Data Visualization – Part 3

Data 1 - mtcars

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3

Data 2 – penguins

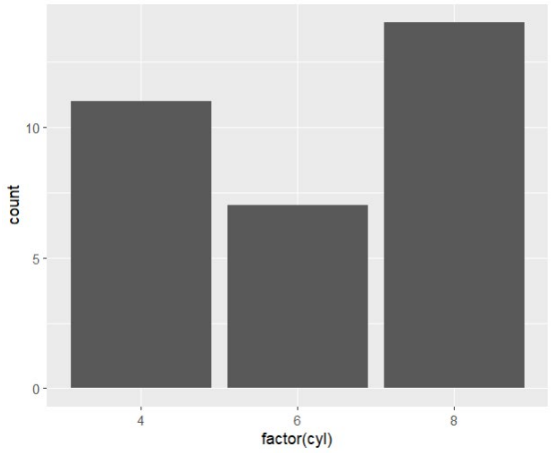
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
1	Adelie	Torgersen	39.1	18.7	181	3750	male	2007
2	Adelie	Torgersen	39.5	17.4	186	3800	female	2007
3	Adelie	Torgersen	40.3	18.0	195	3250	female	2007
4	Adelie	Torgersen	*NA*	*NA*	*NA*	*NA*	*NA*	2007
5	Adelie	Torgersen	36.7	19.3	193	3450	female	2007
6	Adelie	Torgersen	39.3	20.6	190	3650	male	2007
7	Adelie	Torgersen	38.9	17.8	181	3625	female	2007
8	Adelie	Torgersen	39.2	19.6	195	4675	male	2007
9	Adelie	Torgersen	34.1	18.1	193	3475	*NA*	2007
10	Adelie	Torgersen	42.0	20.2	190	4250	*NA*	2007
11	Adelie	Torgersen	37.8	17.1	186	3300	*NA*	2007
12	Adelie	Torgersen	37.8	17.3	180	3700	*NA*	2007
13	Adelie	Torgersen	41.1	17.6	182	3200	female	2007
14	Adelie	Torgersen	38.6	21.2	191	3800	male	2007

Changing Colors in Plots

Univariate Displays

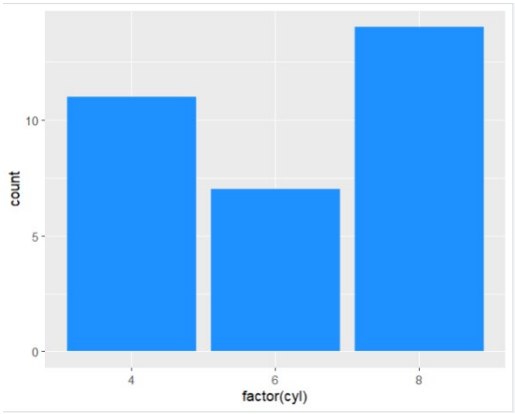
Example 1: Univariate simple bar graph (no color)

```
fig_1 <- mtcars %>%
  ggplot(aes(x = factor(cyl))) +
  geom_bar()
fig_1
```



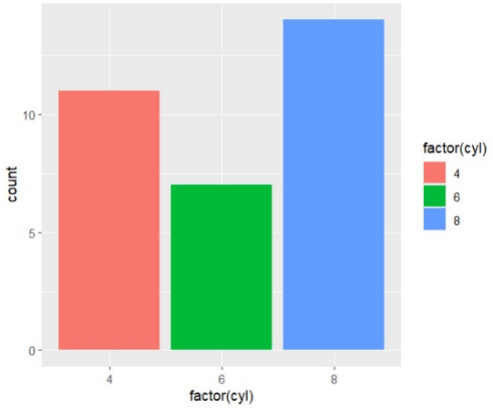
Example 2: Fills all the bars with the same color.

```
fig_2 <- mtcars %>%
  ggplot(aes(x = factor(cyl))) +
  geom_bar(fill = "dodgerblue")
fig_2
```



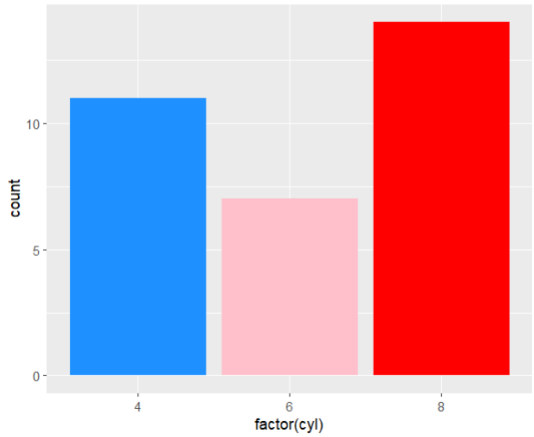
Example 3: Change each bar to a different color using fill inside "aes"

```
fig_3 <- mtcars %>%
  ggplot(aes(x = factor(cyl),
             fill = factor(cyl))) +
  geom_bar()
fig_3
```



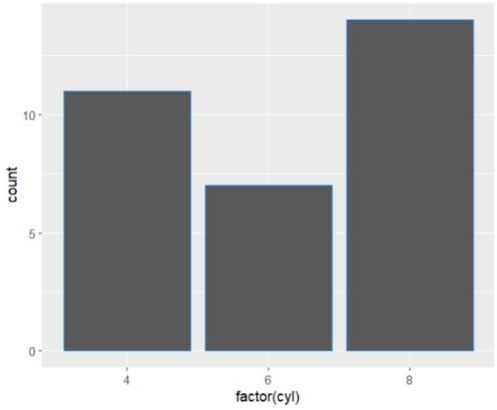
Example 4: Change each bar to a different color without the legend.

```
fig_4 <- mtcars %>%
  ggplot(aes(x = factor(cyl))) +
  geom_bar(fill = c("dodgerblue", "pink", "red"))
fig_4
```



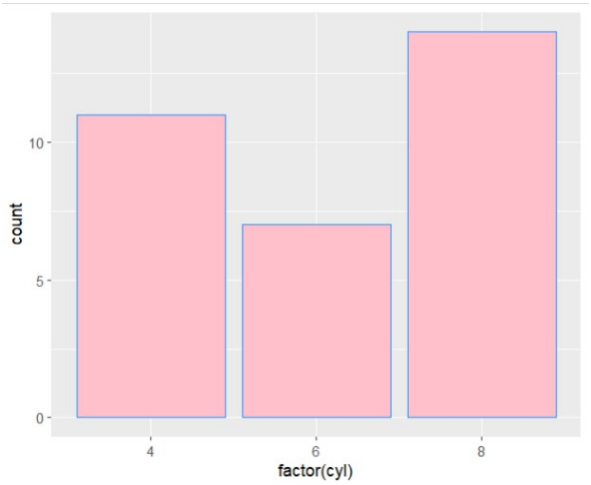
Example 5: Attempt to change bar color using color instead of fill. What happens?

```
fig_5 <- mtcars %>%
  ggplot(aes(x = factor(cyl))) +
  geom_bar(color = "dodgerblue")
fig_5
```



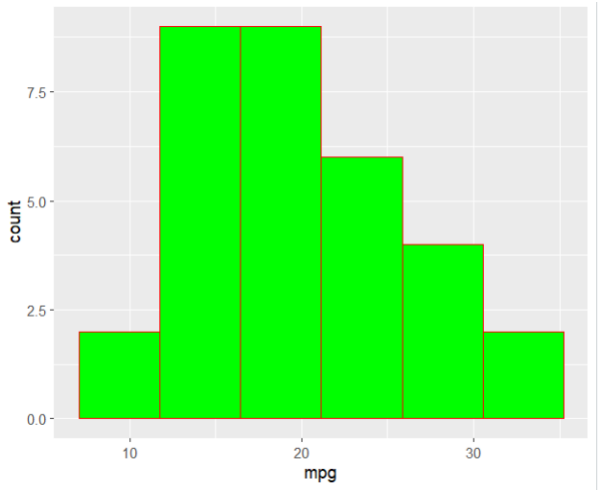
Example 6: Change the lines of the bars to dodgerblue and the bar colors to pink.

```
fig_6 <- mtcars %>%
  ggplot(aes(x = factor(cyl))) +
  geom_bar(fill = "pink", color = "dodgerblue")
fig_6
```



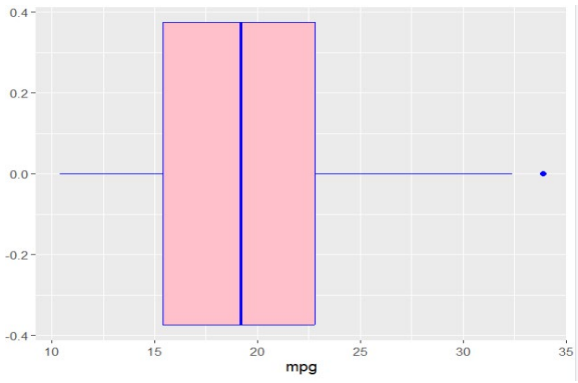
Example 7: Create a histogram for mpg variable with 6 bins. Then, change the lines of the histogram to red and the bar colors to green.

```
fig_7 <- mtcars %>%
  ggplot(aes(x = mpg)) +
  geom_histogram(bins = 6, fill = "green", color = "red")
fig_7
```



Example 8: Create a boxplot for mpg variable. Then, change the lines of the plot to blue and the inside colors to pink.

```
fig_8 <- mtcars %>%
  ggplot(aes(x = mpg)) +
  geom_boxplot(color = "blue", fill = "pink")
fig_8
```



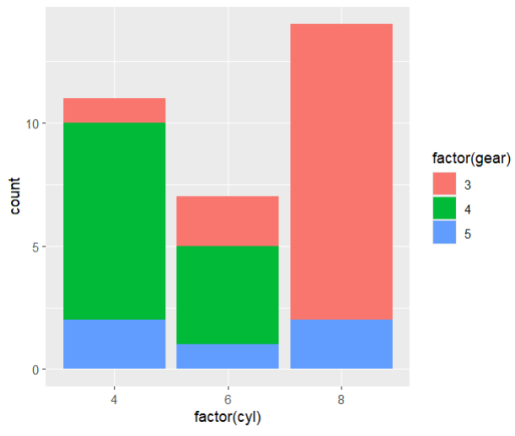
Multivariate Displays

For multivariate displays, you typically need to use color or fill inside the aes. To change the colors for a multivariable plot, you will need the functions:

- `scale_fill_manual()`: manually changes fill in the aes.
- `scale_color_manual()`: manually changes color in the aes.

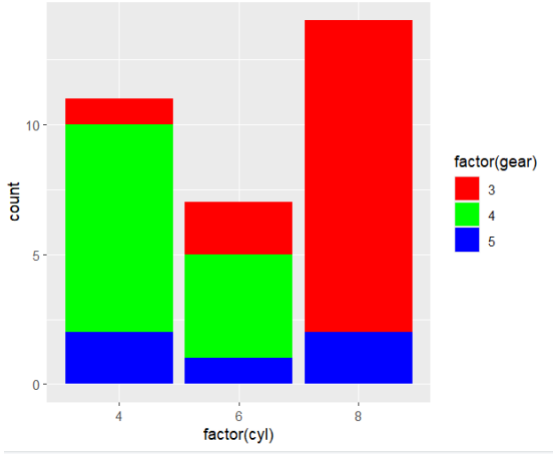
Example 9: Create a stacked bar graph with cyl in the x axis and gear as the categorical variable to split each column.

```
fig_9 <- mtcars %>%
  ggplot(aes(x = factor(cyl), fill = factor(gear)))+
  geom_bar()
fig_9
```



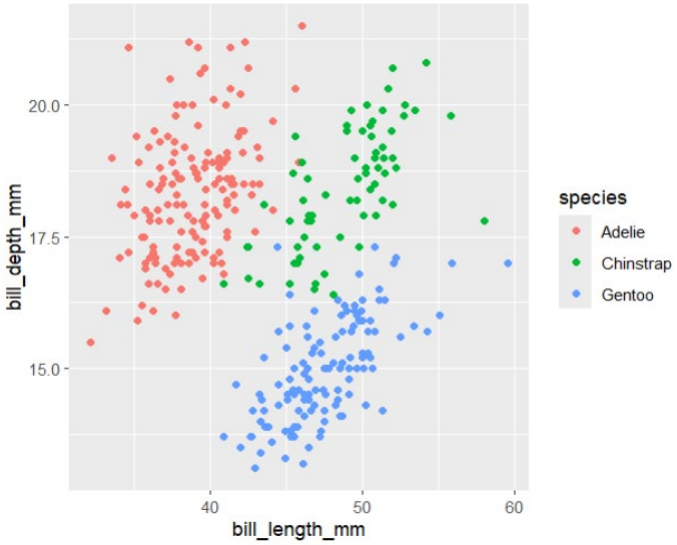
Example 10: Change the colors from Example 9 to red, green, and blue (in that order).

```
fig_10 <- mtcars %>%
  ggplot(aes(x = factor(cyl), fill = factor(gear)))+
  geom_bar()+
  scale_fill_manual(values = c("red", "green", "blue"))
fig_10
```



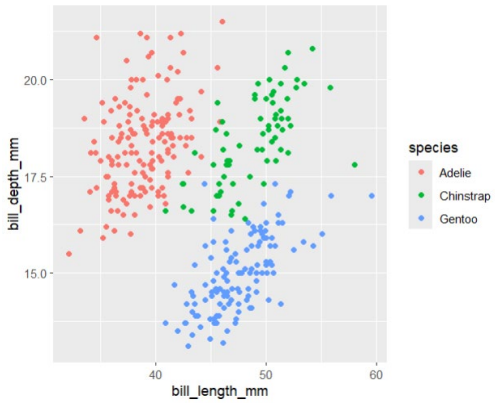
Example 11: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable.

```
fig_11 <- penguins %>%
  ggplot(aes(x = bill_length_mm,
             y = bill_depth_mm,
             color = species)) +
  geom_point()
fig_11
```



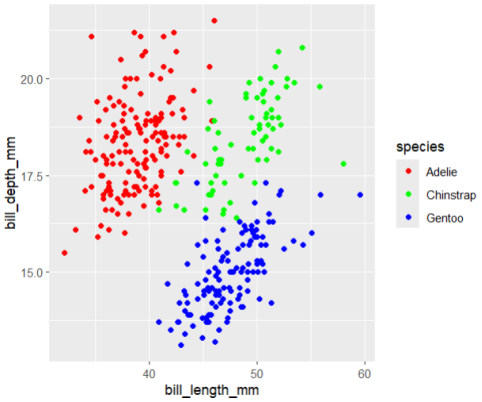
Example 12: Change the colors from Example 11 to red, green, and blue, in that order. What is wrong with these codes? Why isn't this one working?

```
fig_12 <- penguins %>%
  ggplot(aes(x = bill_length_mm,
             y = bill_depth_mm,
             color = species)) +
  geom_point() +
  scale_fill_manual(values = c("red", "green", "blue"))
fig_12
```



Example 13: Change the colors from Example 11 to red, green, and blue, in that order.

```
fig_13 <- penguins %>%
  ggplot(aes(x = bill_length_mm,
             y = bill_depth_mm,
             color = species)) +
  geom_point() +
  scale_color_manual(values = c("red", "green", "blue"))
fig_13
```

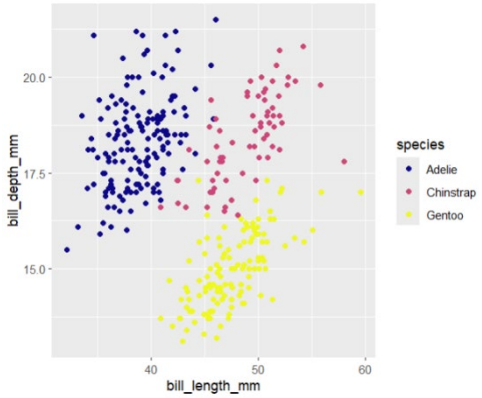


Using External Color Packages

Sometimes, you may want to use specialized color palettes from external packages to enhance your data visualizations. Packages like `wesanderson` and `viridis` provide aesthetically pleasing color palettes that can make your plots more visually appealing.

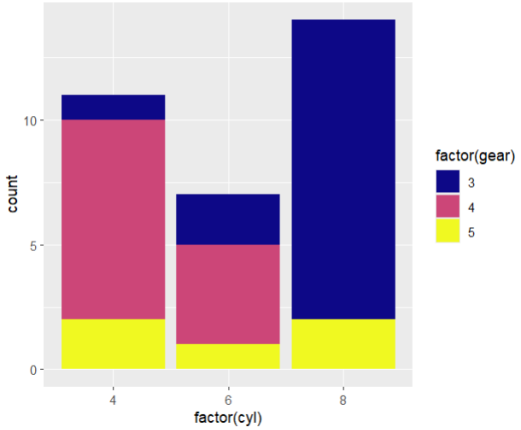
Example 14: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable. Use the viridis package to change the colors.

```
fig_14 <- penguins %>%
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point() +
  scale_color_viridis(discrete = TRUE, option = "C", alpha = 1)
fig_14
```



Example 15: Create a stacked bar graph with cyl in the x axis and gear as the categorical variable to split each column. Use the viridis package to change the colors.

```
fig_15 <- mtcars %>%
  ggplot(aes(x = factor(cyl), fill = factor(gear)))+
  geom_bar() +
  scale_fill_viridis(discrete = TRUE, option = "C", alpha = 1)
fig_15
```



Themes

1. `theme_gray()`:

   - This is the default theme in ggplot2.
   - It uses a simple gray background with white gridlines.
   - A good choice when you want a clean, minimalist look for your plot.

2. `theme_bw()`:

   - This theme provides a white background with black gridlines.
   - It offers a high-contrast, black-and-white appearance.
   - Useful for creating plots that need to be easily readable in black and white.

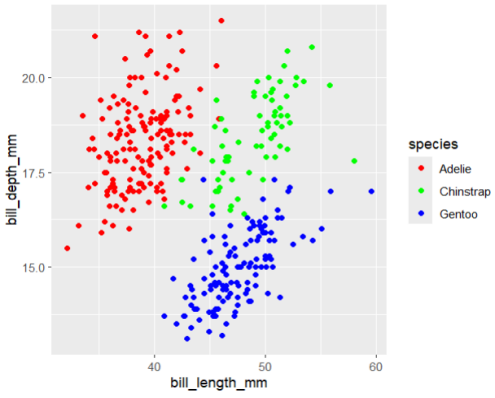
3. `theme_minimal()`:

   - As the name suggests, this theme is minimalistic.
   - It removes gridlines and most background elements, leaving a clean, white canvas.
   - Suitable for plots where you want to focus on the data without distractions.

4. `theme_void()`:

   - This theme removes nearly all elements, providing a blank canvas.
   - It's useful when you want to start with a clean slate and add custom elements.

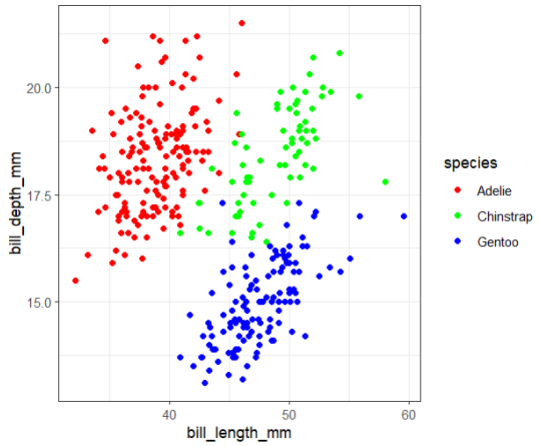
Example 16: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable. Use a gray theme




```
fig_16 <- penguins %>%
  ggplot(aes(x = bill_length_mm,
             y = bill_depth_mm,
             color = species)) +
  geom_point() +
  scale_color_manual(values = c("red", "green", "blue")) +
  theme_gray()
fig_16
```

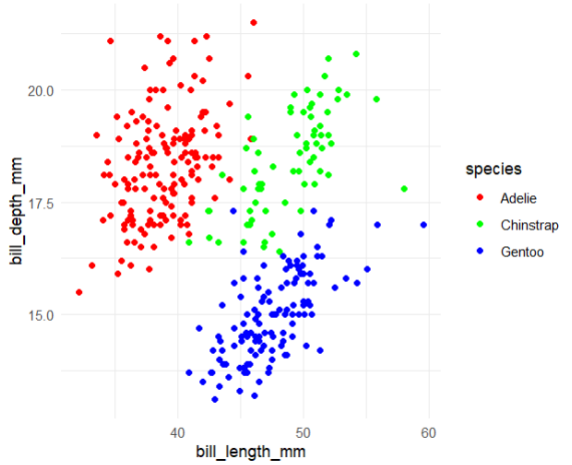
Example 17: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable. Use a black and white theme

```
fig_17 <- penguins %>%
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point() +
  scale_color_manual(values = c("red", "green", "blue")) +
  theme_bw()
fig_17
```



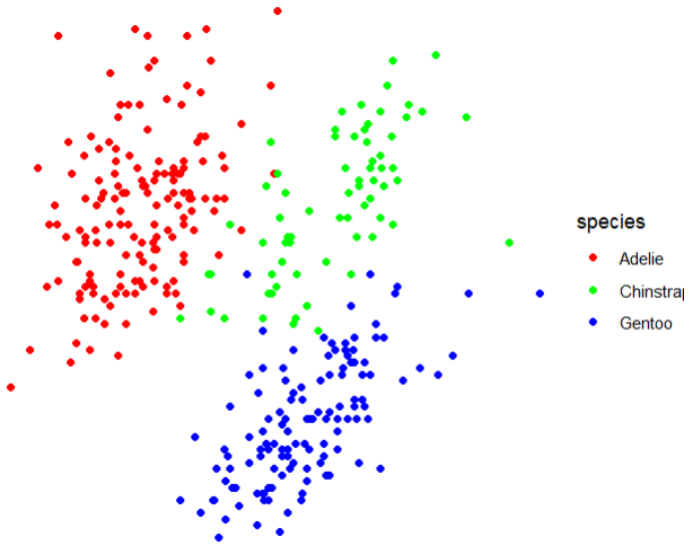
Example 18: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable. Use a minimal theme.

```
fig_18 <- penguins %>%
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point() +
  scale_color_manual(values = c("red", "green", "blue")) +
  theme_minimal()
fig_18
```



Example 19: Create a scatter plot with x = bill_length_mm, y = bill_depth_mm, and species as the categorical variable. Use a void theme.

```
fig_19 <- penguins %>%
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point() +
  scale_color_manual(values = c("red", "green", "blue")) +
  theme_void()
fig_19
```

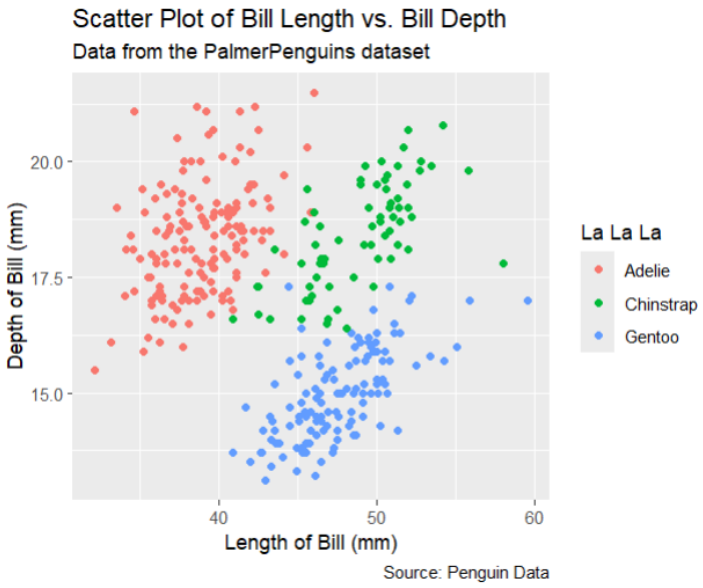


Customizing Labels with labs()

The `labs()` function in ggplot2 allows you to customize plot labels, including titles, subtitles, captions, and axis labels. You can use `labs()` to change the text displayed in various parts of your plot.

Example 20: Customizing Plot Labels

```
fig_20 <- penguins %>%
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) +
  geom_point() +
  labs(
    title = "Scatter Plot of Bill Length vs. Bill Depth",
    x = "Length of Bill (mm)",
    y = "Depth of Bill (mm)",
    subtitle = "Data from the PalmerPenguins dataset",
    caption = "Source: Penguin Data",
    color = "La La La"
  )
fig_20
```



Example 21: Customizing Plot Labels

```
fig_21 <- mtcars %>%
  ggplot(aes(x = factor(cyl), fill = factor(gear)))+
  geom_bar() +
  labs(
    title = "Bar Graph",
    x = "Cylinders",
    y = "Total Number",
    fill = "La La La"
  )

fig_21
```