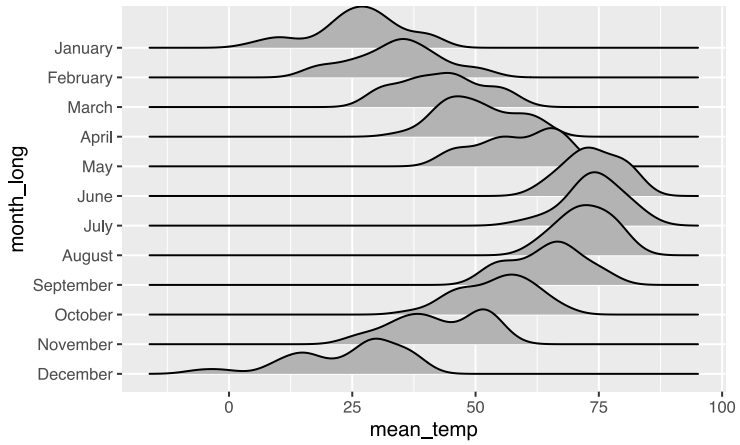
Figure design

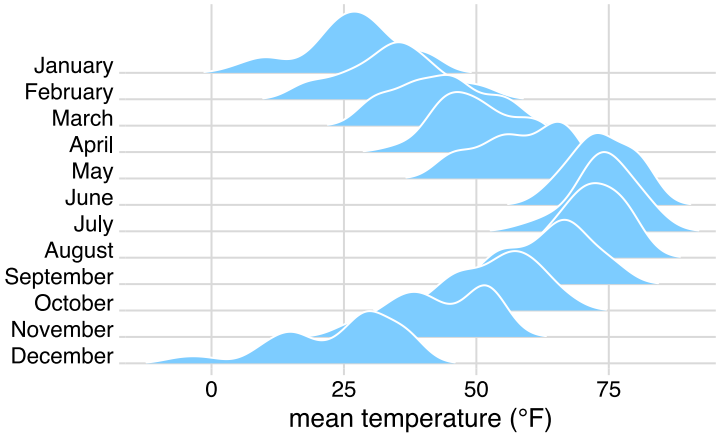
Claus O. Wilke

last updated: 2021-02-11

How do you go from this ...



... to this?

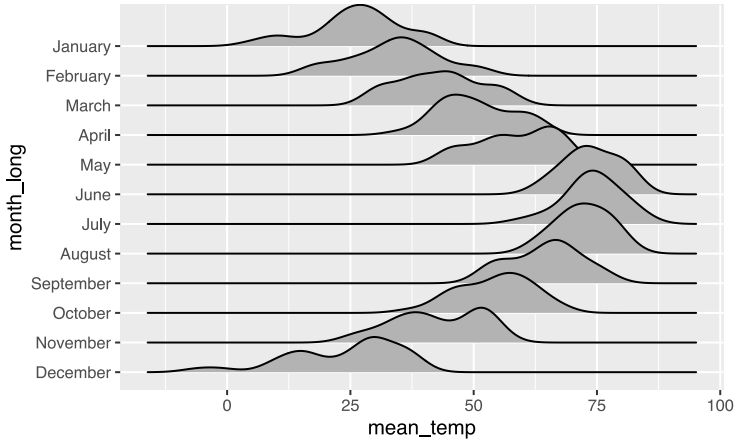


Requires coordinated modification of multiple elements:

- geoms (via arguments to geoms)
- scales (via `scale_*()` functions)
- plot appearance (via themes)

The starting point, a rough draft

```
ggplot(lincoln_temps) +
  aes(x = mean_temp, y = month_lo
  geom_density_ridges()
```

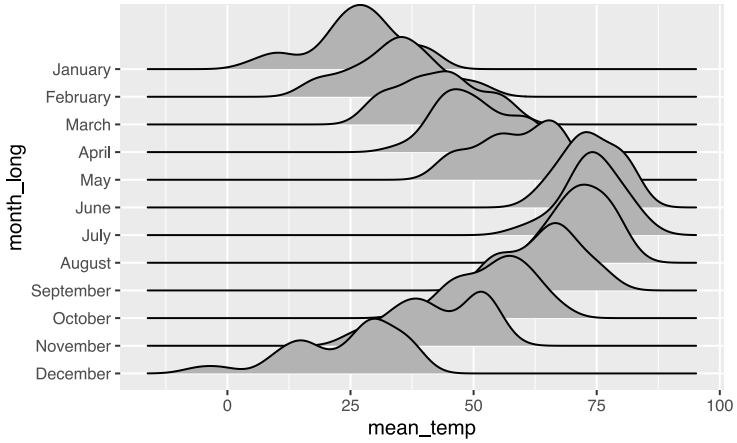


You can download the dataset using this code:

```
lincoln_temps <- readRDS(
  url("https://wilkelab.org/SDS375/datasets/lincoln_temps.rds")
)
```

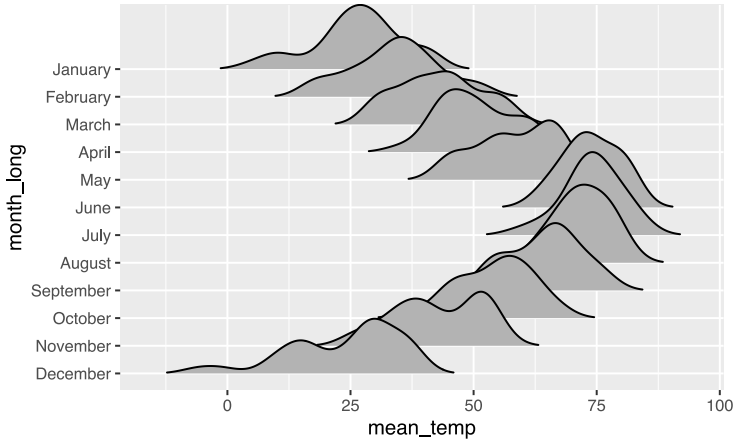
Set `scale` and `bandwidth` to shape ridgelines

```
ggplot(lincoln_temps) +
  aes(x = mean_temp, y = month_lo
  geom_density_ridges(
    scale = 3, bandwidth = 3.4
  )
```



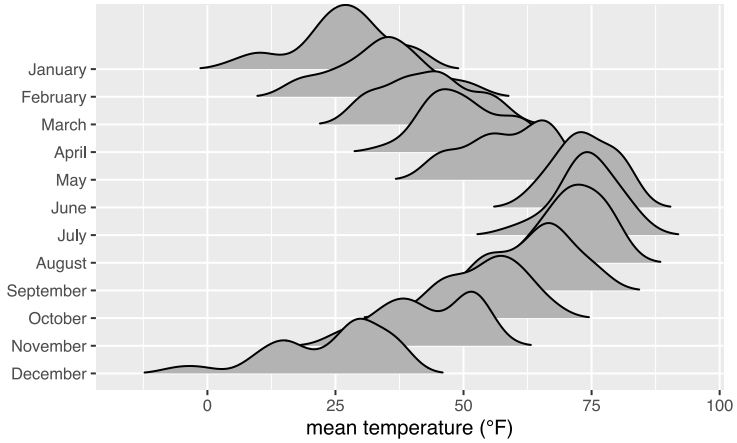
Set `rel_min_height` to cut ridgelines near zero

```
ggplot(lincoln_temps) +
  aes(x = mean_temp, y = month_lo
  geom_density_ridges(
    scale = 3, bandwidth = 3.4,
    rel_min_height = 0.01
  )
```



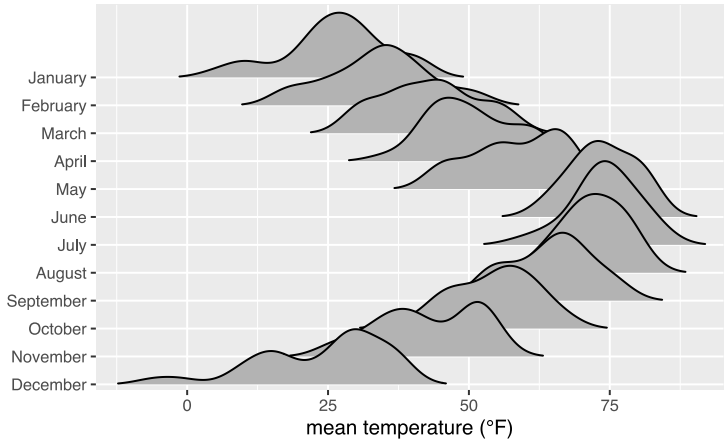
Use `scale_*()` functions to specify axis labels

```
ggplot(lincoln_temps) +
  aes(x = mean_temp, y = month_lo
  geom_density_ridges(
    scale = 3, bandwidth = 3.4,
    rel_min_height = 0.01,
  ) +
  scale_x_continuous(
    name = "mean temperature (°F)
  ) +
  scale_y_discrete(
    name = NULL  # NULL means no
  )
```



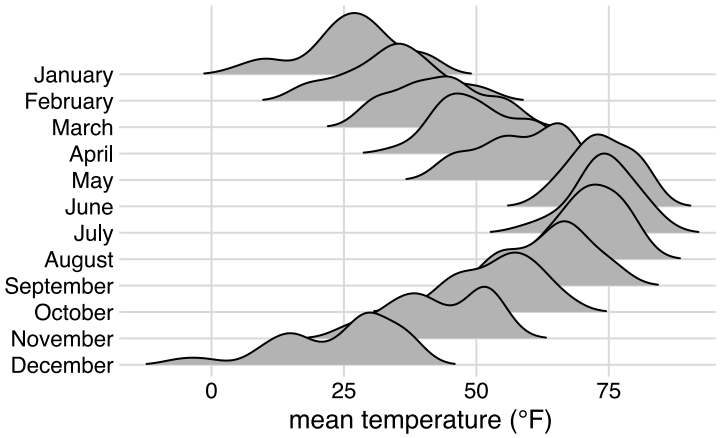
Specify scale expansion

```
ggplot(lincoln_temps) +
  aes(x = mean_temp, y = month_lo
  geom_density_ridges(
    scale = 3, bandwidth = 3.4,
    rel_min_height = 0.01
  ) +
  scale_x_continuous(
    name = "mean temperature (°F)
    expand = c(0, 0)
  ) +
  scale_y_discrete(
    name = NULL,
    expand = expansion(add = c(0.
  )
```



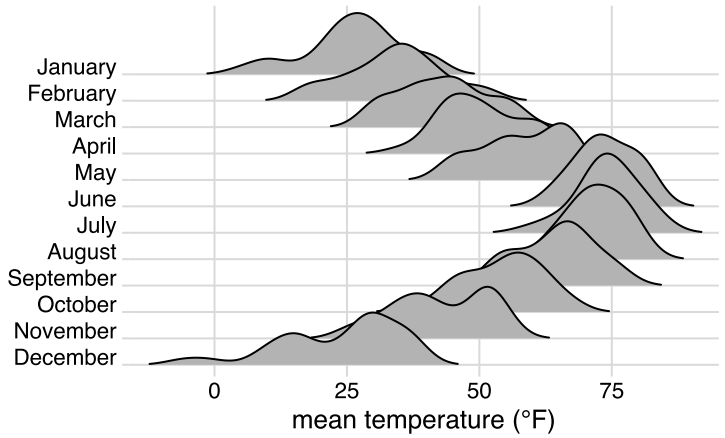
Set overall plot theme

```
ggplot(lincoln_temps) +
  aes(x = mean_temp, y = month_lo
  geom_density_ridges(
    scale = 3, bandwidth = 3.4,
    rel_min_height = 0.01
  ) +
  scale_x_continuous(
    name = "mean temperature (°F)
    expand = c(0, 0)
  ) +
  scale_y_discrete(
    name = NULL,
    expand = expansion(add = c(0.
  ) +
  theme_minimal_grid()  # from co
```



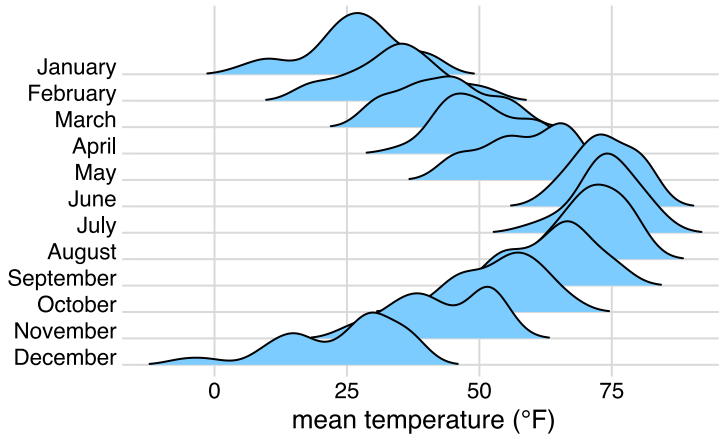
Align y axis labels to grid lines

```
ggplot(lincoln_temps) +
  aes(x = mean_temp, y = month_lo
  geom_density_ridges(
    scale = 3, bandwidth = 3.4,
    rel_min_height = 0.01
  ) +
  scale_x_continuous(
    name = "mean temperature (°F)
    expand = c(0, 0)
  ) +
  scale_y_discrete(
    name = NULL,
    expand = expansion(add = c(0.
  ) +
  theme_minimal_grid() +
  theme(
    axis.text.y = element_text(vj
  )
```



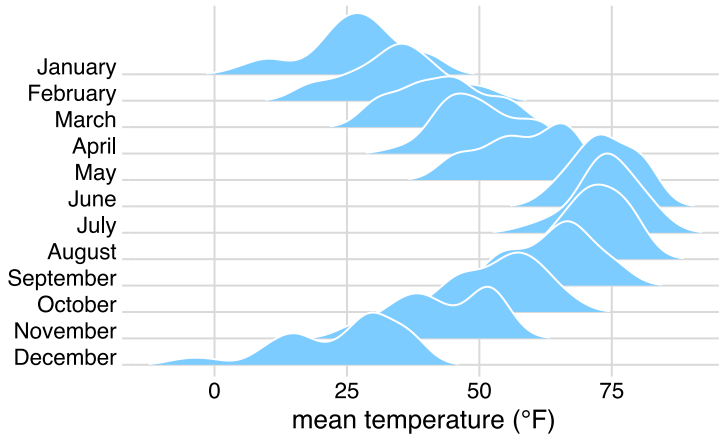
Change fill color from default gray to blue

```
ggplot(lincoln_temps) +
  aes(x = mean_temp, y = month_lo
  geom_density_ridges(
    scale = 3, bandwidth = 3.4,
    rel_min_height = 0.01,
    fill = "#7DCCFF"
  ) +
  scale_x_continuous(
    name = "mean temperature (°F)
    expand = c(0, 0)
  ) +
  scale_y_discrete(
    name = NULL,
    expand = expansion(add = c(0.
  ) +
  theme_minimal_grid() +
  theme(
    axis.text.y = element_text(vj
  )
```



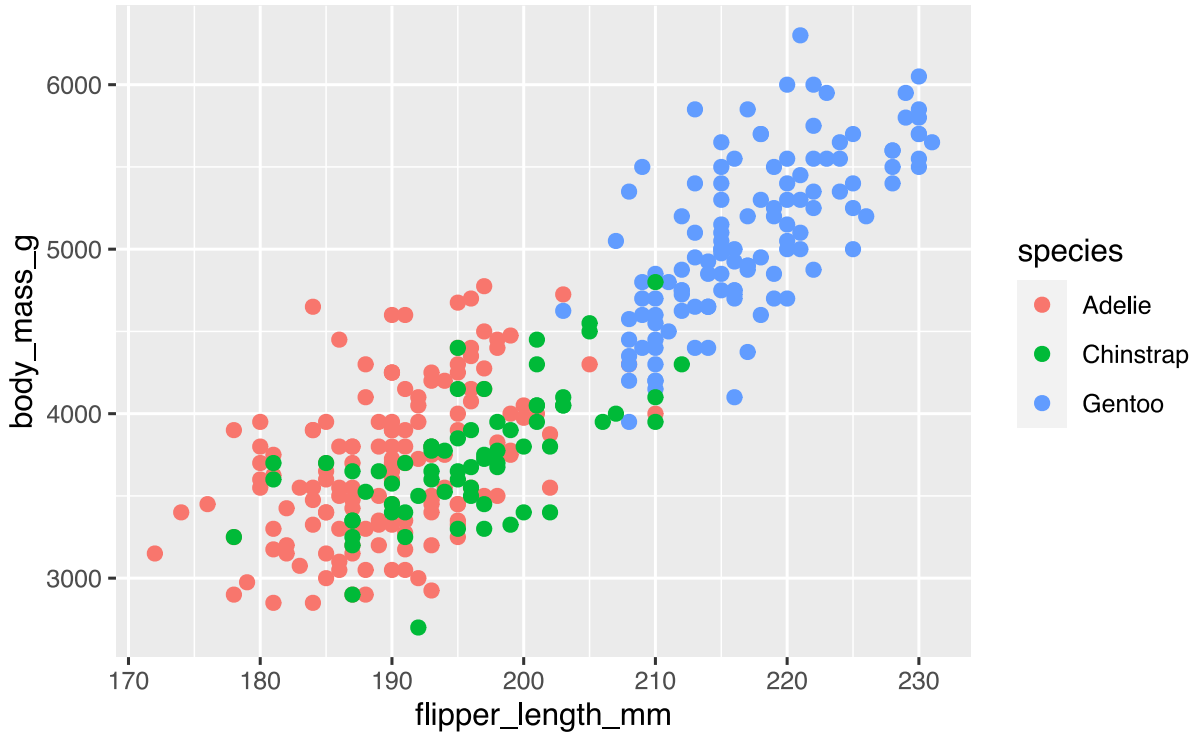
Draw lines in white instead of black

```
ggplot(lincoln_temps) +
  aes(x = mean_temp, y = month_lo
  geom_density_ridges(
    scale = 3, bandwidth = 3.4,
    rel_min_height = 0.01,
    fill = "#7DCCFF",
    color = "white"
  ) +
  scale_x_continuous(
    name = "mean temperature (°F)
    expand = c(0, 0)
  ) +
  scale_y_discrete(
    name = NULL,
    expand = expansion(add = c(0.
  ) +
  theme_minimal_grid() +
  theme(
    axis.text.y = element_text(vj
  )
```



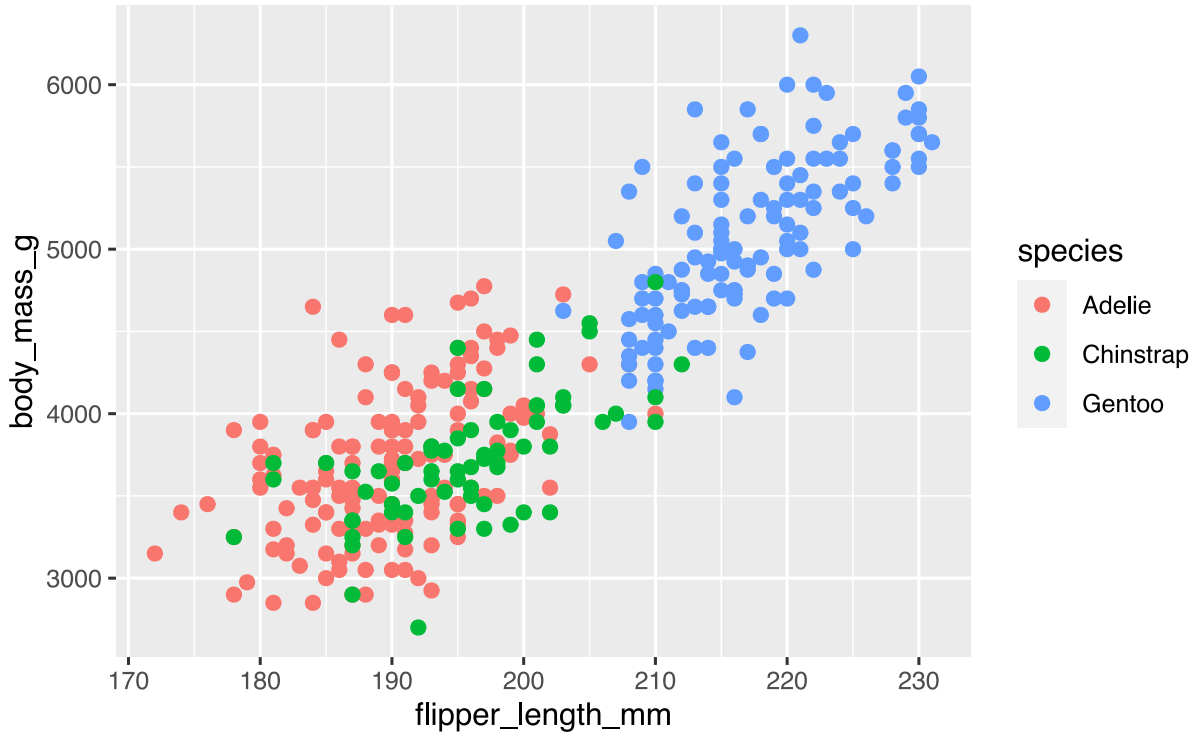
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point()
  # default theme is theme_gray()
```



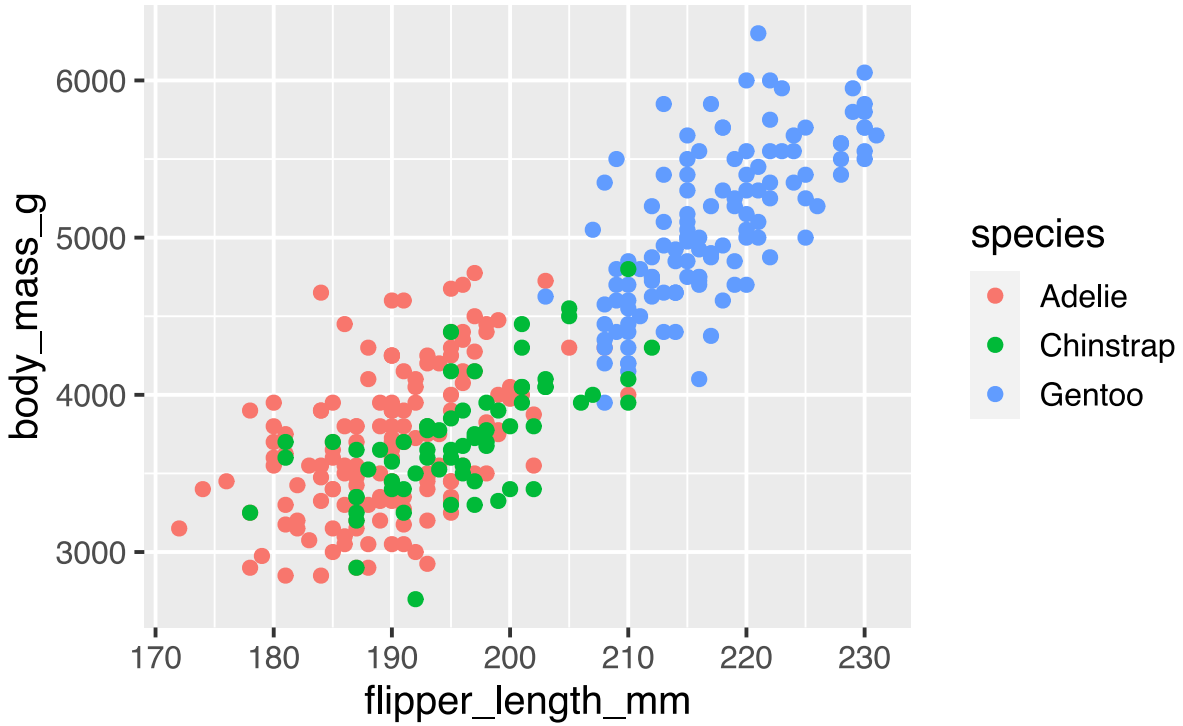
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_gray()
```



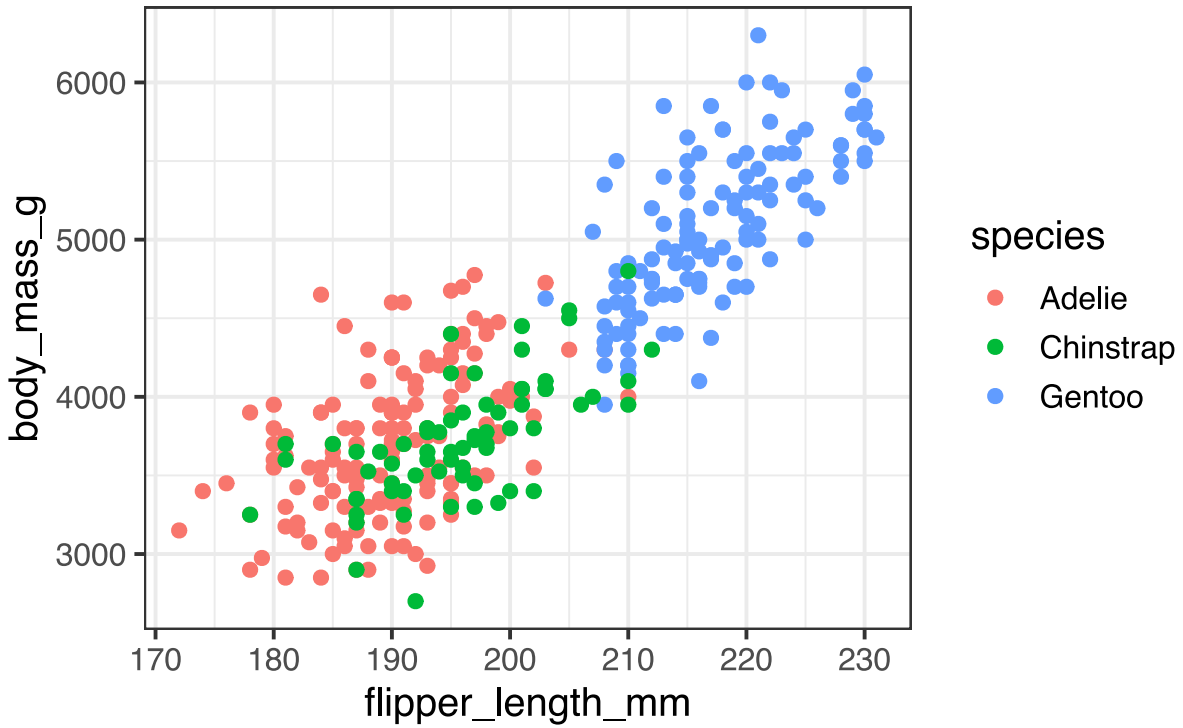
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_gray(14) # most themes take a font-size argument to scale text s
```



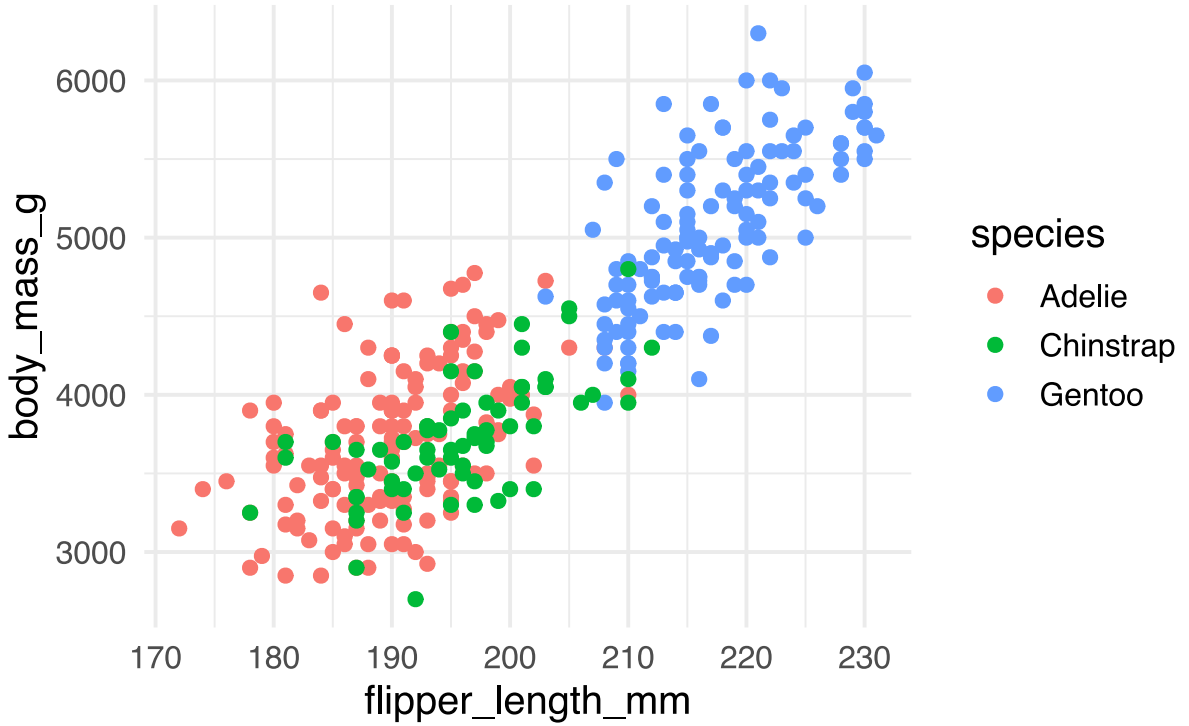
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_bw(14)
```



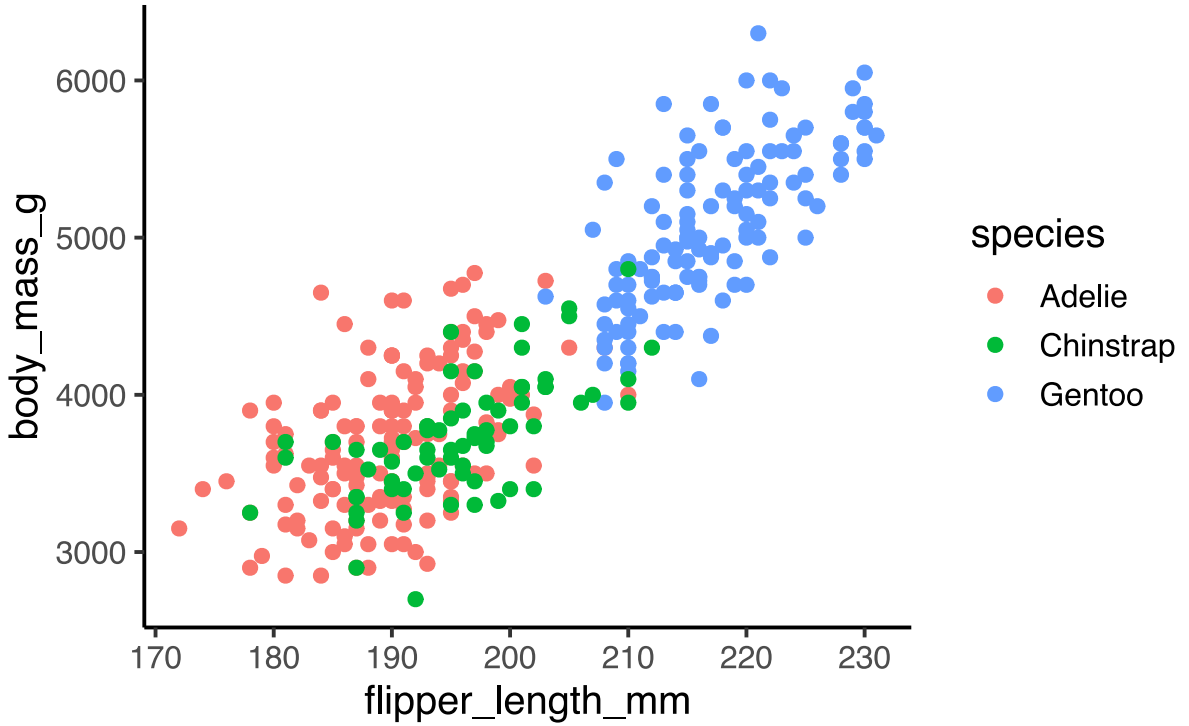
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_minimal(14)
```



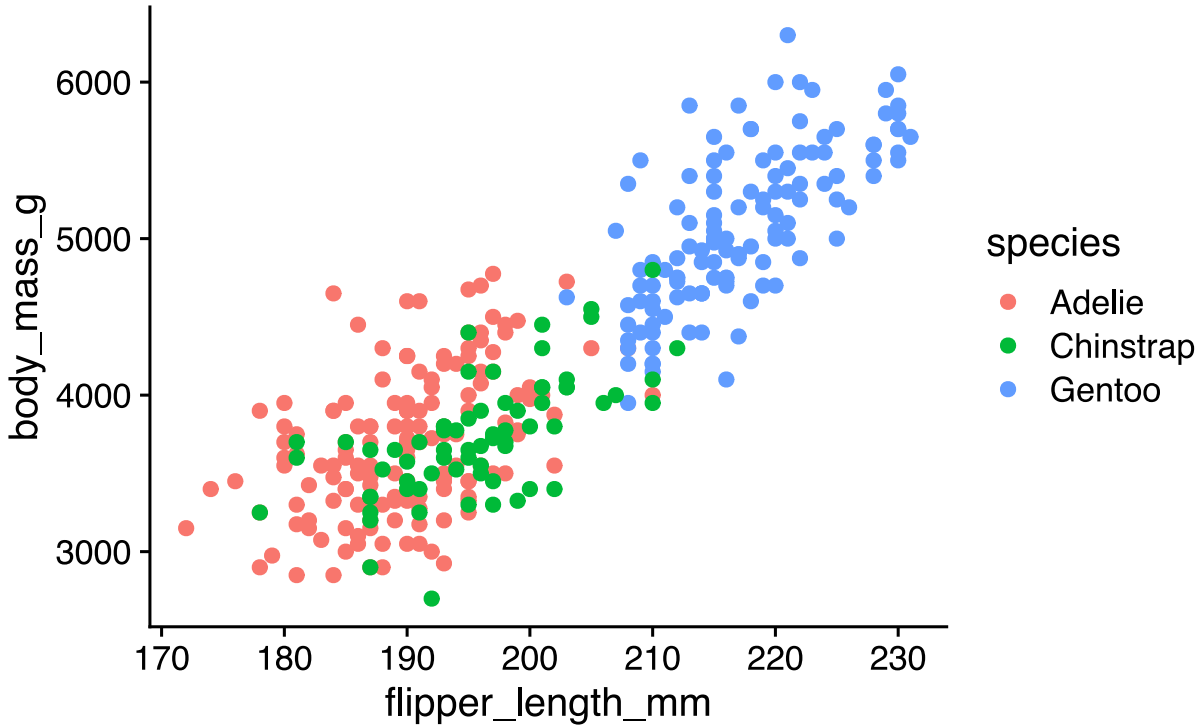
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_classic(14)
```



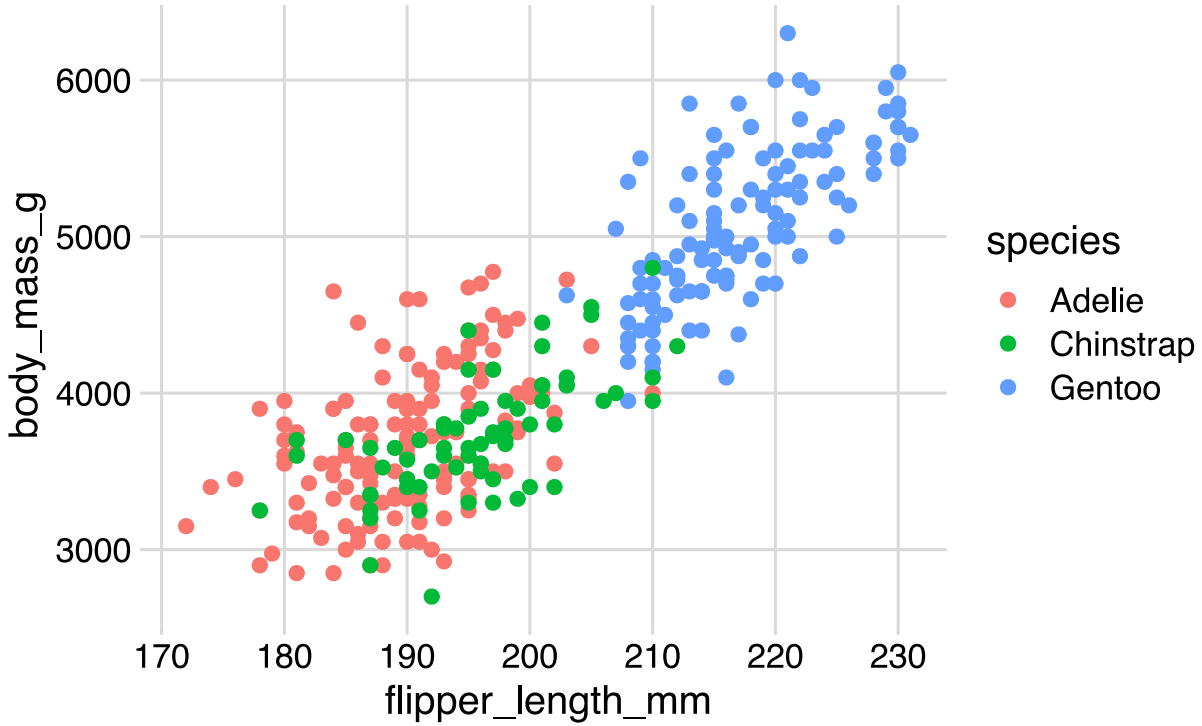
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_half_open()  # from package cowplot
```



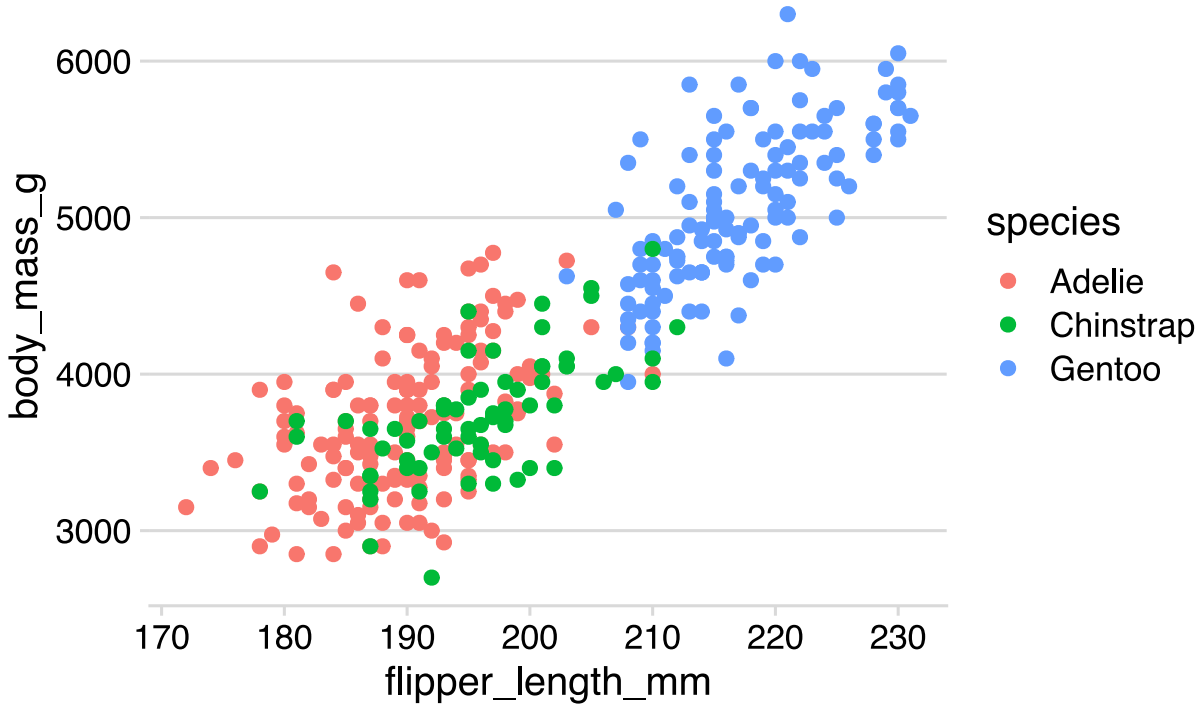
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_minimal_grid()  # from package cowplot
```



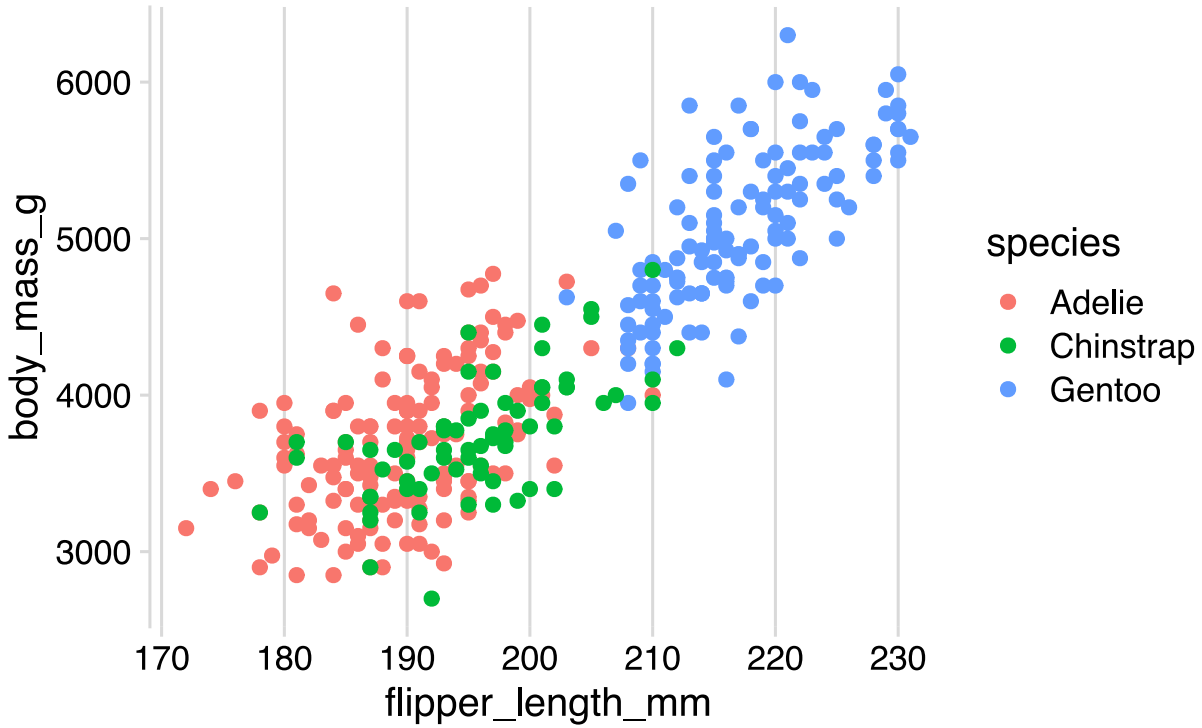
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_minimal_hgrid()  # from package cowplot
```



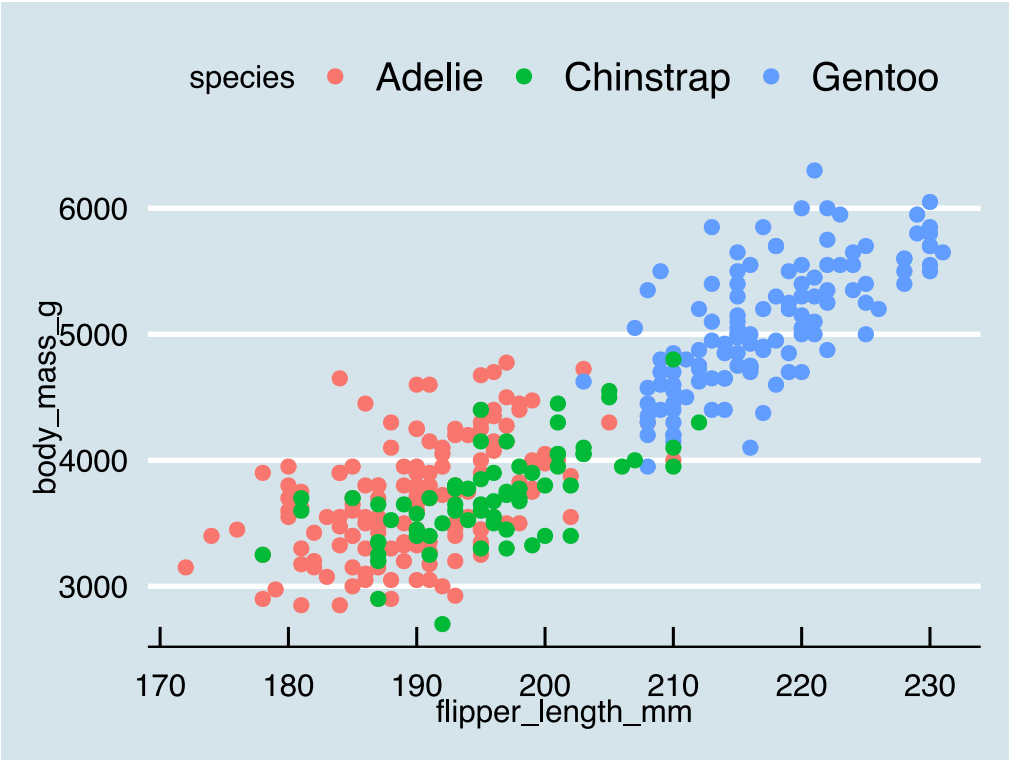
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_minimal_vgrid()  # from package cowplot
```



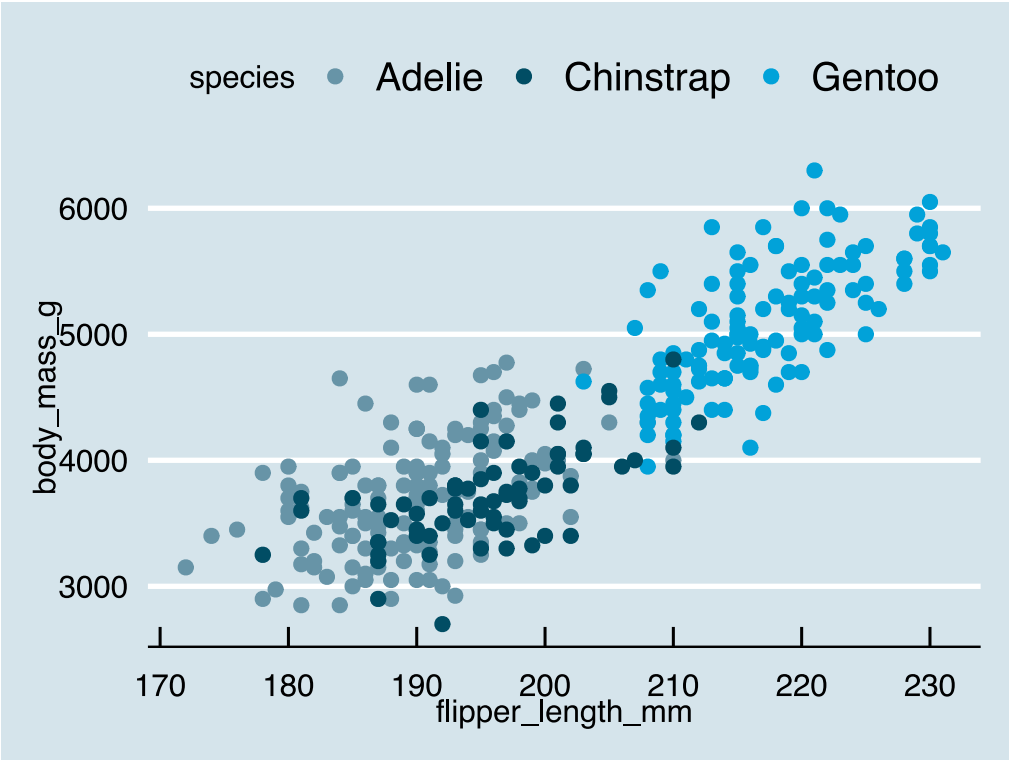
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_economist(14)       # from package ggthemes
```



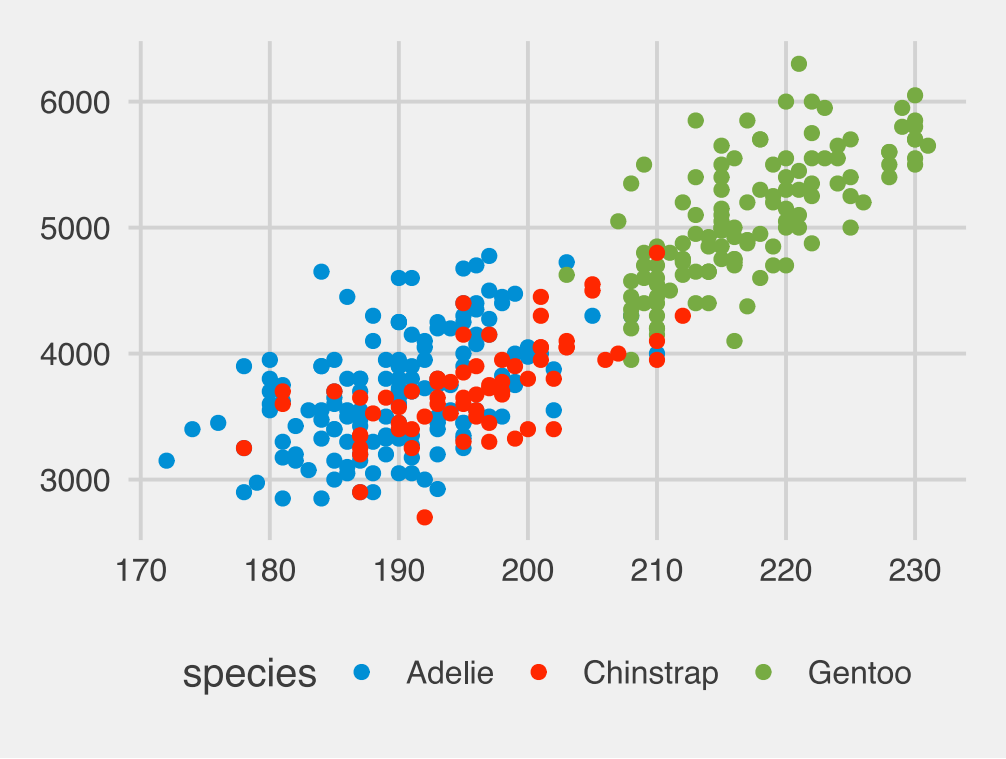
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_economist(14) + scale_color_economist() # from package ggthemes
```



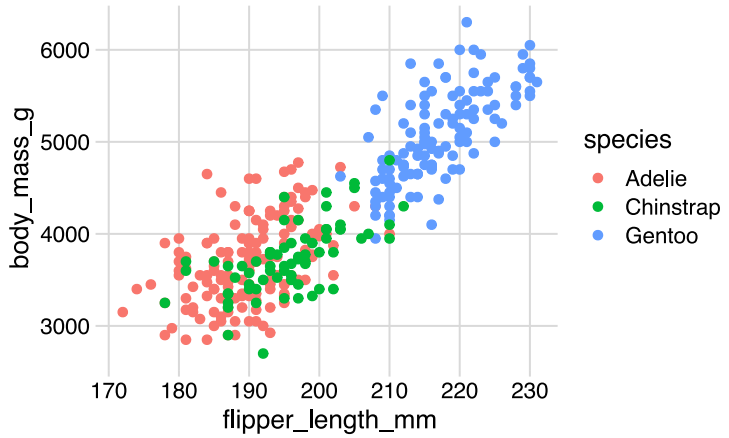
Using ready-made themes

```
ggplot(penguins, aes(flipper_length_mm, body_mass_g, color = species)) +
  geom_point() +
  theme_fivethirtyeight(14) + scale_color_fivethirtyeight() # from packa
```



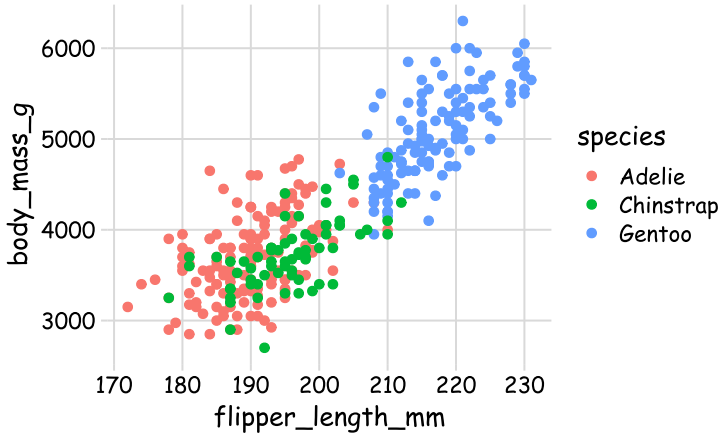
Customizing theme elements

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid()
```



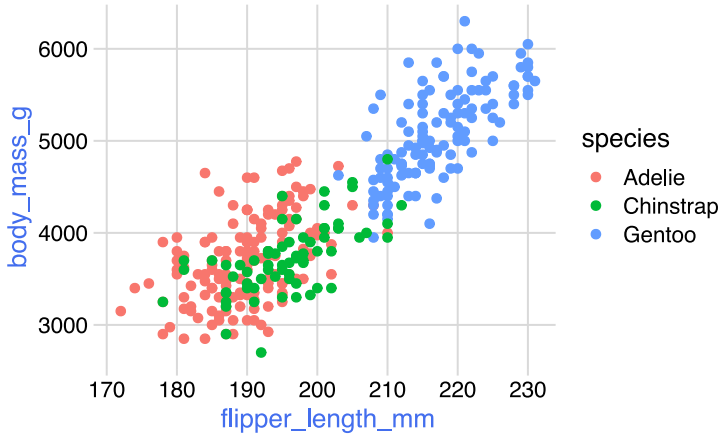
Customizing theme elements

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    # change overall font family
    # (requires font to be availa
    text = element_text(
      family = "Comic Sans MS"
    )
  )
```



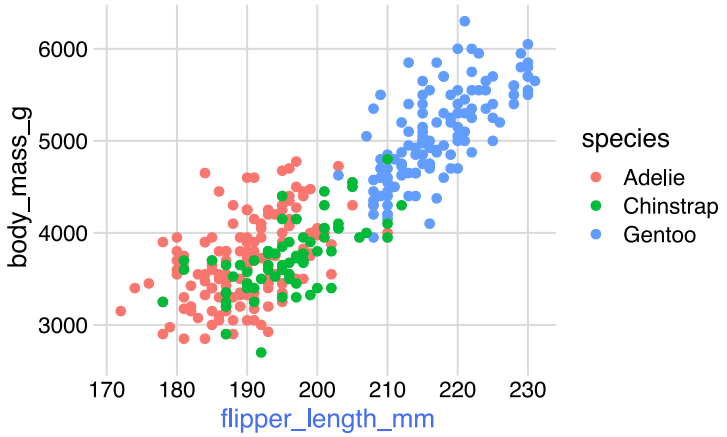
Customizing theme elements

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    # change color of axis titles
    axis.title = element_text(
      color = "royalblue2"
    )
  )
```



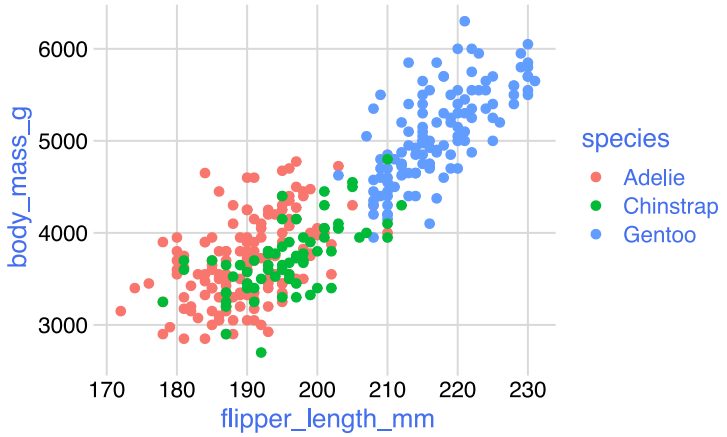
Customizing theme elements

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    # change color of only the x
    axis.title.x = element_text(
      color = "royalblue2"
    )
  )
```



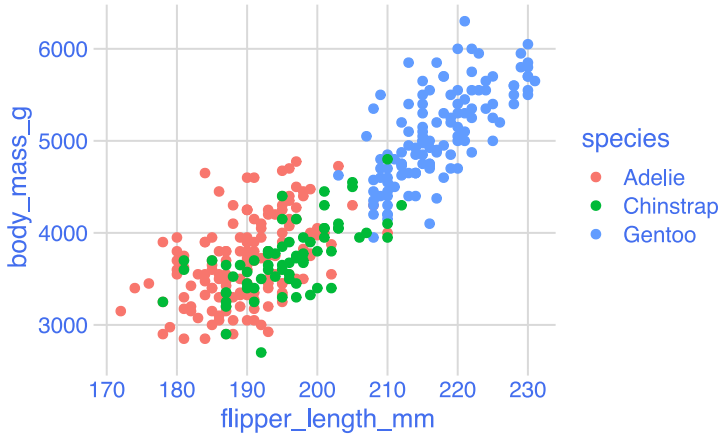
Customizing theme elements

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    # change all text colors?
    # why does it not work?
    text = element_text(color = "
  )
```



Customizing theme elements

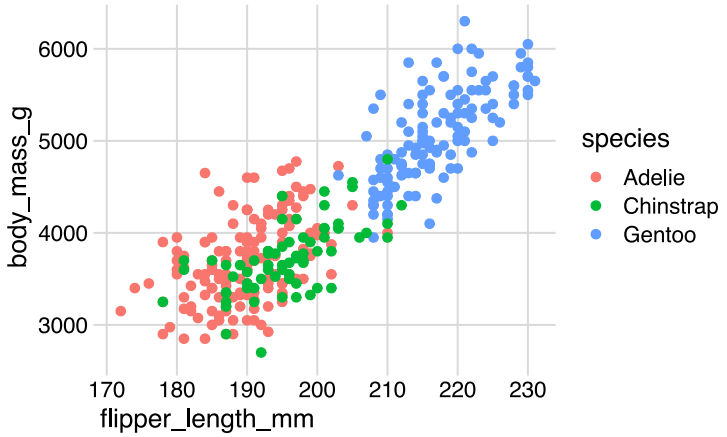
```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    text = element_text(color = "
    axis.text = element_text(
      color = "royalblue2"
    )
  )
```



The element `axis.text` has its own color set in the theme. Therefore it doesn't inherit from `text`.

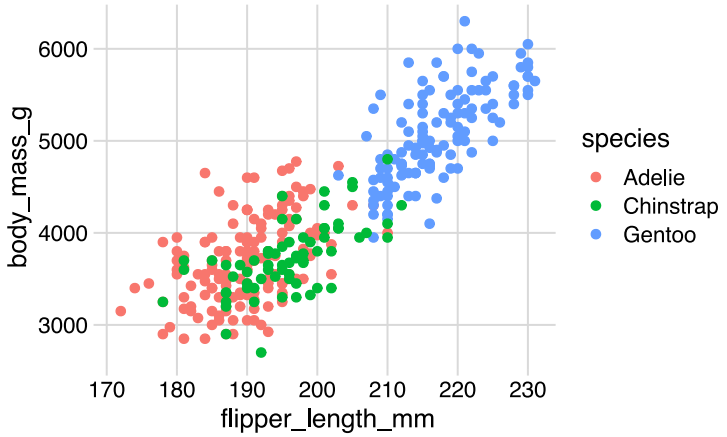
Horizontal and vertical alignment

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    axis.title.x = element_text(
      # horizontal justification
      # (0 = left)
      hjust = 0
    )
  )
```



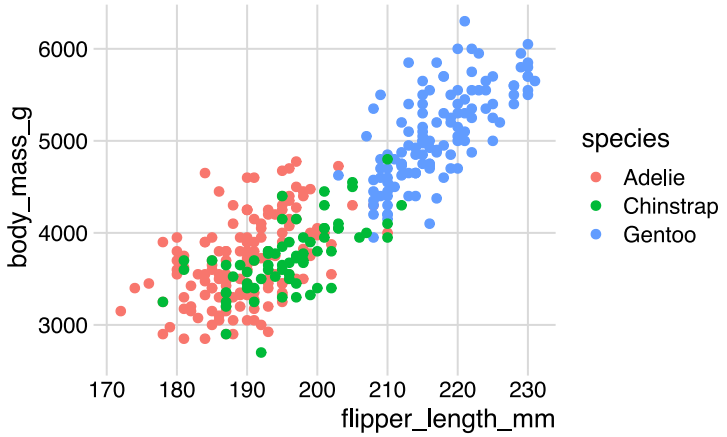
Horizontal and vertical alignment

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    axis.title.x = element_text(
      # horizontal justification
      # (0.5 = center)
      hjust = 0.5
    )
  )
```



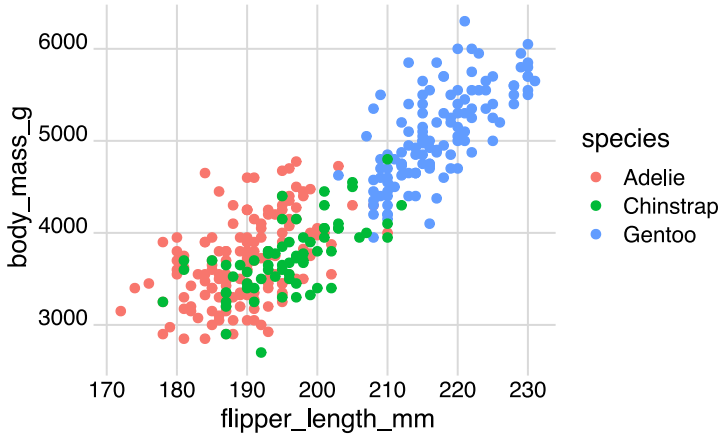
Horizontal and vertical alignment

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    axis.title.x = element_text(
      # horizontal justification
      # (1 = right)
      hjust = 1
    )
  )
```



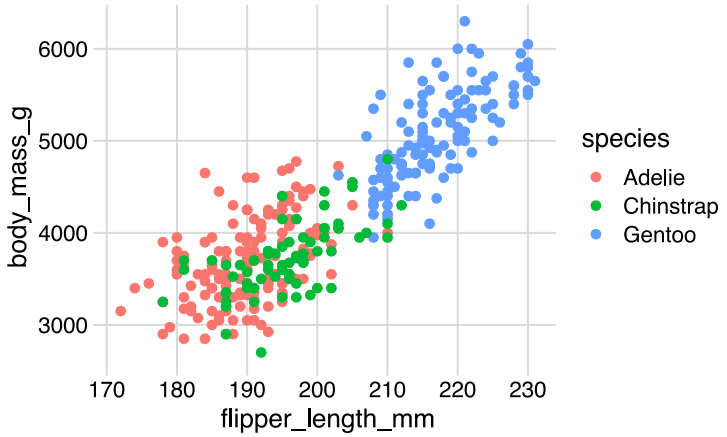
Horizontal and vertical alignment

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    axis.text.y = element_text(
      # vertical justification
      # (0 = bottom)
      vjust = 0
    )
  )
```



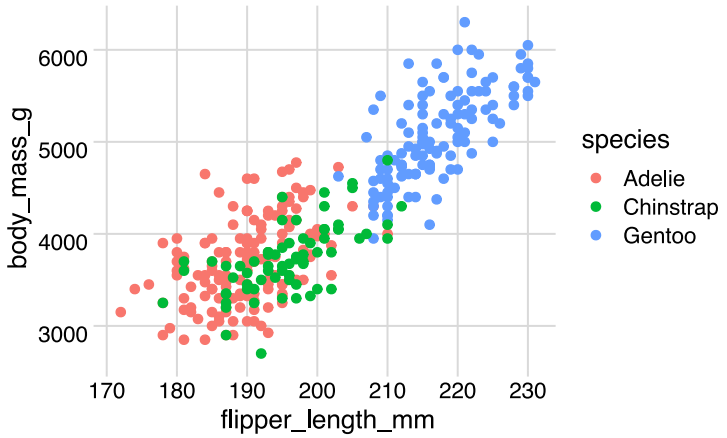
Horizontal and vertical alignment

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    axis.text.y = element_text(
      # vertical justification
      # (0.5 = center)
      vjust = 0.5
    )
  )
```



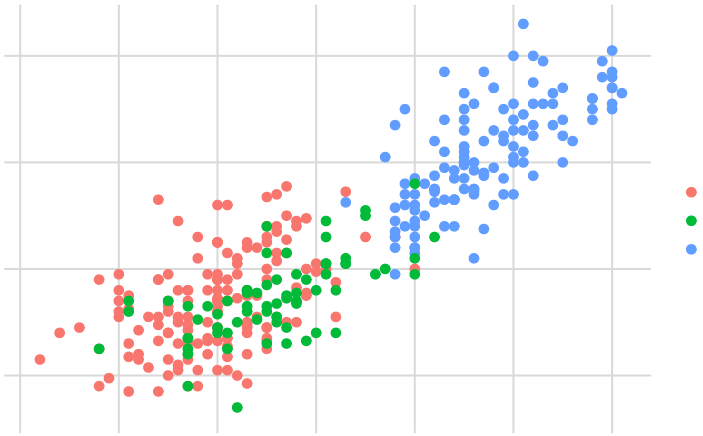
Horizontal and vertical alignment

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    axis.text.y = element_text(
      # vertical justification
      # (1 = top)
      vjust = 1
    )
  )
```



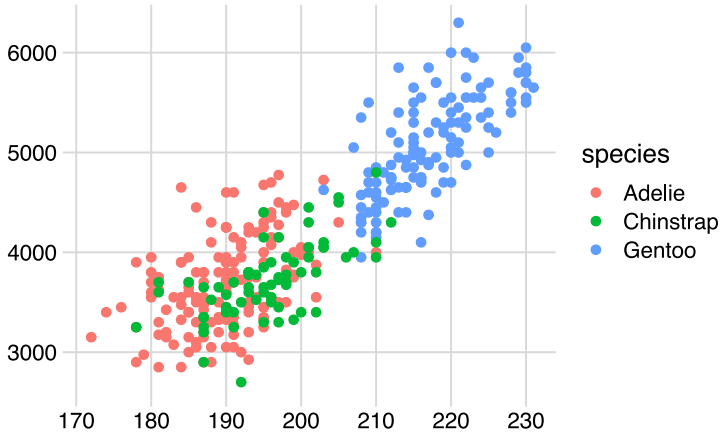
Remove elements entirely: `element_blank()`

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    # all text gone
    text = element_blank()
  )
```



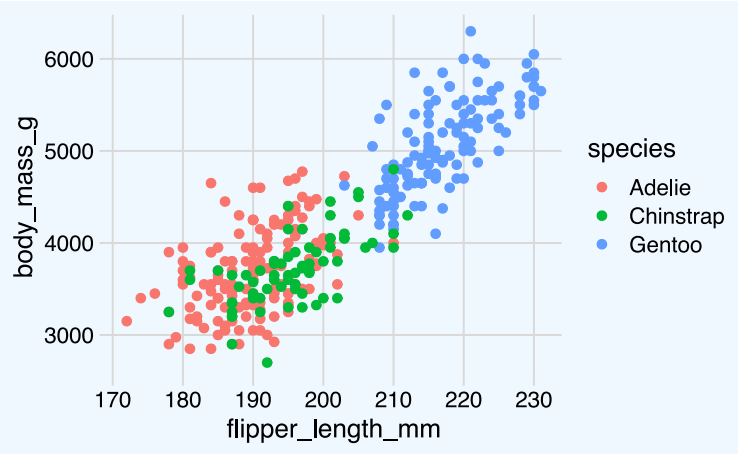
Remove elements entirely: `element_blank()`

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    # no axis titles
    axis.title = element_blank()
  )
```



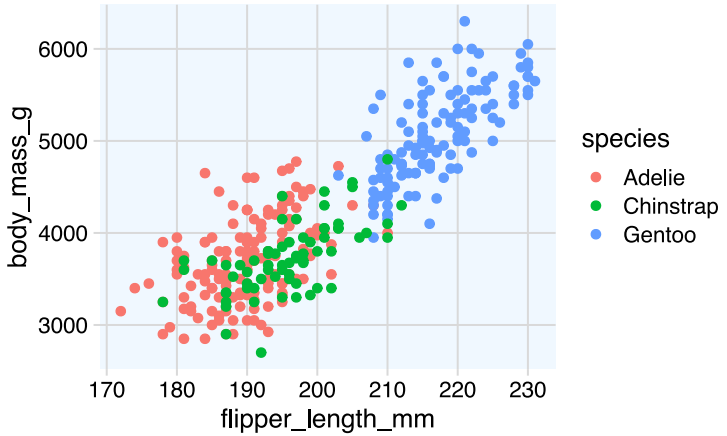
Set background color: `element_rect()`

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    plot.background = element_rec
      fill = "aliceblue"
    )
  )
```



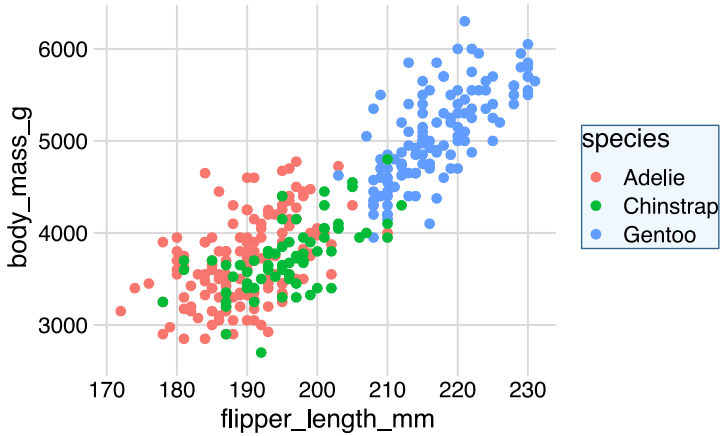
Set background color: `element_rect()`

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    panel.background = element_re
      fill = "aliceblue"
    )
  )
```



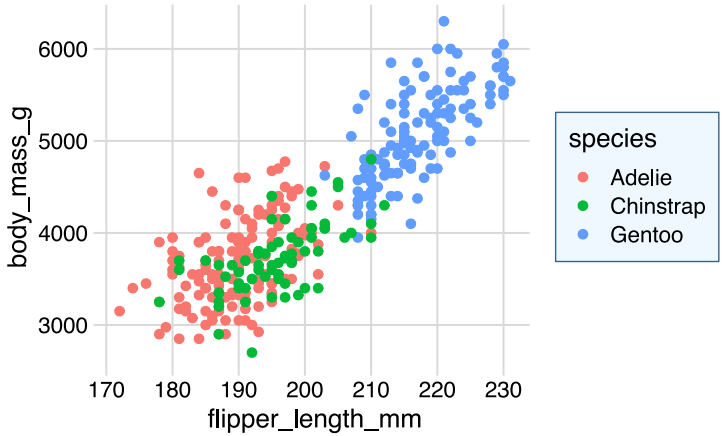
Set background color: `element_rect()`

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    legend.box.background = eleme
      fill = "aliceblue",
      color = "steelblue4" # line
    )
  )
```



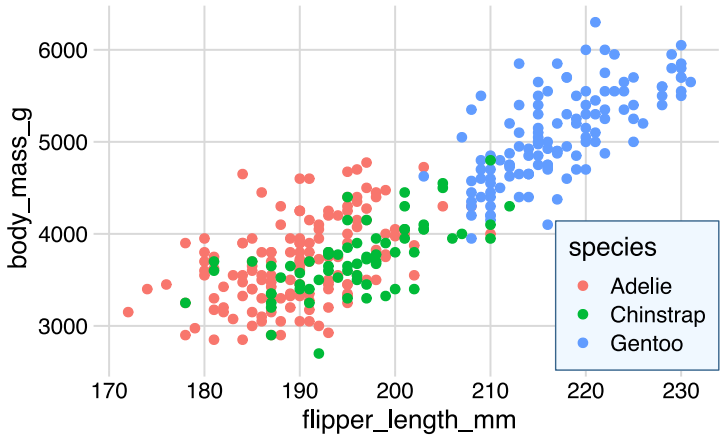
Set background color: `element_rect()`

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    legend.box.background = eleme
      fill = "aliceblue",
      color = "steelblue4" # line
    ),
    legend.box.margin = margin(7,
  )
```



Move the legend: `legend.position`

```
ggplot(penguins) +
  aes(flipper_length_mm, body_mas
  geom_point(aes(color = species)
  theme_minimal_grid() +
  theme(
    legend.box.background = eleme
      fill = "aliceblue",
      color = "steelblue4" # line
    ),
    legend.box.margin = margin(7,
    # relative position inside pl
    legend.position = c(1, 0),
    # justification relative to p
    legend.justification = c(1, 0
  )
```



Further reading

- Fundamentals of Data Visualization: Chapter 23: Balance the data and the context
- Data Visualization—A Practical Introduction: Chapter 8.3: Change the appearance of plots with themes
- ggplot2 reference documentation: Complete themes
- ggplot2 reference documentation: Modify components of a theme