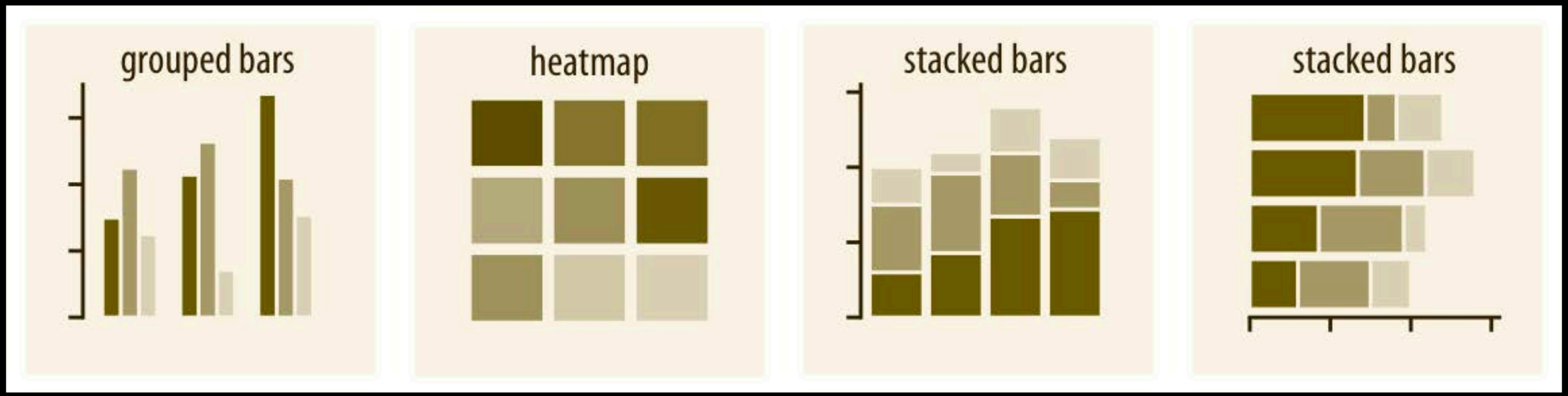
ANT 6973: DATA VISUALIZATION AND EXPLORATION

VISUALIZING DISTRIBUTIONS AND UNCERTAINTY

PREVIOUSLY...

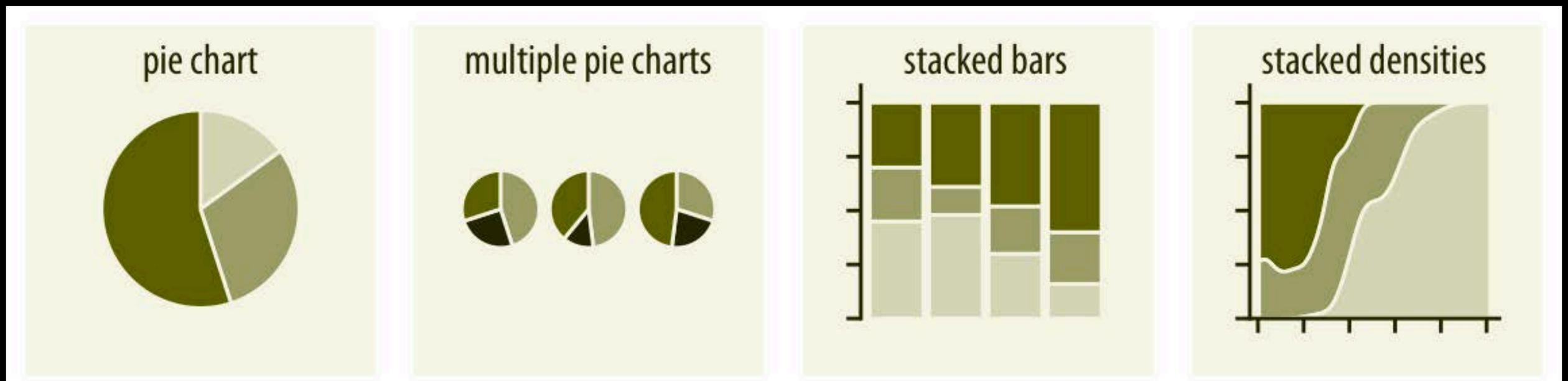
PREVIOUSLY...

- Visualizing amounts with bar charts (and variants)



PREVIOUSLY...

- Visualizing proportions with pies, stacked bars, and stacked areas



PREVIOUSLY...

- Import from a .csv file using `read_csv()` the readr package.

```
my_data <- read_csv(“path/to/data.csv”)
```

New R object where data will be stored

`readr` function

Path of data file relative to project home

PREVIOUSLY...

- Stats
- Position adjustments
- Coordinates
- Facets
- Scales
- Themes

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

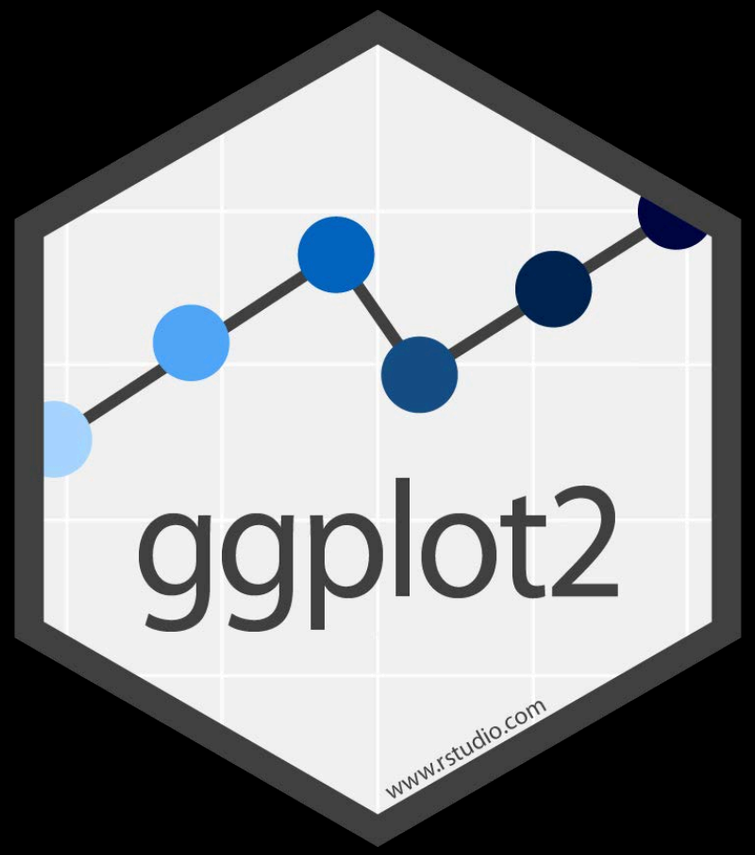
Required

Not required, sensible defaults supplied

TODAY'S TOPICS

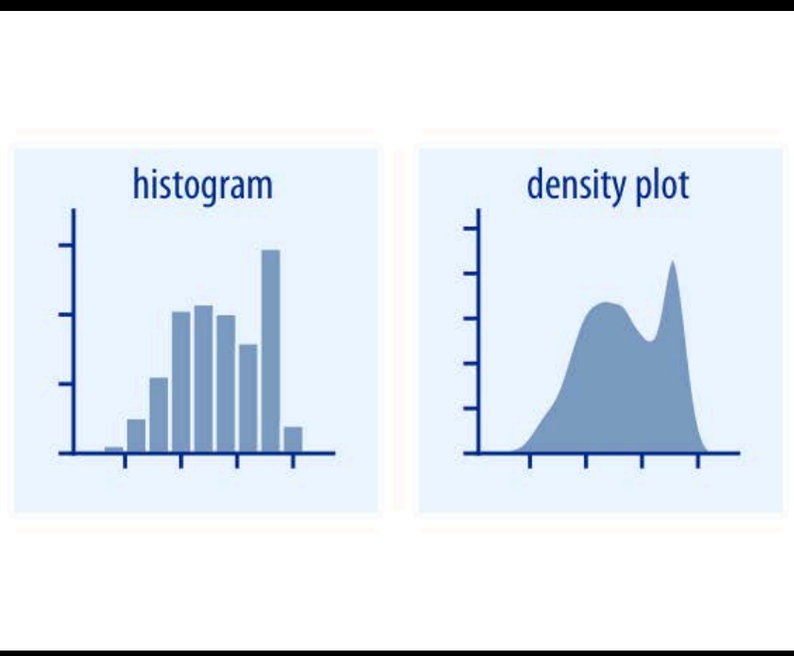
- Visualizing single distributions
- Visualizing multiple distributions
- Visualizing uncertainty
- Activities:
  - Body mass distributions of late Quaternary mammals

VISUALIZING DISTRIBUTIONS

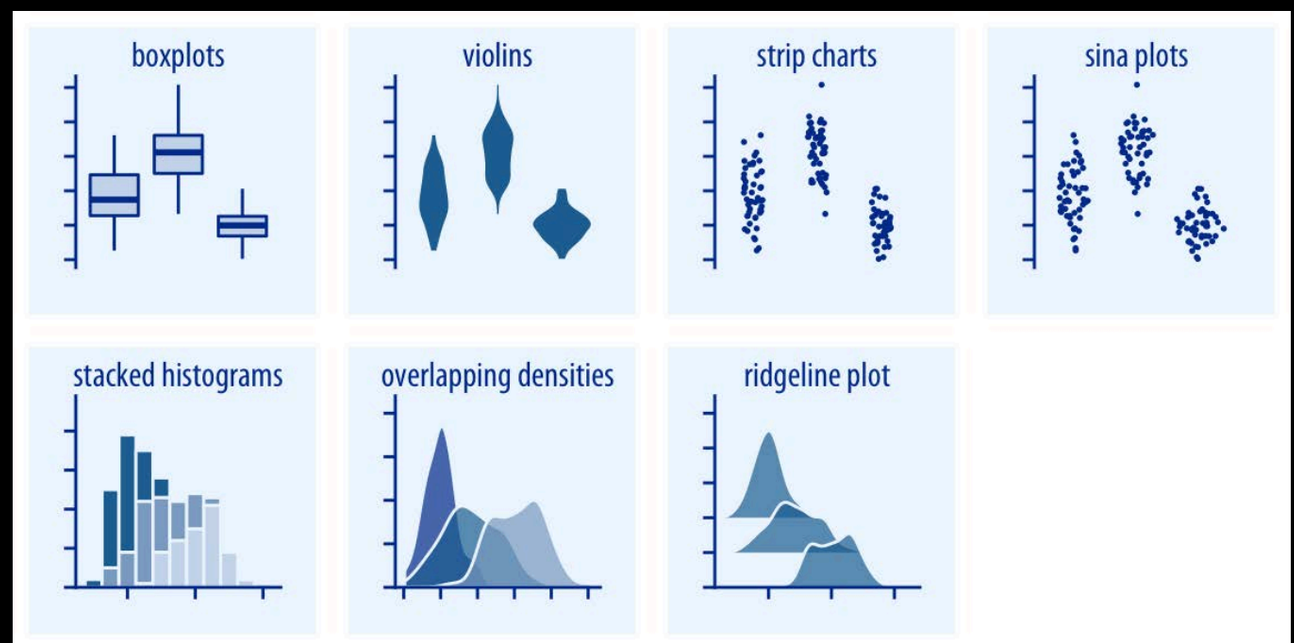


VISUALIZING DISTRIBUTIONS

Single distributions



Multiple distributions



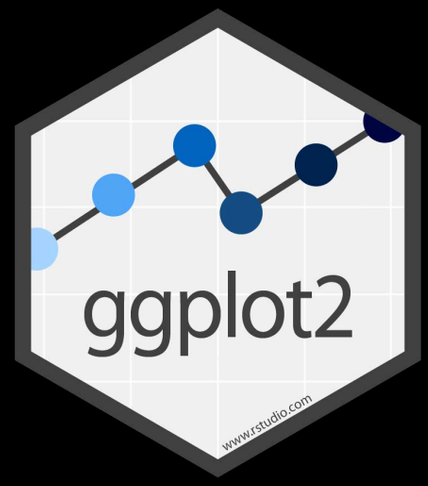
HISTOGRAMS

HISTOGRAMS

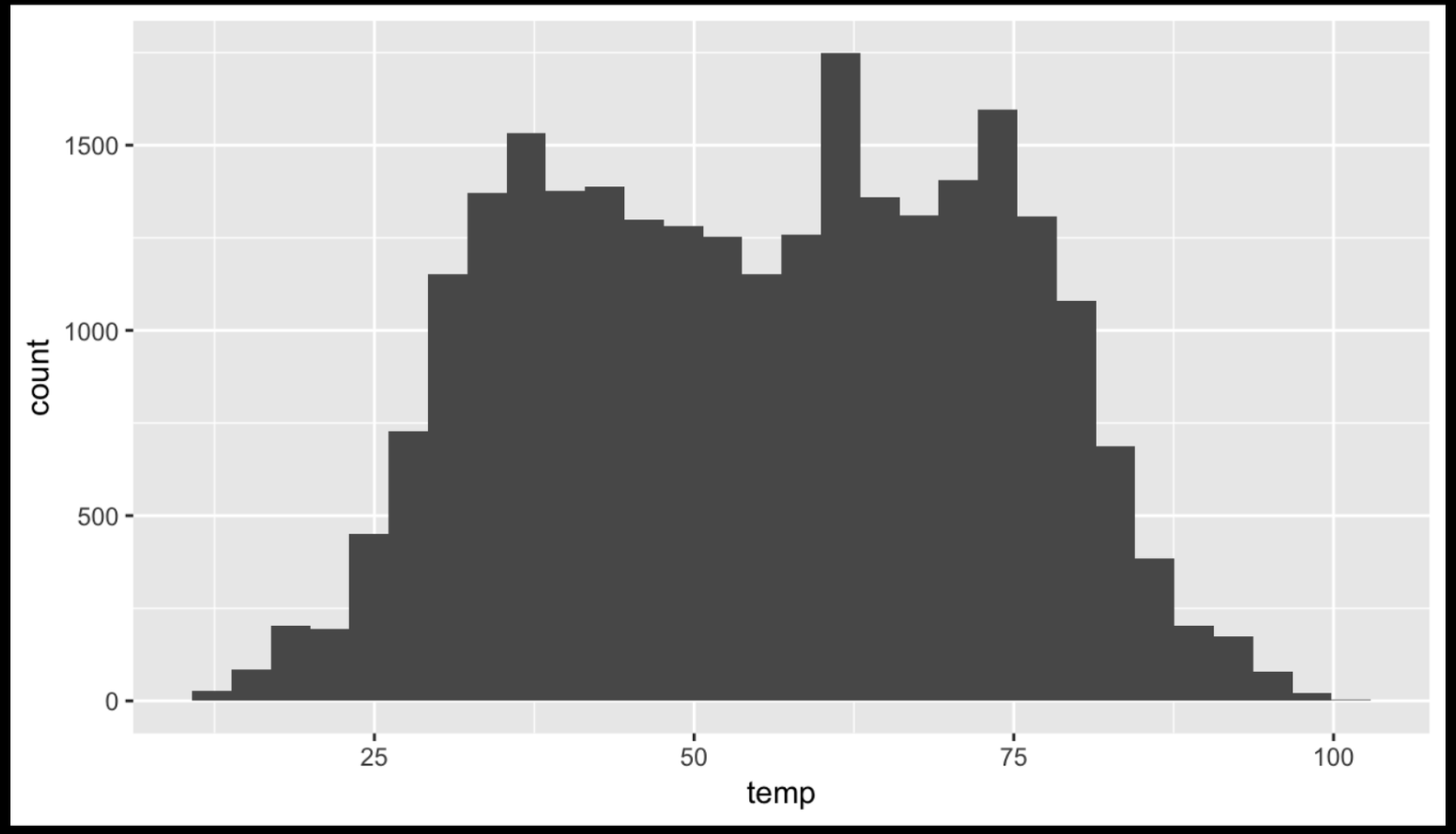
- Shows the distribution of a single variable
- Breaks the data apart into specific bins and shows *count* of cases in each bin

HISTOGRAMS

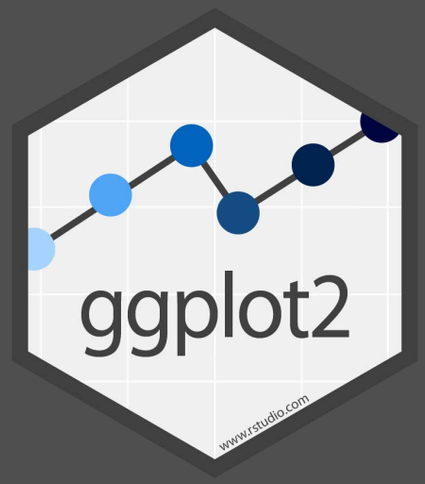
Major difference with bar charts



- Geometric object is `geom_histogram()`
- One numeric variable to mapped to x in `aes()`
- y-aesthetic computed automatically (don't map anything to y)



YOUR TURN



- Download the late Quaternary mammals data set from the course website ("late_quaternary_mammals.RDS"). This is a file format for single R objects.

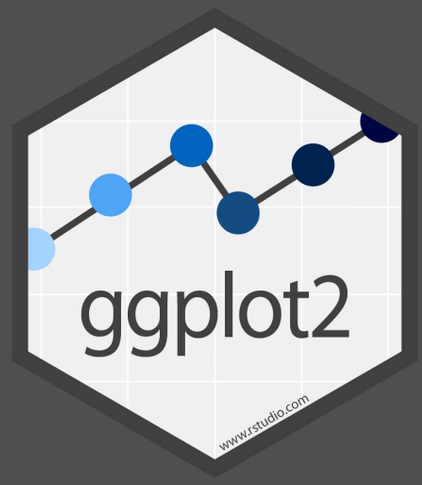
- Read the file into R using:

```
lqm <- readRDS("your-path/late_quaternary_mammals.RDS")
```

YOUR TURN

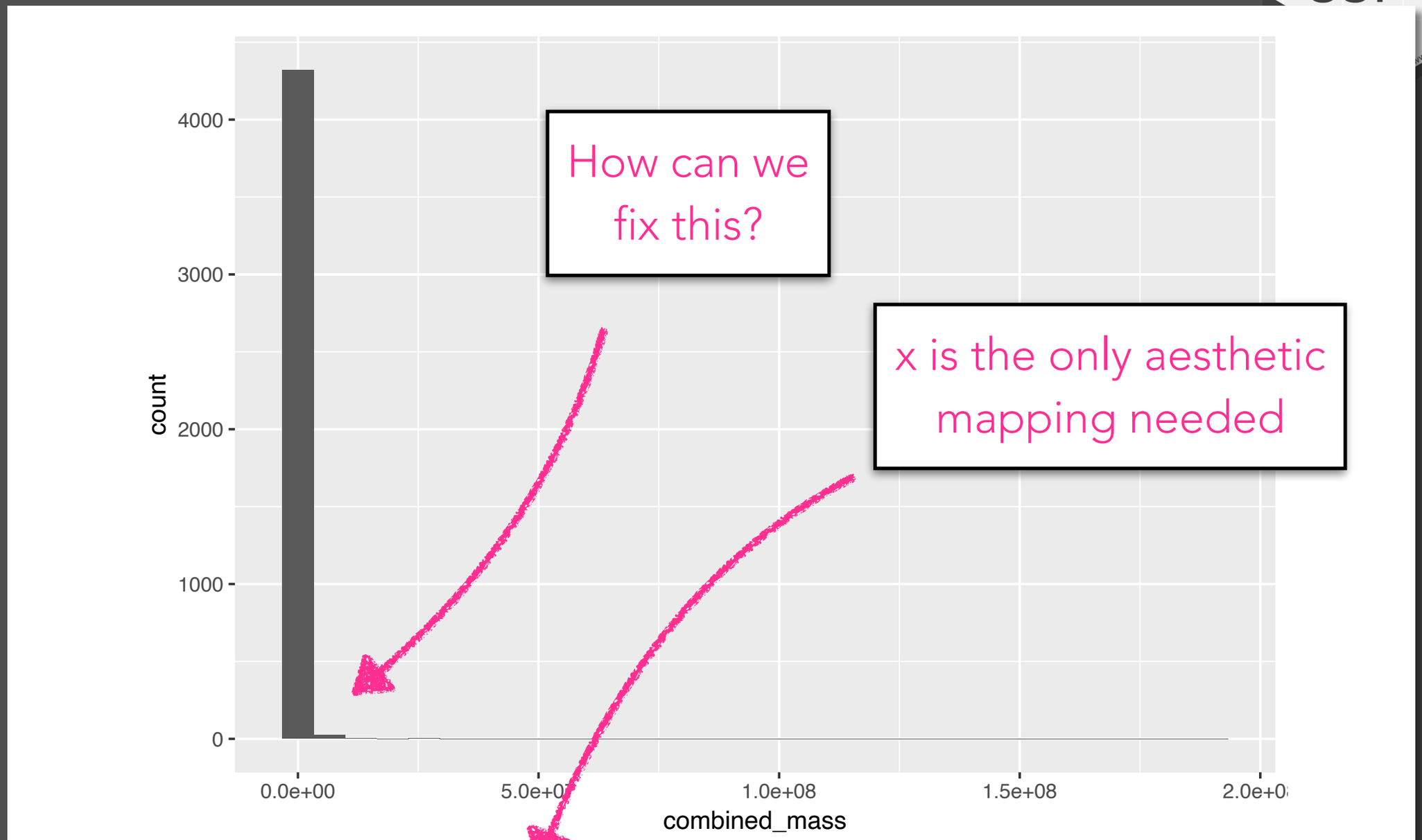
Average of M/F body masses (g)

log_{10}(combined_mass)



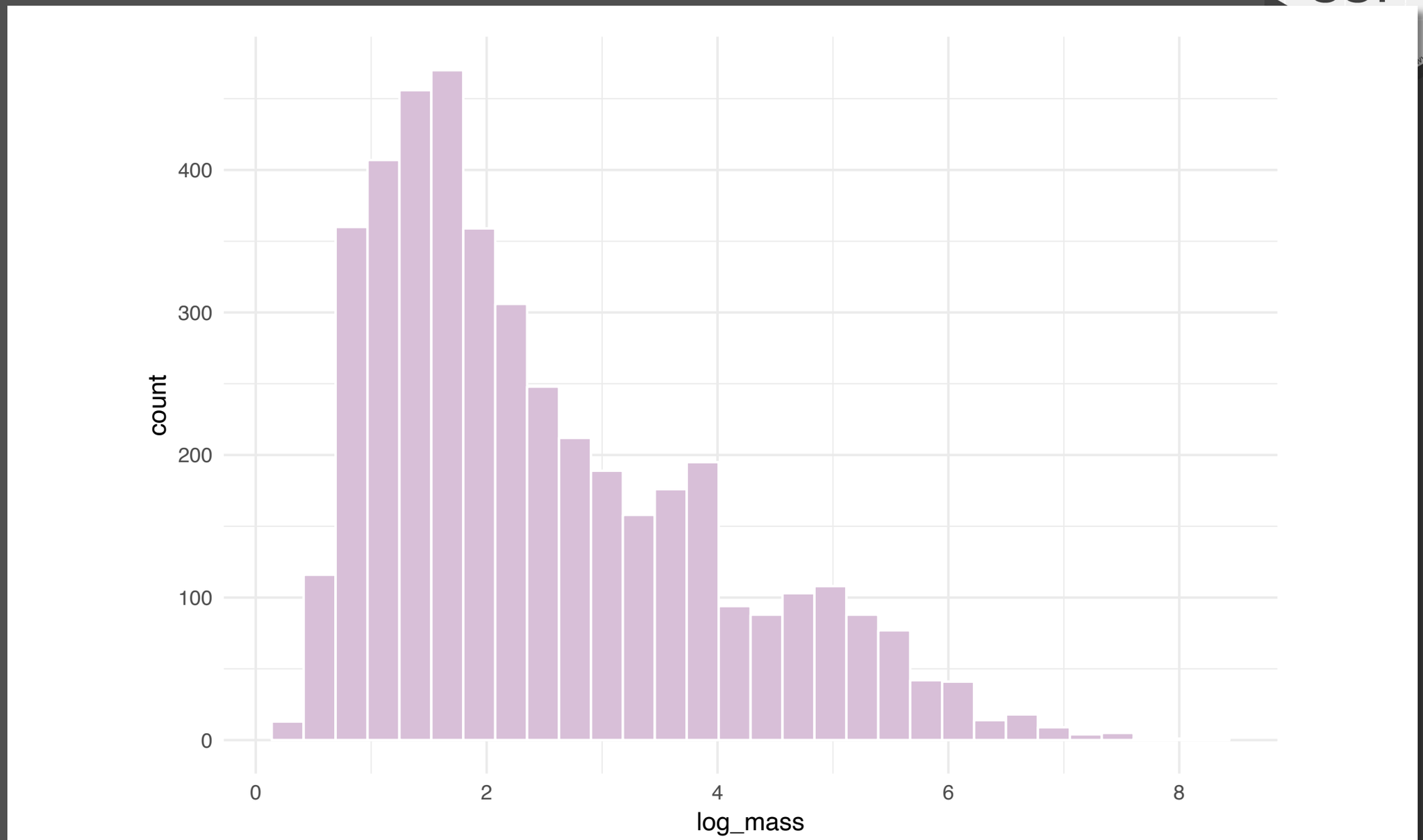
continent	status	order	family	genus	species	log_mass	combined_mass	reference
Africa	extant	Artiodactyla	Bovidae	Addax	nasomaculatus	4.85	70000.3	60
Africa	extant	Artiodactyla	Bovidae	Aepyceros	melampus	4.72	52500.1	63, 70
Africa	extant	Artiodactyla	Bovidae	Alcelaphus	buselaphus	5.23	171001.5	63, 70
Africa	extant	Artiodactyla	Bovidae	Ammodorcas	clarkei	4.45	28049.8	60
Africa	extant	Artiodactyla	Bovidae	Ammotragus	lervia	4.68	48000.0	75
Africa	extant	Artiodactyla	Bovidae	Antidorcas	marsupialis	4.59	39049.9	60
Africa	extinct	Artiodactyla	Bovidae	Antidorcas	bondi	4.53	34000.0	1
Africa	extinct	Artiodactyla	Bovidae	Antidorcas	australis	4.60	40000.0	2
Africa	extant	Artiodactyla	Bovidae	Bos	taurus	5.95	900000.0	-999
Africa	extant	Artiodactyla	Bovidae	Capra	walie	5.00	100000.0	-999

YOUR TURN



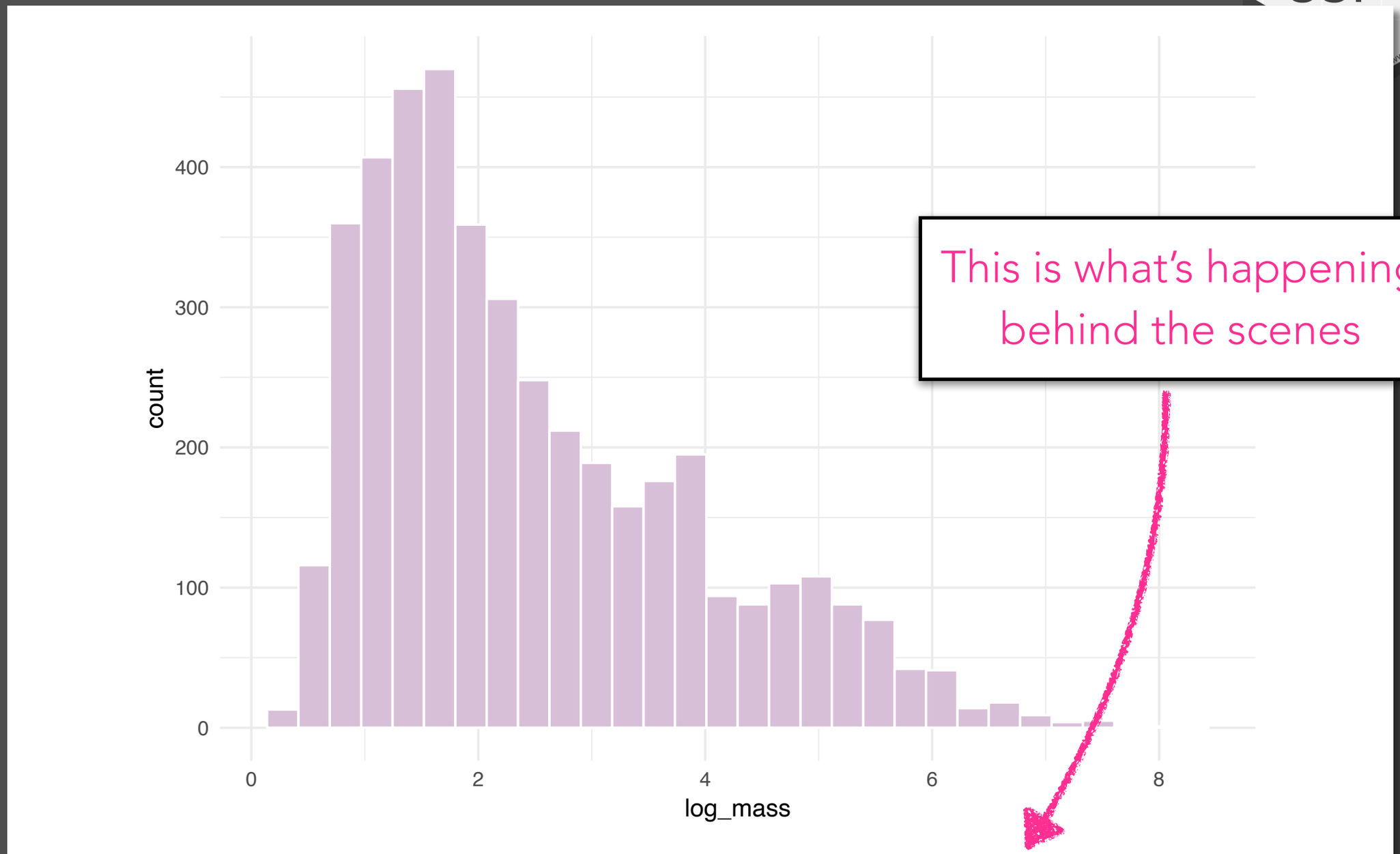

```
ggplot(lqm, aes(x = combined_mass)) +
  geom_histogram()
```

YOUR TURN




```
ggplot(lqm, aes(x = log_mass)) +
  geom_histogram(fill = "thistle", color = "white")
```

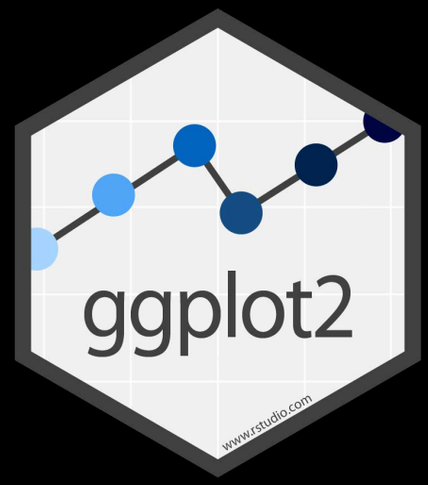
YOUR TURN



This is what's happening behind the scenes

```
ggplot(lqm, aes(x = log_mass, y = stat(count))) +
  geom_histogram(fill = "thistle", color = "white")
```

HISTOGRAMS



- *When making a histogram, always explore multiple bin widths.*

- Binning behavior can be modified using:

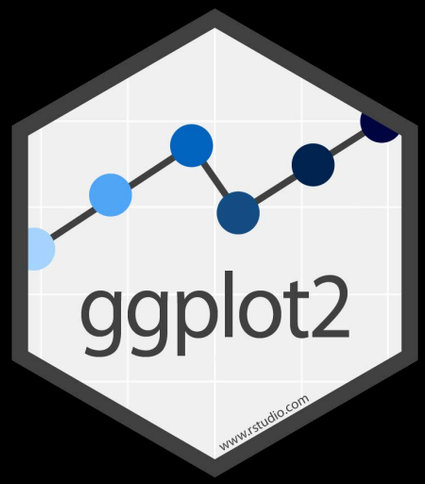
  - bins: `geom_histogram(bins = 30)`
  - binwidth: `geom_histogram(binwidth = 1)`
  - breaks: `geom_histogram(breaks = 0:10)`

# of bins

width of one bin in units of x

sequence of bin boundaries

HISTOGRAMS



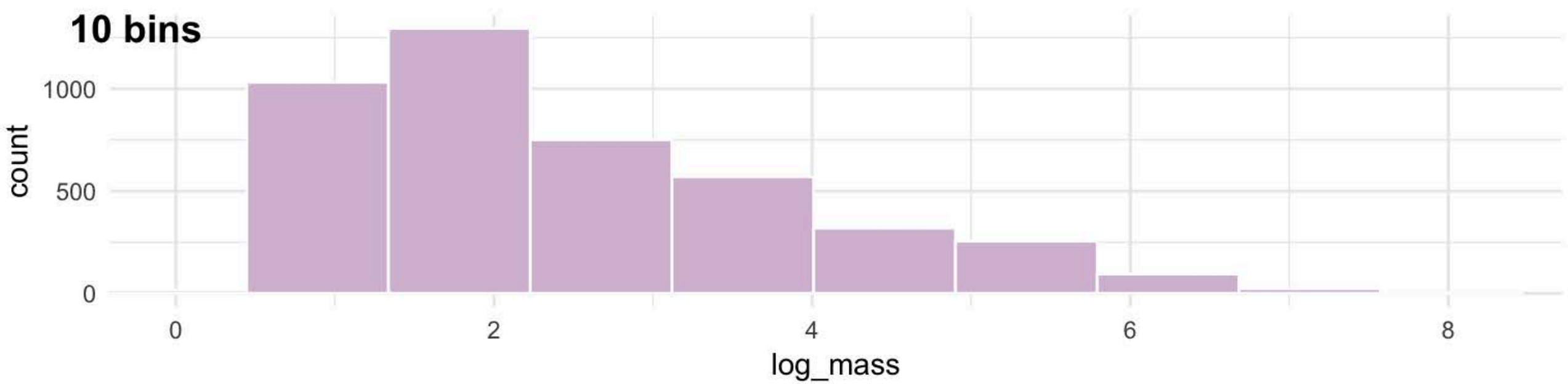
- *When making a histogram, always explore multiple bin widths.*

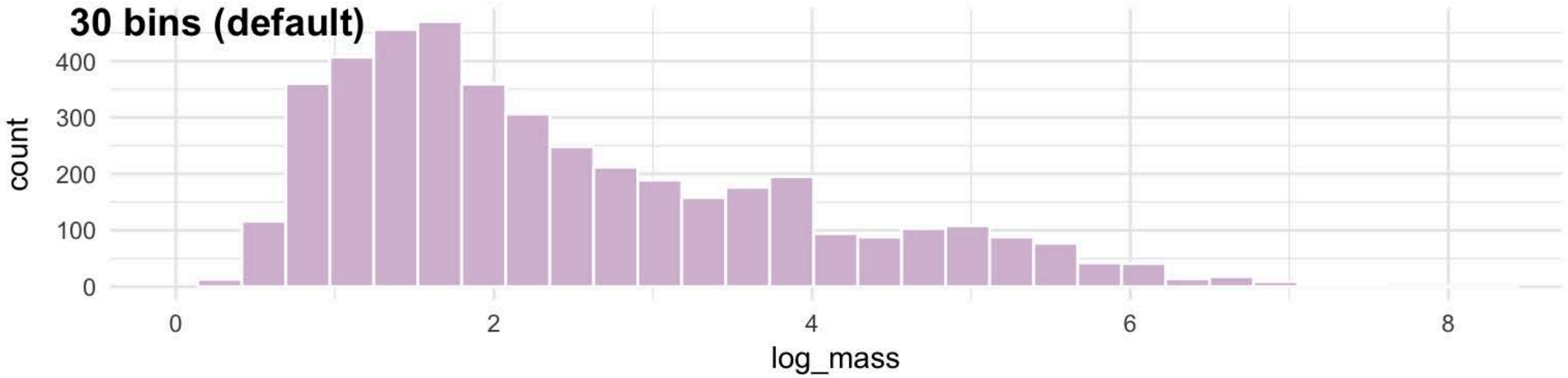
- Binning behavior can be modified using:

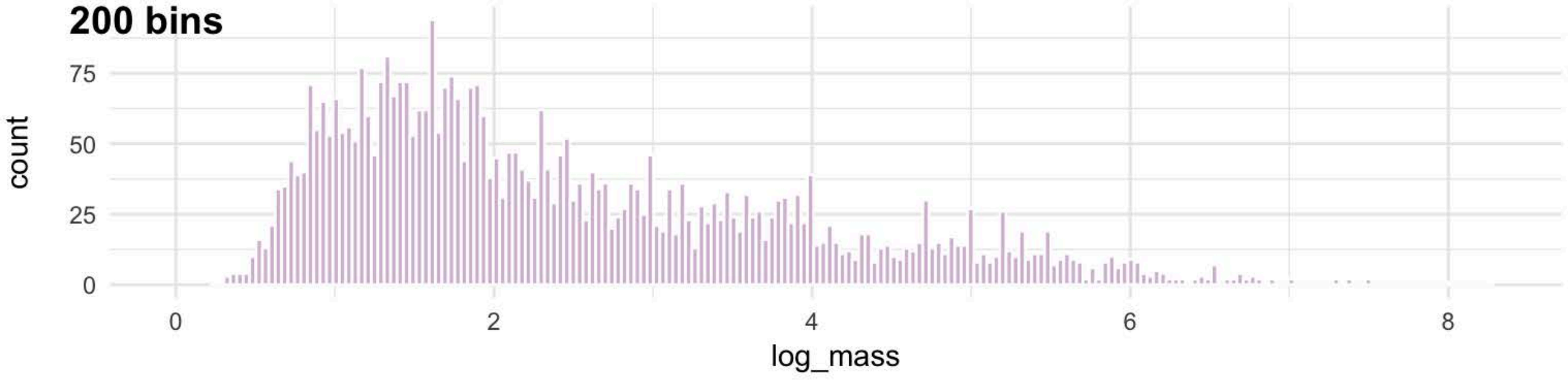
  - bins: `geom_histogram(bins = 30)`

  - binwidth: `geom_histogram(binwidth = 1)`

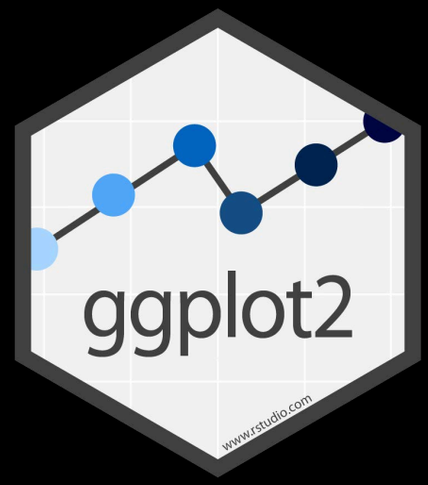
  - breaks: `geom_histogram(breaks = 0:10)`


10 bins
count
1000
500
0
0
2
4
6
8
log_mass


30 bins (default)
count
400
300
200
100
0
0
2
4
6
8
log_mass


200 bins
count
75
50
25
0
0
2
4
6
8
log_mass

HISTOGRAMS

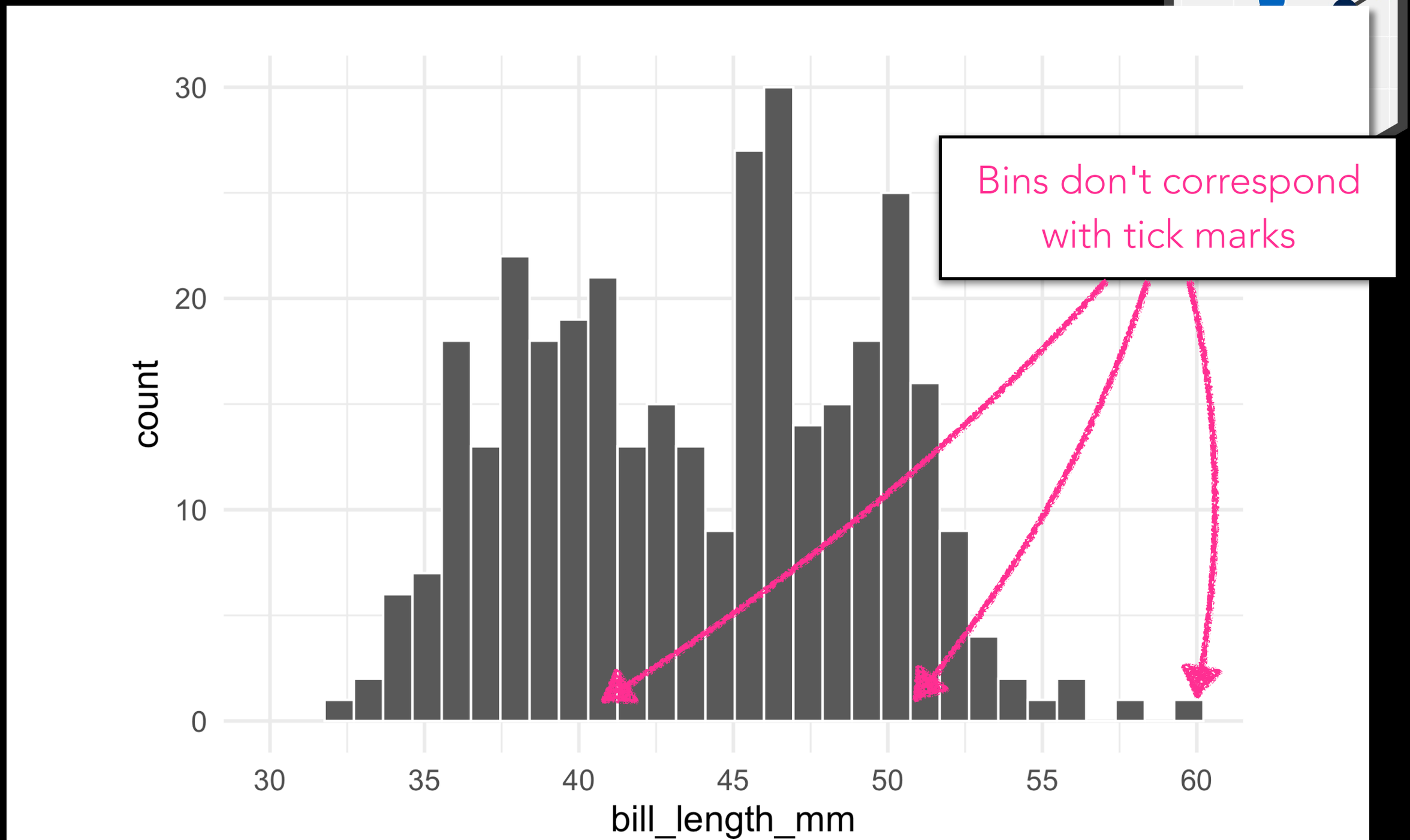


- *Also pay attention to the bin positions!*
  The defaults don't always make sense.

- Binning positions can be modified with ONE (not both) of the following:

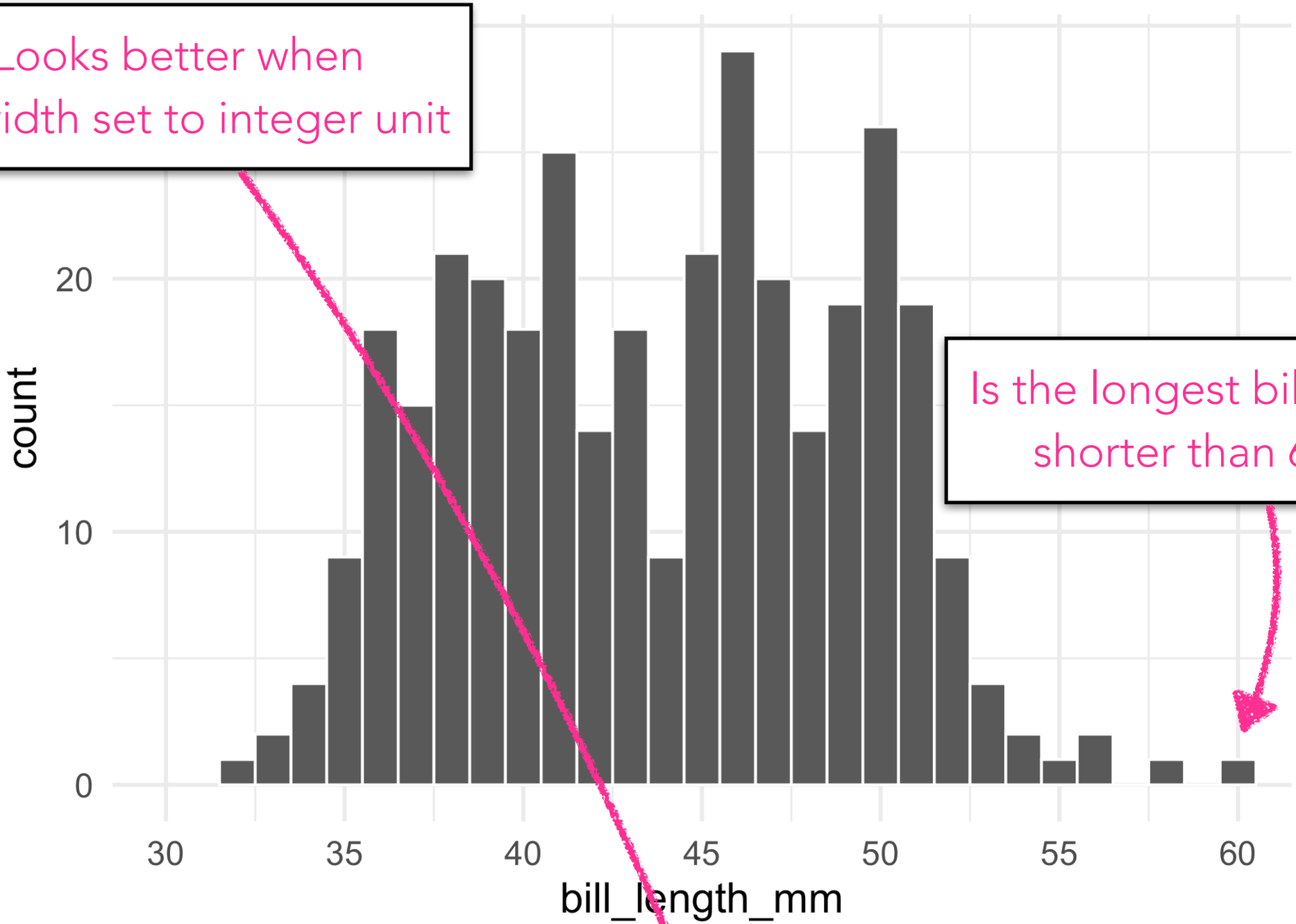
  - center: an x value for the center of one of the bins (the rest will be calculated)

  - boundary: an x value for the boundary between any two bins (the rest will be calculated)




```
ggplot(penguins, aes(x = bill_length_mm)) +
  geom_histogram(color = "white", bins = 30)
```

Looks better when binwidth set to integer unit

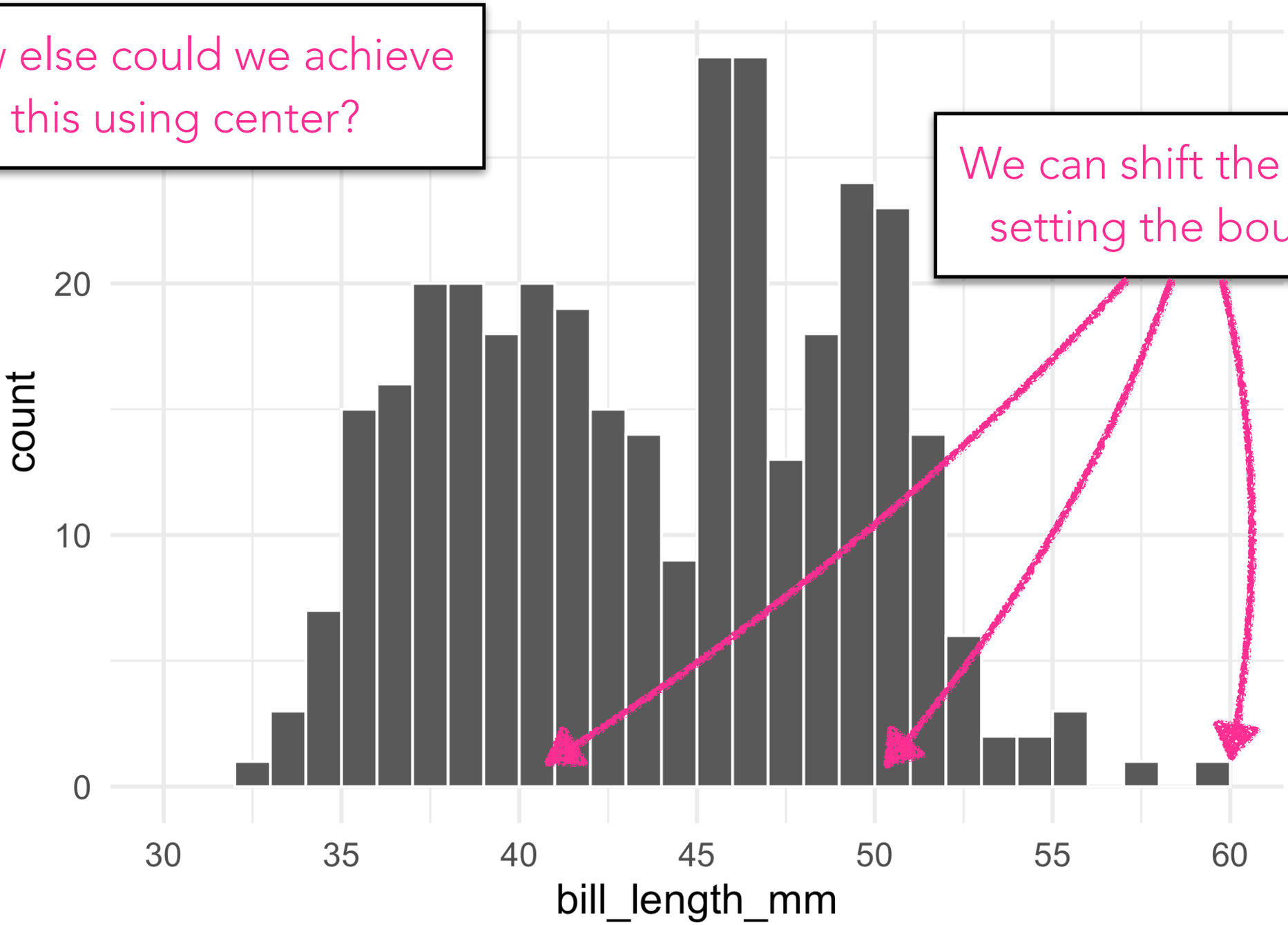


Is the longest bill longer or shorter than 60 mm?

```
ggplot(penguins, aes(x = bill_length_mm)) +
  geom_histogram(color = "white", binwidth = 1)
```

How else could we achieve this using center?

We can shift the bars by setting the boundary



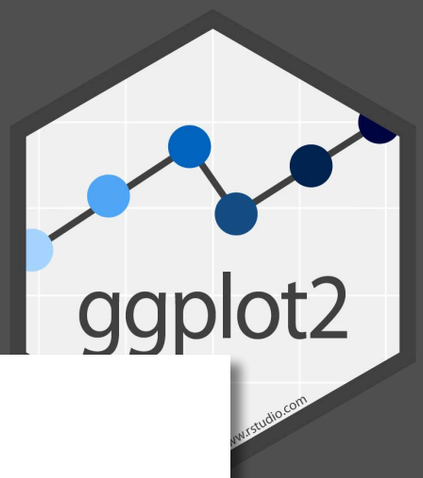

```
ggplot(penguins, aes(x = bill_length_mm)) +
  geom_histogram(color = "white", binwidth = 1, boundary = 40)
```

HISTOGRAMS

- Histograms are generally bad at showing multiple distributions together

- For example…

YOUR TURN



Try to predict what this will do…

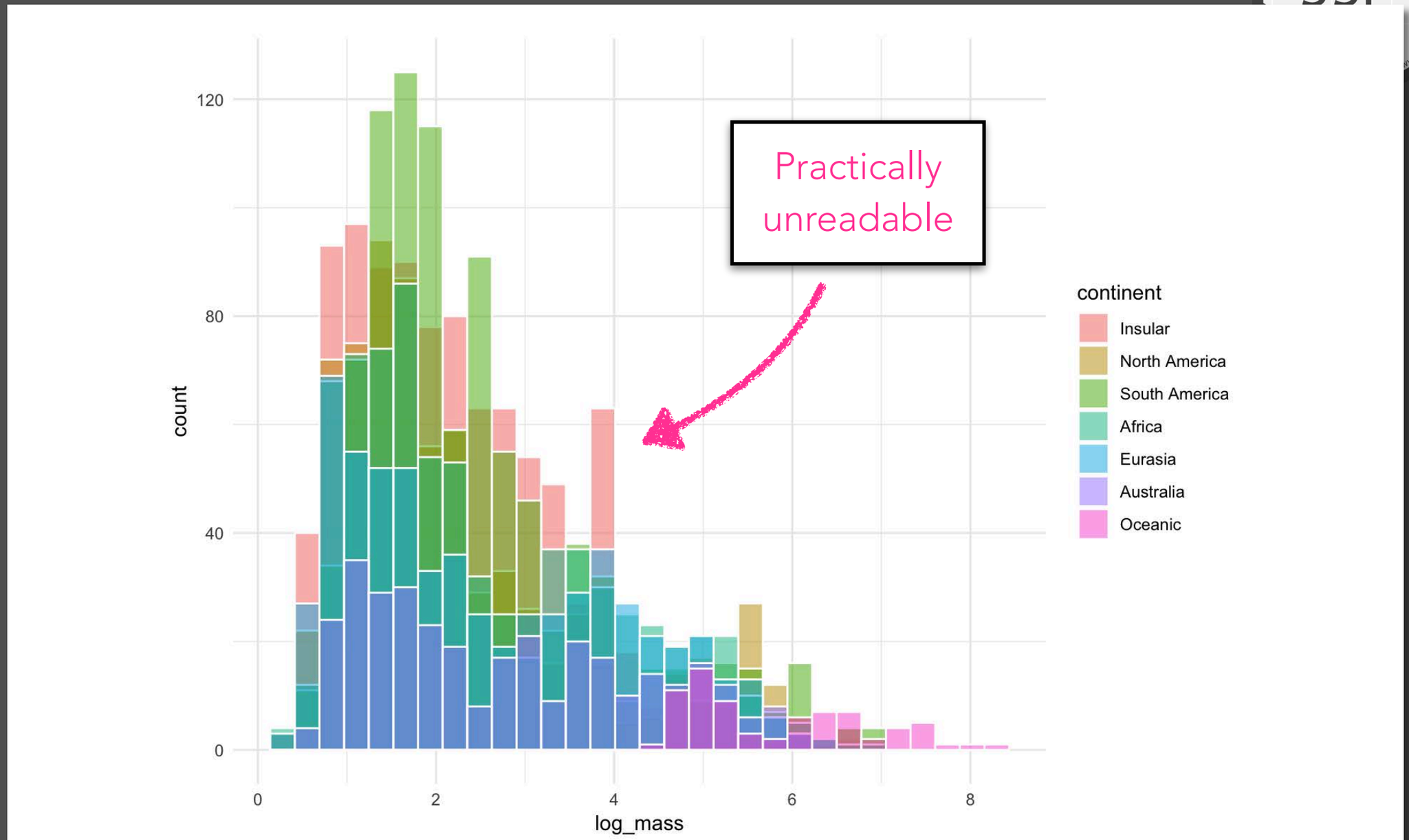
Different fill color for each continent

Makes histograms overlap

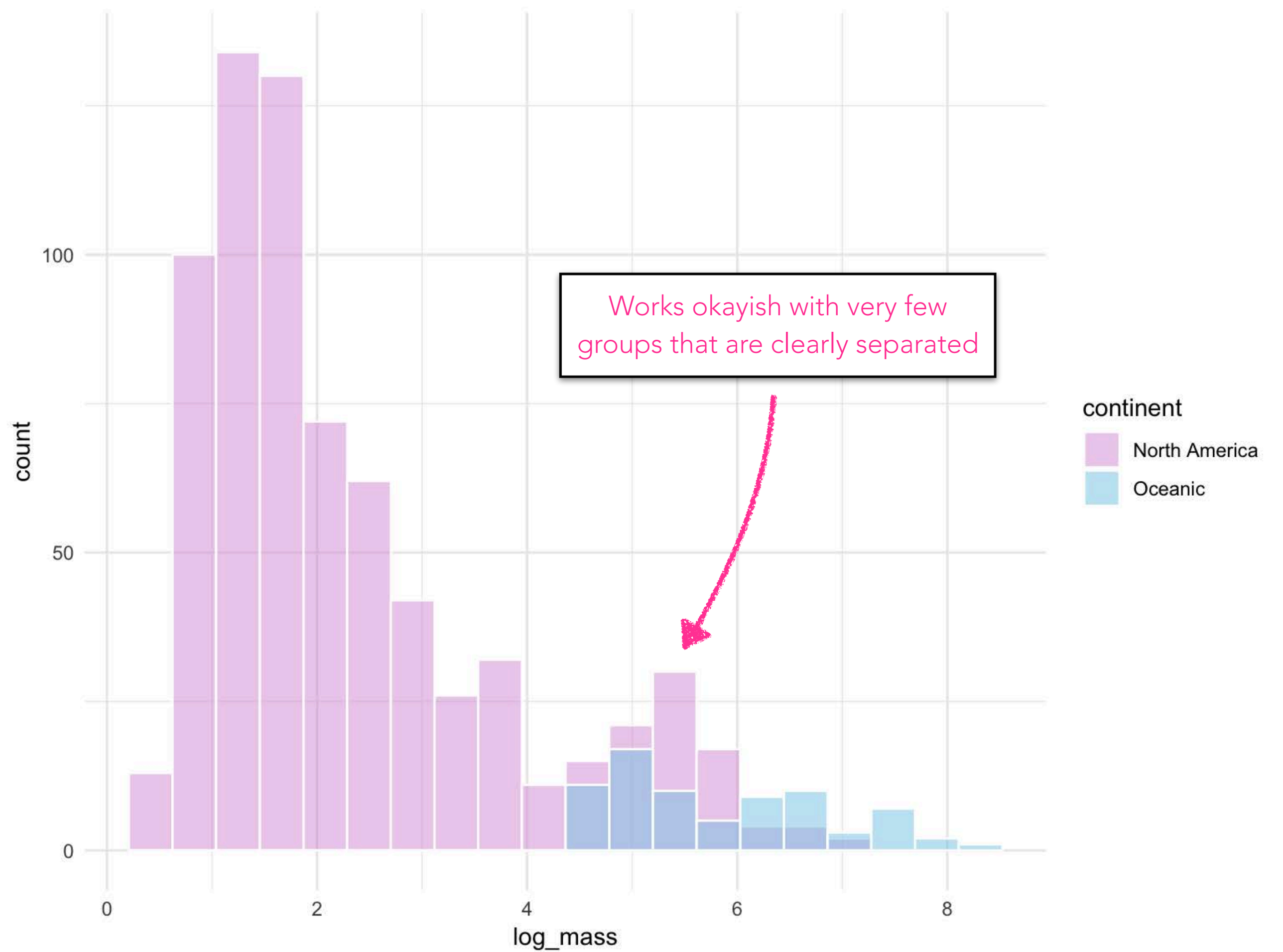
Makes them partially transparent

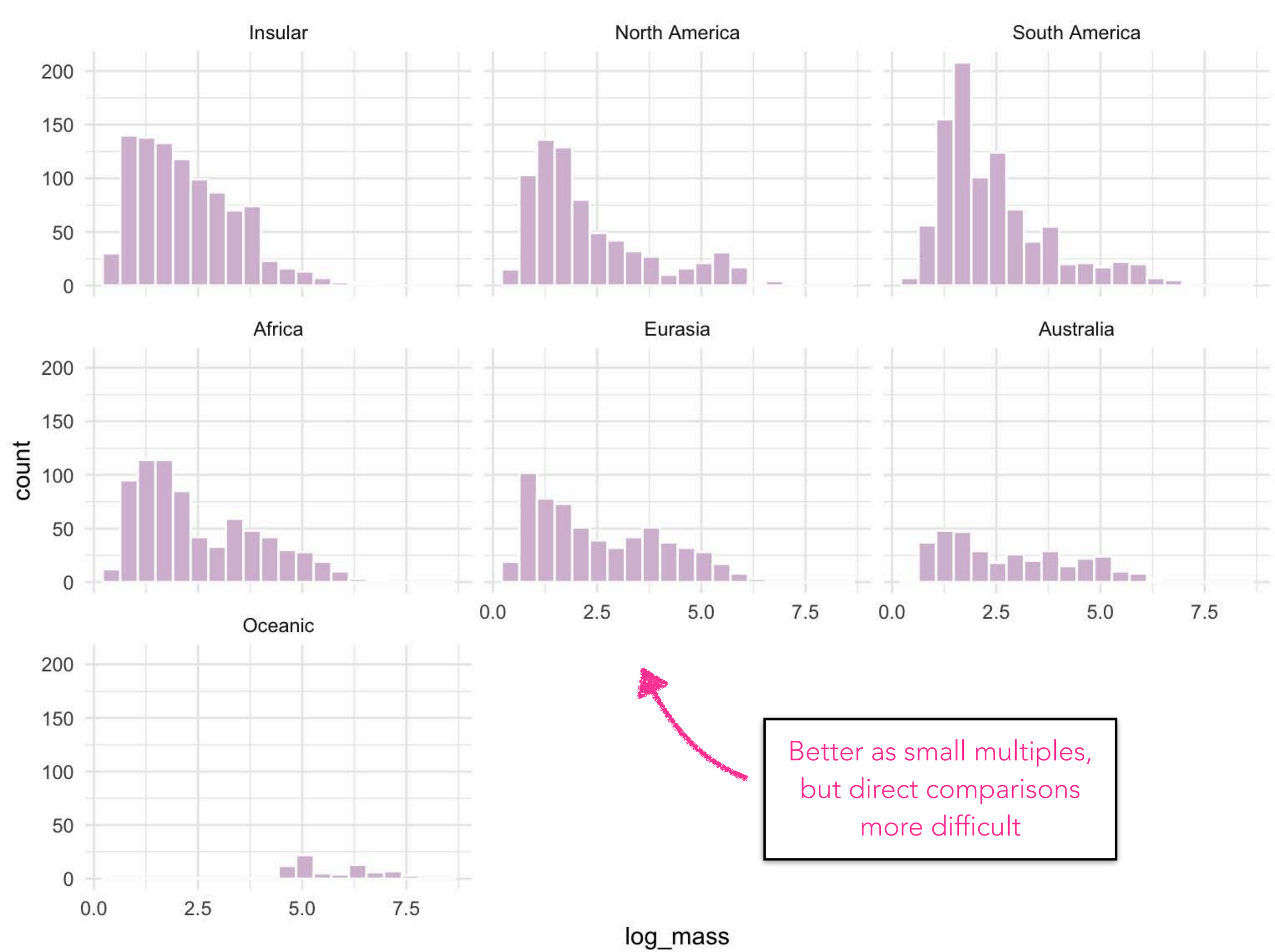
```
ggplot(lqm, aes(x = log_mass, fill = continent)) +
  geom_histogram(color = "white", position = "identity", alpha = 0.5)
```

YOUR TURN



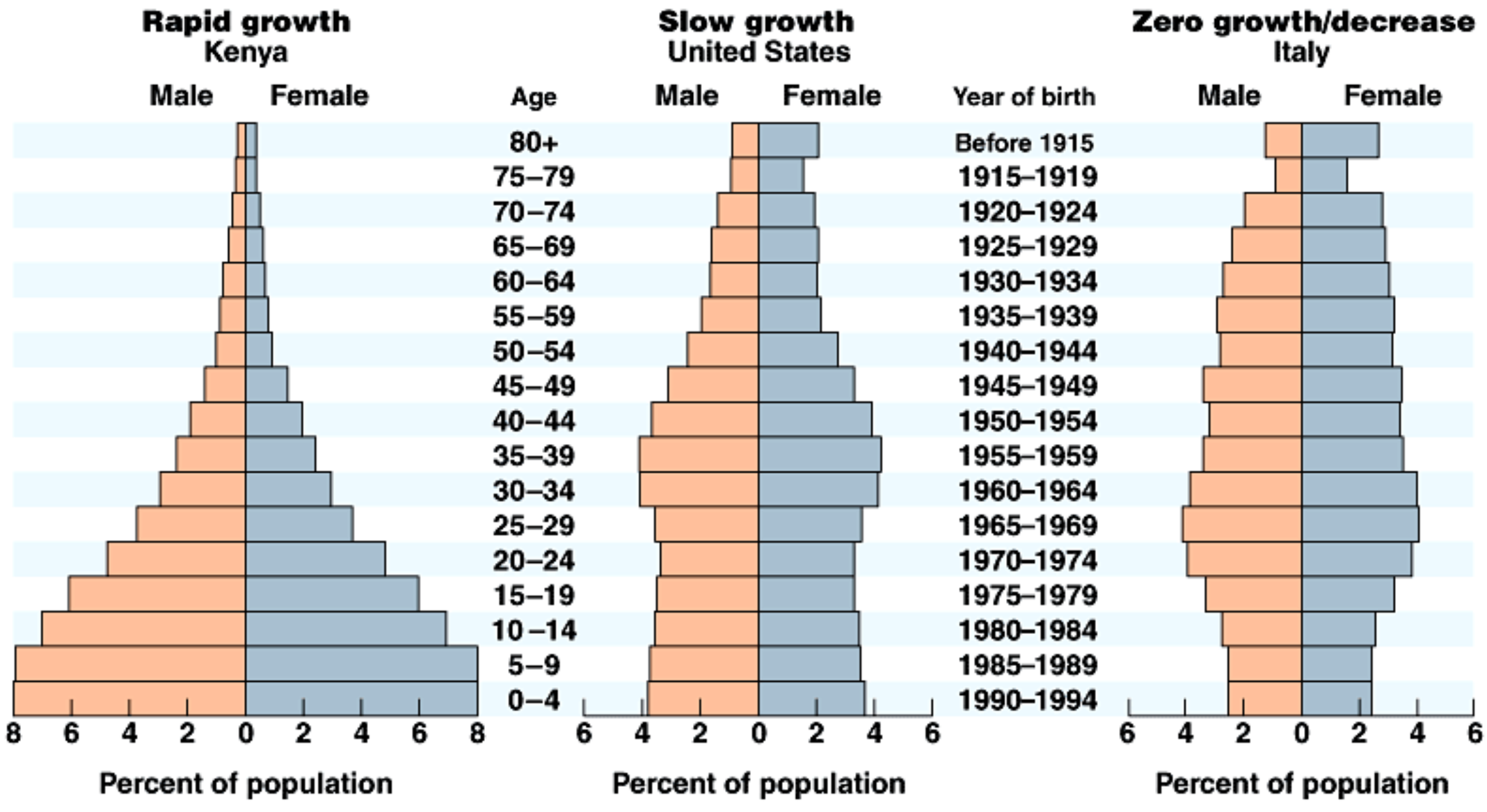

```
ggplot(lqm, aes(x = log_mass, fill = continent)) +
  geom_histogram(color = "white", position = "identity", alpha = 0.5)
```

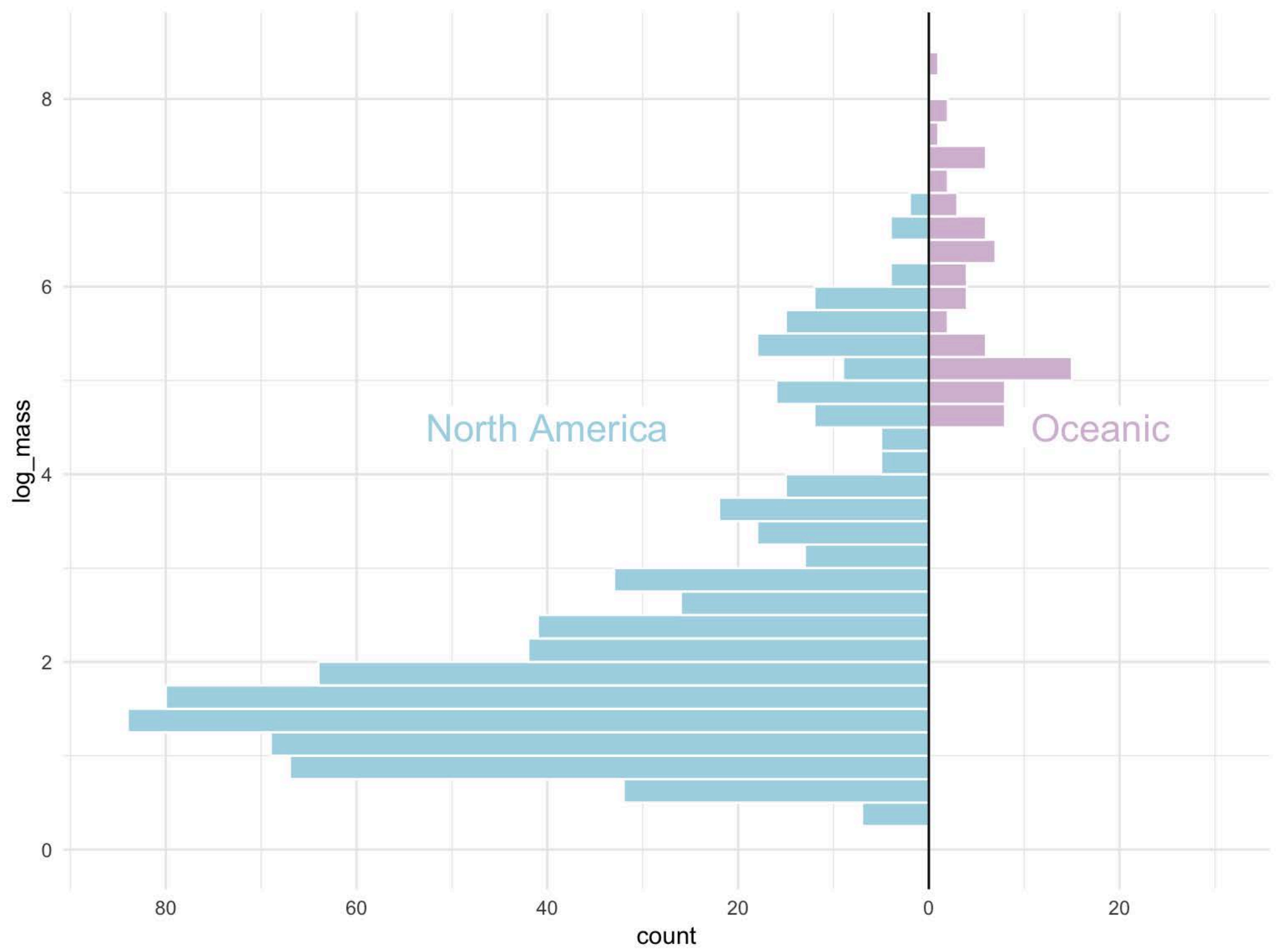

Works okayish with very few groups that are clearly separated
count
100
50
0
0
2
4
6
8
log_mass
continent
North America
Oceanic



Better as small multiples, but direct comparisons more difficult

"Age pyramid" style plot is ideal for comparing two (and only two) distributions




log_mass
8
6
4
2
0
North America
Oceanic
80
60
40
20
0
20
count

Works well even if distributions overlap extensively

North America

Africa

log_mass

6

4

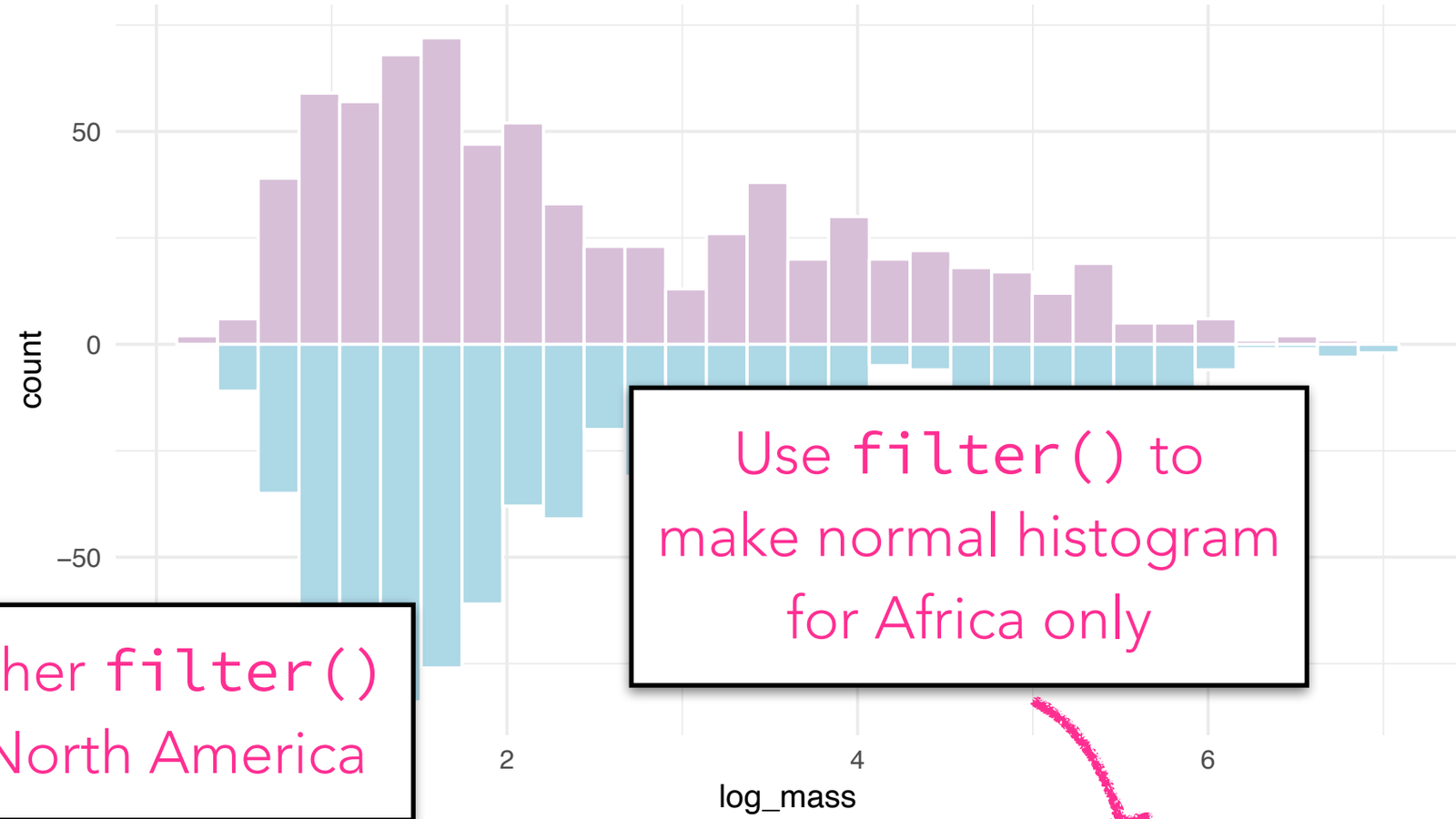
2

0

80 60 40 20 0 20 40 60 80

count

A STARTER...



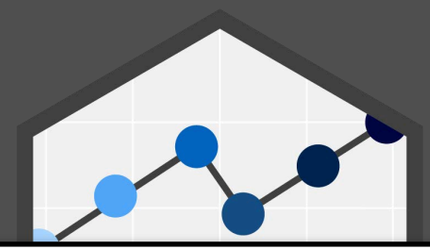
Use `filter()` to make normal histogram for Africa only

Use the `stat()` trick to get counts with flipped sign

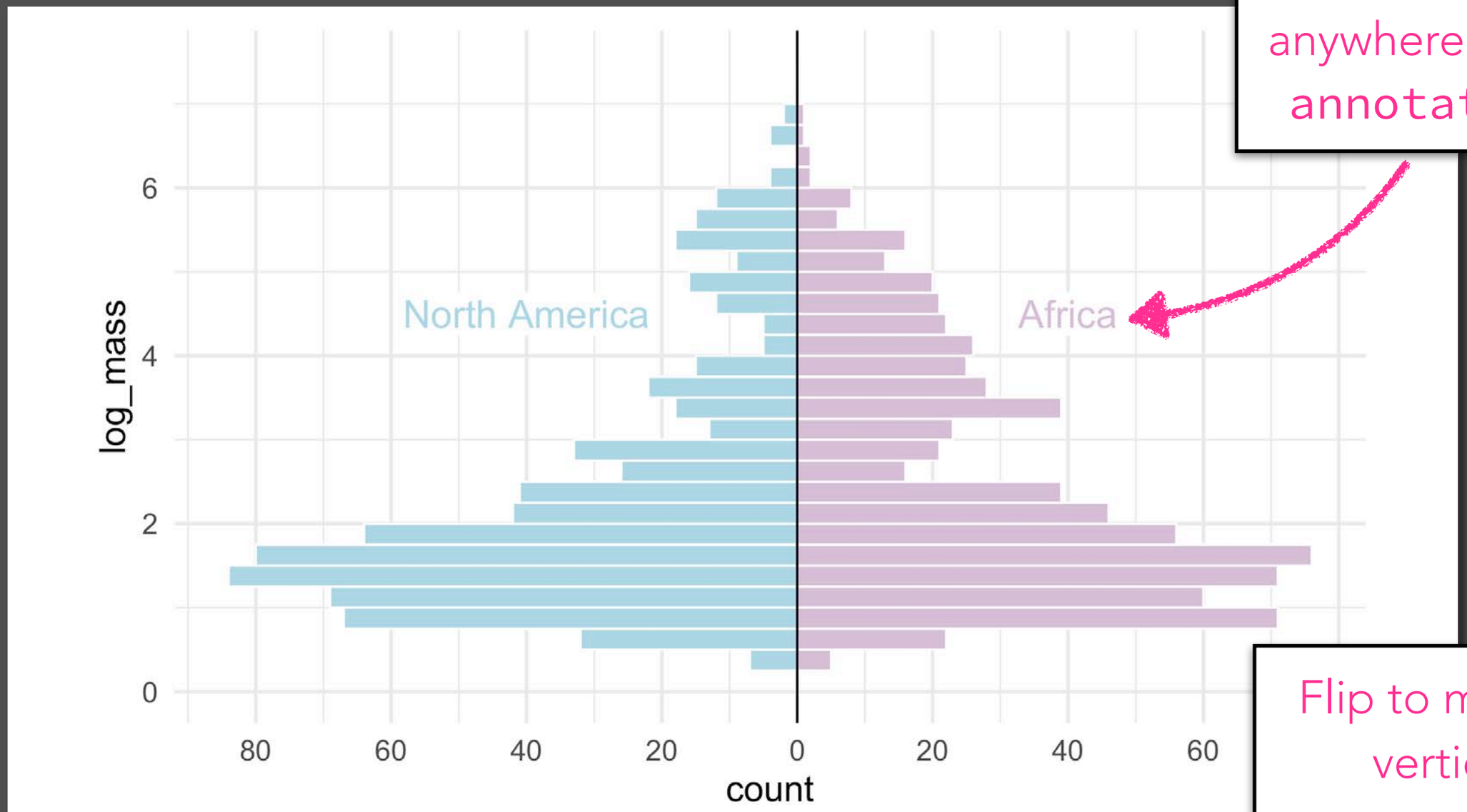
Another `filter()` for North America

```
ggplot() +
  geom_histogram(data = filter(lqm, continent == "Africa"),
                 aes(x = log_mass),
                 fill = "thistle", color = "white") +
  geom_histogram(data = filter(lqm, continent == "North America"),
                 aes(x = log_mass, y = -stat(count)),
                 fill = "lightblue", color = "white")
```

A STARTER...



Add text anywhere using `annotate()`



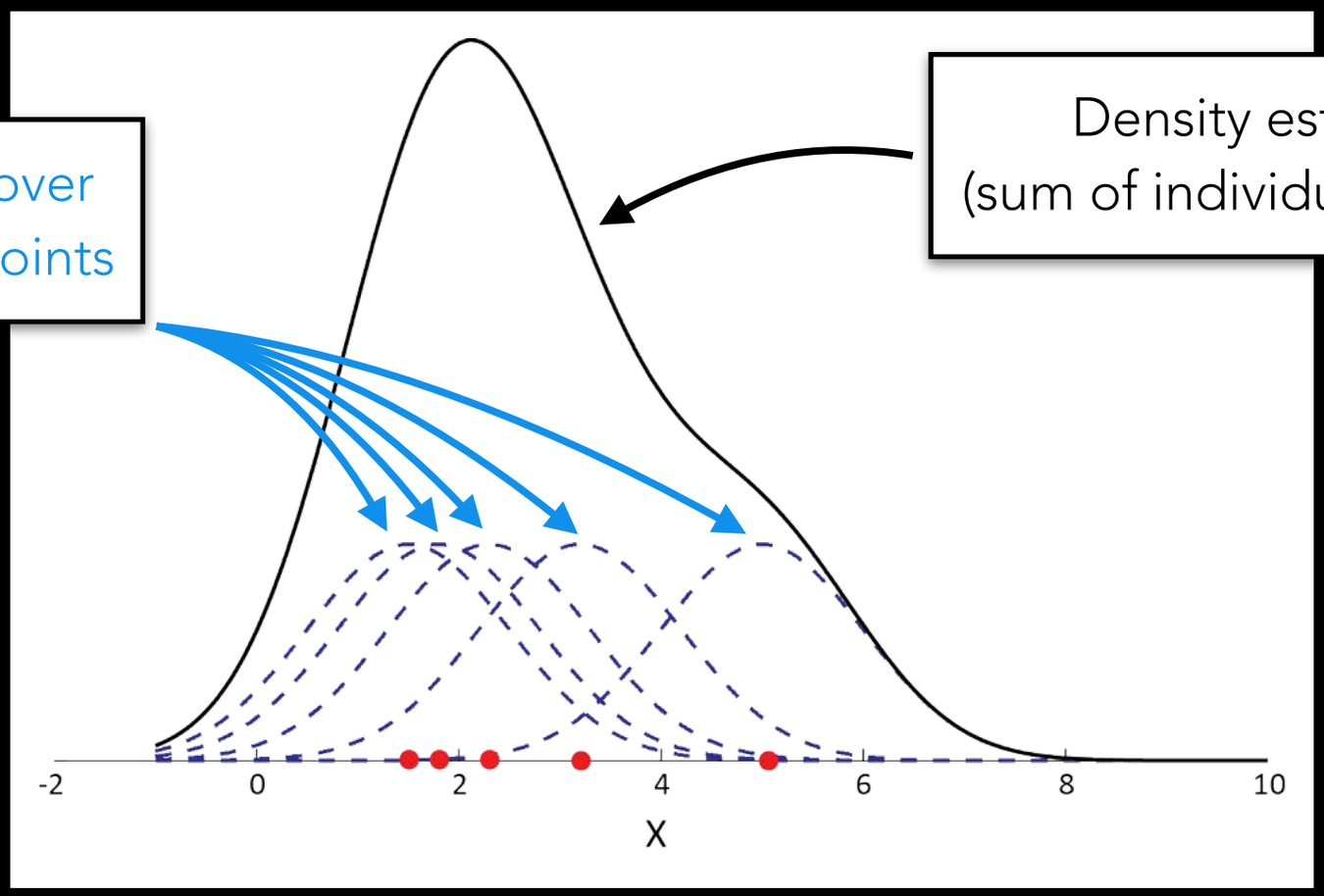
Flip to make it vertical

```
...   +
annotate(geom = "text", y = 40, x = 4.5, label = "Africa",
         color = "thistle", size = 6) +
annotate(geom = "text", y = -40, x = 4.5, label = "North America",
          color = "lightblue", size = 6) +
coord_flip()
```

DENSITY PLOTS

DENSITY PLOTS

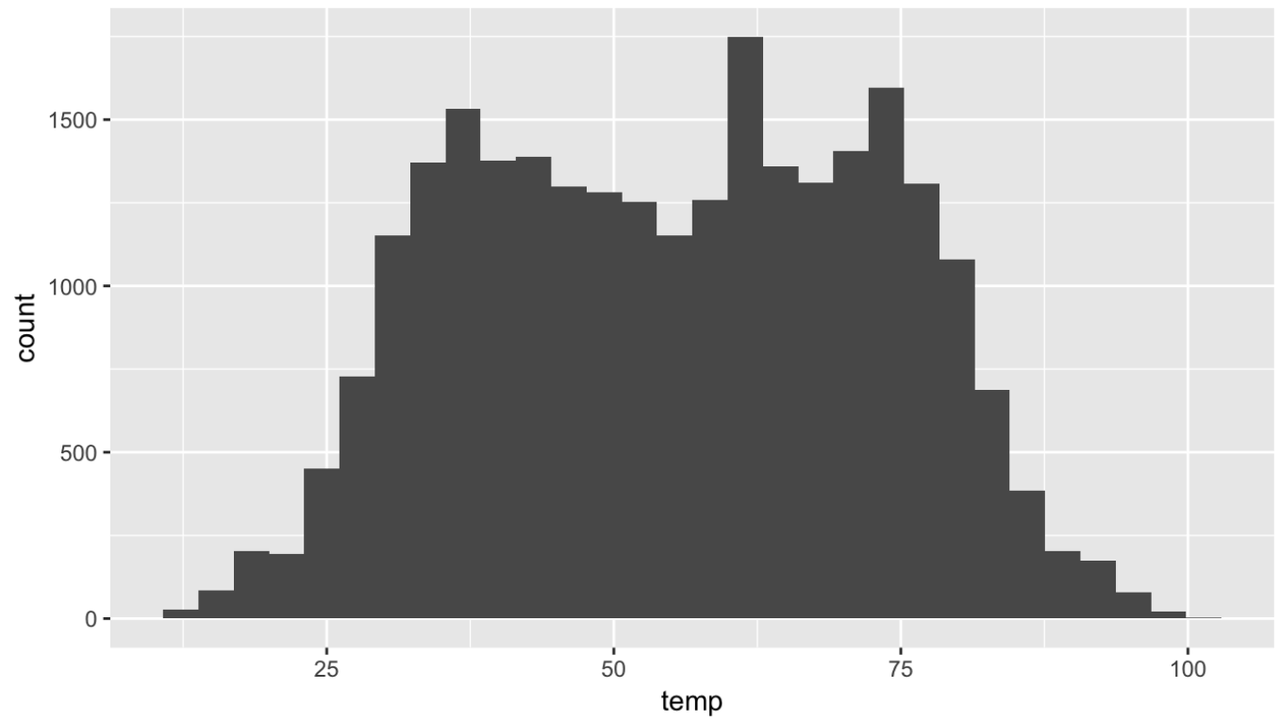
- Like histograms, density plots show the distribution of a single variable
- Uses smoothing to estimate the *density* of points over a *continuous* range of data values.



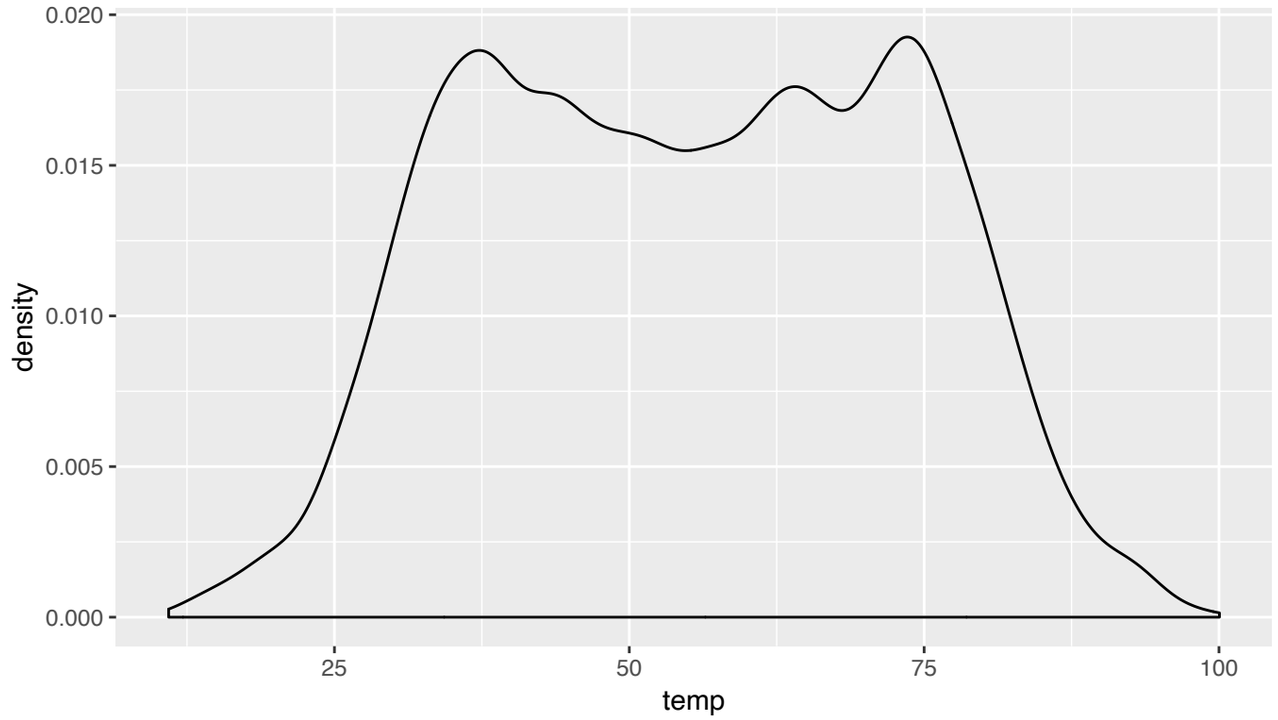
DENSITY PLOTS

- Geometric object is `geom_density()`
- Aesthetic mappings are same as for histogram

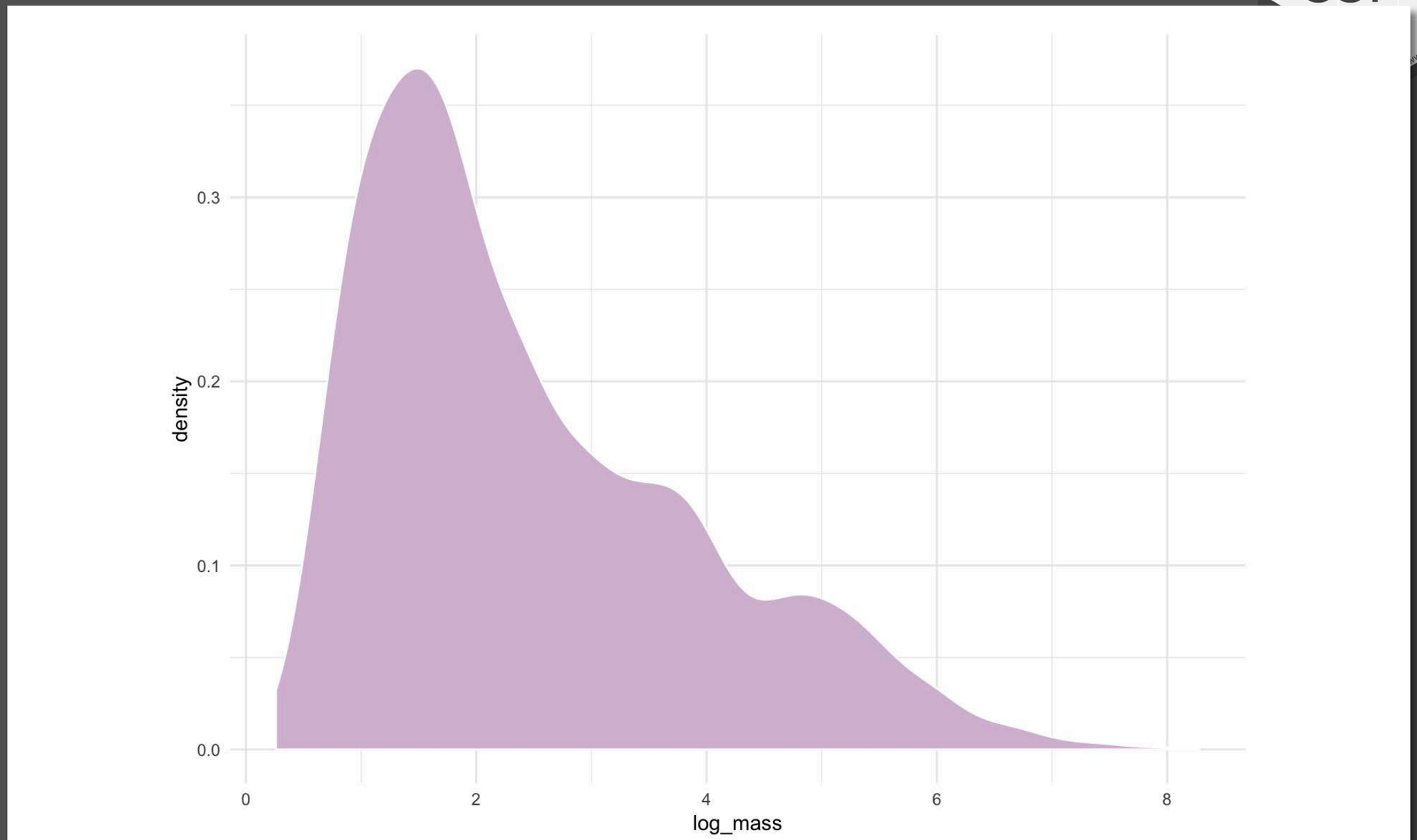
Histogram



Density plot

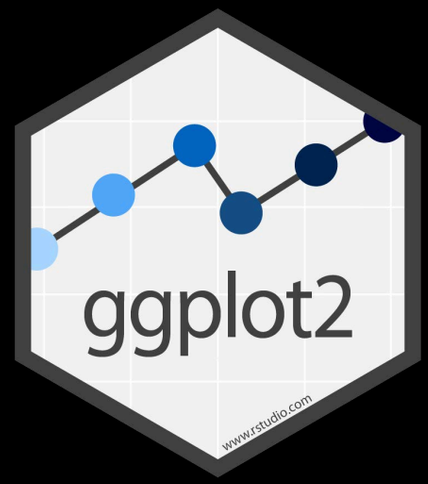


YOUR TURN




```
ggplot(lqm, aes(x = log_mass)) +
  geom_density(fill = "thistle", color = "white")
```

DENSITY PLOTS

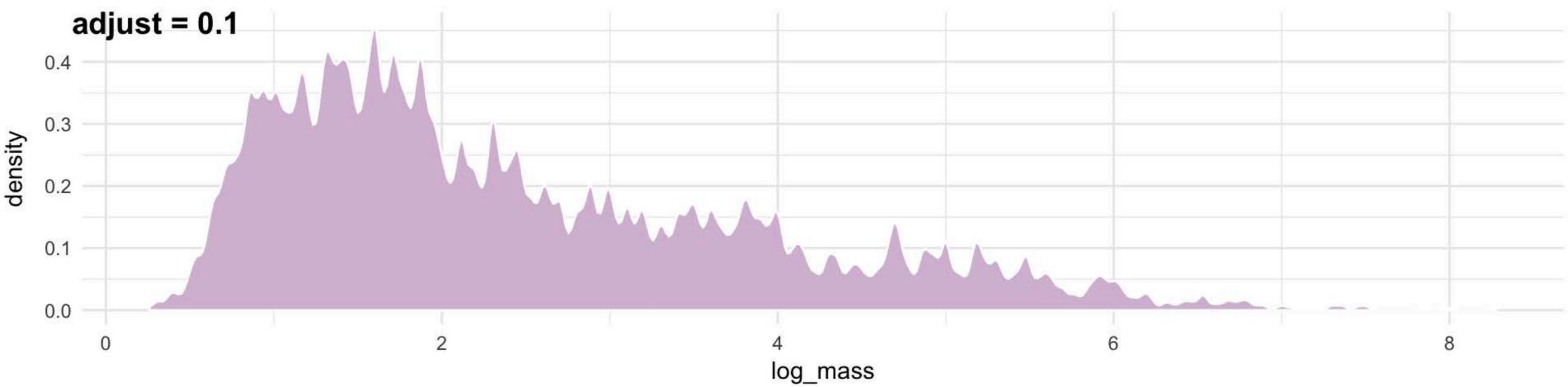


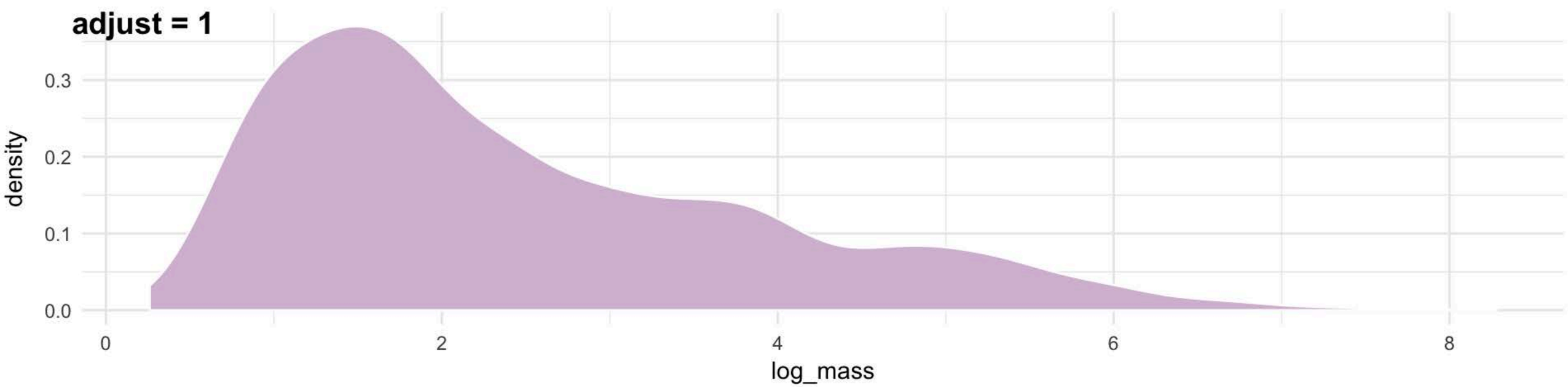
- Like histograms, the visual appearance of density plots depends on the amount of smoothing.

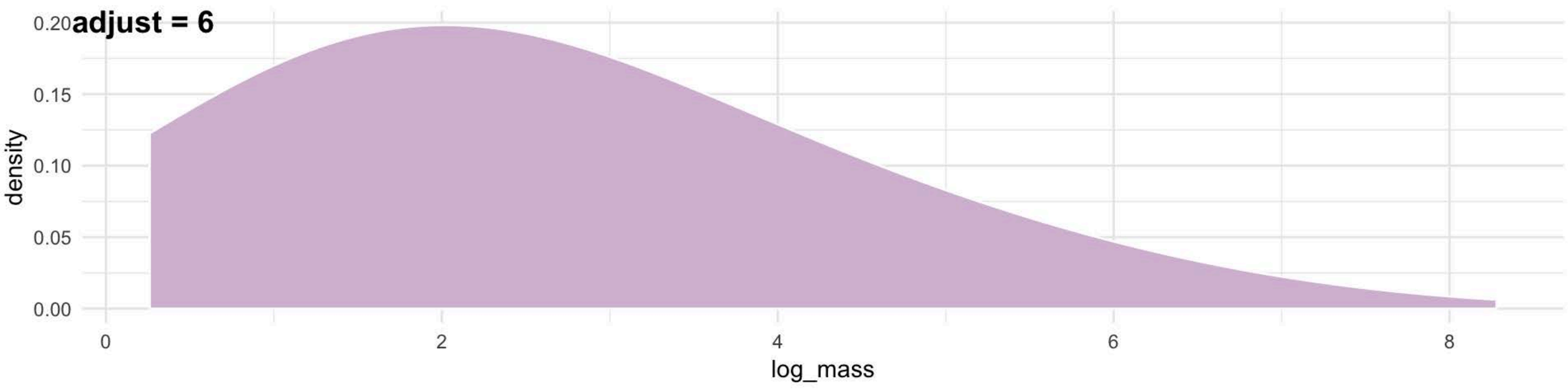
- Smoothing behavior can be adjusted in various complex ways, but the only one you're likely to use is:

  - adjust: `geom_density(adjust = 1)`

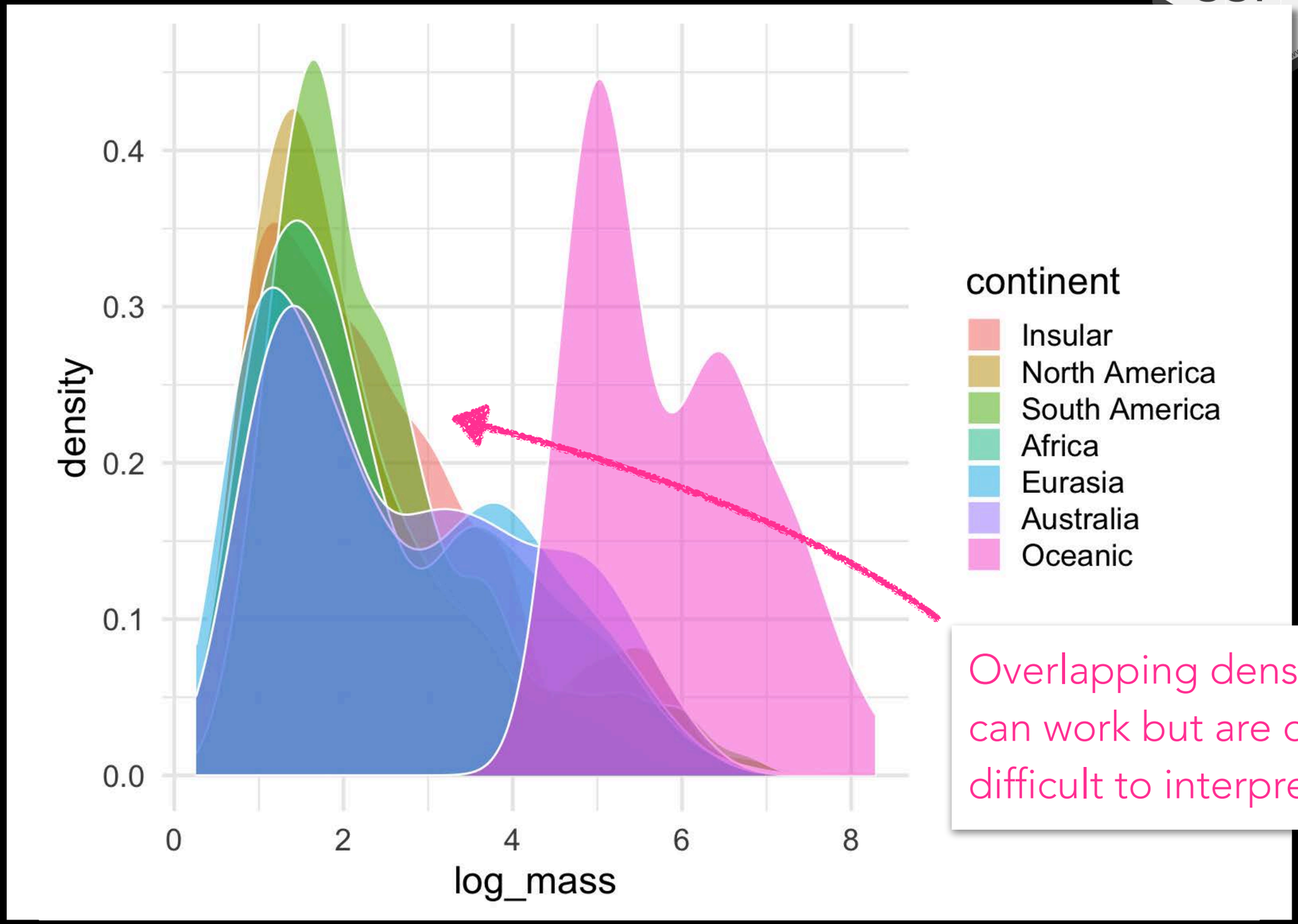
Multiplier for the amount of smoothing (1 = default)


adjust = 0.1
density
0.4
0.3
0.2
0.1
0.0
0
2
4
6
8
log_mass


adjust = 1
density
0.3
0.2
0.1
0.0
0
2
4
6
8
log_mass

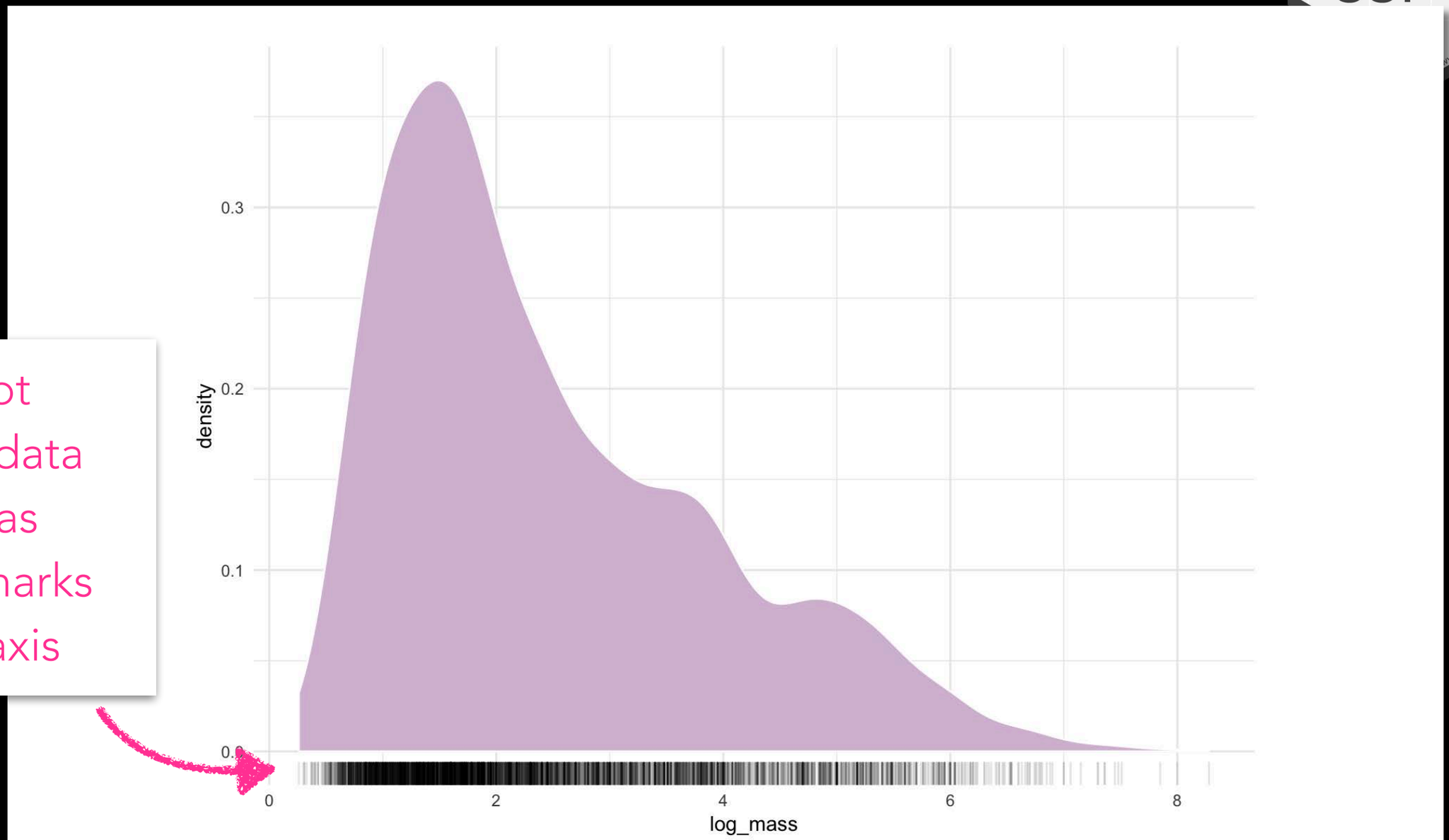

adjust = 6
density
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
log_mass

DENSITY PLOTS



Overlapping densities can work but are often difficult to interpret

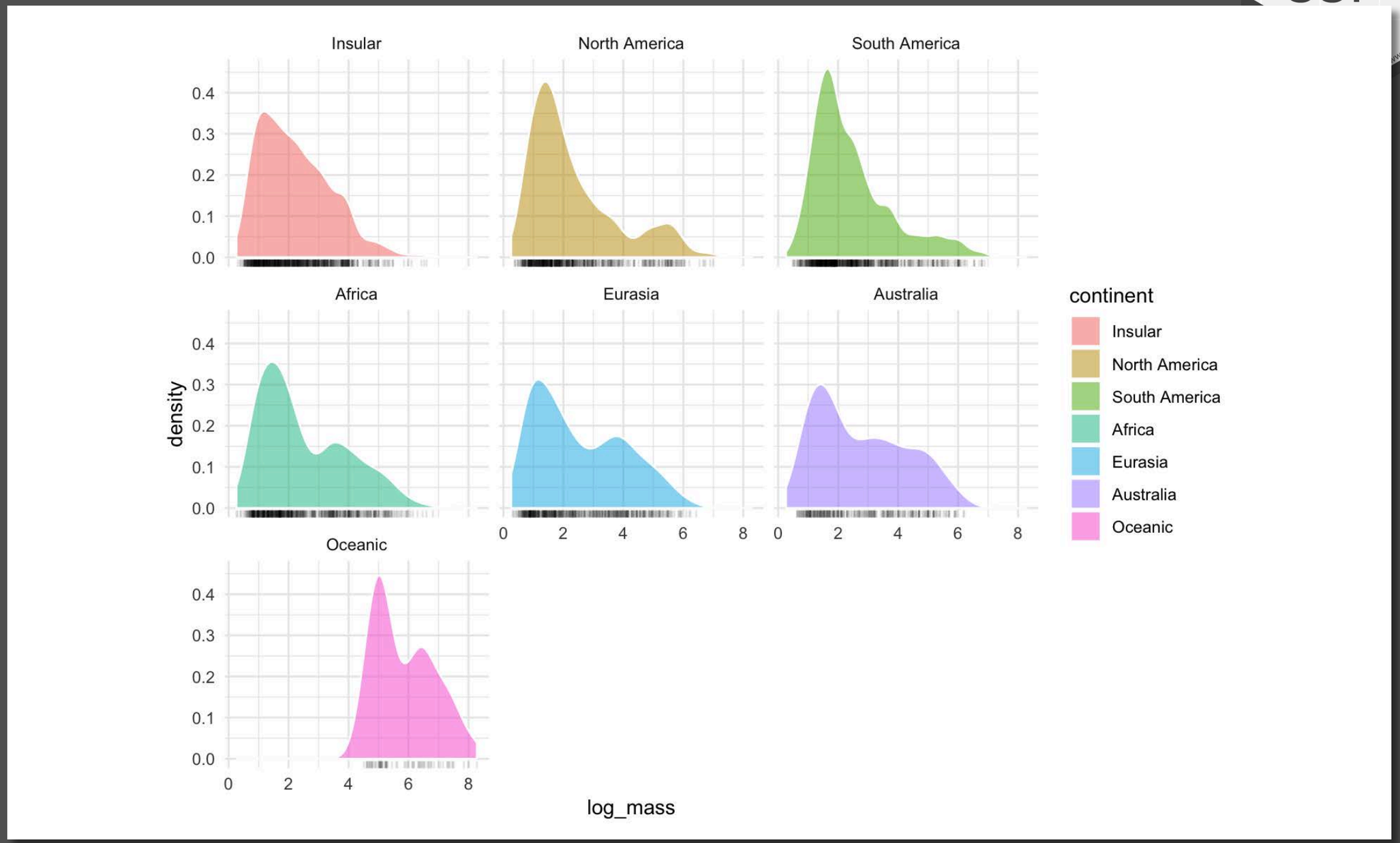
DENSITY PLOTS



Rug plot shows data points as small marks along axis

```
ggplot(lqm, aes(x = log_mass)) +
  geom_density(fill = "thistle", color = "white") +
  geom_rug(alpha = 0.1)
```

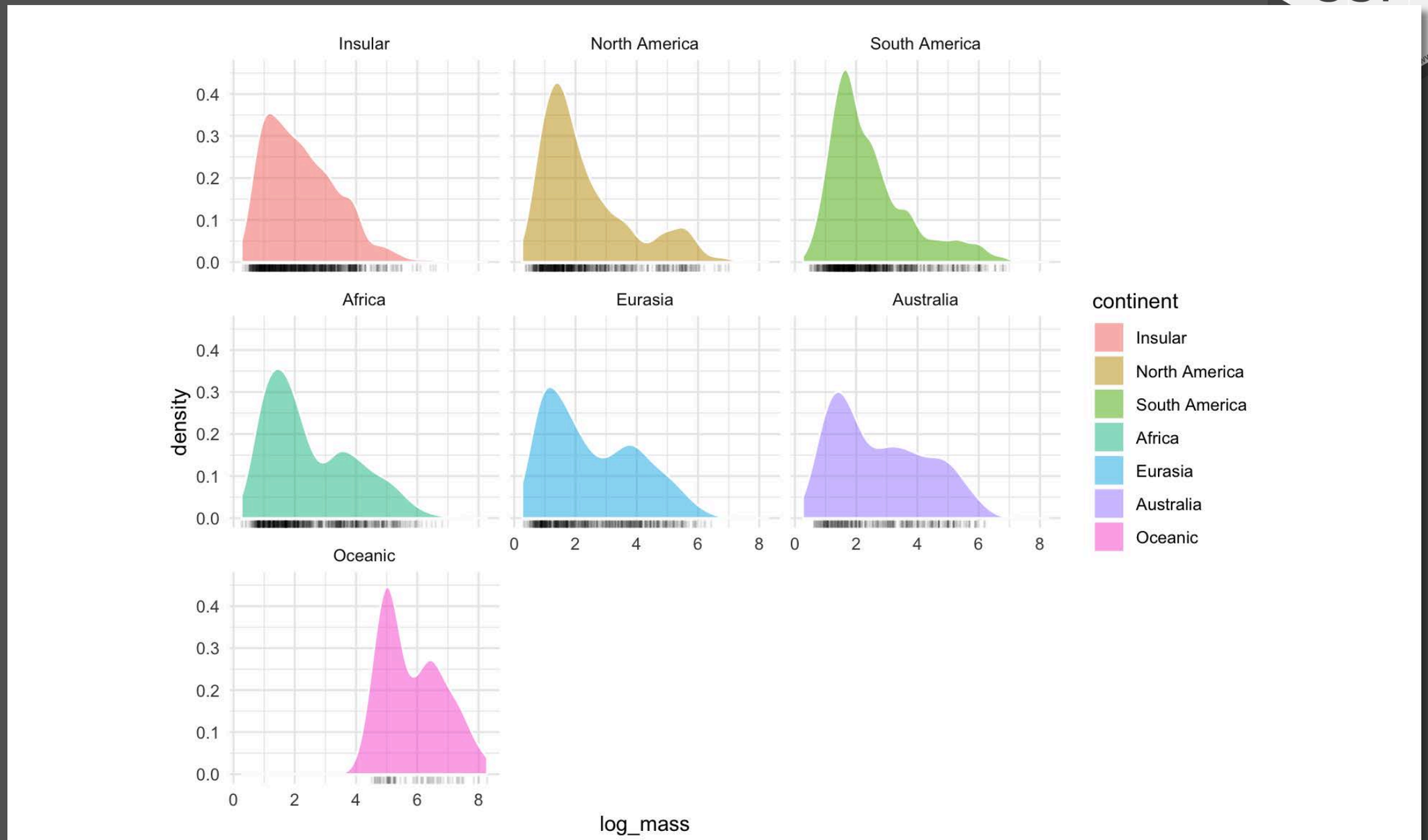
YOUR TURN



Try to reverse-engineer this plot.

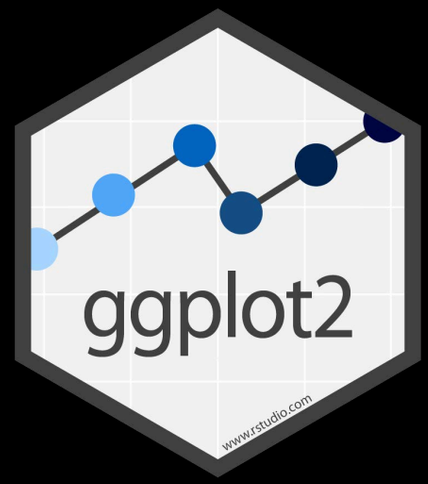
5:00

YOUR TURN




```
ggplot(lqm, aes(x = log_mass, fill = continent)) +
  geom_density(color = NA, alpha = 0.5) +
  geom_rug(alpha = 0.1) +
  facet_wrap(vars(continent))
```

HISTOGRAMS VS. DENSITY PLOTS

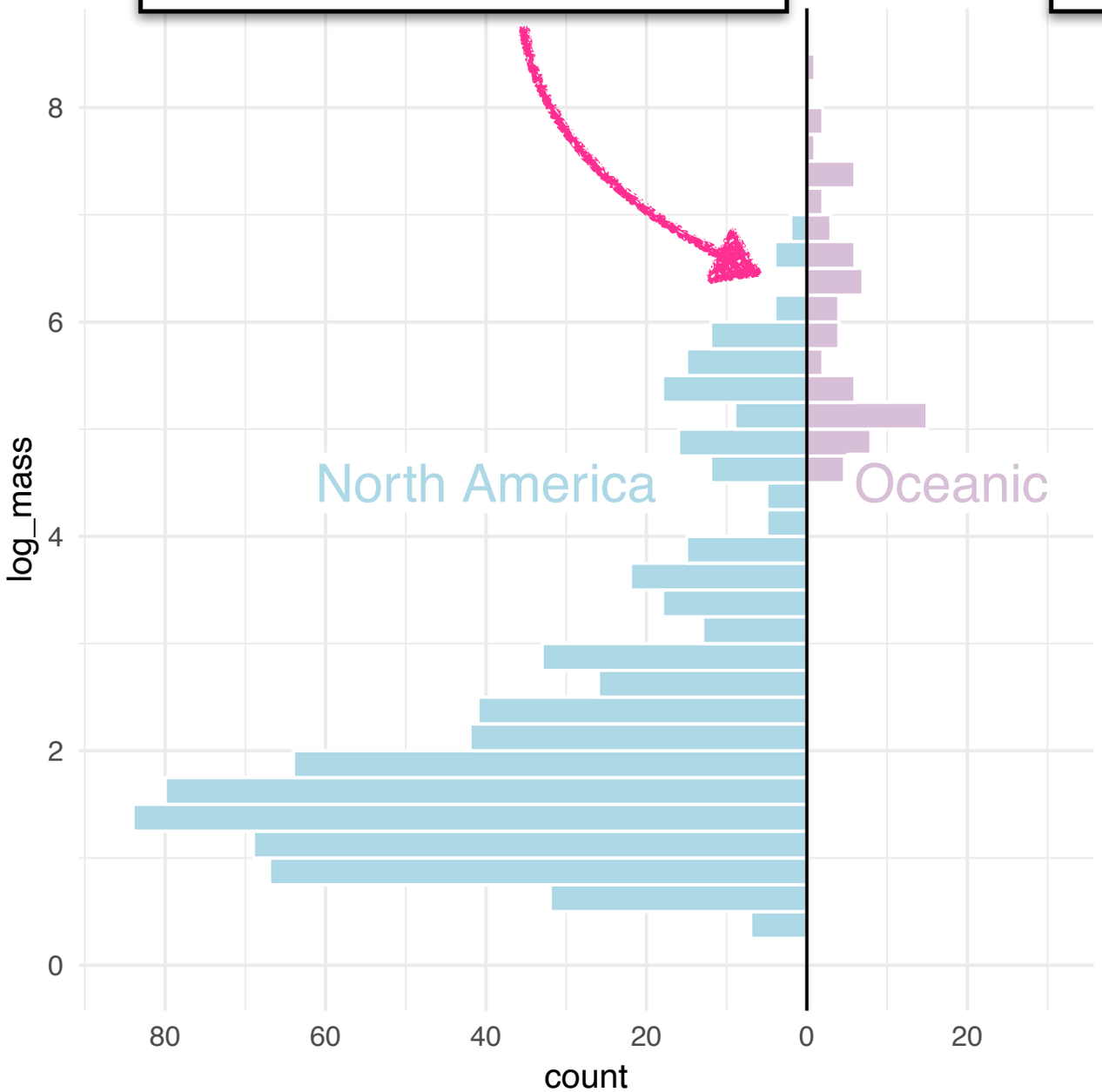


- Histograms are much more common, and are therefore more familiar to a wide audience

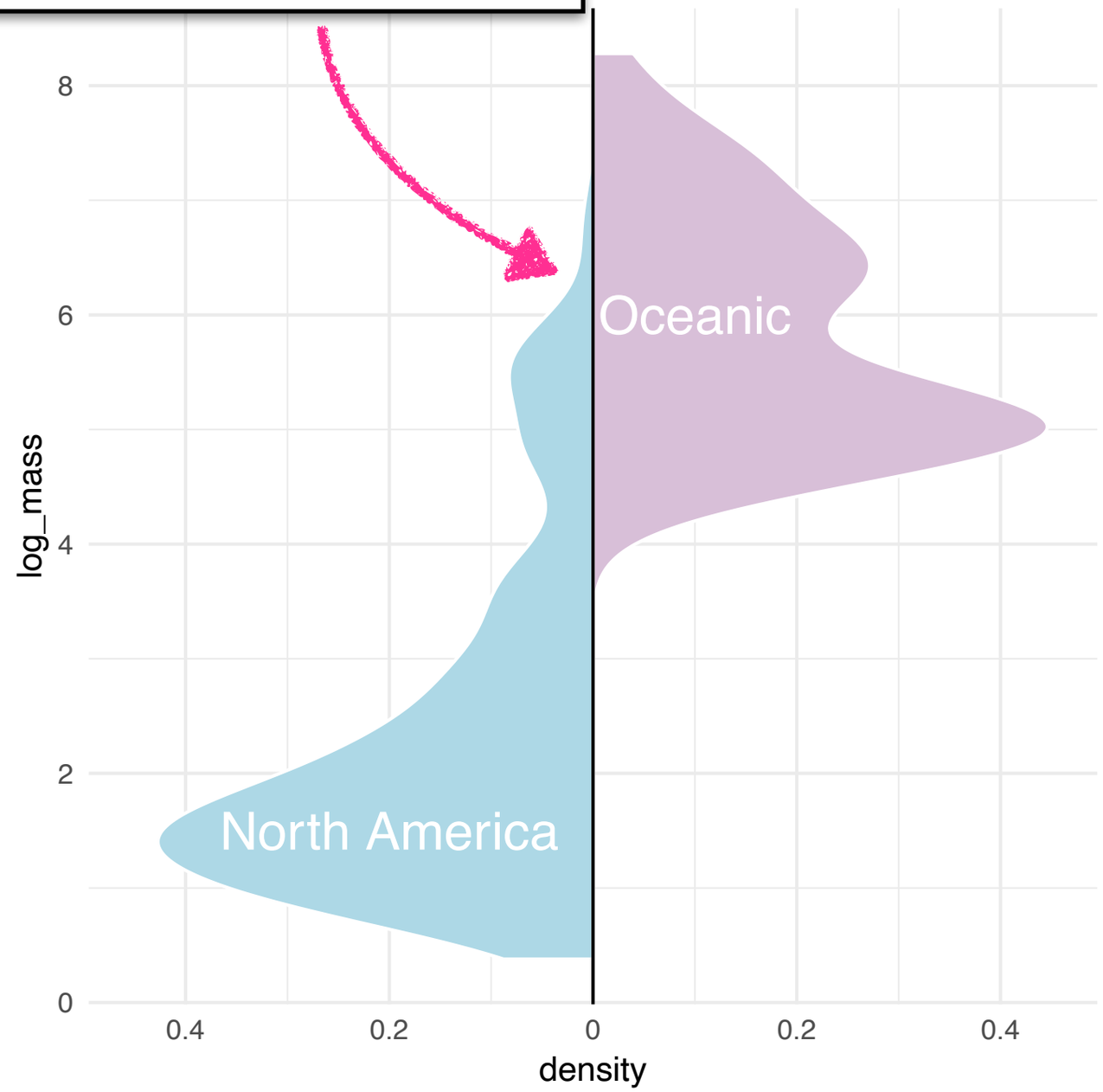
- Histograms use a more intuitive stat (count vs. density)

- Whereas the appearance of a histograms depends on sample size, densities are *scaled*

Smaller sample size for "Oceanic" means bars are small

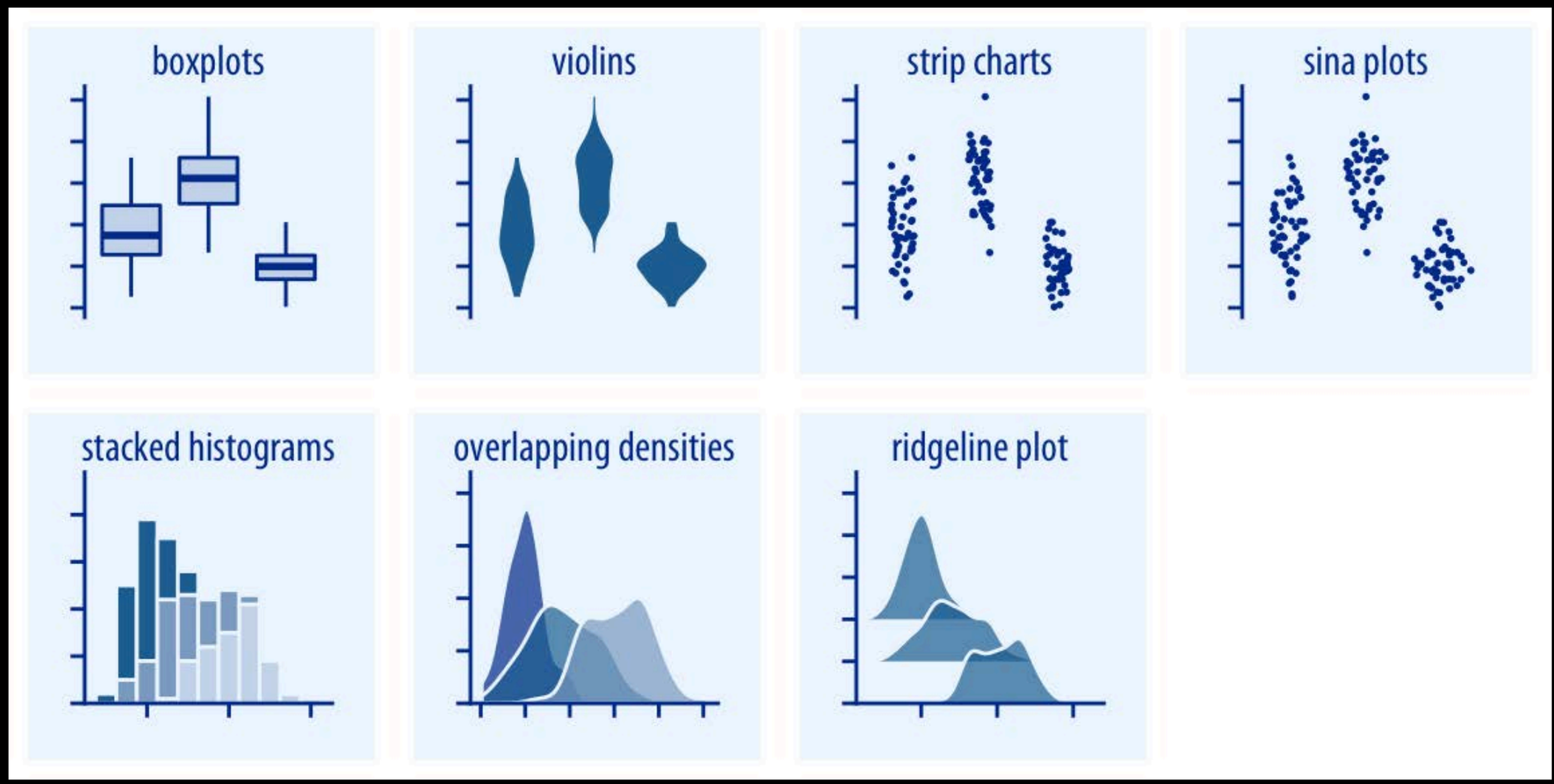


Densities have equal areas (area under the curve sums to 1)

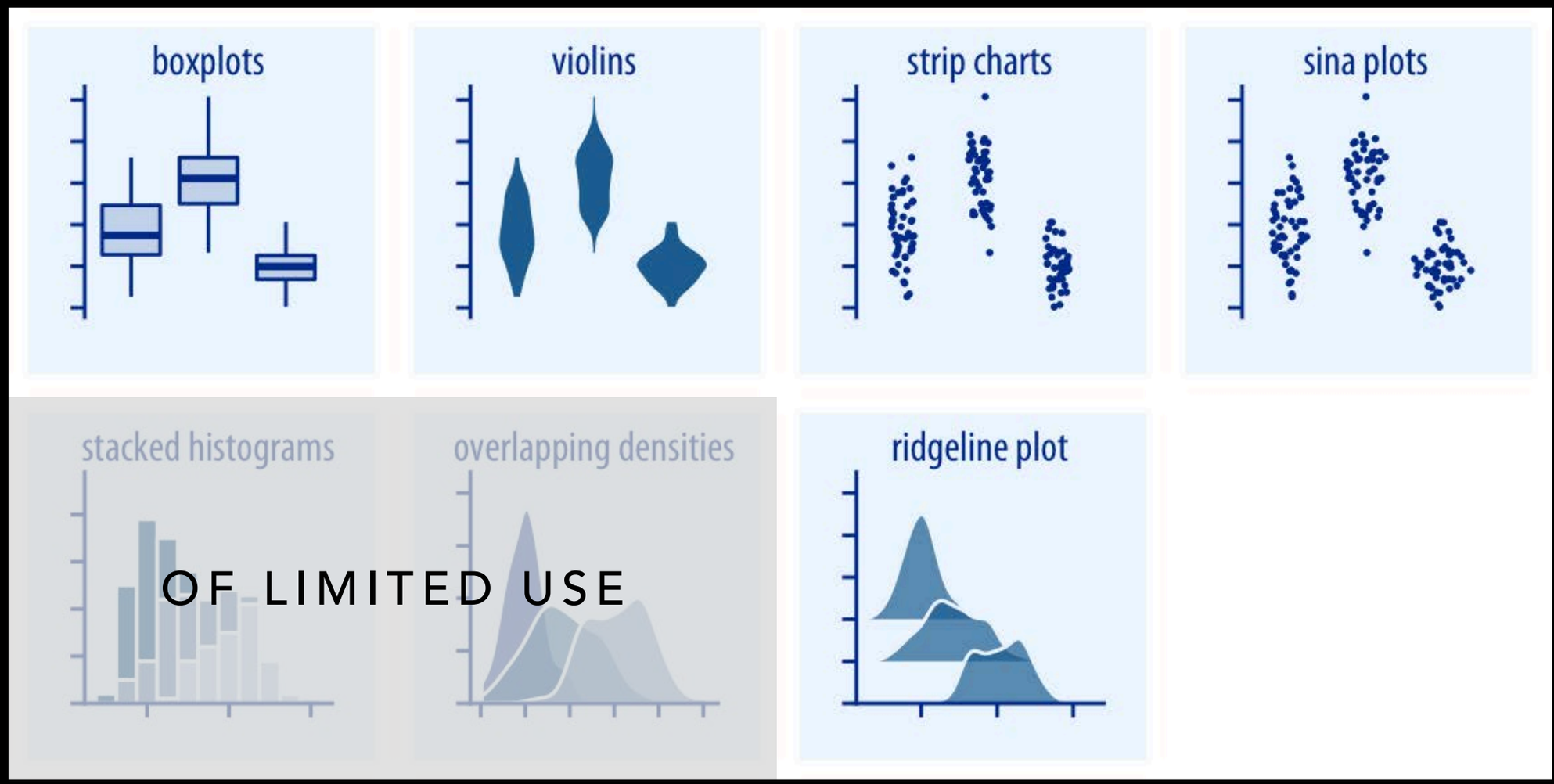


WHAT IF YOU NEED TO SHOW MULTIPLE DISTRIBUTIONS TOGETHER?

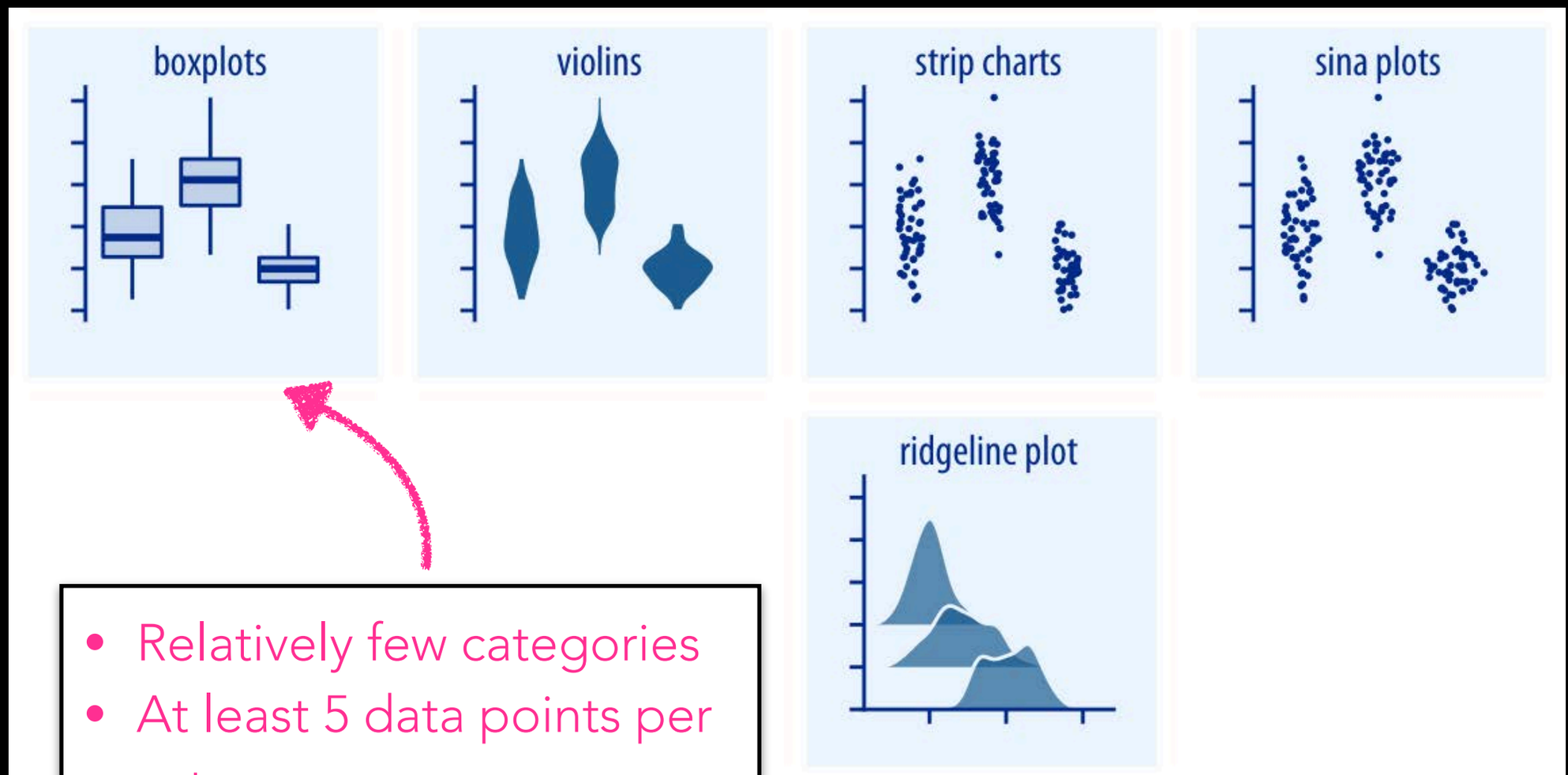
WHAT IF YOU NEED TO SHOW MULTIPLE DISTRIBUTIONS TOGETHER?



WHAT IF YOU NEED TO SHOW MULTIPLE DISTRIBUTIONS TOGETHER?

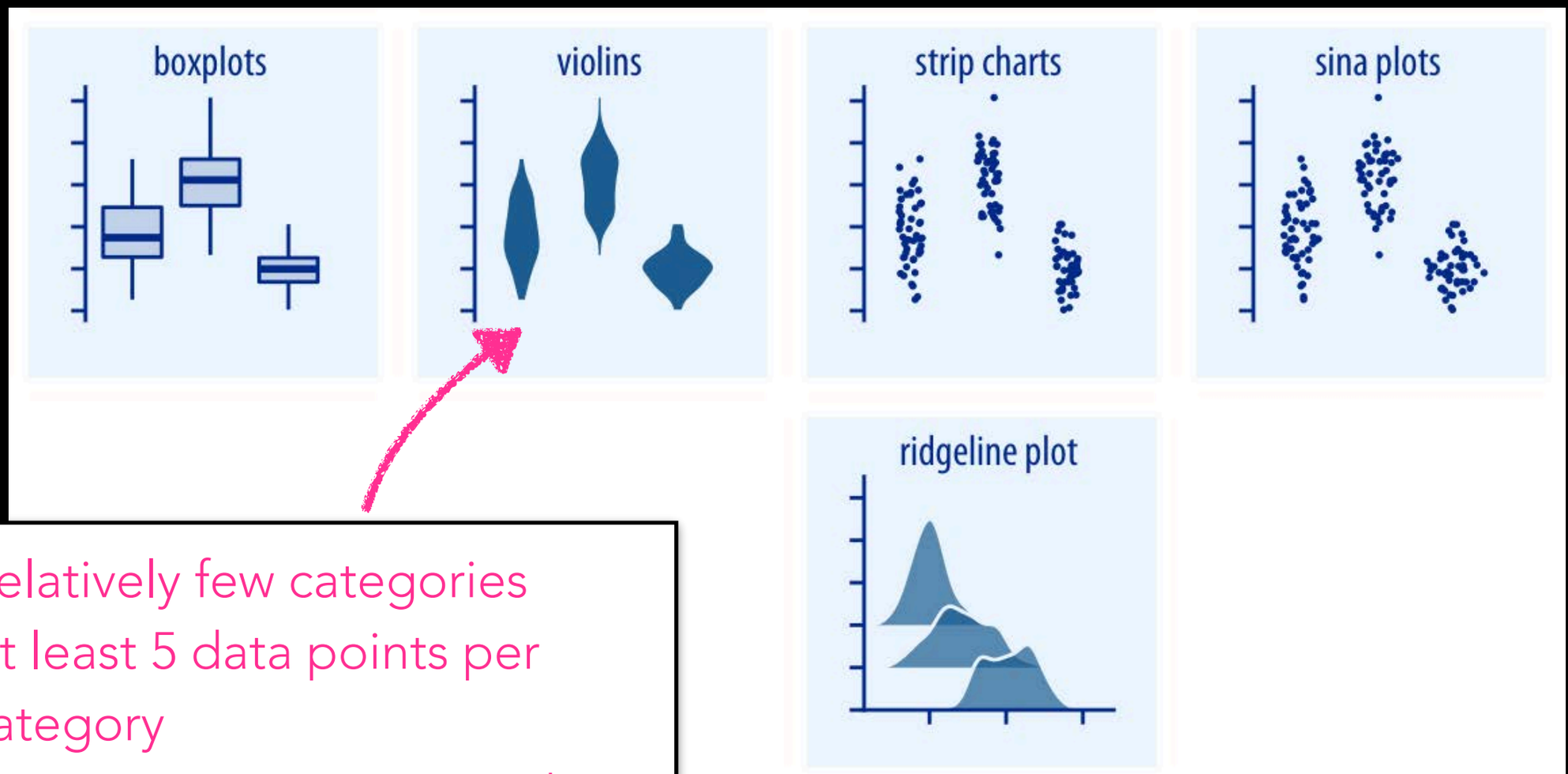


USAGE GUIDELINES



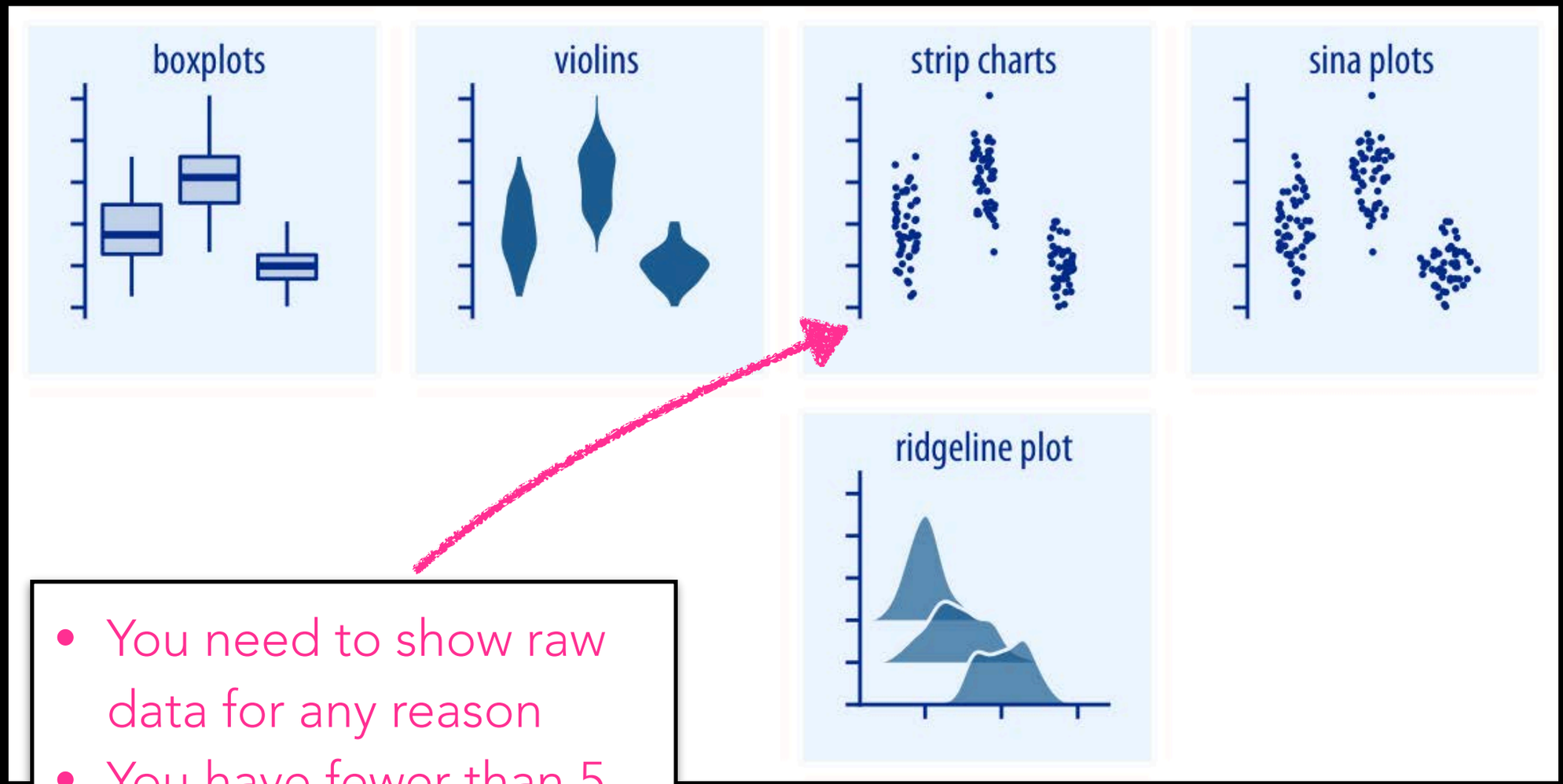
- Relatively few categories
- At least 5 data points per category
- Statistical comparisons are important

USAGE GUIDELINES



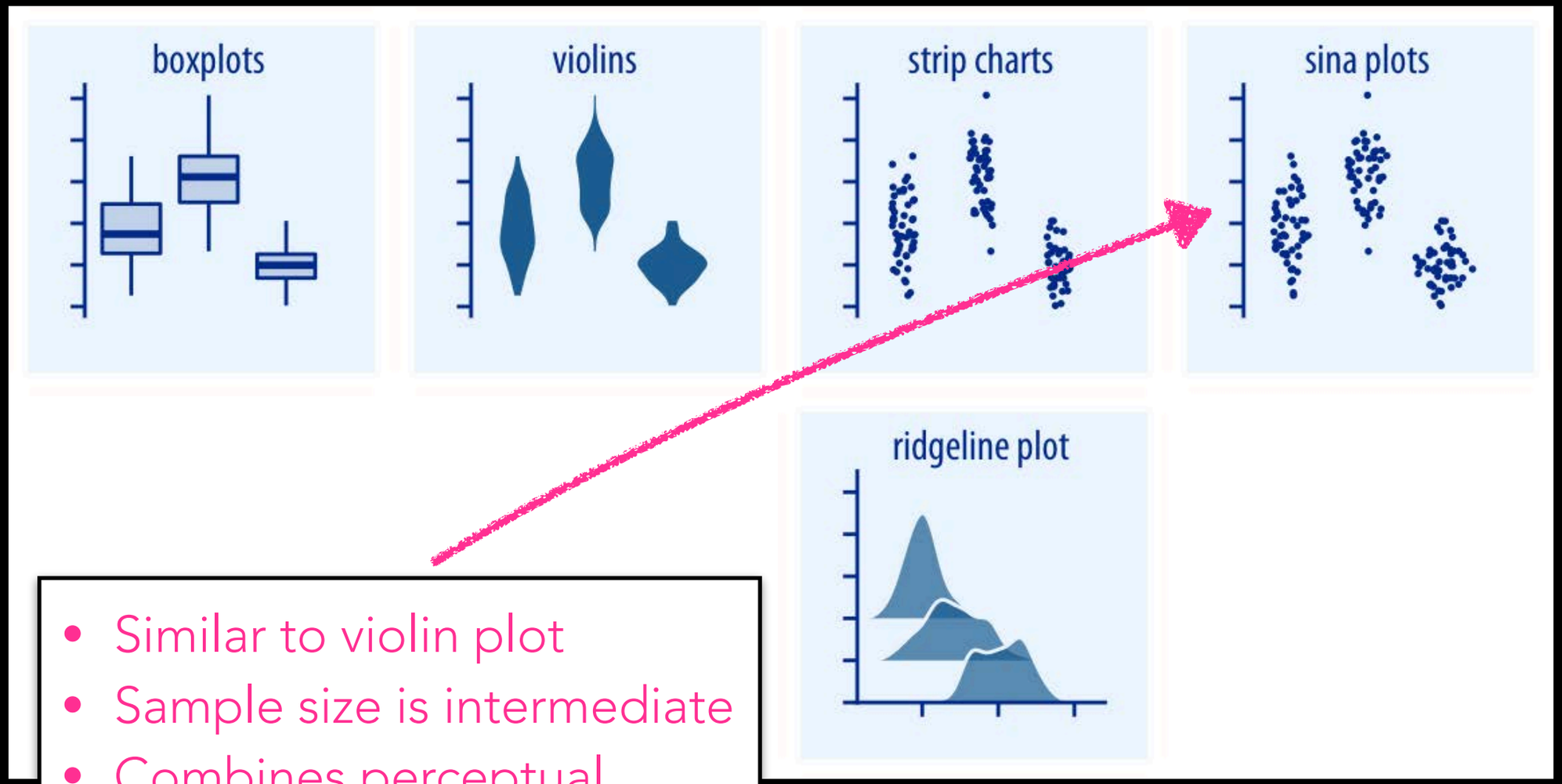
- Relatively few categories
- At least 5 data points per category
- You want a more nuanced comparison of the *shapes* of the distributions rather than statistical values

USAGE GUIDELINES



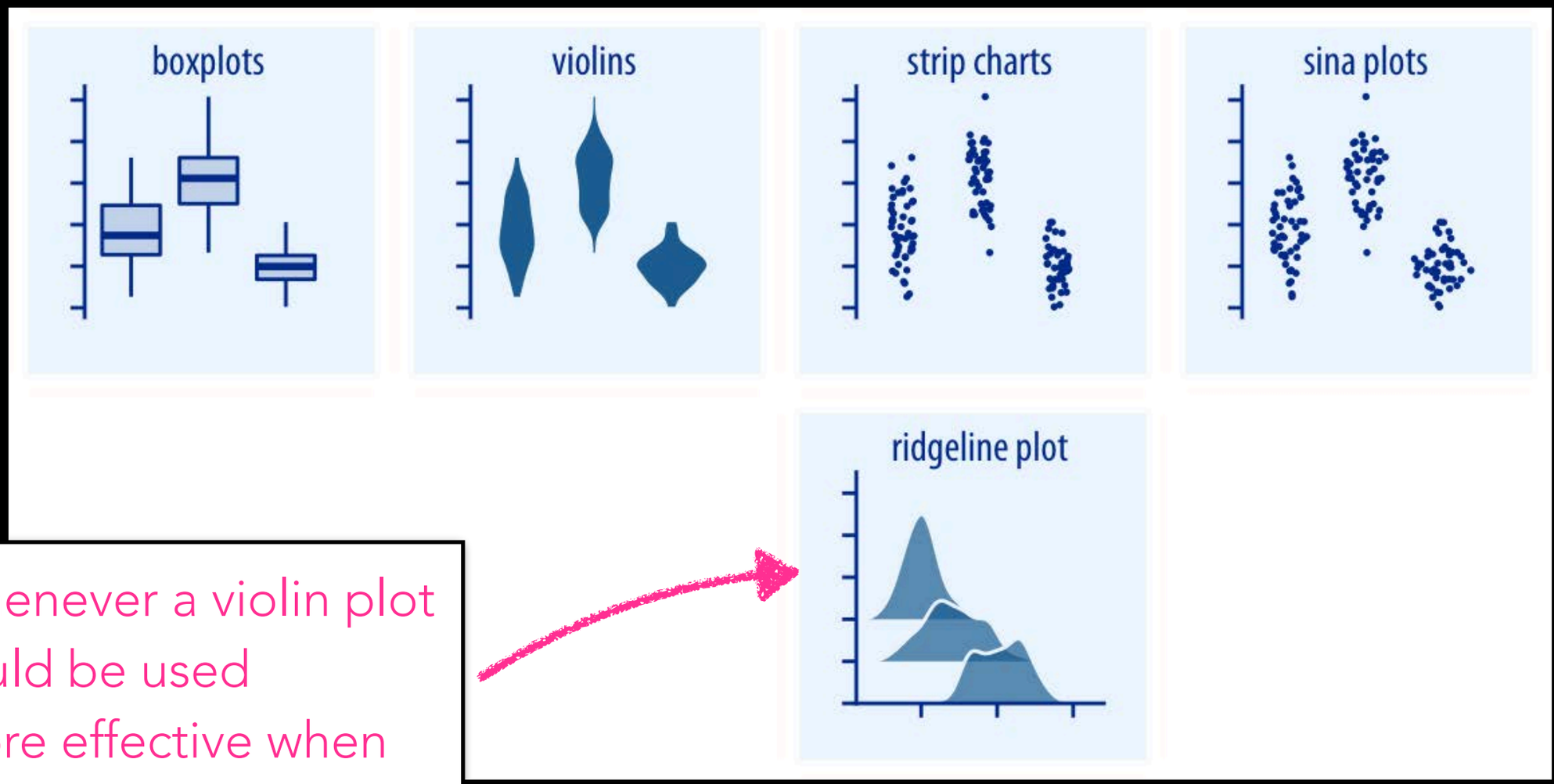
- You need to show raw data for any reason
- You have fewer than 5 data points in some categories

USAGE GUIDELINES



- Similar to violin plot
- Sample size is intermediate
- Combines perceptual properties of violins and strip charts

USAGE GUIDELINES

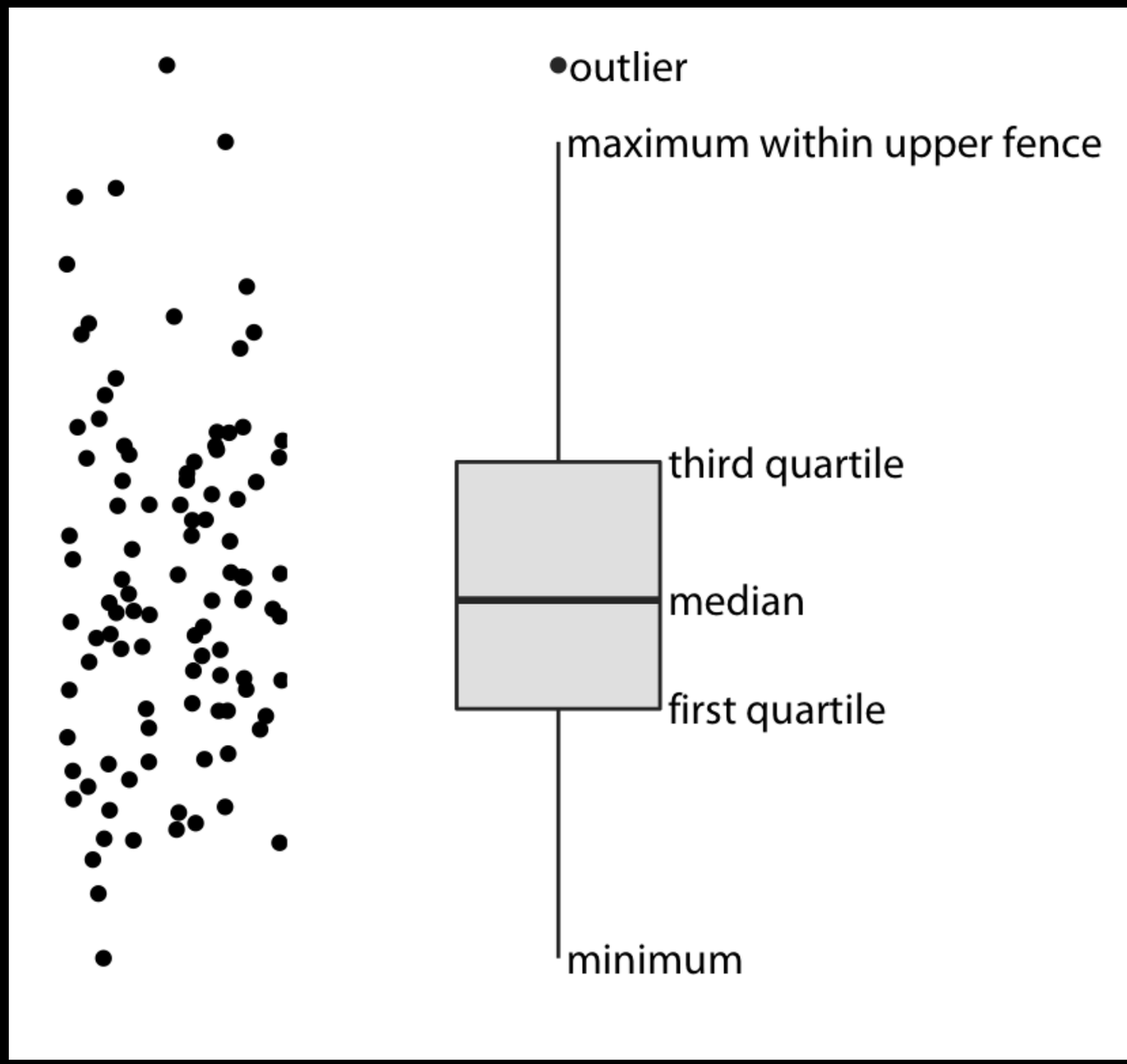


- Whenever a violin plot could be used
- More effective when there are *many* categories that you want to compare

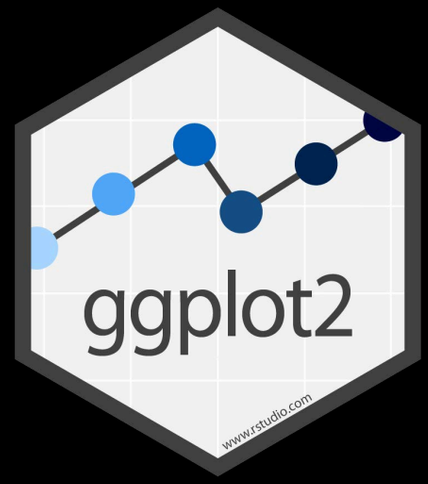
BOXPLOTS

BOXPLOTS

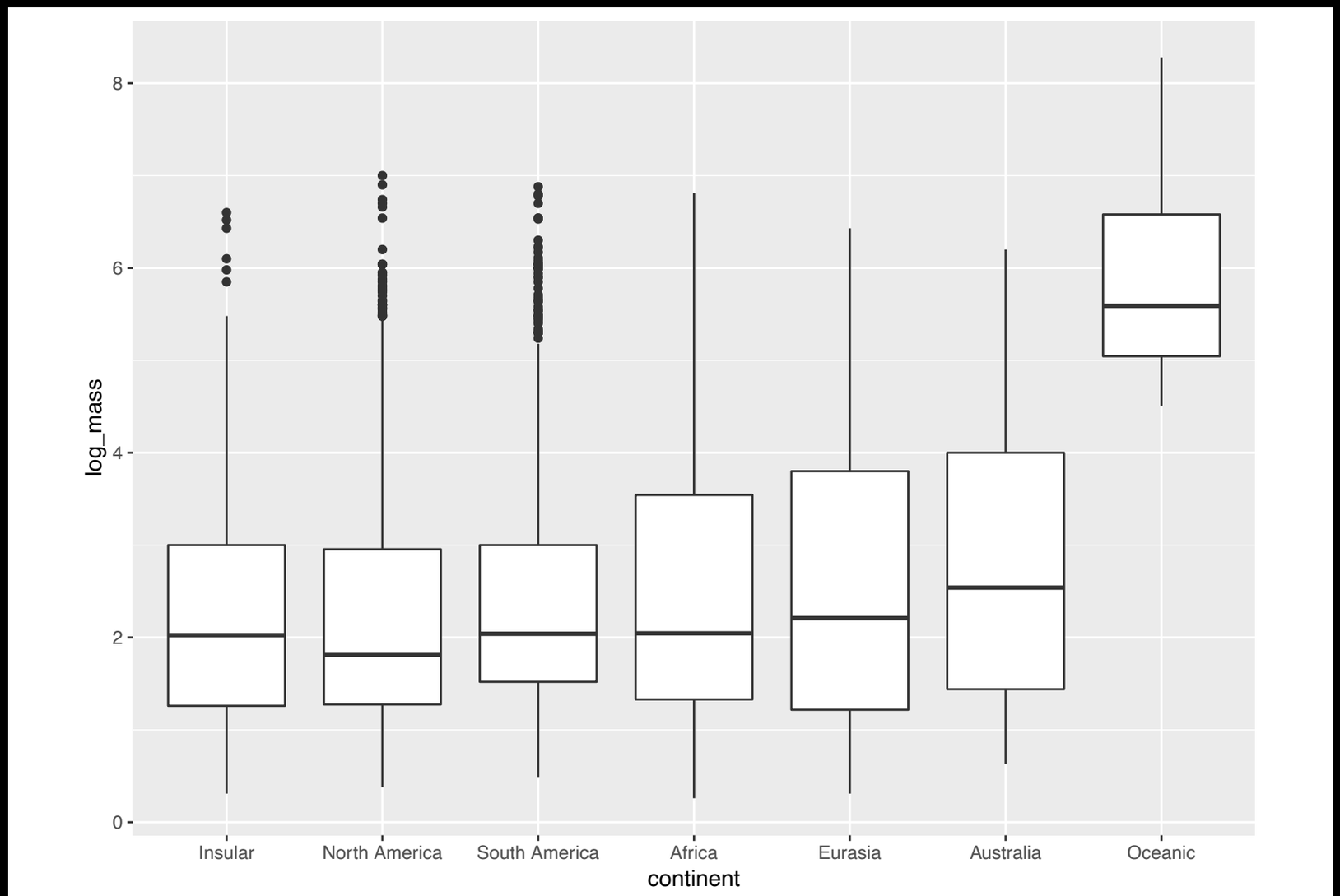
- Each box plot shows a standardized set of 5 statistical summaries of the distribution, as well as individual outliers
- Very common in scientific lit
- But beware of using with lay audience (e.g., “what does the line sticking out of the top show exactly?”)



BOXPLOTS

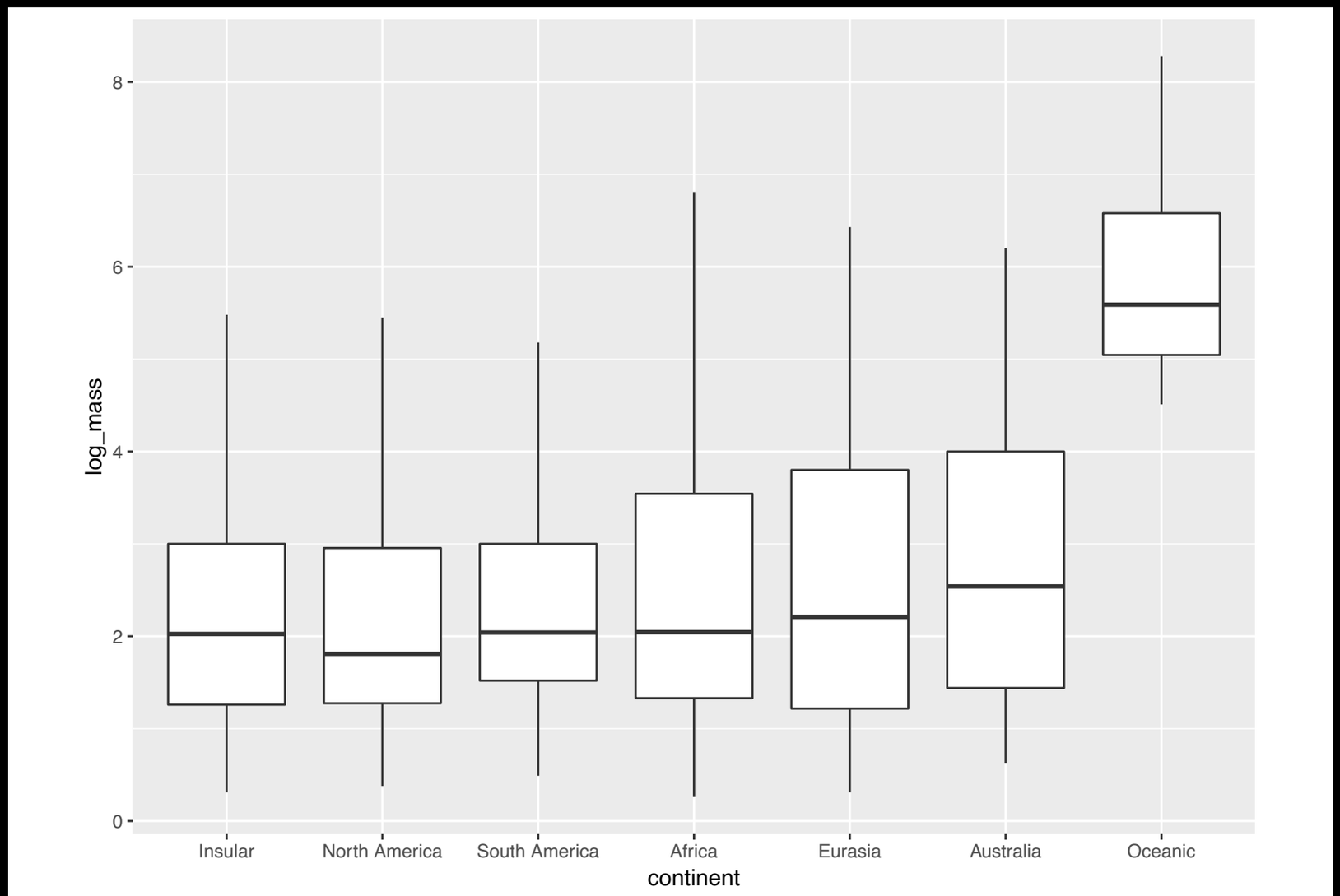


- Geometric object is `geom_boxplot()`
- Categorical variable to mapped to x, and numeric variable mapped to y



BOXPLOTS

- Suppress outliers with `outlier.shape = NA`

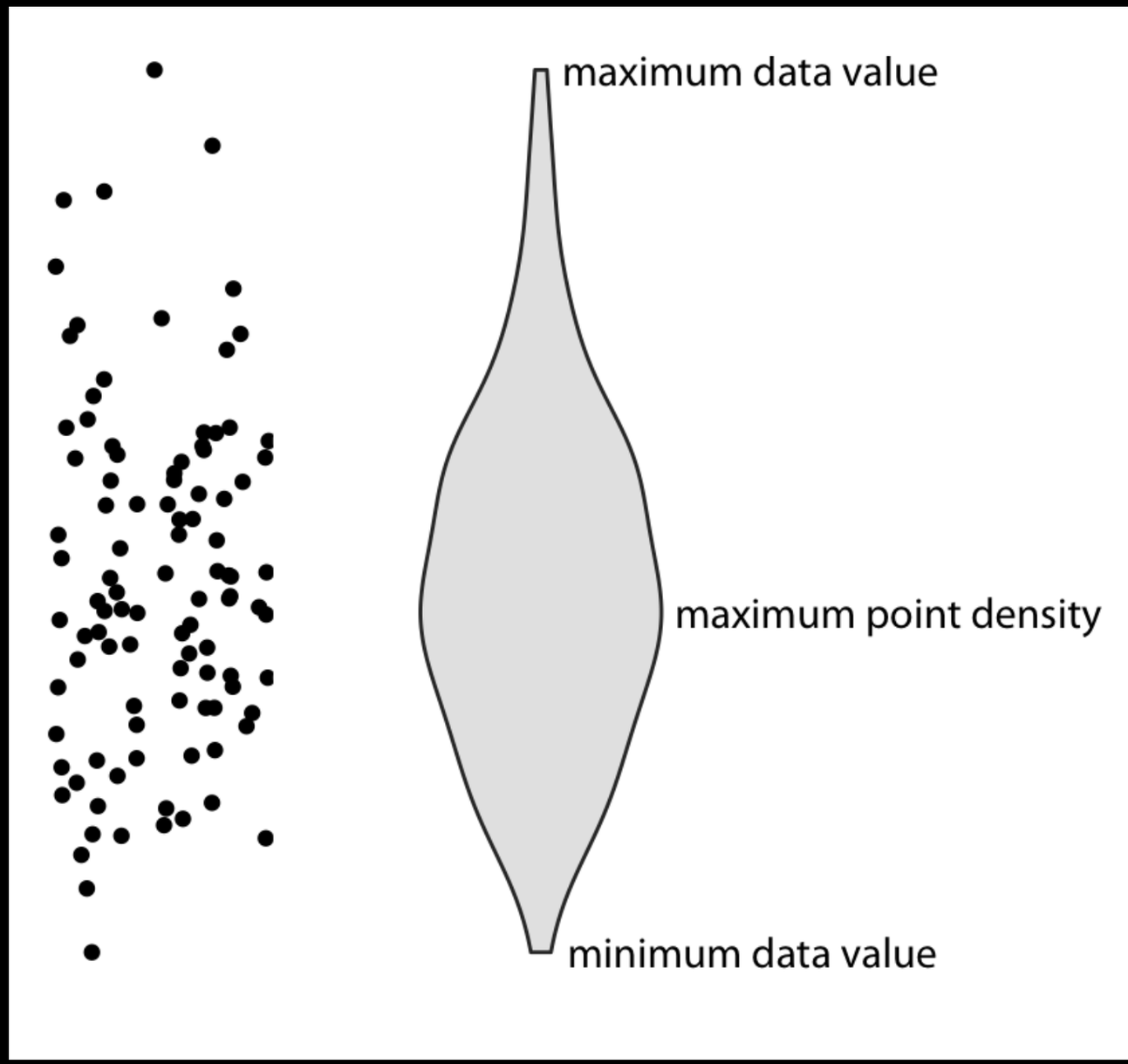


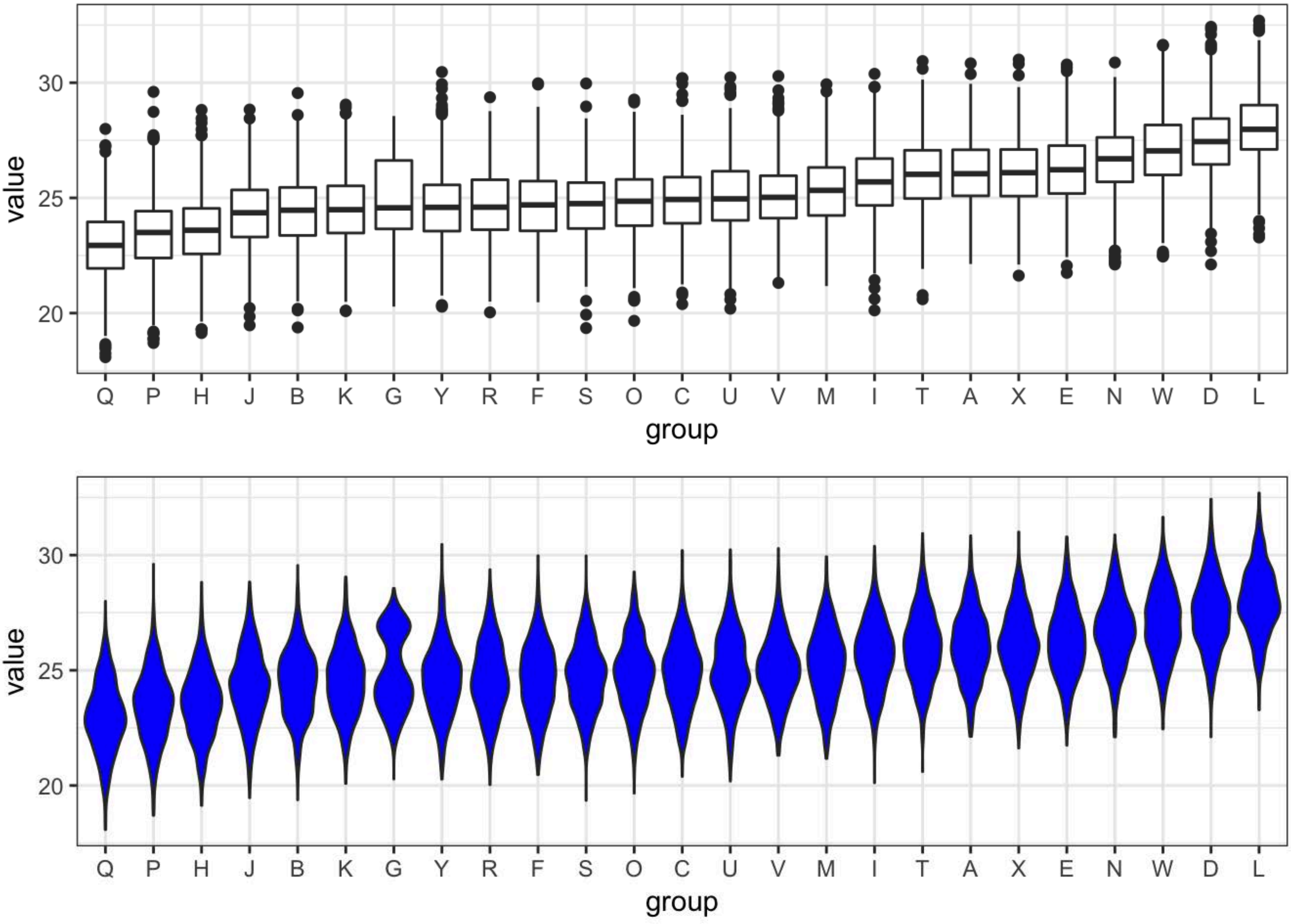

```
ggplot(lqm, aes(x = continent, y = log_mass)) +
  geom_boxplot(outlier.shape = NA)
```

VIOLIN PLOTS

VIOLIN PLOTS

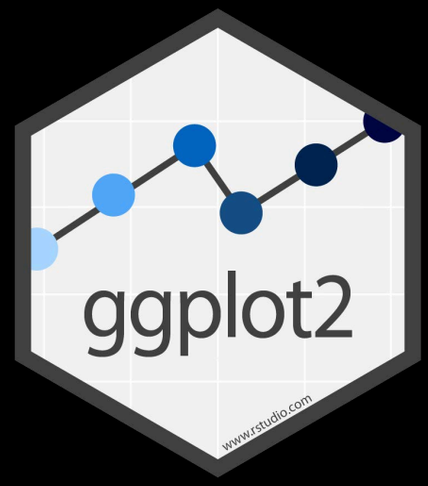
- Each violin is essentially a density plot rotated vertically and mirrored across the vertical
- Less standardized and precise, but also easier to intuit
- Better at showing multi-modal distributions



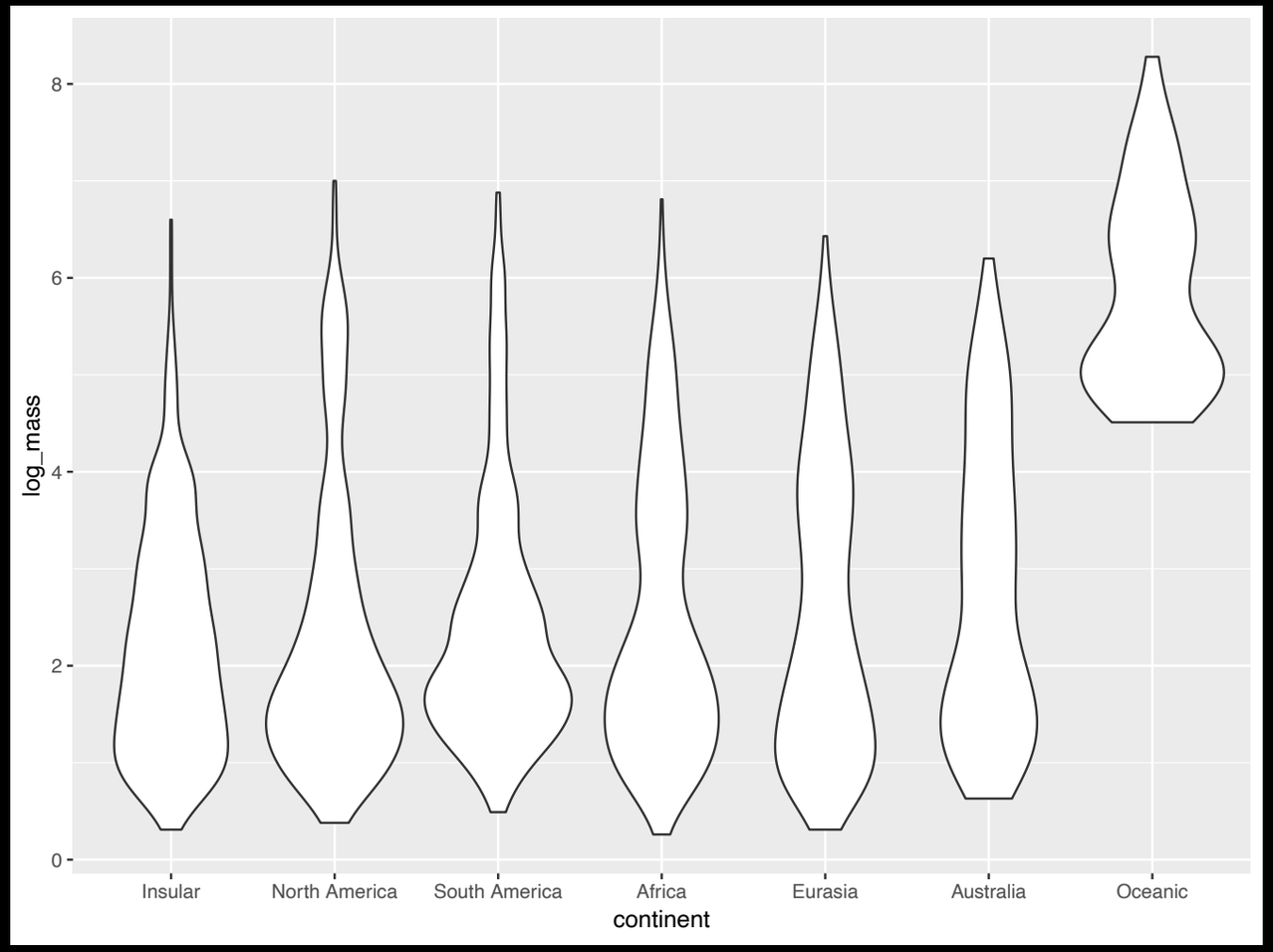

value
30
25
20
Q P H J B K G Y R F S O C U V M I T A X E N W D L
group

value
30
25
20
Q P H J B K G Y R F S O C U V M I T A X E N W D L
group

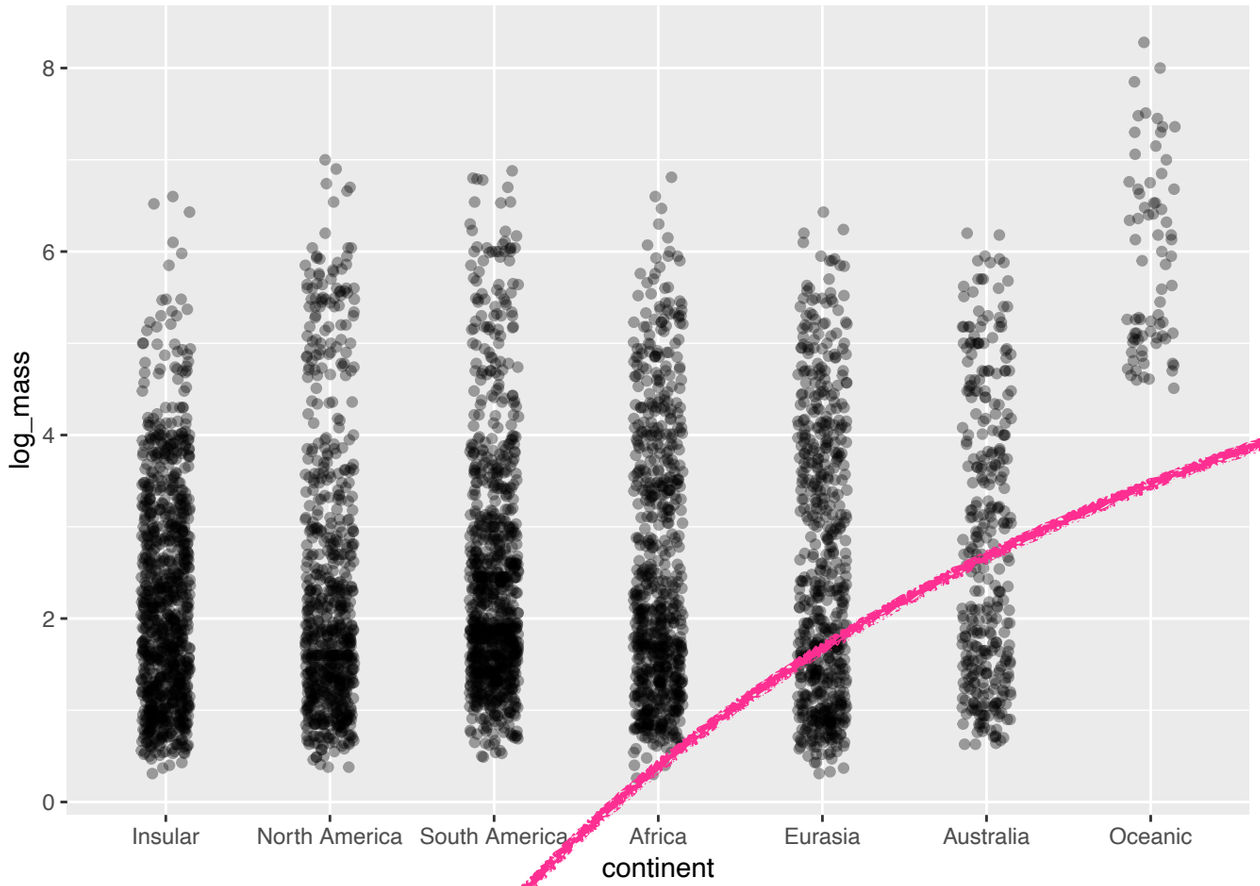
VIOLIN PLOTS



- Geometric object is `geom_violin()`
- Mappings same as boxplot



STRIP PLOTS



`geom_jitter()` makes randomly displaced points

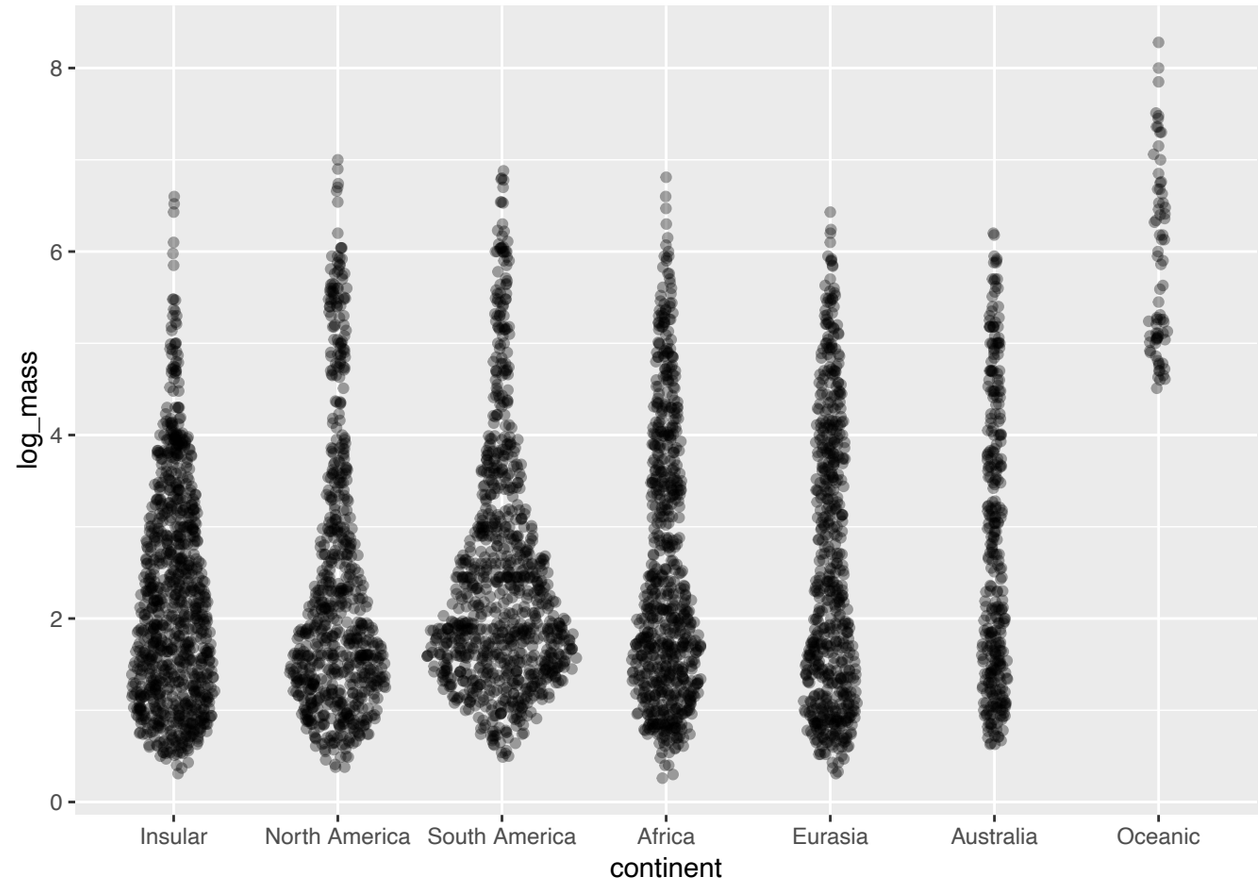
`height` and `width` parameters control amount of jitter

```
ggplot(lqm, aes(x = continent, y = log_mass)) +
  geom_jitter(width = 0.15, height = 0, alpha = 0.35)
```

SINA PLOTS

Not a default geom!

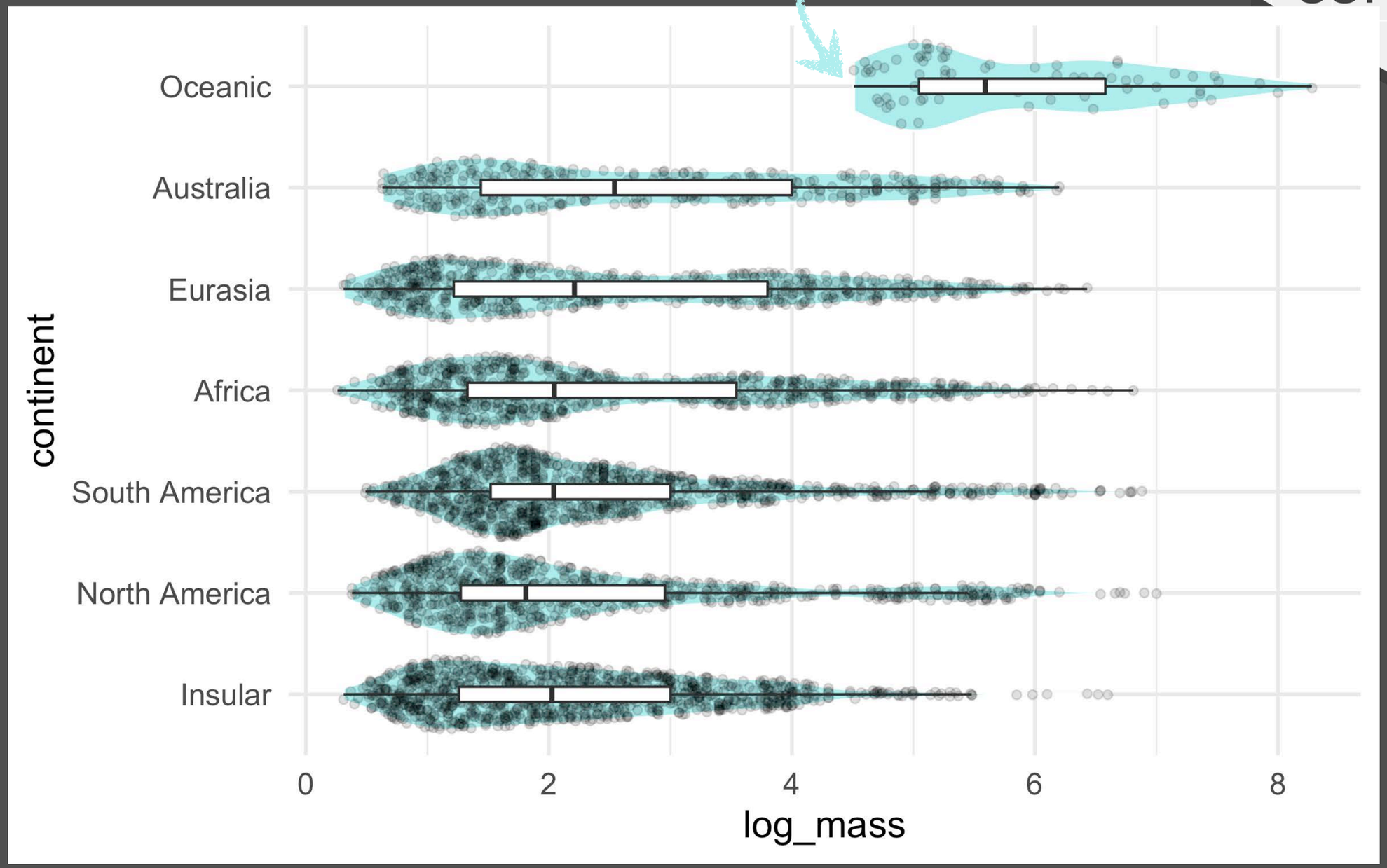
Provided by the "ggforce" package.




```
library("ggforce")

ggplot(lqm, aes(x = continent, y = log_mass)) +
  geom_sina(alpha = 0.15)
```

YOUR TURN

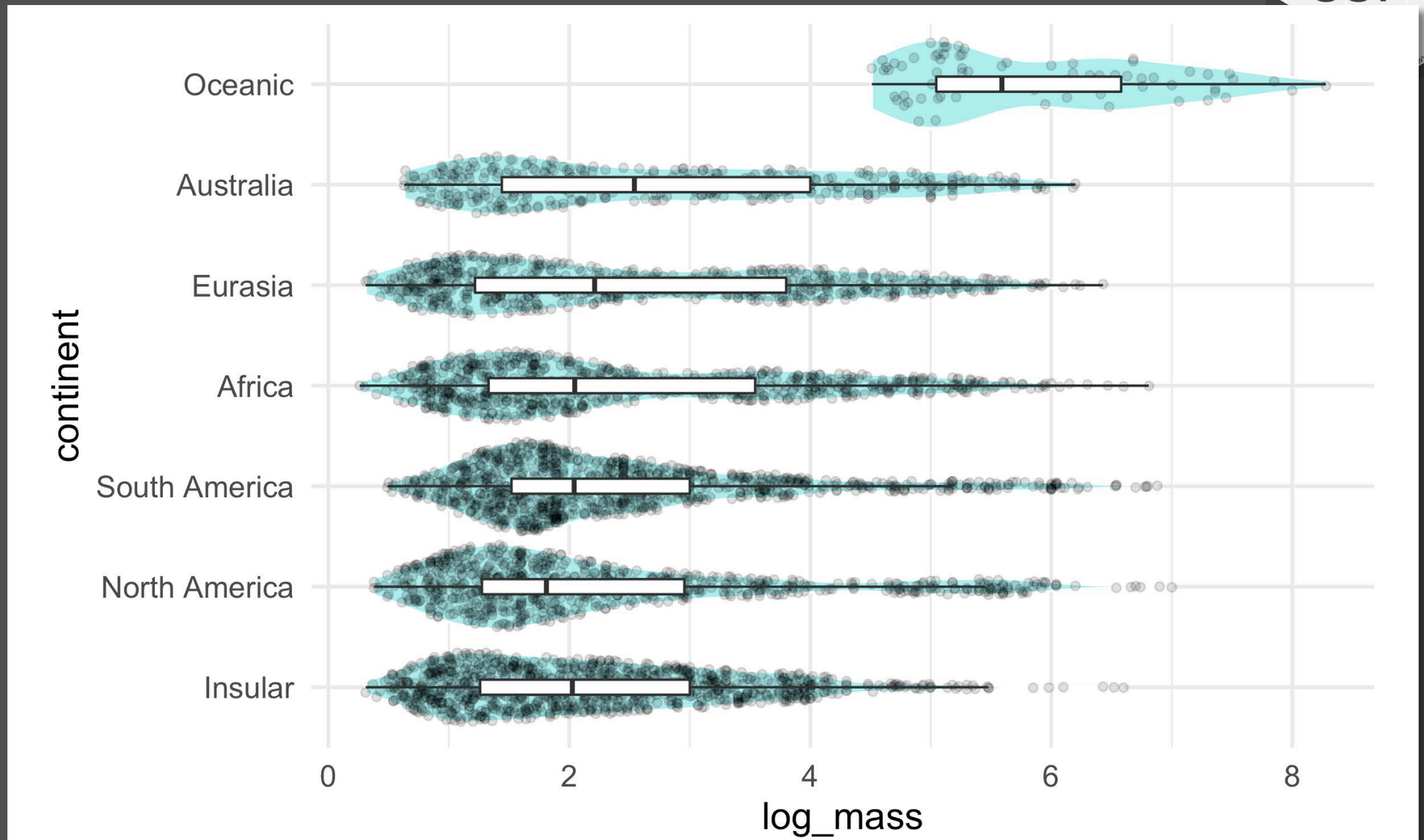
fill is "paleturquoise"



Try to reverse-engineer this (rather busy) plot.

5:00

YOUR TURN

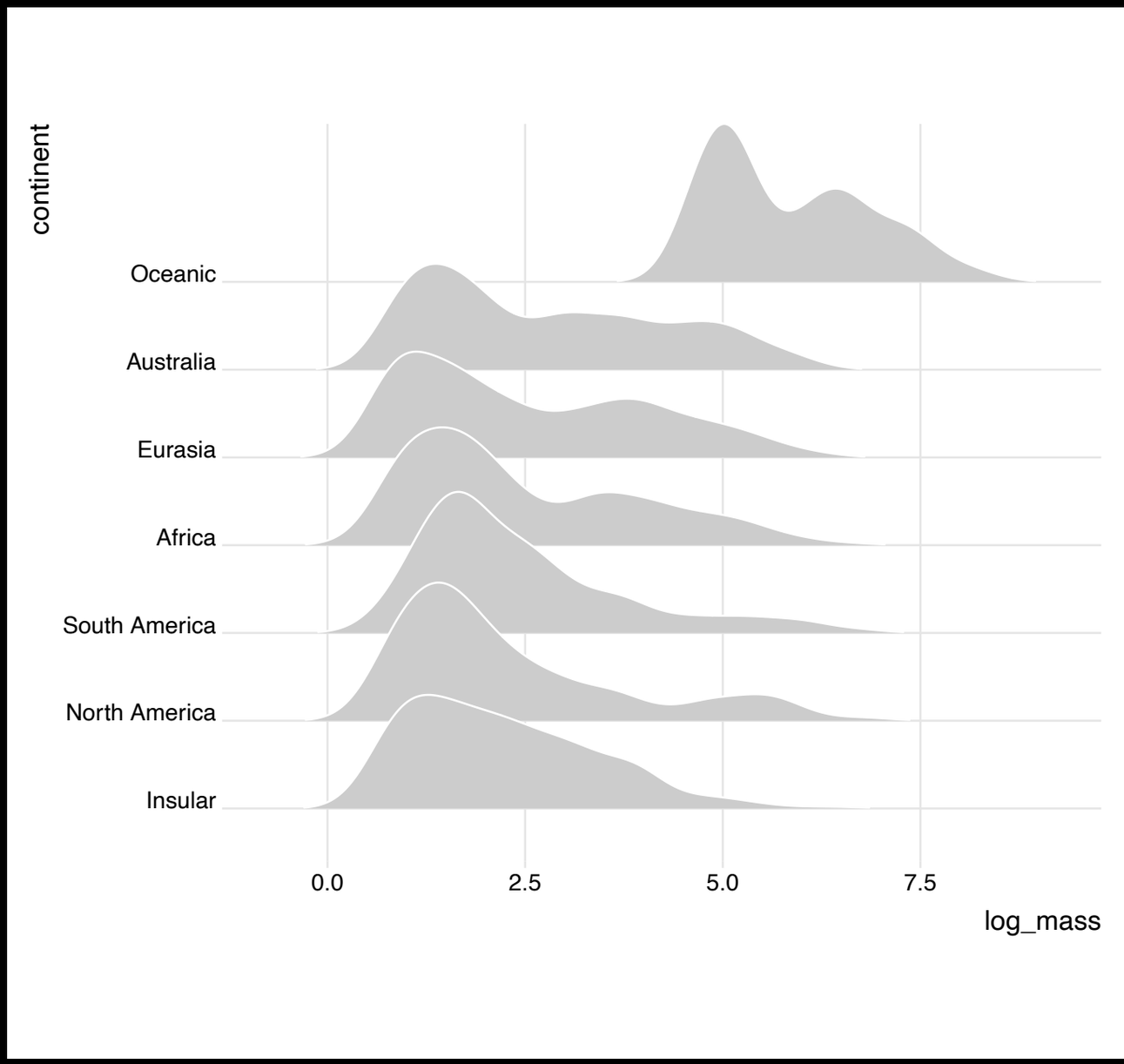


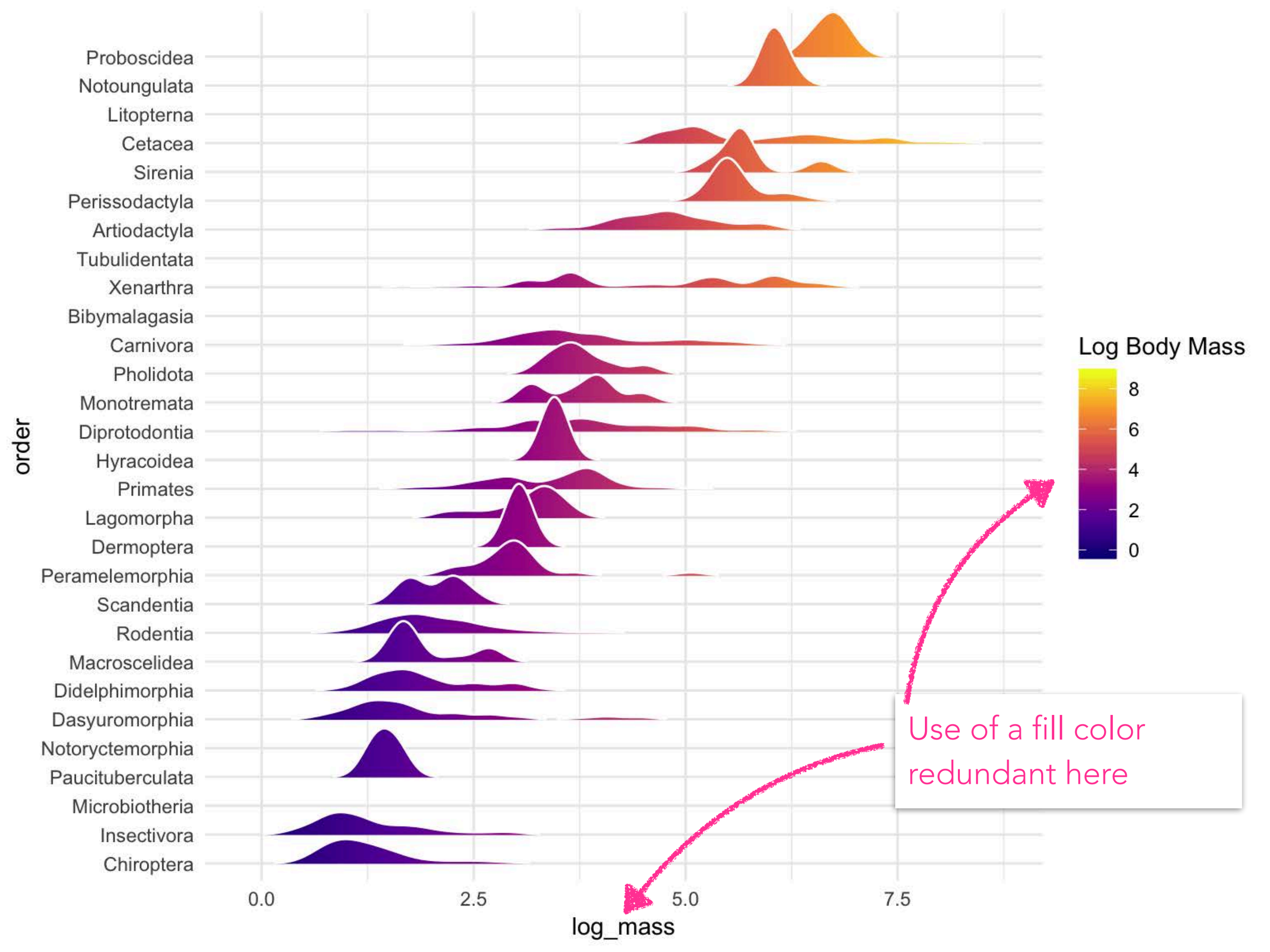

```
ggplot(lqm, aes(x = continent, y = log_mass)) +
  geom_violin(fill = "paleturquoise", color = NA) +
  geom_sina(alpha = 0.15) +
  geom_boxplot(fill = "white", width = 0.15, outlier.shape = NA) +
  coord_flip()
```

RIDGELINE PLOTS

RIDGELINE PLOTS

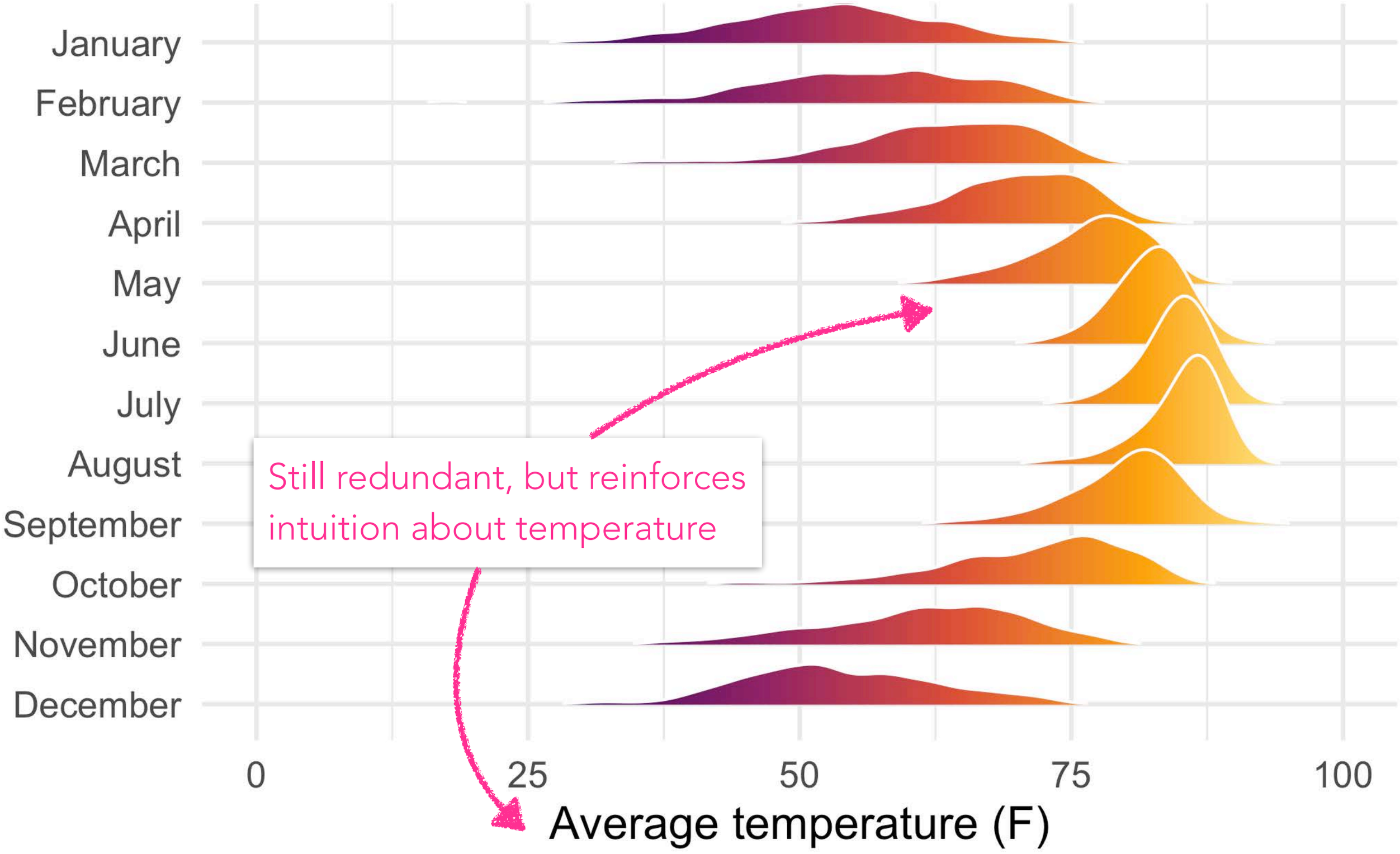
- Each ridge is a vertically offset density plot
- Rarely used, but easy to compare shapes and relative heights
- Not good if specific statistical values are of interest.




order
Proboscidea
Notoungulata
Litopterna
Cetacea
Sirenia
Perissodactyla
Artiodactyla
Tubulidentata
Xenarthra
Bibymalagasia
Carnivora
Pholidota
Monotremata
Diprotodontia
Hyracoidea
Primates
Lagomorpha
Dermoptera
Peramelemorphia
Scandentia
Rodentia
Macroscelidea
Didelphimorphia
Dasyuromorphia
Notoryctemorphia
Paucituberculata
Microbiotheria
Insectivora
Chiroptera
0.0
2.5
5.0
7.5
log_mass
Log Body Mass
8
6
4
2
0
Use of a fill color redundant here

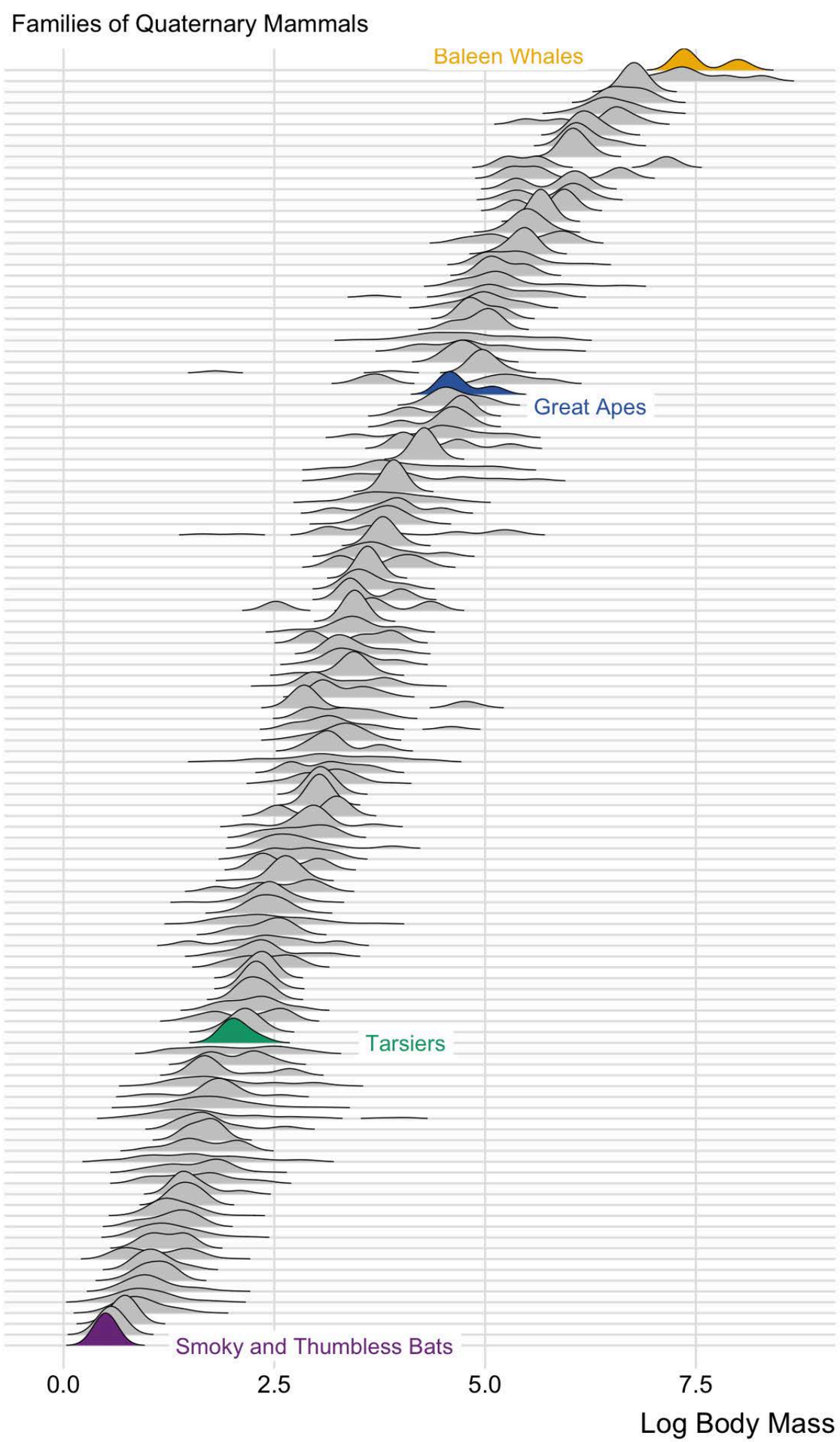
SATX Climate

1946 - 2021



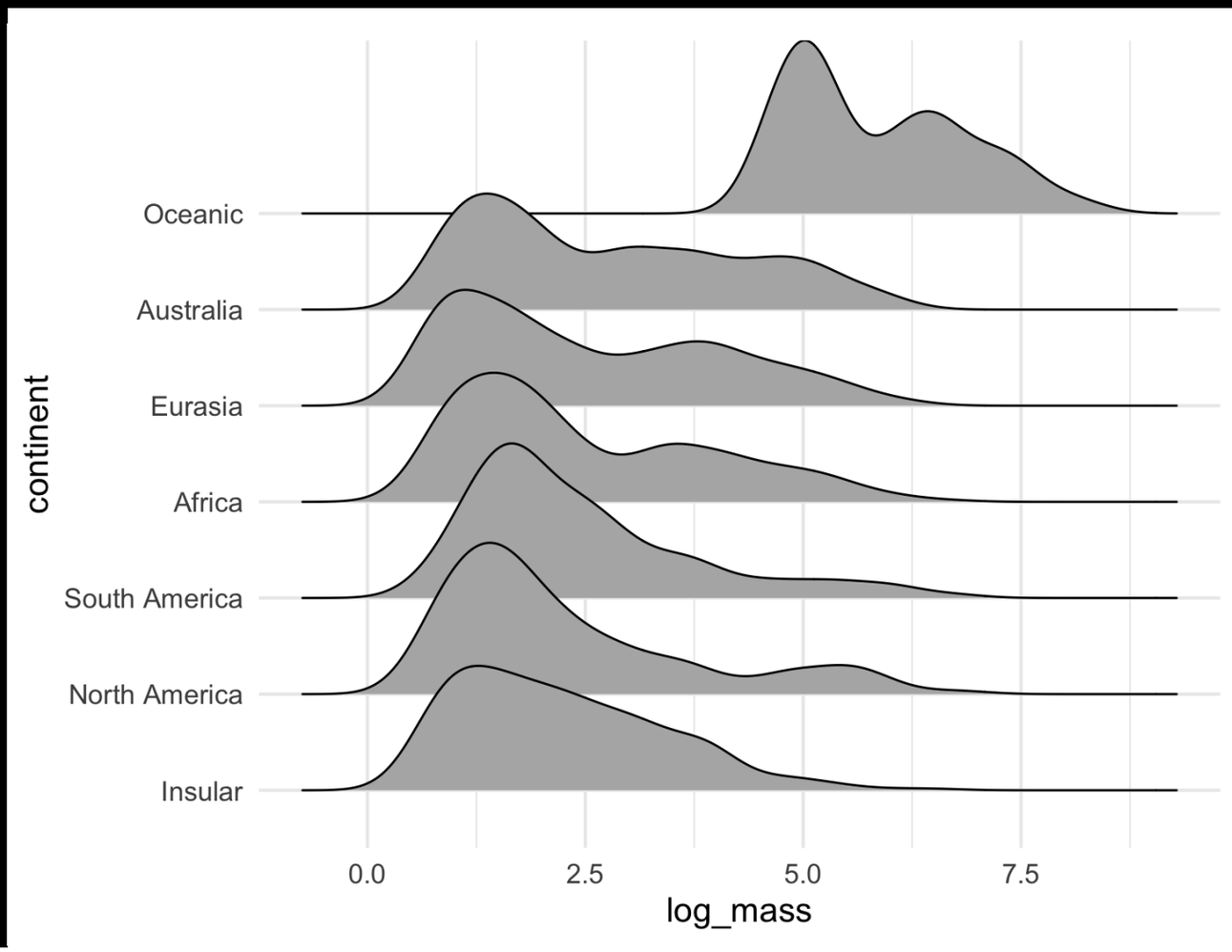
RIDGELINE PLOT

- Only method we have discussed that remains readable with large number of distributions



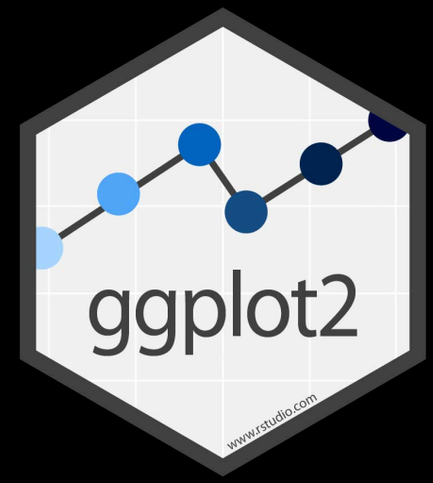
RIDGELINE PLOTS

- Not a default geom—provided by “ggridges” package
- Basic geometric object is `geom_density_ridges()`
- Like density plot, numeric variable mapped to x but categorical variable to mapped to y

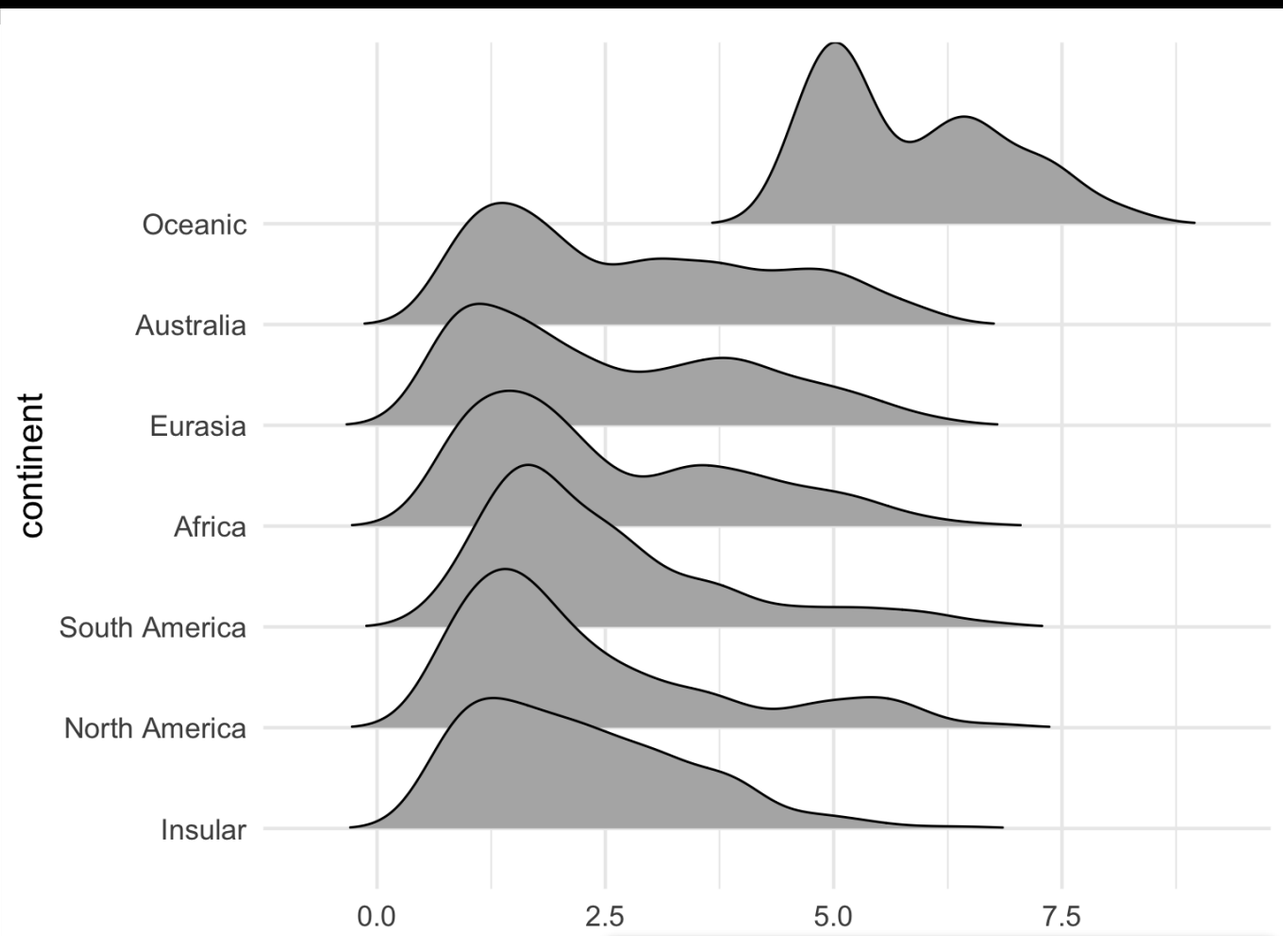



```
library("ggridges")

ggplot(lqm, aes(x = log_mass, y = continent)) +
  geom_density_ridges()
```

RIDGELINE PLOTS



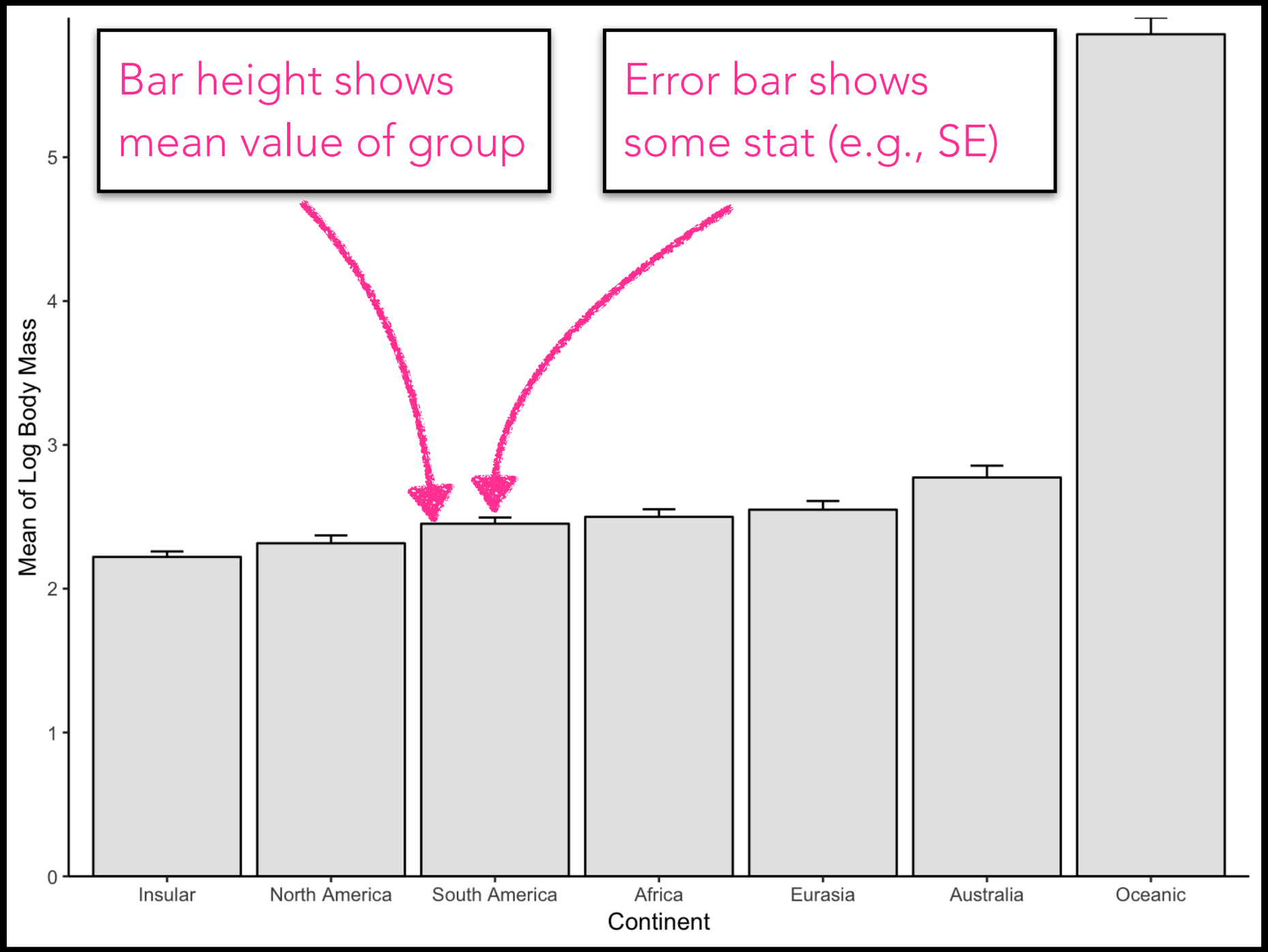
- Not a default geom—provided by "ggridges" package
- Basic geometric object is `geom_density_ridges()`
- Like density plot, numeric variable mapped to x but categorical variable to mapped to y



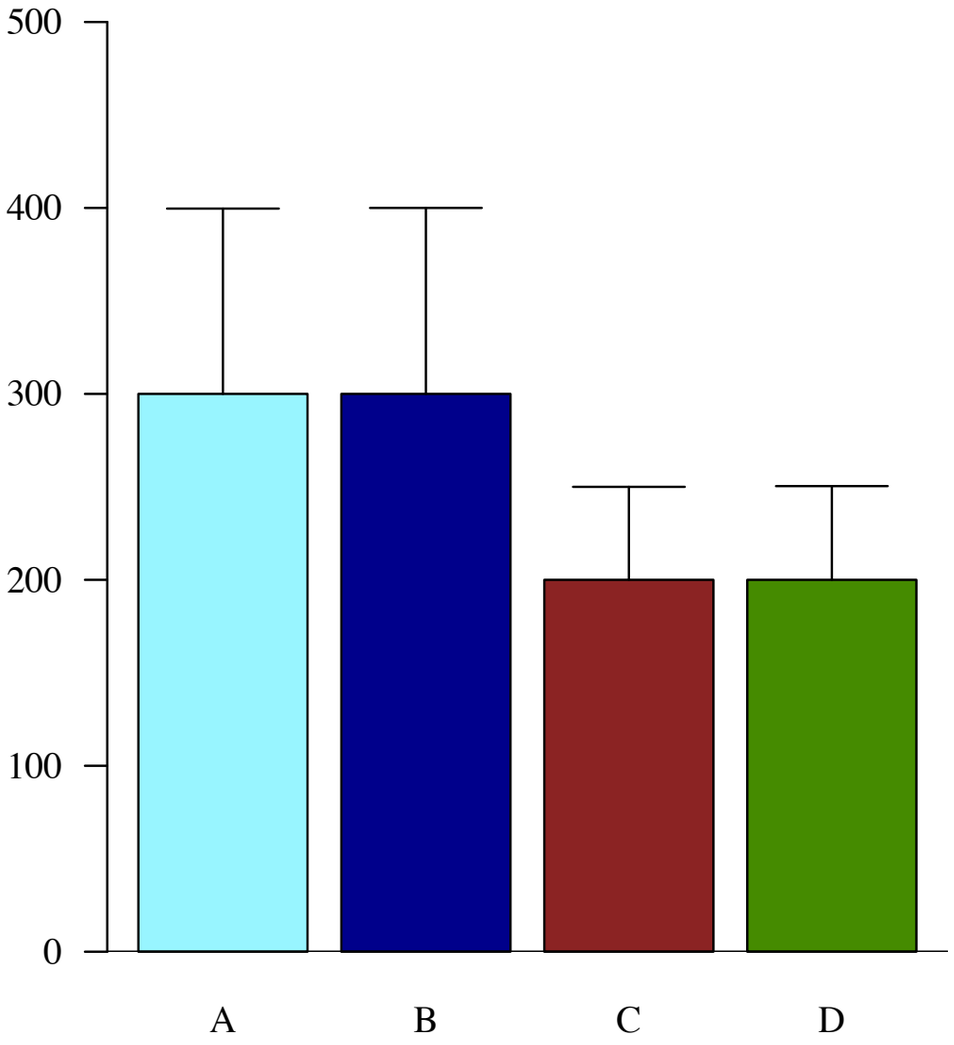

```
library("ggridges")

ggplot(lqm, aes(x = log_mass, y = continent)) +
  geom_density_ridges(rel_min_height = 0.005)
```

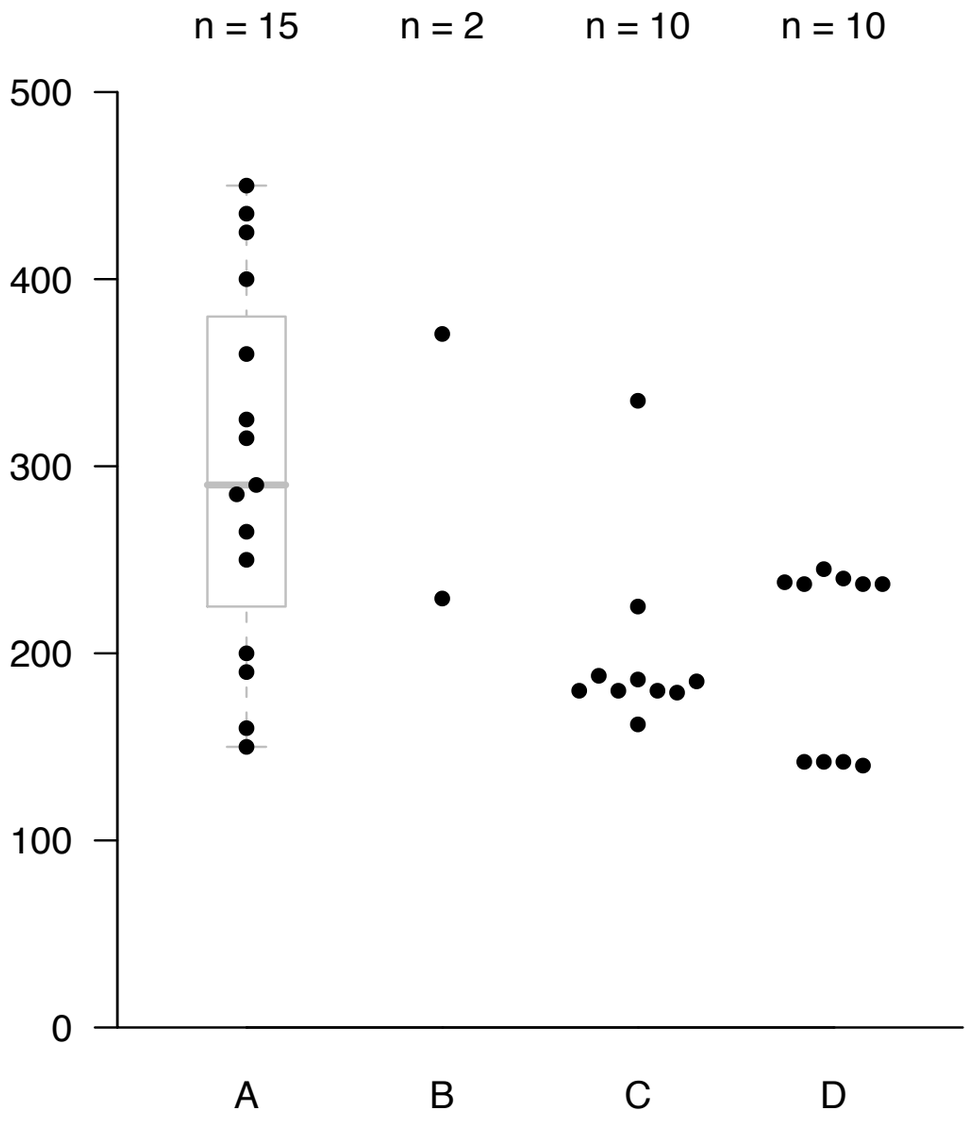
Cleaner look: cuts off long tails.

WHAT ABOUT A "DYNAMITE PLOT"?



DON'T USE THEM FOR DISTRIBUTIONS





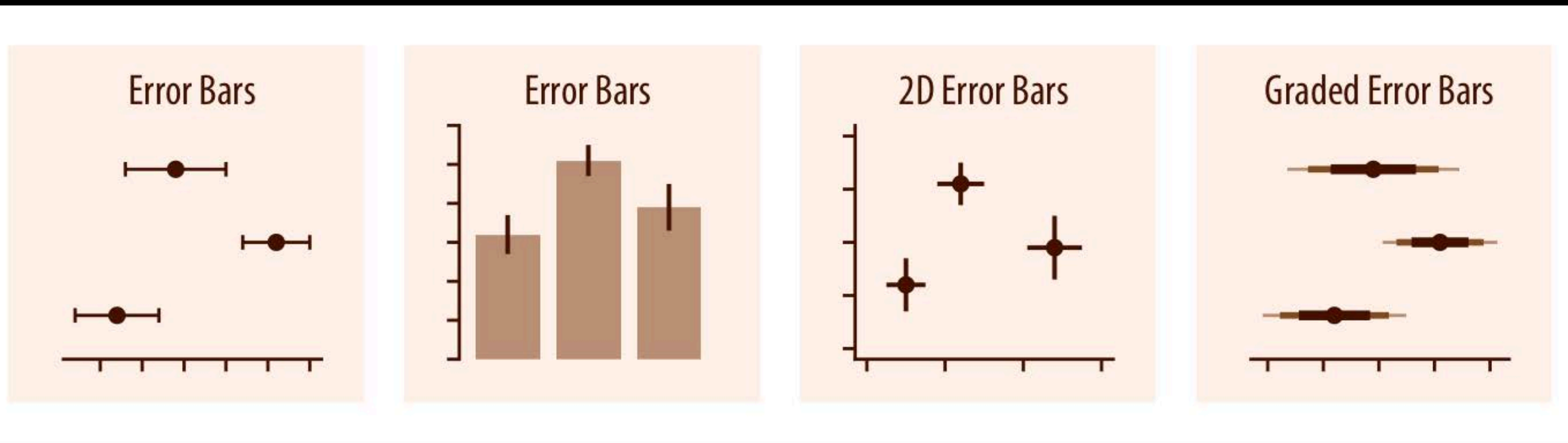
http://biostat.mc.vanderbilt.edu/wiki/pub/Main/TatsukiRcode/Poster3.pdf

PROBLEMS WITH DYNAMITE PLOTS

- Hides the data for no particular reason
- Does not show possible skew or multi-modality
- Anchors data at zero, *which might not make sense for the distribution*
  - Data in small range: : part of the range covered by the bar might have never been observed in the sample
  - Log scale (log(0) is undefined)
- *Bar plots are good for showing counts and single amounts of categories, but not for summarizing continuous data*

VISUALIZING UNCERTAINTY

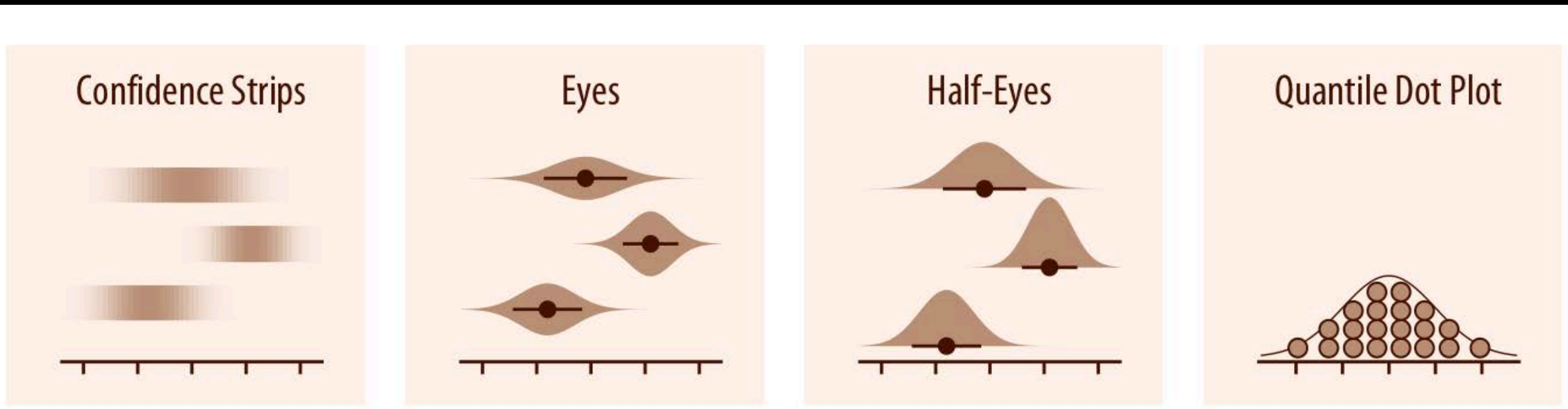
Showing uncertainty intervals around a measurement/estimate



Very common approaches

VISUALIZING UNCERTAINTY

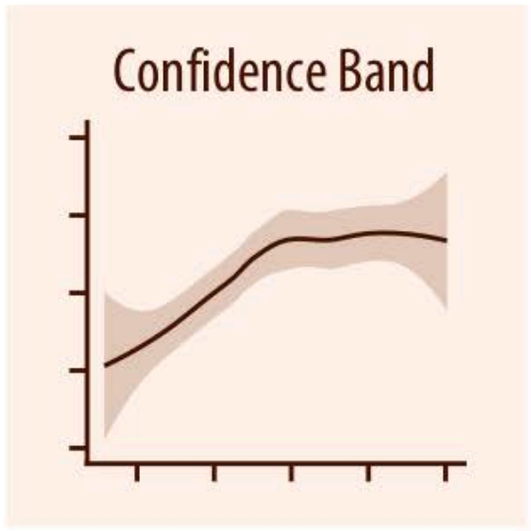
Showing continuous uncertainty around an estimate

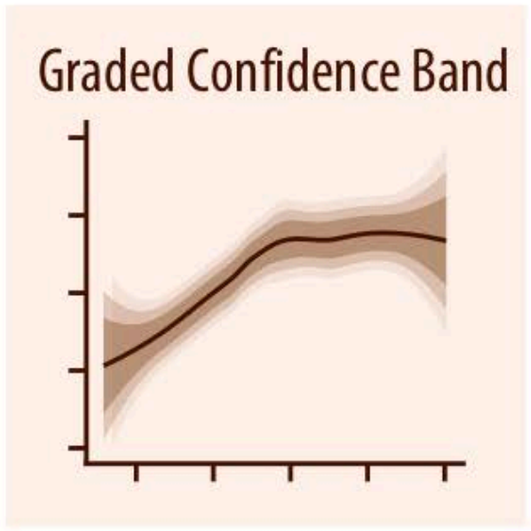


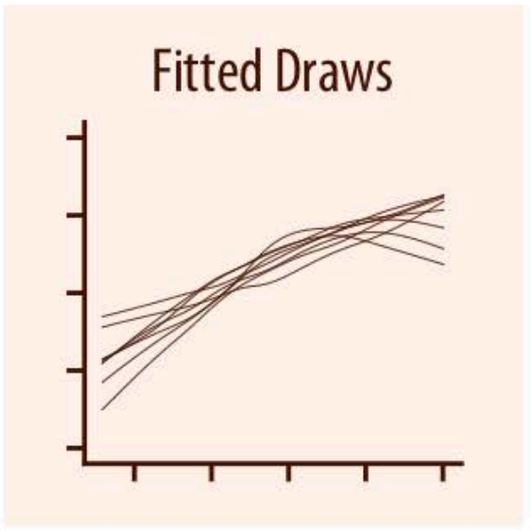
Less-common approaches

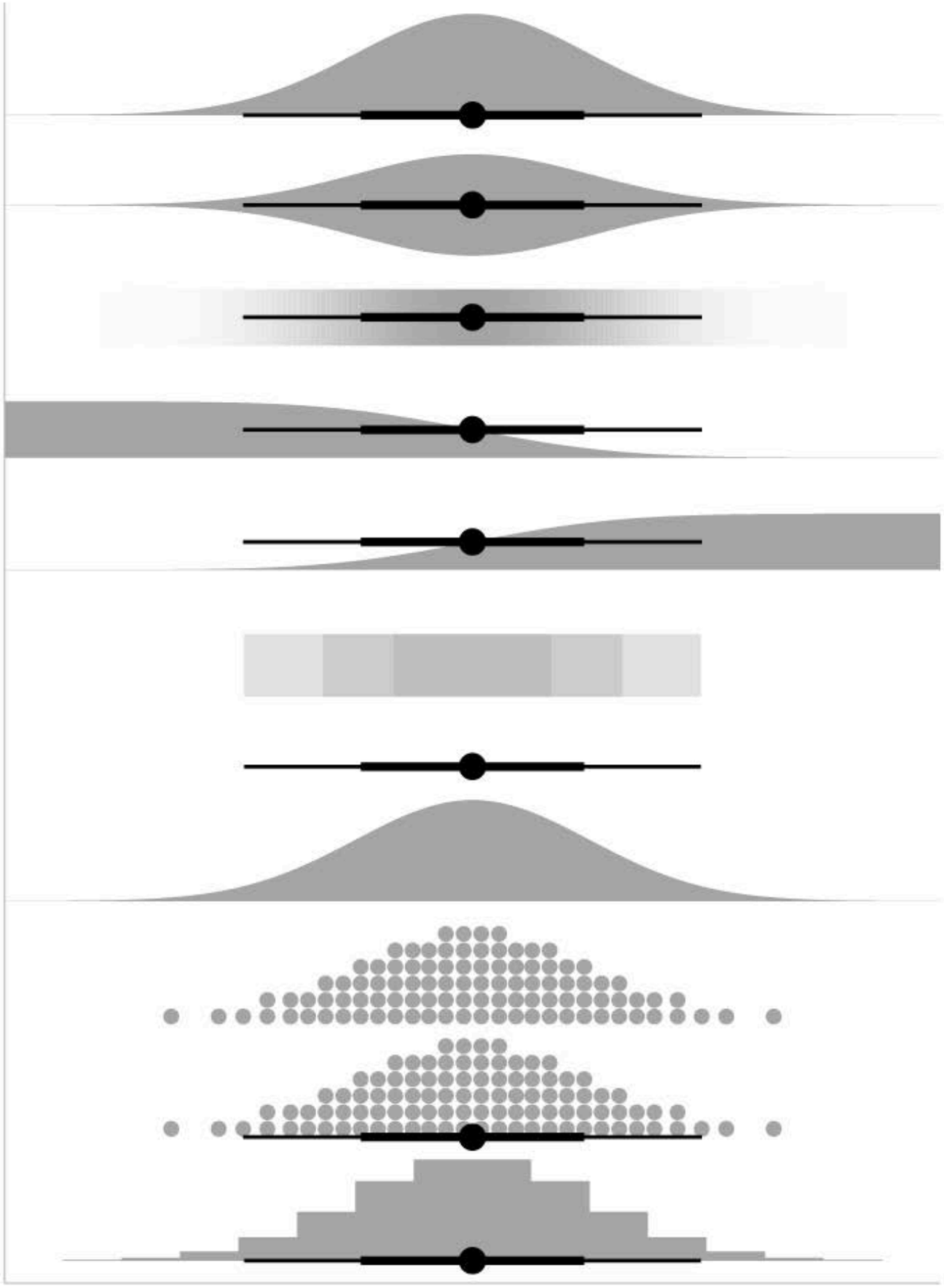
VISUALIZING UNCERTAINTY

Showing uncertainty around smooth lines

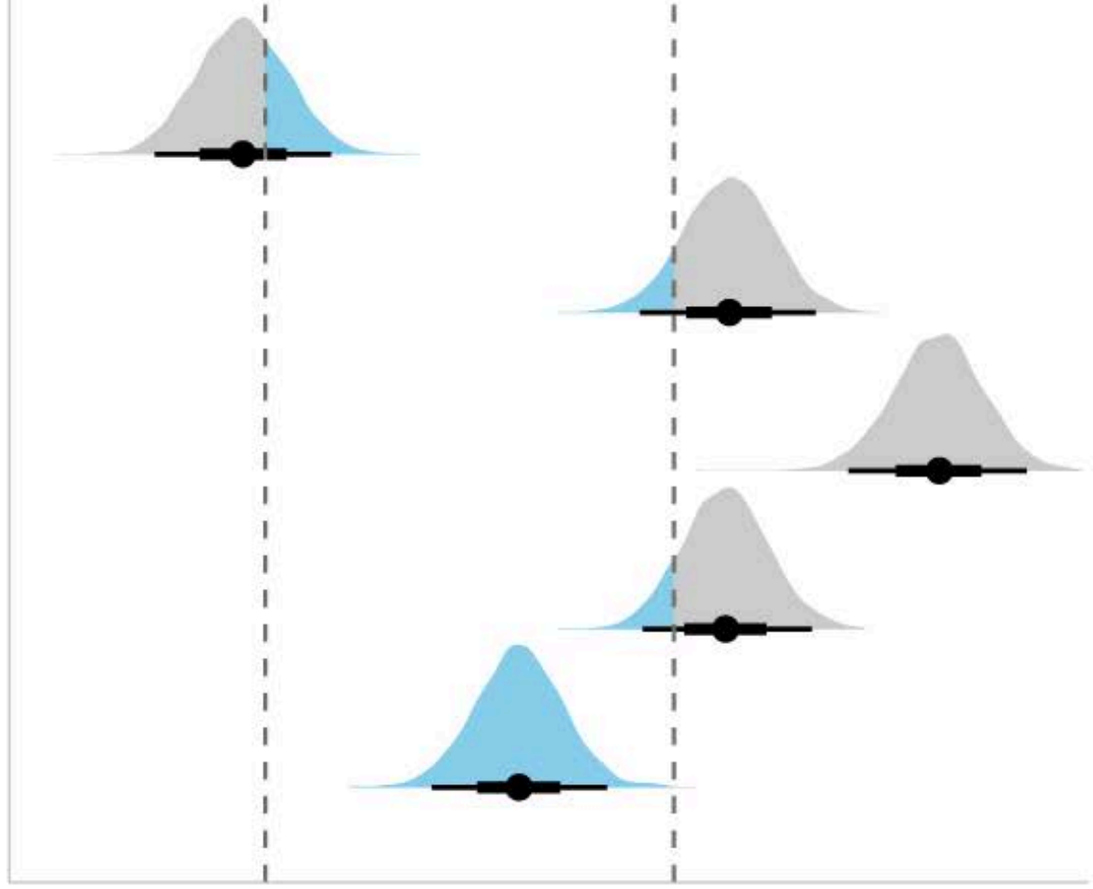






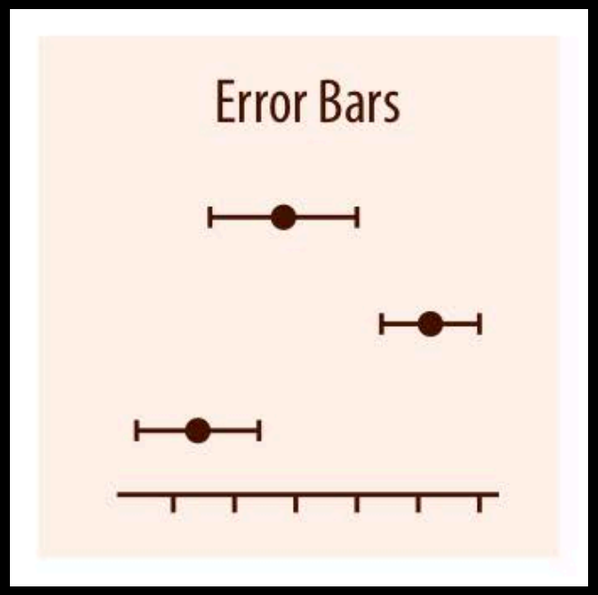



GGDIST



VISUALIZING UNCERTAINTY

Showing uncertainty intervals around a measurement/estimate



The workhorse: error bars

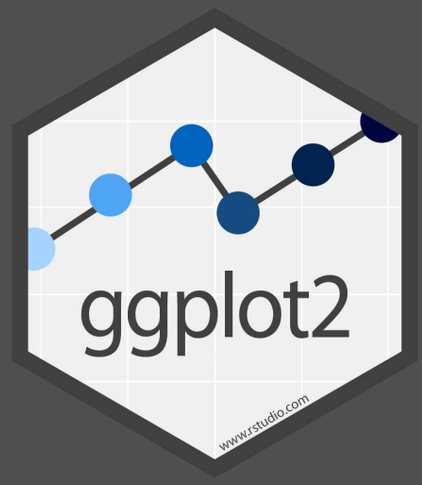
YOUR TURN

- Download the model summary data set from the course website ("model_summary.csv"). This is a file format for single R objects.

- Read the file into R using:

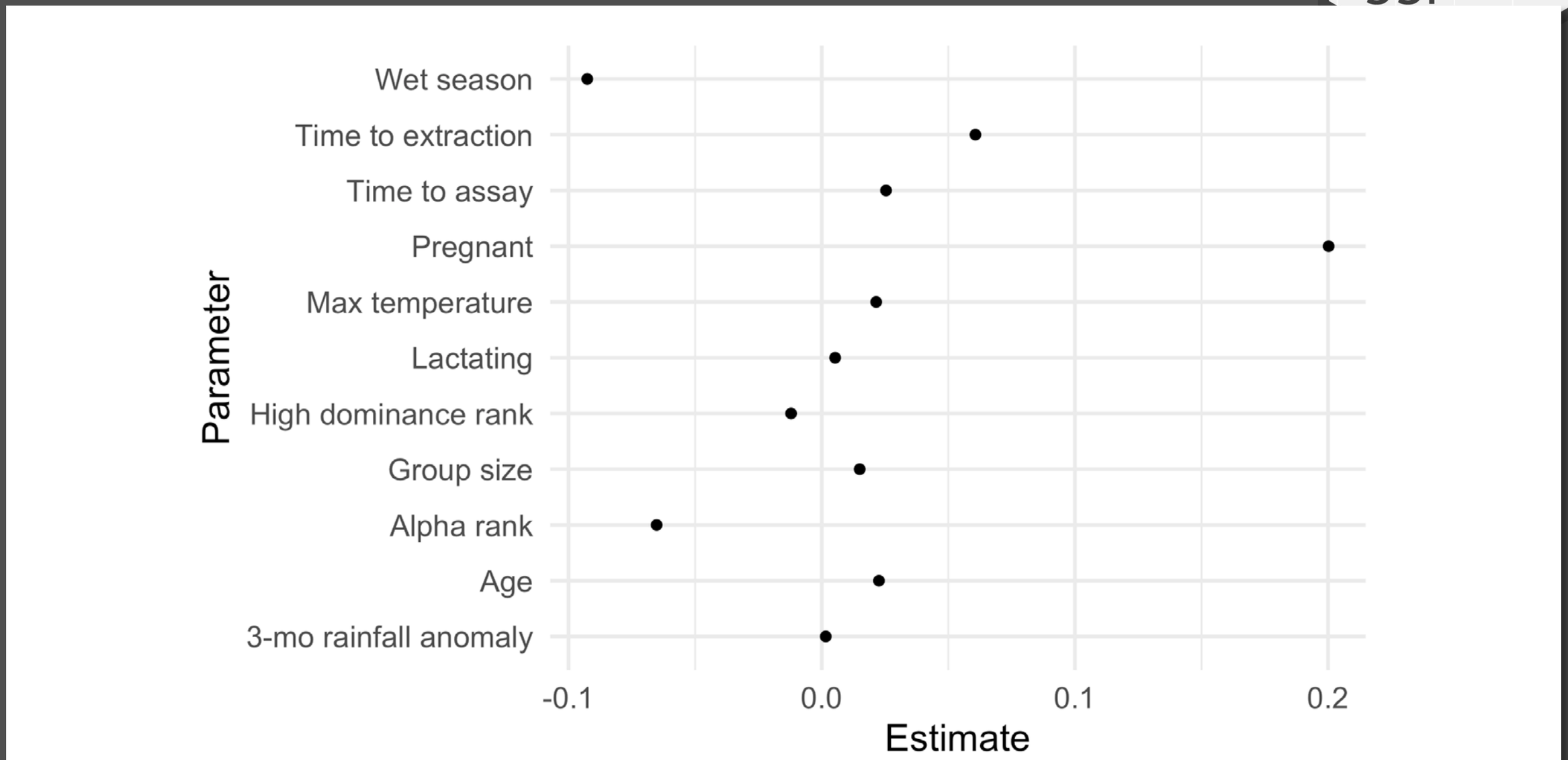
```
model_summary <- read_csv("your-path/model_summary.csv")
```

YOUR TURN



parameter	mean	std_error	lower	upper
Lactating	0.0053	0.0089	-0.0094	0.0198
Pregnant	0.2002	0.0097	0.1842	0.2163
Wet season	-0.0926	0.0084	-0.1063	-0.0786
Group size	0.0150	0.0071	0.0033	0.0267
Alpha rank	-0.0652	0.0316	-0.1171	-0.0133
High dominance rank	-0.0121	0.0104	-0.0293	0.0050
Max temperature	0.0215	0.0043	0.0144	0.0286
3-mo rainfall anomaly	0.0016	0.0037	-0.0044	0.0077
Time to extraction	0.0607	0.0044	0.0534	0.0680
Time to assay	0.0254	0.0040	0.0189	0.0321

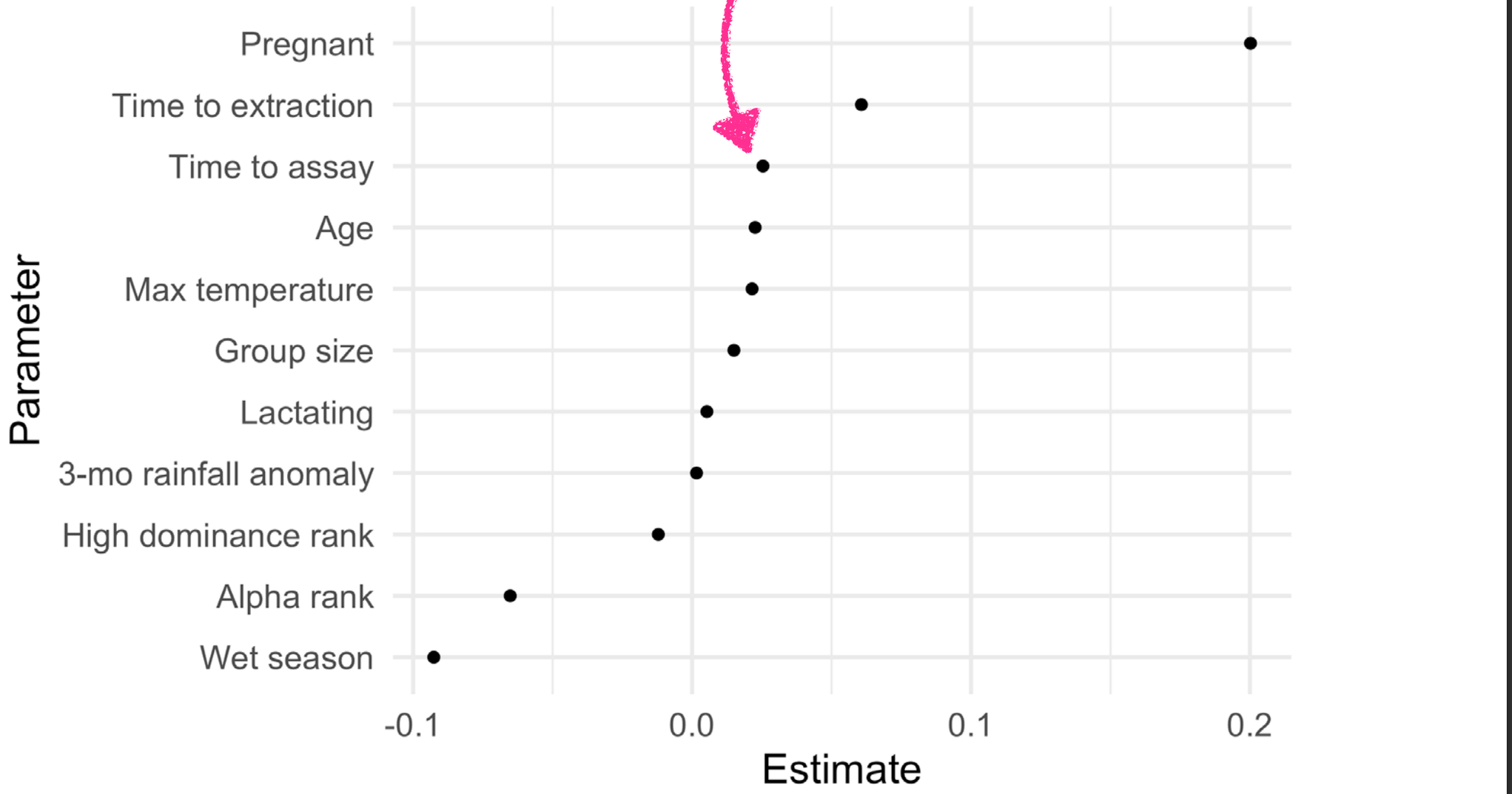
YOUR TURN




```
ggplot(model_summary, aes(x = mean, y = parameter)) +
  geom_point(size = 2) +
  labs(x = "Estimate", y = "Parameter")
```

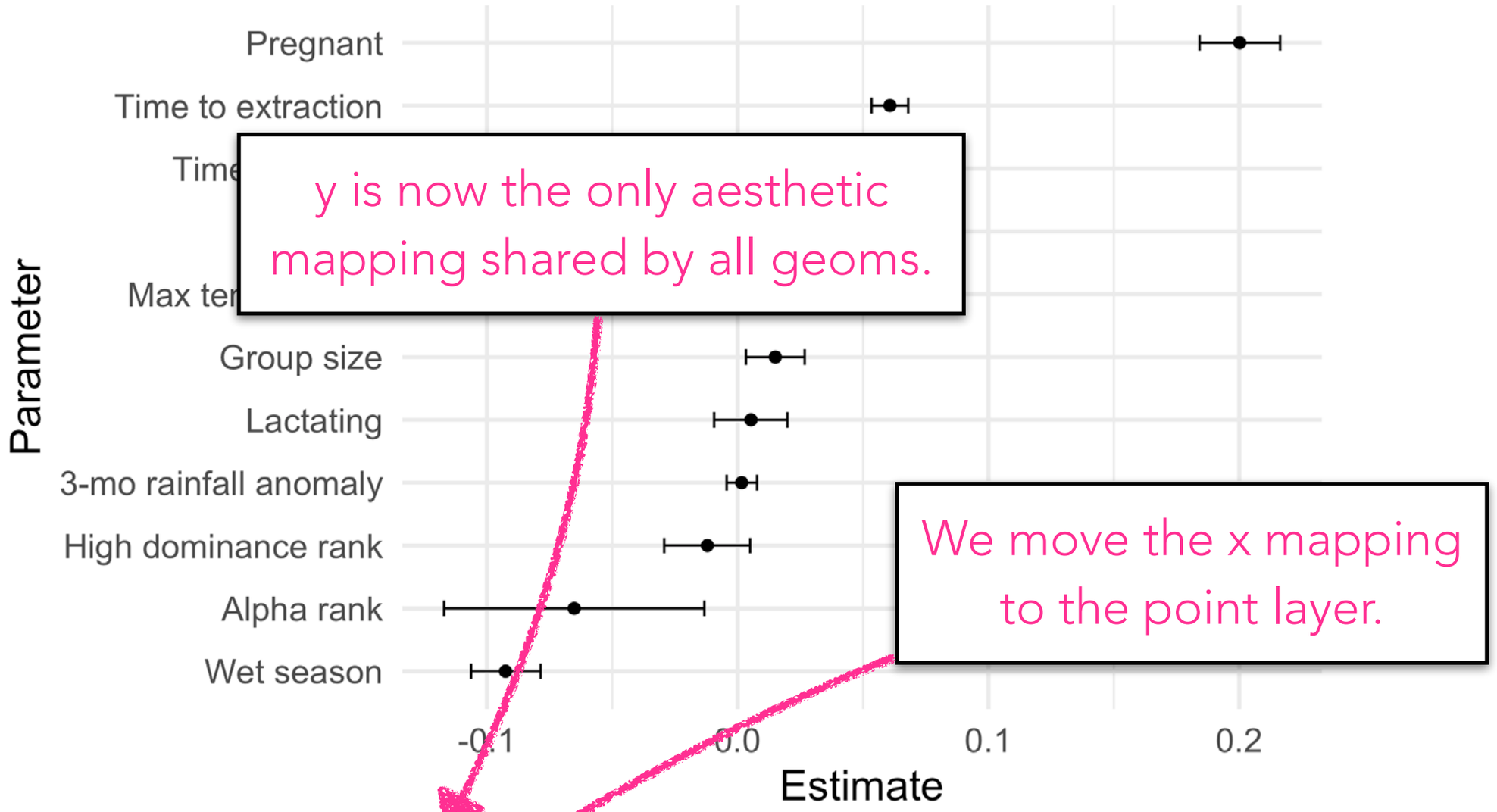
YOUR TURN

Reordering by mean looks cleaner and improves interpretation.



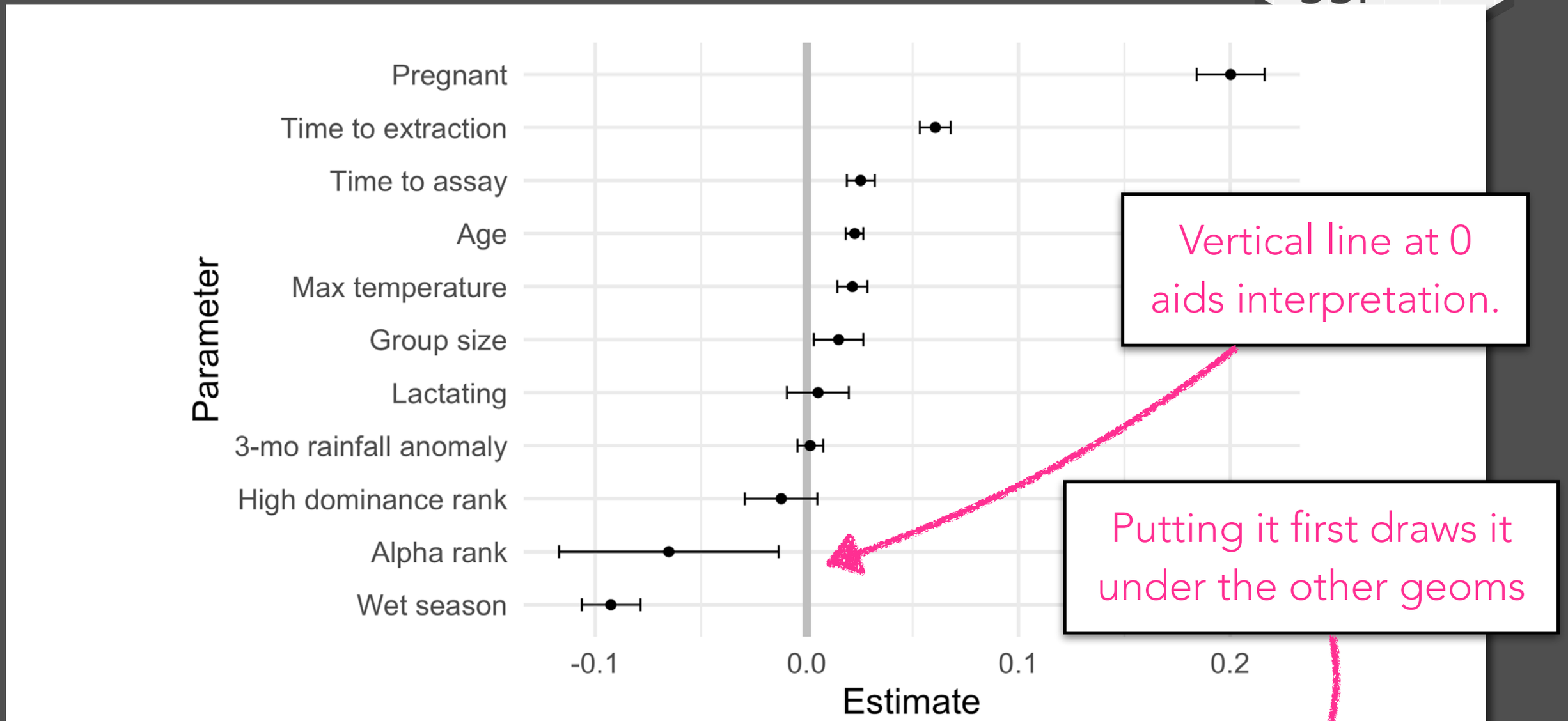

```
ggplot(model_summary, aes(x = mean, y = fct_reorder(parameter, mean))) +
  geom_point(size = 2) +
  labs(x = "Estimate", y = "Parameter")
```

YOUR TURN




```
ggplot(model_summary, aes(y = fct_reorder(parameter, mean))) +
  geom_errorbarh(aes(xmin = lower, xmax = upper), height = 0.25) +
  geom_point(aes(x = mean), size = 2) +
  labs(x = "Estimate", y = "Parameter")
```

YOUR TURN



Vertical line at 0 aids interpretation.

Putting it first draws it under the other geoms

```
ggplot(model_summary, aes(y = fct_reorder(parameter, mean))) +
  geom_vline(xintercept = 0, color = "gray", size = 2) +
  geom_errorbarh(aes(xmin = lower, xmax = upper), height = 0.25) +
  geom_point(aes(x = mean), size = 2) +
  labs(x = "Estimate", y = "Parameter")
```