Data Visualization Analysis

David

2024-12-02

Table of contents

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.

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 and `gcookbook` is for using `pg_mean` dataset, but in the next section we will use more dataset from `gcookbook`.

```
library(ggplot2)
library(gcookbook)
```

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).

Below graph initializes a ggplot with the dataset `pg_mean`.

`aes(x = group, y = weight)` specifies the aesthetics:

- `x = group`: Assign the `group` variable to the x-axis (categorical data, such as `ctrl, trt1, trt2`).
- `y = weight`: Assign the `weight` variable to the y-axis (numerical data).

`geom_col()`:

- Adds a column geometry to the plot.
- `geom_col()` creates bars where the height of each bar corresponds to the value of weight for each group.

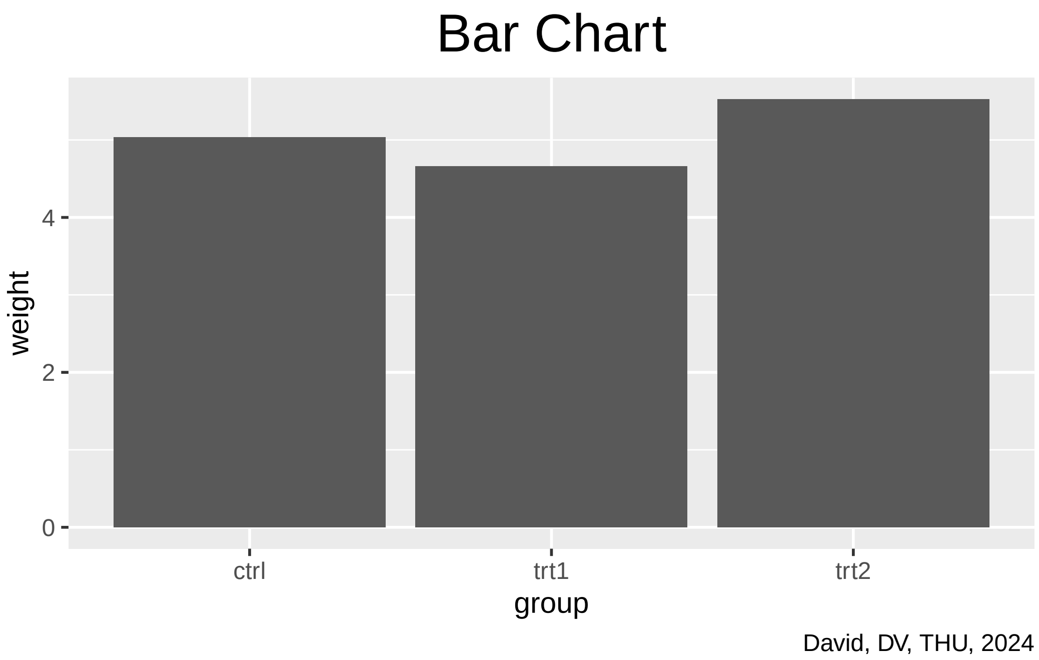
`labs()` function in `ggplot2` is used to customize labels in a plot.

- `title`: To specifies the main title of the plot.
- `caption`: To Add a additional information.
- `x`: To change x label name.
- `y`: To change y label name.

`theme()` function in `ggplot2` is used to customize the appearance of a plot.

- `plot.title`: To control the main title.
- `element_text()`: To customize the appearance of text elements in a plot.
  - `hjust`: To adjust the title position.
  - `size`: To control size of the title.

```r
ggplot(pg_mean, aes(x = group, y = weight)) +
  geom_col() +
  labs(title = "Bar Chart", caption = 'David, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5, size = 20))
```

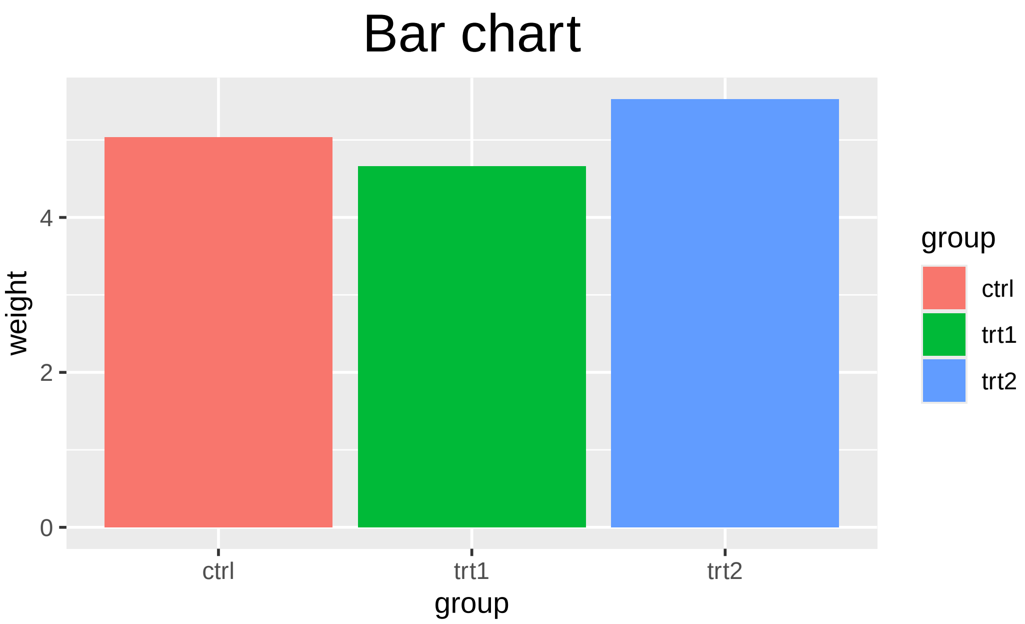


4 Bar chart with color

To add color in the bar chart you can add a function `fill` in the `ggplot()` function.

`fill = group`: Assign to color the bar of the `group` bar.

```
ggplot(pg_mean, aes(x = group, y = weight, fill = group)) +
  geom_col() +
  labs(title = "Bar chart",
       caption = "David, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))
```



David, DV, THU, 2024

5 Line chart

In this section, we will draw a line chart using `BOD` dataset. The dataset has two columns: `Time`, `demand`

```
BOD
```

```
  Time demand
1    1    8.3
2    2   10.3
3    3   19.0
4    4   16.0
5    5   15.6
6    7   19.8
```

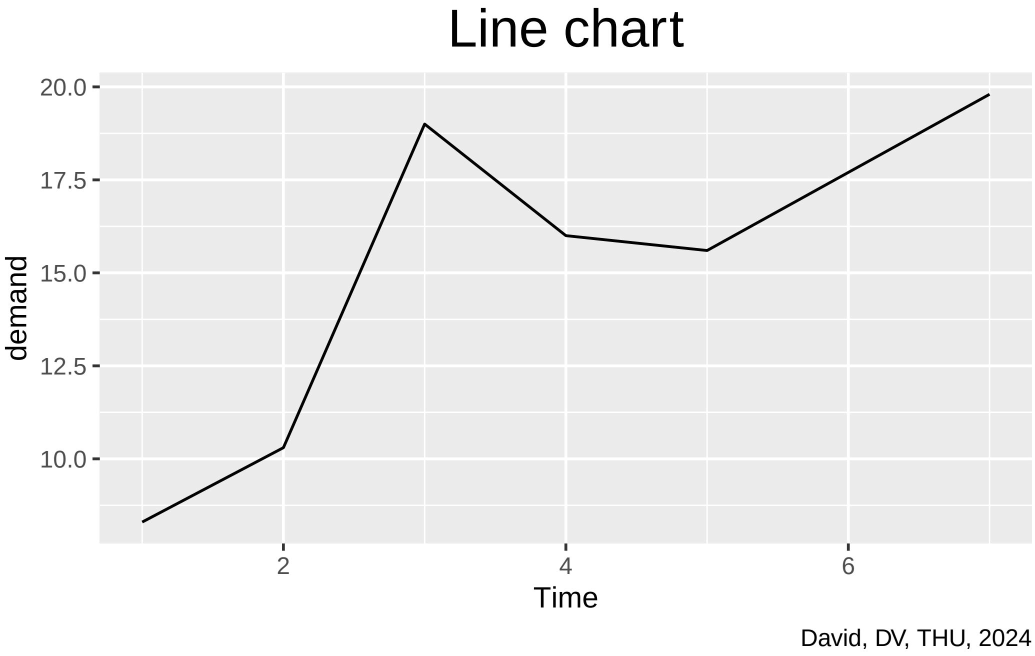
This dataset compares the data across two groups:

- `Time`: Represents the time in days at which the biochemical oxygen demand is measured.
- `demand`: Represents the biochemical oxygen demand (BOD) in milligrams per liter (mg/L).

`geom_line()`:

- Used to add lines to a plot, typically to visualize trends or relationships between data points in sequential or continuous data.
- It connects data points in the order of their x-values.

```
ggplot(BOD, aes(x = Time, y = demand)) +
  geom_line() +
  labs(title = "Line chart",
       caption = "David, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))
```



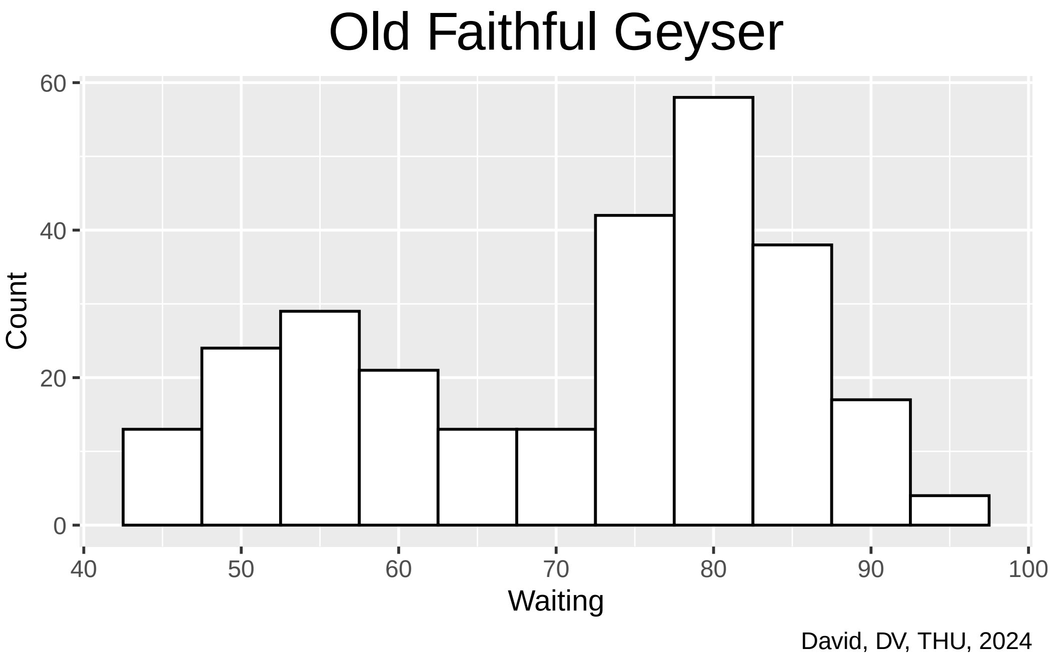
6 Histogram

In this section, we will draw a histogram using `faithful` dataset. Our goal is to count `waiting` in the dataset.

`geom_histogram()`: To visualize the frequency distribution of a continuous variable.

- `binwidth`: To set the range of one bar.
- `fill`: To give a color to the bar.
- `colour`: To give color to the border of the bar.

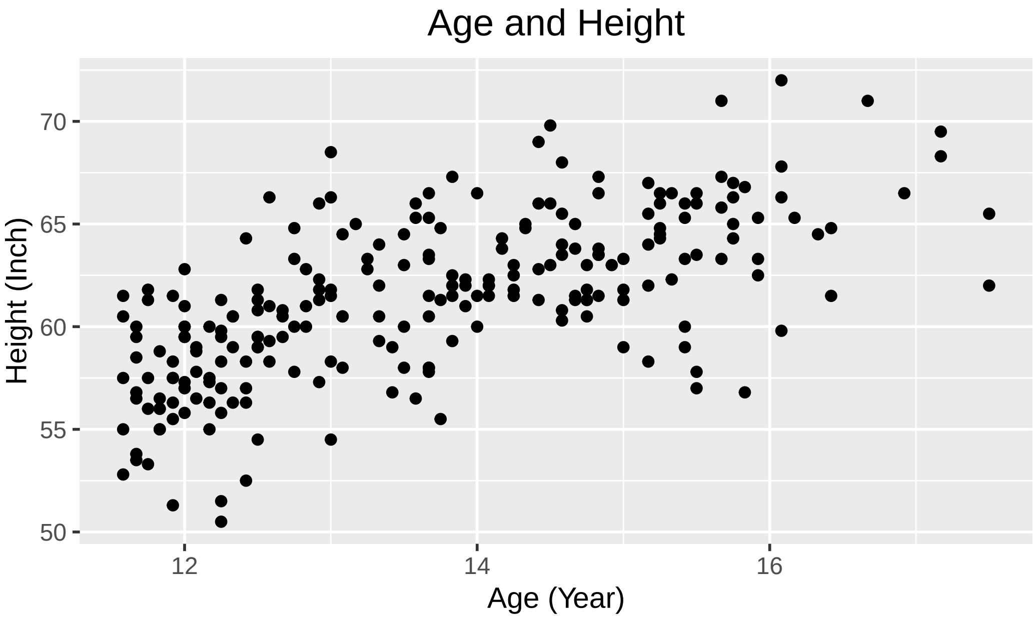
```
ggplot(faithful, aes(x = waiting)) +
  labs(title = "Old Faithful Geyser",
       x = "Waiting",
       y = "Count",
       caption = "David, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  geom_histogram(binwidth = 5, fill = "white", colour = "black")
```



7 Correlation chart

In this section, we will draw correlation chart using `heightweight` dataset. Our goal is to make a correlation between two varialbles. In this case, we correlate `ageYear` and `heightIn` by using `geom_point`.

```
ggplot(heightweight, aes(x = ageYear, y = heightIn)) +
  geom_point()+
  labs(title = "Age and Height",
       x = 'Age (Year)',
       y = "Height (Inch)",
       caption = "David, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 14))
```



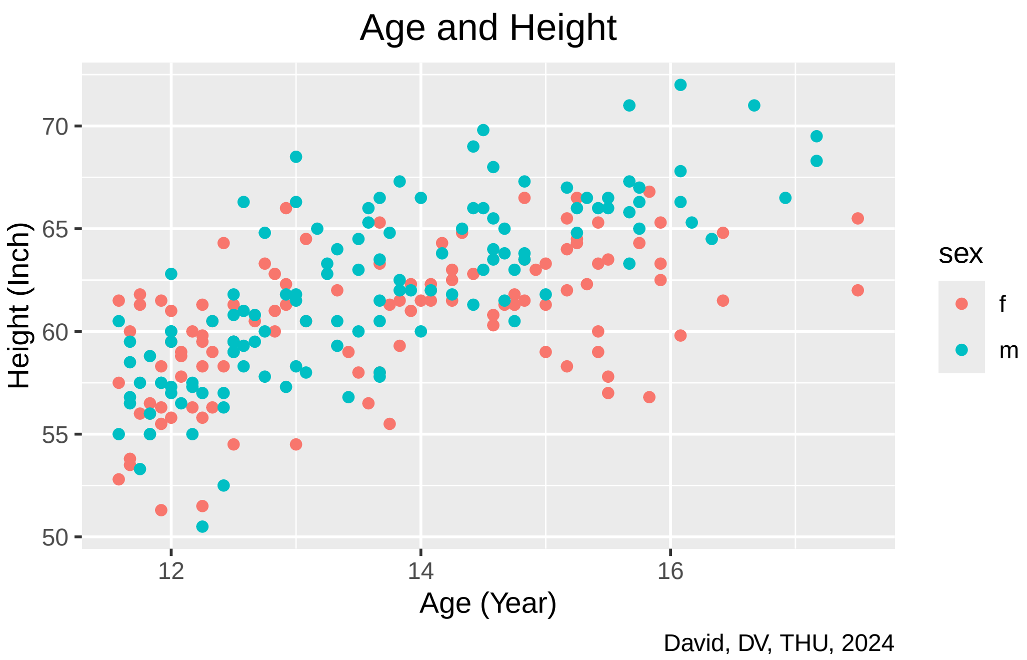
David, DV, THU, 2024

8 Correlation chart: Color by group

To add color in the chart we can add `color` function to `ggplot` function.

We can add different color to determine different group by adding what group we want to differentiate. (`color = sex`)

```
ggplot(heightweight, aes(x = ageYear, y = heightIn, color = sex)) +
  geom_point()+
  labs(title = "Age and Height",
       x = 'Age (Year)',
       y = "Height (Inch)",
       caption = "David, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 14))
```

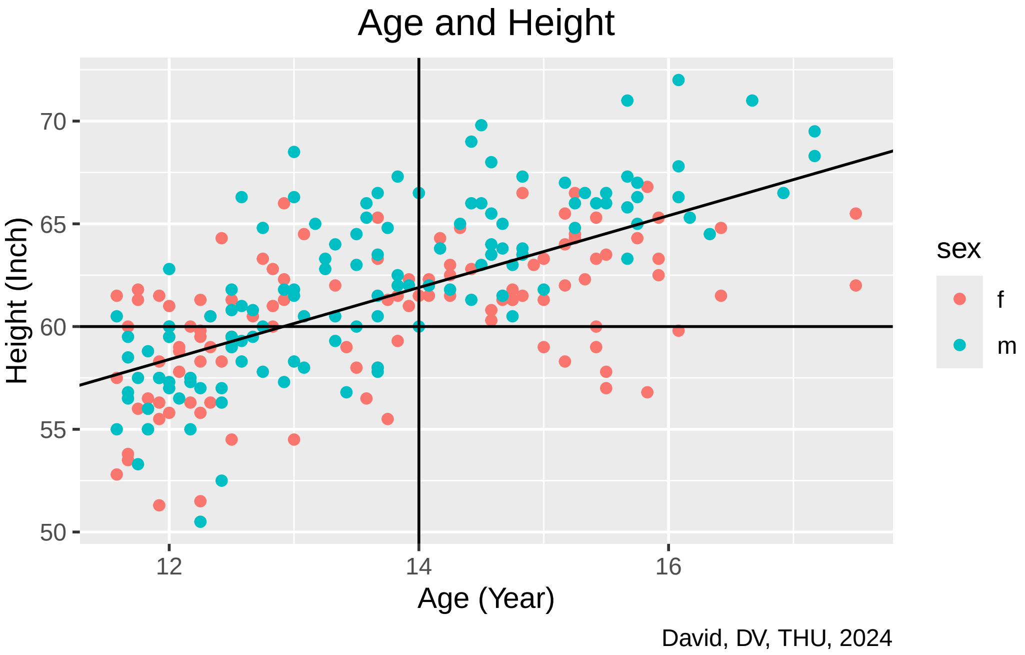


9 Correlation chart: Add lines

We can use these function to add lines in our graph:

- `geom_vline()`: To add a vertical line.
  - `yintercept =`: To determine coordinate where we want to add a horizontal line in y line.
- `geom_hline()`: To add a horizontal line.
  - `xintercept =`: To determine coordinate where we want to add a vertical line in x line.
- `geom_abline()`: To add a line according to our decision.
  - `intercept`: To determine coordinate where we want to add a line.
  - `slope`: To define the slop of the line.

```
ggplot(heightweight, aes(x = ageYear, y = heightIn, colour = sex)) +
  geom_point() +
  geom_hline(yintercept = 60) +
  geom_vline(xintercept = 14) +
  geom_abline(intercept = 37.4, slope = 1.75) +
  labs(title = "Age and Height",
       x = 'Age (Year)',
       y = "Height (Inch)",
       caption = "David, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 14))
```

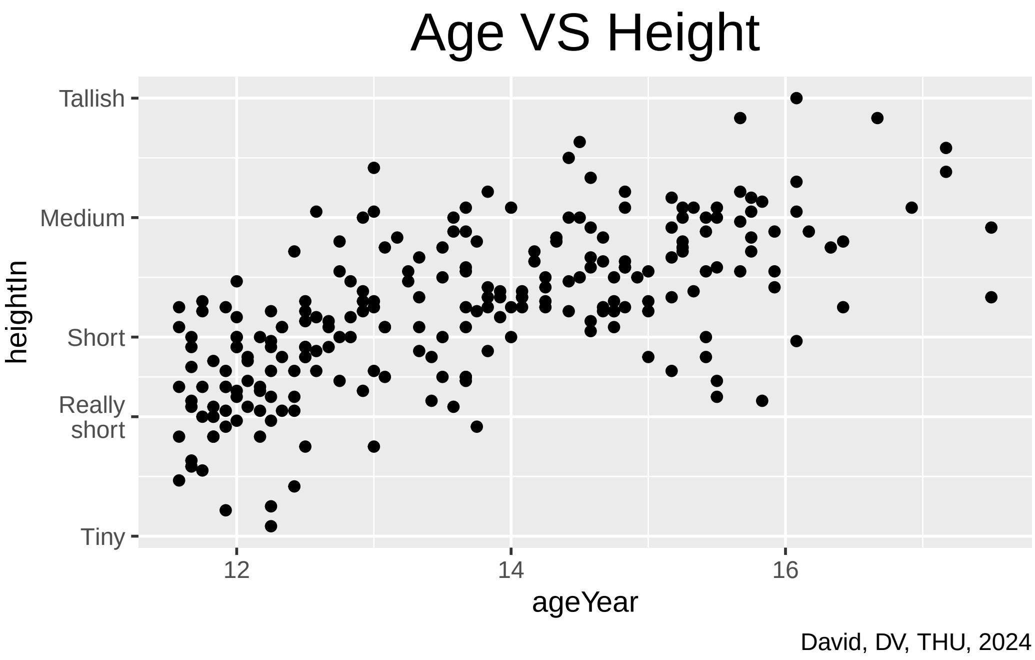


10 Correlation Chart: Change the text ticks of labels

Now we want to change the text of tick labels by using `heightwight` dataset.

- `breaks`: Specifying the locations of the tick marks and labels on the y-axis (breaks = seq(min, max, by = interval)). This gives you more control over the spacing and readability of the axis labels.
- `labels`: Customizing the text of the axis labels (labels = function(x) …). This allows for formatting, unit specification, or abbreviation of the labels.

```
ggplot(heightweight, aes(x = ageYear, y = heightIn)) +
  geom_point() +
  scale_y_continuous(
    breaks = c(50, 56, 60, 66, 72),
    labels = c("Tiny", "Really\nshort", "Short", "Medium", "Tallish")
  ) +
  labs(title = "Age VS Height", caption = 'David, DV, THU, 2024') +
  theme(plot.title = element_text(hjust = 0.5, size = 20))
```



11 Multigroup histogram

To make a double histogram in a single chart, firstly we can add `MASS` to our `library()`, so we can use `facet_grid()` function.

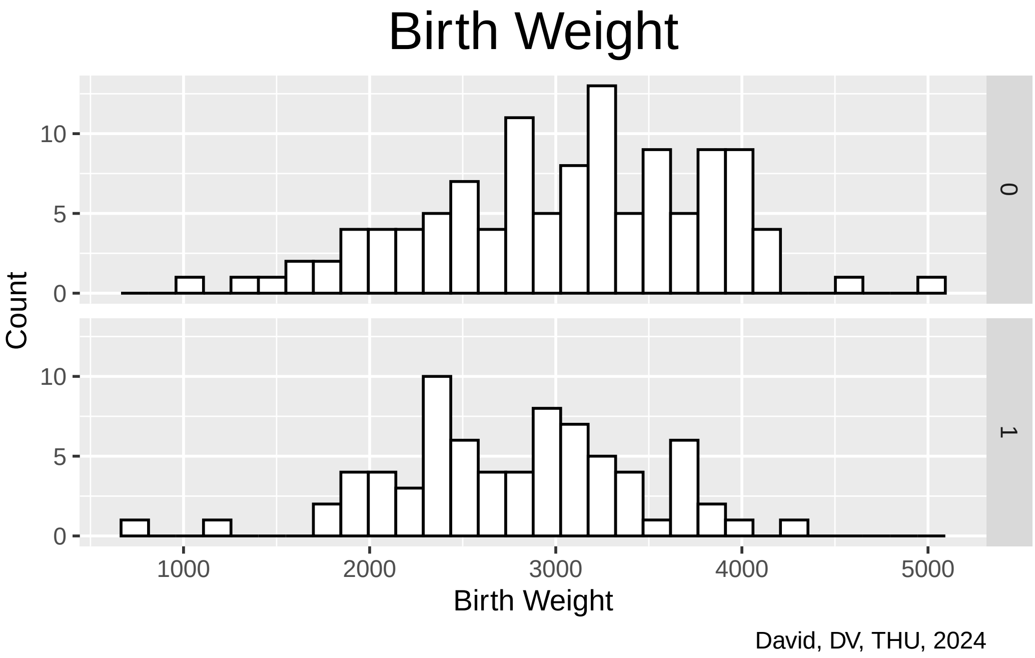
```
library(MASS)
```

In this section, we will draw a histogram using `birthwt` dataset. Our goal is to count `smoking` and `nonsmoking` in the dataset.

`facet_grid()`: To determine whether the histogram wants to split horizontally or vertically.

- `facet_grid(smoke ~ .)`: To split the histogram horizontal.
- `facet_grid(. ~ smoke)`: To split the histogram vertical.

```
ggplot(birthwt, aes(x = bwt)) +
  labs(title = "Birth Weight",
      x = "Birth Weight",
      y = "Count",
      caption = "David, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  geom_histogram(fill = "white", colour = "black") +
  facet_grid(smoke ~ .)
```



12 Multigroup Histogram: Change the name of histogram

To change the name of the histogram we can use `recode_factor()`, but first we must input `tidyverse` in our `library()`.

```
library(tidyverse)
```

We can create a new variable called `birthwt_mod`.

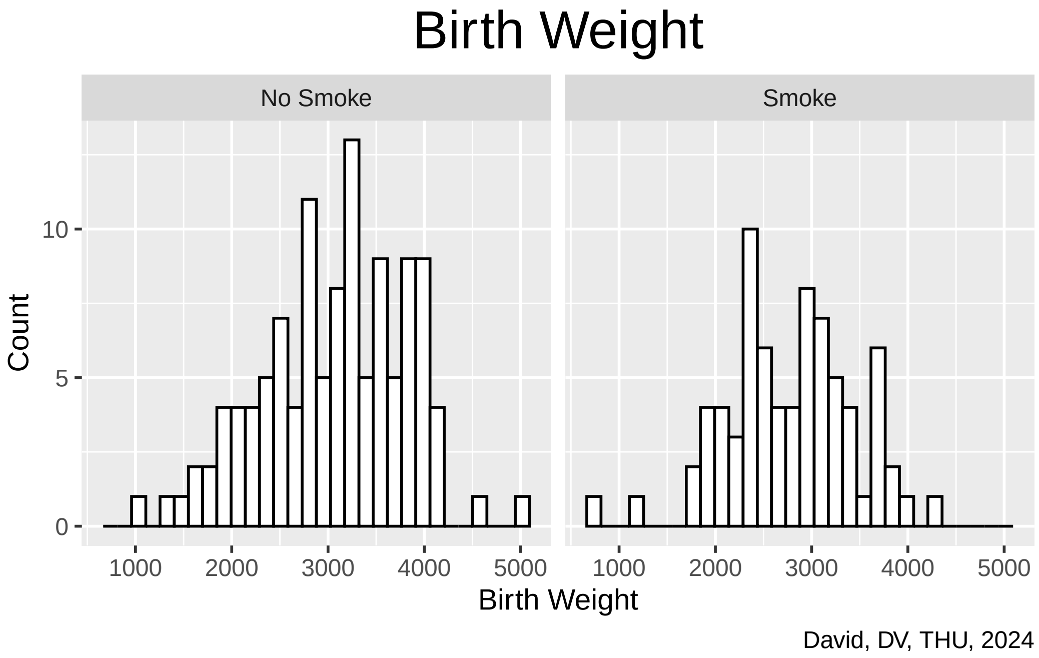
`birthwt_mod$smoke <- recode_factor(birthwt_mod$smoke, '0' = 'No Smoke', '1' = 'Smoke'):`

- We want to change the name in the `smoke row`.
    - By using `$` to mention the row name.
- Change `0` to `No Smoke`.
- change `1` to `Smoke`.

```
birthwt_mod <- birthwt

birthwt_mod$smoke <- recode_factor(birthwt_mod$smoke, '0' = 'No Smoke', '1' = 'Smoke')

ggplot(birthwt_mod, aes(x = bwt)) +
  labs(title = "Birth Weight",
       x = "Birth Weight",
       y = "Count",
       caption = "David, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  geom_histogram(fill = "white", colour = "black") +
  facet_grid(. ~ smoke)
```

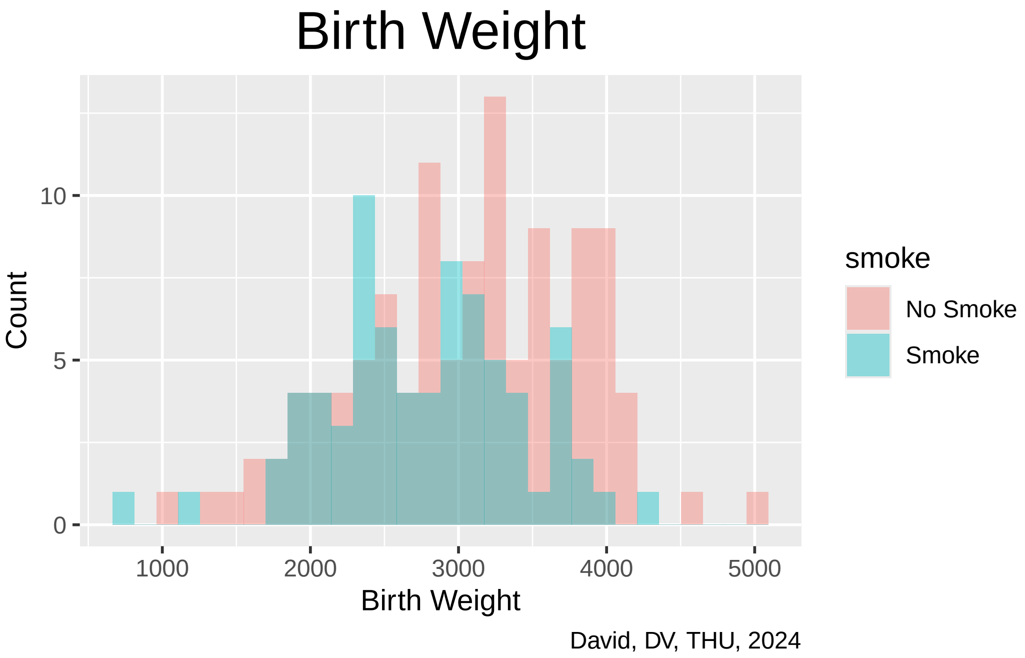


13 Multigroup Histogram: Add color

We can give color to the histogram by using `fill` in the `ggplot` function.

To make the both data become one graph we should add `position = 'identity'` and we can add `aplha` to adjust histogram density in the `geom_histogram` function.

```r
ggplot(birthwt_mod, aes(x = bwt, fill = smoke)) +
  labs(title = "Birth Weight",
       x = "Birth Weight",
       y = "Count",
       caption = "David, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  geom_histogram(position = "identity", alpha = 0.4)
```



14 Multigroup Histogram: Add more than 2 histogram chart

In this section we will use functions from `gridExtra` library.

```r
library(gridExtra)
```

To make a multigroup chart we need `grid.arrange()` function from the `gridExtra` package in R to arrange four plots (named plot1, plot2, plot3, and plot4) into a single figure.

We using `faithful` dataset.

`grid.arrange()`: To arrange a lot of charts into a single figure.

- `ncol`: To set how much charts in one row.
- `top`: To specifies the main title of a figure.
- `bottom`: To specifies the caption of a figure.

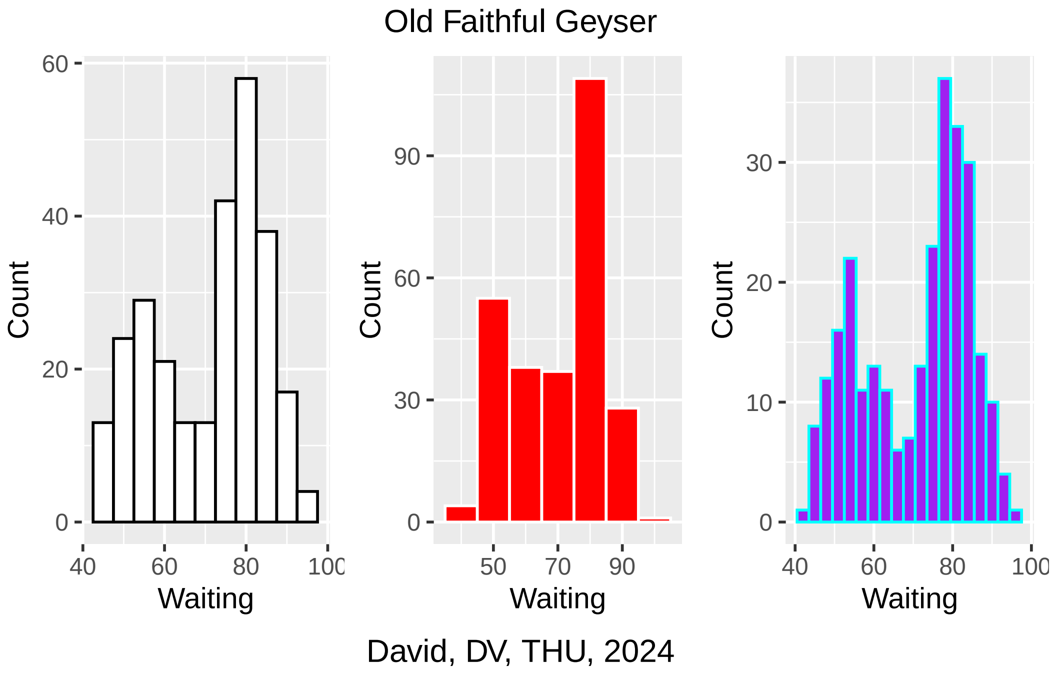
```r
tabs <- ggplot(faithful, aes(x = waiting)) +
  labs(x = "Waiting",
       y = "Count")  +
  geom_histogram()

p1 <- tabs +
  geom_histogram(binwidth = 5, fill = "white", colour = "black")

p2 <- tabs +
  geom_histogram(binwidth = 10, fill = "red", colour = "white")

p3 <- tabs +
  geom_histogram(binwidth = 3, fill = "purple", colour = "cyan")

grid.arrange(p1, p2, p3, ncol=3,
             top = 'Old Faithful Geyser',
             bottom = "David, DV, THU, 2024")
```



15 Multigroup Histogram: Remove tick marks

We can remove the tick marks in `faithful` dataset.

- To remove the tick marks, use `theme(axis.ticks=element_blank())`. This will remove the tick marks on both axes.
- To remove the tick marks, the labels, and the grid lines, set breaks to `NULL`

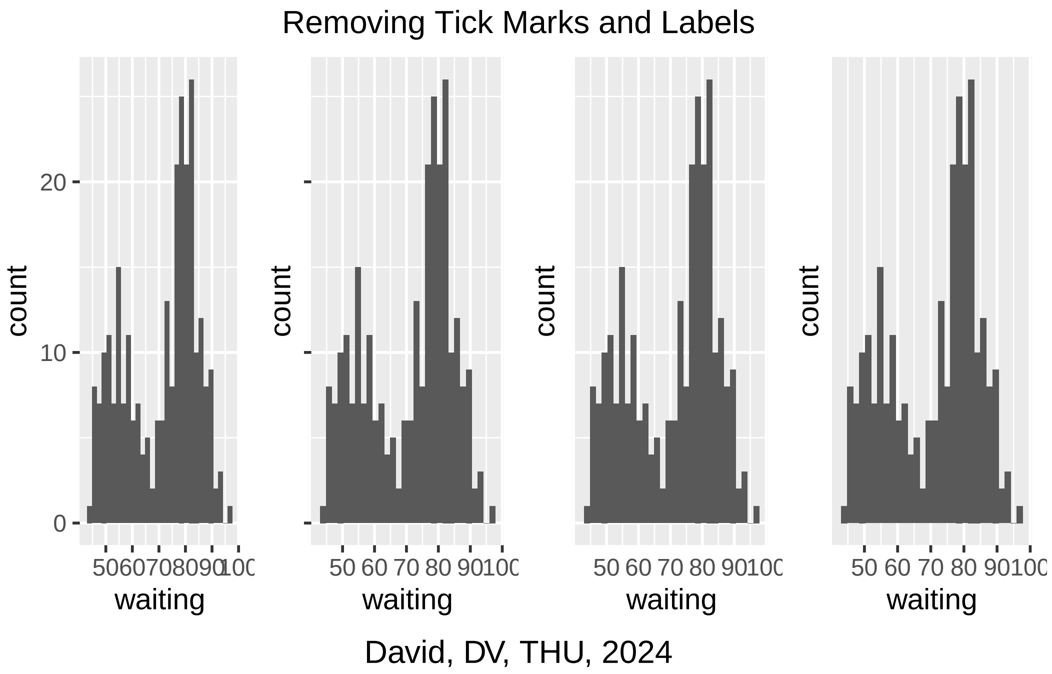
```r
p1 <- ggplot(faithful, aes(x = waiting)) +
  geom_histogram()  +
  theme(plot.title = element_text(hjust = 0.5, size = 12))

p2 <- ggplot(faithful, aes(x = waiting)) +
  geom_histogram() +
  theme(axis.text.y = element_blank())+
  theme(plot.title = element_text(hjust = 0.5, size = 12))

p3 <- ggplot(faithful, aes(x = waiting)) +
  geom_histogram() +
  theme(axis.ticks = element_blank(), axis.text.y = element_blank())  +
  theme(plot.title = element_text(hjust = 0.5, size = 12))

p4 <- ggplot(faithful, aes(x = waiting)) +
  geom_histogram() +
  scale_y_continuous(breaks = NULL) +
  theme(plot.title = element_text(hjust = 0.5, size = 12))

grid.arrange(p1, p2, p3, p4, ncol = 4,
             top = 'Removing Tick Marks and Labels',
             bottom = "David, DV, THU, 2024")
```



16 Density chart

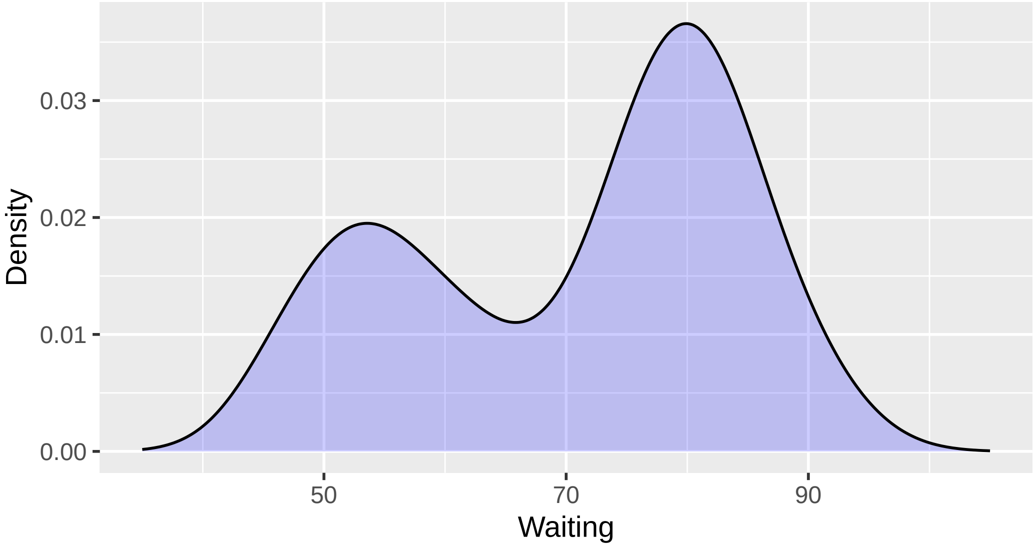
In this section, we will make a density chart using `faithful` dataset.

We can use `geom_density` to add a density chart

- `xlim()`: To determine the x line range of density chart.
- `fill`: To add a color to the density chart.
- `alpha`: To determine color density in the chart

```r
ggplot(faithful, aes(x = waiting)) +
  labs(title = "Density Chart",
       x = "Waiting",
       y = "Density",
       caption = "By David, Data Visualization Course, Tunghai University, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  geom_density(fill = "blue", alpha = .2) +
  xlim(35, 105)
```

Density Chart



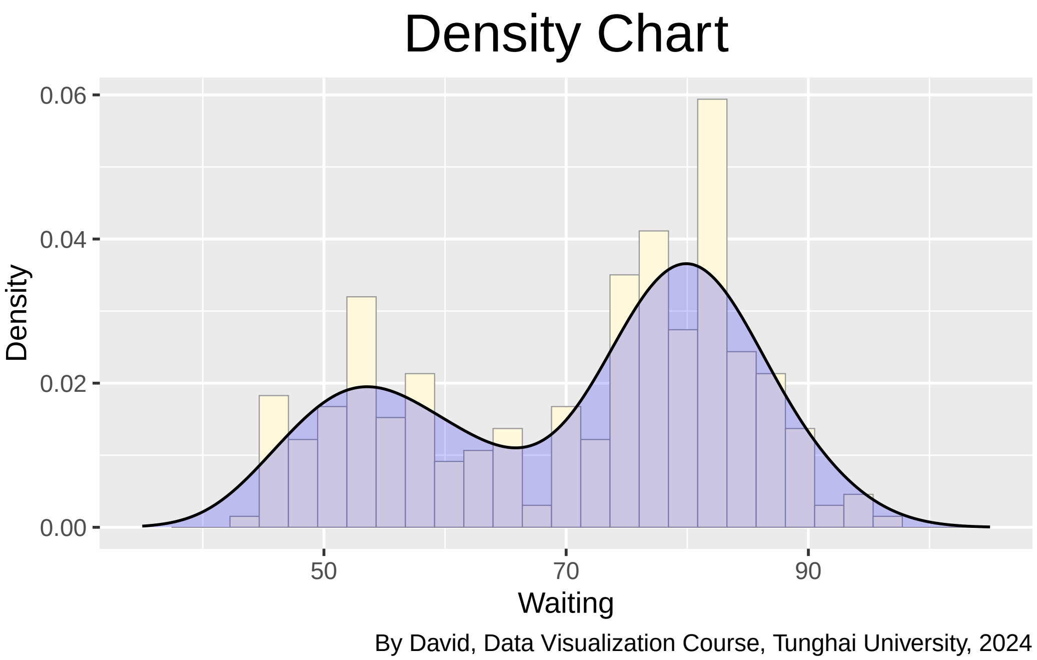
By David, Data Visualization Course, Tunghai University, 2024

17 Histogram and Density chart

To add a histogram chart in the density chart we can directly add `geom_histogram`.

- `fill`: To add a color in the density or histogram chart.
- `colour`: To change histogram border color.
- `size`: To set a size of the histogram chart.

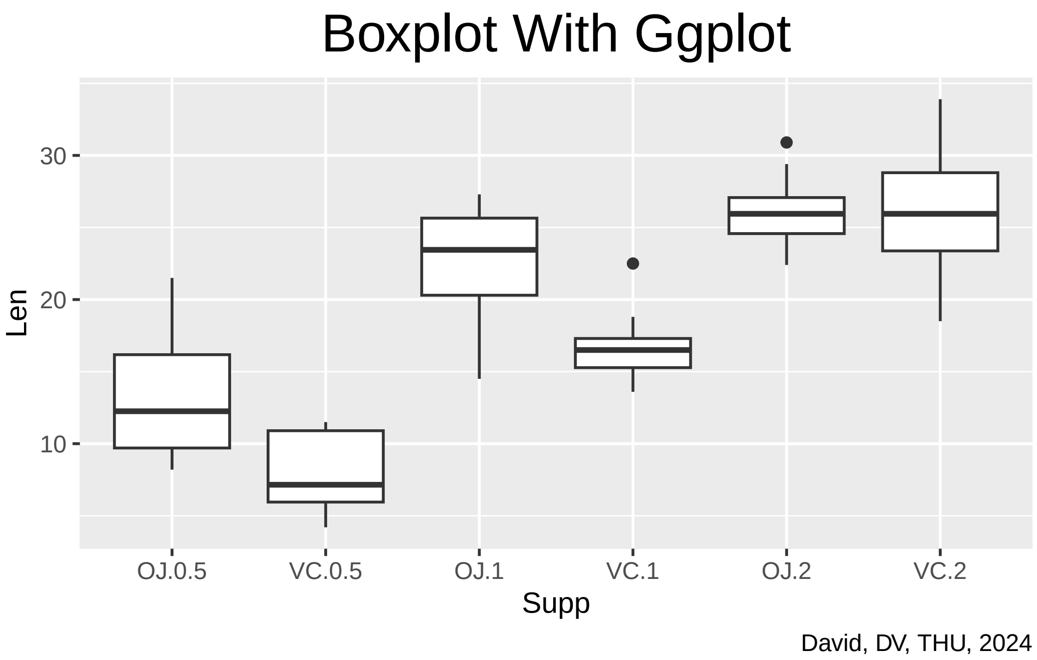
```
ggplot(faithful, aes(x = waiting, y = after_stat(density))) +
  labs(title = "Density Chart",
       x = "Waiting",
       y = "Density",
       caption = "By David, Data Visualization Course, Tunghai University, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  geom_histogram(fill = "cornsilk", colour = "grey60", size = .2) +
  geom_density(fill = "blue", alpha = .2) +
  xlim(35, 105)
```



18 Box plot

In this section, we will make a boxplot using `geom_boxplot` function in the `ggplot`. We will use `ToothGrowth` function.

```r
ggplot(ToothGrowth, aes(x  = interaction(supp, dose), y = len)) +
  labs(title = "Boxplot With Ggplot",
       x = "Supp",
       y = "Len",
       caption = "David, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  geom_boxplot()
```



19 Boxplot: Add labels and rotate labels

We use `PlantGrowth` dataset to try use these functions:

- Use `scale_x_discrete()` to change the text of the axis labels
- Use `breaks = c()` to break the axis labels
- Use `labels = c()` to add a name in axis labels
- `axis.text.x = element_text(...)`: This part specifically targets the text elements of the x-axis. element_text is a function that controls the formatting of text within the plot.
- `angle = 30`: This rotates the x-axis labels by 30 degrees. This is often useful when labels are long and overlapping.

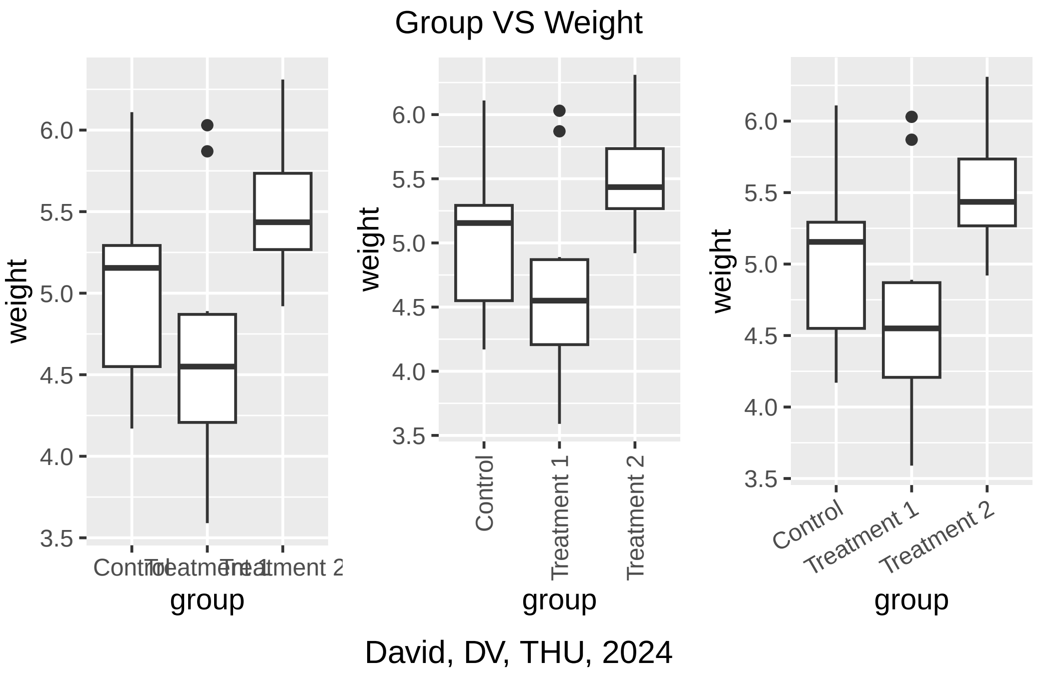
```
pg_plot <- ggplot(PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot() +
  scale_x_discrete(
    breaks = c("ctrl", "trt1", "trt2"),
    labels = c("Control", "Treatment 1", "Treatment 2")
  ) +
  theme(plot.title = element_text(hjust = 0.5, size = 16))

p1 <- pg_plot
```

```r
p2 <- pg_plot +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = .5))

p3 <- pg_plot +
  theme(axis.text.x = element_text(angle = 30, hjust = 1, vjust = 1))

grid.arrange(p1, p2, p3, ncol = 3,
             top = 'Group VS Weight',
             bottom = "David, DV, THU, 2024")
```


Group VS Weight
weight
Control
Treatment 1
Treatment 2
group
David, DV, THU, 2024