Scales in `ggplot2`

Introduction

Scales in `ggplot2` control the mapping between data values and their visual representation in a plot. They define how data values map to axes, colors, sizes, shapes, and other aesthetics. By customizing scales, you can fine-tune the appearance and readability of your visualizations.

Theoretical Overview of Scales

Scales serve two primary functions in `ggplot2`:

- **Data Transformation:** Convert raw data values into a format suitable for visualization (e.g., log transformation).
- **Aesthetic Mapping:** Define how data values map to visual properties (e.g., colors, shapes, and sizes).

Types of Scales Scales in `ggplot2` are specific to aesthetics:

- **Position scales:** Control the mapping of data to x- and y-axes (`scale_x_` and `scale_y_`).
- **Color and fill scales:** Map data values to colors (`scale_color_`, `scale_fill_`).
- **Size scales:** Map data values to sizes (`scale_size_`).
- **Shape and linetype scales:** Control mapping of data to shapes and line types.
- **Manual scales:** Allow explicit control of mappings.

Continuous vs. Discrete Scales Scales are automatically chosen based on the type of data:

- **Continuous Scales:** For numeric data.
- **Discrete Scales:** For categorical data.

Common Parameters in Scales

Scales share several common parameters:

- **limits:** Defines the range of the scale.
- **breaks:** Specifies where ticks appear on the axis.
- **labels:** Sets custom labels for ticks.
- **trans:** Applies a transformation (e.g., `"log"`, `"sqrt"`).
- **expand:** Controls the space between the axis and data points.

Examples of Scale Customization

Example 1: Customizing Axes with `scale_x_` and `scale_y_`

Customizing axis limits, breaks, and labels.

```r
library(ggplot2)

ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  scale_x_continuous(limits = c(1, 6), breaks = seq(1, 6, by
      = 1), labels = paste0("Weight ", 1:6)) +
  scale_y_continuous(limits = c(10, 35), breaks = seq(10,
     35, by = 5)) +
  labs(title = "Custom Axes", x = "Weight (Custom)", y = "
     Miles per Gallon (Custom)") +
  theme_minimal()
```

Example 2: Transforming Scales

Applying a logarithmic transformation to the axes.

```r
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  scale_x_log10() +
  scale_y_sqrt() +
  labs(title = "Logarithmic and Square Root Transformed
     Scales", x = "Log Weight", y = "Sqrt MPG") +
  theme_light()
```

Example 3: Customizing Colors with `scale_color_manual`

Manually defining colors for a categorical variable.

```
ggplot(data = mtcars, aes(x = wt, y = mpg, color = factor(
    cyl))) +
  geom_point(size = 3) +
  scale_color_manual(values = c("red", "green", "blue")) +
  labs(title = "Manual Color Scale", x = "Weight", y = "
      Miles per Gallon", color = "Cylinders") +
  theme_classic()
```

Example 4: Gradient Colors with scale_color_gradient

Using a gradient scale for continuous data.

```
ggplot(data = mtcars, aes(x = wt, y = mpg, color = hp)) +
  geom_point(size = 3) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Gradient Color Scale", x = "Weight", y = "
      Miles per Gallon", color = "Horsepower") +
  theme_minimal()
```

Example 5: Size Scales with scale_size_continuous

Mapping a continuous variable to point sizes.

```
ggplot(data = mtcars, aes(x = wt, y = mpg, size = hp)) +
  geom_point(alpha = 0.7) +
  scale_size_continuous(range = c(2, 10)) +
  labs(title = "Continuous Size Scale", x = "Weight", y = "
      Miles per Gallon", size = "Horsepower") +
  theme_light()
```

Example 6: Shape Scales with scale_shape_manual

Mapping categorical data to custom shapes.

```
ggplot(data = mtcars, aes(x = wt, y = mpg, shape = factor(
    cyl))) +
  geom_point(size = 3) +
  scale_shape_manual(values = c(16, 17, 18)) +
  labs(title = "Custom Shape Scale", x = "Weight", y = "
      Miles per Gallon", shape = "Cylinders") +
  theme_classic()
```

Example 7: Fill Scales for Bar Charts

Using a gradient fill scale for a bar chart.

```
ggplot(data = mtcars, aes(x = factor(cyl), fill = mpg)) +
  geom_bar(stat = "identity") +
  scale_fill_gradient(low = "yellow", high = "red") +
  labs(title = "Gradient Fill Scale for Bar Chart", x = "
     Cylinders", y = "Miles per Gallon", fill = "MPG") +
  theme_minimal()
```

Example 8: Custom Labels with scale_y_continuous

Adding custom labels to the y-axis.

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  scale_y_continuous(labels = scales::dollar_format()) +
  labs(title = "Custom Y-Axis Labels", x = "Weight", y = "
     Miles per Gallon ($)") +
  theme_light()
```

Tips for Customizing Scales

- Use scale_x_continuous() and scale_y_continuous() for numerical transformations.

- Use manual scales (scale_color_manual(), scale_shape_manual()) for full control over mappings.

- Combine discrete and continuous scales for hybrid visualizations.

- Leverage scales package functions (e.g., scales::percent_format(), scales::dollar_format()).

- Use limits to focus on specific data ranges, and expand to adjust margins around data points.