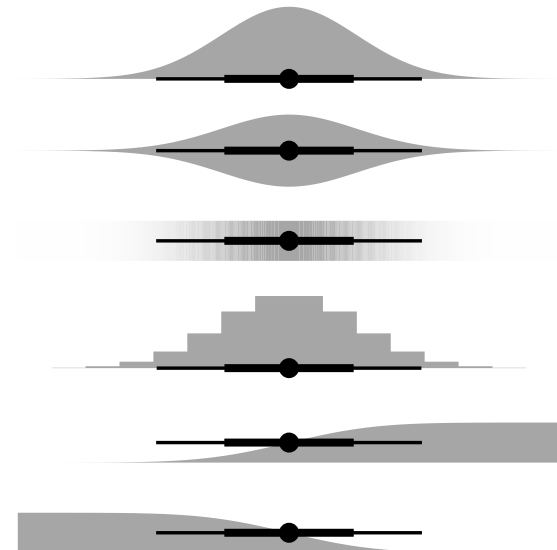


Shortcut slab+interval geometries

The `stat_slabinterval()` stat is a flexible meta-geometry for visualizing **sample data** and **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:



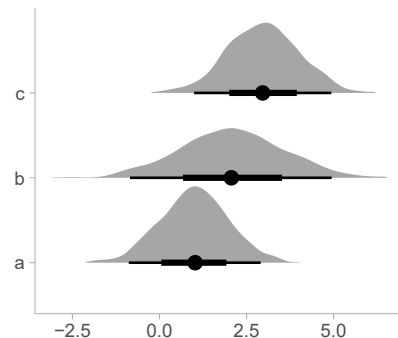
This geometry uses these defaults:				
	mapping = <i>aesthetic mapping</i>	slab_type = <i>function assigned to the computed aesthetic f</i>	side = <i>side to draw the slab on</i>	justification = <i>position of interval relative to slab</i>	normalize = <i>What groups to normalize max height of slab thickness within</i>
<code>stat_slabinterval()</code>	<code>aes(thickness = f)</code>	"pdf"	"topright"	0	"all"
<code>stat_halfeye()</code>	<code>aes(thickness = f)</code>	"pdf"	"topright"	0	"all"
<code>stat_eye()</code>	<code>aes(thickness = f)</code>	"pdf"	"both"	0	"all"
<code>stat_gradientinterval()</code>	<code>aes(slab_alpha = f)</code>	"pdf"	"topright"	0.5	"all"
<code>stat_histinterval()</code>	<code>aes(thickness = f)</code>	"histogram"	"topright"	0	"all"
<code>stat_cdfinterval()</code>	<code>aes(thickness = f)</code>	"cdf"	"topleft"	0.5	"none"
<code>stat_ccdfinterval()</code>	<code>aes(thickness = f)</code>	"ccdf"	"topleft"	0.5	"none"



Example on **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 on **distributional vectors**

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

ggplot(df) +
  aes(
    y = group,
    xdist = dist_normal(mean, sd)
  ) +
  stat_halfeye()
```

