

EDS 240: Discussion 4

Recreating US Drought Monitor viz

Part 2: ggplot themes

Week 4 | January 27th, 2025

Themes modify *non-data* plot elements

Themes are used to modify the **non-data components** of plots (e.g. titles, labels, fonts, background, gridlines, legend).

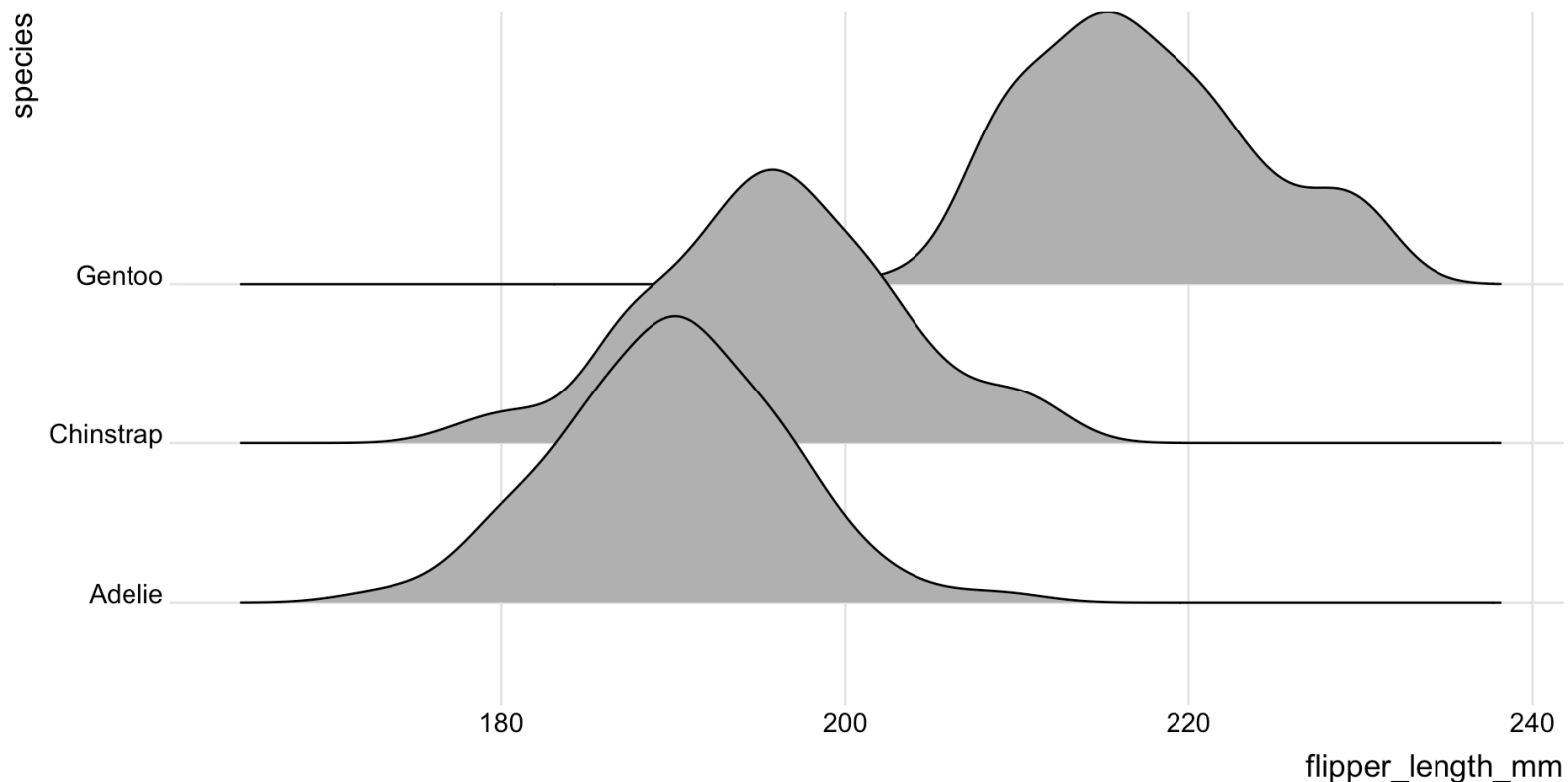
In addition to using **pre-built themes** (available via the `{ggplot2}` package and also from extension packages), you can **fine-tune** the appearance of your plots theme by making adjustments using the `theme()` function.

Pre-built themes

`{ggplot2}` comes with **eight complete themes**, which can be applied as-is, or further modified using `theme()`. There are also *many* **additional themes** which can be applied via **ggplot extension packages**. A small handful of packages with additional themes:

Keep an eye out for plot-specific themes

Keep your eye out for extension **packages that supply both a geom(s) *and* a pre-build theme(s) designed specifically to work with that geom.** For example, the `{ggridges}` package provides both a few different ridgeline plot geoms and a pre-built `theme_ridges()` theme to pair with them:



Use `ggplot2::theme()` for fine-tune control

`{ggplot2}`

theme elements

Set a complete theme as baseline theme, then modify as needed:

```
plot +  
  theme_*() +  
  theme(theme.element = element_type())
```

Use `element_blank()` to remove an element

Axis titles, text, ticks, and lines can be specified per axis using theme inheritance by putting `.x / .y` at the end of the theme element.

`axis.line.y = element_line()`

`axis.title.y = element_line()`

`panel.grid.major = element_line()`
(gridlines at values)

`panel.grid.minor = element_line()`
(gridlines in-between values)

`axis.text.y`

`axis.text = element_text()`

`axis.text.x`

```
plot.title.position = "plot"  
plot.subtitle.position = "plot"  
plot.title = element_text()  
plot.subtitle = element_text()
```

< "plot" means that they will be left-aligned with the plot (rather than the panel)

`plot.margin = margin(t = 25, r = 25, b = 25, l = 25)`

`plot.background = element_rect()`

`panel.background = element_rect()`

`legend.title = element_text()`

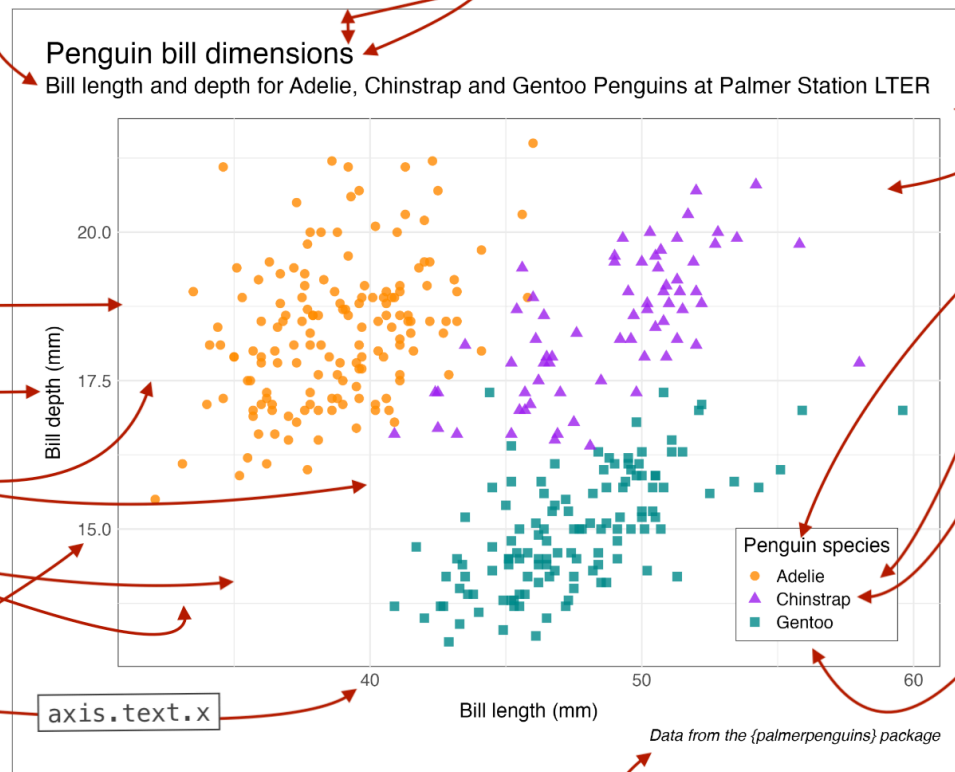
`legend.background = element_rect()`

`legend.text = element_text()`

`legend.position = c(0.85, 0.15) /`
"none" / "left" /
"right" / "bottom" /
"top"

`plot.caption = element_text()`

`text = element_text()` < modifications will be applied to all text elements



Recreate this USDM plot

Reference the graphic on the previous slide (and of course, documentation – run `?theme` or check out [online documentation](#)) to start tweaking plot elements until it matches the original USDM graphic (below). It's common to start with a pre-built theme and modify from there.

