

Implementing the Grammar of Graphics with `{ggplot2}`

Introduction to Data Science

PDF version of slides

{ggplot2}

A package that implements a *layered* grammar of graphics



{tidyverse}



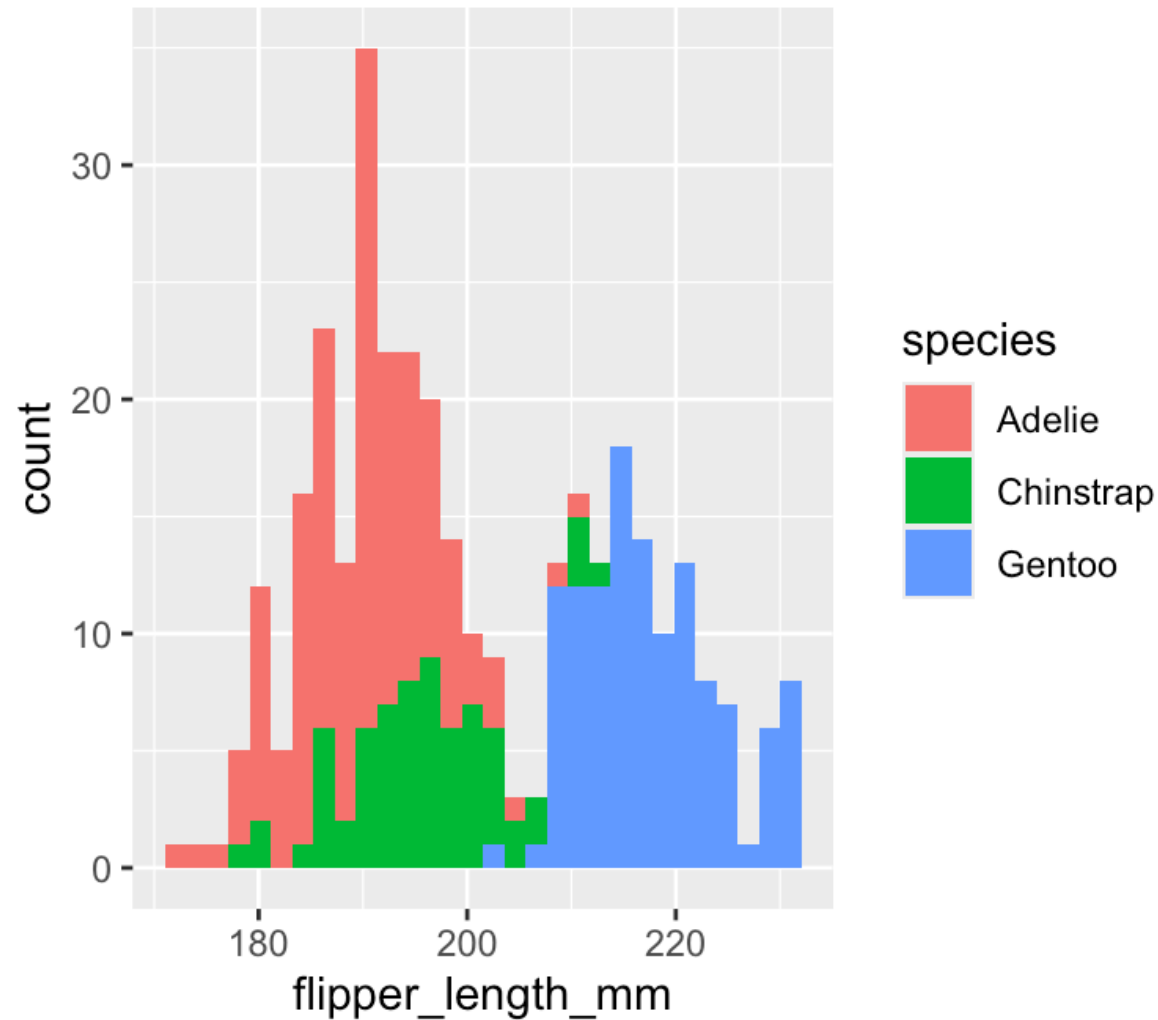
- An opinionated collection of R packages designed for data science
- All packages share an underlying philosophy and a common grammar

`library(tidyverse)`
tidyverse.org

Minimalist plot template


```
1 ggplot(data = <DATA>) +  
2   <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```


Example





```
1 ggplot(data = penguins) +  
2   geom_histogram(mapping = aes(x = flipper_length_mm, fill = species))
```

RStudio cheatsheet lays out all geom options

 **a + geom_path**(lineend = "butt", linejoin = "round", linemitre = 1) - x, y, alpha, color, group, linetype, size


 **a + geom_polygon**(aes(alpha = 50)) - x, y, alpha, color, fill, group, subgroup, linetype, size


 **b + geom_rect**(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size


 **a + geom_ribbon**(aes(ymin = unemploy - 900, ymax = unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

 **b + geom_abline**(aes(intercept = 0, slope = 1))

 **b + geom_hline**(aes(yintercept = lat))


 **b + geom_vline**(aes(xintercept = long))


b + geom_segment(aes(yend = lat + 1, xend = long + 1))


b + geom_spoke(aes(angle = 1:1155, radius = 1))


ONE VARIABLE continuous


c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)


 **c + geom_area**(stat = "bin") - x, y, alpha, color, fill, linetype, size

 **c + geom_density**(kernel = "gaussian") - x, y, alpha, color, fill, group, linetype, size, weight

 **c + geom_dotplot**() - x, y, alpha, color, fill


 **c + geom_freqpoly**() - x, y, alpha, color, group, linetype, size


 **c + geom_histogram**(binwidth = 5) - x, y, alpha, color, fill, linetype, size, weight


 **c2 + geom_qq**(aes(sample = hwy)) - x, y, alpha, color, fill, linetype, size, weight


discrete


d <- ggplot(mpg, aes(fl))

 **d + geom_bar**() - x, alpha, color, fill, linetype, size, weight

 **e + geom_quantile**() - x, y, alpha, color, group, linetype, size, weight


 **e + geom_rug**(sides = "bl") - x, y, alpha, color, linetype, size


 **e + geom_smooth**(method = lm) - x, y, alpha, color, fill, group, linetype, size, weight


 **e + geom_text**(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust


one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))

 **f + geom_col**() - x, y, alpha, color, fill, group, linetype, size


 **f + geom_boxplot**() - x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight


 **f + geom_dotplot**(binaxis = "y", stackdir = "center") - x, y, alpha, color, fill, group

 **f + geom_violin**(scale = "area") - x, y, alpha, color, fill, group, linetype, size, weight

both discrete


g <- ggplot(diamonds, aes(cut, color))


 **g + geom_count**() - x, y, alpha, color, fill, shape, size, stroke


 **e + geom_jitter**(height = 2, width = 2) - x, y, alpha, color, fill, shape, size

THREE VARIABLES

seals\$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))


 **l + geom_contour**(aes(z = z)) - x, y, z, alpha, color, group, linetype, size, weight


 **l + geom_contour_filled**(aes(fill = z)) - x, y, alpha, color, fill, group, linetype, size, subgroup


 **h + geom_hex**() - x, y, alpha, color, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))

 **i + geom_area**() - x, y, alpha, color, fill, linetype, size


 **i + geom_line**() - x, y, alpha, color, group, linetype, size


 **i + geom_step**(direction = "hv") - x, y, alpha, color, group, linetype, size

visualizing error


df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)


j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

 **j + geom_crossbar**(fatten = 2) - x, y, ymax, ymin, alpha, color, fill, group, linetype, size

 **j + geom_errorbar**() - x, ymax, ymin, alpha, color, group, linetype, size, width

Also **geom_errorbarh**()

 **j + geom_linerange**() - x, ymin, ymax, alpha, color, group, linetype, size


 **j + geom_pointrange**() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

data <- data.frame(murder = USArrests\$Murder, state = tolower(rownames(USArrests)))

map <- map_data("state")

k <- ggplot(data, aes(fill = murder))

 **k + geom_map**(aes(map_id = state), map = map) + **expand_limits**(x = map\$long, y = map\$lat) - map_id, alpha, color, fill, linetype, size

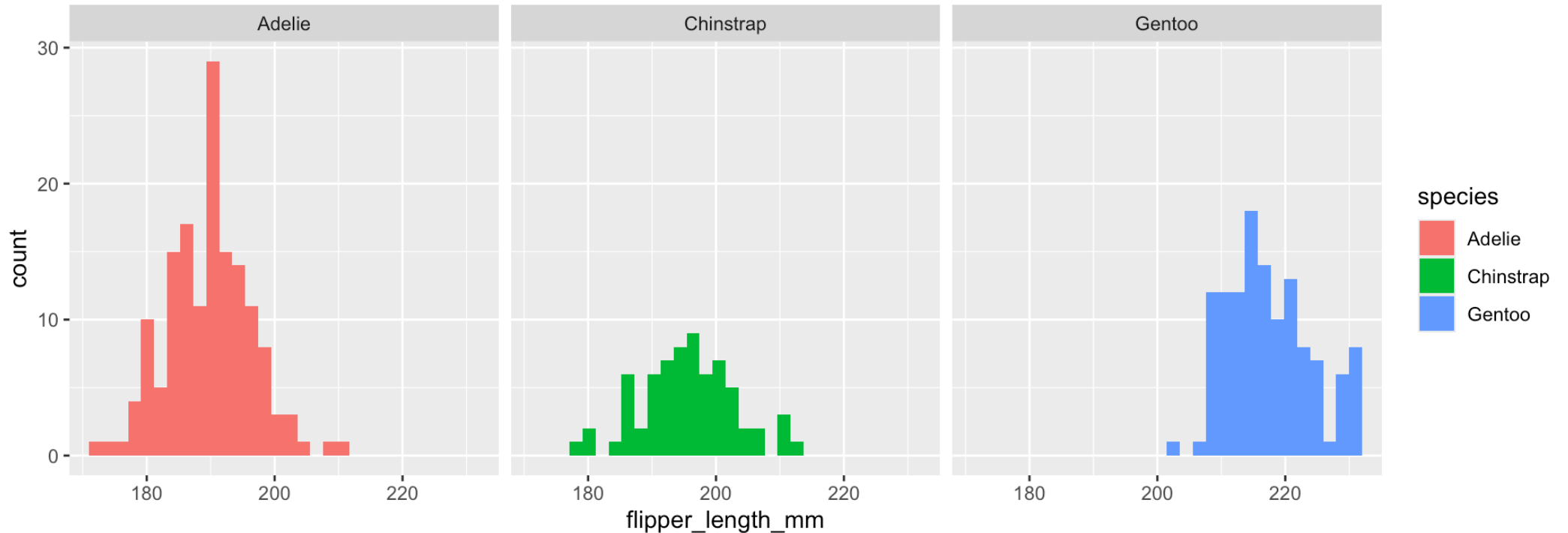
Aesthetics options

Visual characteristics of plotting characters that can be **mapped to a specific variable** in the data are

- `color`
- `size`
- `shape`
- `alpha` (transparency)

Facets

Small multiples are created through **facets**

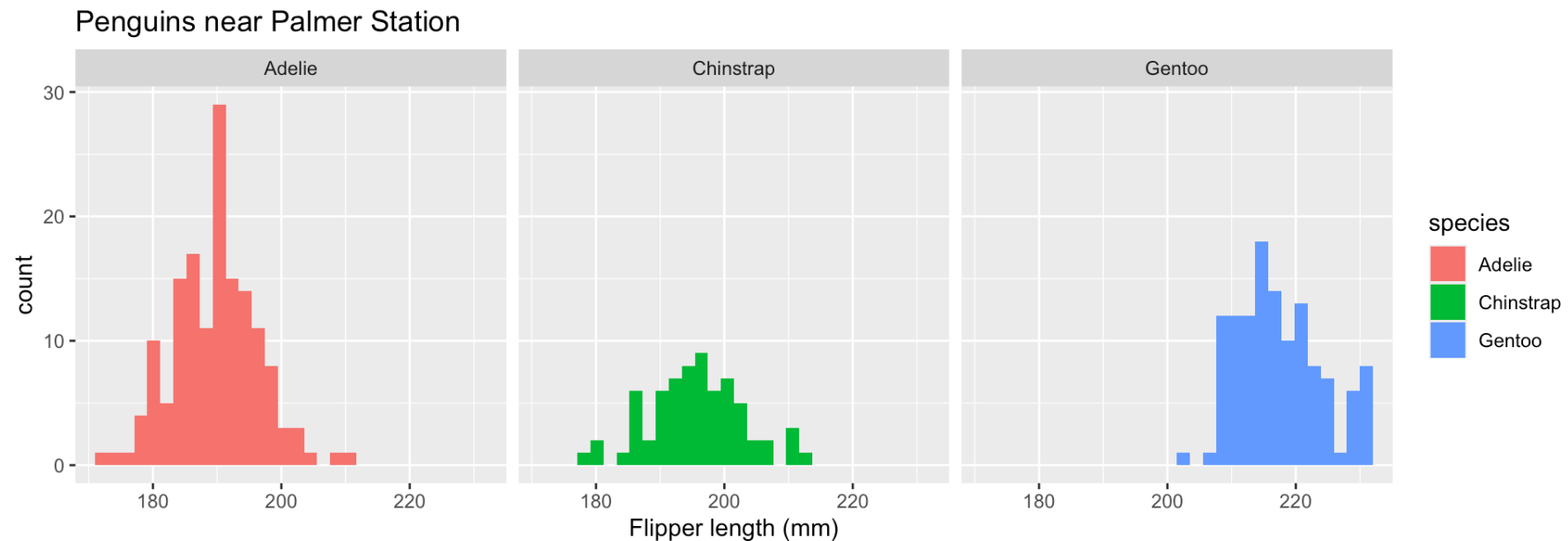


```
1 ggplot(data = penguins) +  
2   geom_histogram(mapping = aes(x = flipper_length_mm, fill = species)) +  
3   facet_wrap(~ species)
```

Theme elements

Non-data components of the plot that can be modified with **theme elements**

```
1 ggplot(data = penguins) +  
2   geom_histogram(mapping = aes(x = flipper_length_mm, fill = species)) +  
3   facet_wrap(~ species) +  
4   labs(x = "Flipper length (mm)",  
5         title = "Penguins near Palmer Station")
```



Updated template

```
1 ggplot(data = <DATA>) +  
2   <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>)) +  
3   <FACET_FUNCTION> +  
4   <THEME_FUNCTION>
```