Data Visualization Analysis

Chen

2024-01-01

0.1 Introduction

This tutorial is designed to help you learn data visualization analysis by providing simple and useful information in a way that is easy to follow and understand.

0.2 Preparation

In order to draw a chart, we need to include the required packages for visualization and dataset. For example, `ggplot2` package is for drawing charts, `gcookbook` is for using `pg_mean` dataset, and `palmerpenguins` provides the `penguins` dataset.

0.3 Bar Chart

In this section, we will draw a bar chart using `pg_mean` dataset. The dataset has two columns: `group`, `weight`.

```
pg_mean
```

```
  group weight
1  ctrl  5.032
2  trt1  4.661
3  trt2  5.526
```

This dataset compares the weight across three groups:

- `ctrl`: Control group (baseline, weight = 5.032).
- `trt1`: Treatment 1 group (weight = 4.661).
- `trt2`: Treatment 2 group (weight = 5.526).

```r
ggplot(pg_mean, aes(x = group, y = weight)) +
  geom_col()
```

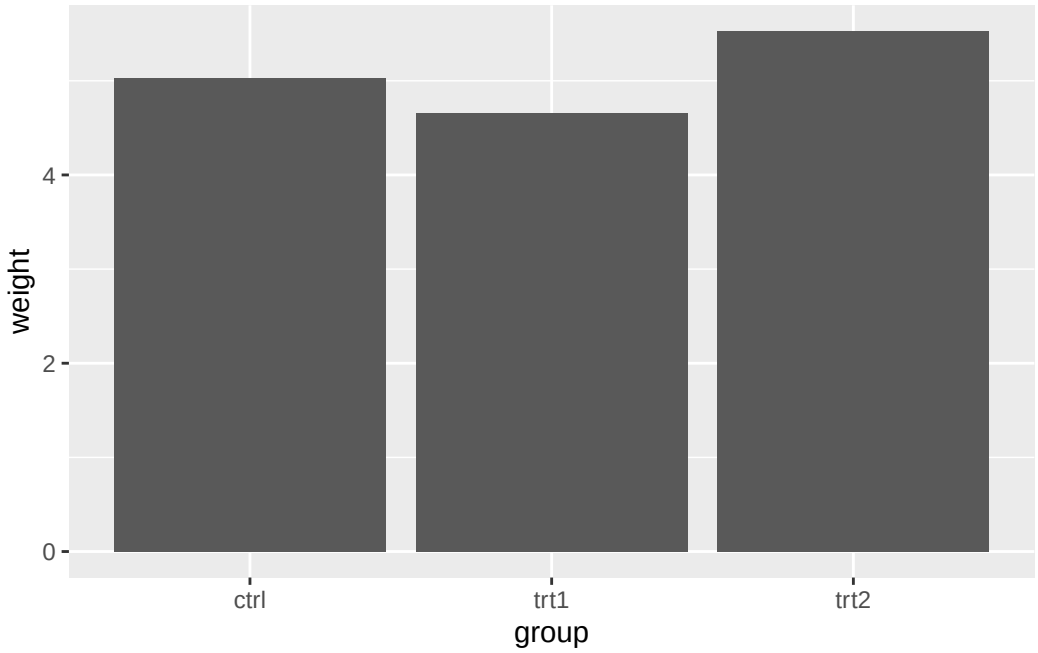

Figure 1: Bar Chart Example

0.4 Bar Chart with Color

```r
# Sample data
data <- penguins %>% filter(!is.na(body_mass_g))

# Bar chart by species
ggplot(data, aes(x = species, fill = species)) +
  geom_bar() +
  labs(title = "Bar Chart by Species", x = "Species", y = "Count")
```

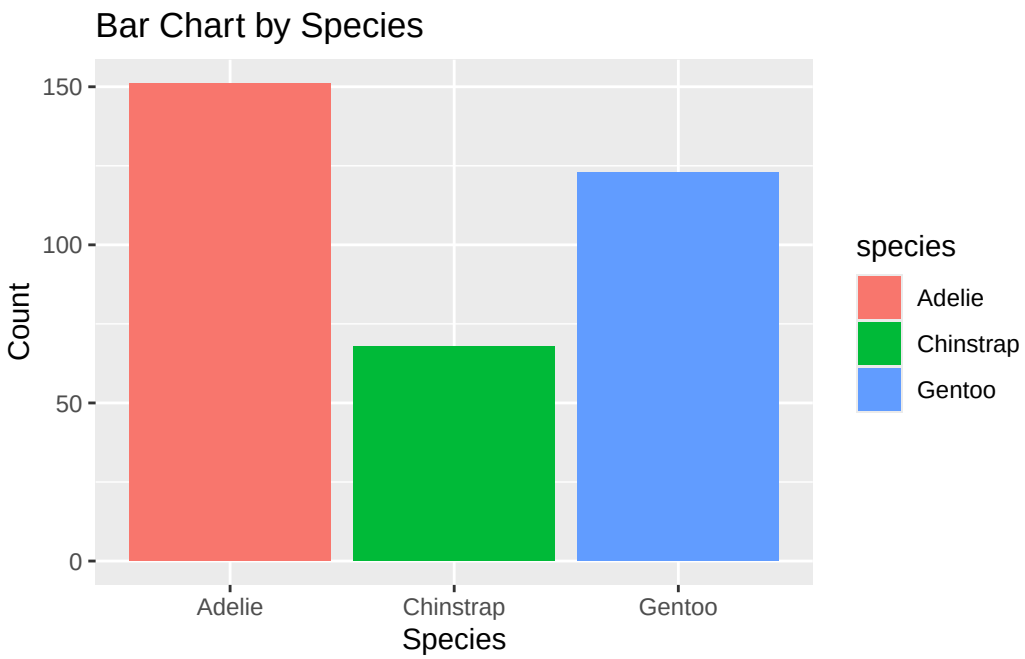

Figure 2: Bar Chart by Species

0.5 Line Chart

```
# Sample data
line_data <- tibble(
  x = 1:100,
  y = cumsum(rnorm(100))
)

# Line chart
ggplot(line_data, aes(x = x, y = y)) +
  geom_line(color = "blue") +
  labs(title = "Line Chart", x = "X-axis", y = "Y-axis")
```

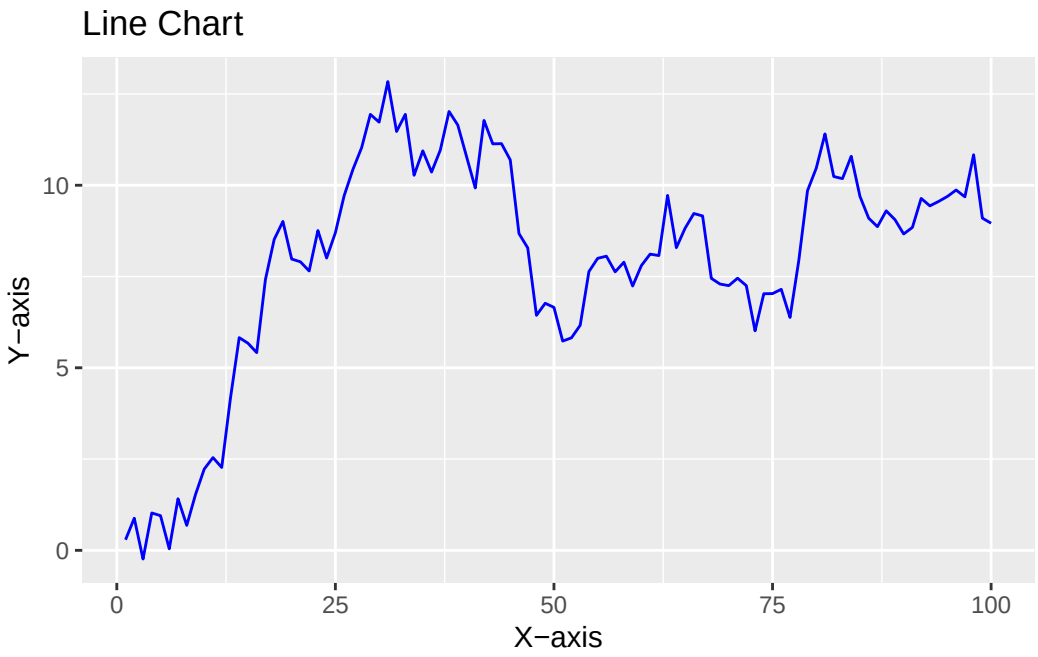

Figure 3: Line Chart

0.6 Histogram

```
# Histogram of penguin body mass
ggplot(data, aes(x = body_mass_g)) +
  geom_histogram(binwidth = 100, fill = "steelblue", color = "black") +
  labs(title = "Histogram of Body Mass", x = "Body Mass (g)", y = "Frequency")
```

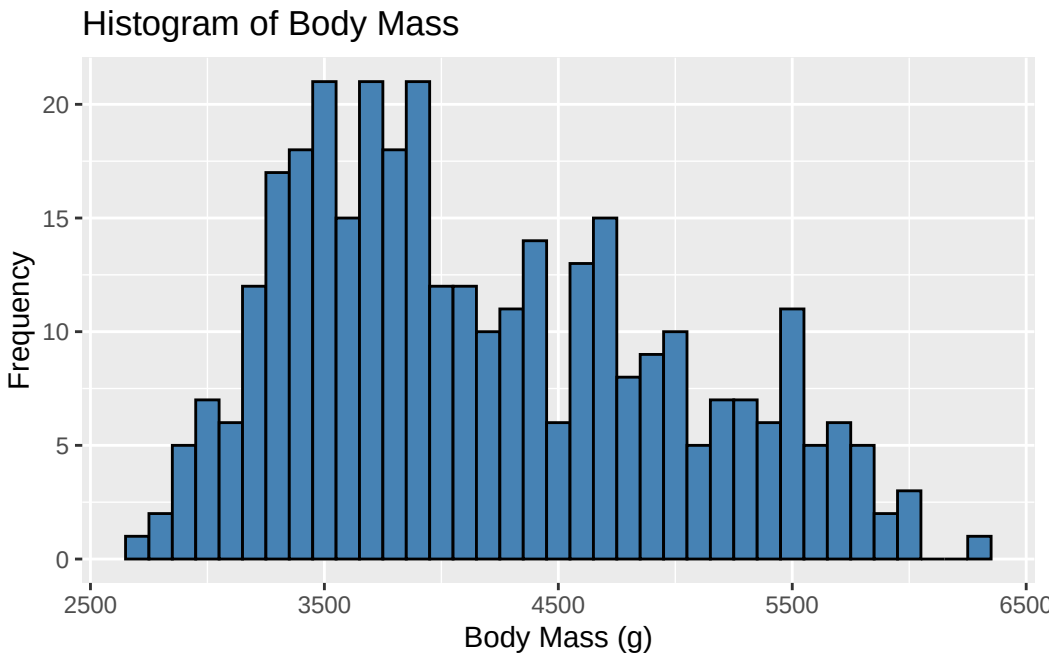

Figure 4: Histogram of Body Mass

0.7 Advanced Bar Chart

0.7.1 Grouped Bar Chart

```
# Sample data
data_grouped <- data %>% group_by(species, island) %>% summarise(mean_mass = mean(body_mass_g
```

```
`summarise()` has grouped output by 'species'. You can override using the
`.groups` argument.
```

```
# Grouped bar chart
ggplot(data_grouped, aes(x = species, y = mean_mass, fill = island)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Grouped Bar Chart", x = "Species", y = "Mean Body Mass (g)")
```

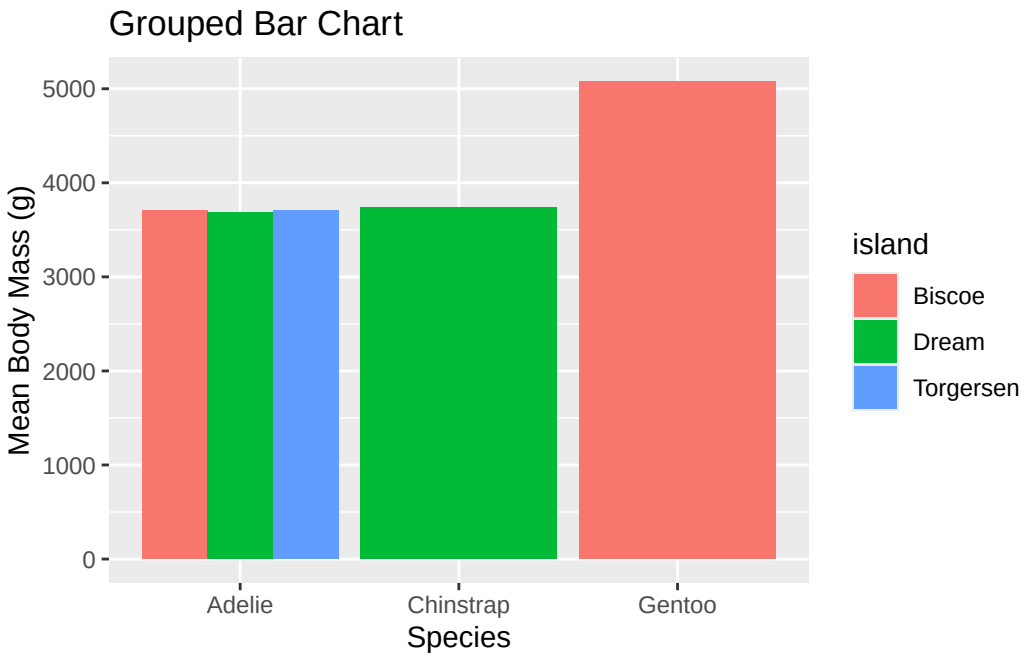

Figure 5: Grouped Bar Chart

0.8 Scatter Plot with Regression Line

```
# Scatter plot with regression line
ggplot(data, aes(x = body_mass_g, y = flipper_length_mm)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Scatter Plot with Regression Line", x = "Body Mass (g)", y = "Flipper Length
```

```
`geom_smooth()` using formula = 'y ~ x'
```

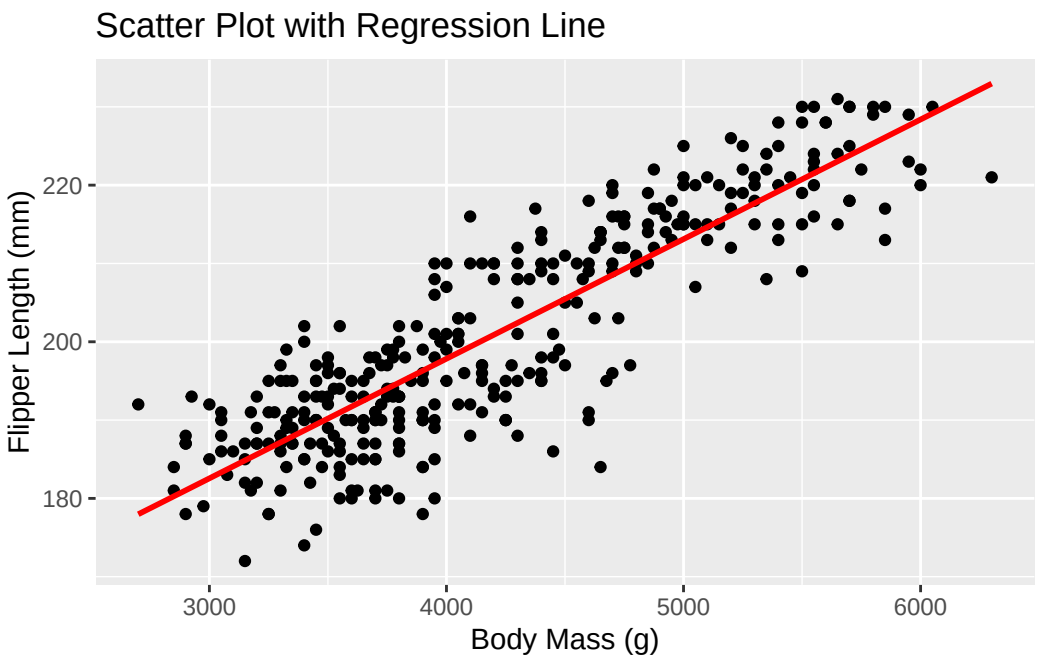

Figure 6: Scatter Plot with Regression Line

0.9 Faceted Plots

0.9.1 Faceted Scatter Plot

```
# Faceted scatter plot by species
ggplot(data, aes(x = body_mass_g, y = flipper_length_mm, color = species)) +
  geom_point() +
  facet_wrap(~species) +
  labs(title = "Faceted Scatter Plot by Species", x = "Body Mass (g)", y = "Flipper Length (
```

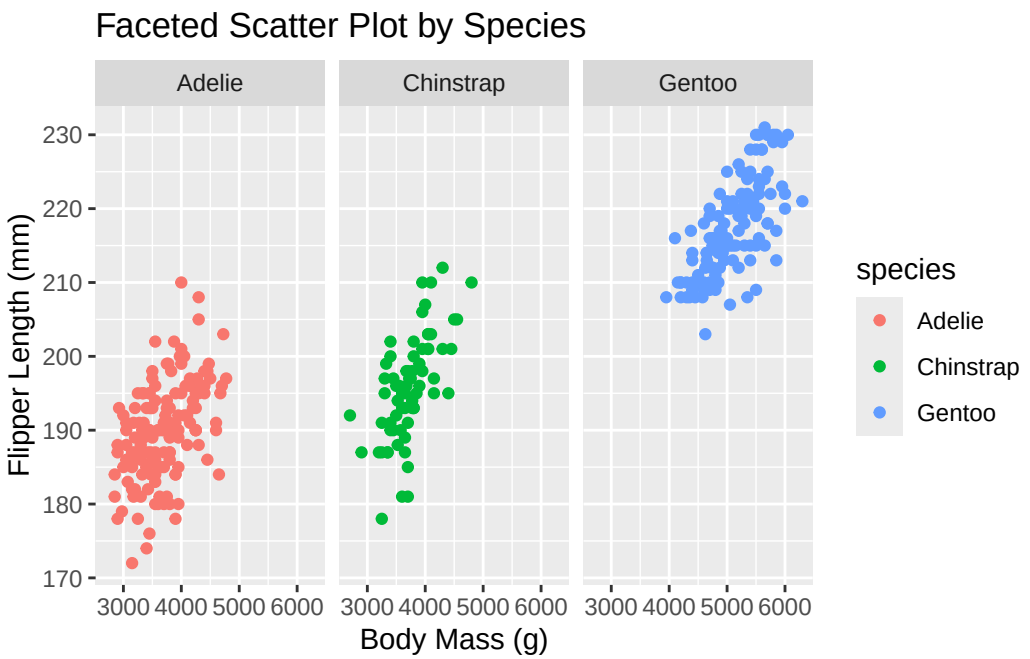

Figure 7: Faceted Scatter Plot by Species

0.10 Box Plot with Jitter

```
# Box plot with jittered points
ggplot(data, aes(x = species, y = body_mass_g, fill = species)) +
  geom_boxplot() +
  geom_jitter(width = 0.2, alpha = 0.5) +
  labs(title = "Box Plot with Jitter", x = "Species", y = "Body Mass (g)")
```

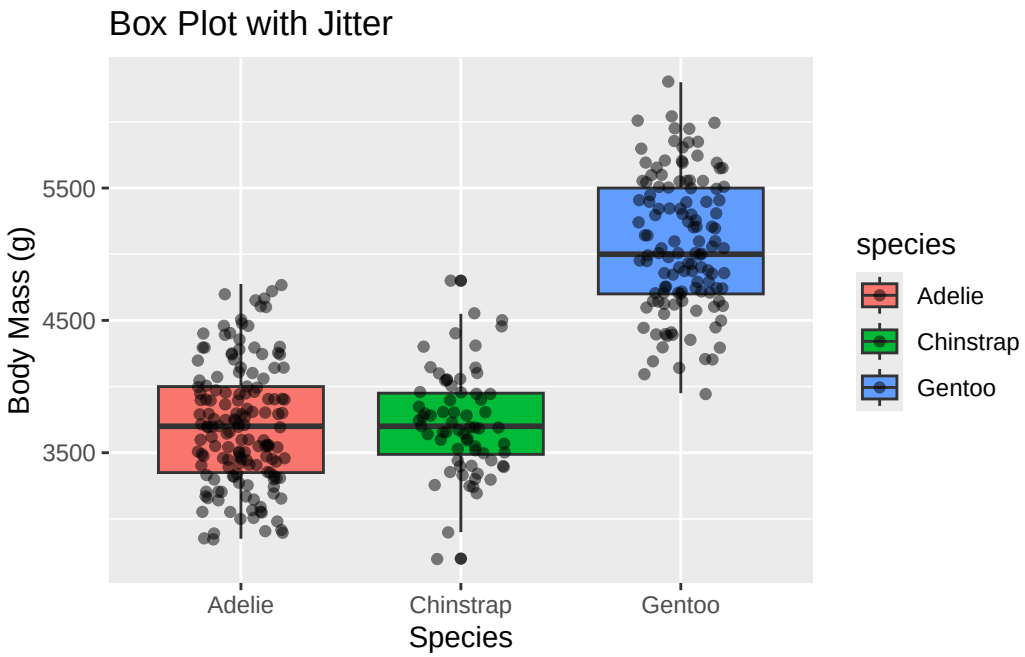

Figure 8: Box Plot with Jitter

0.11 Heatmap

```
# Create heatmap data
data_heatmap <- data %>% group_by(species, island) %>% summarise(mean_mass = mean(body_mass_g
```

```
`summarise()` has grouped output by 'species'. You can override using the
`.groups` argument.
```

```
# Heatmap
ggplot(data_heatmap, aes(x = species, y = island, fill = mean_mass)) +
  geom_tile() +
  labs(title = "Heatmap of Mean Body Mass", x = "Species", y = "Island")
```

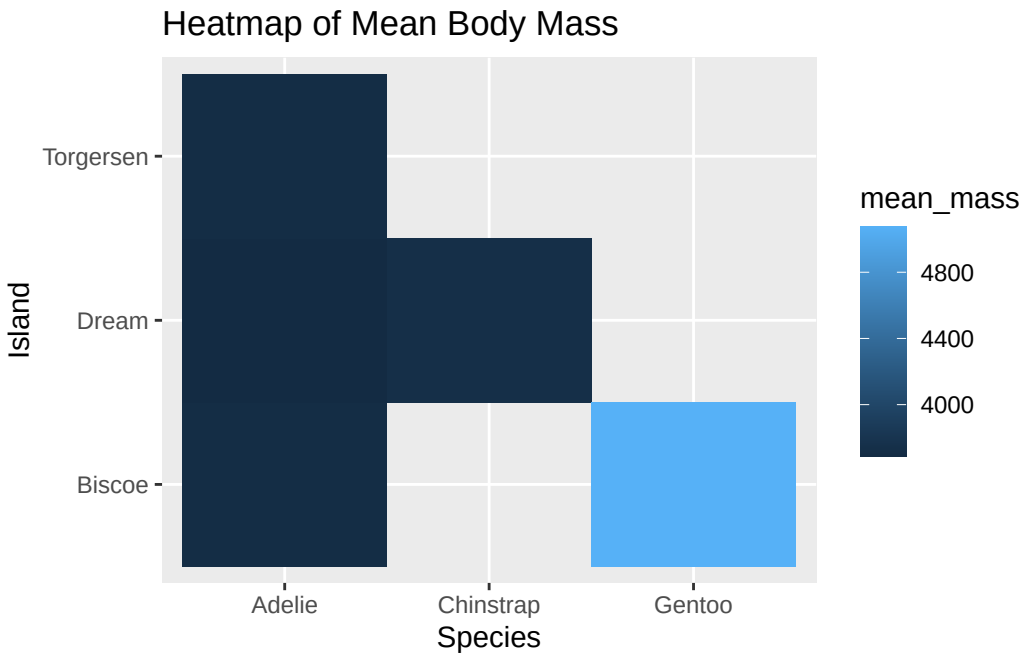

Figure 9: Heatmap of Mean Body Mass

0.12 Radar Chart

0.12.1 Using Reshaped Data

```
# Create radar chart data
data_radar <- data %>%
  group_by(species) %>%
  summarise(across(where(is.numeric), mean, na.rm = TRUE)) %>%
  column_to_rownames(var = "species")
data_radar <- rbind(max = apply(data_radar, 2, max), min = apply(data_radar, 2, min), data_ra

# Radar chart
radarchart(data_radar, axistype = 1, title = "Radar Chart Example")
```

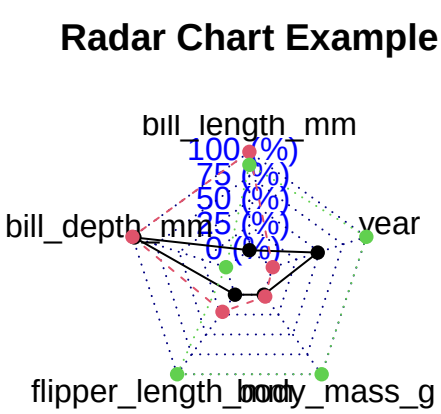

Figure 10: Radar Chart Example

0.13 Conclusion

This tutorial provided an overview of various visualization techniques in R, from basic plots to advanced visualizations. By applying these methods, you can explore, analyze, and communicate data effectively.