Making Data Speak: Principles and Practice of Scientific Visualization

Part 1: Code-Along

Ronnie Bailey-Steinitz

2026-02-06

Some useful keyboard shortcuts: - New code chunk: Ctrl+Alt+i // Cmd+Option+i - Comment or uncomment: Ctrl+Shift+C // Cmd+Shift+C - Run current selected code line: Ctrl+Enter // Cmd+Return - Select Word: Ctrl+Shift+Left/Right // Option+Shift+Left/Right - Select to Line Start: Alt+Shift+Left // Cmd+Shift+Left - Select to Line End: Alt+Shift+Right // Cmd+Shift+Right

Attach Packages

```
# Core
library(ggplot2)
library(here)

# Data import + wrangling
library(readr)
library(dplyr)
library(tidyr)
library(stringr)
library(lubridate)
library(forcats)
library(janitor)

# Plot helpers
library(scales)
library(patchwork)
```

0. Read in Data

```
crossings <- readr::read_csv(here("SDSU Data Viz 2026/crossings.csv")) %>%
  janitor::clean_names()

penguins <- readr::read_csv(here("SDSU Data Viz 2026/penguins.csv")) %>%
  janitor::clean_names()
```

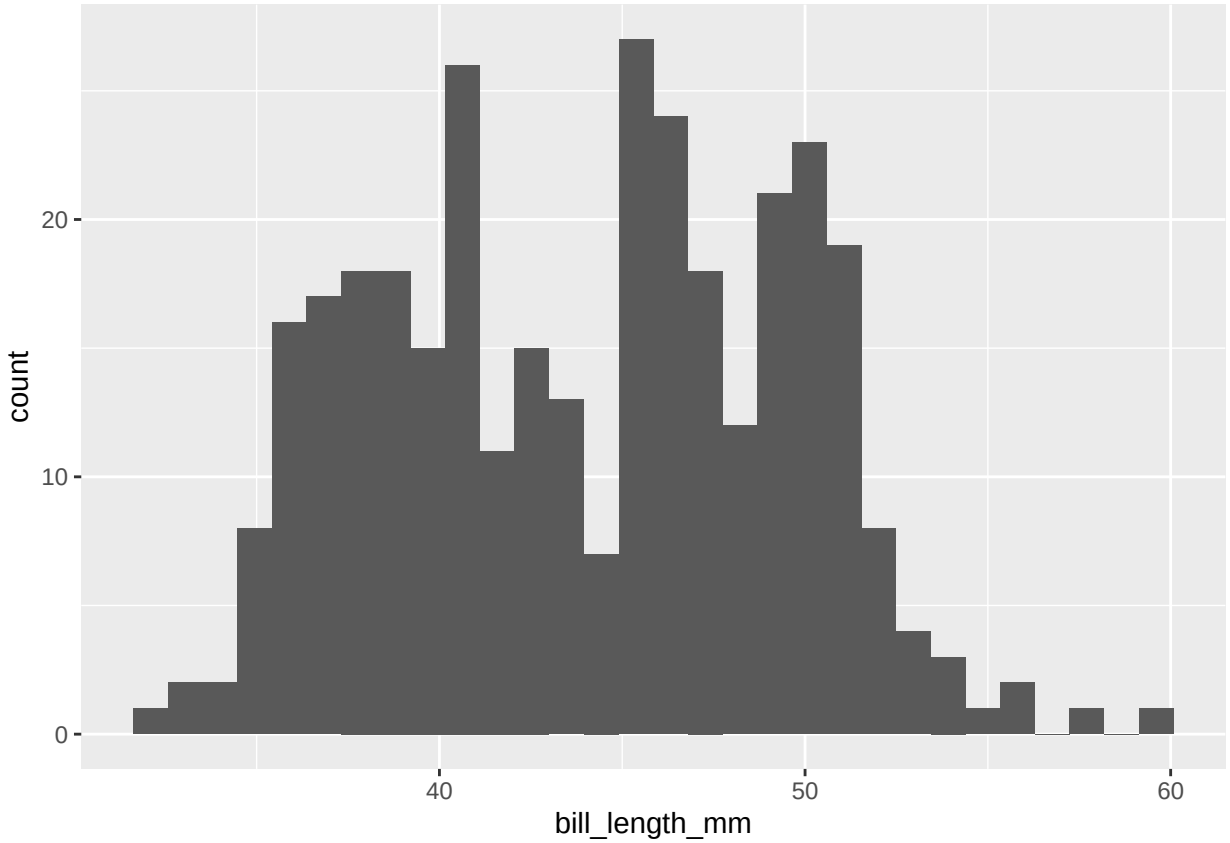
1. Look at data

1.1 summary stats

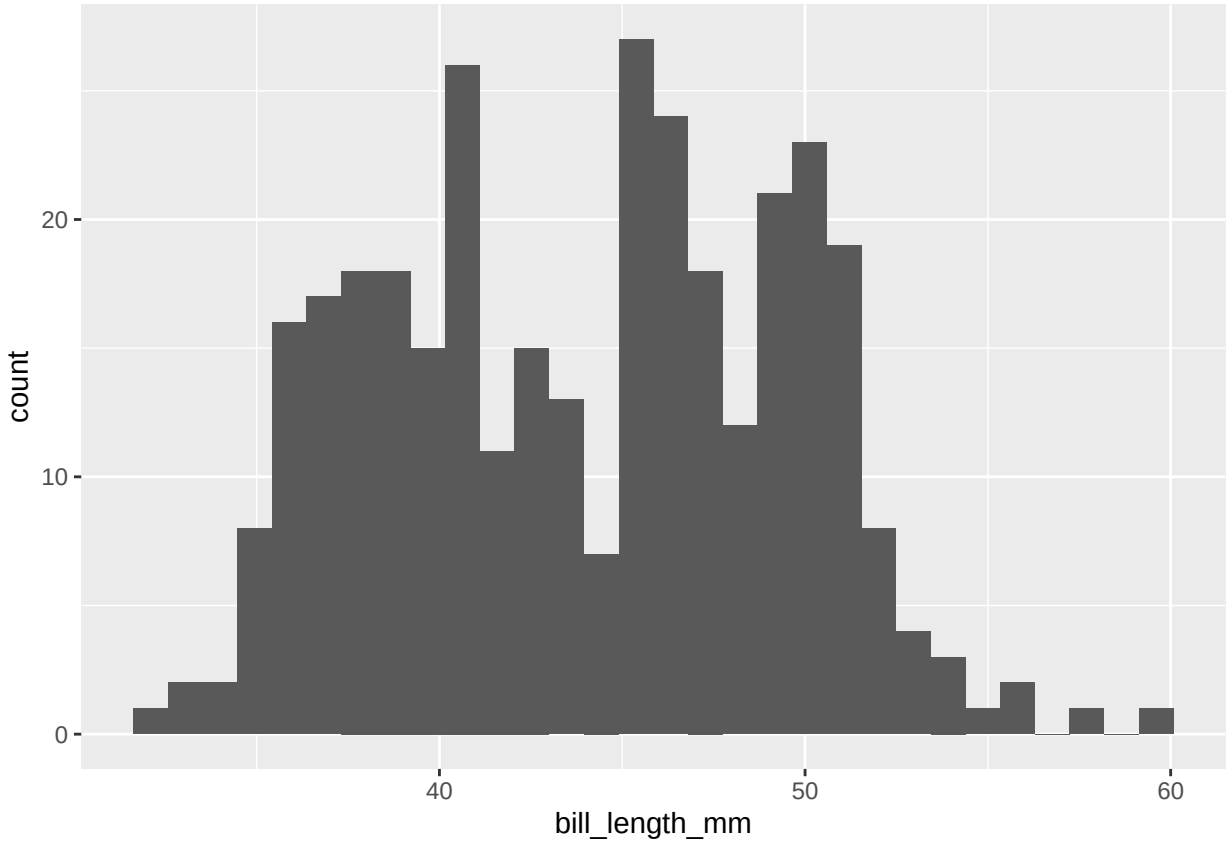
```
dplyr::glimpse(penguins)
```

```
## Rows: 333
## Columns: 20
## $ x1                <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ study_name        <chr> "PAL0708", "PAL0708", "PAL0708", "PAL0708", "PAL0708~
## $ sample_number     <dbl> 1, 2, 3, 5, 6, 7, 8, 13, 14, 15, 16, 17, 18, 19, 20,~
## $ species           <chr> "Adelie Penguin (Pygoscelis adeliae)", "Adelie Pengu~
## $ region            <chr> "Anvers", "Anvers", "Anvers", "Anvers", "Anvers", "A~
## $ island            <chr> "Torgersen", "Torgersen", "Torgersen", "Torgersen", ~
## $ stage             <chr> "Adult, 1 Egg Stage", "Adult, 1 Egg Stage", "Adult, ~
## $ individual_id     <chr> "N1A1", "N1A2", "N2A1", "N3A1", "N3A2", "N4A1", "N4A~
## $ clutch_completion <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "No", "No", "Yes"~
## $ date_egg          <chr> "11/11/07", "11/11/07", "11/16/07", "11/16/07", "11/~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, 36.7, 39.3, 38.9, 39.2, 41.1, 38.6~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, 19.3, 20.6, 17.8, 19.6, 17.6, 21.2~
## $ flipper_length_mm <dbl> 181, 186, 195, 193, 190, 181, 195, 182, 191, 198, 18~
## $ body_mass_g       <dbl> 3750, 3800, 3250, 3450, 3650, 3625, 4675, 3200, 3800~
## $ sex               <chr> "MALE", "FEMALE", "FEMALE", "FEMALE", "MALE", "FEMAL~
## $ delta_15_n_o_oo   <dbl> NA, 8.94956, 8.36821, 8.76651, 8.66496, 9.18718, 9.4~
## $ delta_13_c_o_oo   <dbl> NA, -24.69454, -25.33302, -25.32426, -25.29805, -25.~
## $ comments          <chr> "Not enough blood for isotopes.", NA, NA, NA, NA, "N~
## $ spp               <chr> "Adelie", "Adelie", "Adelie", "Adelie", "Adelie", "A~
## $ date_e            <date> 2007-11-11, 2007-11-11, 2007-11-16, 2007-11-16, 200~
```

```
# Every ggplot has 3 components: data, aes (aka mapping), geom:
ggplot(penguins, aes(x = bill_length_mm)) +
  geom_histogram()
```



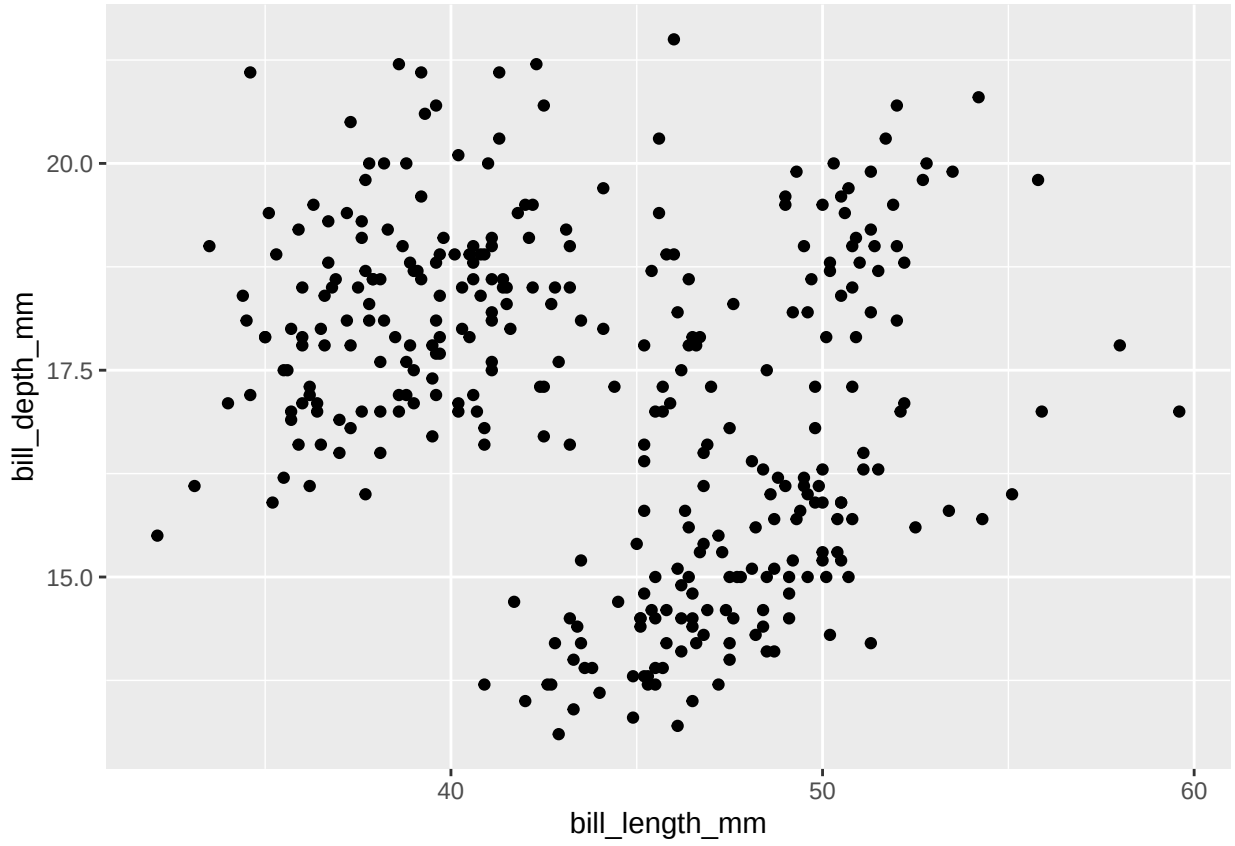

```
ggplot(penguins, aes(x = bill_length_mm)) +
  geom_histogram()
```



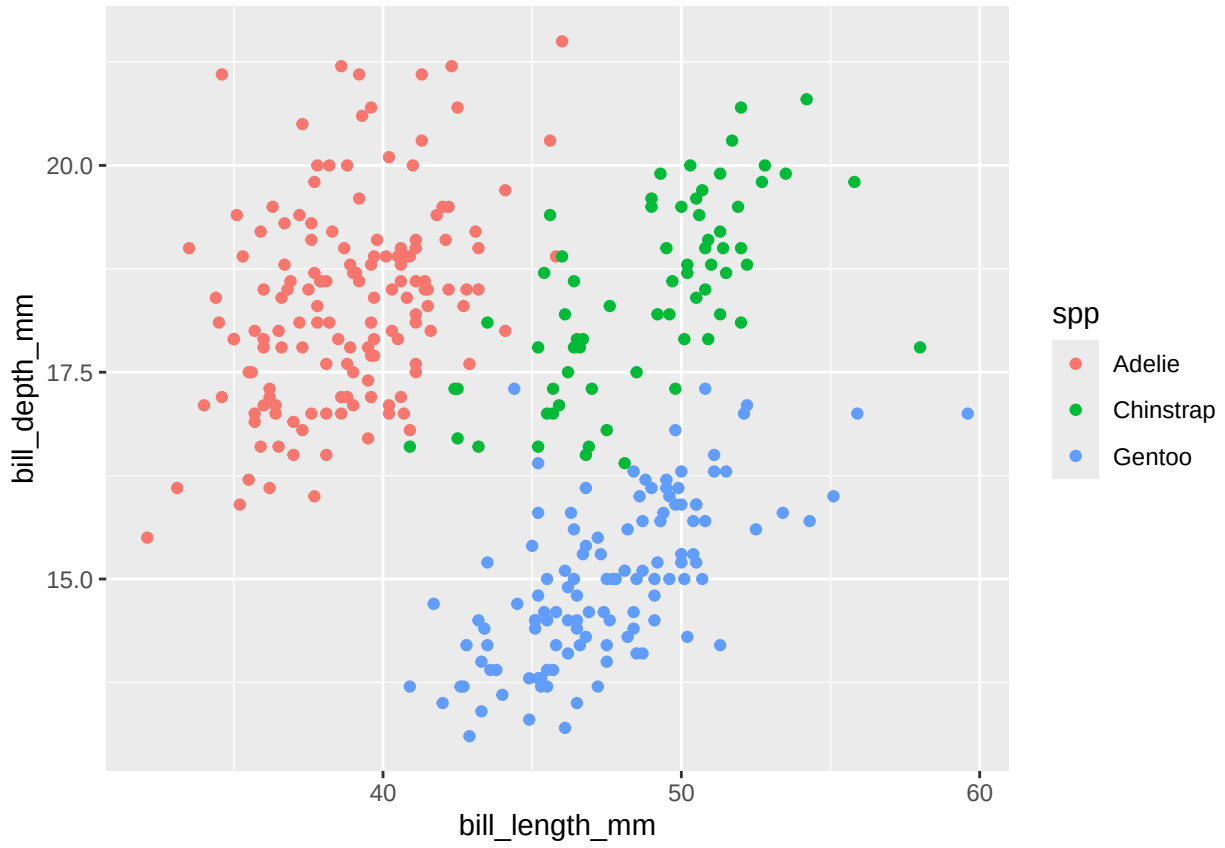
2. Plotting

2.1 Scatterplot

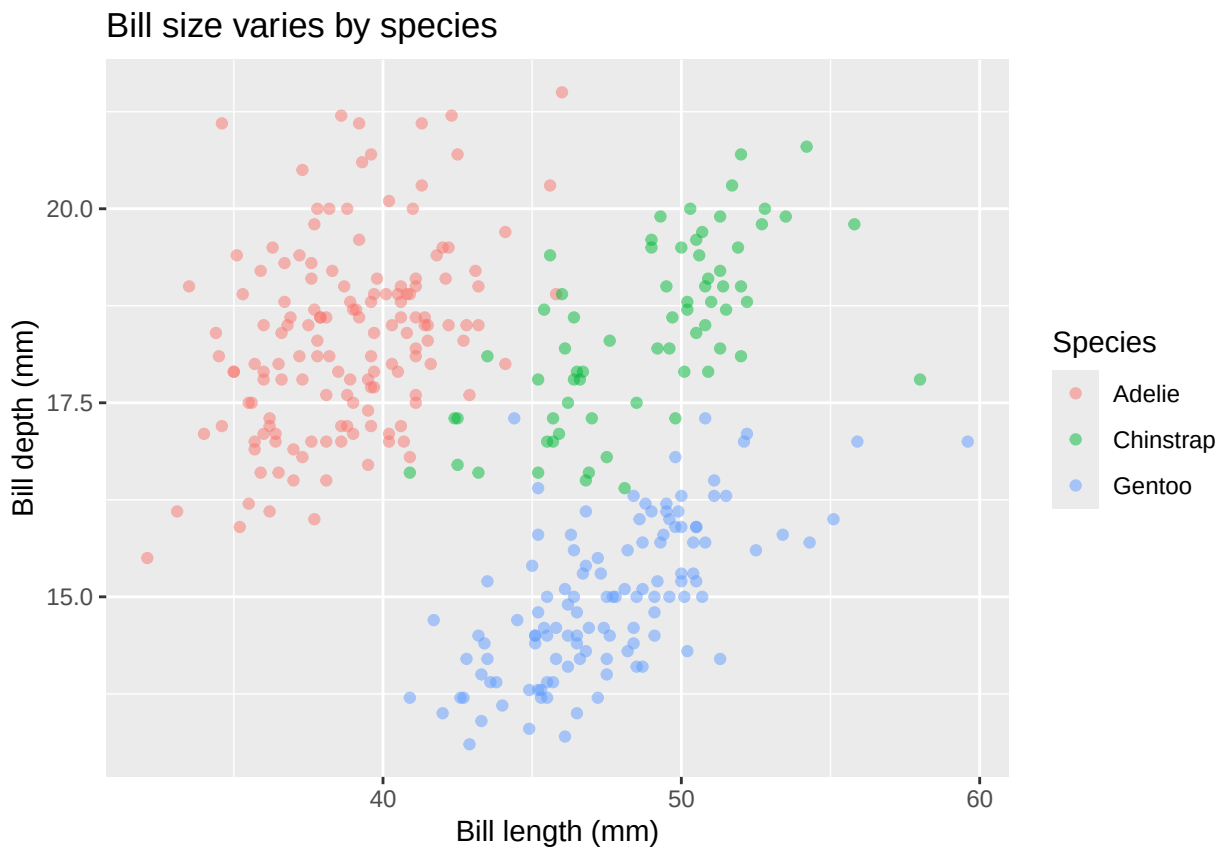
```
# BASE
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm)) +
  geom_point()
```



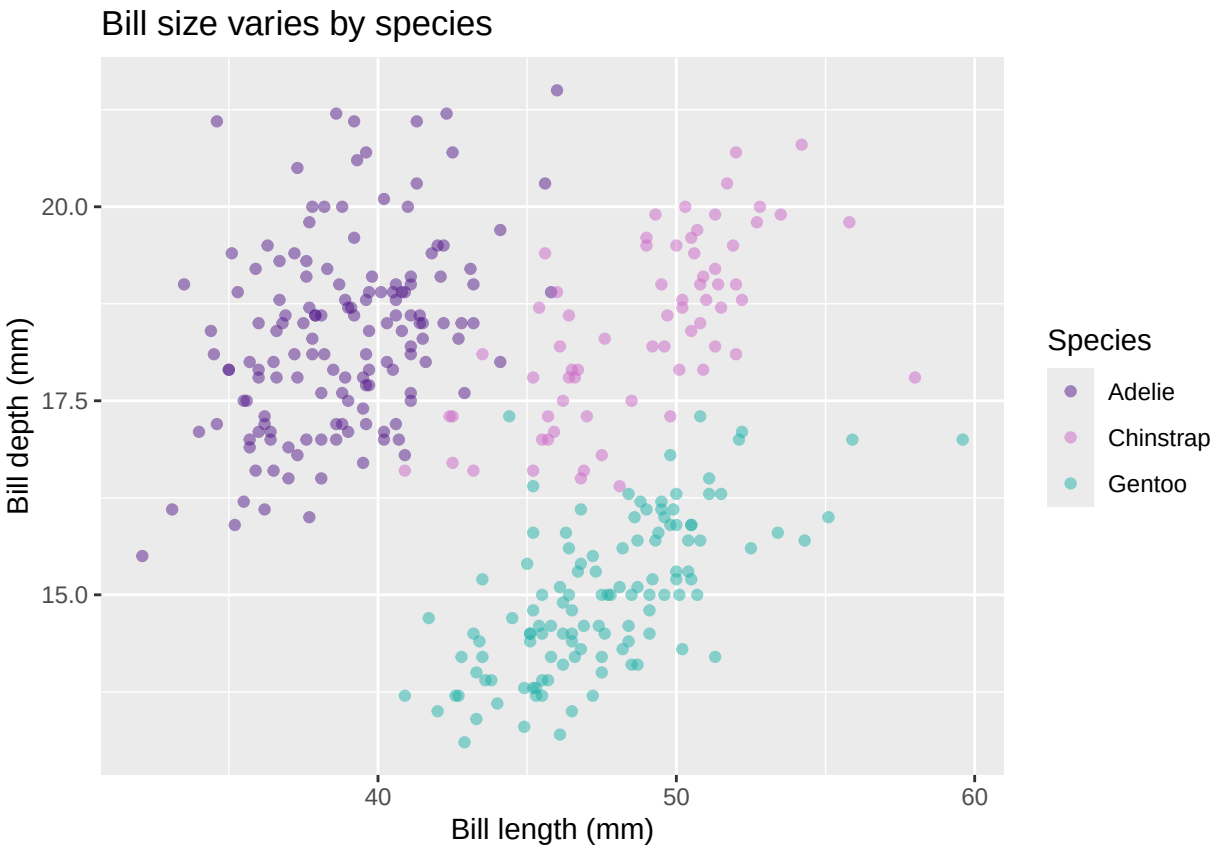

```
# Add color
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp)) +
  geom_point()
```




```
# # Move legend
# ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp)) +
#   geom_point() +
#   theme(legend.position = "bottom")

# Improve readability - alpha & labels & theme!
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Bill size varies by species",
    x = "Bill length (mm)",
    y = "Bill depth (mm)",
    color = "Species"
  )
```




```
# Custom colors
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Bill size varies by species",
    x = "Bill length (mm)",
    y = "Bill depth (mm)",
    color = "Species"
  ) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen"))
```

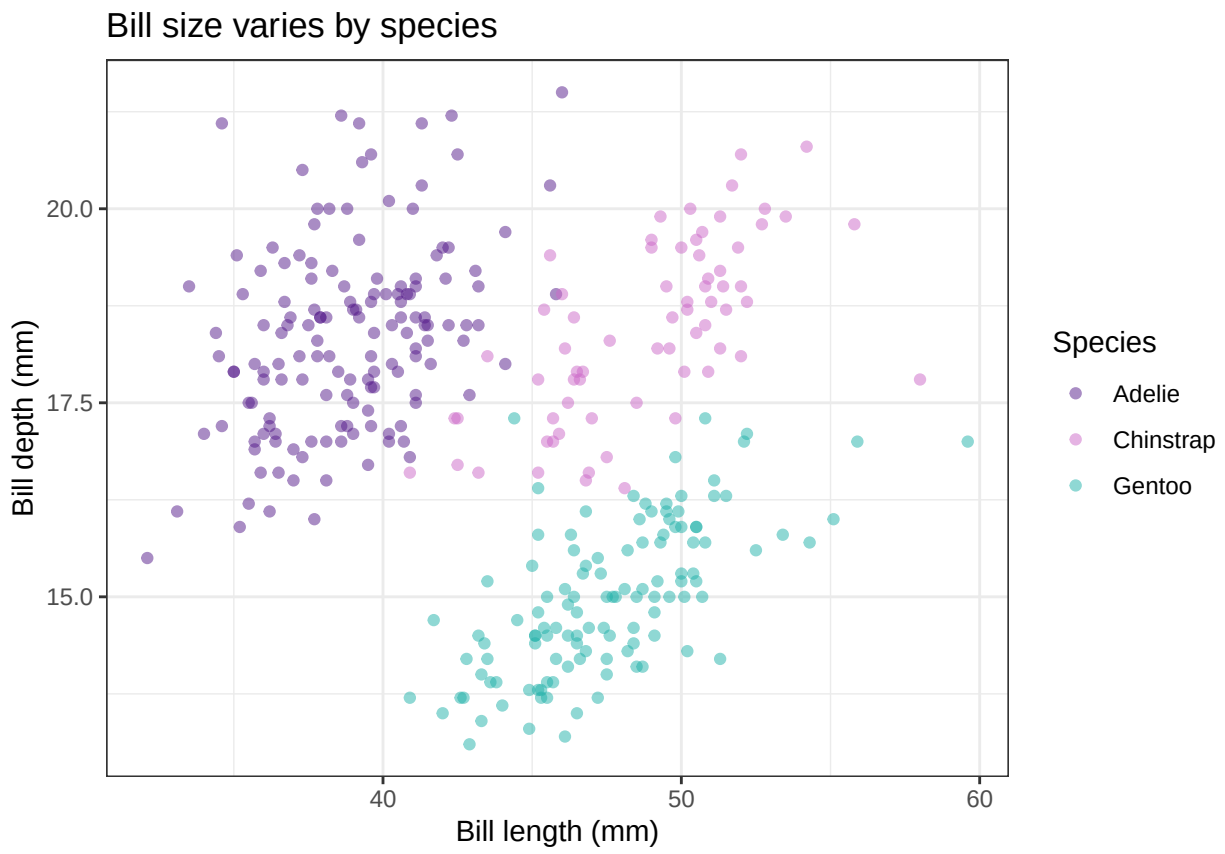


You can find all the keys and hex codes to innate R colors here: https://r-charts.com/colors/
. . . and more on how to use colors in ggplots here: https://r-graph-gallery.com/ggplot2-color.html

2.2 Adding themes

Themes are pre-defined sets of parameters that give ggplots specific types of appearances

```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Bill size varies by species",
    x = "Bill length (mm)",
    y = "Bill depth (mm)",
    color = "Species"
  ) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  theme_bw()
```



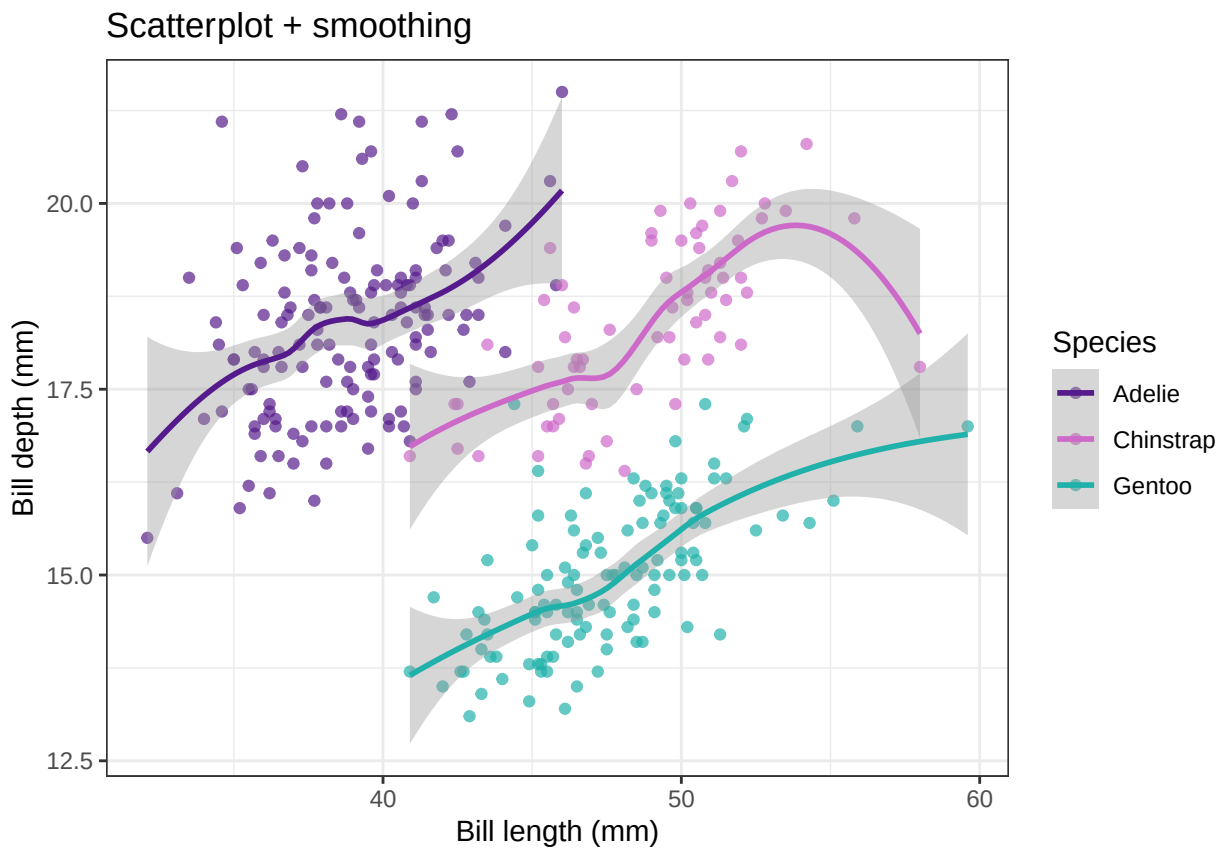

```
  # theme_light()
  # theme_dark()
  # theme_minimal()
  # theme_classic() # classic data plot ready for publication
  # theme_void() #excludes anything but the geom itself


# you can save the plot and define the dimensions!
ggsave("Bill size by species.png") # if you don't give dimensions, it'll appear like it does in your Pl
ggsave("Bill size by species 5x4.png", width = 5, height = 4) # but you can also define dimensions.
# do this to keep multiple plot dimensions consistent in your publication/poster!
```

More on themes here: https://ggplot2.tidyverse.org/reference/ggtheme.html

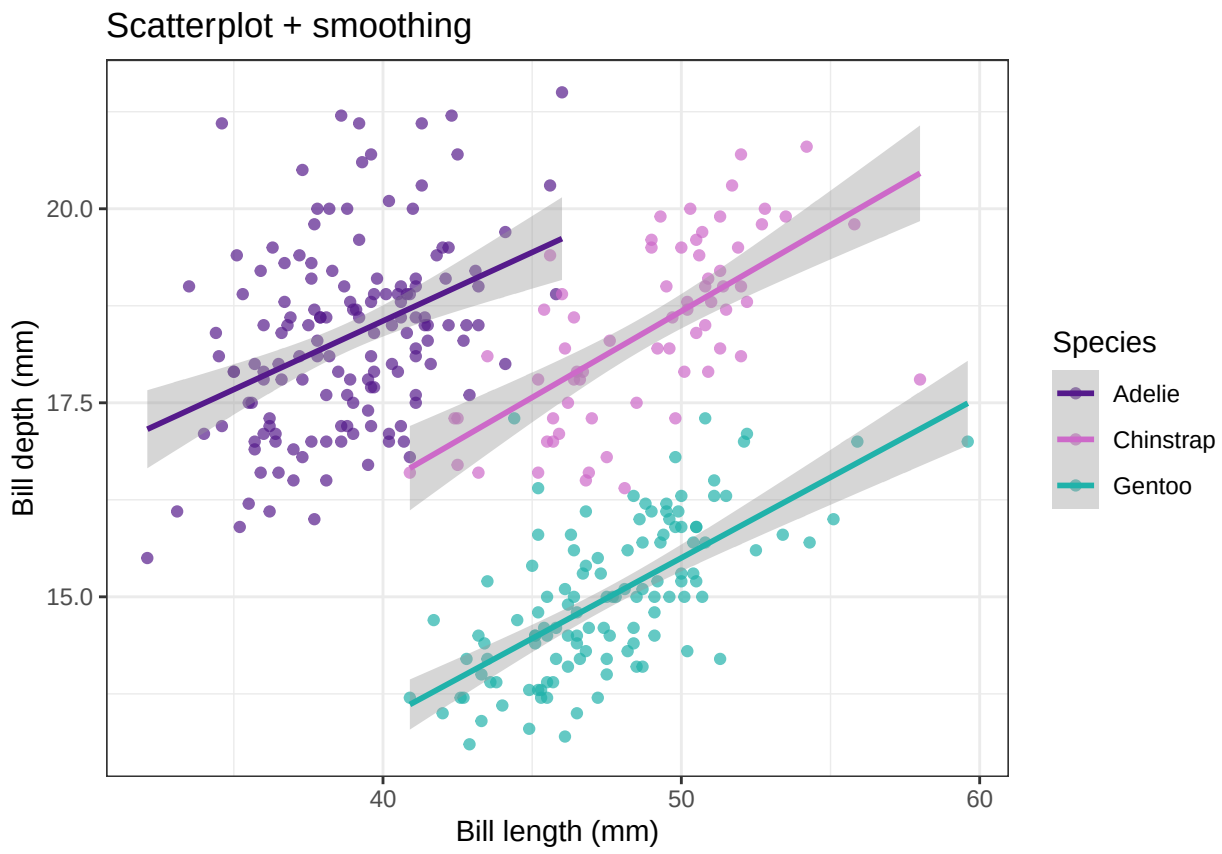
2.3 Add trendlines

```
# smooth (curved) lines
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp)) +
  geom_point(alpha = 0.7) +
  geom_smooth() +
  labs(
    title = "Scatterplot + smoothing",
    x = "Bill length (mm)",
```

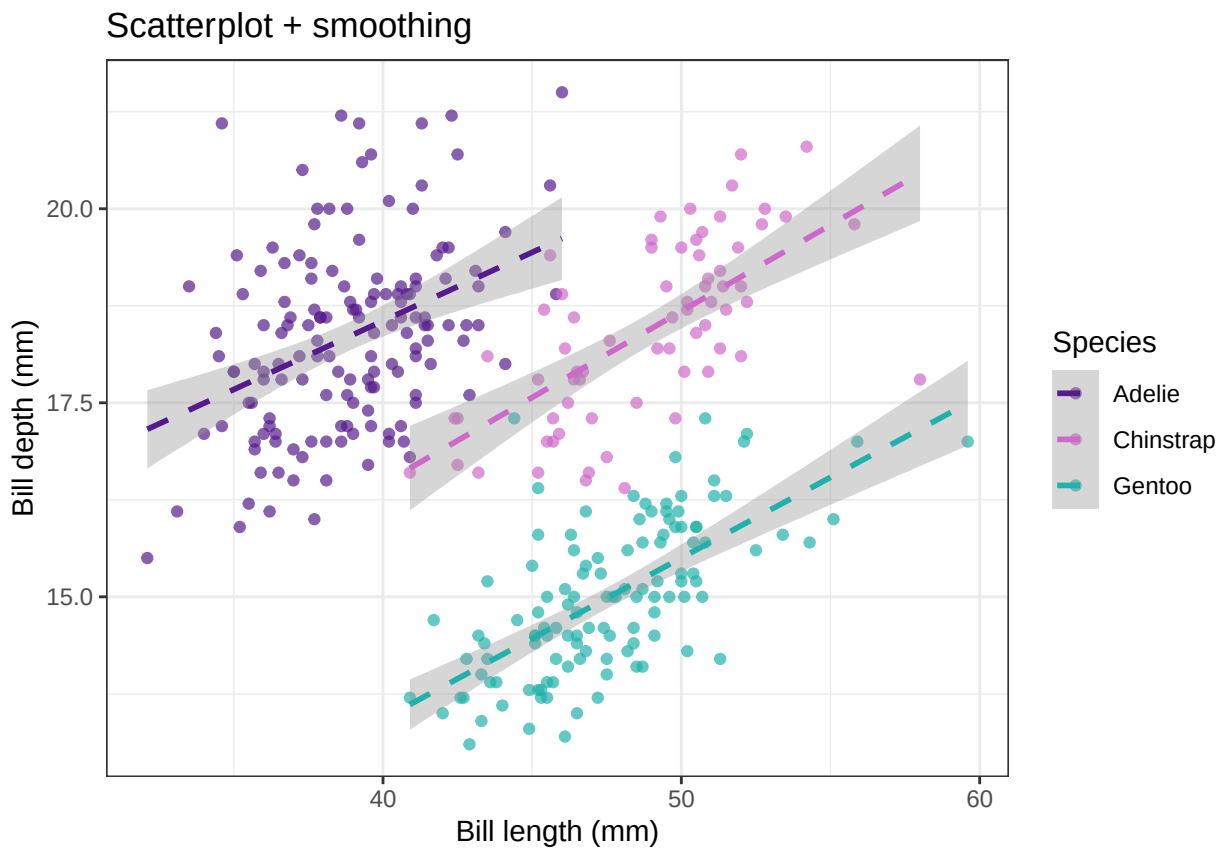
```
    y = "Bill depth (mm)",
    color = "Species"
  ) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  theme_bw()
```



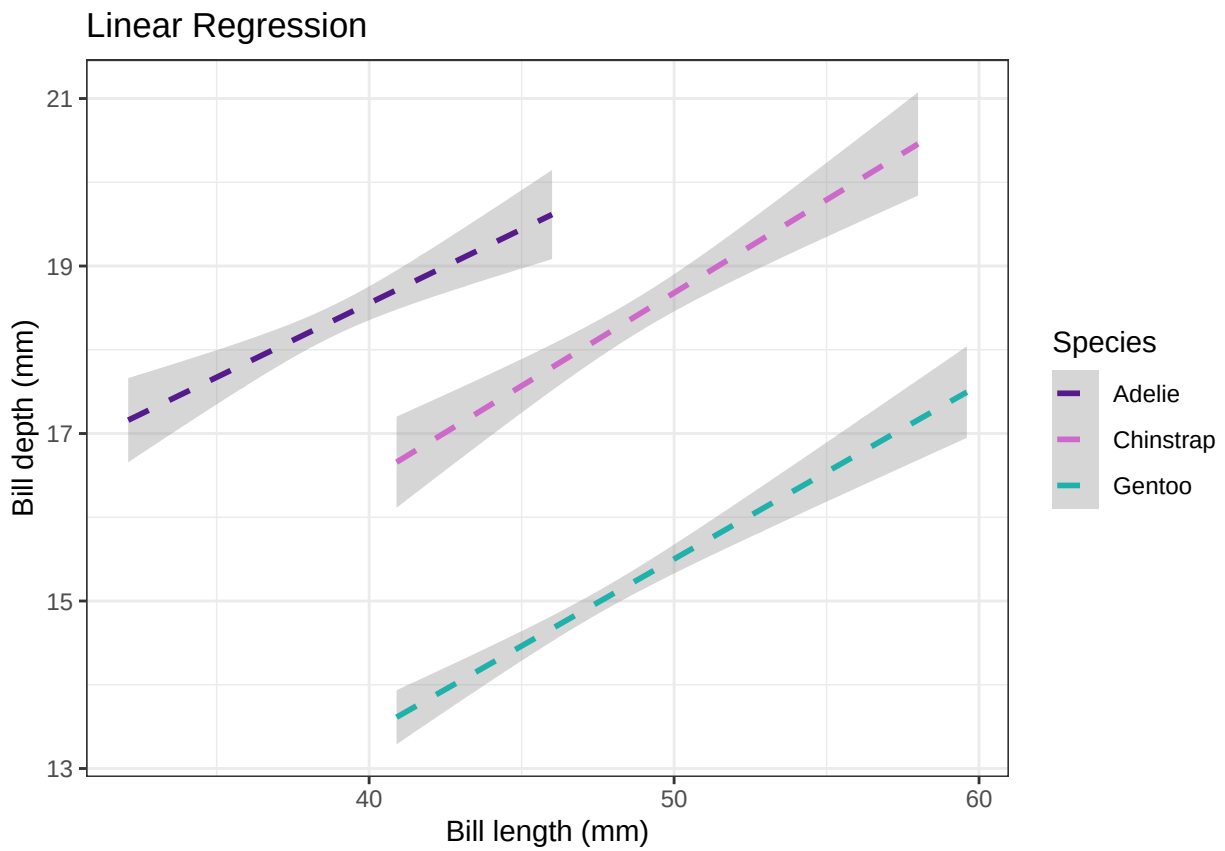

```
# make it linear!
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm") +
  labs(
    title = "Scatterplot + smoothing",
    x = "Bill length (mm)",
    y = "Bill depth (mm)",
    color = "Species"
  ) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  theme_bw()
```




```
# Modify the lines
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", linetype = "dashed") +
  labs(
    title = "Scatterplot + smoothing",
    x = "Bill length (mm)",
    y = "Bill depth (mm)",
    color = "Species"
  ) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  theme_bw()
```



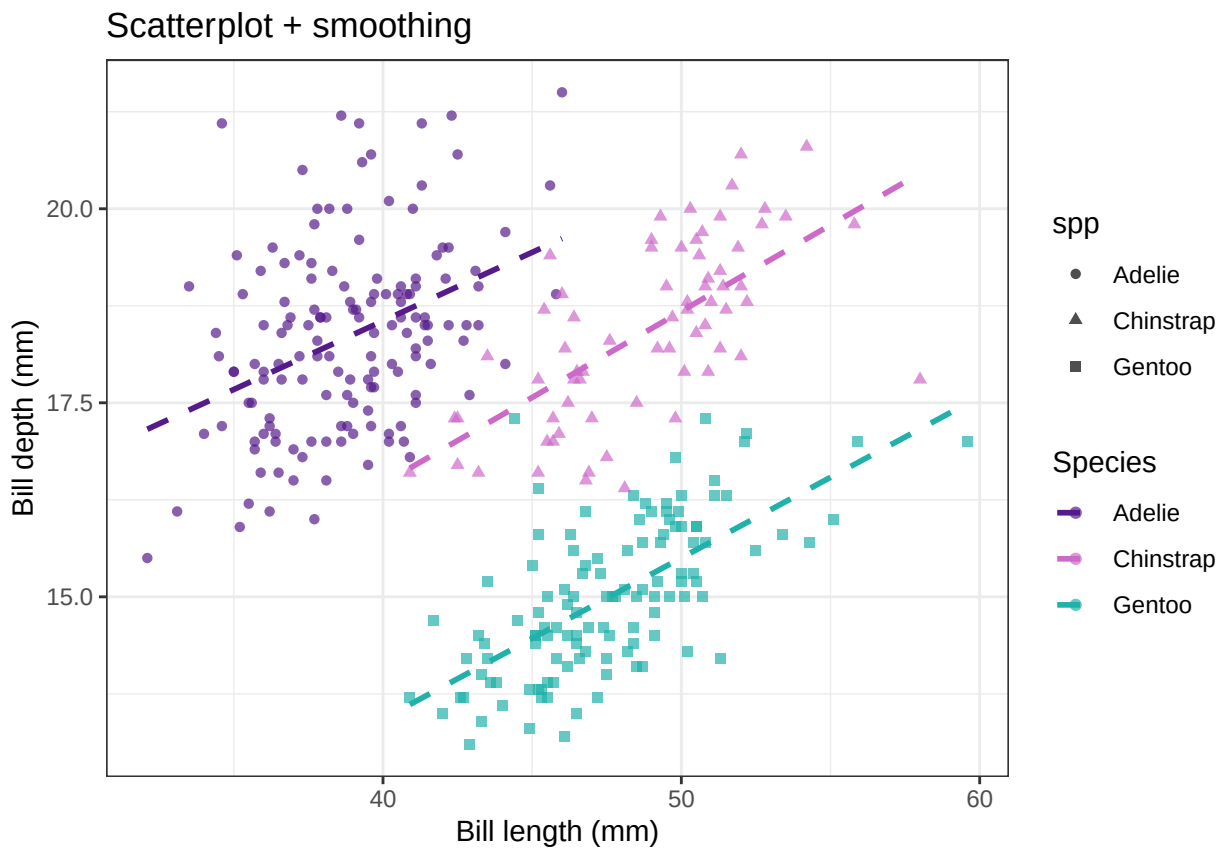

```
# or only have lines:
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp)) +
  geom_smooth(method = "lm", linetype = "dashed") +
  labs(
    title = "Linear Regression",
    x = "Bill length (mm)",
    y = "Bill depth (mm)",
    color = "Species"
  ) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  theme_bw()
```



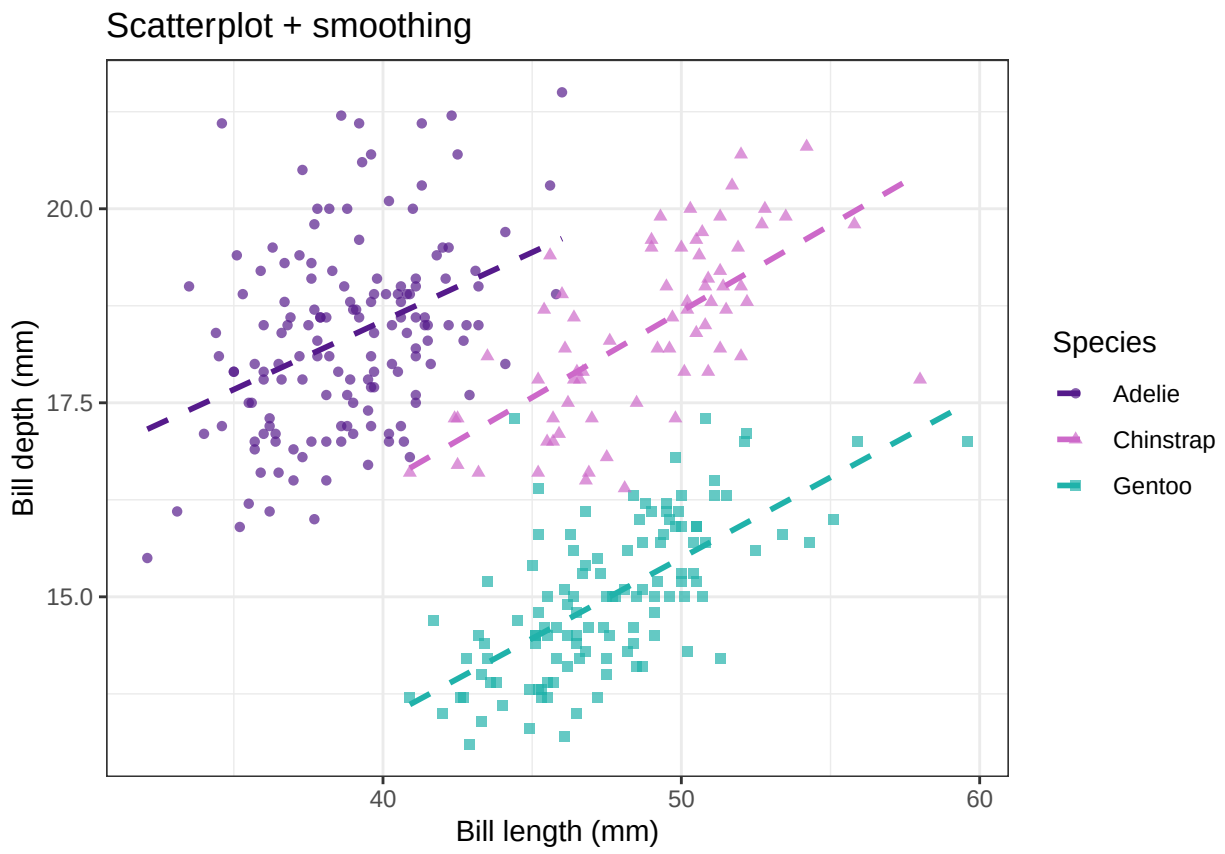
Other linetype options are: twodash; solid; longdash; dotted; dotdash; dashed.

2.4 Shapes

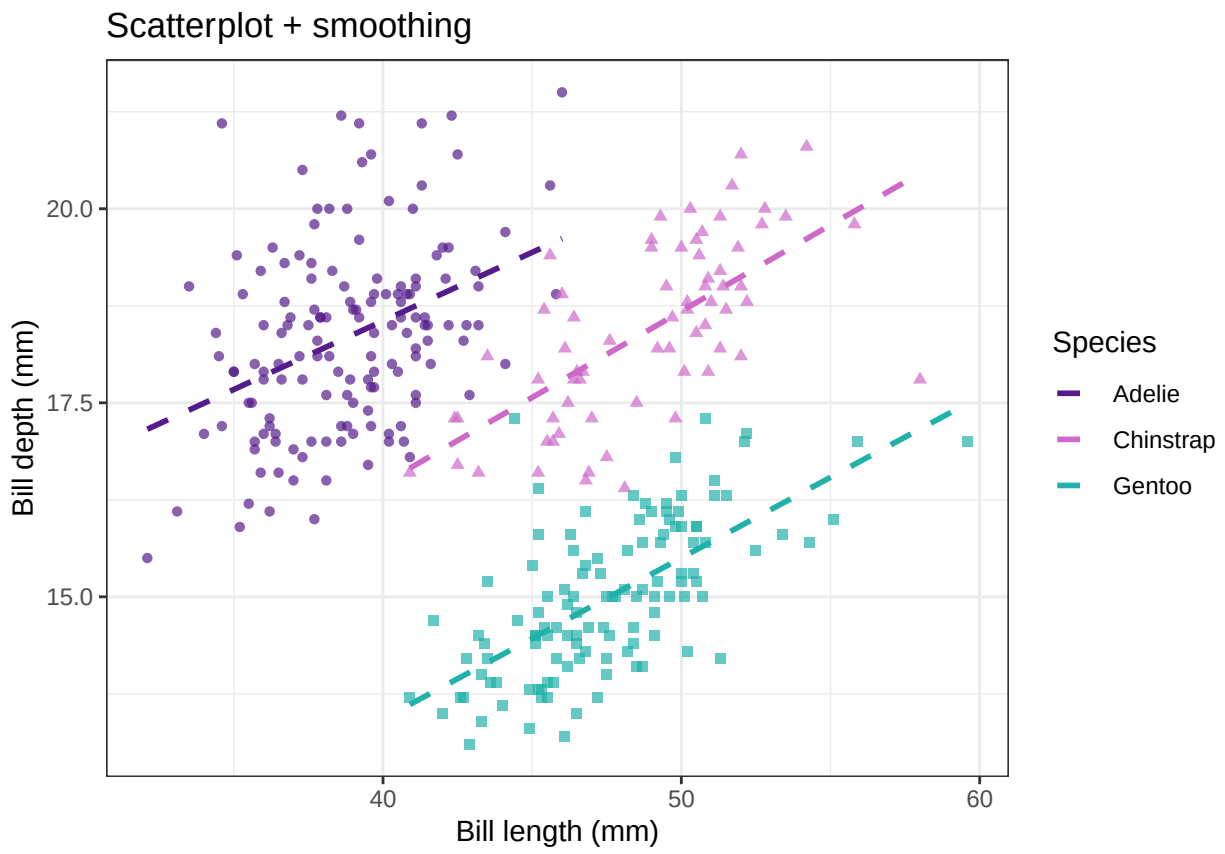
```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp, shape = spp)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, linetype = "dashed") +
  labs(
    title = "Scatterplot + smoothing",
    x = "Bill length (mm)",
    y = "Bill depth (mm)",
    color = "Species"
  ) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  theme_bw()
```




```
# combine the legends
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp, shape = spp)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, linetype = "dashed") +
  labs(
    title = "Scatterplot + smoothing",
    x = "Bill length (mm)",
    y = "Bill depth (mm)",
    color = "Species",
    shape = "Species" # <----- note that we tell ggplot that the legend for the shape has the same name
  ) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  theme_bw()
```




```
# ...or...
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, color = spp, shape = spp)) +
  geom_point(alpha = 0.7, show.legend = FALSE) + # < ---- just tell geom_point NOT to plot legend at al
  geom_smooth(method = "lm", se = FALSE, linetype = "dashed") +
  labs(
    title = "Scatterplot + smoothing",
    x = "Bill length (mm)",
    y = "Bill depth (mm)",
    color = "Species"
    # no "shape" in labels section
  ) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  theme_bw()
```

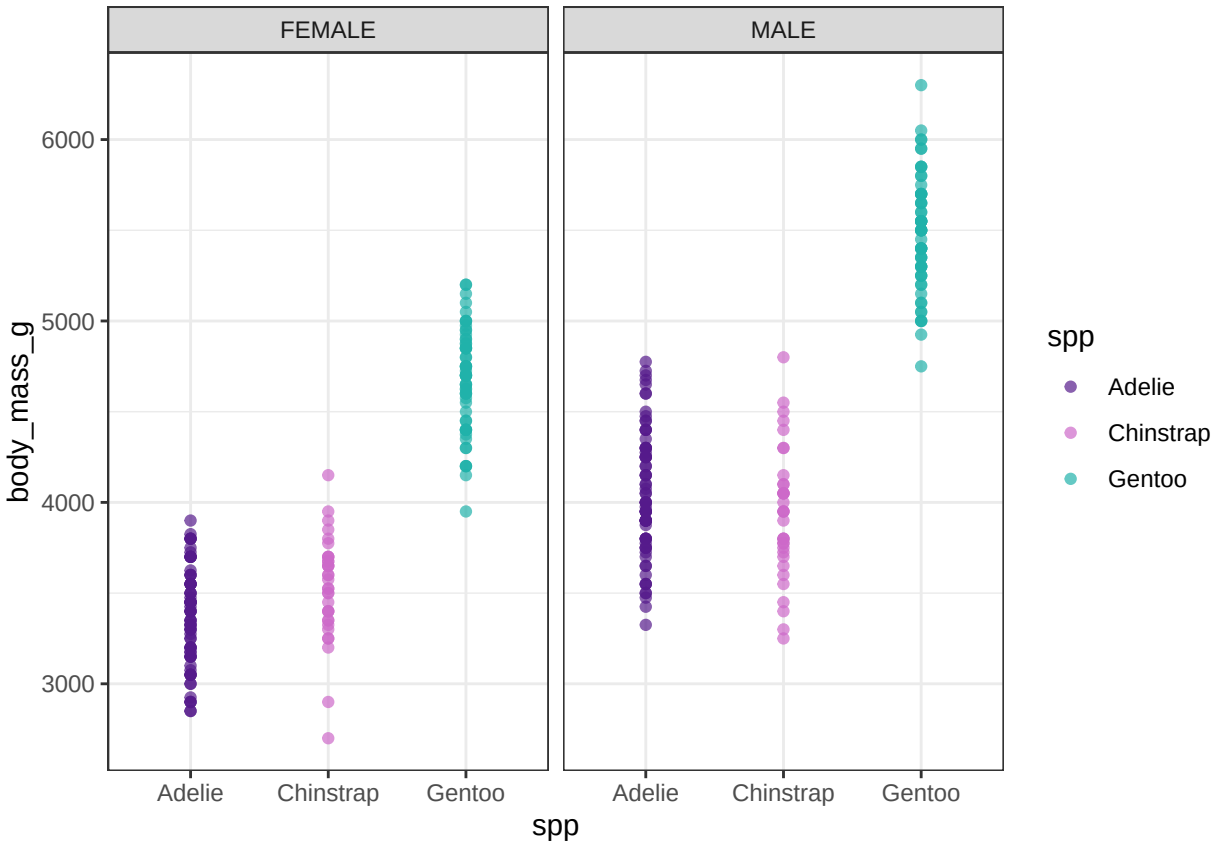



```
#  Now ggplot knows color and shape belong to the same variable. It automatically merges them into a si
```

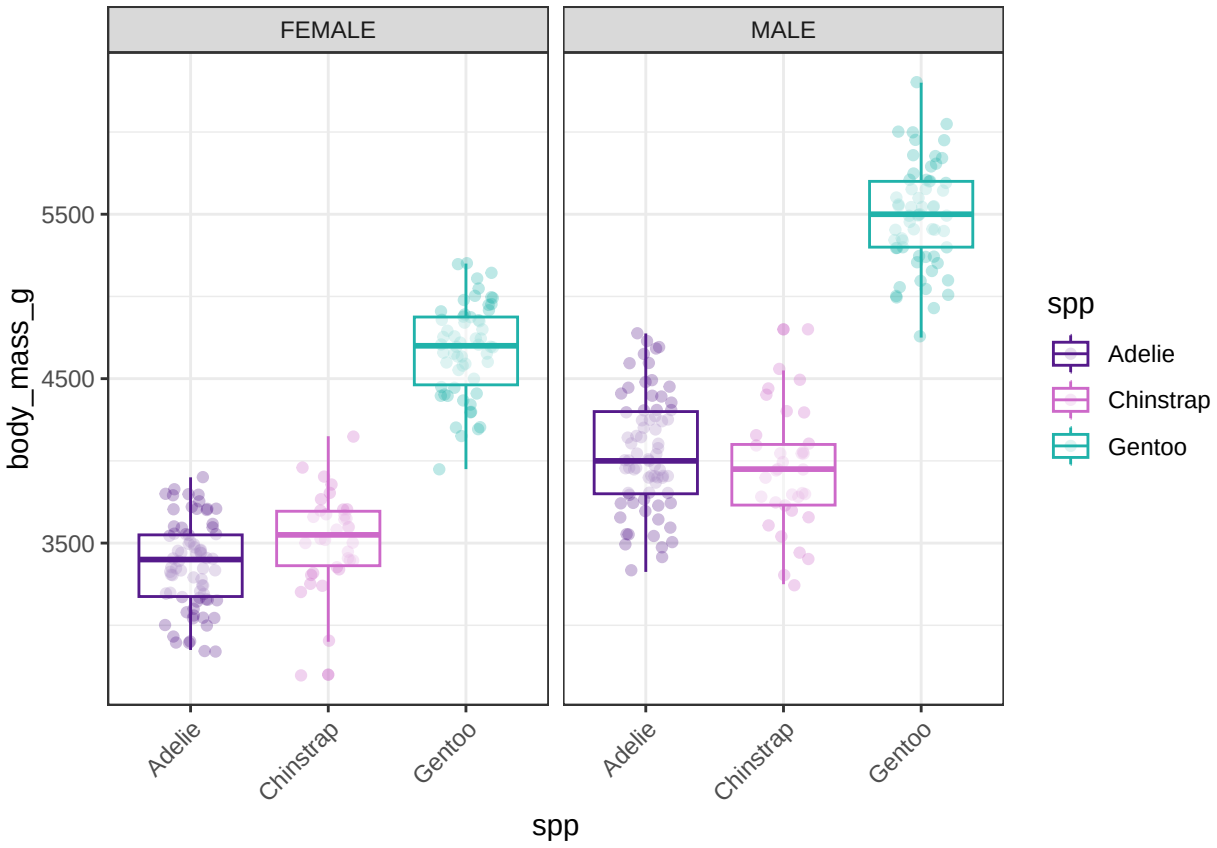
2.5 Facetting

Also, boxplots and axis text

```
# you can divide by a categorical variable
ggplot(penguins, aes(x = spp, y = body_mass_g, color = spp)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~ sex) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  theme_bw()
```




```
# You can use this to show relationships between three variables: species, sex, body mass

# Boxplots might be a better geom here:
ggplot(penguins, aes(x = spp, y = body_mass_g, color = spp)) +
  geom_jitter(alpha = 0.3, width = 0.2) +
  geom_boxplot(alpha = 0.5) +
  facet_wrap(~ sex) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```



To use `vjust` and `hjust` for axis text in ggplot2, modify the `theme()` function with `axis.text.x` or `axis.text.y` and the `element_text()` function. The values for `vjust` (vertical justification) and hjust (horizontal justification) range from 0 to 1.

Next, we'll look at how to manage your viewer's attention, and in the process, we'll see new types of charts.

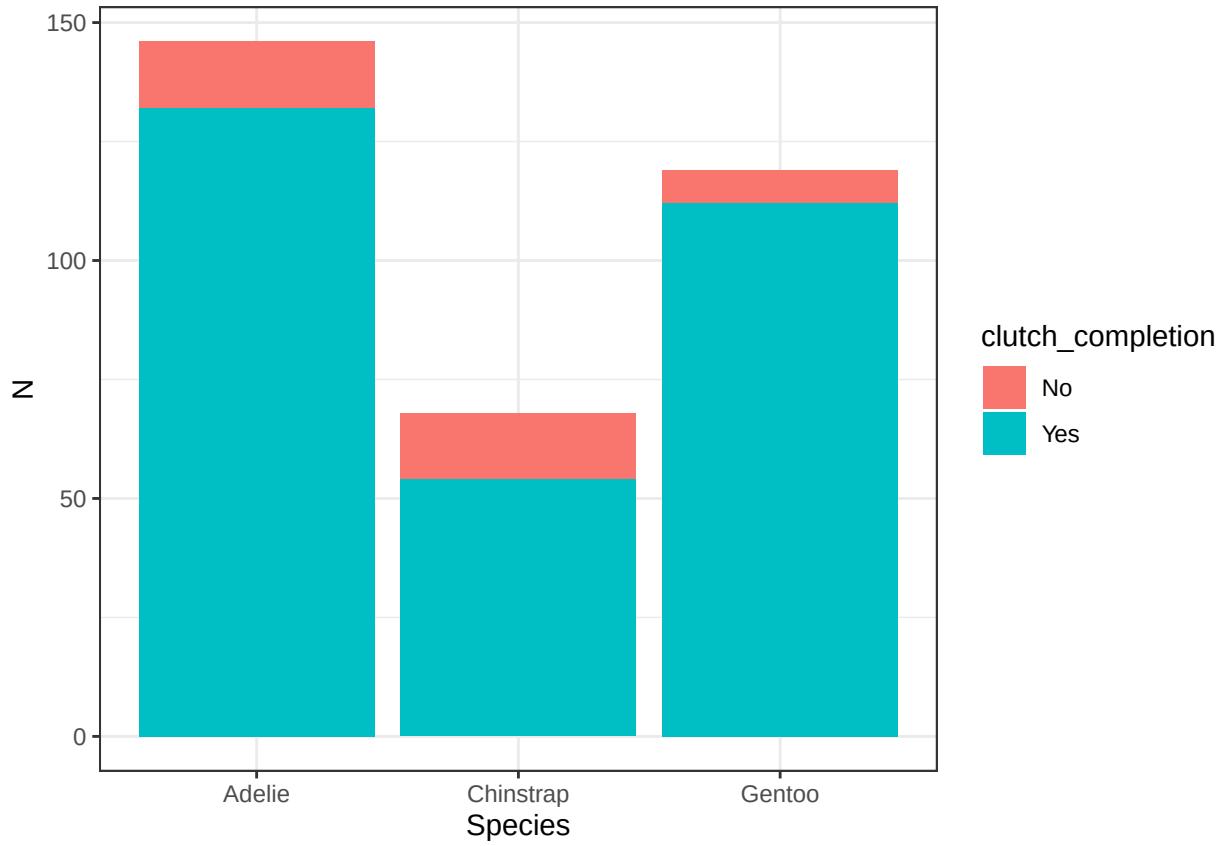
3. Storytelling: Managing the viewer's attention

3.1 Using colors

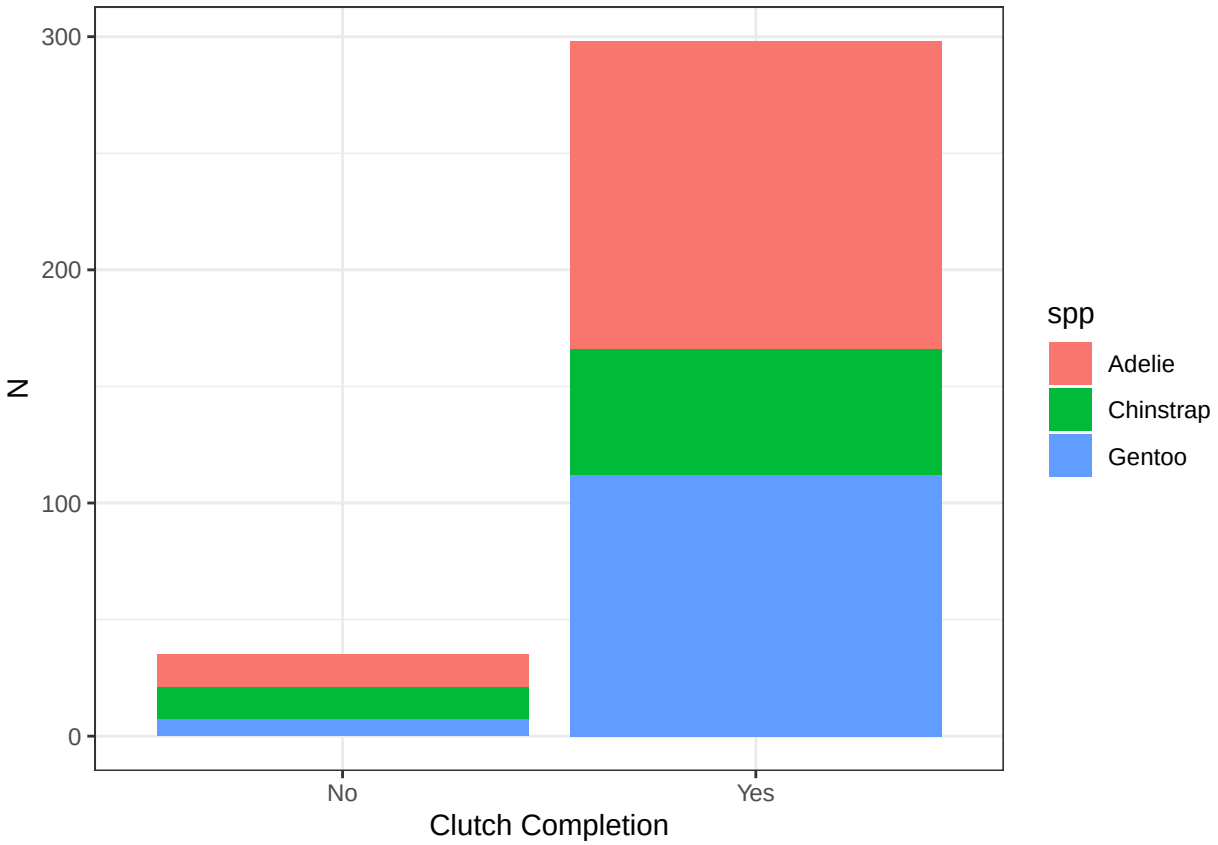
```
# let's look at some other aspects of the data
penguins %>% dplyr::count(clutch_completion, spp)
```

```
## # A tibble: 6 x 3
##   clutch_completion spp           n
##   <chr>             <chr>     <int>
## 1 No                Adelie       14
## 2 No                Chinstrap    14
## 3 No                Gentoo        7
## 4 Yes               Adelie      132
## 5 Yes               Chinstrap    54
## 6 Yes               Gentoo      112
```

```r
# basic stacked bar chart
ggplot(penguins, aes(x = spp, fill = clutch_completion)) +
  geom_bar() +

  # scale_fill_manual(values = c("green4", "firebrick3")) +
      # but notice the colors, what are they telling you???

  # scale_fill_manual(values = c("firebrick", "green4")) +
      # this makes more sense-- play on peoples' familiarity with color-coding!!!

  labs(x = "Species", y = "N") +
  theme_bw()
```

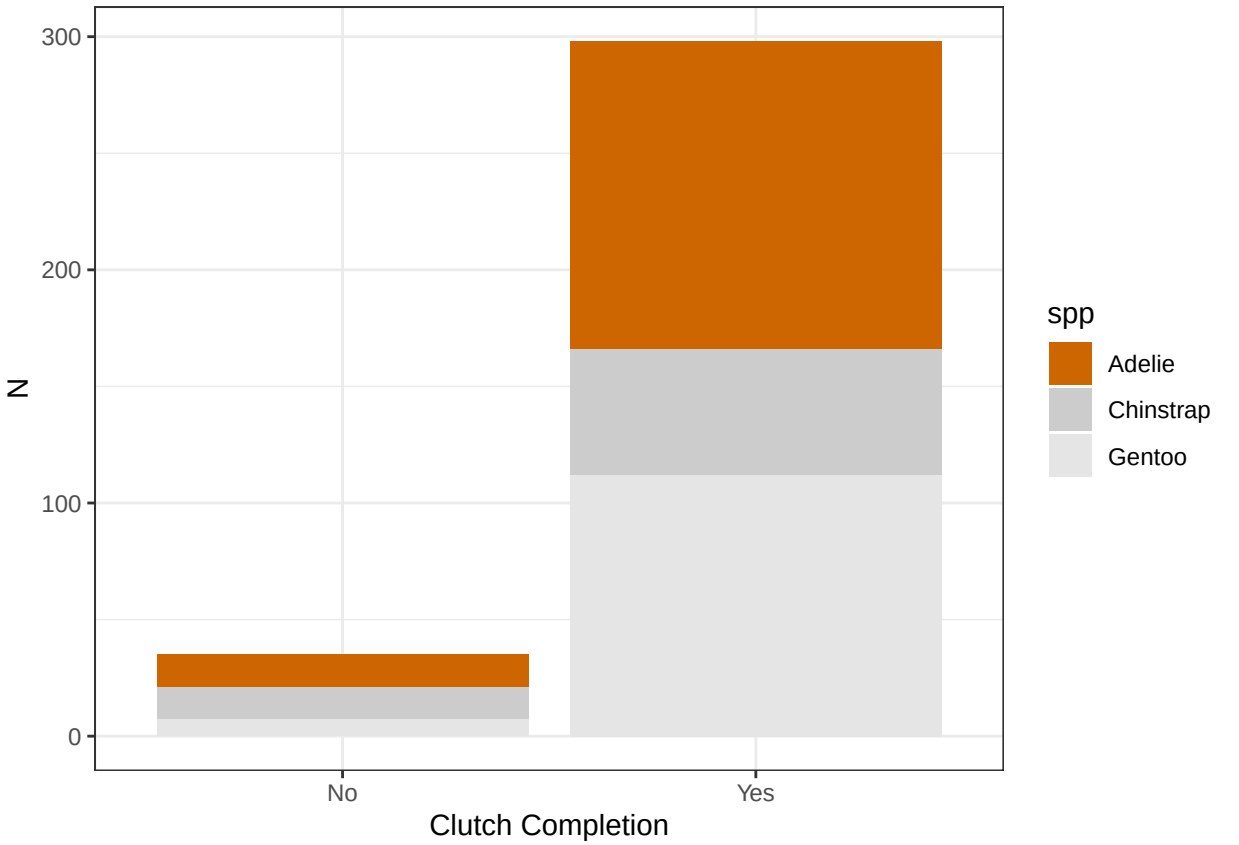



```r
# you can change the focus of the chart (swap color vs. x-axis)
ggplot(penguins, aes(x = clutch_completion, fill = spp)) +
  geom_bar() +
  #scale_fill_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  labs(x = "Clutch Completion", y = "N") +
  theme_bw()
```

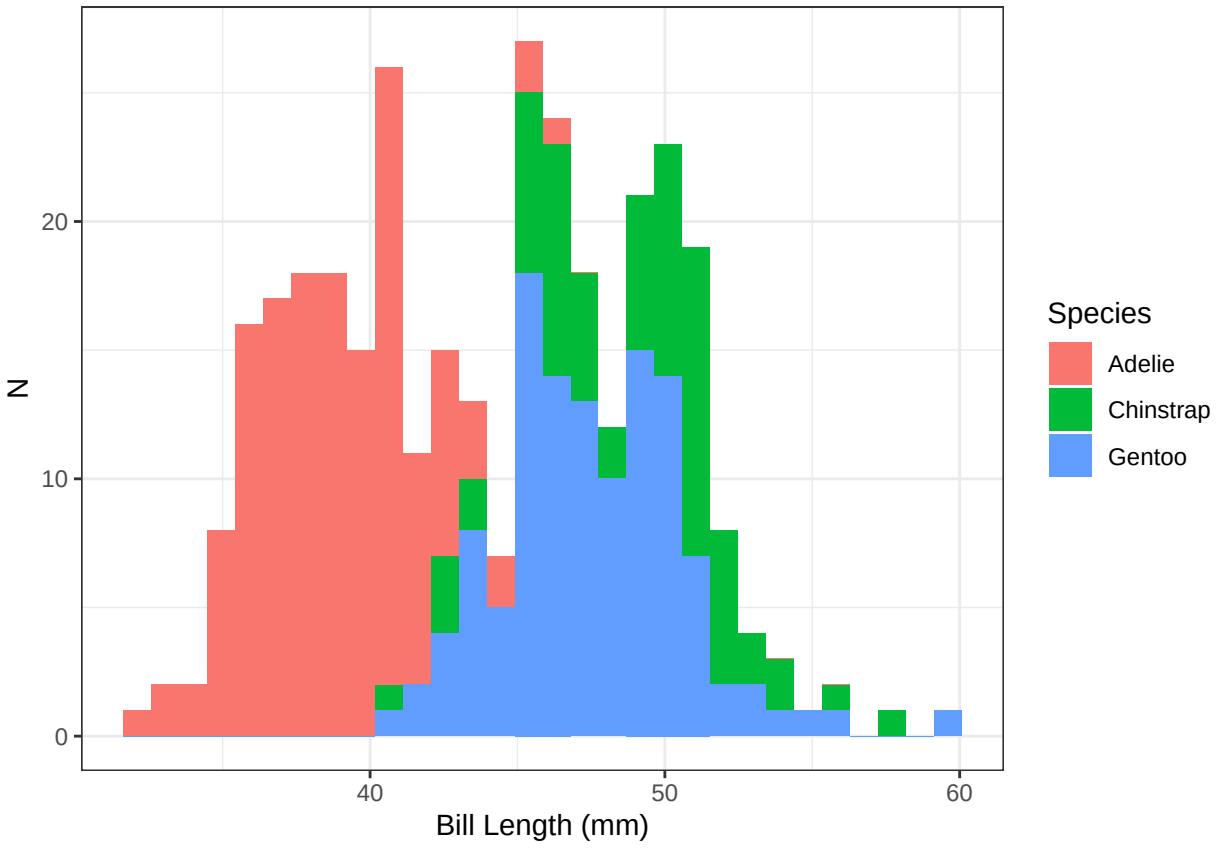



```
# --> we simply switched which variable was on the x-axis, and while is the color fill, but it makes a
```

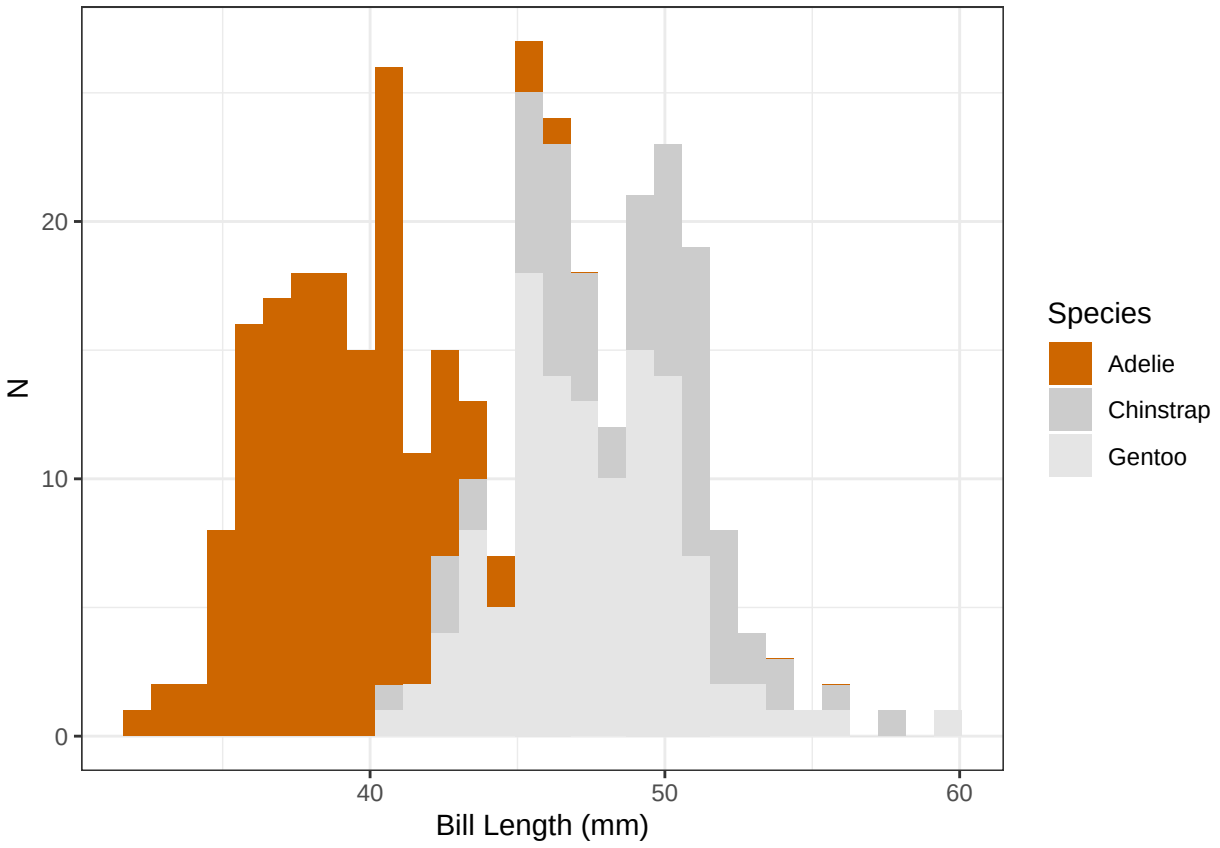
```
# To draw the viewer's attention, you can also highlight one particular aspect using colors
ggplot(penguins, aes(x = clutch_completion, fill = spp)) +
  geom_bar() +
  scale_fill_manual(values = c("darkorange3", "grey80", "grey90")) +
  labs(x = "Clutch Completion", y = "N") +
  theme_bw()
```



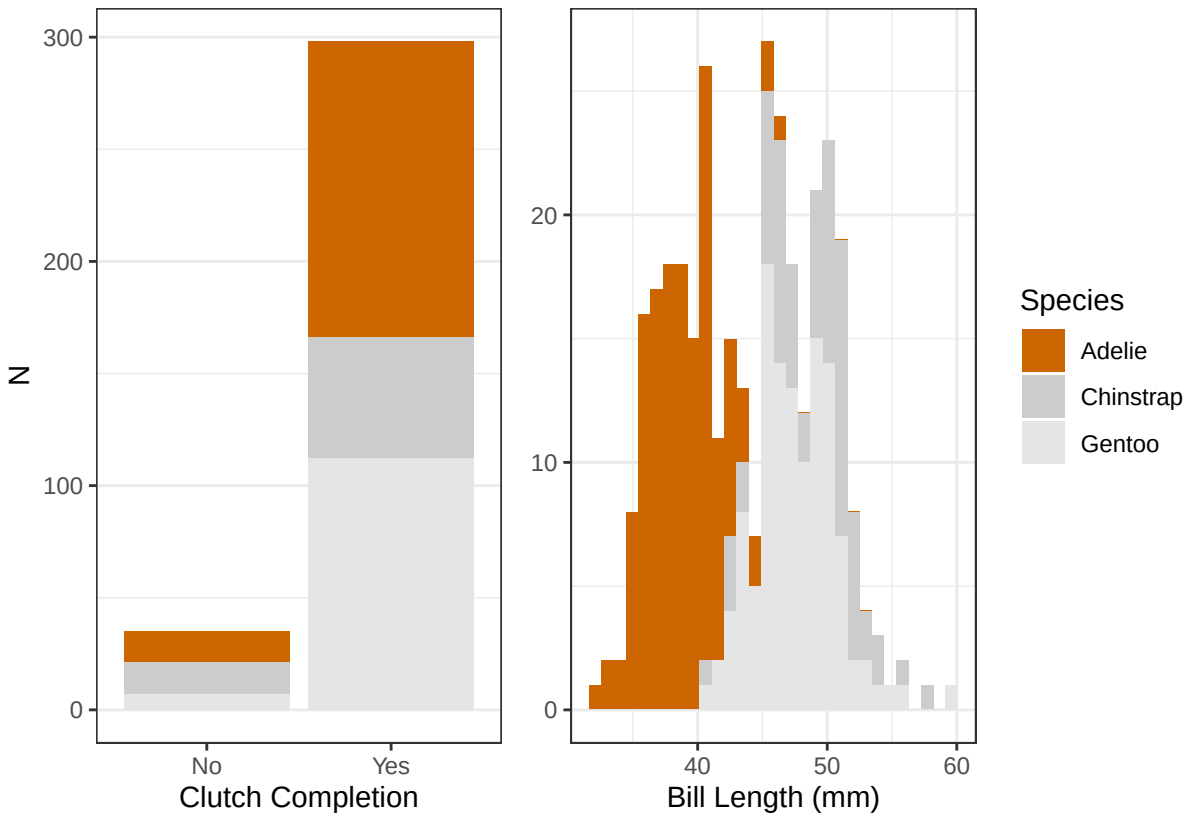

```
#########
# ADD N = numbers to bars???
#########

# you can also go back to you summary stats (the histogram):
ggplot(penguins, aes(x = bill_length_mm, fill = spp)) +
  geom_histogram() +
  labs(x = "Bill Length (mm)", y = "N", fill = "Species") +
  theme_bw()
```




```
ggplot(penguins, aes(x = bill_length_mm, fill = spp)) +
  geom_histogram() +
  scale_fill_manual(values = c("darkorange3", "grey80", "grey90")) +
  labs(x = "Bill Length (mm)", y = "N", fill = "Species") +
  theme_bw()
```

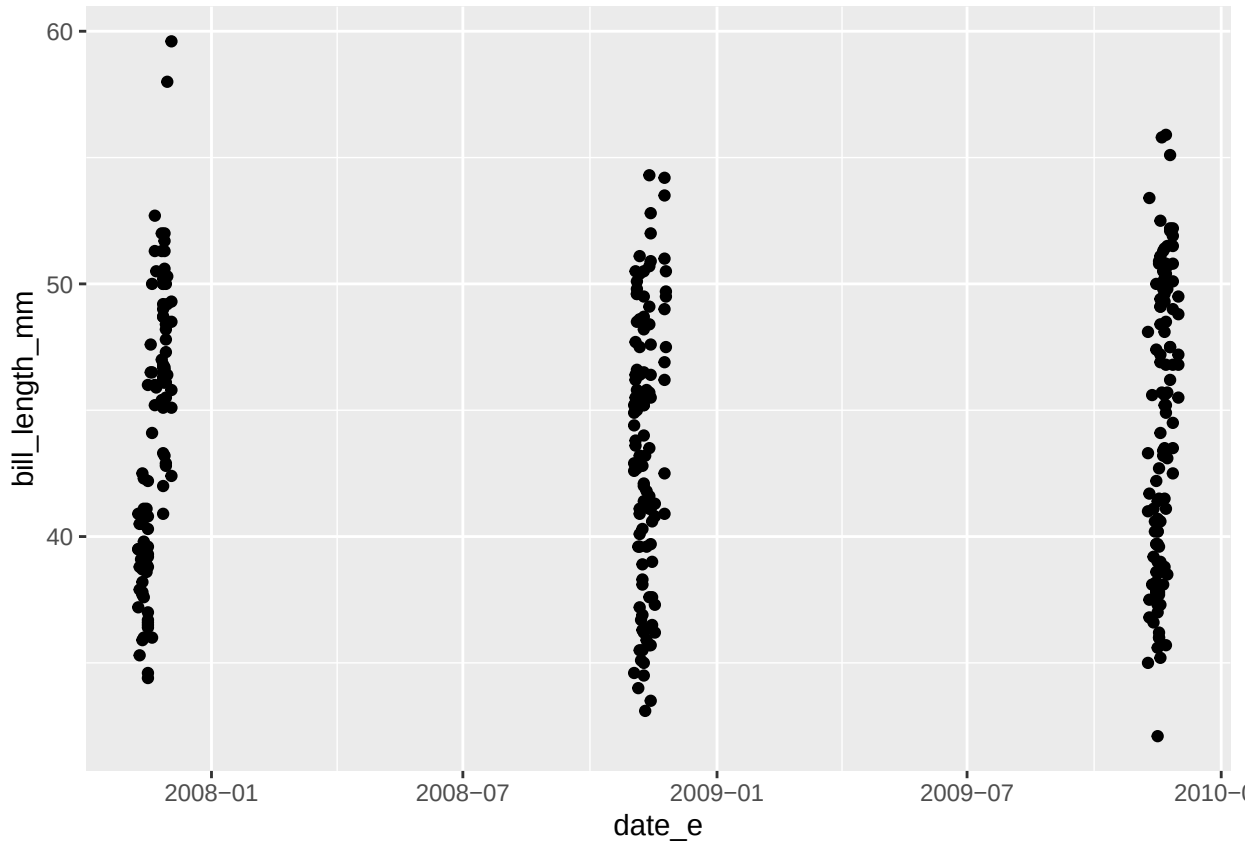



```
# ======================================================

plot1 <- ggplot(penguins, aes(x = clutch_completion, fill = spp)) +
  geom_bar() +
  scale_fill_manual(values = c("darkorange3", "grey80", "grey90")) +
  labs(x = "Clutch Completion", y = "N", fill = "Species") +
  theme_bw() +
  theme(panel.grid.major.x = element_blank())

plot2 <- ggplot(penguins, aes(x = bill_length_mm, fill = spp)) +
  geom_histogram() +
  scale_fill_manual(values = c("darkorange3", "grey80", "grey90")) +
  labs(x = "Bill Length (mm)", y = "N", fill = "Species") +
  theme_bw()

# this is part of the package {patchwork}
plot1 + plot2 + plot_layout(axes = "collect", guides = "collect")
```

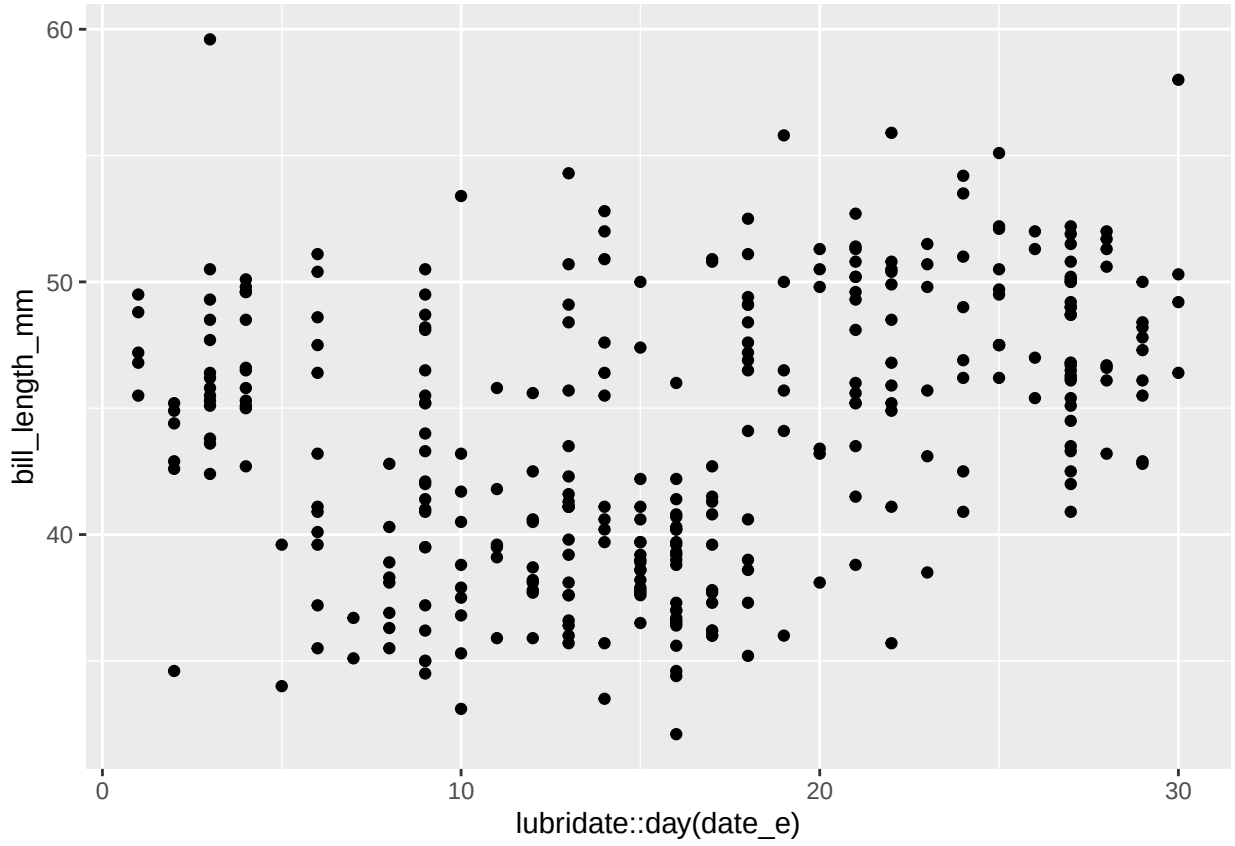


3.2 Using visual cues

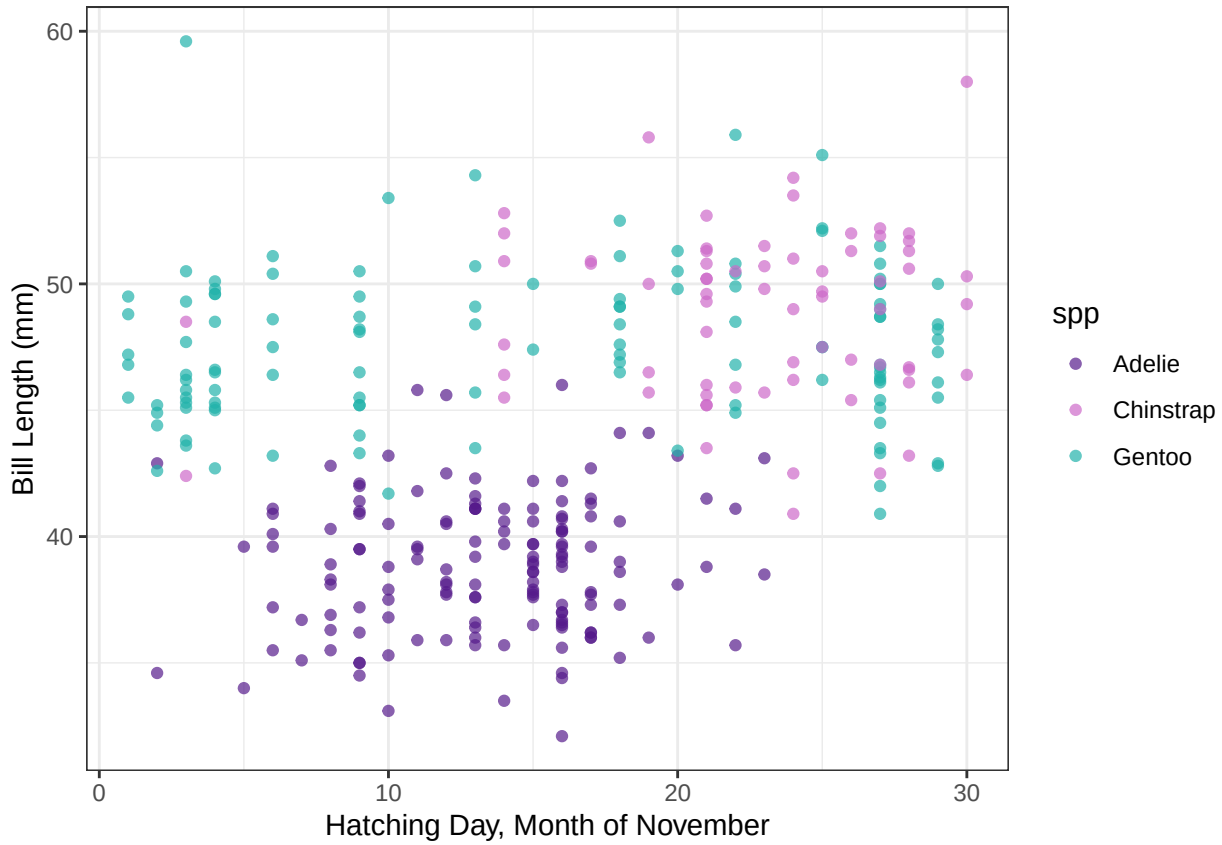
```
# let's look at bill length by egg hatching date 'date_e'
ggplot(penguins, aes(x = date_e, y = bill_length_mm)) +
  geom_point()
```



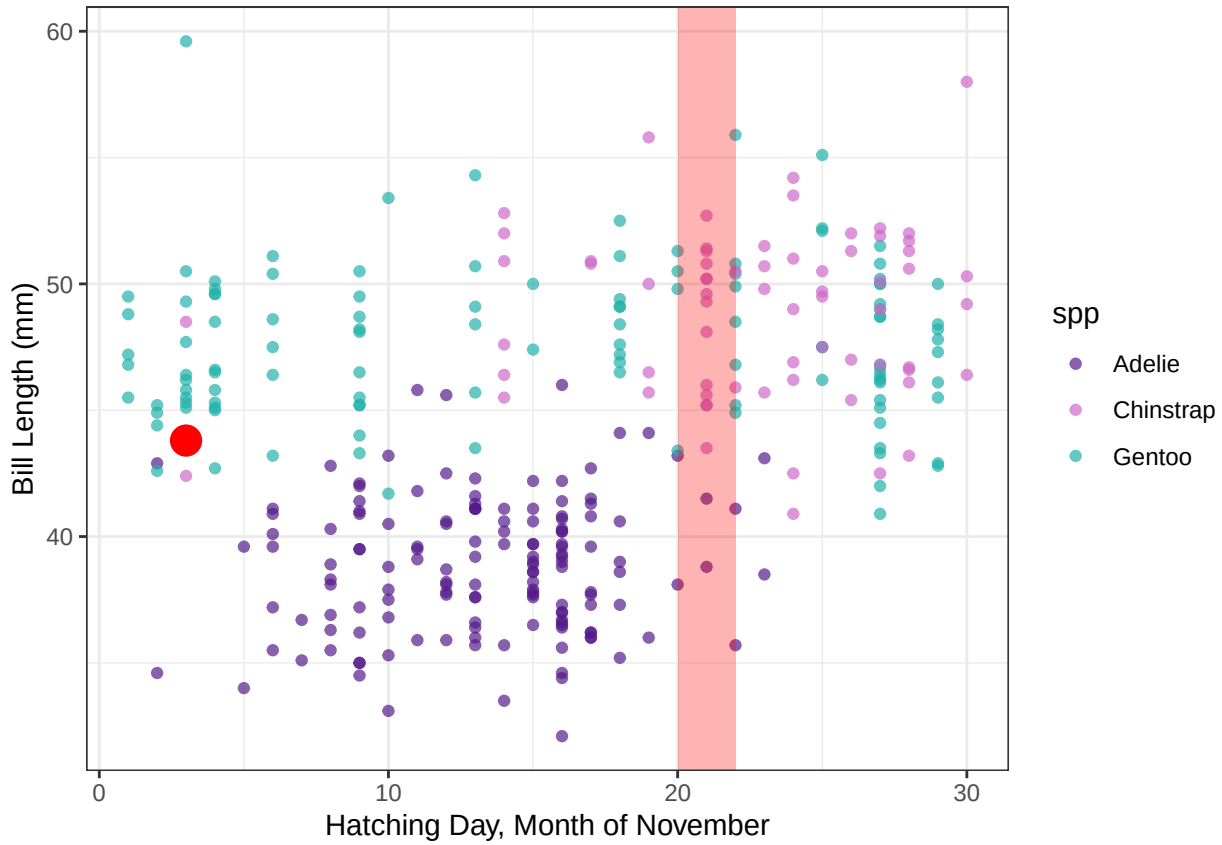

```
# hmm... looks like there was one season each year from 2007-2009...
# so, does it make sense to plot the x-axis as full dates? with months and days?

# seems like a waste of space.

# instead, we can focus on what we want to look at-- egg-laying season, which seems to be days within t
ggplot(penguins, aes(x = lubridate::day(date_e), y = bill_length_mm)) +
  geom_point()
```



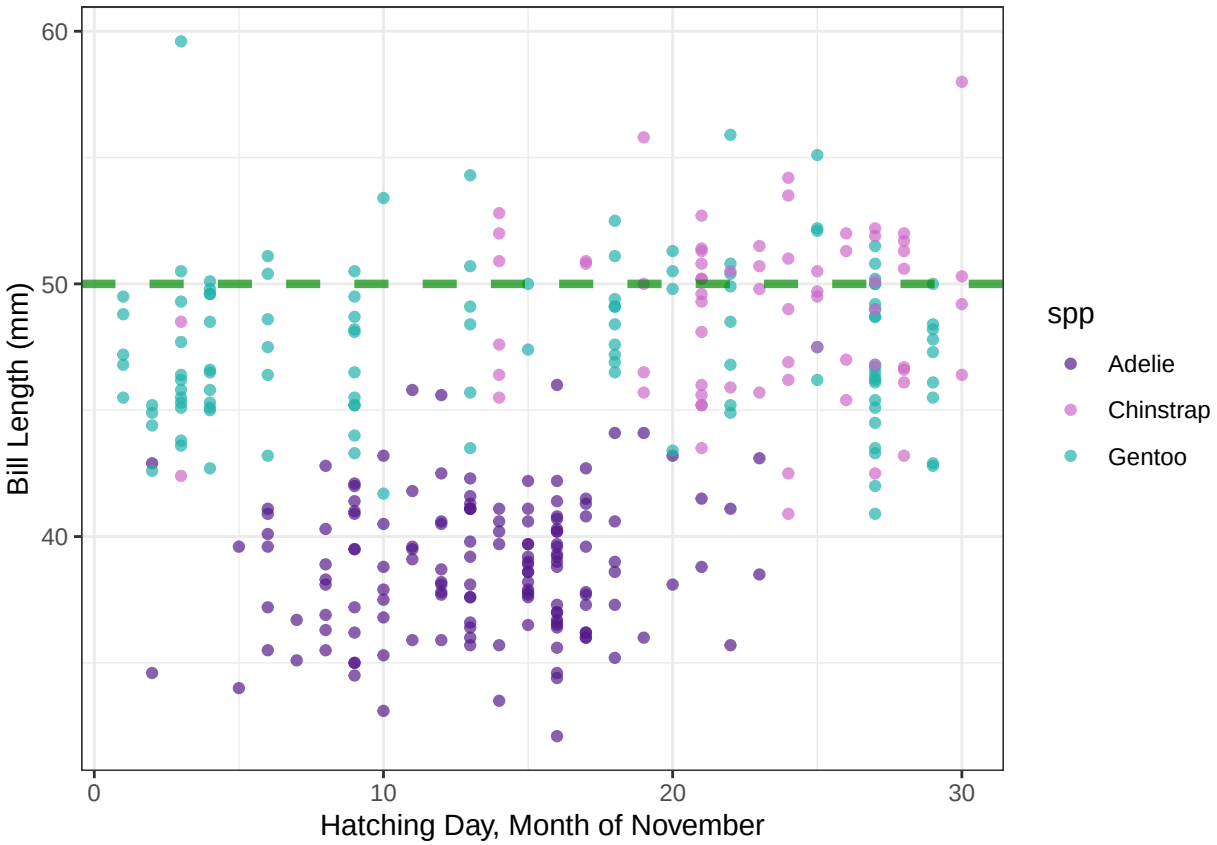

```
# good, but not there yet...

# let's try to see what's happening by species:
# add 'color = spp' to aes() in ggplot
ggplot(penguins, aes(x = lubridate::day(date_e), y = bill_length_mm, color = spp)) +
  geom_point(alpha = 0.7) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  labs(x = "Hatching Day, Month of November", y = "Bill Length (mm)") +
  # scale_x_continuous(breaks = c(1:30)) +
  theme_bw()
```




```
# let's say something happened on a particular date that you want to highlight:
# for example, if a few days had rain:
ggplot(penguins, aes(x = lubridate::day(date_e), y = bill_length_mm, color = spp)) +
  geom_point(alpha = 0.7) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  annotate("rect",
           xmin = 20, # from the 20th
           xmax = 22, # to the 22nd
           ymin = Inf, # set ymin to +infinity
           ymax = -Inf, # set ymax to -infinity
           fill = "red",
           # fill = "skyblue",  # maybe use a color that makes sense for rain instead of red
           alpha = 0.3) +
  geom_point(data = penguins %>% filter(individual_id == "N16A1"), color = "red", size = 5) +
  # annotate("text", label = "<-- Monsoon", x = 25, y = 58, color = "blue") + # include text annotation
  labs(x = "Hatching Day, Month of November", y = "Bill Length (mm)") +
  theme_bw()
```




```
# or if you want to highlight a threshold, use a single line (instead of a rectangle):
ggplot(penguins, aes(x = lubridate::day(date_e), y = bill_length_mm, color = spp)) +
  geom_point(alpha = 0.7) +
  scale_color_manual(values = c("purple4", "orchid3", "lightseagreen")) +
  geom_hline(yintercept = 50, color = "green4", linetype = "dashed", linewidth = 1.5, alpha = 0.7) +
  # geom_vline(xintercept = 15, color = "red", linetype = "solid", linewidth = 1.5, alpha = 0.7) +
  labs(x = "Hatching Day, Month of November", y = "Bill Length (mm)") +
  theme_bw()
```



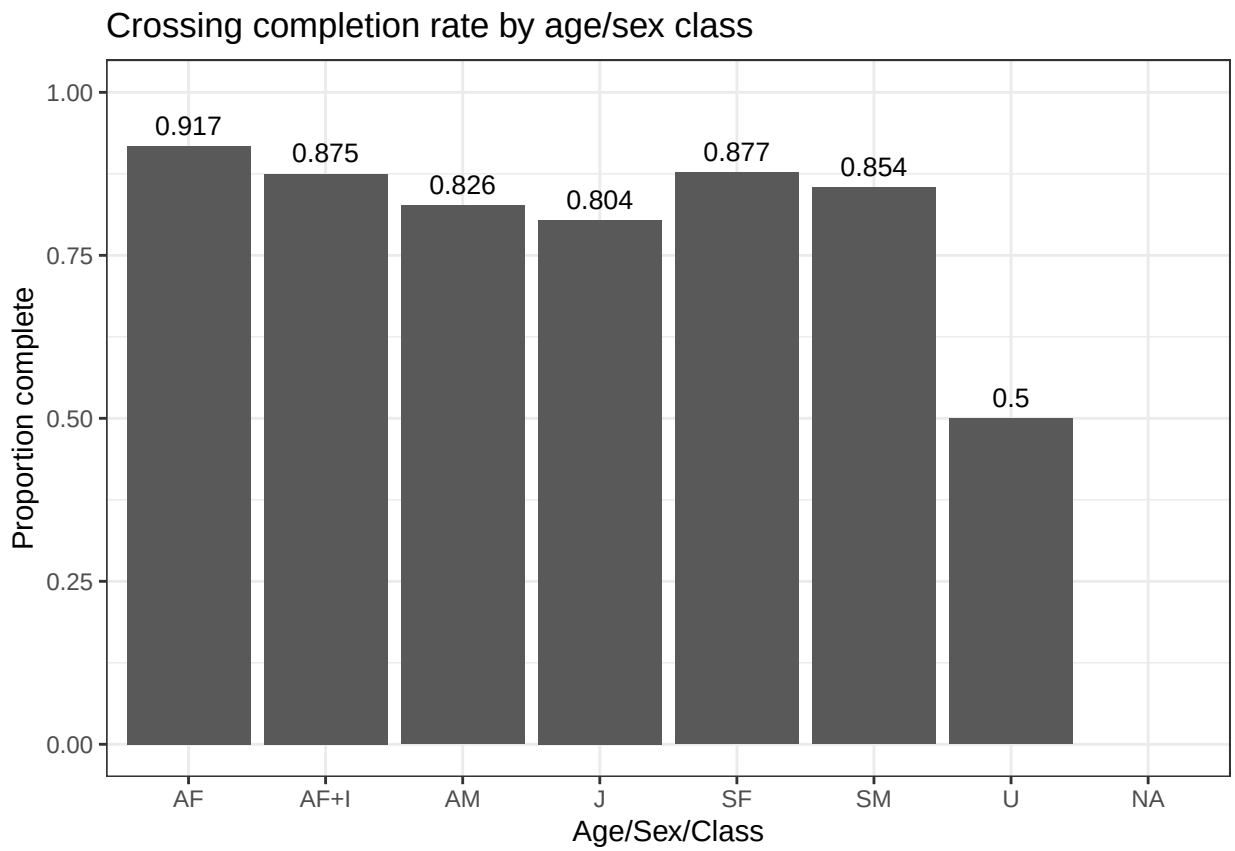
4. Primate Data

4.1 Explore

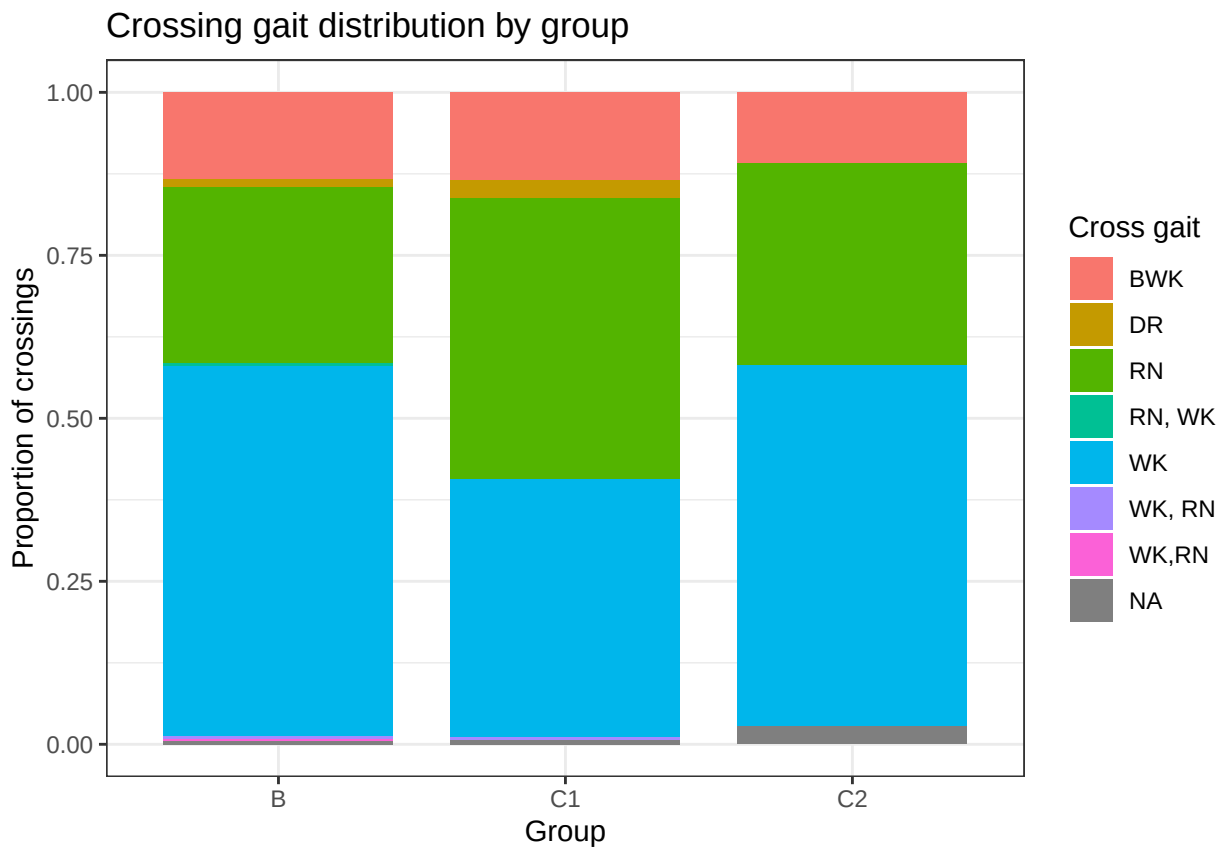
```
# let's have a look at the data:
glimpse(crossings)
```

```
## Rows: 791
## Columns: 13
## $ date                      <chr> "7/29/25", "7/29/25", "7/29/25", "7/29/25", ~
## $ crossing_id_number        <dbl> 10101, 10102, 10103, 10104, 10105, 10106, 10~
## $ start_time                <time> 11:30:00, 11:30:00, 11:30:00, 11:30:00, 11:~
## $ group                     <chr> "B", "B", "B", "B", "B", "B", "B", "B", "B",~
## $ time_of_crossing          <chr> "11:32:00", "11:32:37", "11:32:37", "11:33:2~
## $ id                        <chr> "U", "U", "U", "Rimba", "U", "U", "Rimba", "~
## $ age_sex_class             <chr> "J", "J", "J", "SM", "J", "J", "SM", "J", "S~
## $ cross_gait                <chr> "RN", "WK", "WK", "BWK", "RN", "WK", "WK", "~
## $ mid_cross_behavior        <chr> "TC", NA, "TC", NA, NA, NA, "MST, FE", NA, N~
## $ cross_complete            <chr> "C", "C", "C", "C", "C", "C", "C", "C", "C",~
## $ post_cross_food_behavior  <chr> "FO, FE", "FE", NA, NA, NA, "FE", NA, "FO", ~
## $ post_cross_other_behavior <chr> NA, NA, "MO", "SB", "PL", NA, "SG", NA, "SG"~
## $ notes                     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
```

```r
# Ok, no numeric variables. So, I'd want to compile them somehow.
# Let's group by a/s class, and count the number of crossings per a/s class
# and then convert that to a proportion
# Completion rate by sex
cross1 <- crossings %>%
  # dplyr::filter(!is.na(age_sex_class)) %>%
  group_by(age_sex_class) %>%
  summarise(
    n_crossings = n(),
    prop_complete = mean(cross_complete == "C", na.rm = TRUE)
  )

# This will give:
# Count per age class
# Proportion complete (numeric 0-1)

# plot:
# Bar chart of prop_complete
# Add n_crossings as text labels
ggplot(cross1, aes(x = age_sex_class, y = prop_complete)) +
  geom_col() +
  geom_text(aes(label = round(prop_complete, 3)), vjust = -0.6, size = 3.5) +
  scale_y_continuous(limits = c(0, 1)) +
  labs(
    title = "Crossing completion rate by age/sex class",
    x = "Age/Sex/Class",
    y = "Proportion complete"
  ) +
  theme_bw()
```

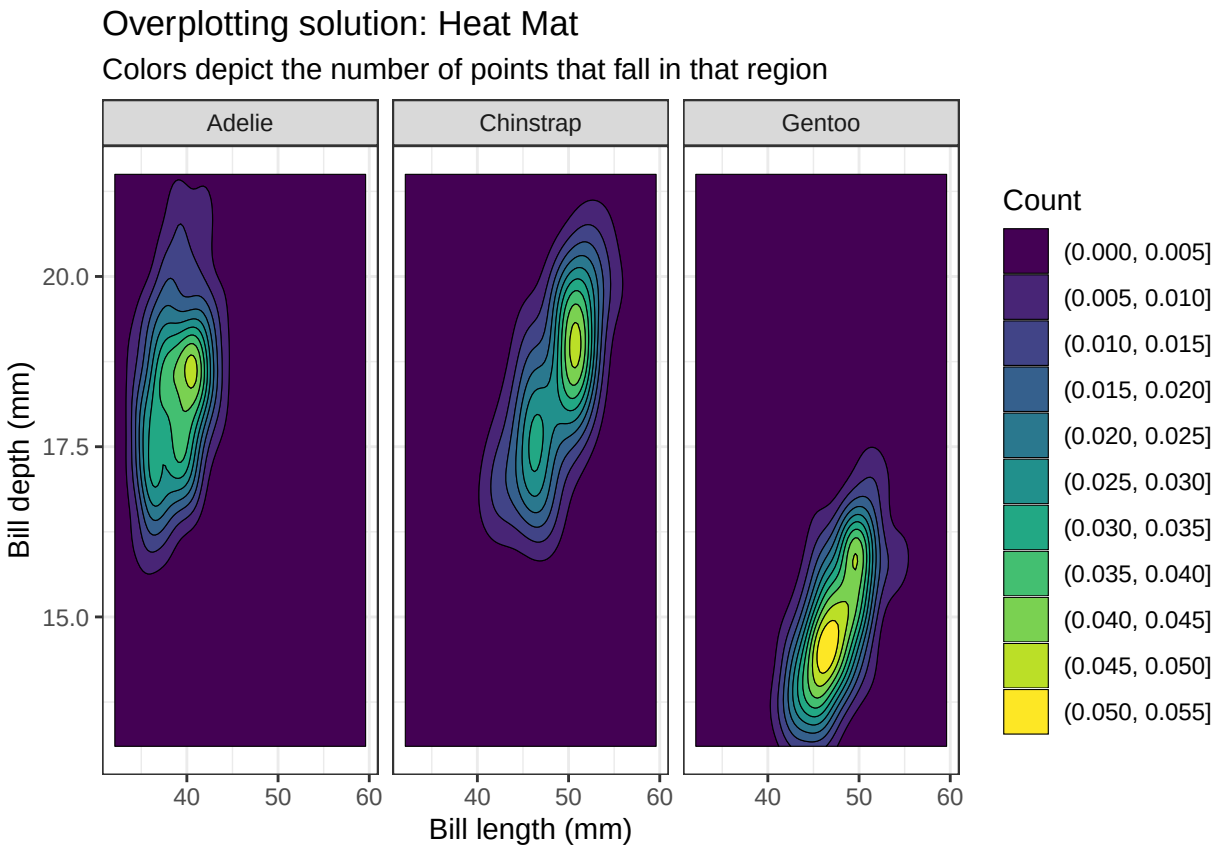



```r
# add: %>% filter(!is.na(age_sex_class)) # to wrangling

# ok, let's look at how
# Gait distribution by group
cross2 <- crossings %>%
  group_by(group, cross_gait) %>%
  summarise(n = n(), .groups = "drop") %>%
  group_by(group) %>%
  mutate(prop = n / sum(n))

ggplot(cross2, aes(x = group, y = prop, fill = cross_gait)) +
  geom_col(width = 0.8) +
  scale_y_continuous(limits = c(0, 1)) +
  labs(
    title = "Crossing gait distribution by group",
    x = "Group",
    y = "Proportion of crossings",
    fill = "Cross gait"
  ) +
  theme_bw()
```



5. Bonus content

```
# heat map;
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm)) +
  geom_density2d_filled(color = "black", linewidth = 0.2) +
  facet_wrap(~ spp) +
  labs(
    title = "Overplotting solution: Heat Mat",
    subtitle = "Colors depict the number of points that fall in that region",
    x = "Bill length (mm)",
    y = "Bill depth (mm)",
    fill = "Count"
  ) +
  theme_bw()
```


Overplotting solution: Heat Mat
Colors depict the number of points that fall in that region
Adelie
Chinstrap
Gentoo
Bill depth (mm)
20.0
17.5
15.0
40
50
60
Bill length (mm)
Count
(0.000, 0.005]
(0.005, 0.010]
(0.010, 0.015]
(0.015, 0.020]
(0.020, 0.025]
(0.025, 0.030]
(0.030, 0.035]
(0.035, 0.040]
(0.040, 0.045]
(0.045, 0.050]
(0.050, 0.055]