Data Visualization Analysis

Sofia

2024-12-18

Table of Contents

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.

Preparation

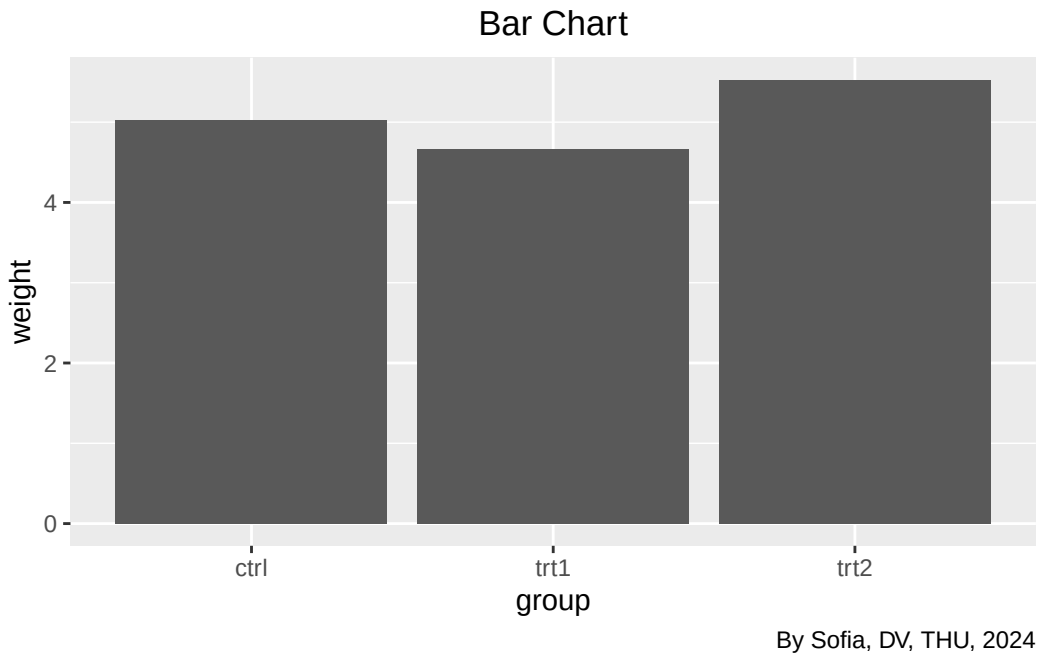
In order to draw a chart, we need to include the required packages for visualization and dataset. For example, the `ggplot2` package is for drawing charts, and the `gcookbook` package is for using the `pg_mean` dataset.

Bar Chart

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

group	weight
ctrl	5.032
trt1	4.661
trt2	5.526

```
ggplot(pg_mean, aes(x = group, y = weight)) +
  geom_col() +
  labs(title = 'Bar Chart', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```

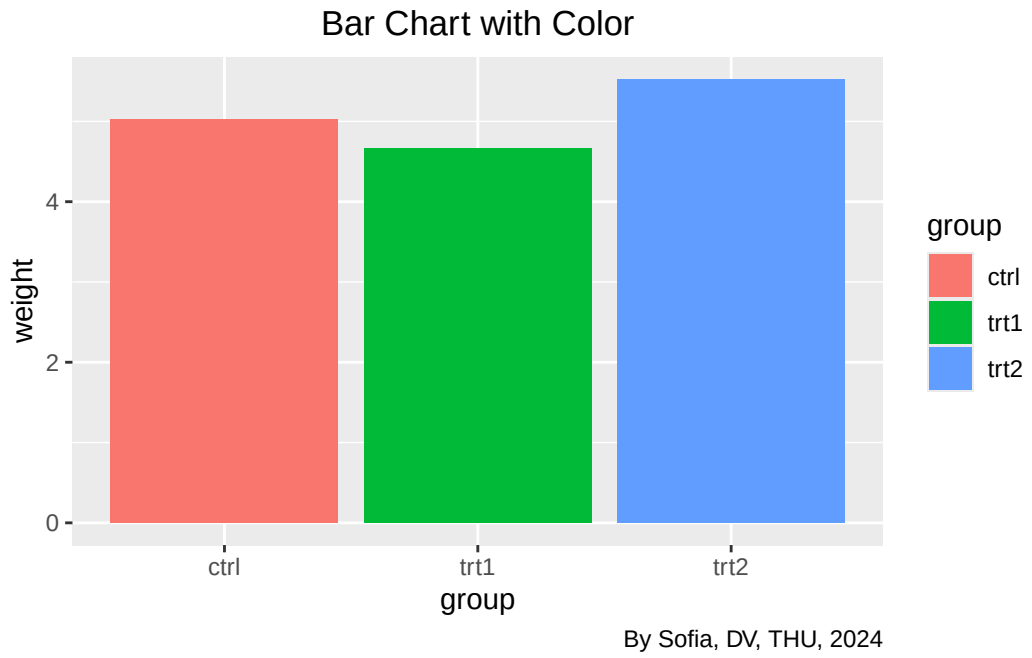


- `aes(x = group, y = weight)` specifies the aesthetics:
  - `x = group`: Assign the group variable to the x-axis.
  - `y = weight`: Assign the weight variable to the y-axis.
- `geom_col()`: Adds a column geometry to the plot.

Bar Chart with Color

This bar chart includes colors for each group to make the data more visually distinct.

```
ggplot(pg_mean, aes(x = group, y = weight, fill = group)) +
  geom_col() +
  labs(title = 'Bar Chart with Color', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```



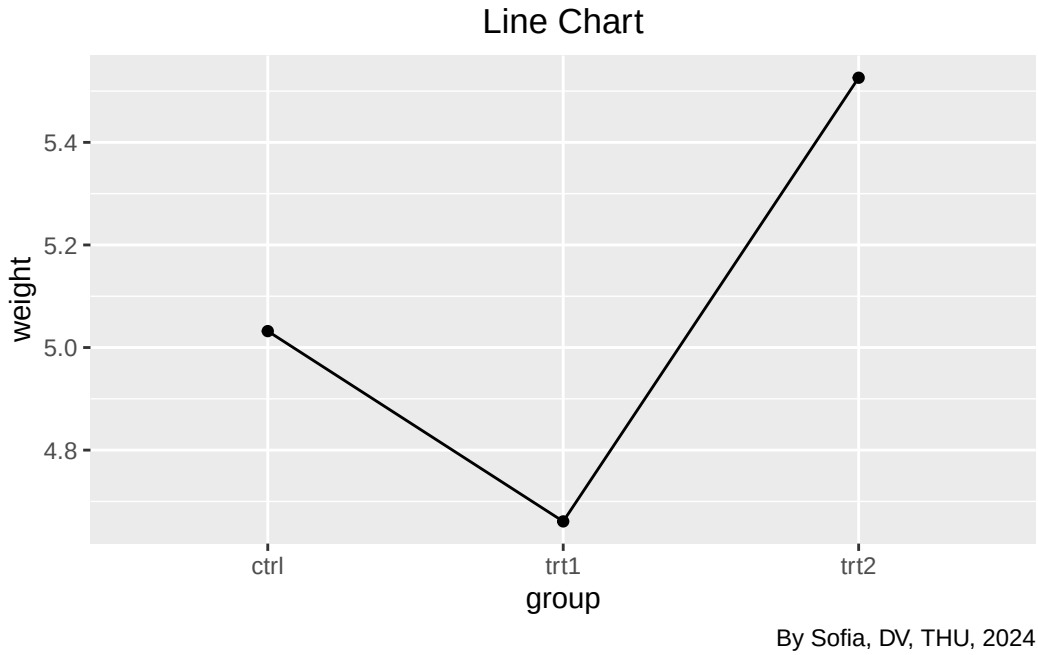
- `aes(x = group, y = weight, fill = group)`: Maps the group variable to the x-axis, the weight variable to the y-axis, and colors the bars based on the group.
- `geom_col()`: Creates a bar chart where the height of the bars represents the weight values.
- `labs()`: Adds a title and caption to the chart.

- `theme()`: Centers the title on the chart.

Line Chart

A line chart is used to show trends over time or categories.

```
library(ggplot2)
ggplot(pg_mean, aes(x = group, y = weight, group = 1)) +
  geom_line() +
  geom_point() +
  labs(title = 'Line Chart', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```



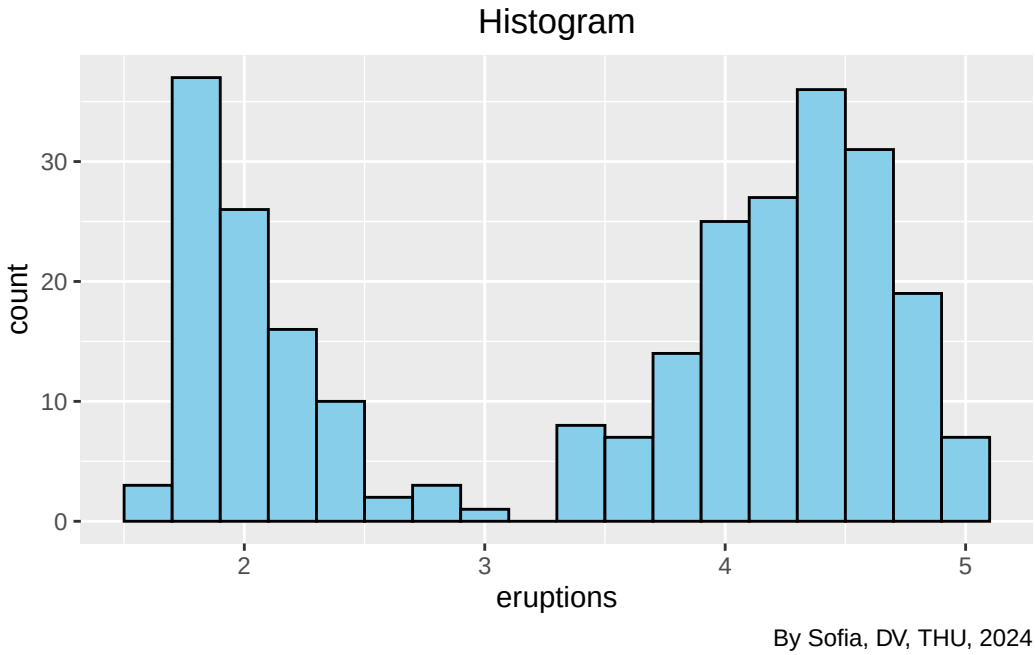
- `aes(x = group, y = weight, group = 1)`: This sets the group variable on the x-axis and the weight variable on the y-axis. The group = 1 ensures that all points are connected by a single line.
- `geom_line()`: Adds a line connecting the points to show the trend.
- `geom_point()`: Adds points to the line chart to highlight individual data values.
- `labs(title = 'Line Chart', caption = 'By Sofia, DV, THU, 2024')`: Adds a title and caption to the chart for clarity.

- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Histogram

A histogram shows the distribution of a numerical variable.

```
ggplot(faithful, aes(x = eruptions)) +
  geom_histogram(binwidth = 0.2, fill = "skyblue", color = "black") +
    labs(title = 'Histogram', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```



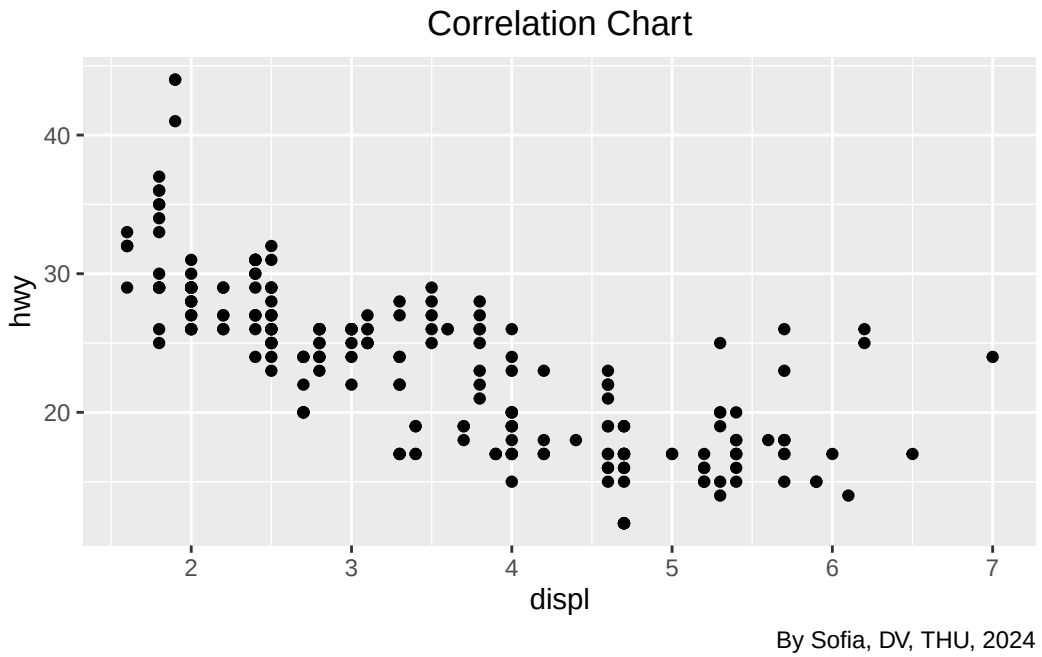
- `aes(x = eruptions)`: This sets the eruptions variable on the x-axis to display its distribution.
- `geom_histogram(binwidth = 0.2, fill = "skyblue", color = "black")`: Creates the histogram with bars representing the frequency of eruptions. The binwidth = 0.2 controls the width of each bar, and fill = "skyblue" and color = "black" set the bar color and the border color, respectively.
- `labs(title = 'Histogram', caption = 'By Sofia, DV, THU, 2024')`: Adds a title and caption to the chart for context.

- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Correlation Chart

Correlation charts show the relationship between two numerical variables.

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
    labs(title = 'Correlation Chart', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```

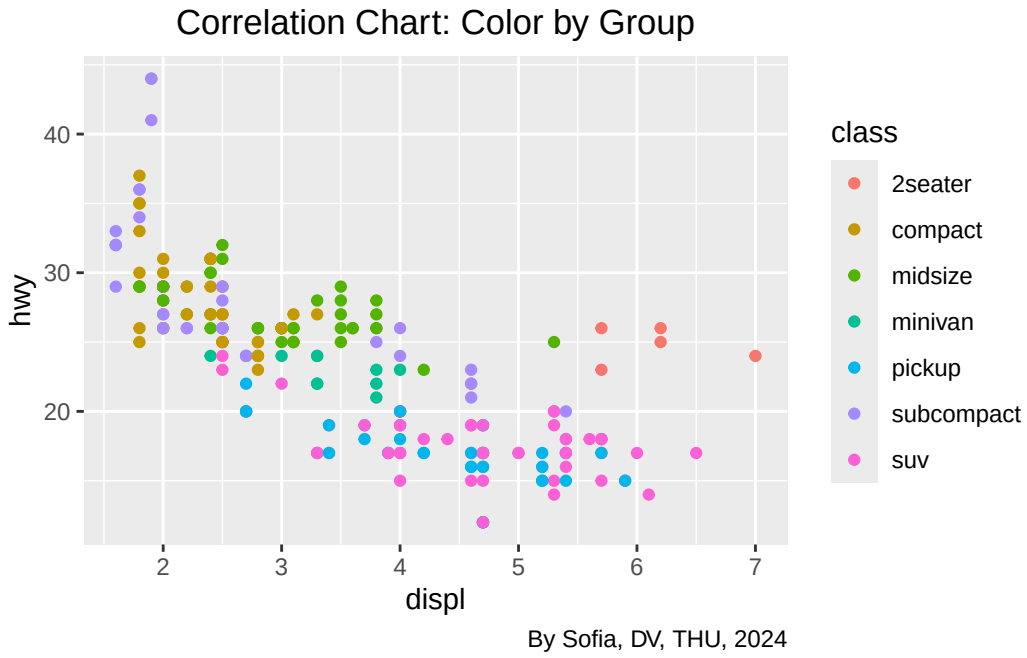


- `aes(x = displ, y = hwy)`: This sets the displ variable (engine displacement) on the `x-axis` and the `hwy` variable (highway miles per gallon) on the `y-axis`.
- `geom_point()`: Adds points to the chart to represent the relationship between `displ` and `hwy`. Each point shows the values of both variables for a specific car.
- `labs(title = 'Correlation Chart', caption = 'By Sofia, DV, THU, 2024')`: Adds a title and caption to the chart for context.
- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Correlation Chart: Color by Group

We can enhance the correlation chart by coloring the points based on a group variable.

```
ggplot(mpg, aes(x = displ, y = hwy, color = class)) +
  geom_point() +
  labs(title = 'Correlation Chart: Color by Group', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```

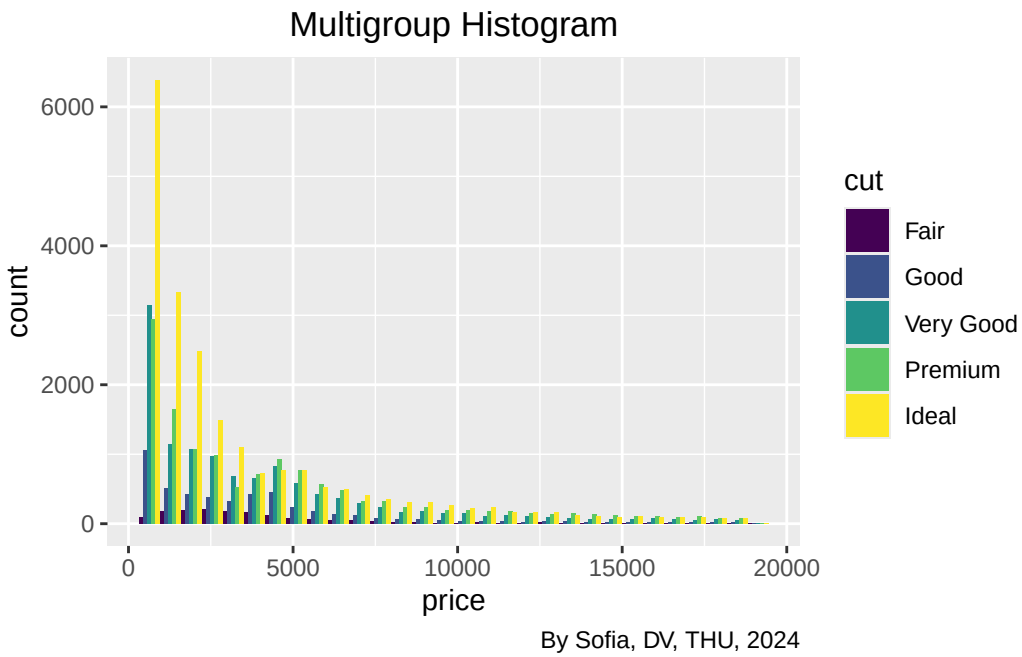


- `aes(x = displ, y = hwy, color = class)`: This sets the `displ` variable on the `x-axis` and the `hwy` variable on the `y-axis`. Additionally, the `color = class` maps the `class` variable to different colors for each group of cars.
- `geom_point()`: Adds points to the chart, where each point represents a car, and the color indicates which class the car belongs to.
- `labs(title = 'Correlation Chart: Color by Group', caption = 'By Sofia, DV, THU, 2024')`: Adds a title and caption to the chart for context.
- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Multigroup Histogram

A multigroup histogram displays distributions for multiple groups side by side.

```
ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_histogram(position = "dodge", bins = 30) +
  labs(title = 'Multigroup Histogram', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```

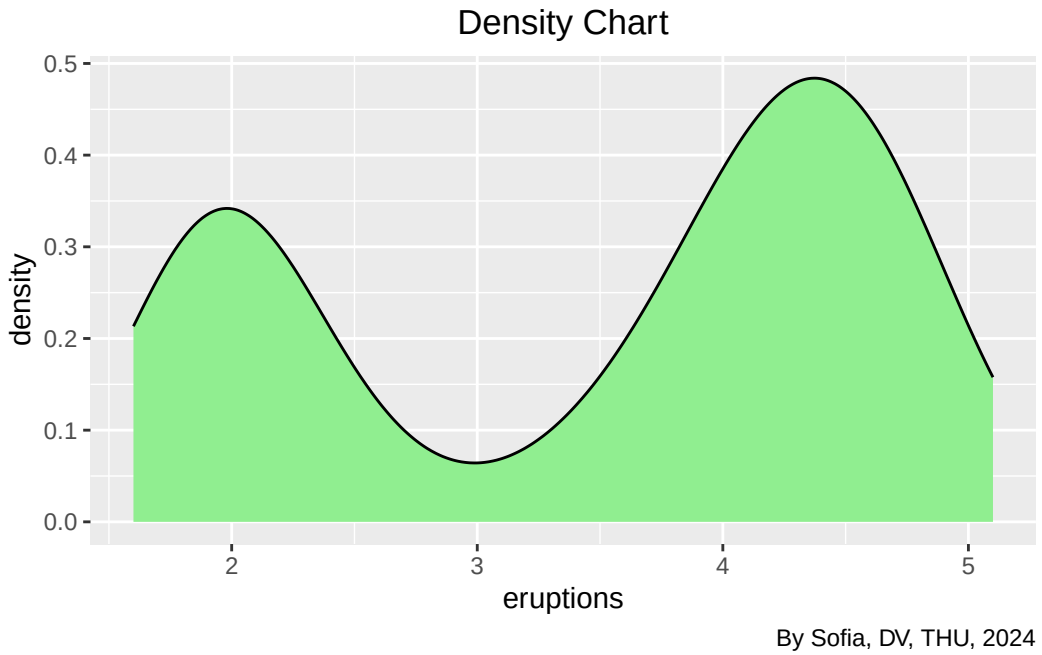


- `aes(x = price, fill = cut)`: This sets the `price` variable on the `x-axis` and uses the `cut` variable to fill the bars with different colors. Each color represents a different diamond cut.
- `geom_histogram(position = "dodge", bins = 30)`: Creates the histogram, with bars positioned side by side (position = "dodge") and 30 bins. Each bin represents a range of price values.
- `labs(title = 'Multigroup Histogram', caption = 'By Sofia, DV, THU, 2024')`: Adds a title and caption to the chart for context.
- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Density Chart

A density chart shows the distribution of a continuous variable, smoothed by a density function.

```
ggplot(faithful, aes(x = eruptions)) +
  geom_density(fill = "lightgreen") +
   labs(title = 'Density Chart', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```



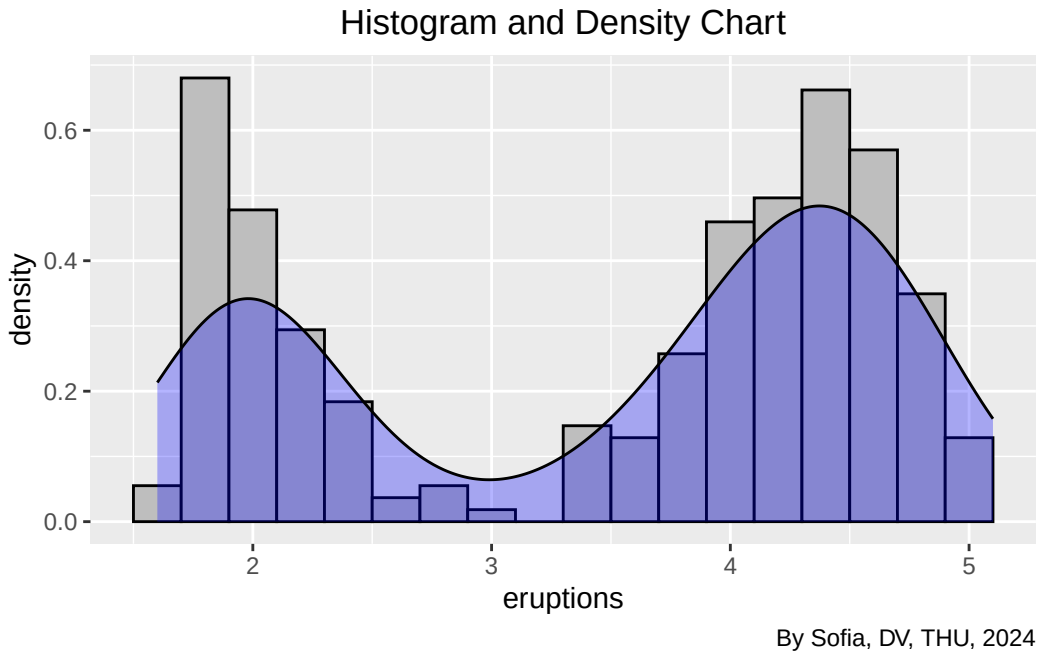
- `aes(x = eruptions)`: This sets the `eruptions` variable on the `x-axis` to display its distribution.
- `geom_density(fill = "lightgreen")`: Creates the density chart with a smoothed curve representing the distribution of eruptions. The area under the curve is filled with a light green color.
- `labs(title = 'Density Chart', caption = 'By Sofia, DV, THU, 2024')`: Adds a title and caption to the chart for context.
- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Histogram and Density Chart

We can combine a histogram with a density chart to compare both visualizations.

```
ggplot(faithful, aes(x = eruptions)) +
  geom_histogram(aes(y = ..density..), binwidth = 0.2, fill = "gray", color = "black") +
  geom_density(fill = "blue", alpha = 0.3) +
   labs(title = 'Histogram and Density Chart', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```

```
Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
i Please use `after_stat(density)` instead.
```



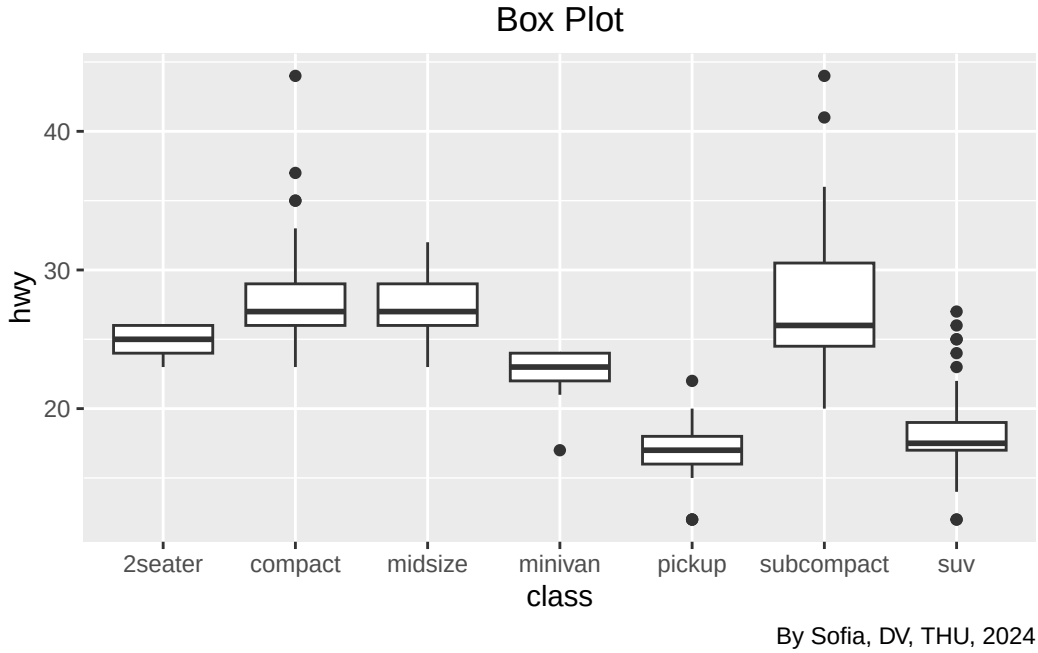
- `aes(x = eruptions)`: This sets the `eruptions` variable on the `x-axis` to show its distribution.
- `geom_histogram(aes(y = ..density..), binwidth = 0.2, fill = "gray", color = "black")`: Creates a histogram where the `y-axis` represents density instead of frequency. The binwidth = 0.2 controls the width of each bin, and the bars are filled with a gray color and outlined with black.
- `geom_density(fill = "blue", alpha = 0.3)`: Adds a density curve on top of the histogram, filled with a blue color and slightly transparent (alpha = 0.3) to allow both visuals to be seen clearly.

- `labs(title = 'Histogram and Density Chart', caption = 'By Sofia, DV, THU, 2024')`: Adds a title and caption to the chart for context.
- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Box Plot

A box plot shows the distribution and identifies outliers within the data.

```
ggplot(mpg, aes(x = class, y = hwy)) +
  geom_boxplot() +
   labs(title = 'Box Plot', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```



- `aes(x = class, y = hwy)`: This sets the `class` variable on the `x-axis` and the `hwy` variable (highway miles per gallon) on the `y-axis`.
- `geom_boxplot()`: Creates a box plot, which shows the distribution of `hwy` values for each class. The plot displays the median, upper and lower quartiles, and any outliers.
- `labs(title = 'Box Plot', caption = 'By Sofia, DV, THU, 2024')`:‘ Adds a title and caption to the chart for context.

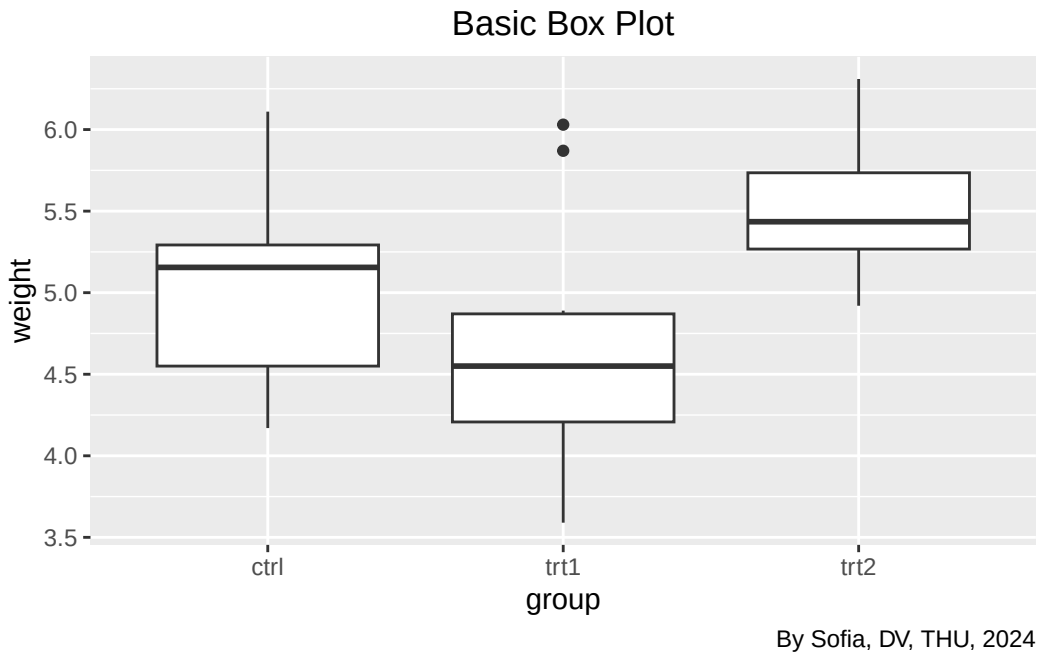
- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Basic Box Plot

This example illustrates a basic box plot for visualizing the distribution of weights across different groups in the `PlantGrowth` dataset.

```
library(ggplot2)

ggplot(PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot() +
  labs(title = 'Basic Box Plot', caption = 'By Sofia, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5))
```



- `aes(x = group, y = weight)`: This sets the `group` variable on the `x-axis` and the `weight` variable on the `y-axis` to show how the distribution of weight differs between groups.
- `geom_boxplot()`: Creates a box plot that displays the distribution of `weight` for each group. It shows the median, quartiles, and potential outliers for each group.

- `labs(title = 'Basic Box Plot', caption = 'By Sofia, DV, THU, 2024')`: Adds a title and caption to the chart for context.
- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

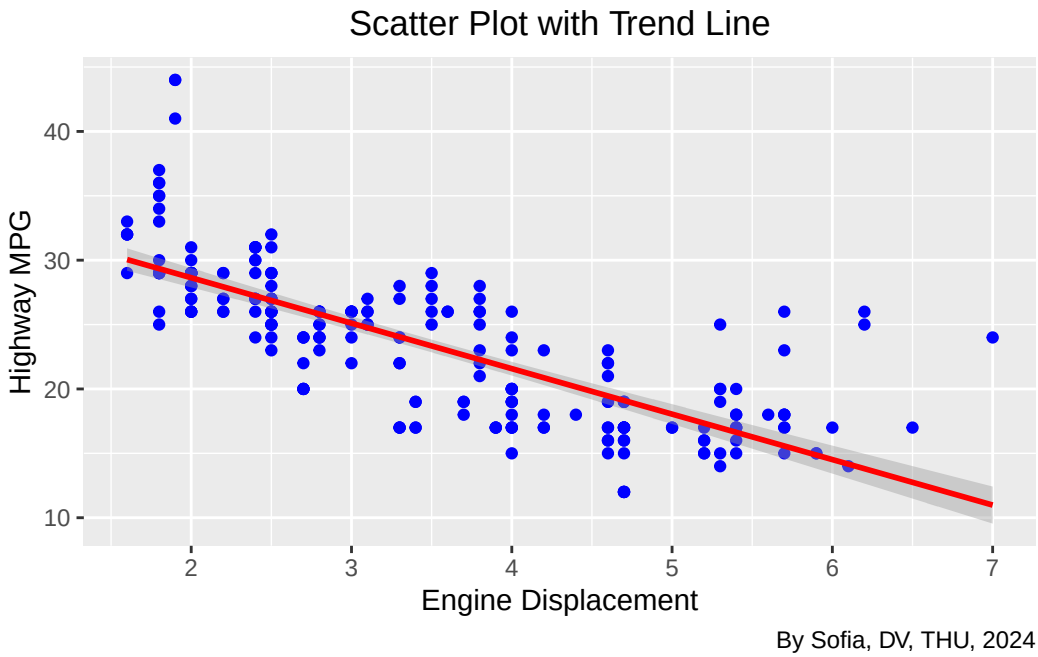
Scatter Plot with Trend Line

A scatter plot with a trend line helps visualize the relationship between two variables and highlight trends.

```
library(ggplot2)

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Scatter Plot with Trend Line",
       x = "Engine Displacement", y = "Highway MPG",
       caption = "By Sofia, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5))
```

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



- `aes(x = displ, y = hwy)`: This sets the `displ` (engine displacement) variable on the `x-axis` and the `hwy` (highway miles per gallon) variable on the `y-axis`.
- `geom_point(color = "blue")`: Creates a scatter plot with blue points to show the individual data points for each combination of `displ` and `hwy`.
- `geom_smooth(method = "lm", color = "red")`: Adds a red trend line using linear regression (method = "lm") to show the overall relationship between `displ` and `hwy`.
- `labs(title = "Scatter Plot with Trend Line", x = "Engine Displacement", y = "Highway MPG", caption = "By Sofia, DV, THU, 2024")`: Adds a title, axis labels, and a caption to the chart for context.

Custom Grid Plot

This example shows how to create customized subplots using a facet grid, visualizing two variables split by a condition.

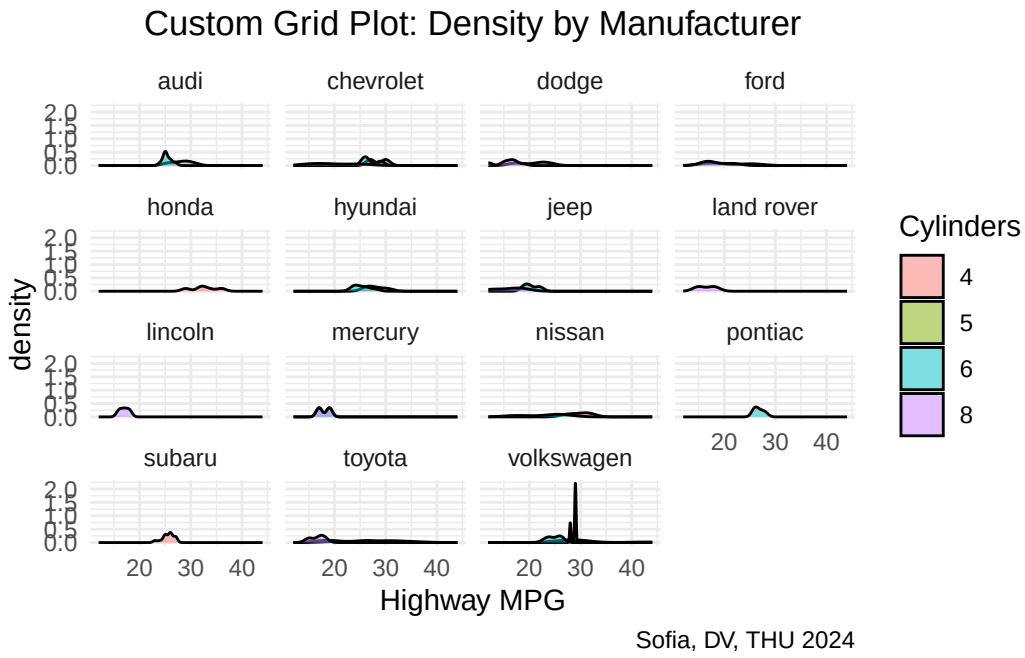
```
library(ggplot2)

ggplot(mpg, aes(x = hwy, fill = factor(cyl))) +
  geom_density(alpha = 0.5) +
  facet_wrap(~ manufacturer) +
```

```
  labs(title = "Custom Grid Plot: Density by Manufacturer",
       x = "Highway MPG", fill = "Cylinders",
       caption = "Sofia, DV, THU 2024") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

```
Warning: Groups with fewer than two data points have been dropped.
Groups with fewer than two data points have been dropped.
Groups with fewer than two data points have been dropped.
Groups with fewer than two data points have been dropped.

Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
-Inf
Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
-Inf
Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
-Inf
Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
-Inf
```



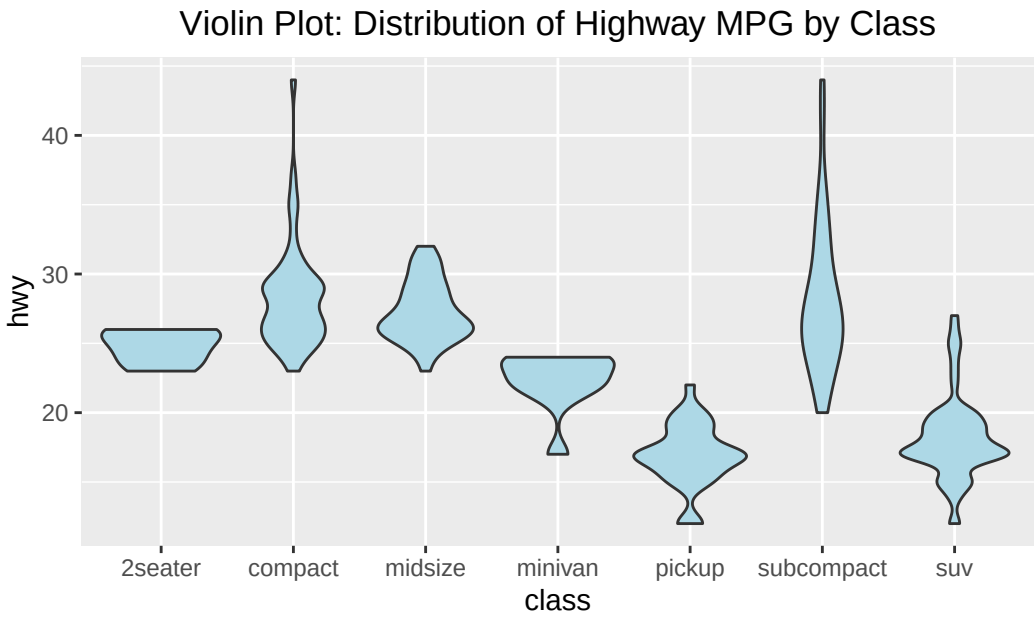
- `aes(x = hwy, fill = factor(cyl))`: This sets the `hwy` (highway miles per gallon) variable on the `x-axis` and uses the `cyl` (number of cylinders) variable to fill the density plots with different colors based on the number of cylinders.

- `geom_density(alpha = 0.5)`: Creates density plots with a transparency level of 0.5 (alpha = 0.5) so that overlapping areas can be seen more clearly.
- `facet_wrap(~ manufacturer)`: Splits the plot into subplots for each manufacturer, creating a grid of plots, one for each manufacturer in the dataset.
- `labs(title = "Custom Grid Plot: Density by Manufacturer", x = "Highway MPG", fill = "Cylinders", caption = "Sofia, DV, THU 2024")`: Adds a title, axis labels, and a caption to the chart. theme_minimal(): Applies a minimal theme to the plot, removing unnecessary elements for a cleaner look.
- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Violin Plot

A violin plot combines aspects of box plots and density plots to show the distribution of a variable.

```
ggplot(mpg, aes(x = class, y = hwy)) +
  geom_violin(fill = "lightblue") +
  labs(title = "Violin Plot: Distribution of Highway MPG by Class",
       caption = "By Sofia, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5))
```

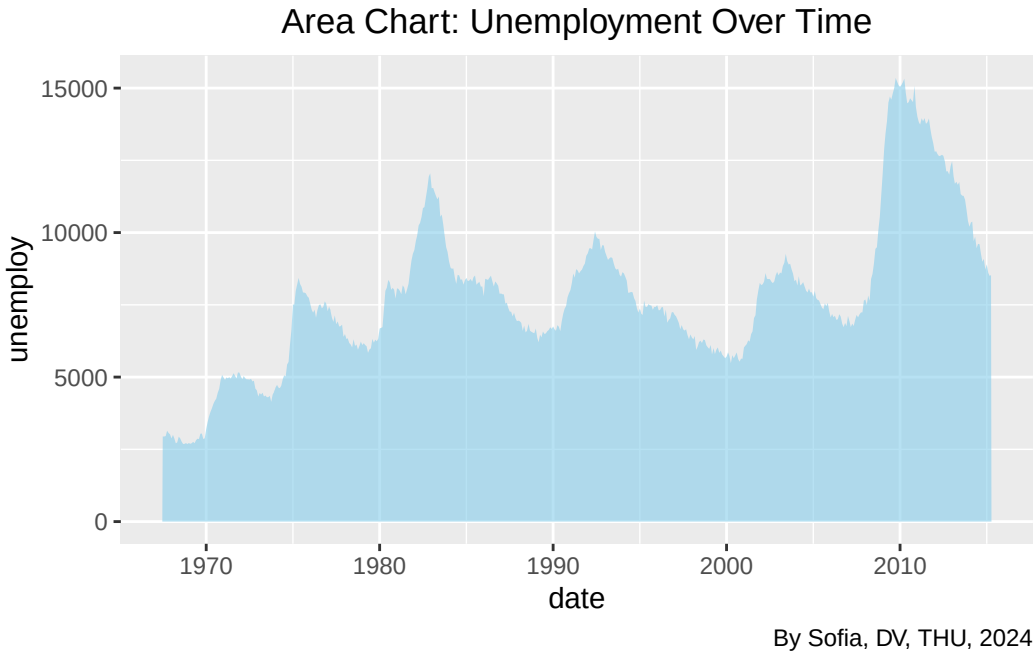


- `aes(x = class, y = hwy)`: This sets the `class` variable on the `x-axis` and the `hwy` (highway miles per gallon) variable on the `y-axis`.
- `geom_violin(fill = "lightblue")`: Creates a violin plot with the data, where the shape represents the density of `hwy` values for each `class`, filled with a light blue color.
- `labs(title = "Violin Plot: Distribution of Highway MPG by Class", caption = "By Sofia, DV, THU, 2024")`: Adds a title and caption to the plot.
- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Area Chart

An area chart is similar to a line chart but with the area below the line filled in to emphasize the magnitude of values.

```
ggplot(economics, aes(x = date, y = unemploy)) +
  geom_area(fill = "skyblue", alpha = 0.6) +
  labs(title = "Area Chart: Unemployment Over Time",
       caption = "By Sofia, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5))
```

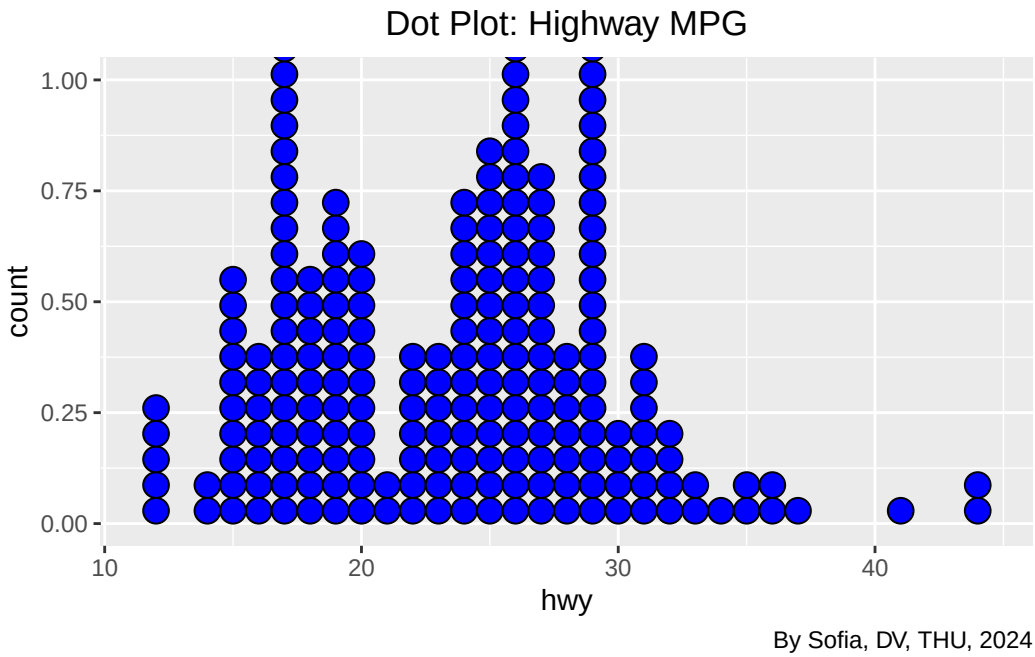


- `aes(x = date, y = unemploy)`: This sets the date variable on the x-axis and the unemploy (unemployment rate) variable on the y-axis.
- `geom_area(fill = "skyblue", alpha = 0.6)`: Creates an area chart where the area under the line is filled with a sky blue color and transparency (alpha = 0.6) to make the chart visually appealing.
- `labs(title = "Area Chart: Unemployment Over Time", caption = "By Sofia, DV, THU, 2024")`: Adds a title and caption to the plot.
- `theme(plot.title = element_text(hjust = 0.5))`: Centers the title at the top of the chart.

Dot Plot

A dot plot is a simple way to represent individual data points on a single axis.

```
ggplot(mpg, aes(x = hwy)) +
  geom_dotplot(binwidth = 1, fill = "blue", color = "black") +
  labs(title = "Dot Plot: Highway MPG", caption = "By Sofia, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5))
```



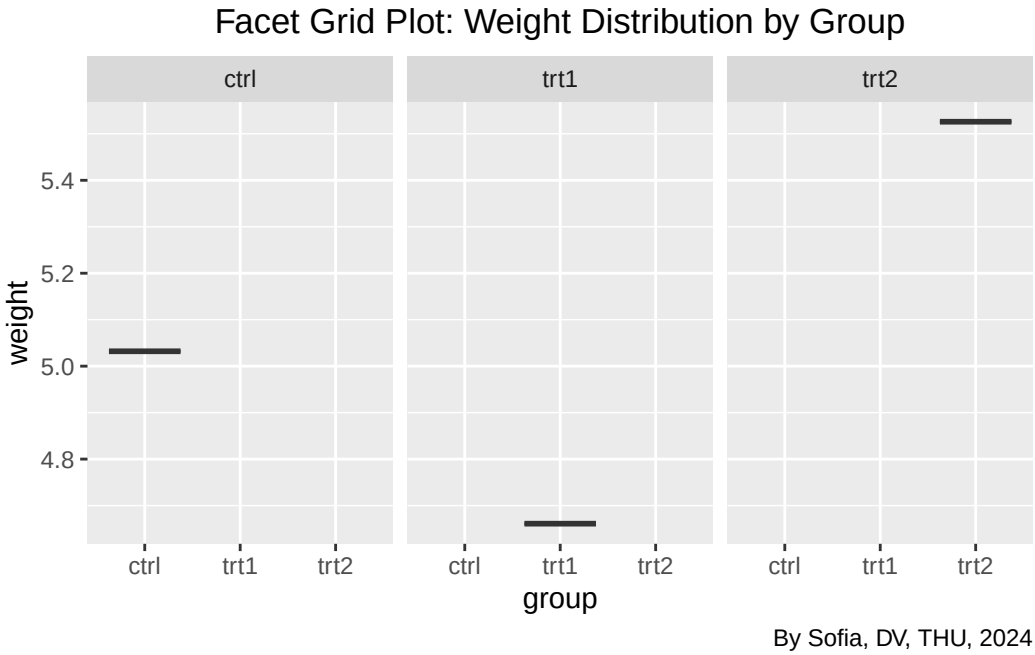
- `geom_dotplot()`: Creates a dot plot where each data point is represented by a dot.
- `binwidth = 1`: Sets the bin width for the dot plot, controlling how closely packed the dots are.
- `labs()`: Adds a title and caption to the plot.

Facet Grid Plot

Facet grids are used to create subplots for visualizing data split by a certain variable.

```
library(ggplot2)
library(gcookbook)

# Facet grid to show weight by group in pg_mean dataset
ggplot(pg_mean, aes(x = group, y = weight)) +
  geom_boxplot() +
  facet_grid(~ group) +
  labs(title = "Facet Grid Plot: Weight Distribution by Group", caption = "By Sofia, DV, THU
  theme(plot.title = element_text(hjust = 0.5))
```



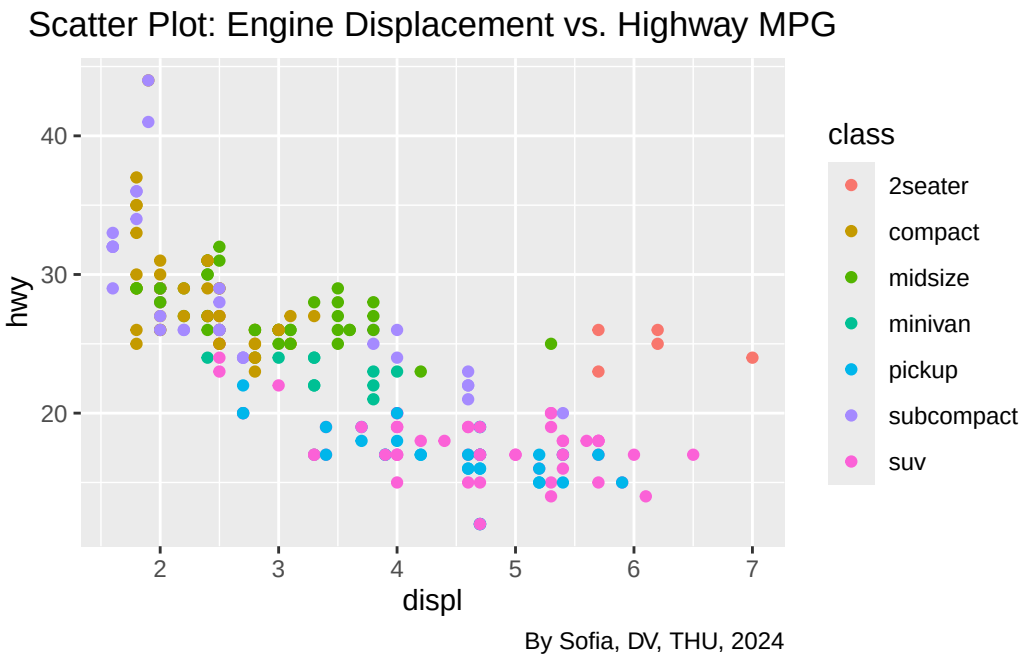
- `acet_grid(~ group)`: The facet_grid() function splits the plot into separate panels, each showing the data for one level of the group variable. This helps us see how weight distributions differ between the groups in the dataset.
- `geom_boxplot()`: This creates a box plot for each group, showing the distribution of the weight variable. The box plot provides a summary of the data's distribution, highlighting the median, quartiles, and any potential outliers.
- `labs()`: The labs() function is used to add a title and caption to the plot. The title provides context for what the plot is showing, and the caption gives credit for the plot's creation.
- `theme(plot.title = element_text(hjust = 0.5))`: The theme() function customizes the plot's appearance, and hjust = 0.5 centers the title horizontally on the plot for better readability.

Scatter Plot with Aesthetic Mappings

A scatter plot is used to visualize the relationship between two continuous variables. In this example, we also use aesthetic mappings to color the data points based on a third categorical variable.

```r
library(ggplot2)

# Create a scatter plot using the mpg dataset and map color to class
ggplot(mpg, aes(x = displ, y = hwy, color = class)) +
  geom_point() +
  labs(title = "Scatter Plot: Engine Displacement vs. Highway MPG", caption = "By Sofia, DV,
  theme(plot.title = element_text(hjust = 0.5))
```



- `color = class`: The color aesthetic is used to differentiate the data points based on the class variable. Each point will be colored according to its class, allowing for easy comparison of how the different classes of vehicles relate to engine displacement (displ) and highway miles per gallon (hwy).
- `geom_point()`: The geom_point() function is used to create the scatter plot. Each point represents a vehicle, with its position determined by its engine displacement and highway miles per gallon.
- `labs()`: The labs() function adds a title and caption to the plot. The title summarizes what the plot is showing, while the caption credits the plot's creation.