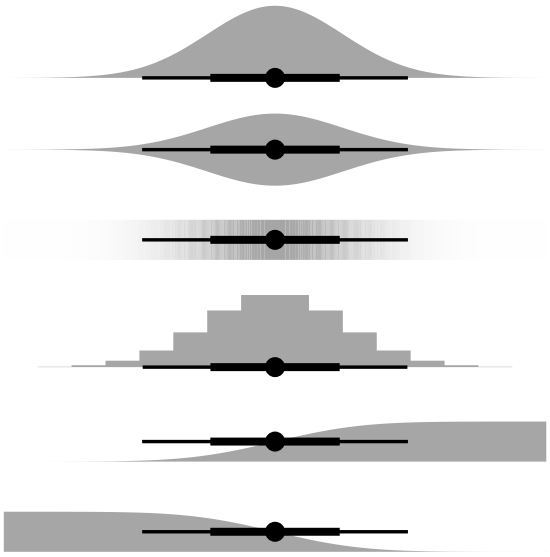
Shortcut slab+interval geometries

The stat_sample_slabinterval() and stat_dist_slabinterval() stats are flexible meta-geometries for visualizing sample data or analytical distributions.
With that flexibility comes a cost in remembering particular combinations of parameters that yield specific visualization types.
Thus, ggdist also provides several shortcut stats with sensible default parameters:

This geometry ...	... uses these defaults: mapping = *aesthetic mapping*	slab_type = *function assigned to the computed aesthetic* **f**	side = *side to draw the slab on*	justification = *position of interval relative to slab*	normalize = *What groups to normalize max height of slab thickness within*
stat_sample_slabinterval() stat_dist_slabinterval()	aes(thickness = **f**)	"pdf"	"topright"	0	"all"
stat_halfeye() stat_dist_halfeye()	aes(thickness = f)	"pdf"	"topright"	0	"all"
stat_eye() stat_dist_eye()	aes(thickness = f)	"pdf"	"both"	0	"all"
stat_gradientinterval() stat_dist_gradientinterval()	aes(slab_alpha = **f**)	"pdf"	"topright"	0.5	"all"
stat_histinterval()	aes(thickness = f)	"histogram"	"topright"	0	"all"
stat_cdfinterval() stat_dist_cdfinterval()	aes(thickness = f)	"cdf"	"topleft"	0.5	"none"
stat_ccdfinterval() stat_dist_ccdfinterval()	aes(thickness = f)	"ccdf"	"topleft"	0.5	"none"

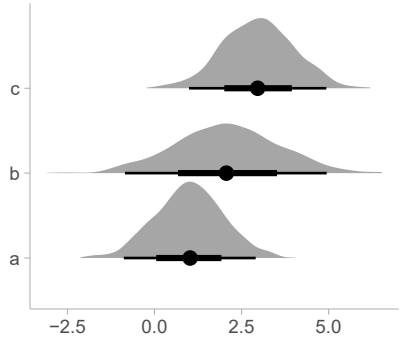


Example from stat_sample_slabinterval() sub-family

Using sample data

```
df = data.frame(
  group = c("a", "b", "c"),
  value = rnorm(
    3000,
    mean = c(1, 2, 3),
    sd = c(1, 1.5, 1)
  )
)

ggplot(df) +
  aes(y = group, x = value) +
  stat_halfeye()
```



Example from stat_dist_slabinterval() sub-family

Using analytical distributions

```
df = data.frame(
  group = c("a", "b", "c"),
  mean = c(1, 2, 3),
  sd = c(1, 1.5, 1)
)

ggplot(df) +
  aes(
    y = group,
    dist = dist_normal(mean, sd)
  ) +
  stat_dist_halfeye()
```