Visualizing Data

Ranji Raj

2021-04-17

Contents

Packages and data

In this analysis, we focus on the Palmer penguins dataset which contains size measurements, clutch observations, and blood isotope ratios for adult foraging Adélie, Chinstrap, and Gentoo penguins observed on islands in the Palmer Archipelago near Palmer Station, Antarctica.

```
glimpse(penguins)
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
## $ sex               <fct> male, female, female, NA, female, male, female, male~
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

We can inspect the dataset's dictionary with `?penguins`.

The dataset has 344 rows and 8 columns.

Okay, now let's investigate whether there are differences in size measurements between the different penguin species about which the dataset contains information. We obtain the number of levels (i.e. unique categories) of a factor variable `x` with `nlevels(x)` or `length(levels(x))`. More generally, we can obtain the number of unique values in any type of variable `y` with `length(unique(x))`.

There are 152, 68, 124 species for which the dataset contains information.

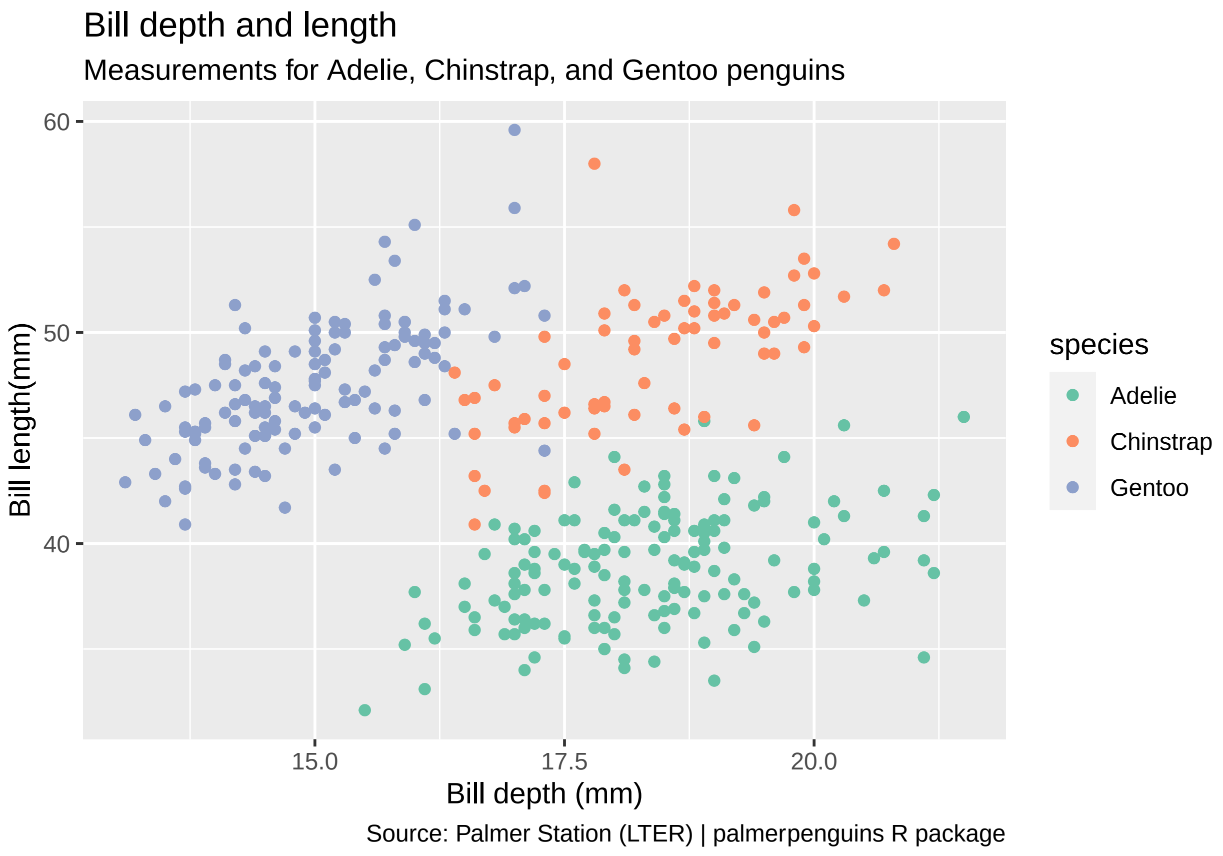
Scatterplot

Build a scatterplot according to the following description:

- Use the `penguins` data frame.
- Map `bill depth` to the x-axis and `bill length` to the y-axis.
- Represent each observation with a point and map species to the color of each point.
- Title the plot “Bill depth and length”, and the subtitle “Measurements for Adelie, Chinstrap, and Gentoo penguins”, label the x and y axes as “Bill depth (mm)” and “Bill length(mm)”, respectively. Label the legend “Species”. Name the source at the bottom: “Palmer Station (LTER) | palmerpenguins R package”.
- Use the discrete color scale “Set2” from the RColorBrewer package. Hint: `scale_color_brewer()`

```
ggplot(penguins, aes(x = bill_depth_mm  , y = bill_length_mm , color = species))+
  geom_point() +
  labs(
    title = "Bill depth and length",
    subtitle = "Measurements for Adelie, Chinstrap, and Gentoo penguins",
    x = "Bill depth (mm)", y = "Bill length(mm)",
    legend = "Species",
    caption = "Source: Palmer Station (LTER) | palmerpenguins R package"
  ) +
  scale_color_brewer(palette = "Set2")
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



Note that we receive a warning. We can verify how many missing values a vector `x` has using `sum(is.na(x))`. First, `is.na()` returns for each element whether it is missing, and then `sum()` calculates the number of `TRUE` values.

There are 2 missing values in `bill_depth_mm` and 11 missing values in `sex`.

Global vs. local aesthetical mapping

Next, let's explore the difference between **global** and **local** aesthetical mapping.

Copy your code from the chunk above (peng-scatter) into the code chunk below. Then, add a linear trend line to the plot. Hint: `?geom_smooth()`

We set the `color` aesthetic globally in the `ggplot` function and both the points and the regression lines are colored accordingly.

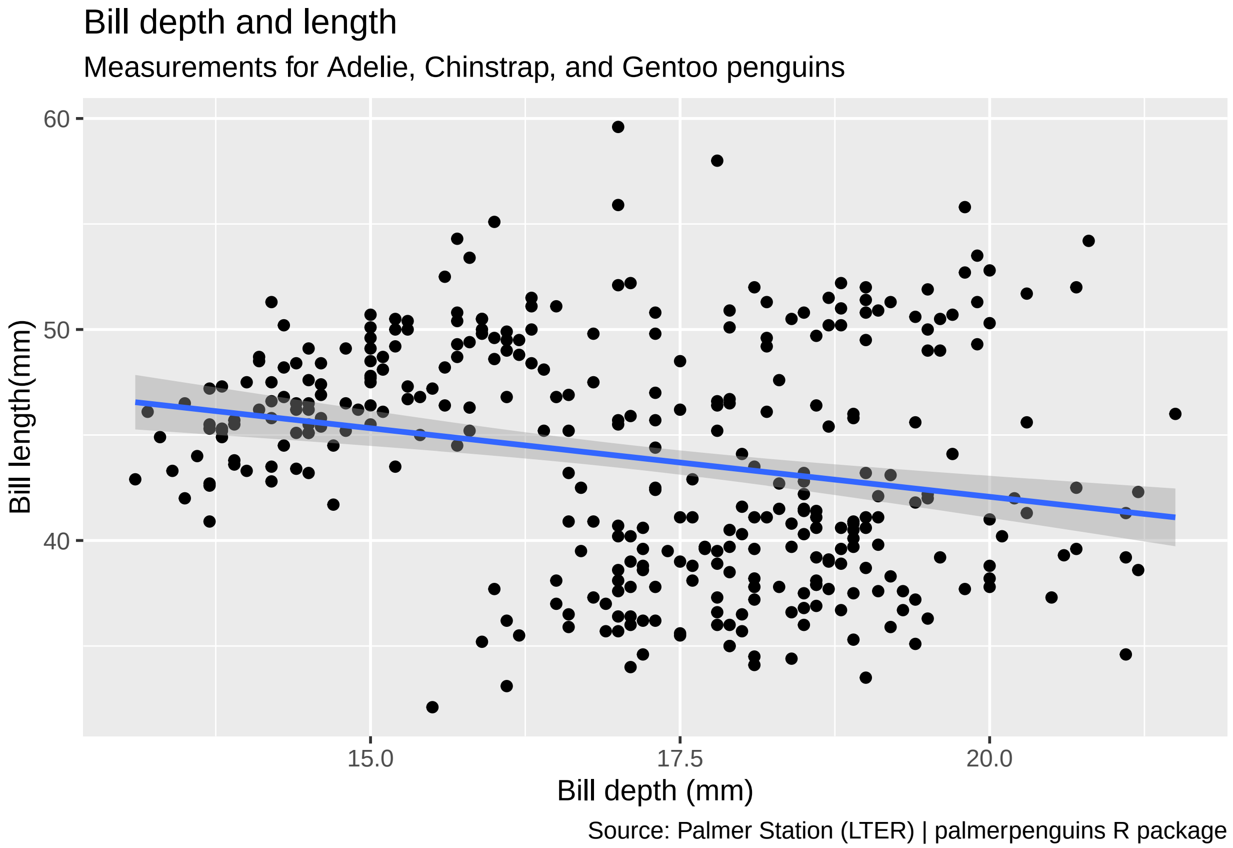
Change the `color` mapping so that it applies *only* for the scatterplot layer and not for the trend line layer. What do you observe?

```
ggplot(penguins, aes(x = bill_depth_mm  , y = bill_length_mm), color = species)+
  geom_point() +
  labs(
    title = "Bill depth and length",
    subtitle = "Measurements for Adelie, Chinstrap, and Gentoo penguins",
    x = "Bill depth (mm)", y = "Bill length(mm)",
    legend = "Species",
    caption = "Source: Palmer Station (LTER) | palmerpenguins R package"
  ) +
  scale_color_brewer(palette = "Set2") +
  geom_smooth(method = "lm")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

```
## Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



Note also that we extensively labeled the plot. For most of the following plots we don't explicitly label title, axis labels, etc. because we want to focus on what is happening in the plots. **But you should always label your plots properly!**

Mapping vs. setting of layer arguments

We can of course also make dataset-independent adjustments to the visual appearance of our plots.

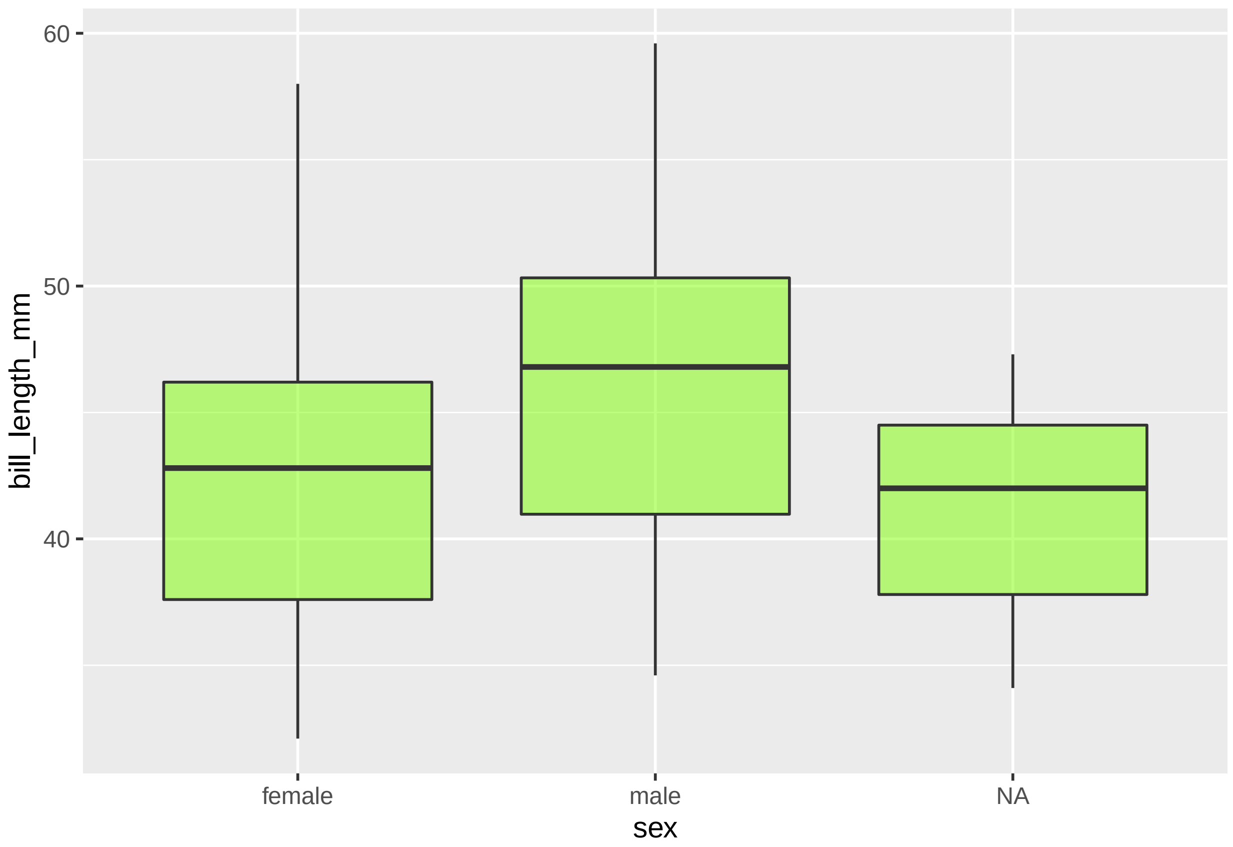
For example, for the boxplots below we mapped sex to fill color and transparency of the boxes.

However, we actually want to customize fill color and transparency, but they should be the same for all boxes.

Change the boxplot fill color to "chartreuse" and the boxplot transparency to 0.5.

```
ggplot(penguins, aes(x = sex, y = bill_length_mm, fill = sex, alpha = sex,)) +
  geom_boxplot(fill = "chartreuse", alpha=.5)
```

```
## Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```



Faceting

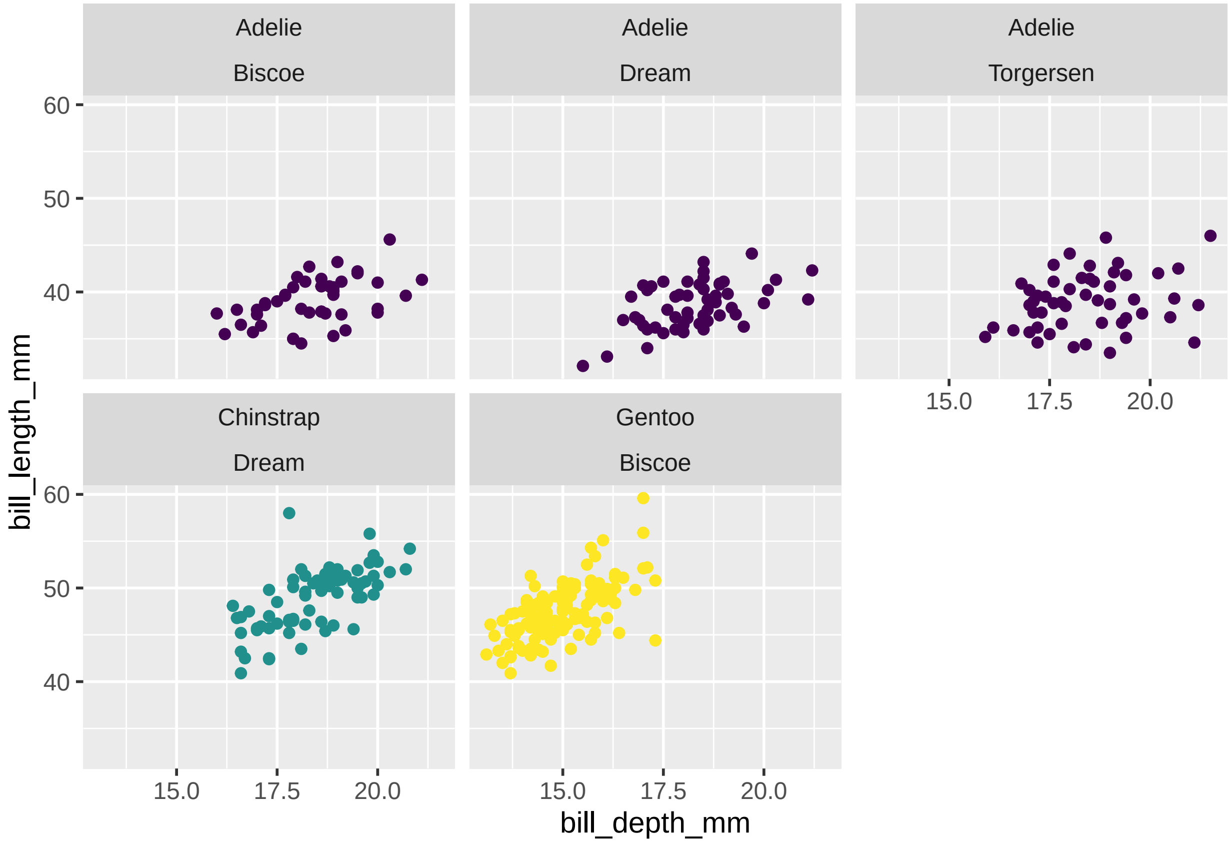
Faceting allows for showing multiple smaller plots that display different subsets of the data. It is useful for exploring conditional relationships, for identifying confounders(*a variable that influences both dependent and independent variable causing spurious association*) or just to spread dense data to reduce problems related to overplotting.

Create a faceted scatterplot:

- Map `bill_depth_mm` and `bill_length_mm` to x- and y-coordinates, respectively. Map `species` to color.
- Arrange the data as a SPLOM where rows represent `species` and columns represent `island`. Hint: Use either `facet_wrap()` or `facet_grid()`. Think about what makes more sense here? What happens if you change the one with the other?

```
ggplot(penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)) +
  geom_point() +
  facet_wrap(species ~ island) +
  scale_color_brewer() +
  guides(color = "none")+
  scale_colour_viridis_d()
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
## Warning: Removed 2 rows containing missing values (geom_point).
```



- `facet_grid()`: Creates facets for all the possible combination including confounders which makes difficult to go though if we have a huge dataset.
- `facet_grid()`: Creates only necessary facets to focus upon.

Additionally, apply the following adjustments:

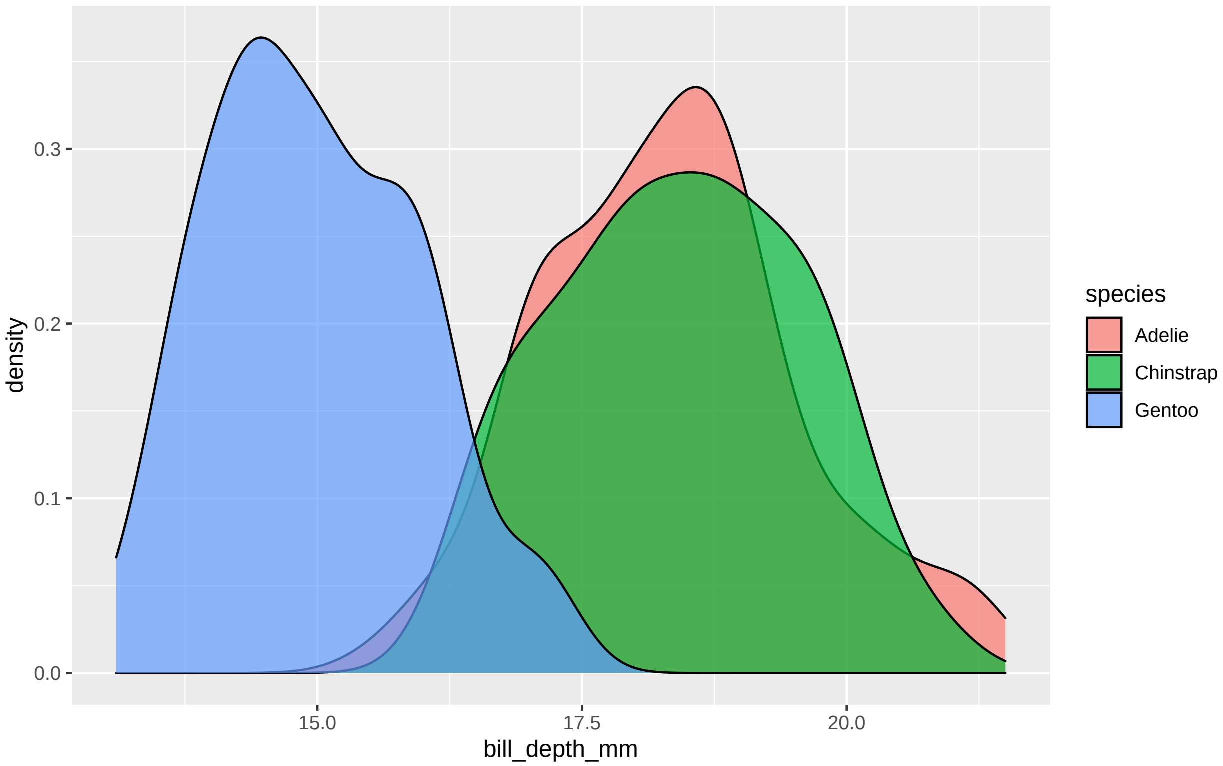
- Use the discrete Viridis color scale. Hint: Look for the corresponding scale function name in the "See Also" section in the help for `?scale_color_discrete`.
- Hide the color legend. Hint: See examples in the help for `?guides`.

Visualizing numerical variables with density curves

Create a density plot that shows the distribution of `bill_depth_mm` for each species. Each species' density curve should be filled with a different color. To help people with color perception deficiencies to distinguish the colors, use the OkabeIto color scale from the `colorblindr` package. Make sure to add some transparency to the density curves.

```
ggplot(penguins, aes(x = bill_depth_mm, fill = species)) +
  geom_density(alpha = 0.7) +
  colorblindr:: scale_color_OkabeIto()
```

```
## Warning: Removed 2 rows containing non-finite values (stat_density).
```


density
bill_depth_mm
species
Adelie
Chinstrap
Gentoo