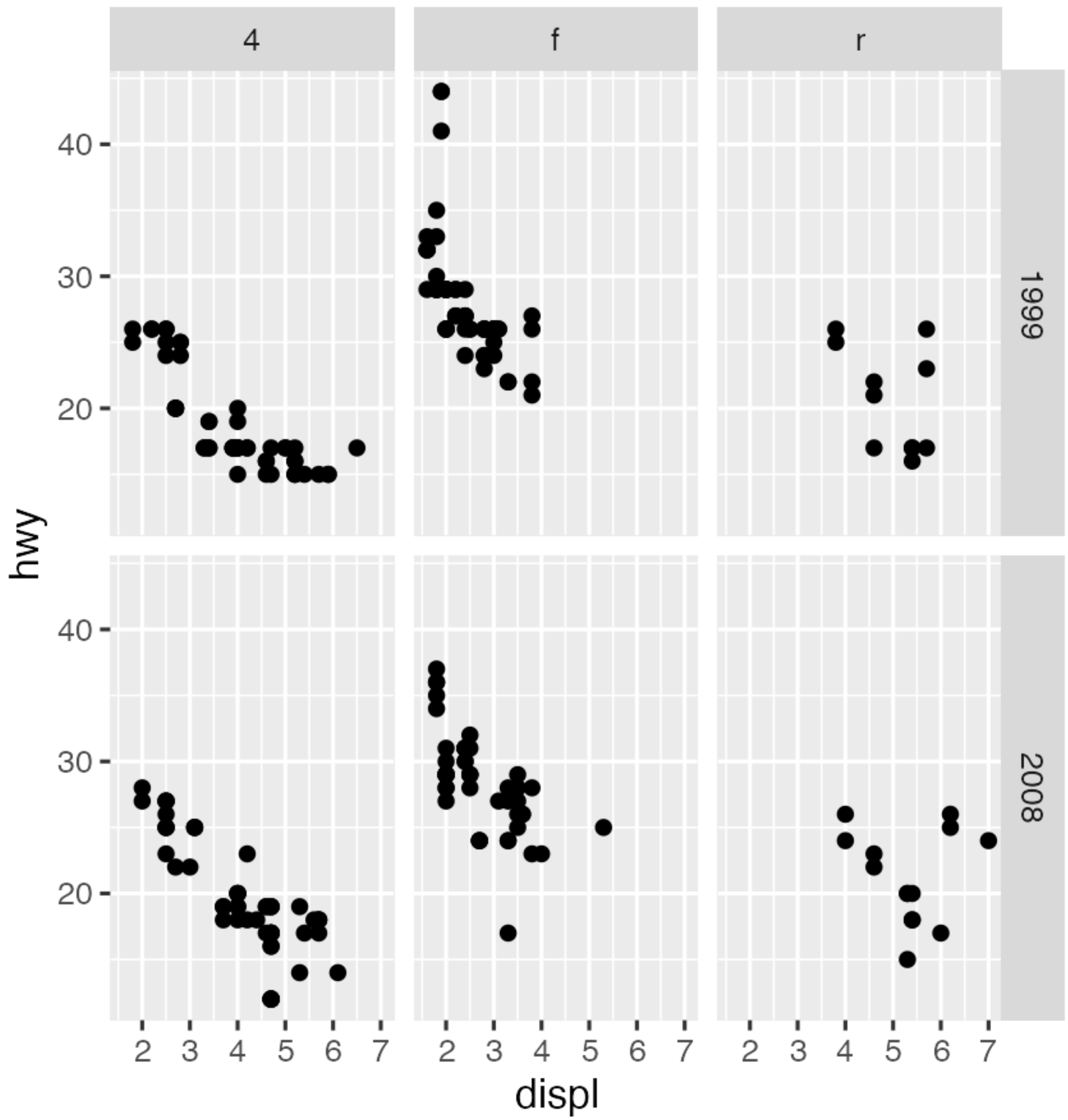


```r
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy)) +
  facet_wrap(~ class)
```

Faceting is often the best way to avoid overplotting



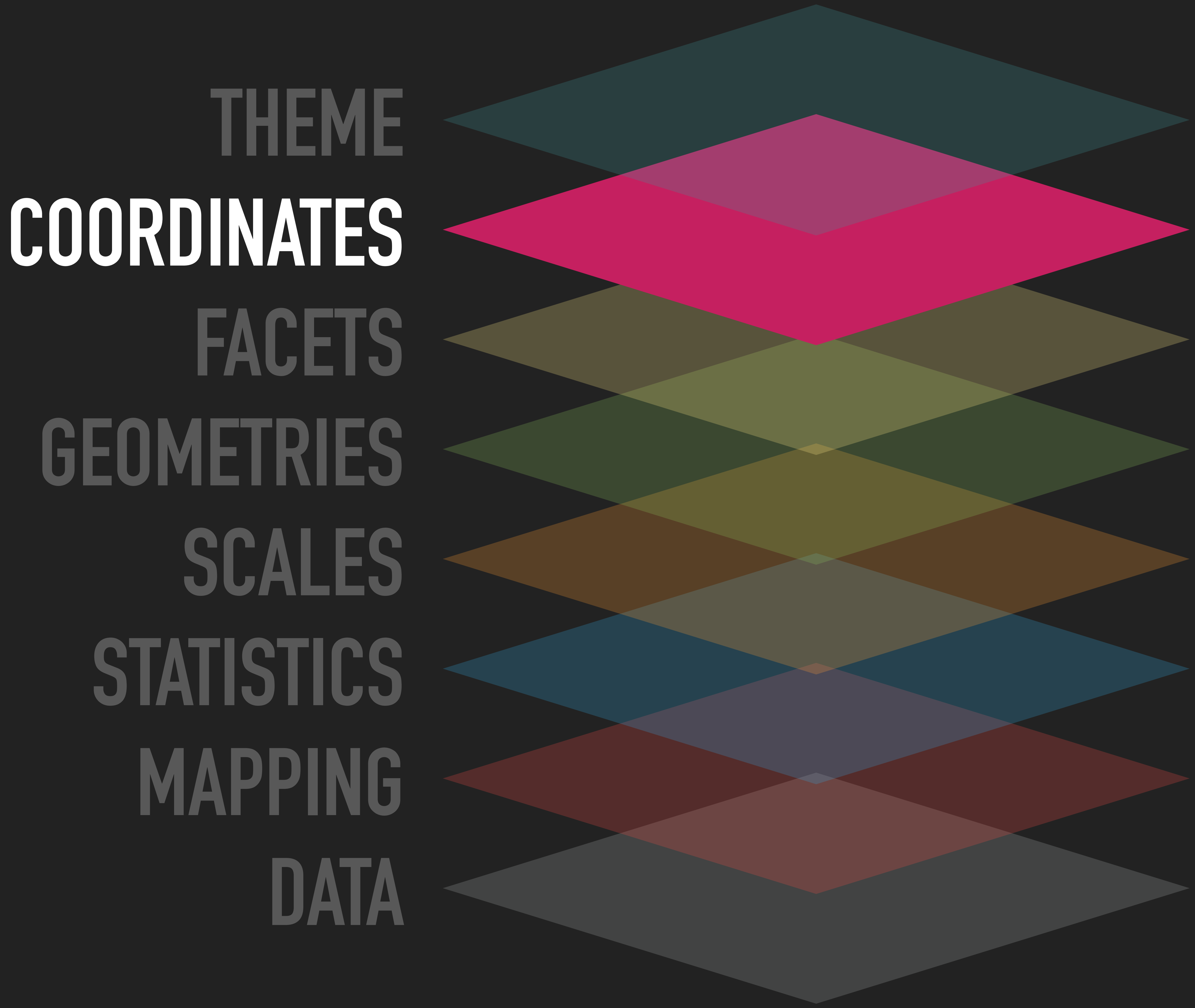
```
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy)) +
  facet_grid(year ~ drv)
```



facet_grid() provides a way of doing graphic pivots

COORDINATES

- What kind of canvas should the final data be drawn on?
    - i.e. how should x and y be interpreted.
- Limits and transformation can be applied in scale or in coord
    - scale will apply it in the beginning
    - coord will apply it in the end
    - you usually want coord
- Extremely useful in cartography (map projections)



THEME

COORDINATES

FACETS

GEOMETRIES

SCALES

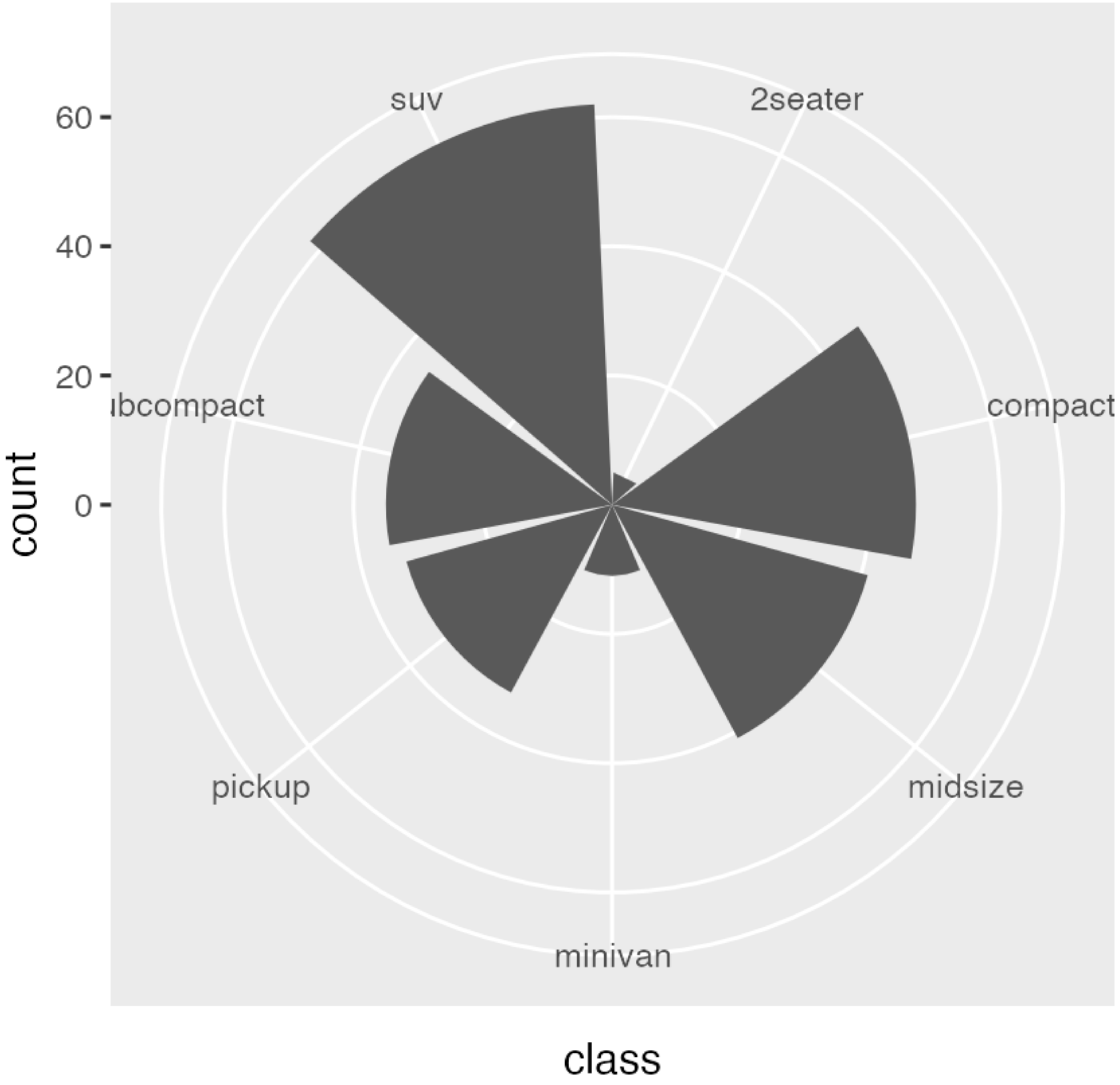
STATISTICS

MAPPING

DATA

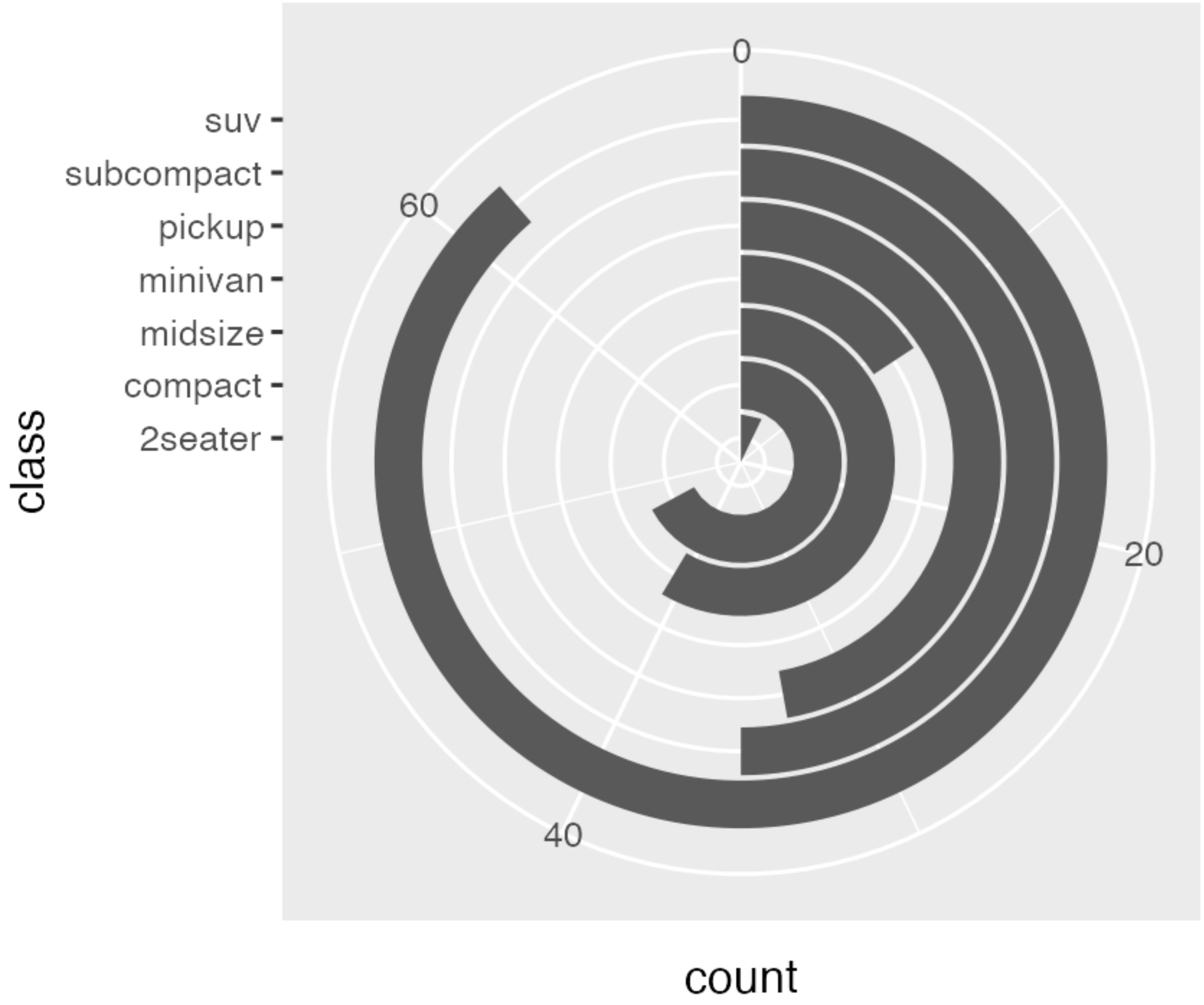
```
ggplot(mpg) +
  geom_bar(aes(x = class)) +
  coord_polar()
```

A polar coordinate system interprets x and y as radius and angle



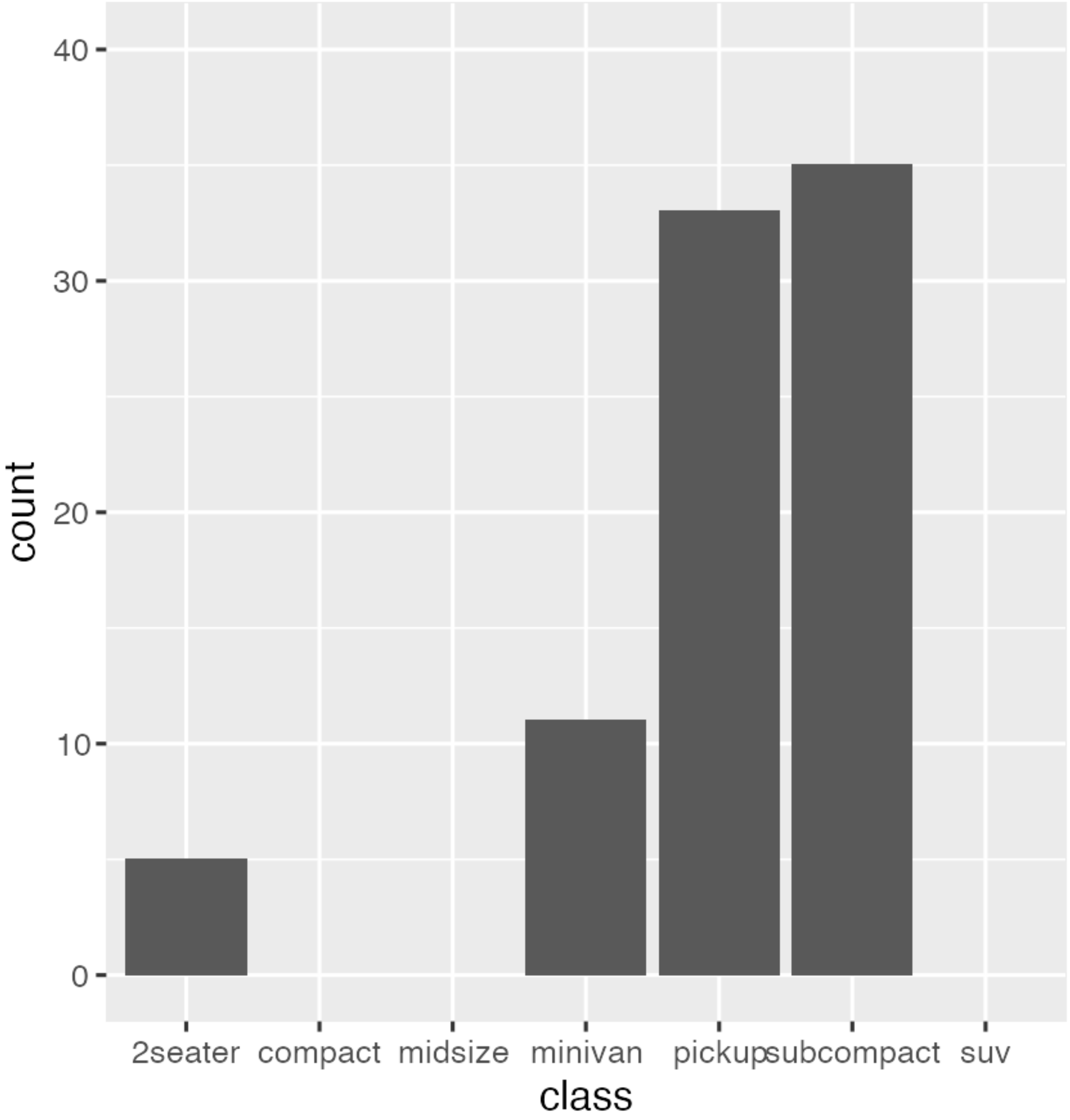
```
ggplot(mpg) +
  geom_bar(aes(x = class)) +
  coord_polar(theta = 'y') +
  expand_limits(y = 70)
```

Changing what is mapped to angle gives a very different plot

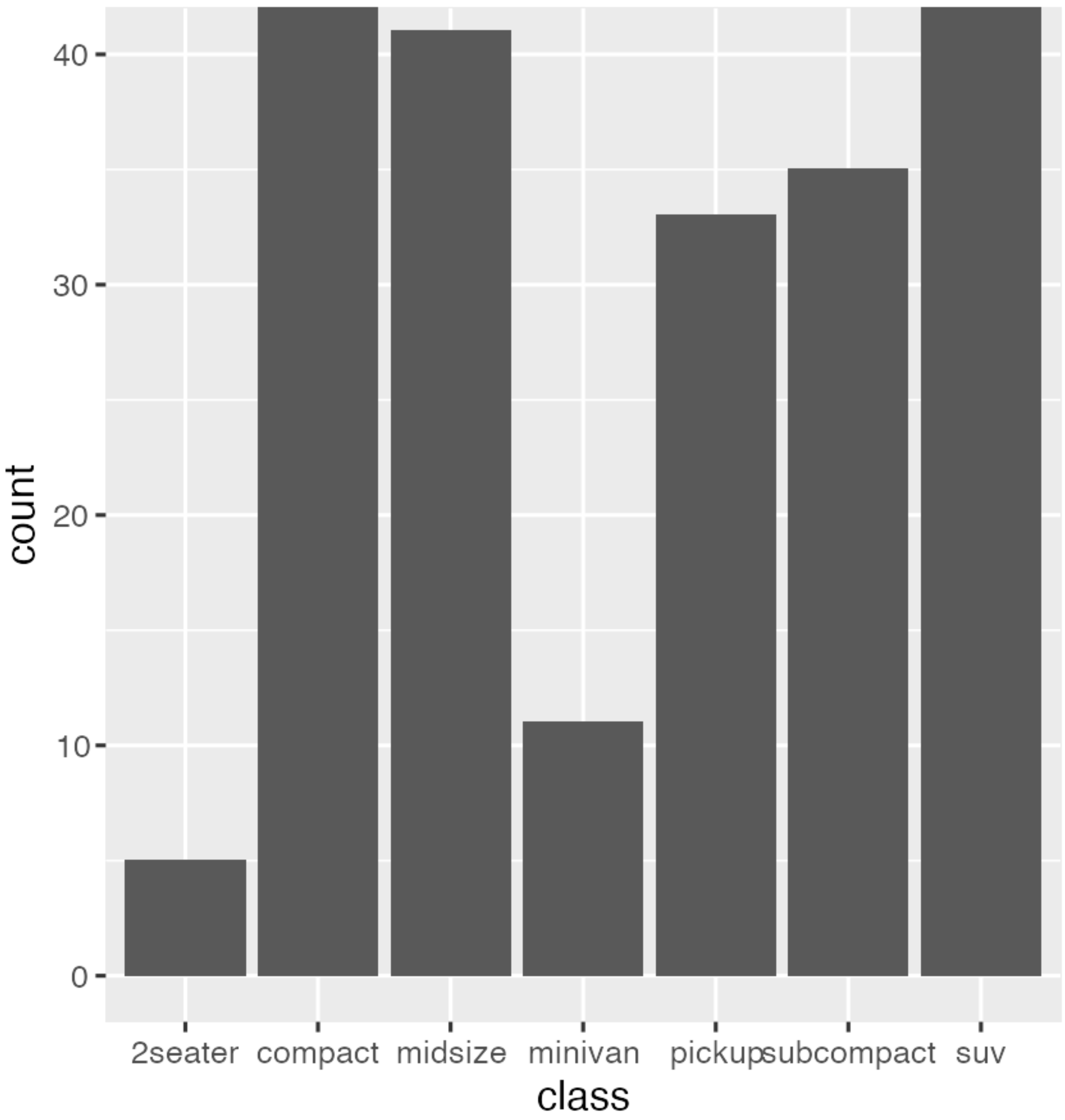


```
ggplot(mpg) +
  geom_bar(aes(x = class)) +
  scale_y_continuous(limits = c(0, 40))
#> Warning message:
#> Removed 3 rows containing missing values
(geom_bar).
```

Zooming with scale removes data outside limits



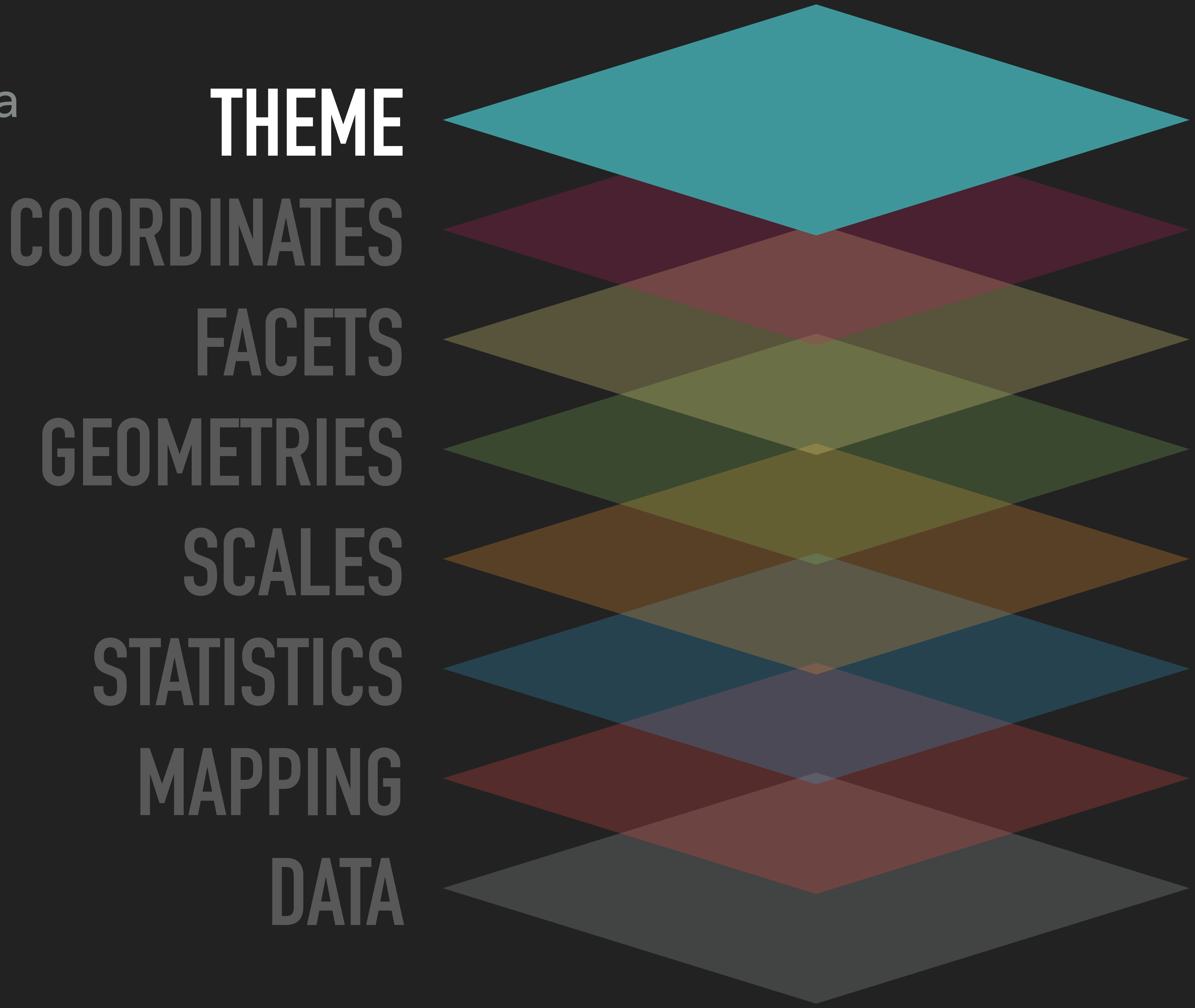
```
ggplot(mpg) +
  geom_bar(aes(x = class)) +
  coord_cartesian(ylim = c(0, 40))
```



Zooming with coord creates a proper zoom

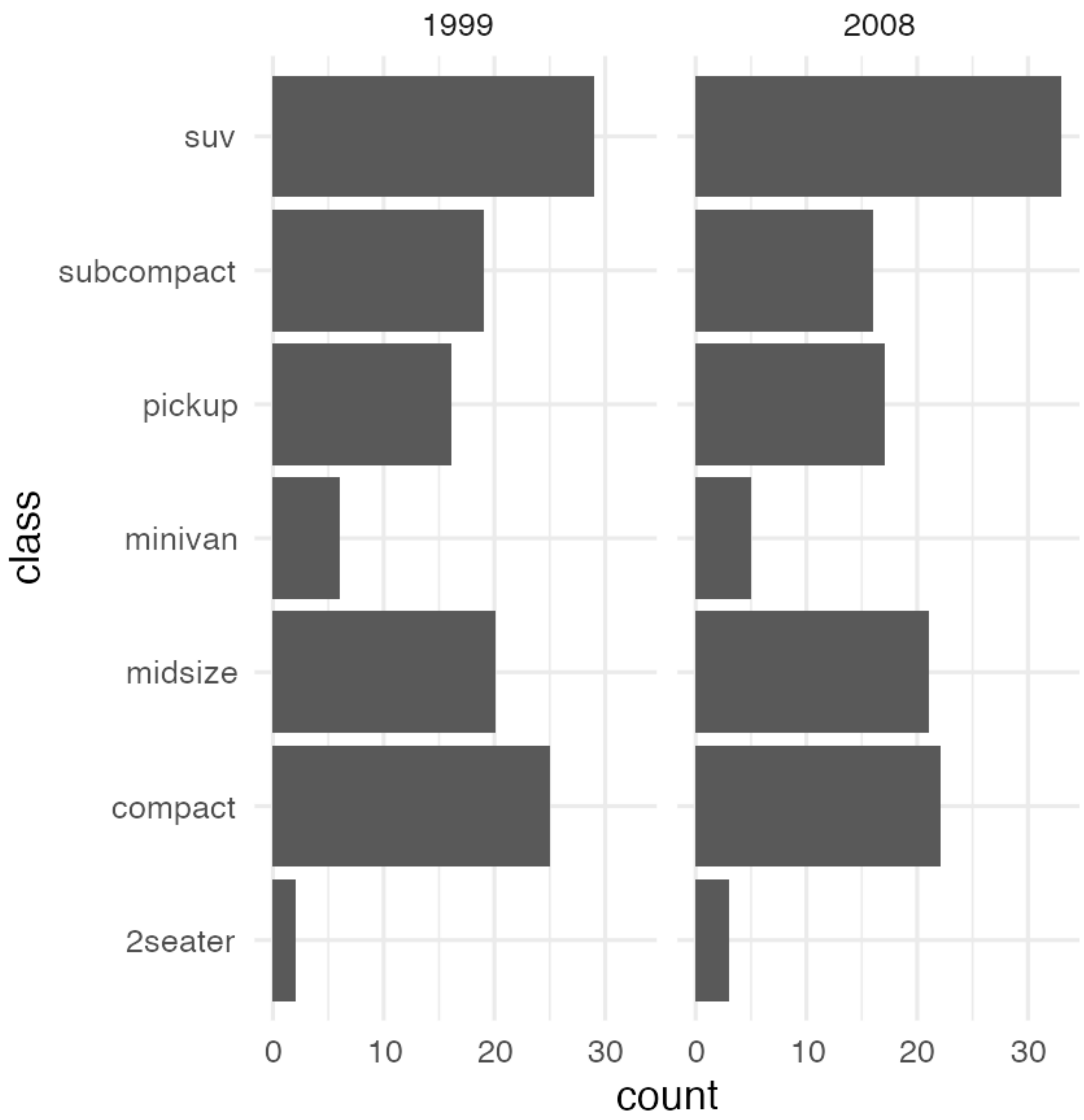
THEME

- Stylistic changes to the plot not related to data
- Can both apply complete themes or modify elements directly
- Theming is hierarchical



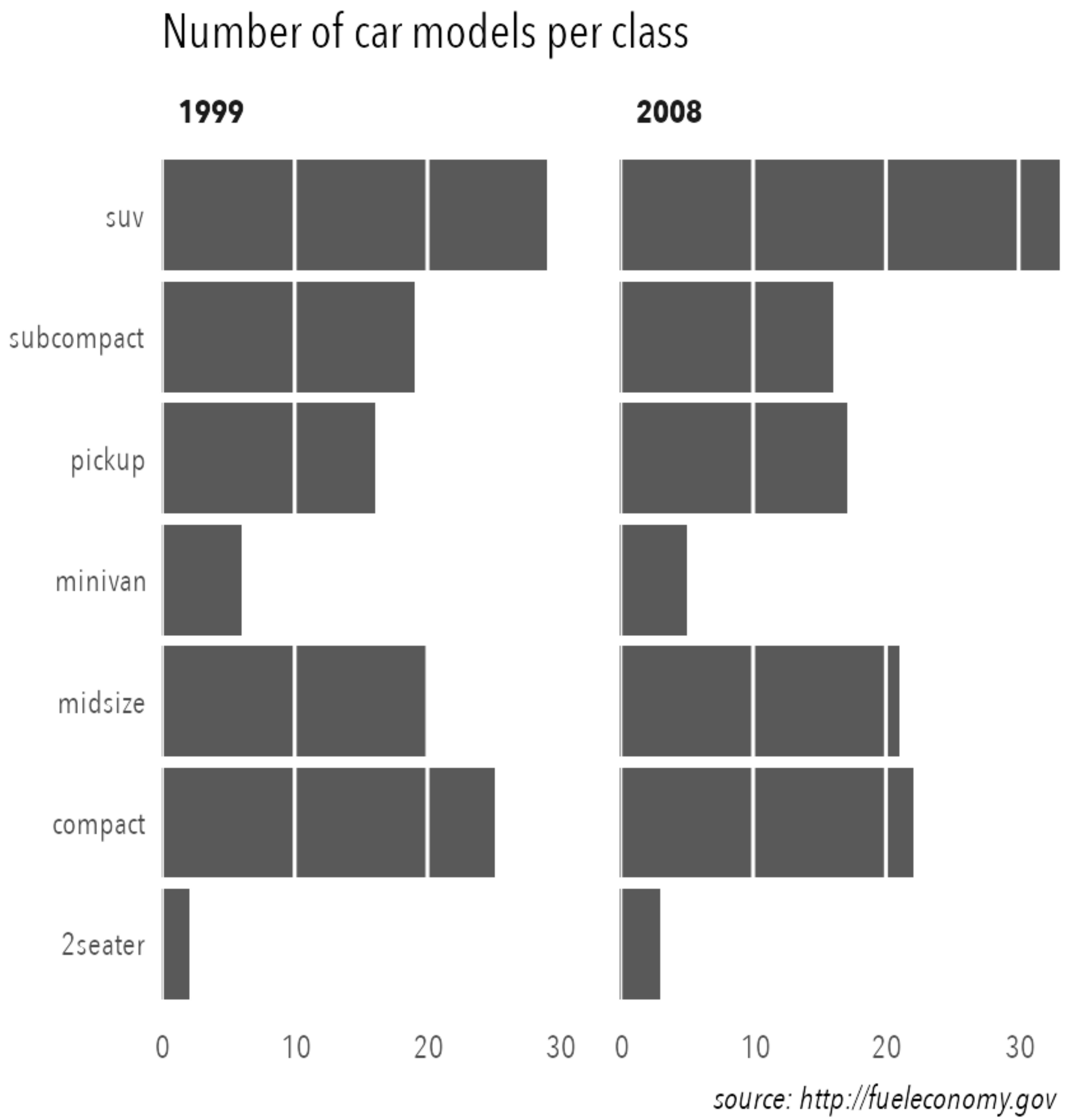
```
ggplot(mpg) +
  geom_bar(aes(y = class)) +
  facet_wrap(~year) +
  theme_minimal()
```

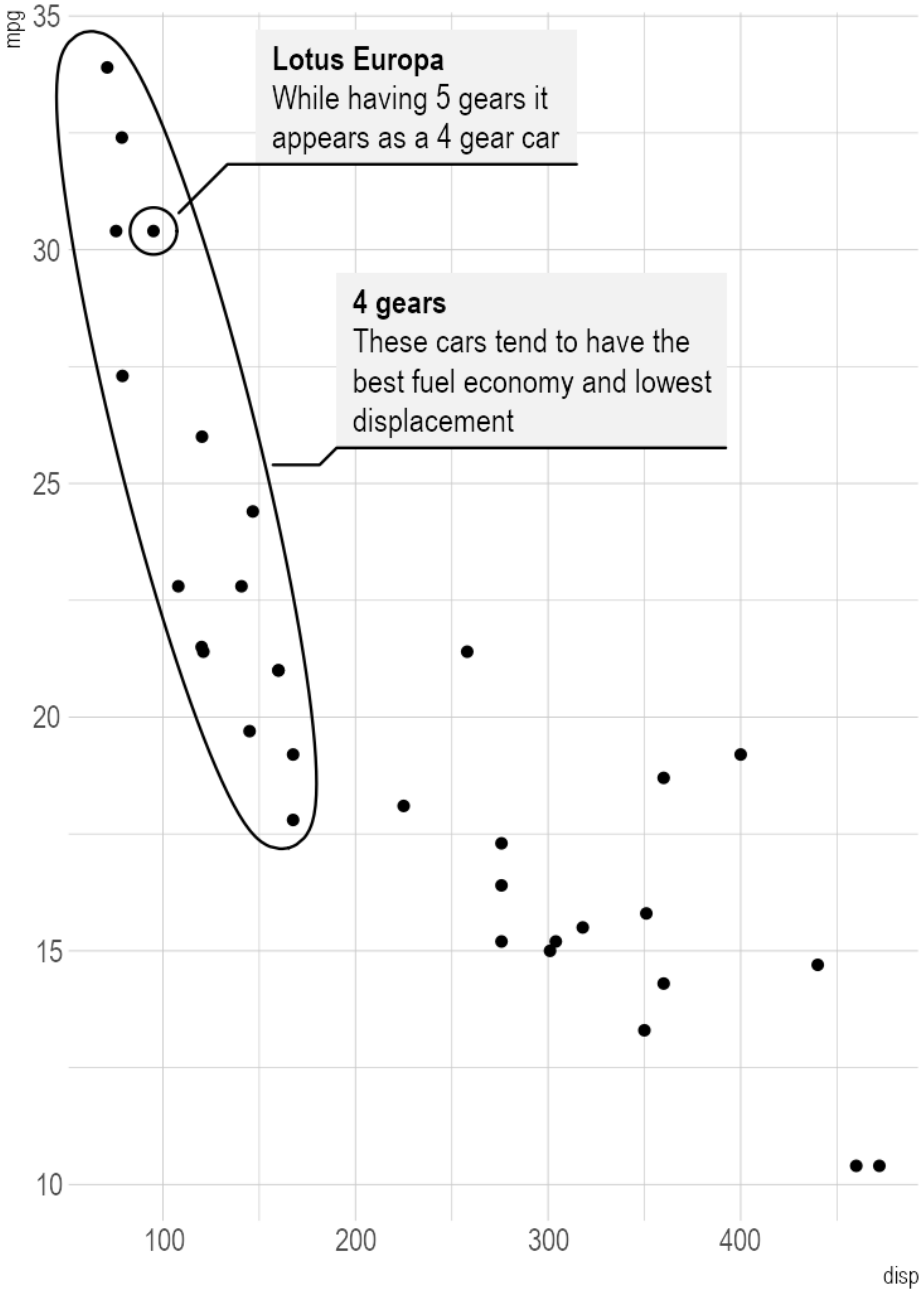
It is quick to change the overall look with a build-in theme…



THE GGPLOT2 API

```
ggplot(mpg) +
  geom_bar(aes(y = class)) +
  facet_wrap(~year) +
  labs(title = "Number of car models per class",
       caption = "source: http://fueleconomy.gov",
       x = NULL,
       y = NULL) +
  scale_x_continuous(expand = c(0, NA)) +
  theme_minimal() +
  theme(
    text = element_text('Avenir Next Condensed'),
    strip.text = element_text(face = 'bold',
                              hjust = 0),
    plot.caption = element_text(face = 'italic'),
    panel.grid.major = element_line('white',
                                    size = 0.5),
    panel.grid.minor = element_blank(),
    panel.grid.major.y = element_blank(),
    panel.ontop = TRUE
  )
```



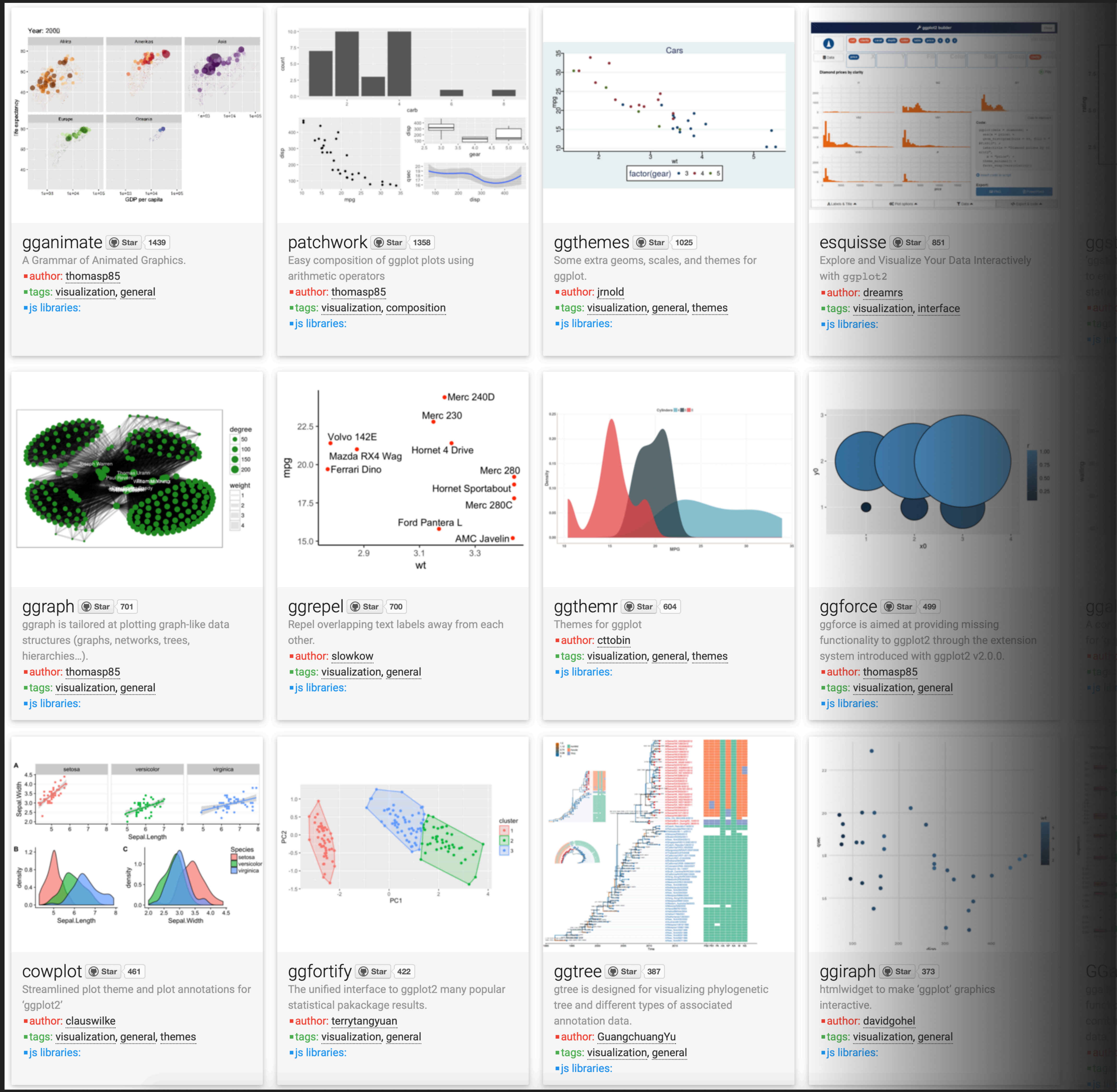


PART 3:

BEYOND GGPLOT2

WHY EXTENSIONS

- ggplot2 is huge! Maintenance is already a team effort
  - 47 geoms
  - 25 stats
  - 62 scales
- Many extensions are very niche specific and better developed by experts in the field
- It is easier to promote focused packages
- www.ggplot2-exts.org



gganimate Star 1439
A Grammar of Animated Graphics.
- author: thomasp85
- tags: visualization, general
- js libraries:

patchwork Star 1358
Easy composition of ggplot plots using arithmetic operators
- author: thomasp85
- tags: visualization, composition
- js libraries:

ggthemes Star 1025
Some extra geoms, scales, and themes for ggplot.
- author: jrnold
- tags: visualization, general, themes
- js libraries:

esquisse Star 851
Explore and Visualize Your Data Interactively with `ggplot2`
- author: dreamrs
- tags: visualization, interface
- js libraries:

ggraph Star 701
ggraph is tailored at plotting graph-like data structures (graphs, networks, trees, hierarchies...).
- author: thomasp85
- tags: visualization, general
- js libraries:

ggrepel Star 700
Repel overlapping text labels away from each other.
- author: slowkow
- tags: visualization, general
- js libraries:

ggthemr Star 604
Themes for ggplot
- author: cttobin
- tags: visualization, general, themes
- js libraries:

ggforce Star 499
ggforce is aimed at providing missing functionality to ggplot2 through the extension system introduced with ggplot2 v2.0.0.
- author: thomasp85
- tags: visualization, general
- js libraries:

cowplot Star 461
Streamlined plot theme and plot annotations for 'ggplot2'
- author: clauswilke
- tags: visualization, general, themes
- js libraries:

ggfortify Star 422
The unified interface to ggplot2 many popular statistical pakackage results.
- author: terrytangyuan
- tags: visualization, general
- js libraries:

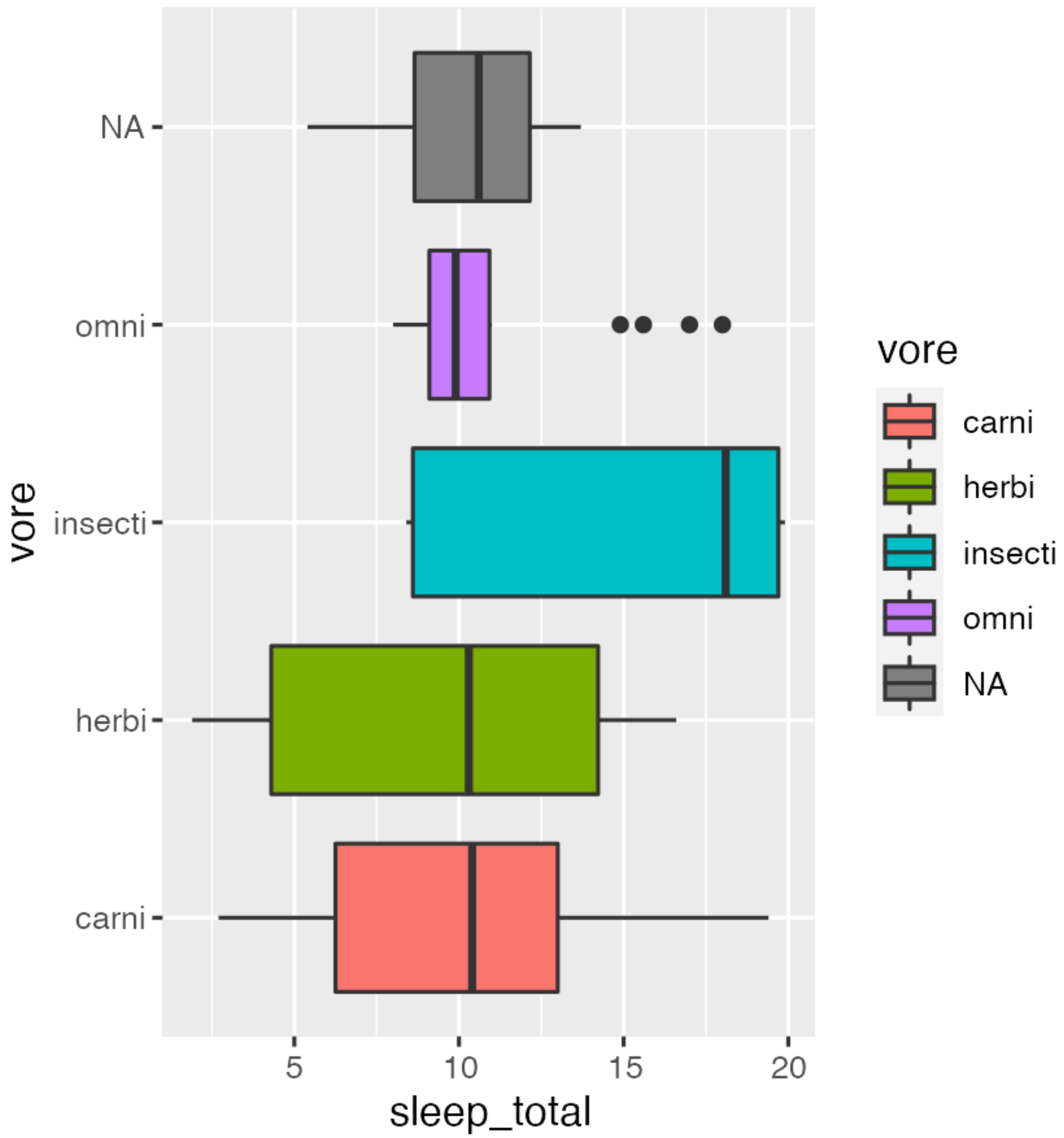
ggtree Star 387
gtree is designed for visualizing phylogenetic tree and different types of associated annotation data.
- author: GuangchuangYu
- tags: visualization, general
- js libraries:

ggiraph Star 373
htmlwidget to make 'ggplot' graphics interactive.
- author: davidgohel
- tags: visualization, general
- js libraries:

PLOT COMPOSITION

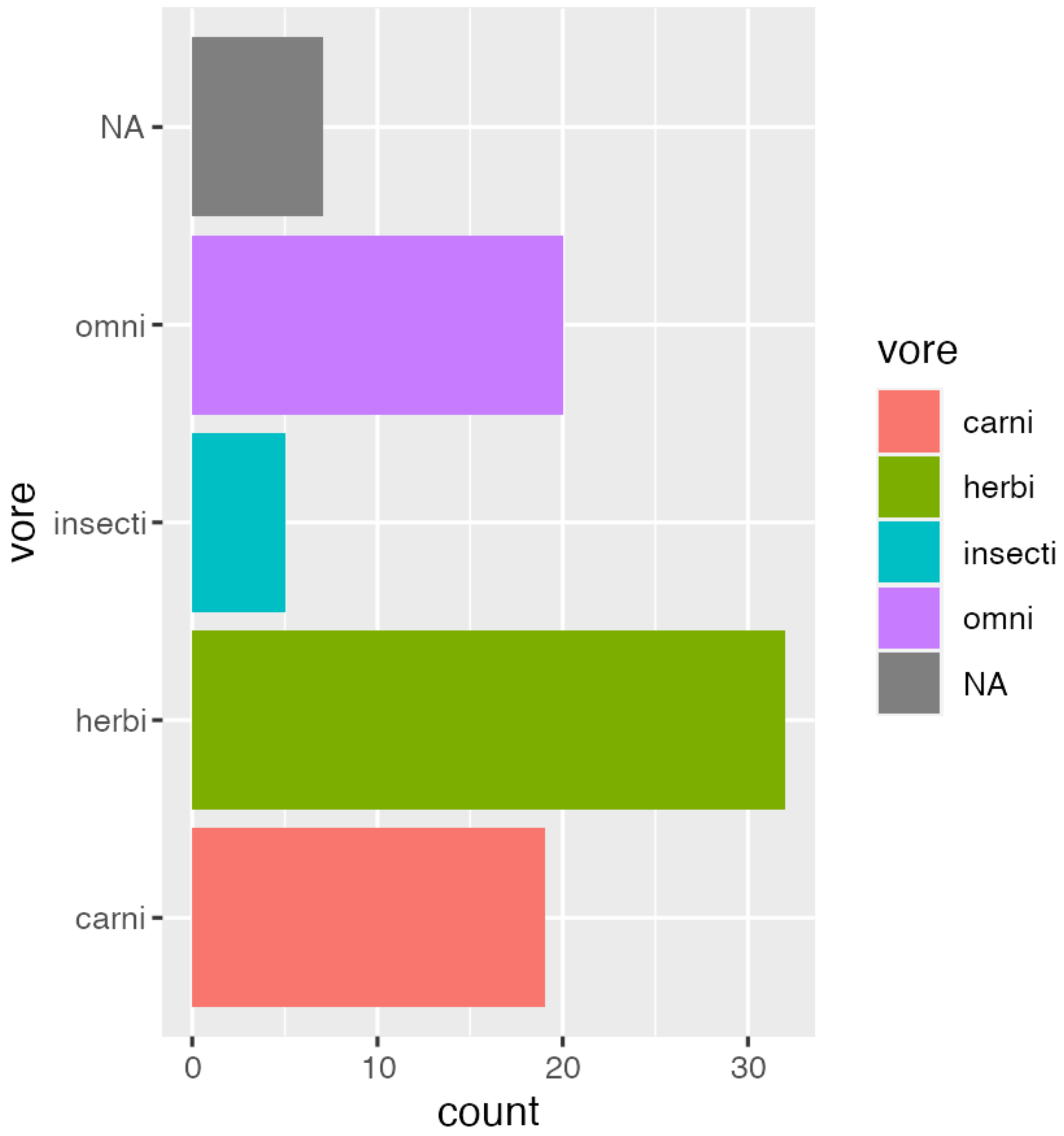
- Stories are told with multiple plots
- Plot layout should be flexible and well aligned
- Many approaches
  - Facet hacking
  - gridExtra::grid.arrange()
  - ggpubr::ggarrange()
  - cowplot::plot_grid()
  - patchwork

```
p1 ← ggplot(msleep) +
  geom_boxplot(aes(x = sleep_total,
                   y = vore,
                   fill = vore))

p2 ← ggplot(msleep) +
  geom_bar(aes(y = vore, fill = vore))

p3 ← ggplot(msleep) +
  geom_point(aes(x = bodywt,
                 y = sleep_total,
                 colour = vore)) +
  scale_x_log10()

p1
```

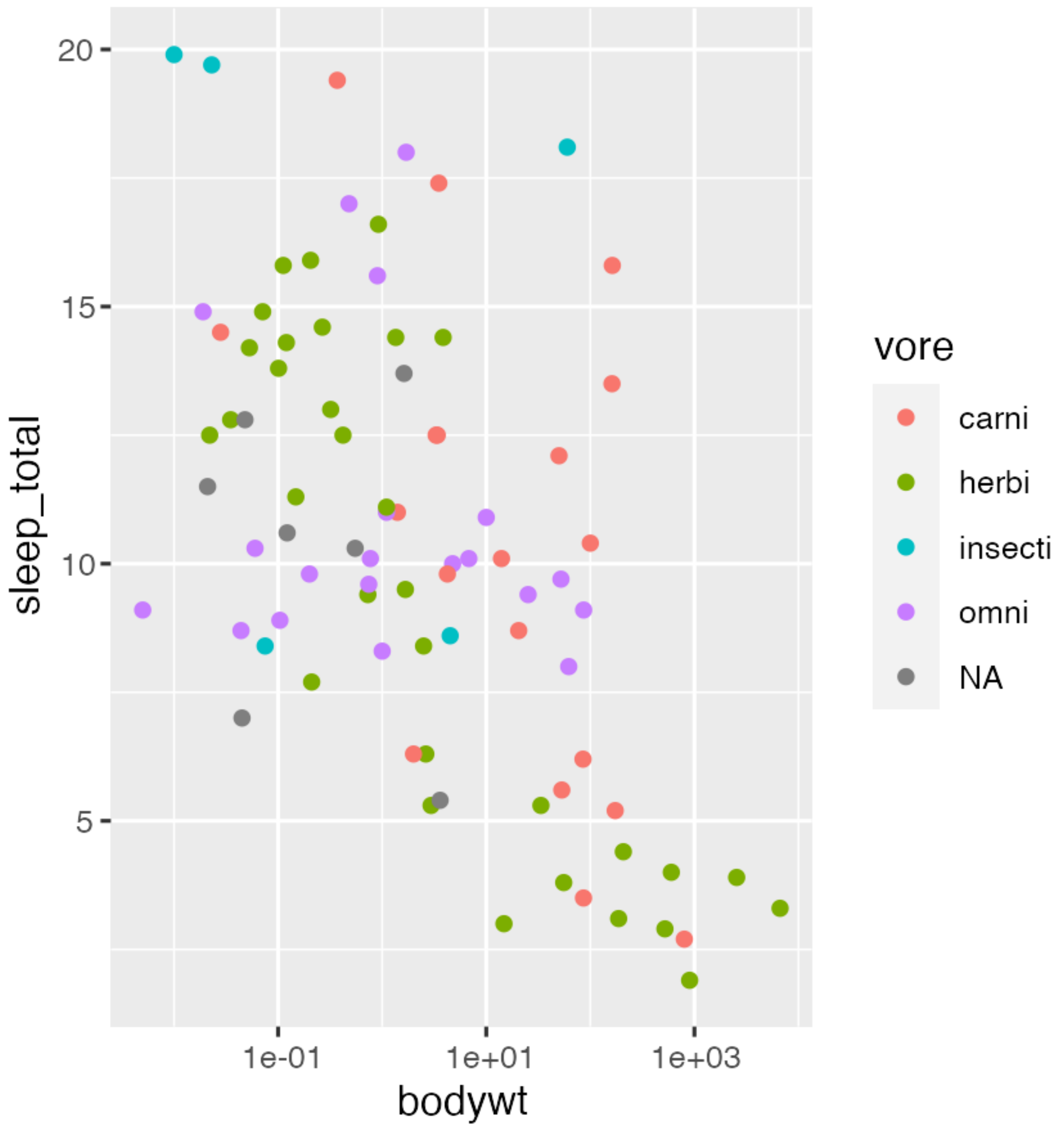


BEYOND GGPLOT2

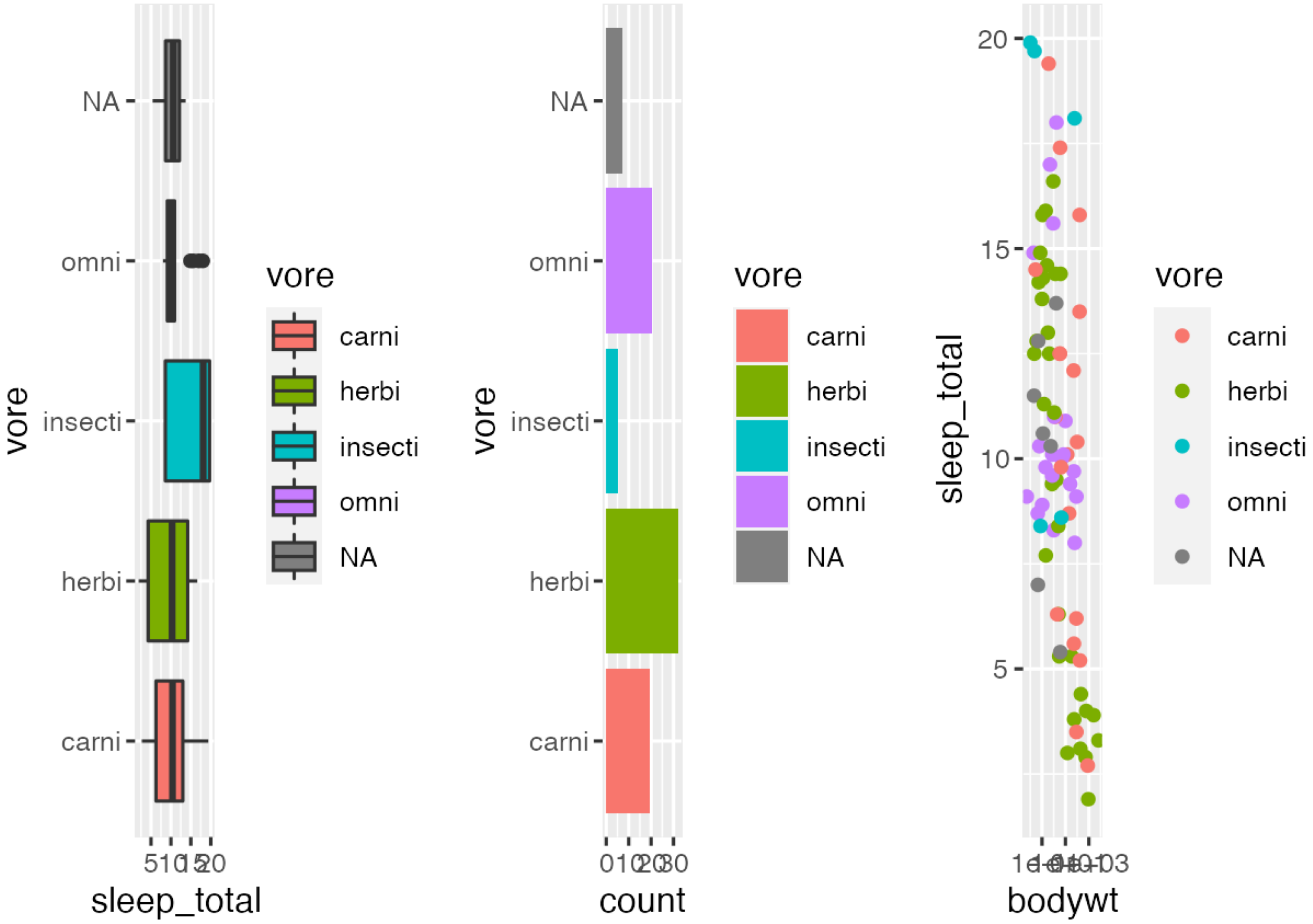
```
p1 ← ggplot(msleep) +
  geom_boxplot(aes(x = sleep_total,
                   y = vore,
                   fill = vore))

p2 ← ggplot(msleep) +
  geom_bar(aes(y = vore, fill = vore))

p3 ← ggplot(msleep) +
  geom_point(aes(x = bodywt,
                 y = sleep_total,
                 colour = vore)) +
  scale_x_log10()

p2
```



```
p1 ← ggplot(msleep) +
  geom_boxplot(aes(x = sleep_total,
                   y = vore,
                   fill = vore))

p2 ← ggplot(msleep) +
  geom_bar(aes(y = vore, fill = vore))

p3 ← ggplot(msleep) +
  geom_point(aes(x = bodywt,
                 y = sleep_total,
                 colour = vore)) +
  scale_x_log10()

p3
```

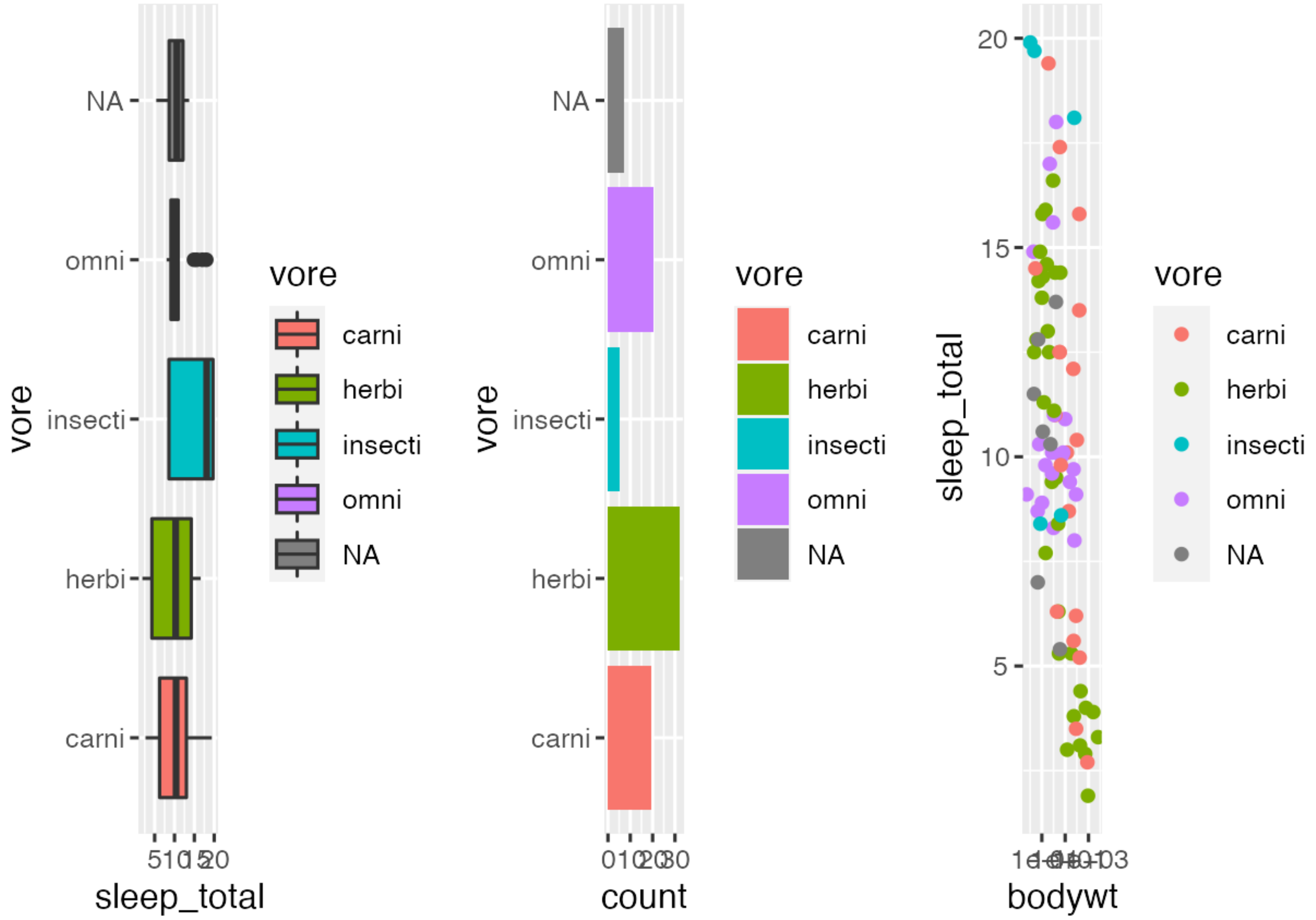


```
p1 + p2 + p3
```


vore
NA
omni
insecti
herbi
carni
510 15 20
sleep_total
vore
carni
herbi
insecti
omni
NA
vore
NA
omni
insecti
herbi
carni
0 10 20 30
count
vore
carni
herbi
insecti
omni
NA
sleep_total
20
15
10
5
1e-04e+01e+03
bodywt
vore
carni
herbi
insecti
omni
NA

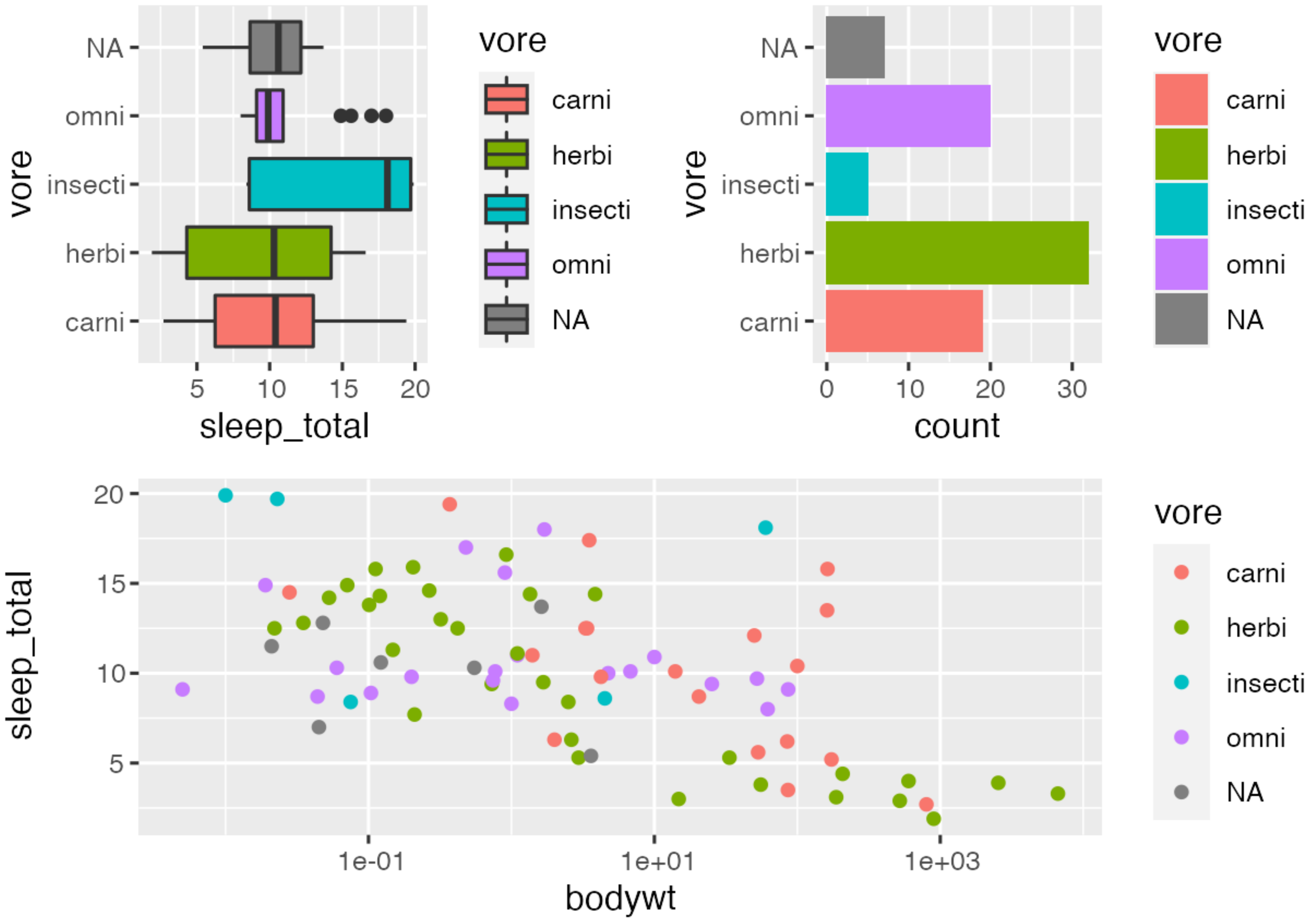
BEYOND GGPLOT2

```
p1 + p2 + p3
```

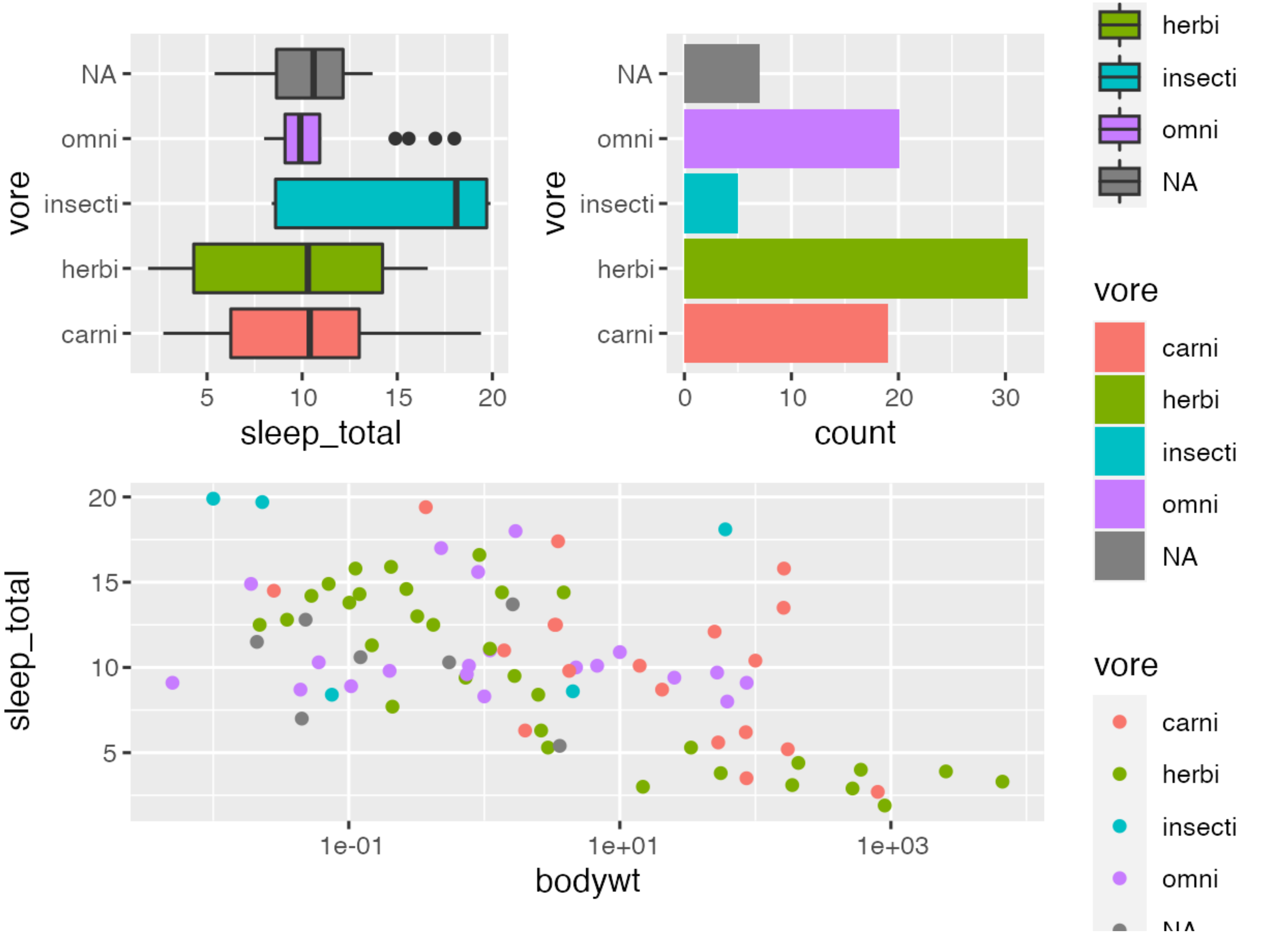


BEYOND GGPLOT2

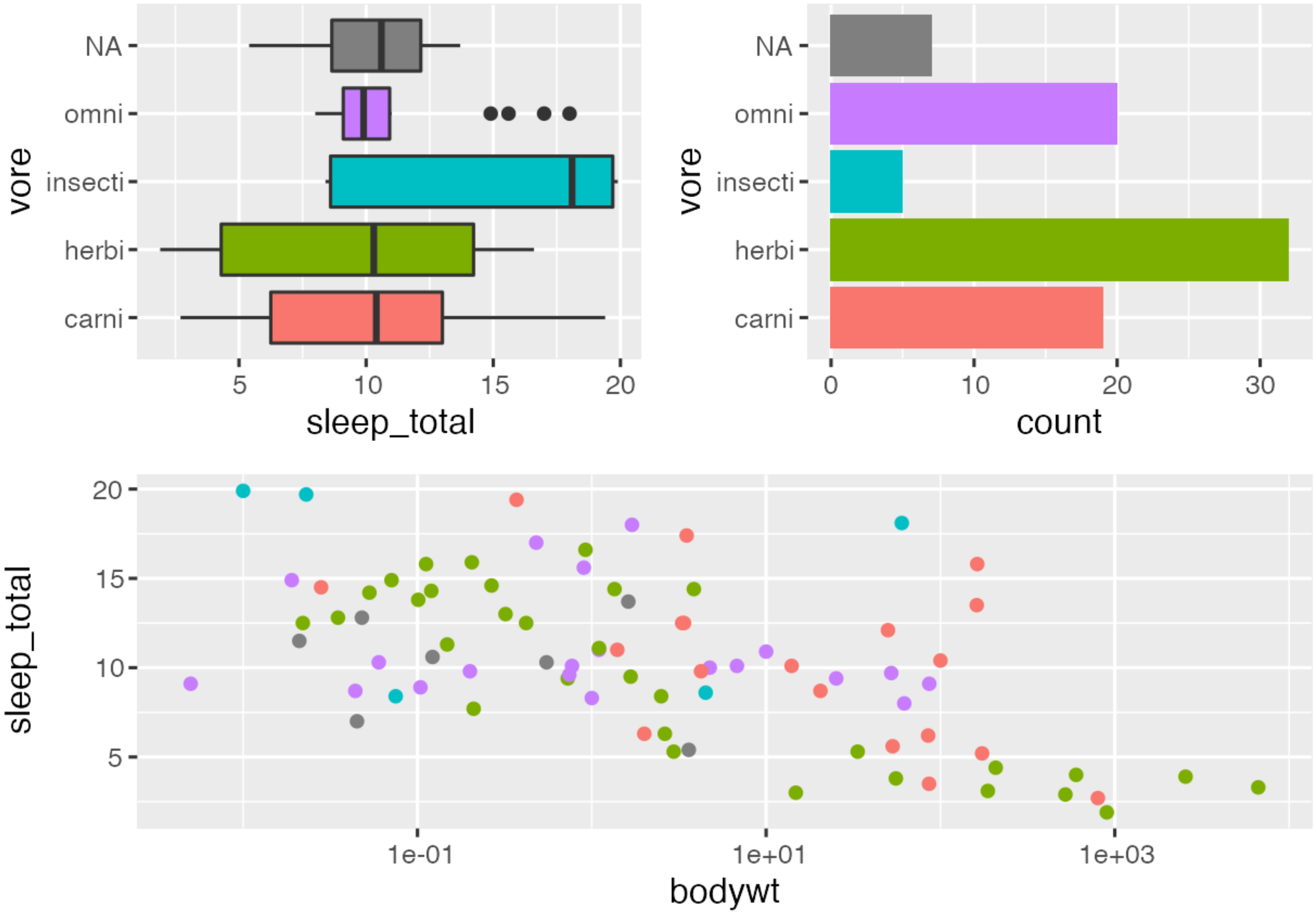
```
(p1 | p2) /
   p3
```



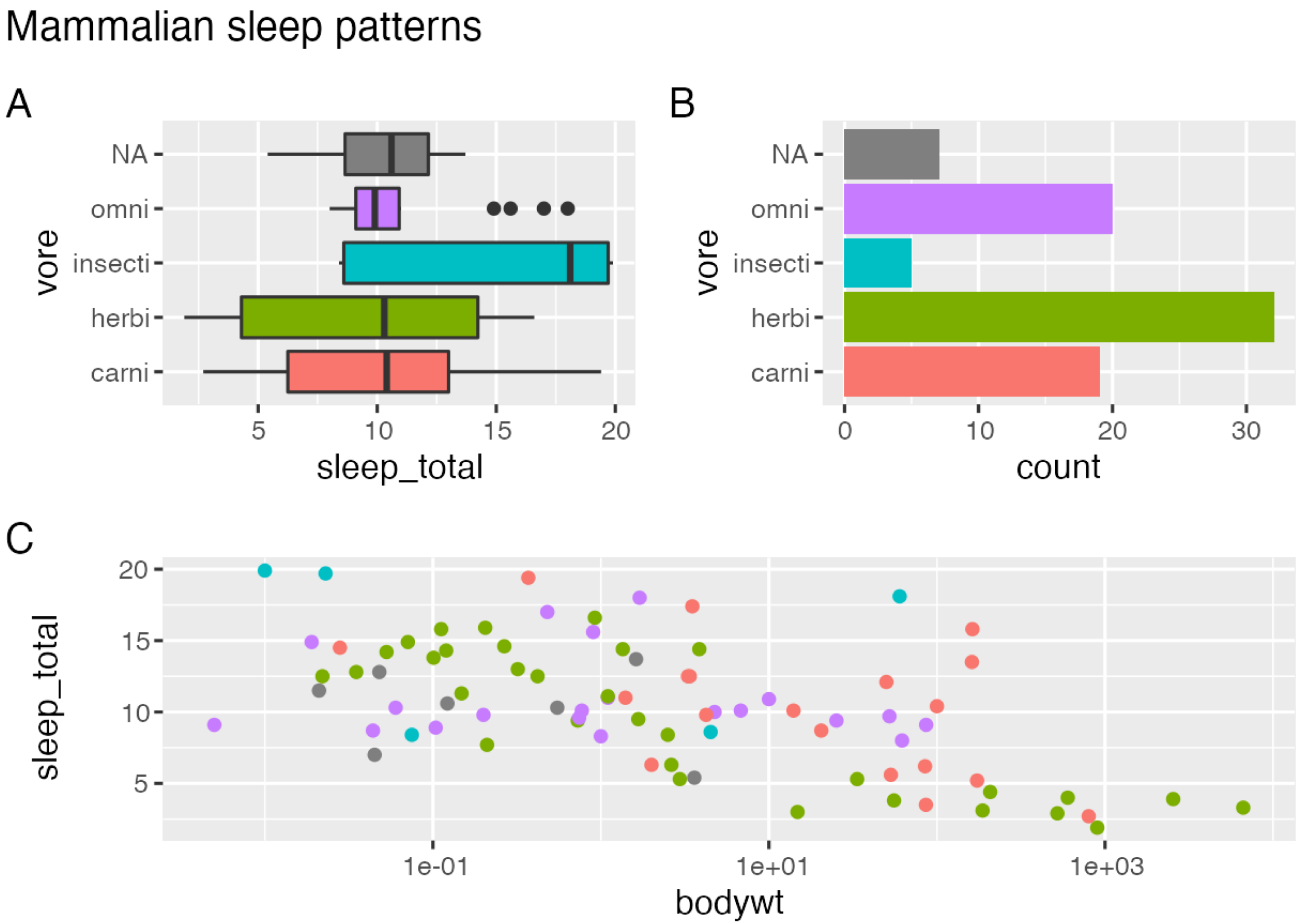
```
p_all ← (p1 | p2) /
            p3

p_all +
  plot_layout(
    guides = 'collect'
  )
```



```
p_all &
  theme(
    legend.position = 'none'
  )
```

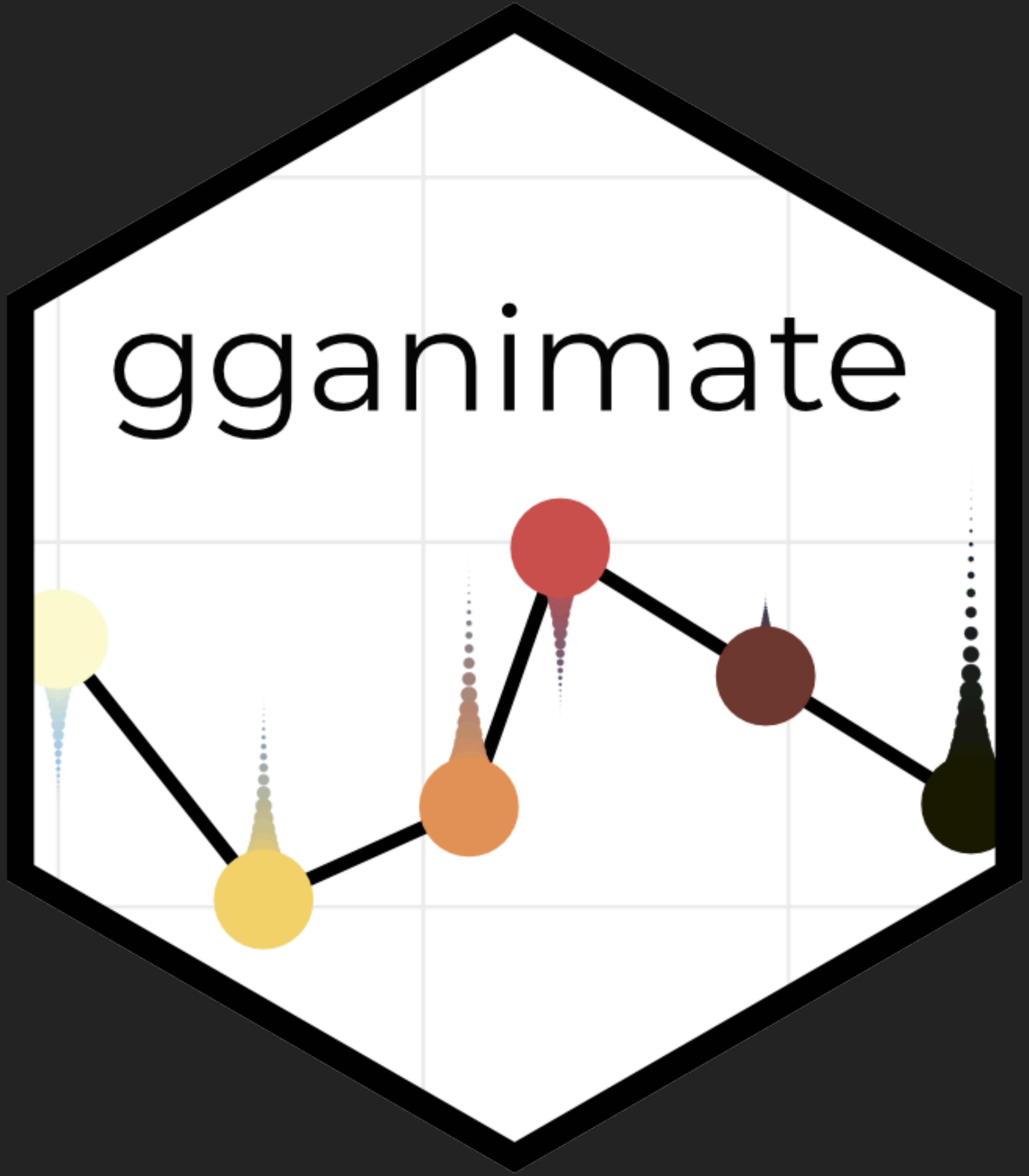


```
p_all + plot_annotation(
  title = 'Mammalian sleep
patterns',
  tag_levels = 'A'
)
```

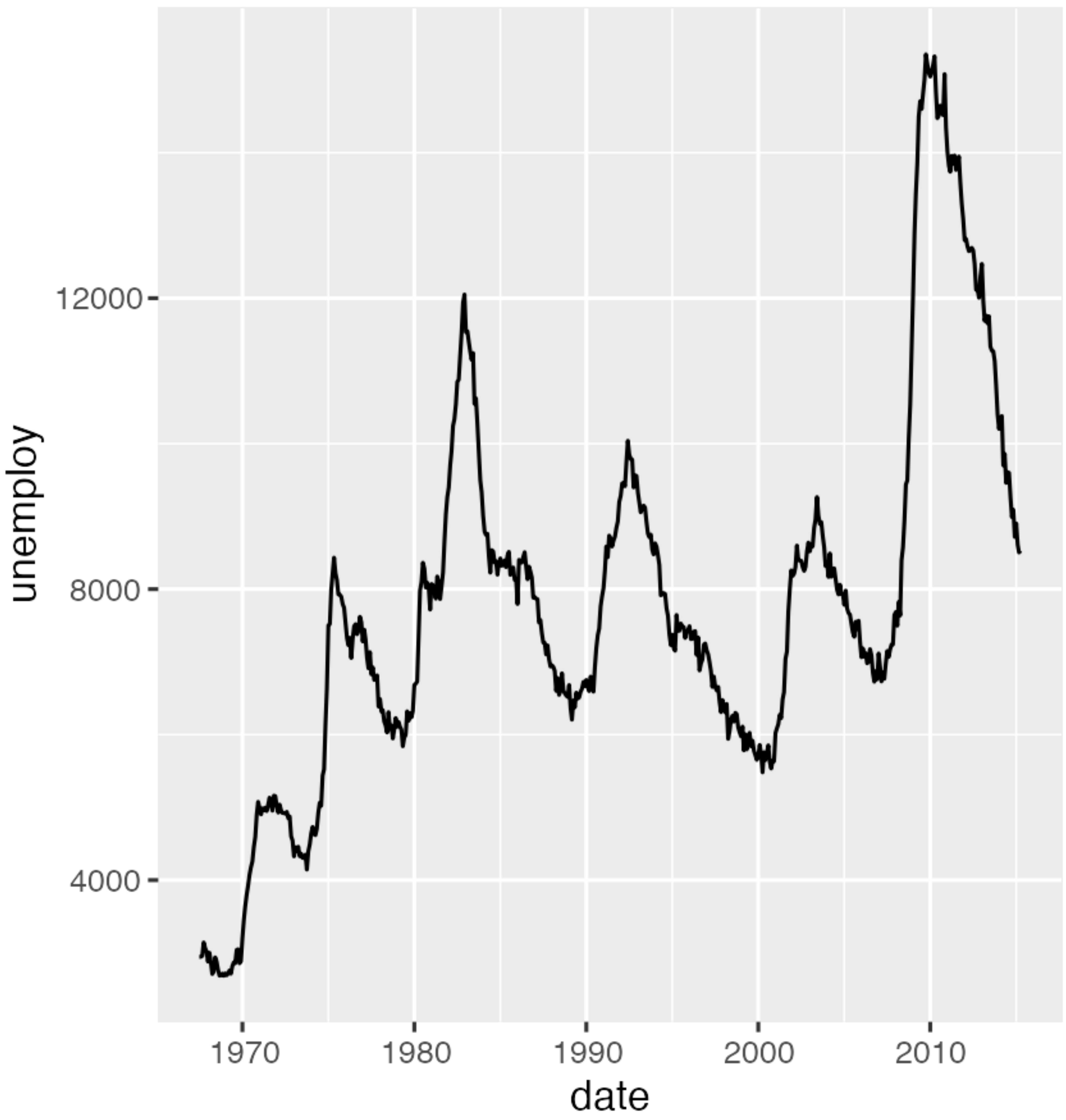


ANIMATION

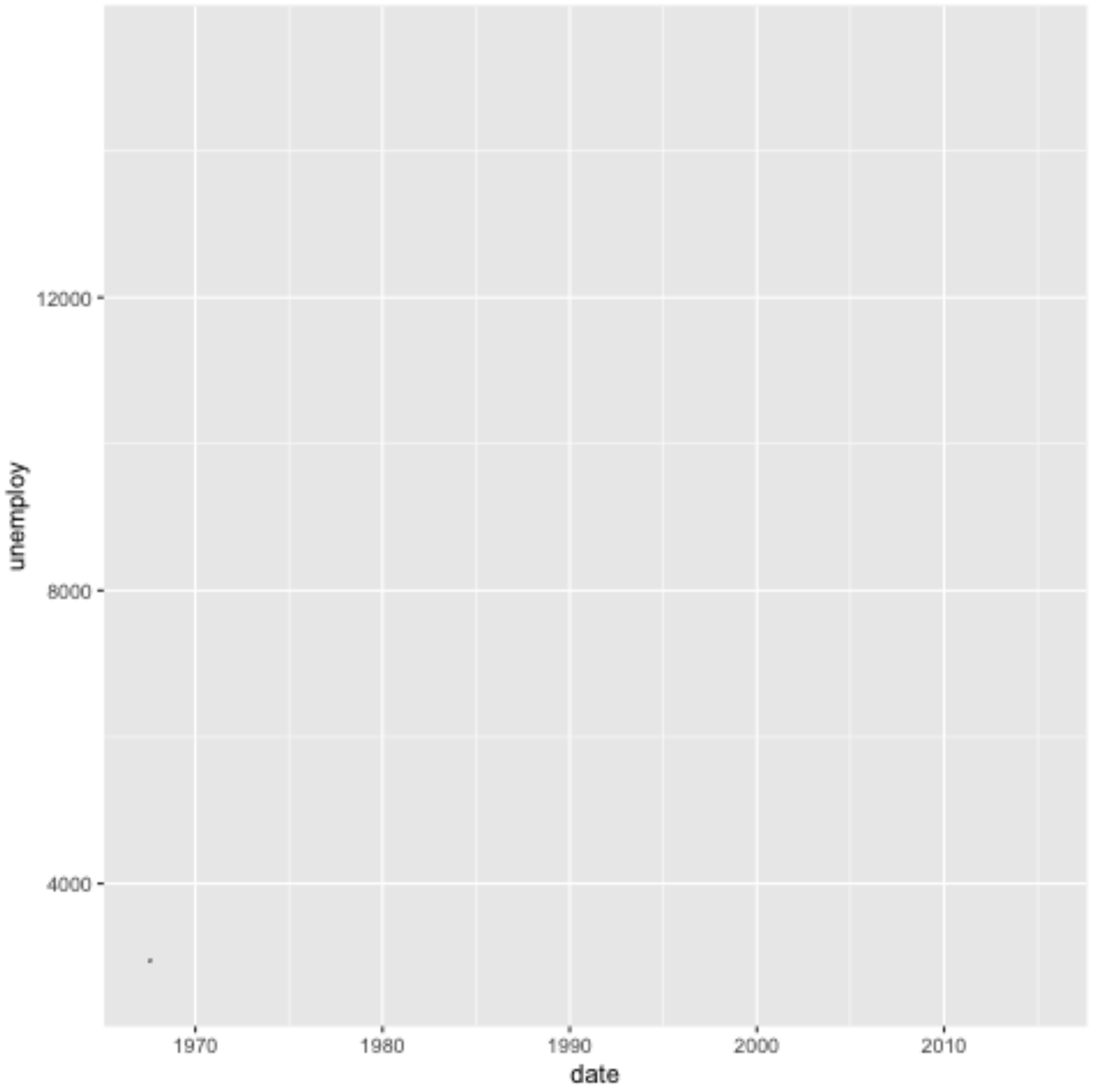
- Story telling and attention grabbing
- May turn boring people off (some valid critique)
- Our eyes are drawn to movement and our mind thinks in motion
- Many ways to make animations – gganimate may be the only one doing it the “grammar” way



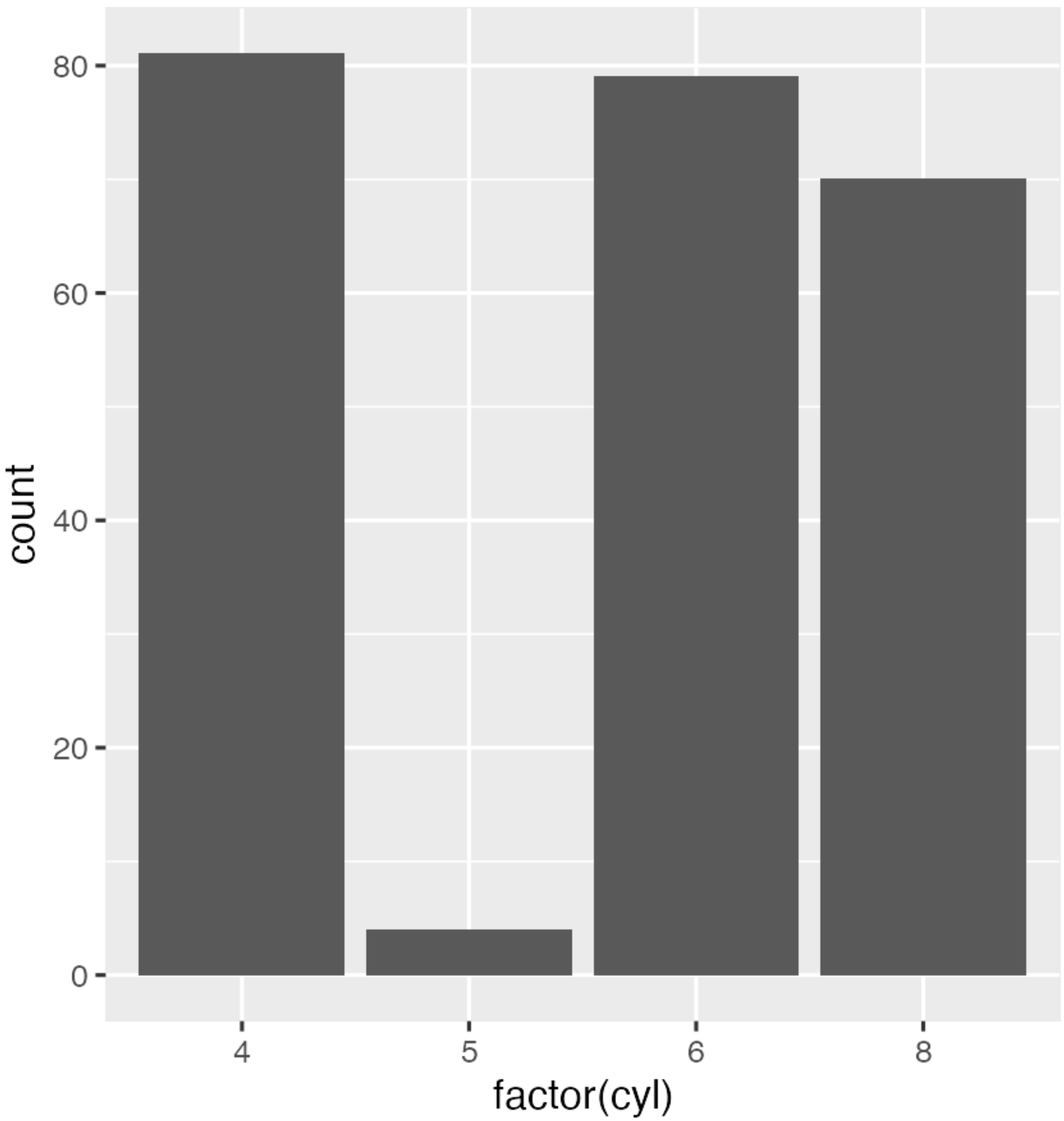
```
ggplot(economics) +
  geom_line(aes(x = date, y = unemploy))
```



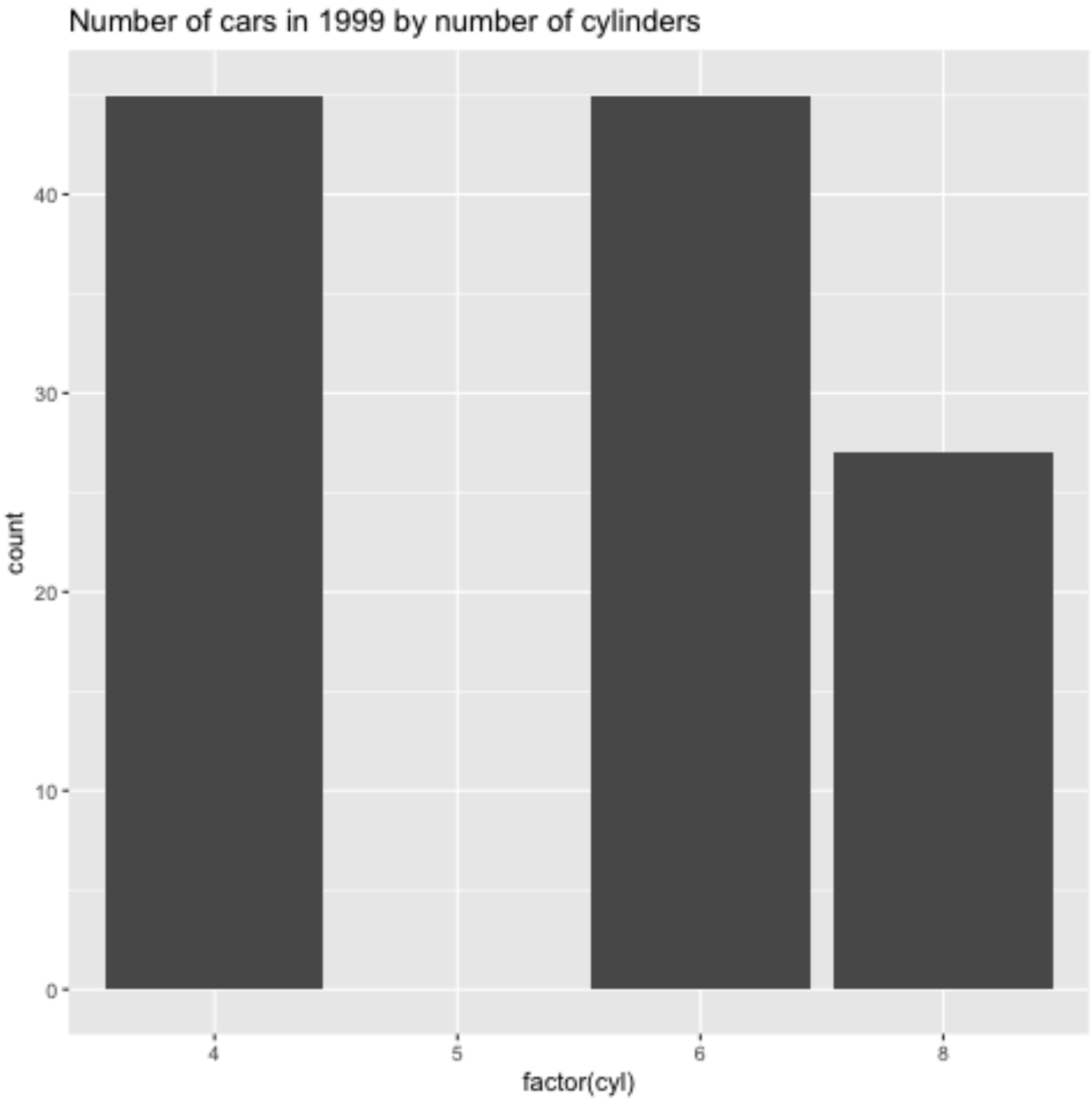
```
library(gganimate)

ggplot(economics) +
  geom_line(aes(x = date, y = unemploy)) +
  transition_reveal(along = date)
```



```
ggplot(mpg) +
  geom_bar(aes(x = factor(cyl)))
```

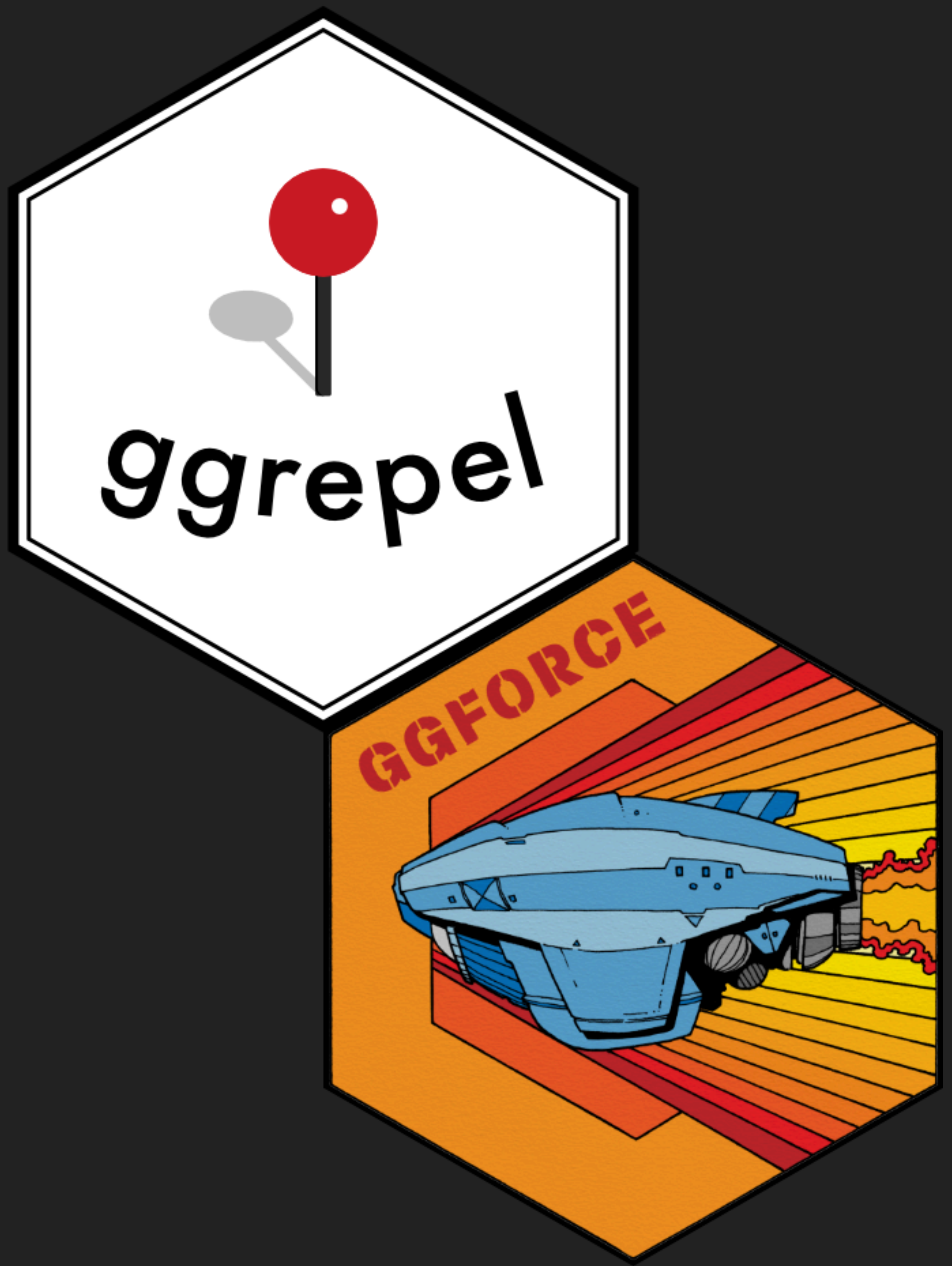


```
ggplot(mpg) +
  geom_bar(aes(x = factor(cyl))) +
  labs(title = 'Number of cars in
{closest_state} by number of cylinders') +
  transition_states(states = year) +
  enter_grow() +
  exit_fade()
```



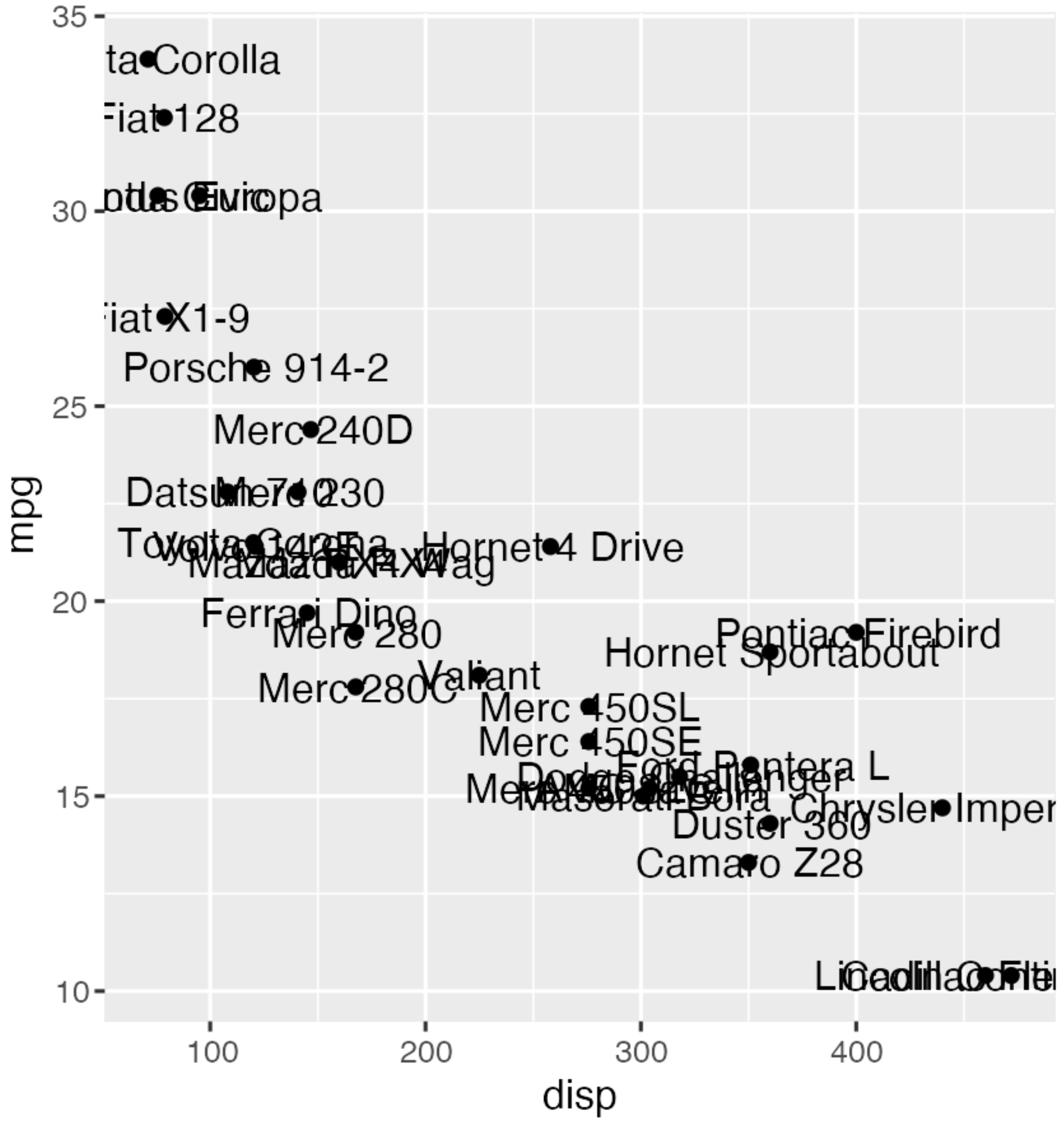
ANNOTATION

- You can't tell a story without words
- This has often been relegated to post processing in e.g. Adobe Illustrator
- Recent improvements means that most can be done with code
- Two key packages:
  - ggrepel
  - ggforce

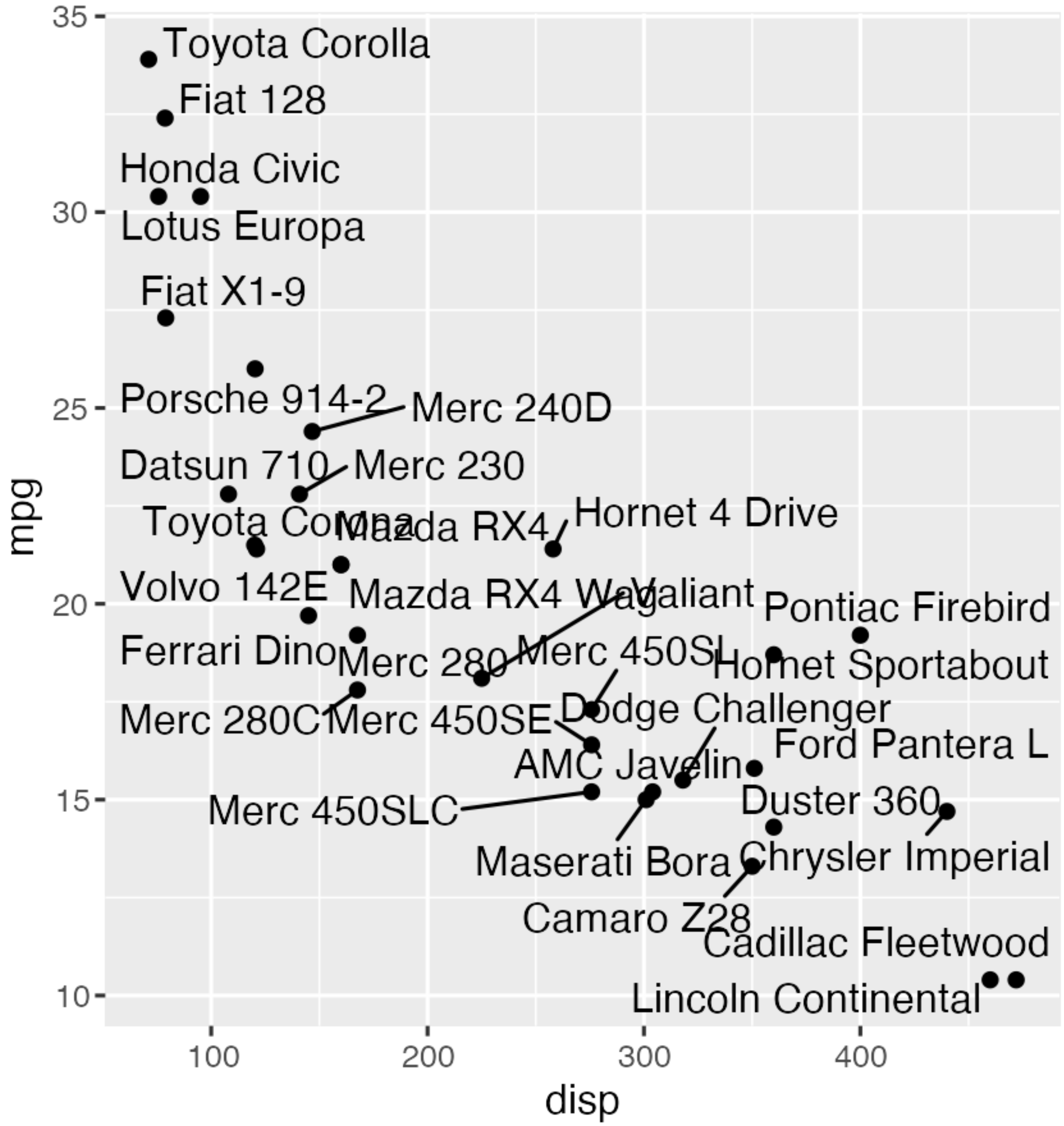


BEYOND GGPLOT2

```
ggplot(mtcars, aes(x = disp, y = mpg)) +
  geom_point() +
  geom_text(aes(label = row.names(mtcars)))
```

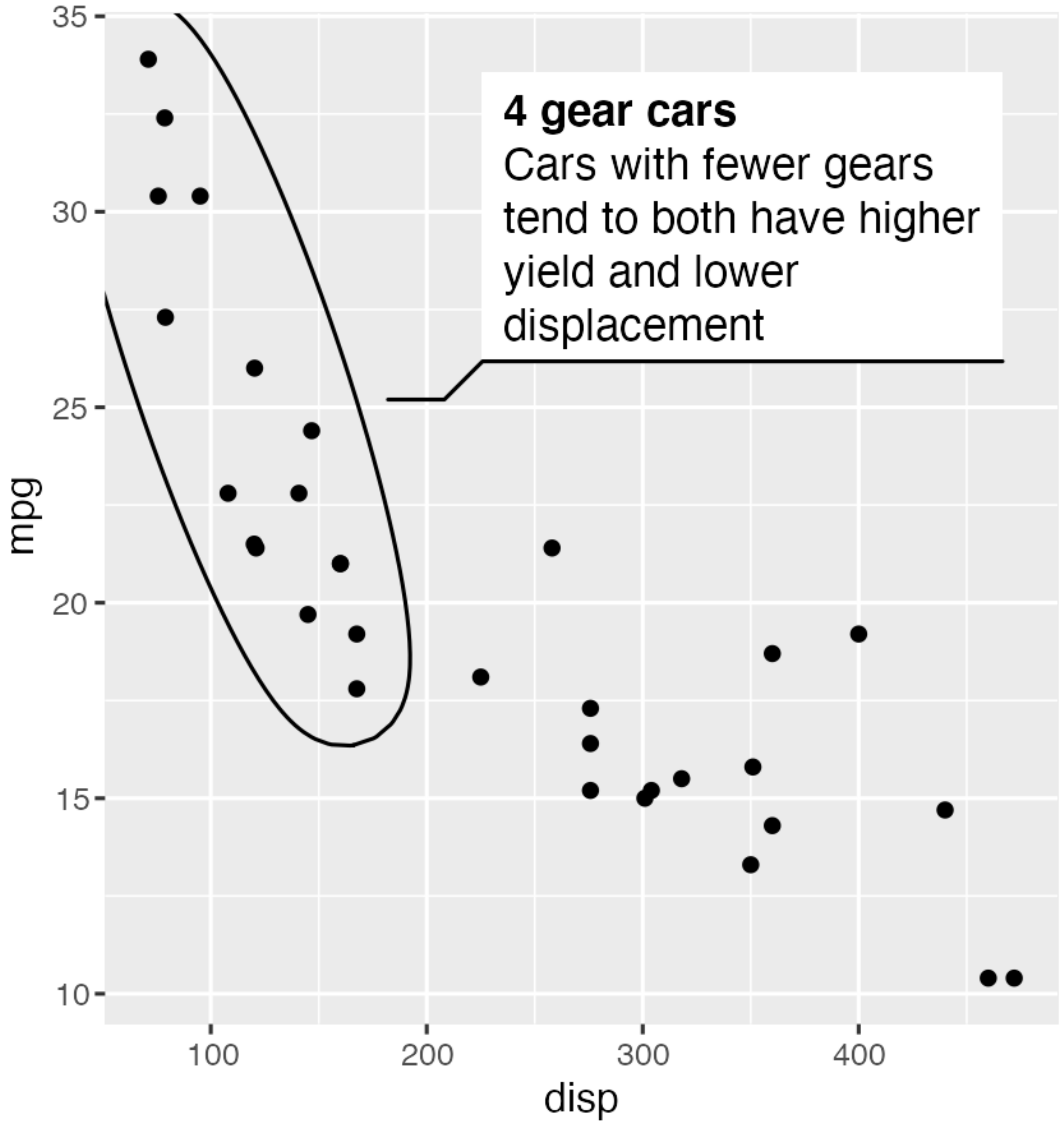


```
library(ggrepel)

ggplot(mtcars, aes(x = disp, y = mpg)) +
  geom_point() +
  geom_text_repel(
    aes(label = row.names(mtcars))
  )
```



BEYOND GGPLOT2

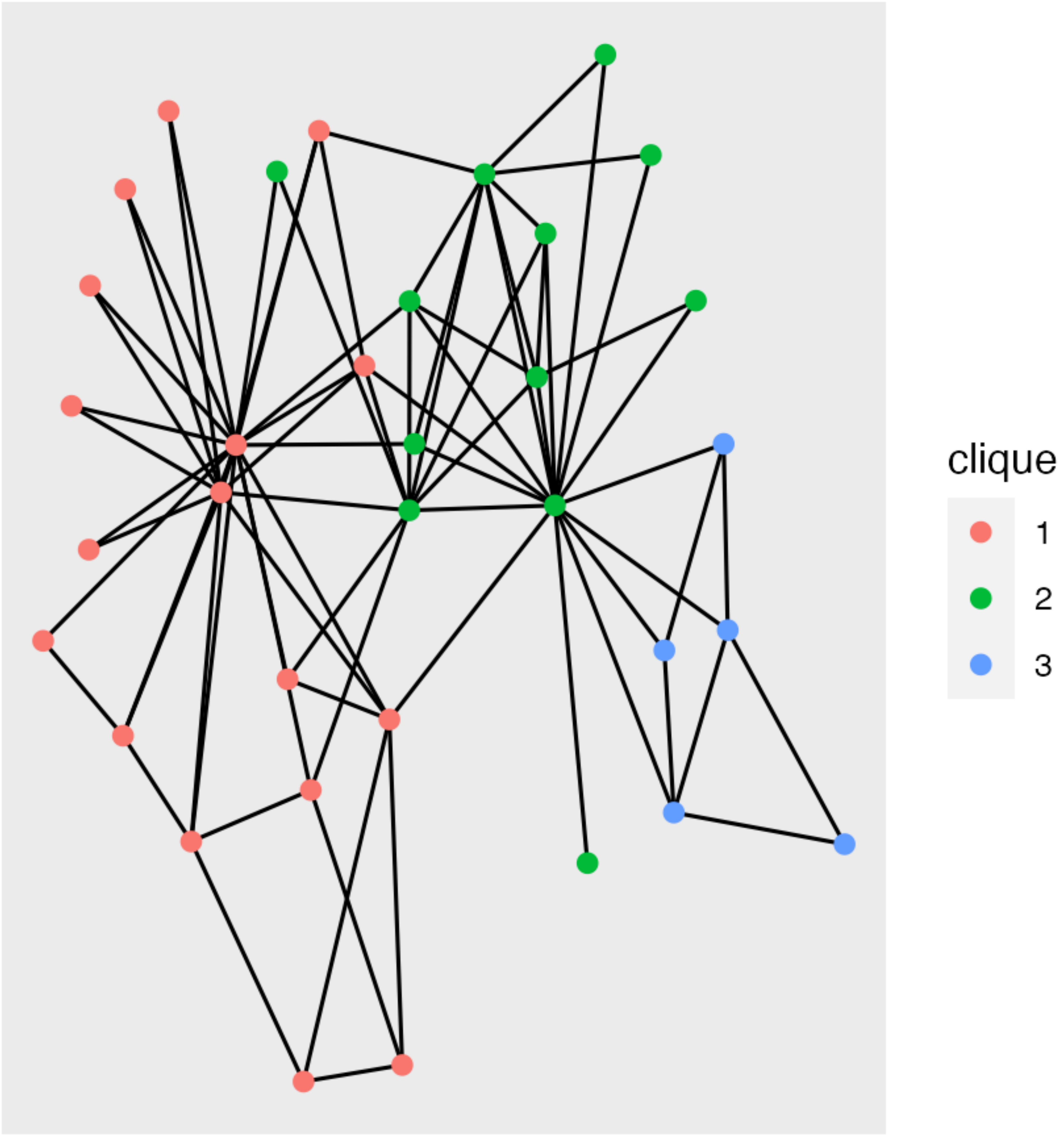
```r
library(ggforce)

ggplot(mtcars, aes(x = disp, y = mpg)) +
  geom_point() +
  geom_mark_ellipse(
    aes(filter = gear == 4,
        label = '4 gear cars',
        description = 'Cars with fewer gears
tend to both have higher yield and lower
displacement'))
```



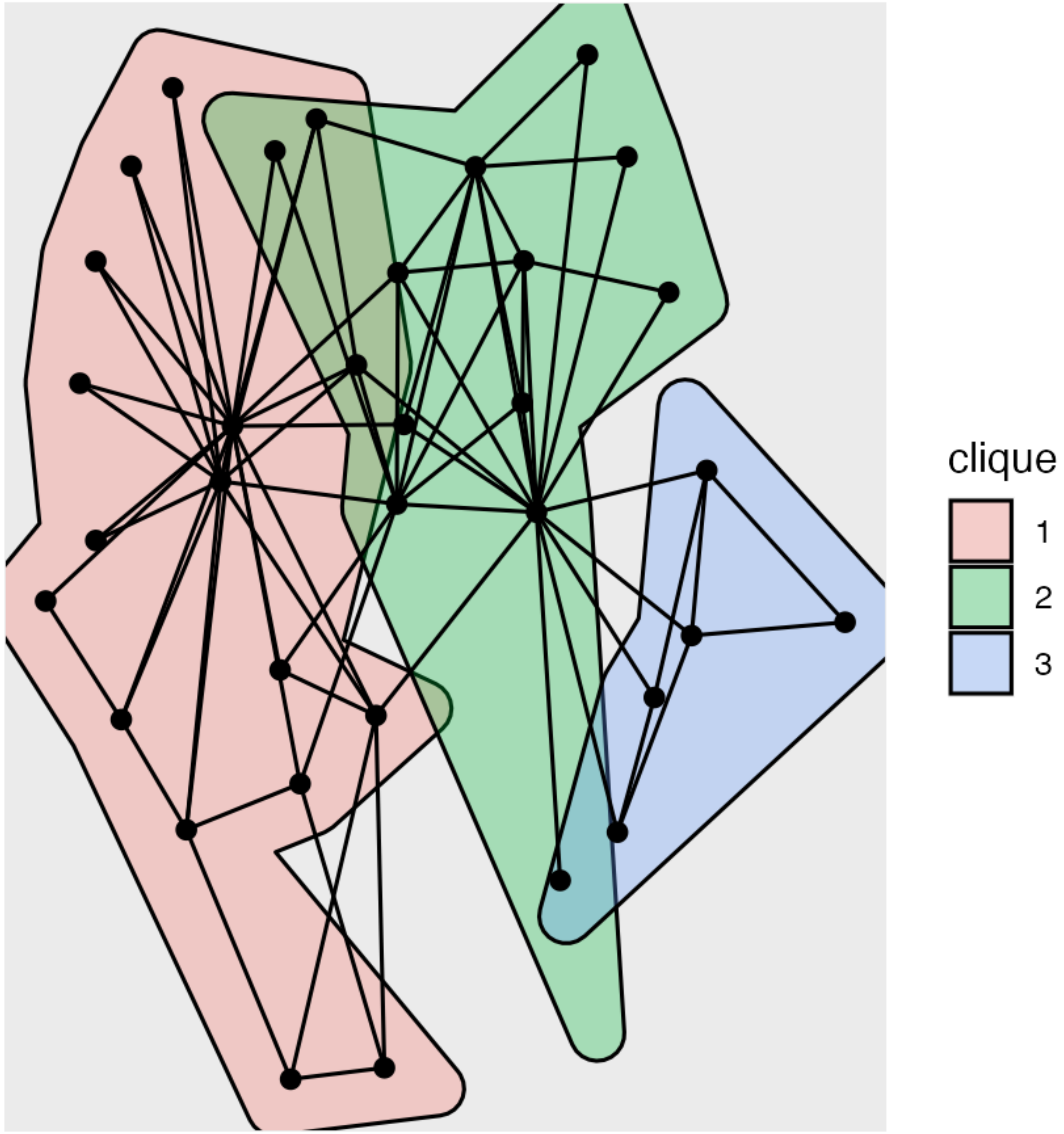
NETWORKS

- ggplot2 was build for tabular data
  - An API decision, not a grammar constraint
- Multiple packages tries to add network capabilities to ggplot2
  - GGally
  - ggnetwork
  - geomnet
  - ggtree
  - ggdag
  - ggraph

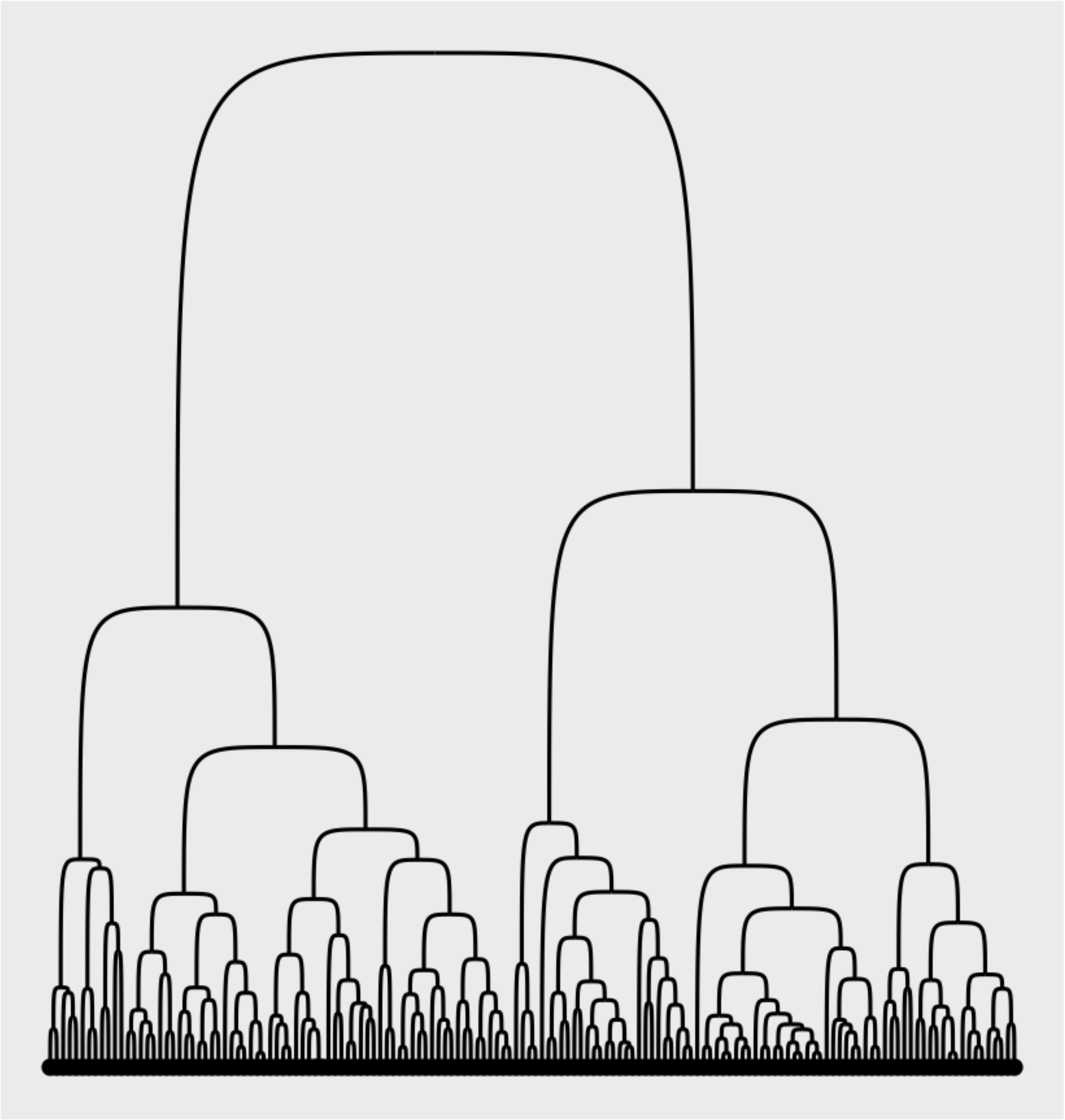
```
library(ggraph)
library(tidygraph)

graph ← create_notable('zachary') %>%
  mutate(clique = as.factor(group_infomap()))

ggraph(graph) +
  geom_edge_link() +
  geom_node_point(aes(colour = clique),
                  size = 2)
```



```
ggraph(graph) +
  geom_mark_hull(aes(x, y, fill = clique)) +
  geom_edge_link() +
  geom_node_point(size = 2)
```

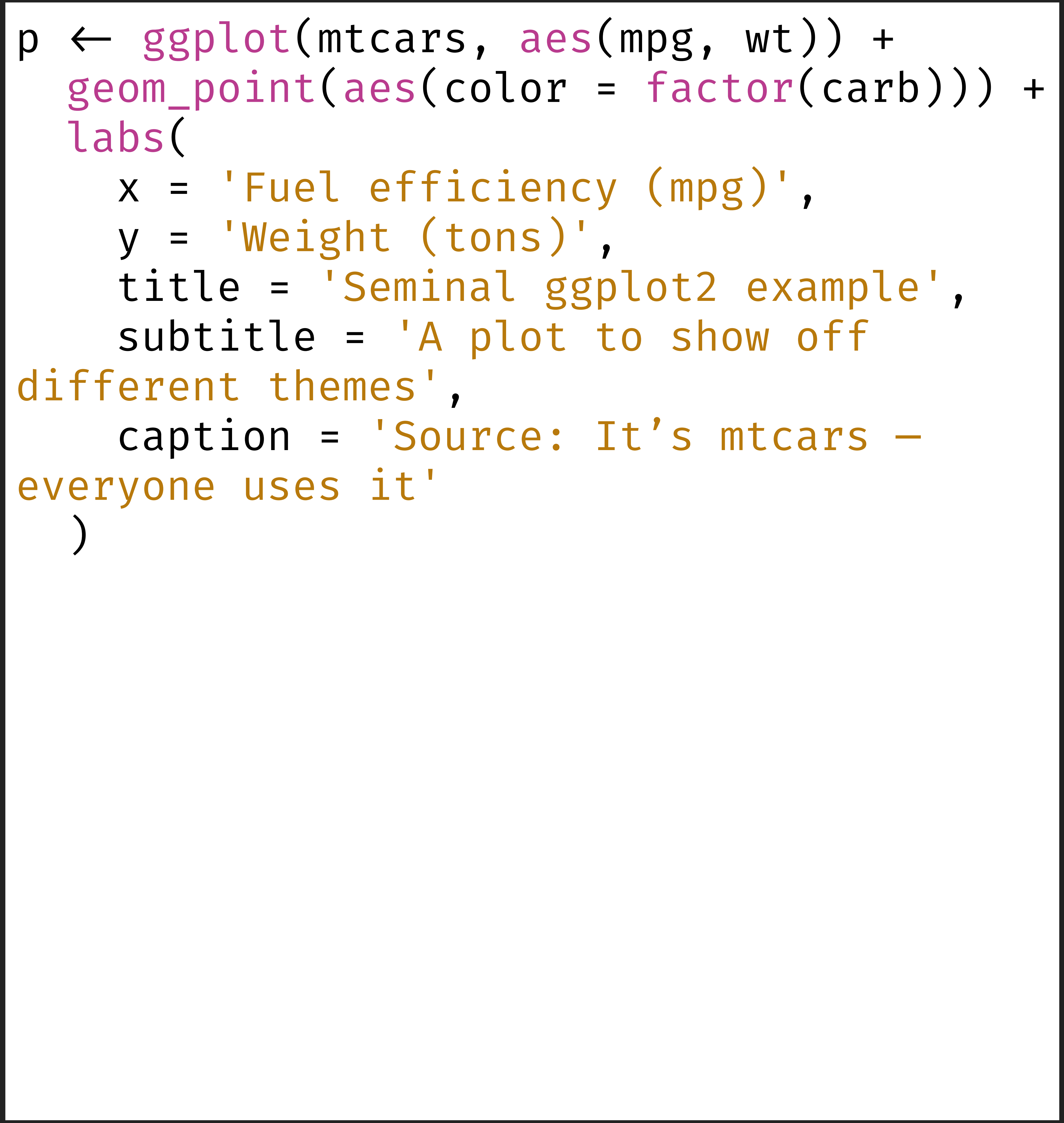


```
iris_clust ← hclust(dist(iris[, 1:4]))

ggraph(iris_clust) +
  geom_edge_bend() +
  geom_node_point(aes(filter = leaf))
```



LOOKS

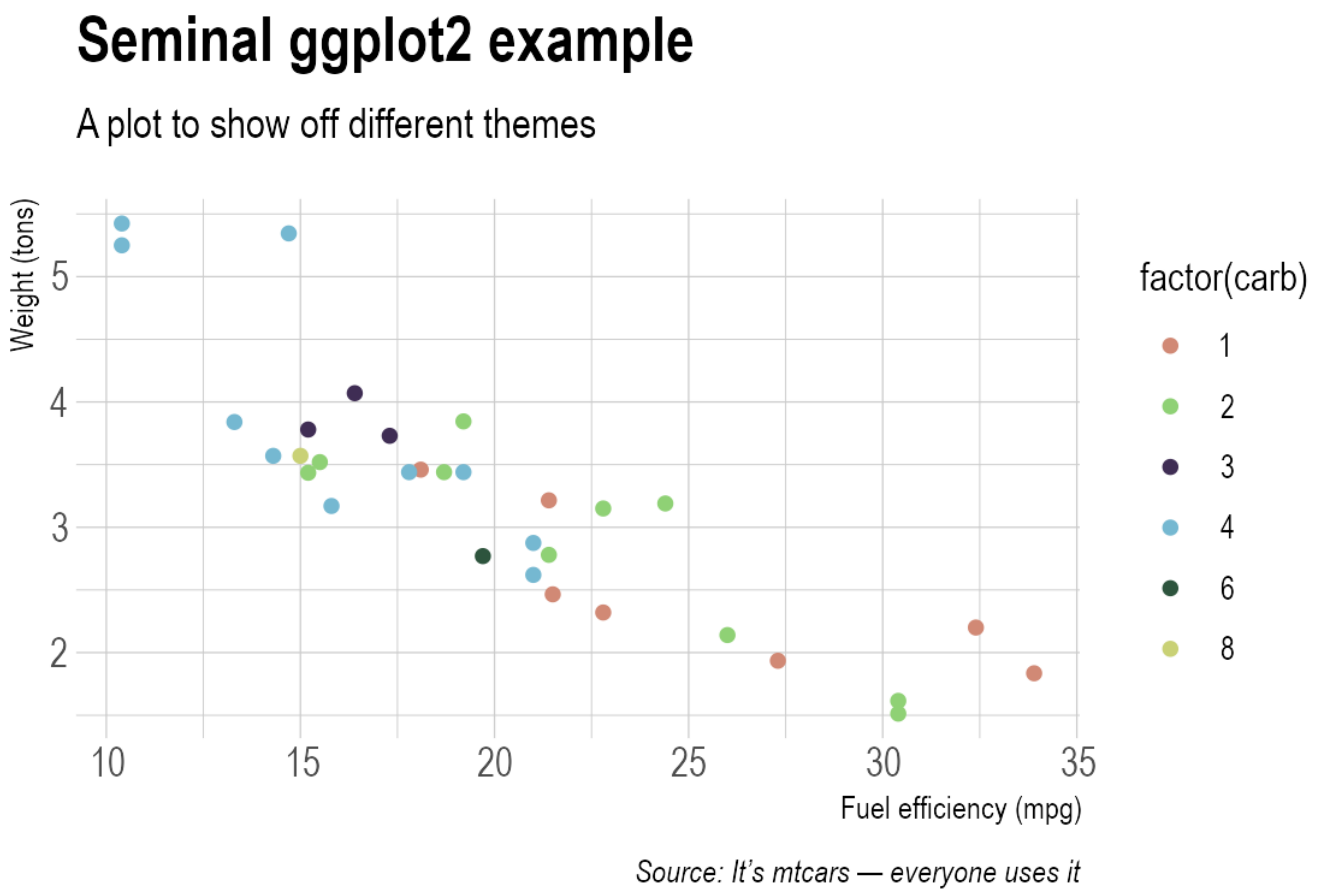
- Theming is one of the (programmatically) easiest things to extend
- Lots of options:
  - ggthemes
  - tvthemes
  - ggtech
  - ggthemr
  - hrbrthemes
- Often coupled with colour scales
- Don’t go too crazy unless warranted



```
p ← ggplot(mtcars, aes(mpg, wt)) +
  geom_point(aes(color = factor(carb))) +
  labs(
    x = 'Fuel efficiency (mpg)',
    y = 'Weight (tons)',
    title = 'Seminal ggplot2 example',
    subtitle = 'A plot to show off
different themes',
    caption = 'Source: It’s mtcars —
everyone uses it'
  )
```

BEYOND GGPLOT2

```r
library(hrbrthemes)

p +
  scale_colour_ipsum() +
  theme_ipsum()
```

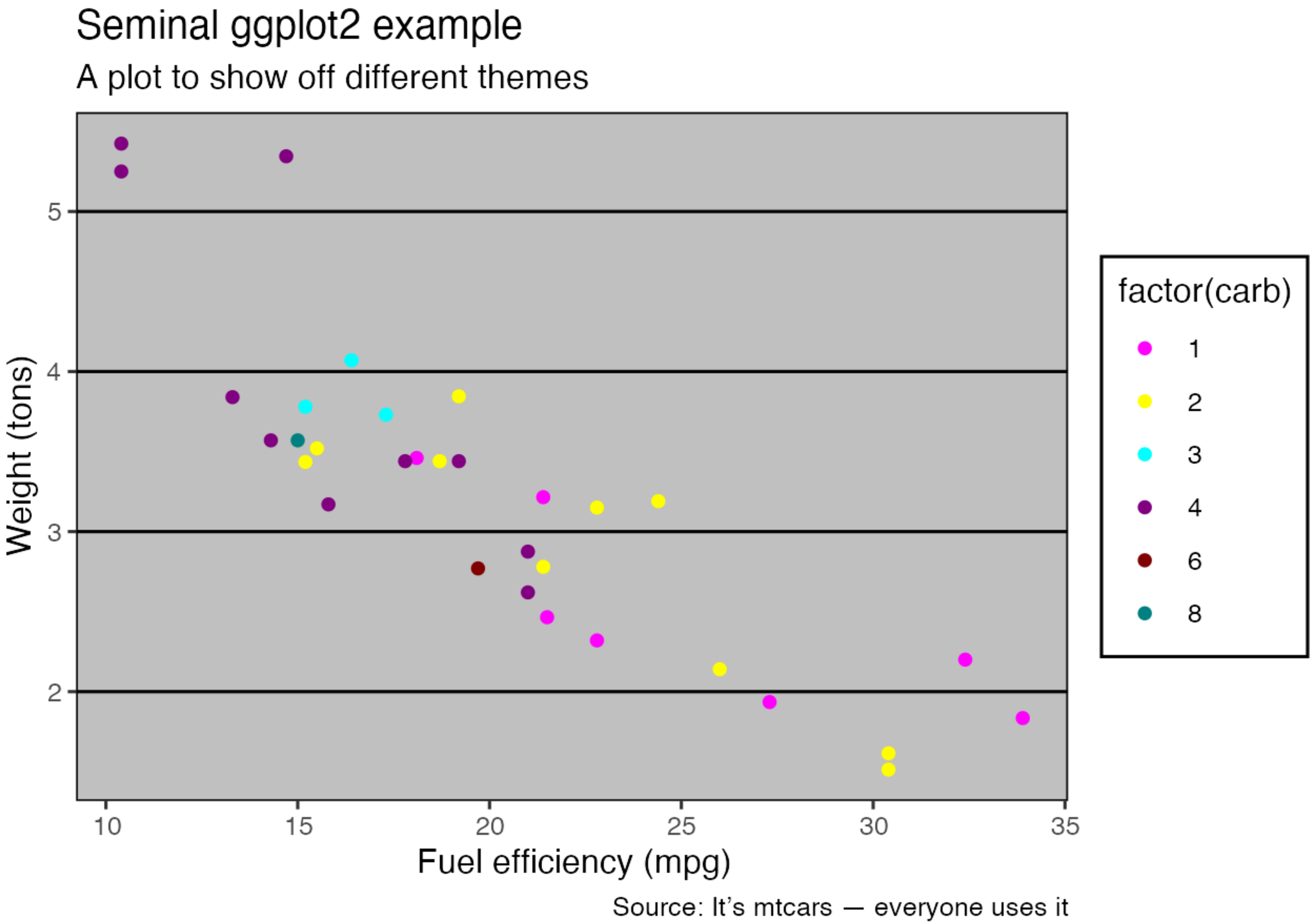


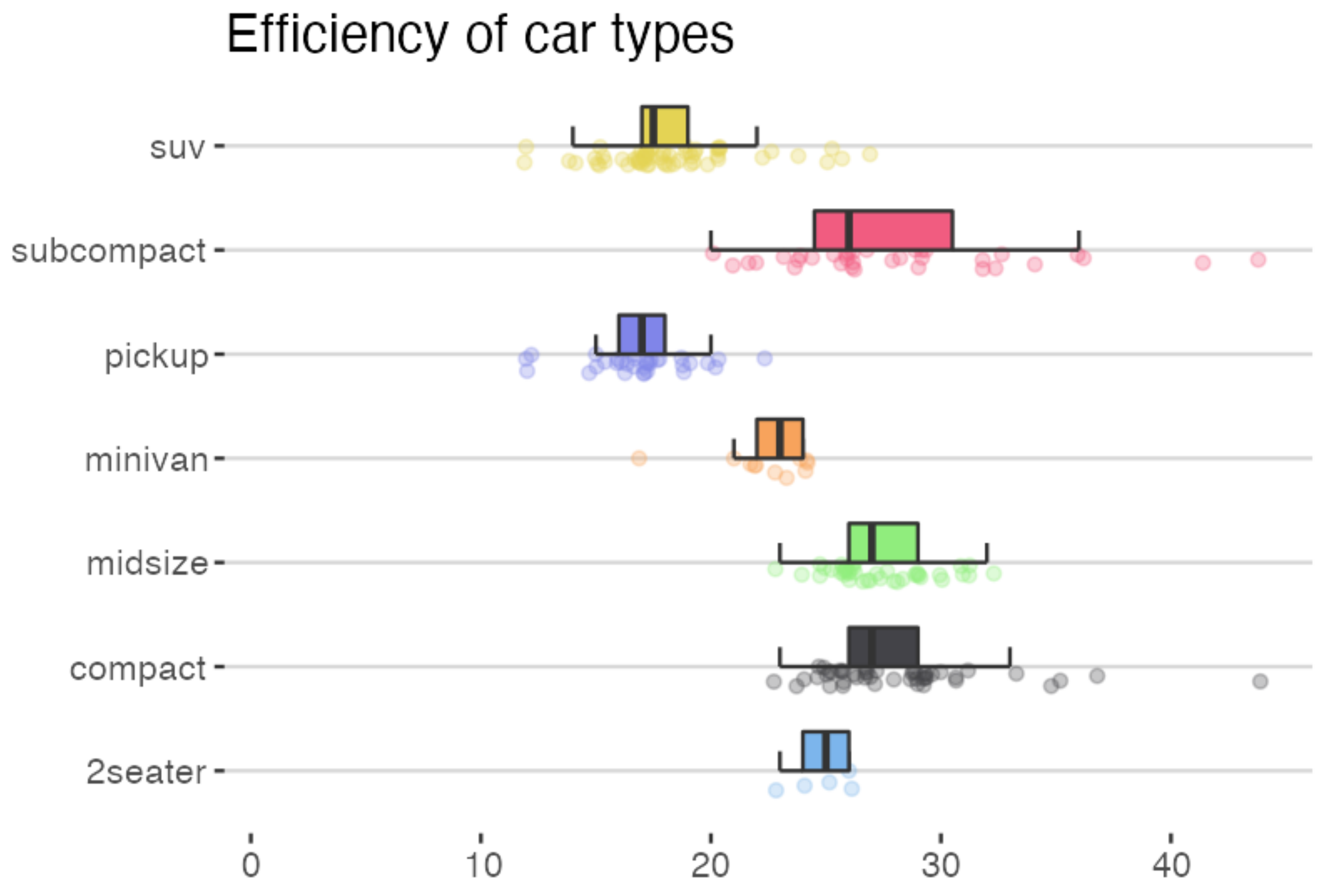
Seminal ggplot2 example

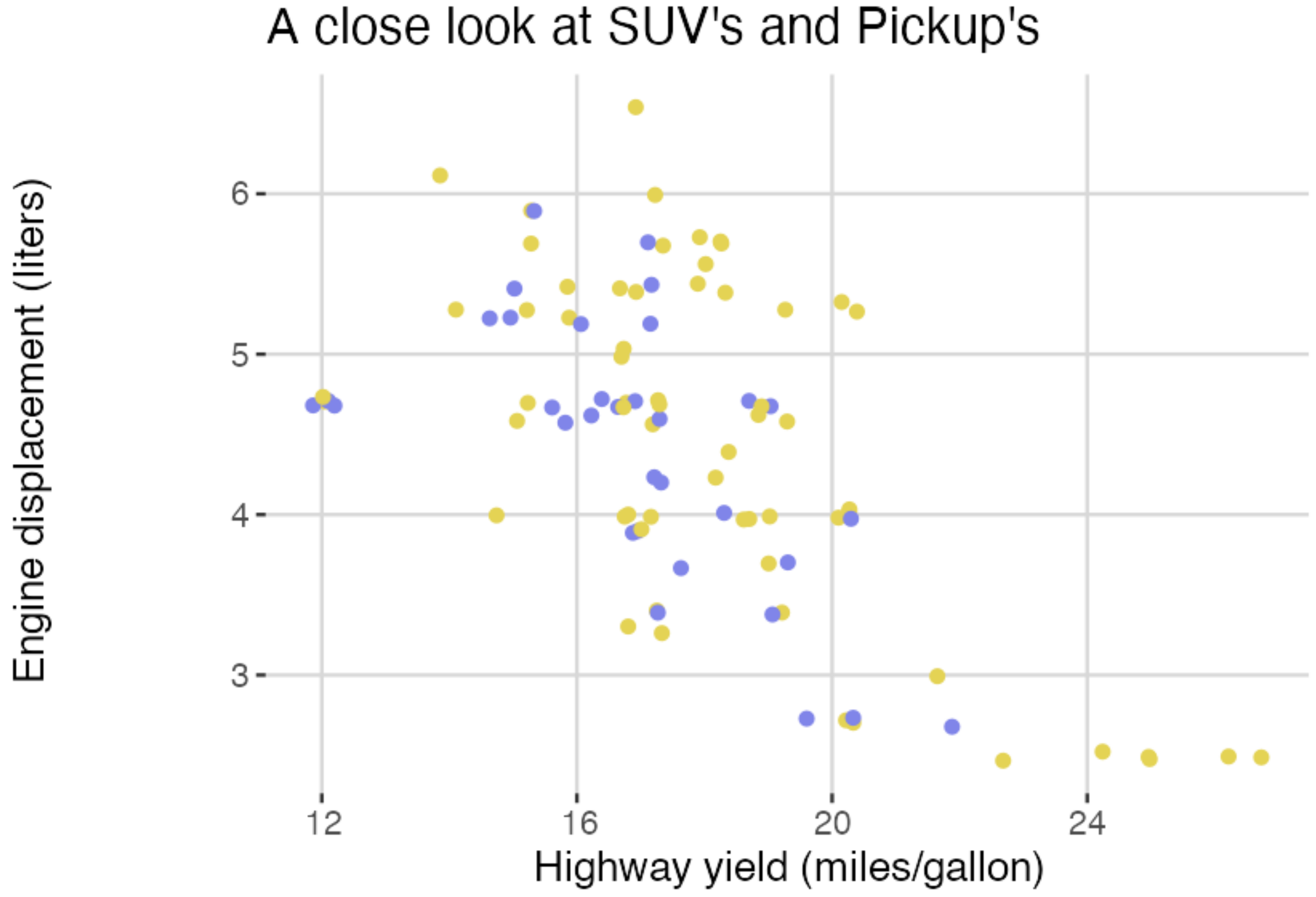
A plot to show off different themes

Source: It's mtcars — everyone uses it

```
library(ggthemes)

p +
  scale_colour_excel() +
  theme_excel()
```





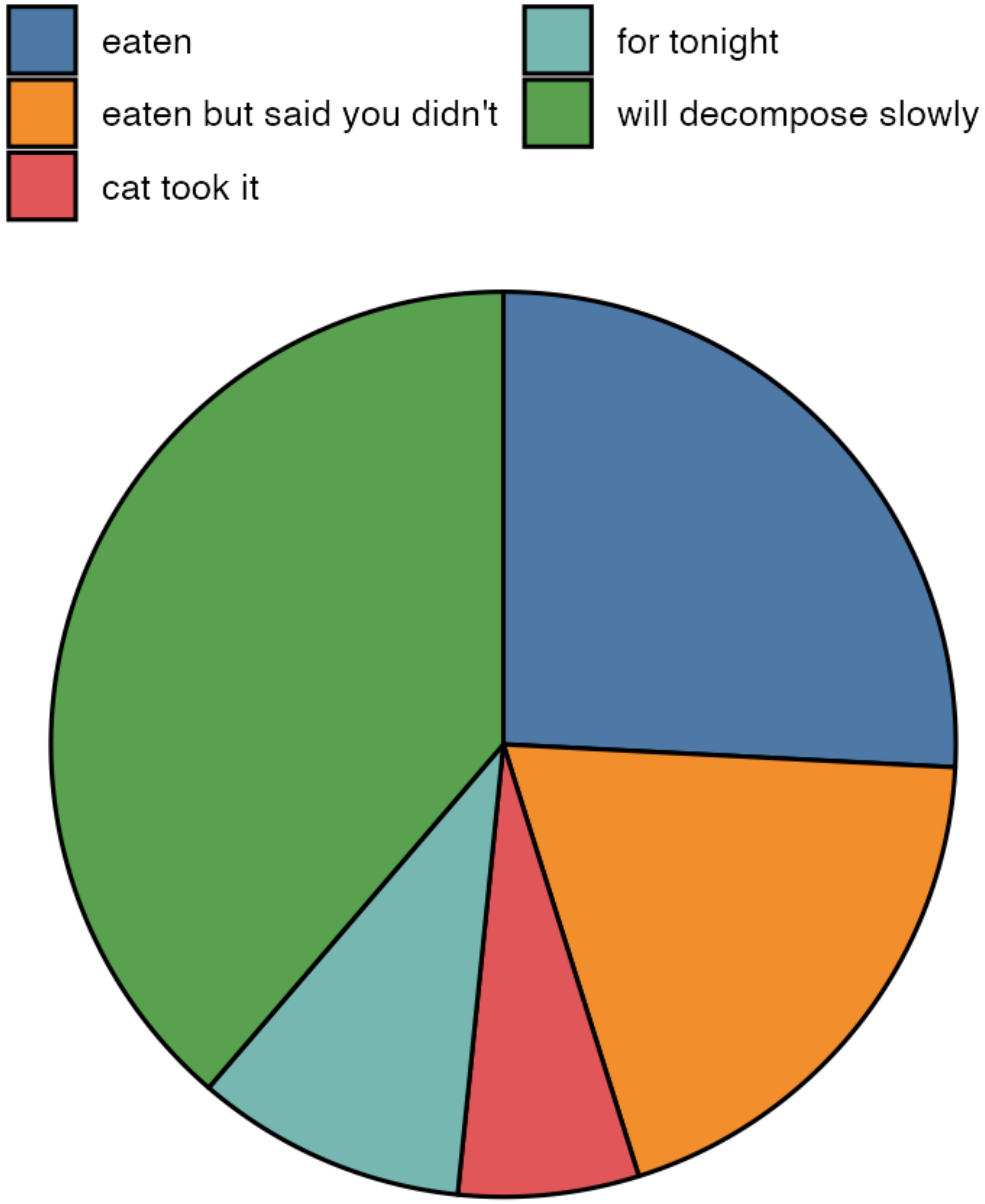


PART 4:

DRAWING ANYTHING

THINKING IN GRAMMAR

- Grammar of graphics is not just for constructing graphs – use it to deconstruct as well!
- The grammar is ambiguous and that's okay
- If all else fails:
    - Everything is just points, lines, polygons, and text
    - If faceting fails, use patchwork
    - Illustrator is ok

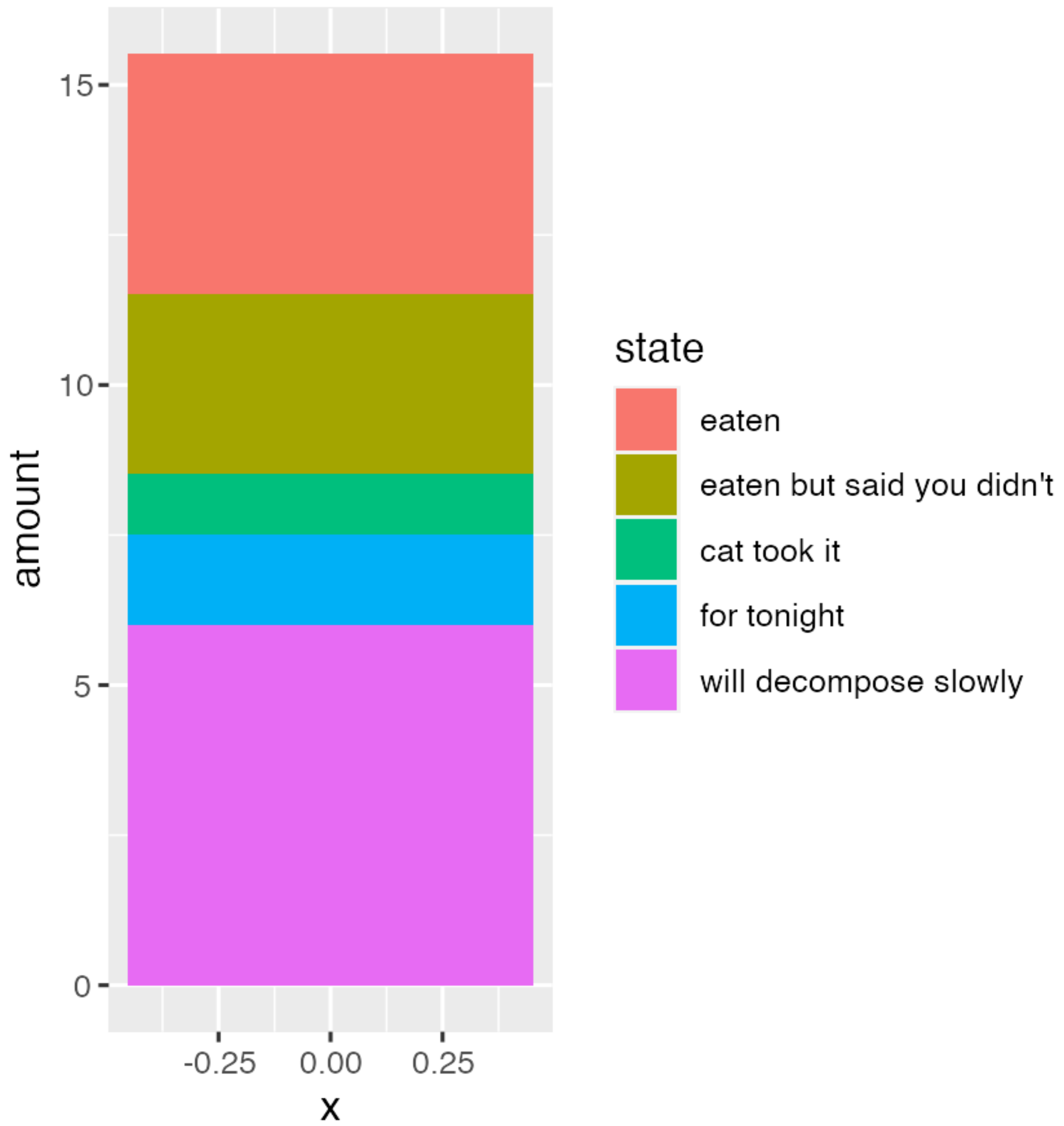


WHAT'S IN A PIE

- How does a pie chart fit into the grammar of graphics?
- What are:
  - the geoms and stats?
  - the coord?
  - the data?

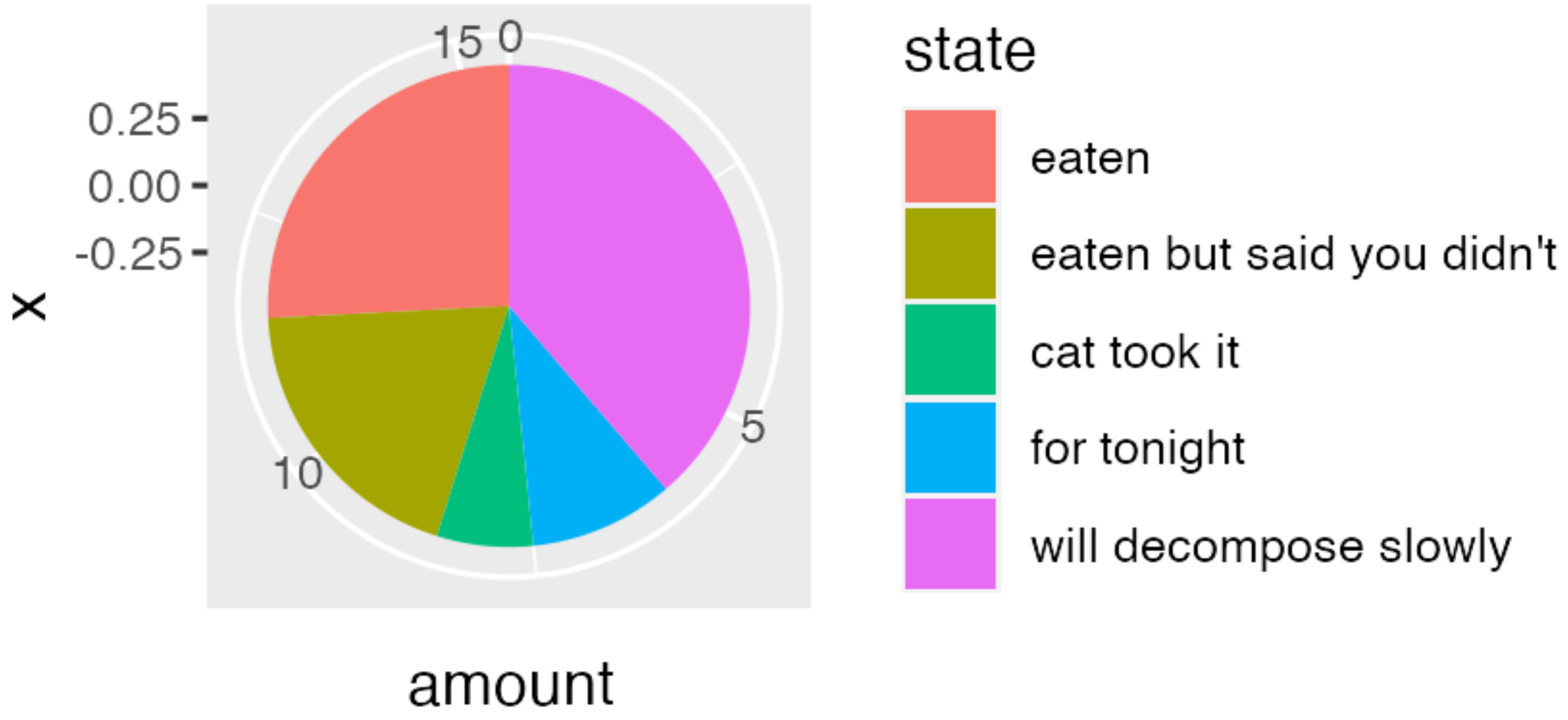
DRAWING ANYTHING

```r
states ← c(
  'eaten',
  "eaten but said you didn\'t",
  'cat took it',
  'for tonight',
  'will decompose slowly'
)
pie ← data.frame(
  state = factor(states, levels = states),
  amount = c(4, 3, 1, 1.5, 6),
  stringsAsFactors = FALSE
)

ggplot(pie) +
  geom_col(
    aes(x = 0, y = amount, fill = state)
  )
```



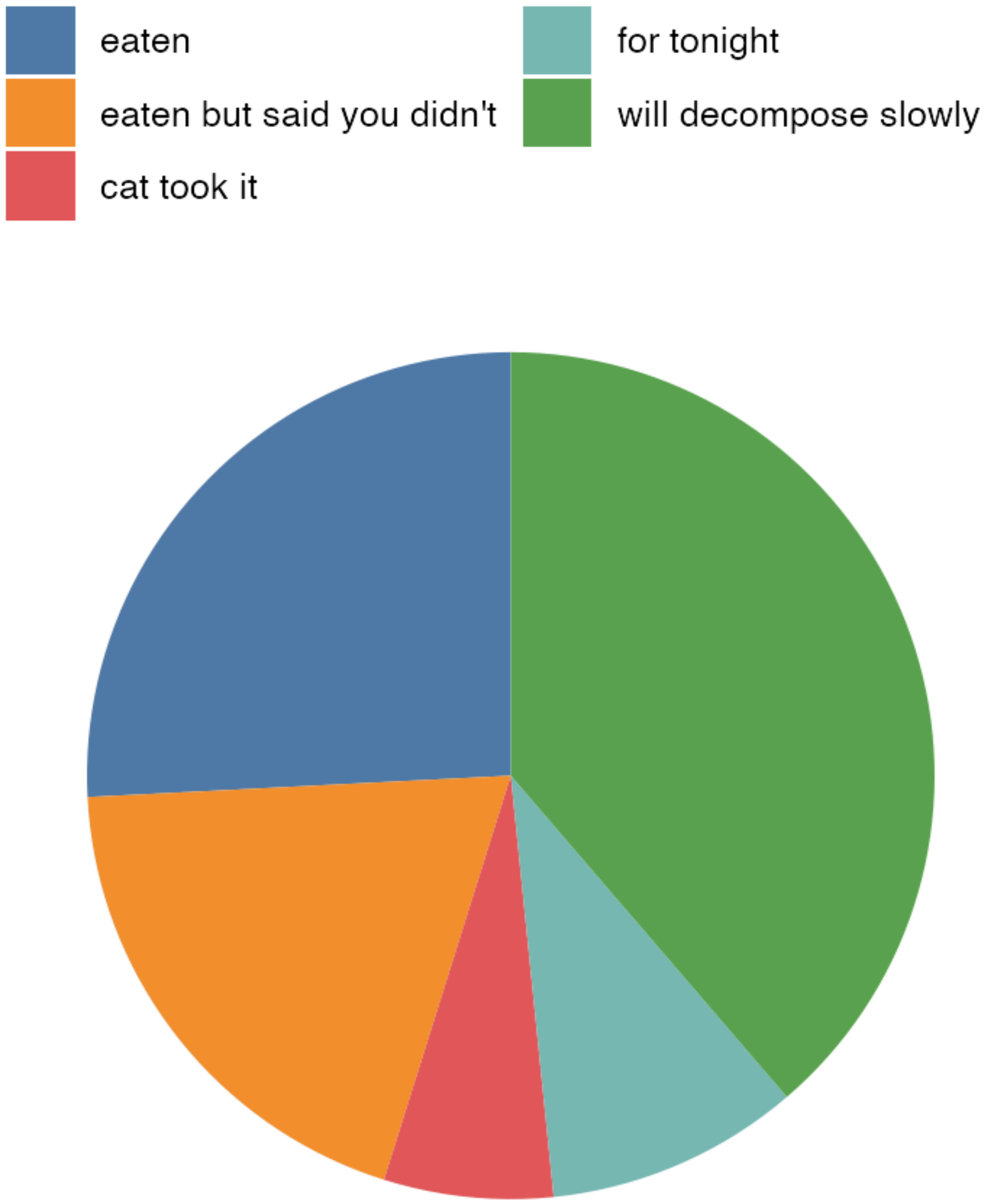
DRAWING ANYTHING

```
ggplot(pie) +
  geom_col(aes(
    x = 0,
    y = amount,
    fill = state)) +
  coord_polar(theta = 'y')
```



DRAWING ANYTHING

```
ggplot(pie) +
  geom_col(aes(
    x = 0,
    y = amount,
    fill = state)) +
  coord_polar(theta = 'y') +
  scale_fill_tableau(
    name = NULL,
    guide = guide_legend(ncol = 2)
  ) +
  theme_void() +
  theme(legend.position = 'top',
        legend.justification = 'left')
```



> A PIE CHART IS A STACKED BAR CHART IN A POLAR COORDINATE SYSTEM…

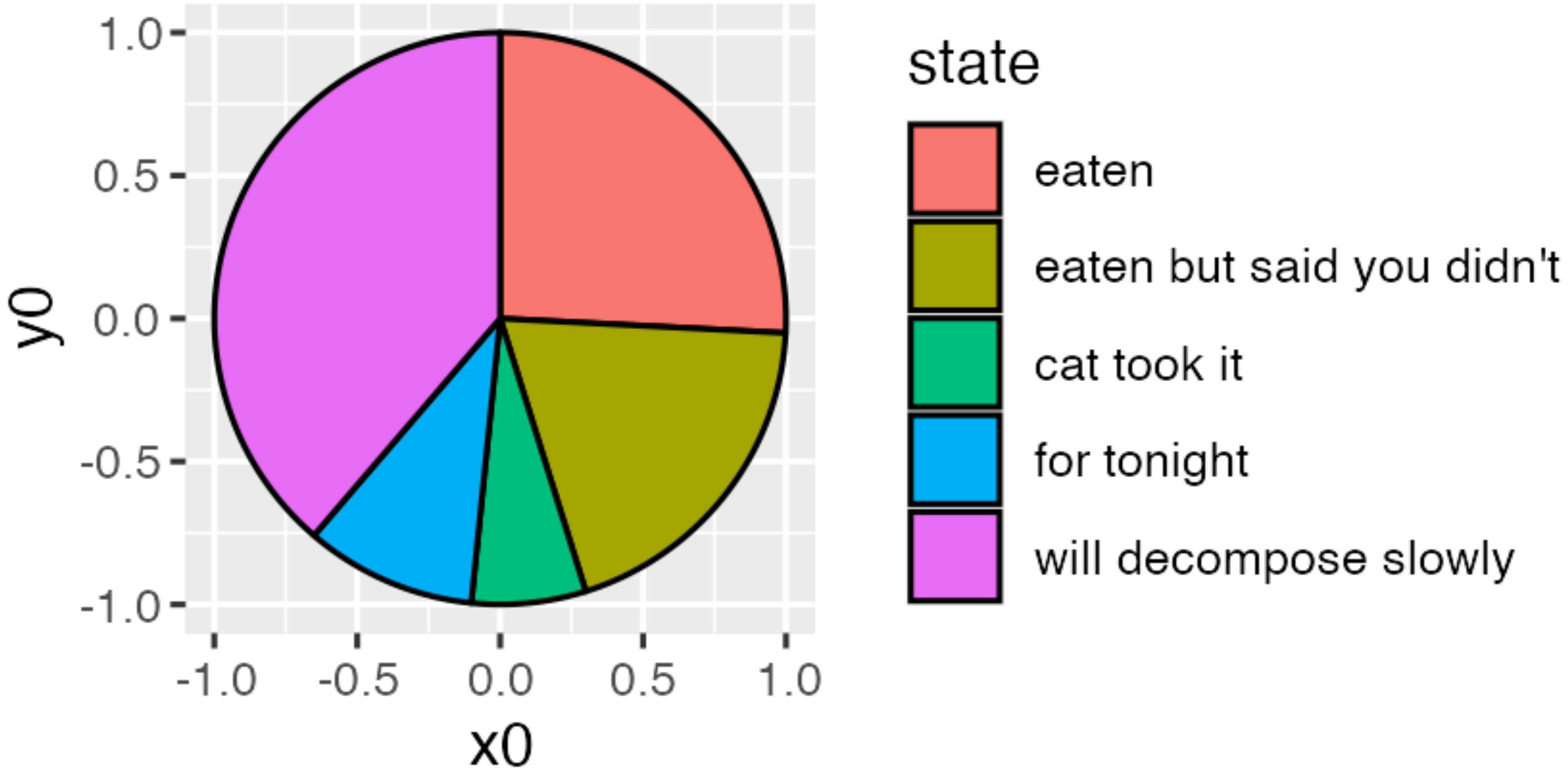
Leland Wilkinson (paraphrased)

> A PIE CHART IS ALSO A WEDGE GEOM COUPLED WITH A STAT FOR CALCULATING RADIANS

Me

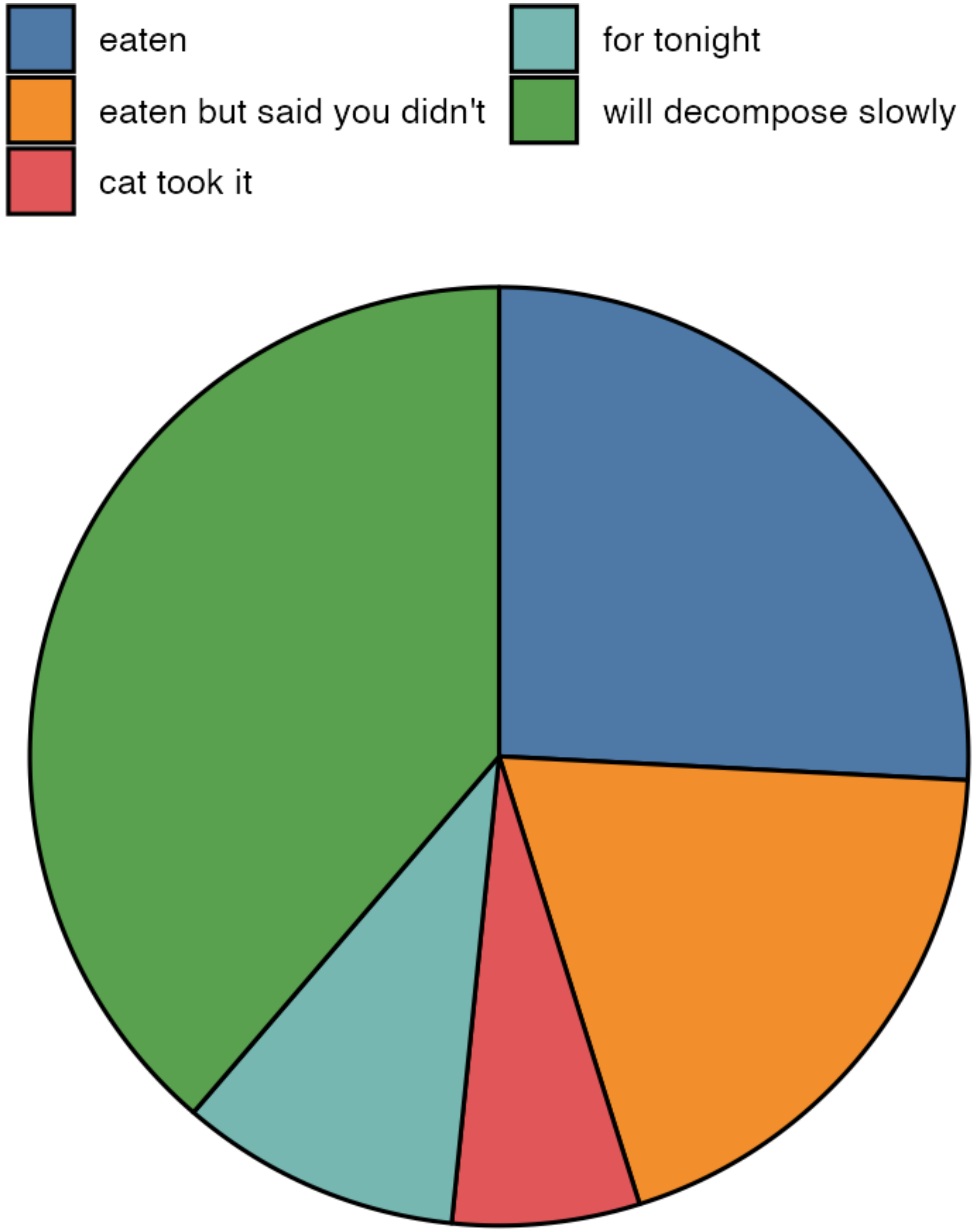
DRAWING ANYTHING

```
ggplot(pie) +
  geom_arc_bar(
    aes(x0 = 0, y0 = 0, r0 = 0, r = 1,
        amount = amount, fill = state),
    stat = ‘pie'
  ) +
  coord_fixed()
```



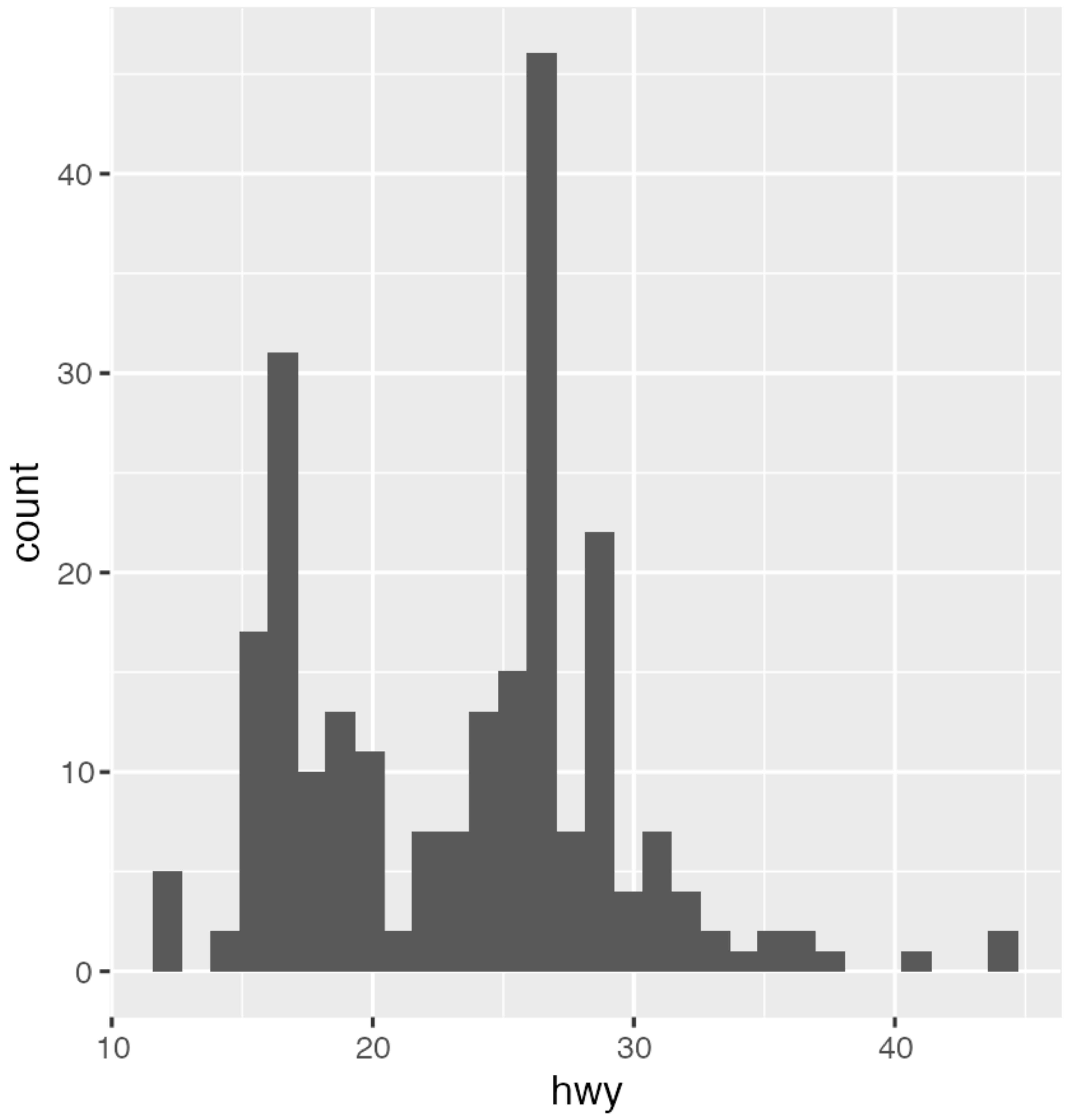
DRAWING ANYTHING

```
ggplot(pie) +
  geom_arc_bar(
    aes(x0 = 0, y0 = 0, r0 = 0, r = 1,
        amount = amount, fill = state),
    stat = ‘pie'
  ) +
  coord_fixed() +
  scale_fill_tableau(
    name = NULL,
    guide = guide_legend(ncol = 2)
  ) +
  theme_void() +
  theme(legend.position = 'top',
        legend.justification = 'left')
```



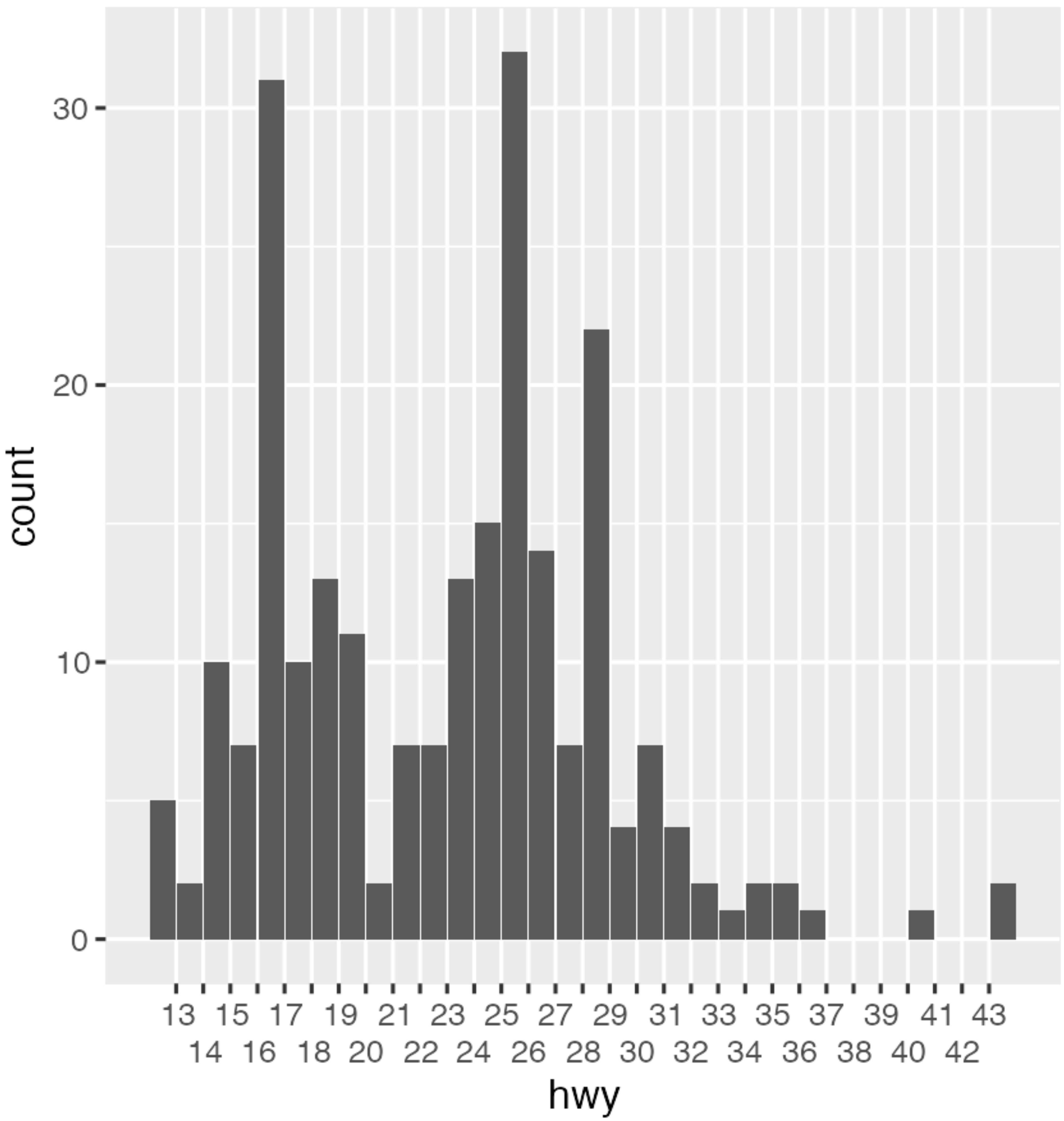
DRAWING ANYTHING

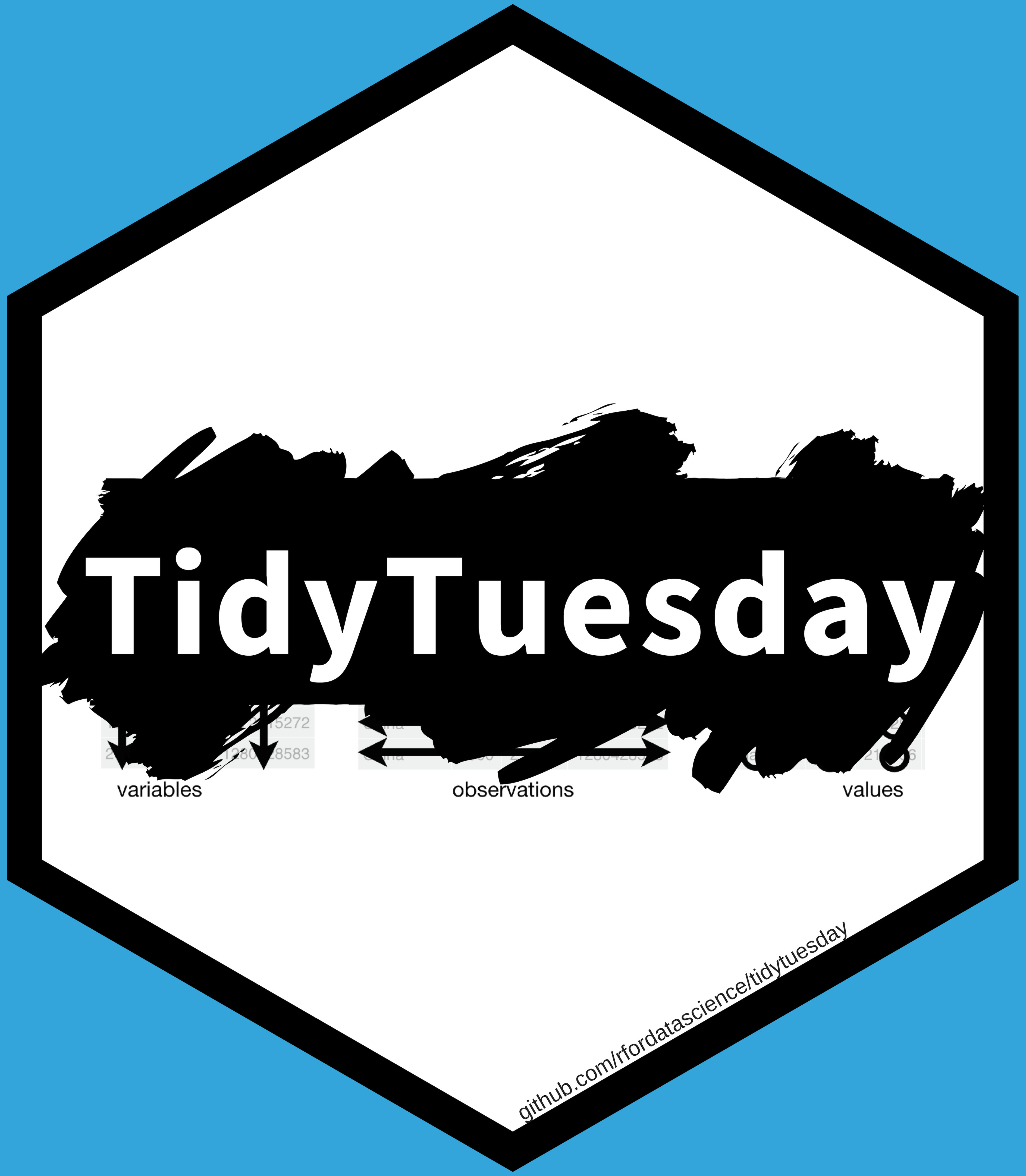
```
ggplot(mpg) +
  # geom_bar(aes(x = hwy), stat = 'bin')
  geom_histogram(aes(x = hwy))
```



DRAWING ANYTHING

```
ggplot(mpg) +
  geom_bar(aes(x = hwy)) +
  scale_x_binned(
    n.breaks = 30,
    guide = guide_axis(n.dodge = 2)
  )
```



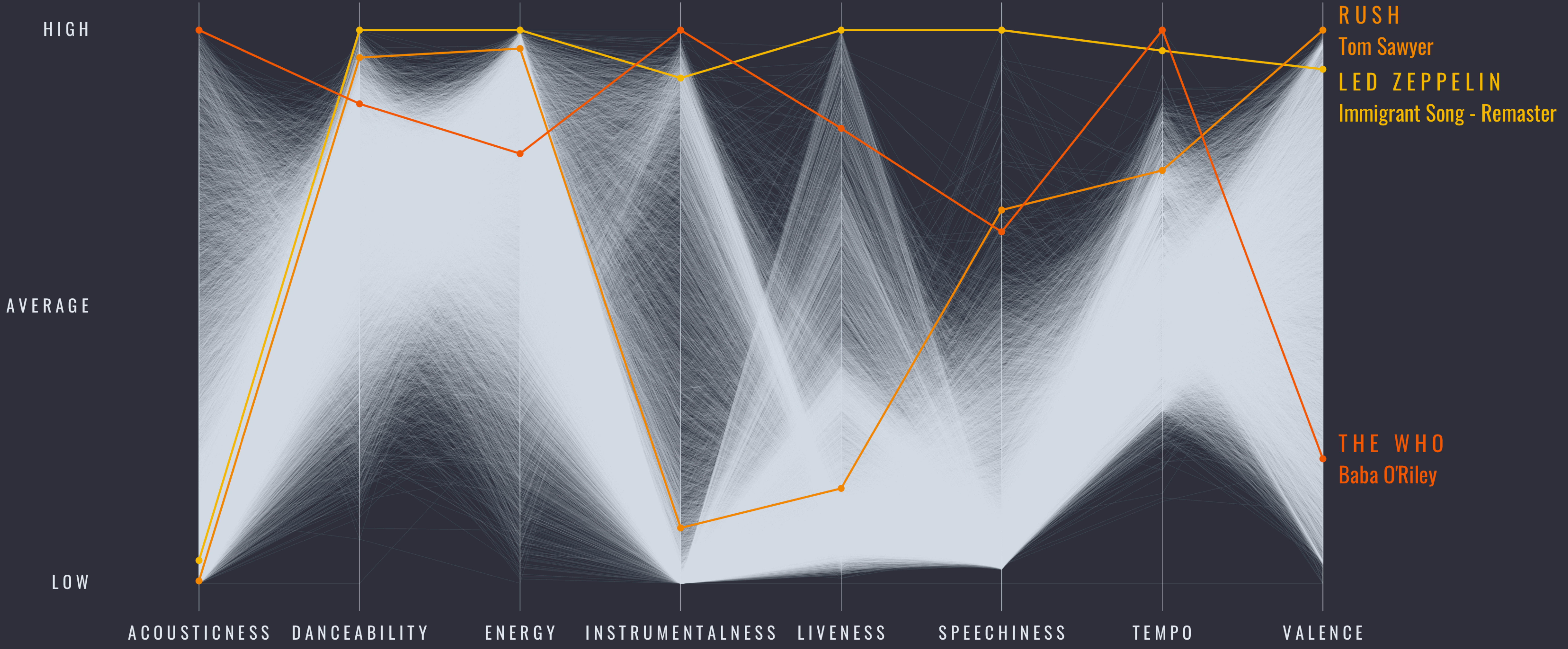


GETTING INSPIRED AND CHALLENGED

- https://github.com/rfordatascience/tidytuesday
- #TidyTuesday on Twitter
- https://nsgrantham.shinyapps.io/tidytuesdayrocks/

Characteristics Of My Favourite Arcade Songs

Ilustrated below is a parallel coordinates plot of Spotify playlist data showing normalized measurements for a variety of audio features accessible via Spotify API. I was introduced **Rush**, **Led Zeppelin** and **The Who** at my local arcade, and look back fondly on those times. These songs still have me look for a quarter and a Tempest machine any time I hear them.

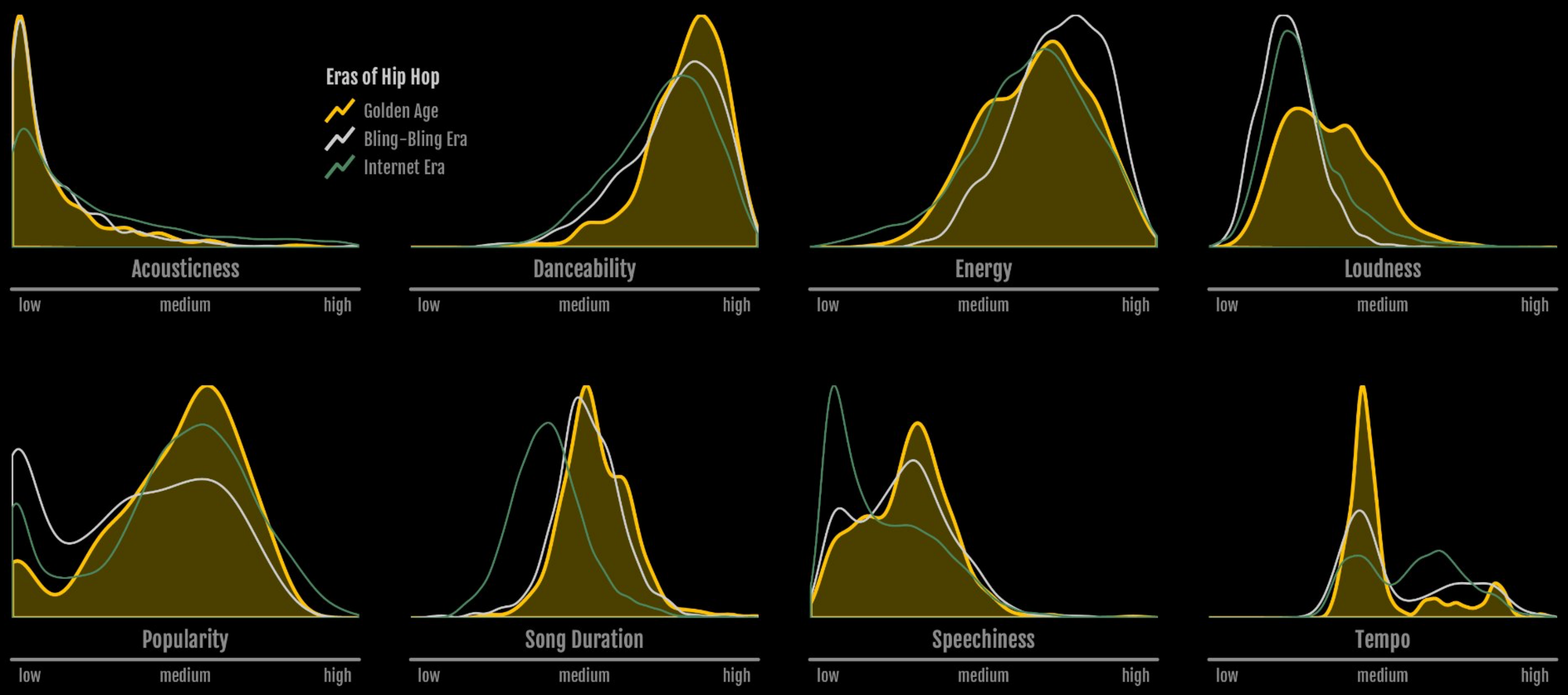


Data: Spotify | **Graphic**: @jakekaupp

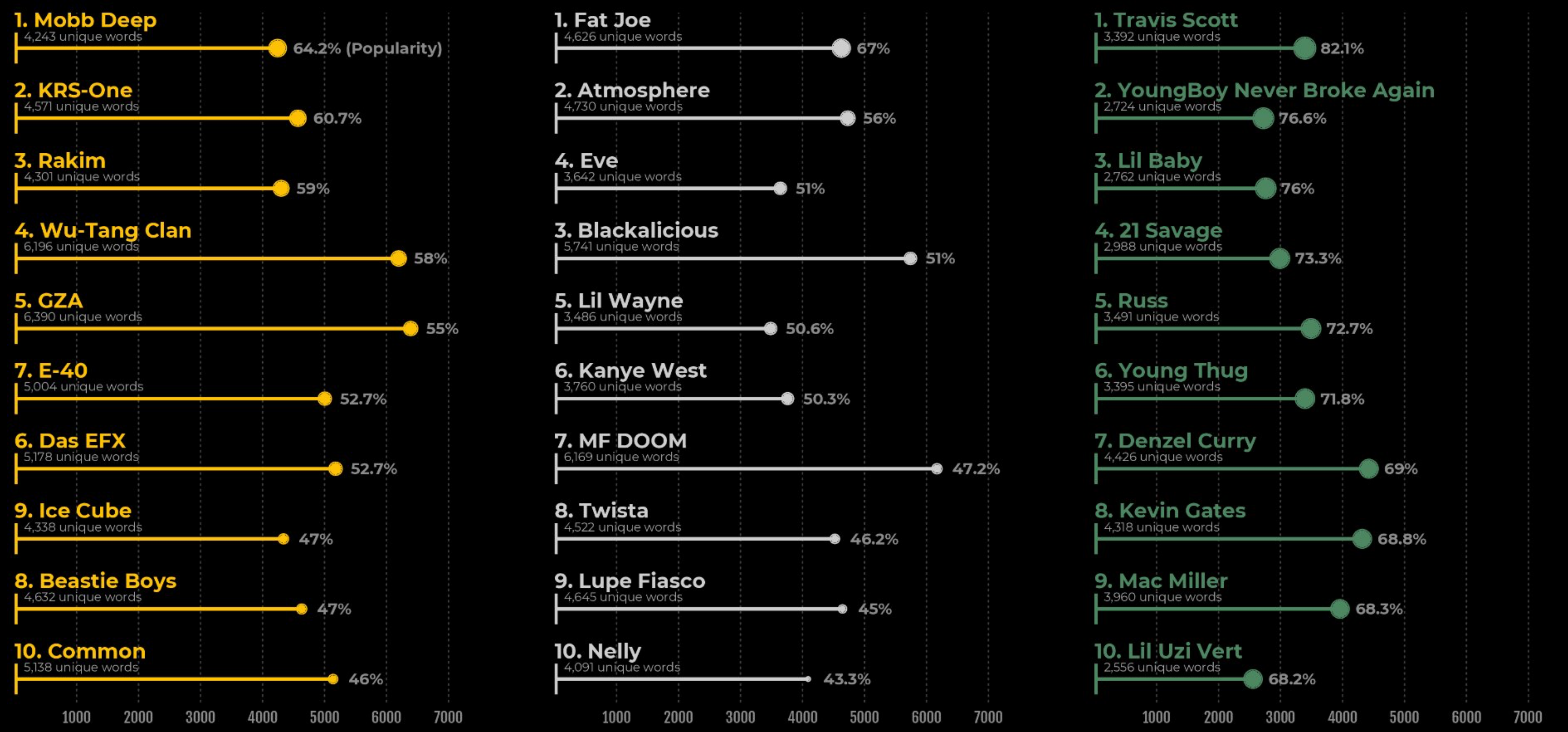
@JAKEKAUPP

The Golden Age of Hip Hop in the Era of Spotify

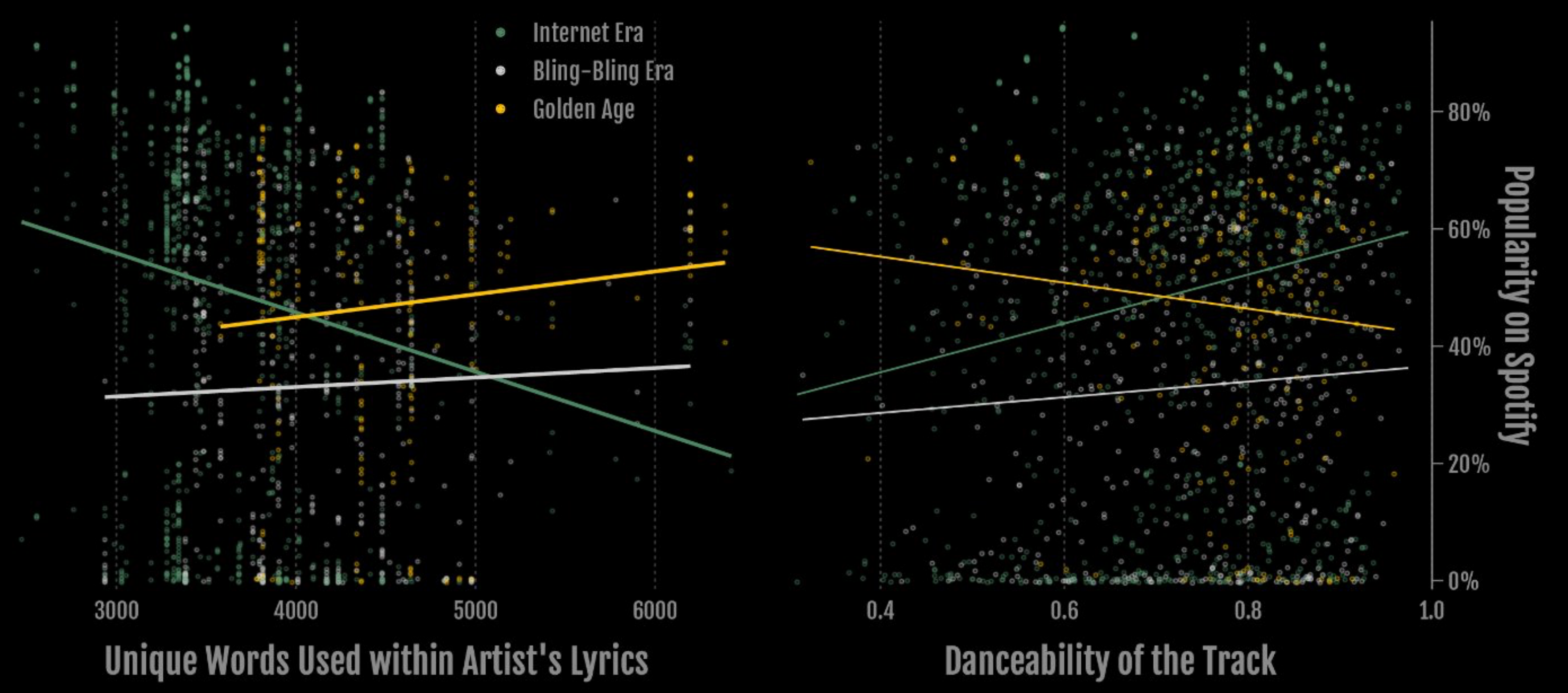
It is generally accepted that the **Golden Age of Hip Hop** occurred from the mid 1980s and mid 1990s. It was then that all the elements of the culture—breaking, graffiti art, DJing, and rap—broke cover to enter the mainstream. **N.W.A., Eric B. & Rakim, Run DMC,** and the **Beastie Boys** allowed rap music to become the culture's crowning glory.With the likes of DMX, Dr. Dre, Eminem, Nelly, and 2Pac all selling albums in their tens of millions, Hip Hop became a game changer, one of the most popular styles in modern music and revolutionized youth culture.



Rap music during the **Golden Age (1985-1996)** was music you would like to shake your booty to! Tracks are often more danceable and louder but a bit slower than rap songs from the **Bling-Bling Era (1997-2008)** and the Internet Era (2009-today) according to the audio features provided by Spotify. Even though the **Golden Age** was way before **Spotify** or even the internet became a mainstream phenomenon, many artists and tracks from back then are popular on Spotify.



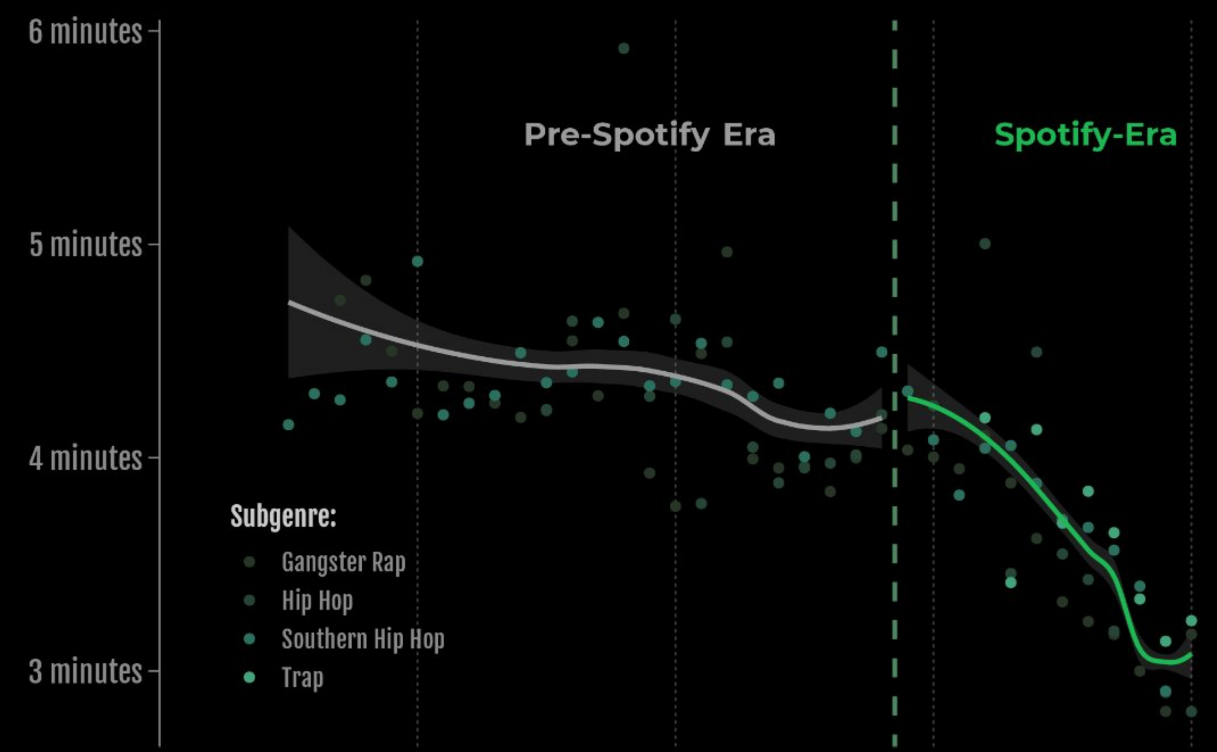
Popular rappers from nowadays, however, use way less unique words. Is that the reason for the higher popularity?



Two variables exhibit distinct patterns
for the different eras of rap:
Indeed, a smaller rap vocabulary
is correlated with higher popularity
for tracks from the Internet Era
while popularity of tracks from
artists of the **Golden Age** and
Bling-Bling Era increases with more
unique words. Also, popularity
increases with danceability for more
recent tracks, while the trend is re-
versed for tracks from the **Golden Age**.

The Spotify Effect on Track Length?

Track duration dropped from around 4.5 minutes to close and even below 3 minutes. Several blame **Spotify** (and other streaming platforms) for this trend since shorter songs bring a higher net revenue. But also a decrease in the human attention span in an ever-faster world in combination with endless amounts of music provided by the streaming platforms are discussed as causes for this **Spotify Effect**. The effect an not be only found for and rap its subgenres but all major music genres with slight variation in the duration of recently released tracks among genres.

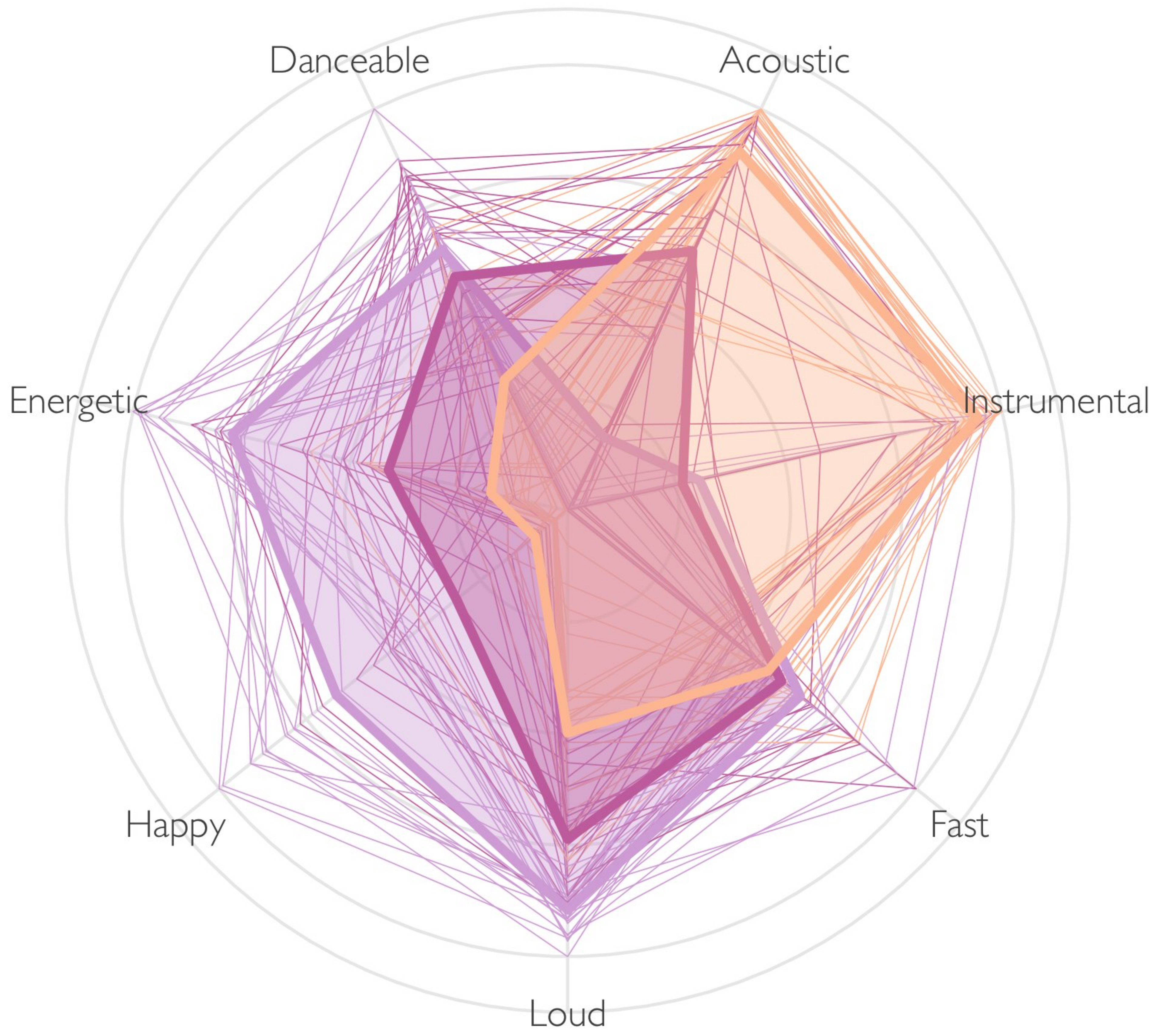


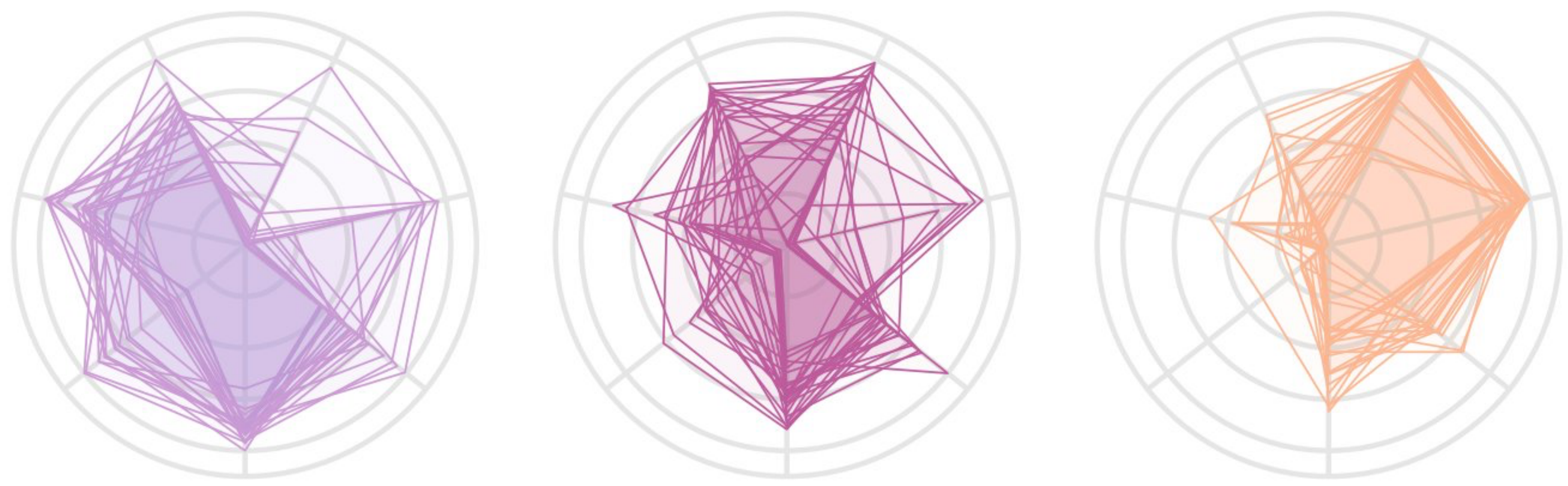
Visualization by **Cédric Scherer** • Song Data by **Spotify** via **{spotifyr}** • Data on Rap Vocabularies by Matt Daniels

@CEDSCHERER

The Flavors of 3 Playlists

Backyard BBQ Mellow Jams Study Songs



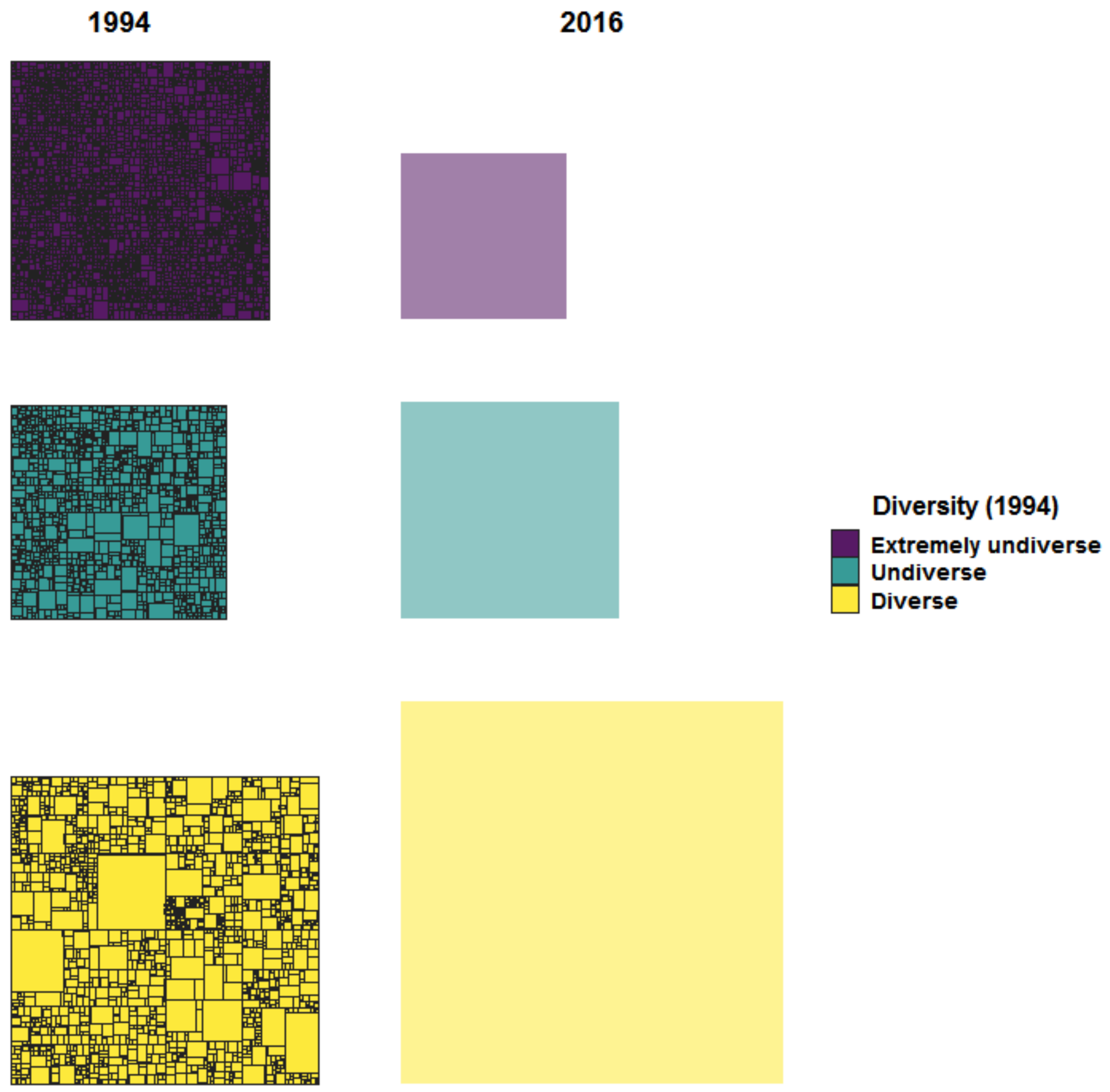


Data from Spotify & SpotifyR | Visualization by @Jake_Lawlor1

@JAKE_LAWLOR1

School Diversity

Diversity Changes of 12,217 U.S. School Districts



#tidytuesday 2019.39 | @jonthegeek

WANT MORE?

- So many great free books:
  - ggplot2: Elegant Graphics for Data Analysis
    ggplot2-book.org
  - Data Visualization: A Practical Introduction
    socviz.co
  - Fundamentals of Data Visualization
    serialmentor.com/dataviz/

WANT MORE?

- So many websites:
  - ggplot2.tidyverse.org
  - gganimate.com
  - patchwork.data-imaginist.com
  - ggforce.data-imaginist.com
  - ggraph.data-imaginist.com
  - ggrepel.slowkow.com

THANK YOU