Data Visualization Analysis

Valencia

2024-12-20

Table of contents

1 Introduction

This tutorial is designed by Valencia 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.

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

---

3 Datasets

In this section, we will discuss all the datasets that are going to be use:

1. `pg_mean` dataset. The dataset has two columns: `group`, `weight`.

```
pg_mean
```

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).

2. Biochemical Oxygen Demand, `BOD`, The dataset has two columns: `Time` and `demmand`

```
BOD
```

This dataset compares Biochemical Oxygen demand over time - `Time`: The time needed - `demmand`: Biochemical Oxygen demanded

3. Edgar Anderson's Iris Data, `iris`, The dataset has 5 columns:

```
iris
```

- `Sepal.Length`: Length of the sepal
- `Sepal.Width`: Width of the sepal
- `Petal.Length`: Length of the petal
- `Petal.Width`: Width of the petal
- `Species`: The species of the flower

4. Risk Factors Associated with Low Infant Birth Weight, `birthwt`, The dataset has 10 columns:

```
birthwt
```

- `low`: Indicator of birth weight less than 2.5 kg
- `age`: Mother's age
- `lwt`: Mother's weight at last menstrual period
- `race`: Mother's race
- `smoke`: Smoking status during pregnancy
- `ptl`: Number of previous premature labors
- `ht`: History of hypertension

- `ui`: Presence of uterine irritability

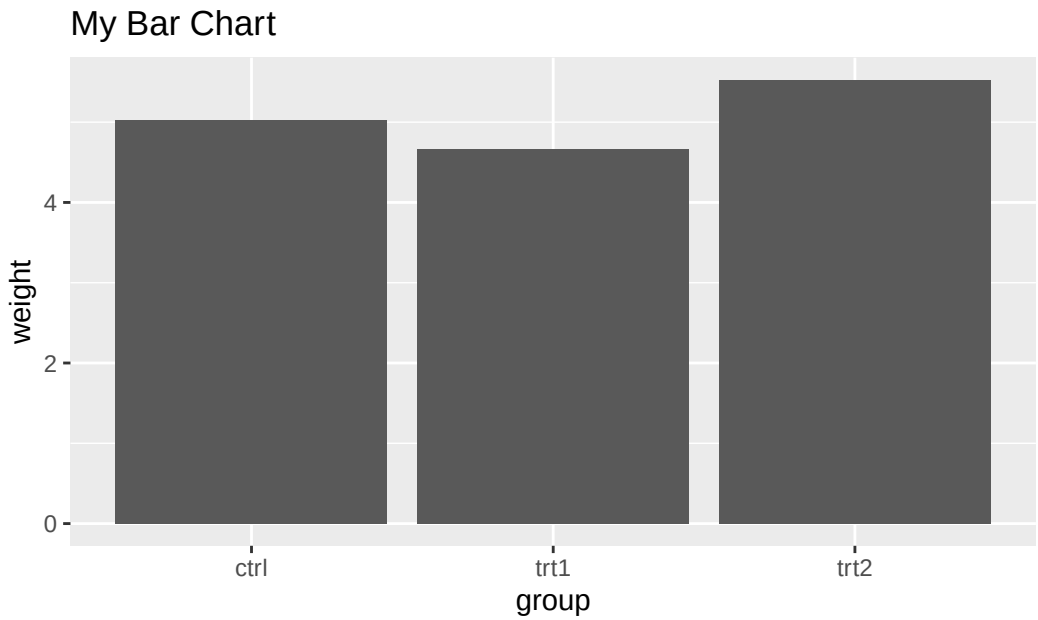
- `ftv`: Number of physician visits during the first trimester
- `bwt`: Birth weight of the baby

---

4 Bar chart

In this section, we will draw a bar chart using the `pg_mean` dataset.

```
ggplot(pg_mean, aes(x = group, y = weight)) +
  geom_col() +
  labs(title='My Bar Chart',
       captions='By Valencia, Data Visualization, THU 2024')
```



It 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.

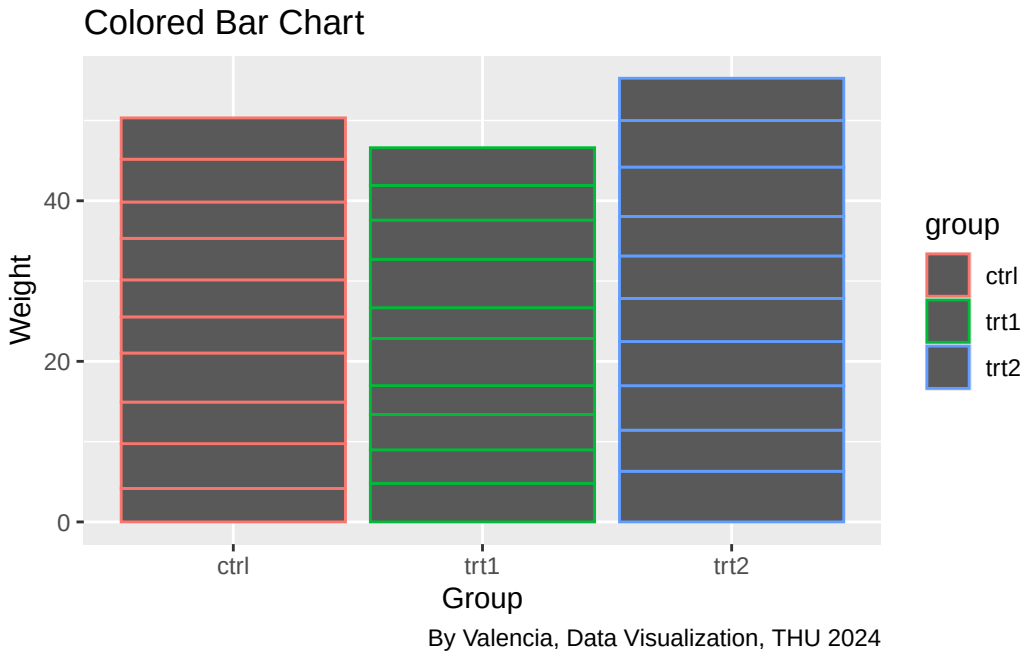
---

5 Bar chart with color

For better visualization, we use `colors`, to create a more distinct appearance of the visualization.

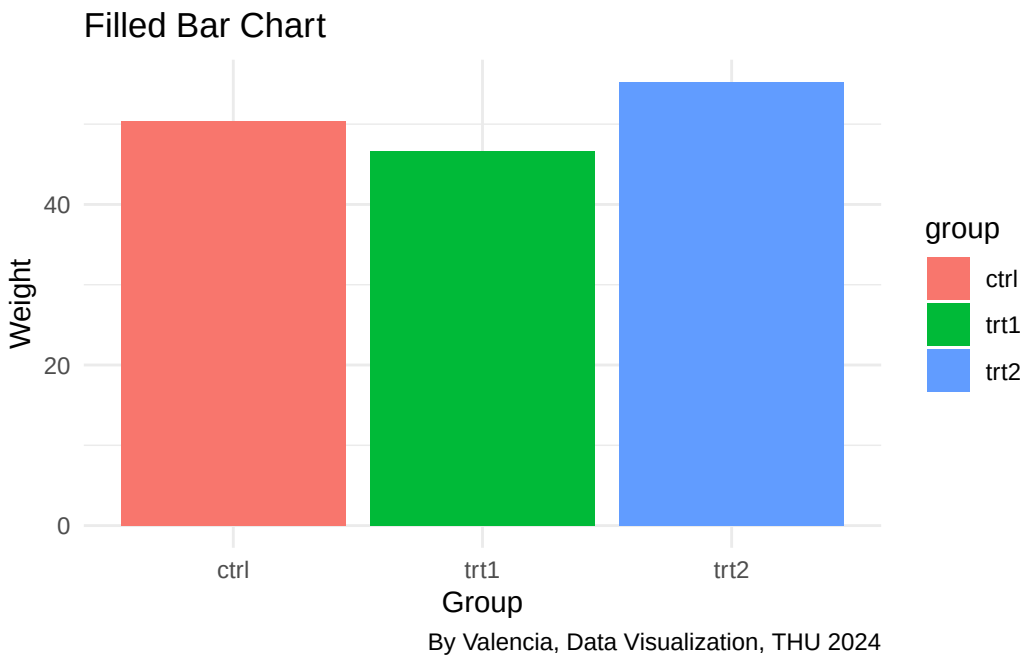
`color` is used to add colors to the outline of the chart

```
ggplot(PlantGrowth, aes(x = group, y = weight, color = group)) +
  geom_col()+
  labs(title = 'Colored Bar Chart',
      x= 'Group',
      y= 'Weight',
      captions= 'By Valencia, Data Visualization, THU 2024')
```



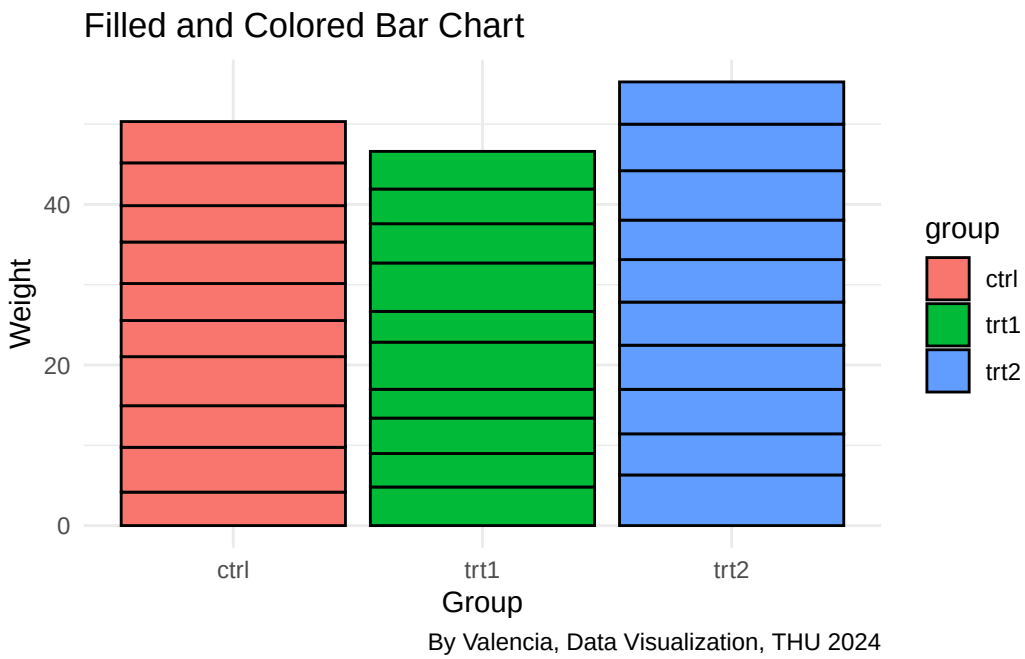
`fill` is used to add colors to the bars

```
ggplot(PlantGrowth, aes(x = group, y = weight, fill = group)) +
  geom_col()+
  theme_minimal()+
  labs(title = 'Filled Bar Chart',
      x= 'Group',
      y= 'Weight',
      captions= 'By Valencia, Data Visualization, THU 2024')
```



We can combine `fill` and `colors`

```
ggplot(PlantGrowth, aes(x = group, y = weight, fill = group)) +
  geom_col(color= 'black')+
  theme_minimal()+
  labs(title = 'Filled and Colored Bar Chart',
      x= 'Group',
      y= 'Weight',
      captions= 'By Valencia, Data Visualization, THU 2024')
```



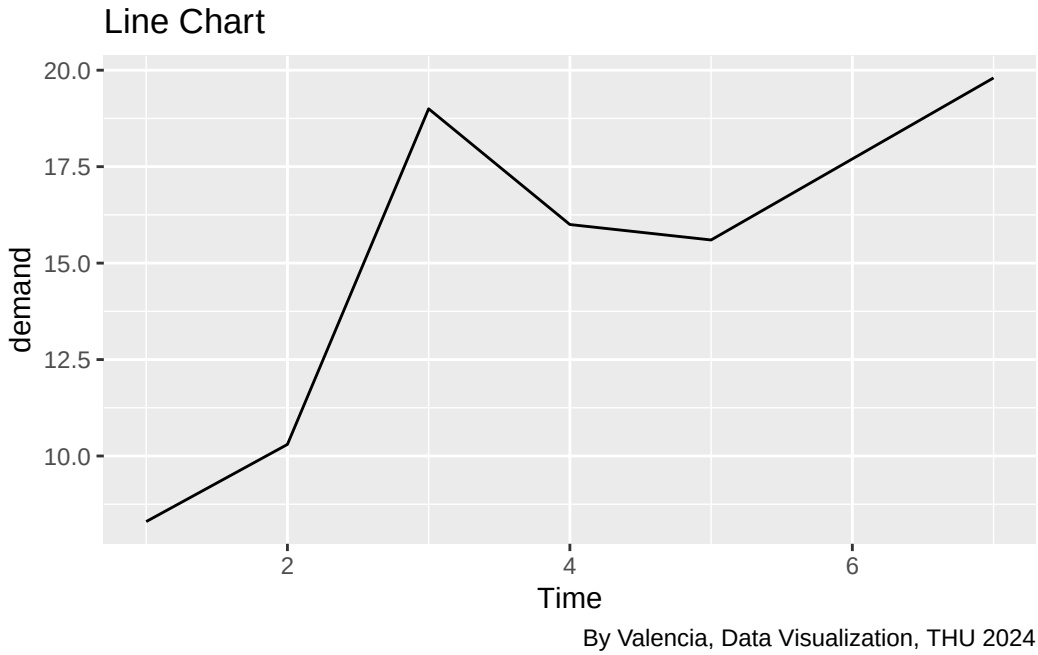
---

6 Line chart

In this section, we will also teach how to make line chart. Using `geom_line()`, the `ggplot2` pacakage helps us plot the line chart for us

For this section we are going to use the `BOD` dataset provided by the `ggplot2` package

```
ggplot(BOD, aes (x = Time, y = demand)) +
  geom_line() +
  labs(title= "Line Chart",
       caption = 'By Valencia, Data Visualization, THU 2024')
```



`aes()`: Defines the following aesthetic:

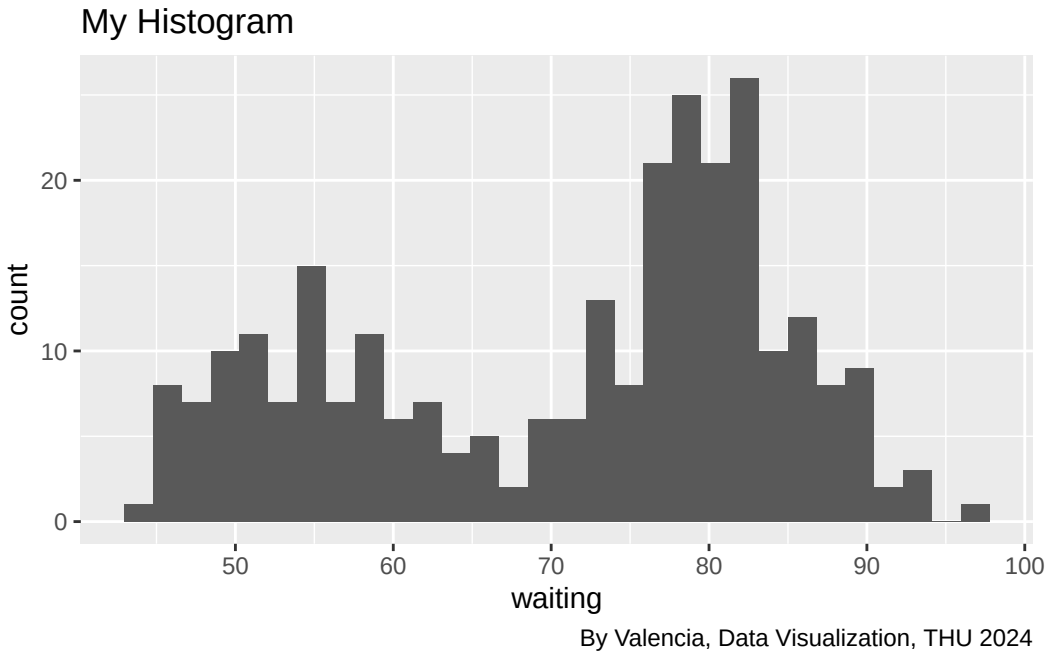
- `x = Time`: Plot the `Time` variable to the x-axis
- `y = demand`: Plot the `demand` variable to the y-axis

---

7 Histogram

In this section, we will learn on how to make a histogram. We can use the `geom_histogram()` function provided by the `ggplot2` package. `faithful` dataset provided by the `ggplot2` will be used in this section

```
ggplot(faithful, aes (x=waiting))+
  geom_histogram()+
  labs(title = "My Histogram",
       caption= "By Valencia, Data Visualization, THU 2024")
```

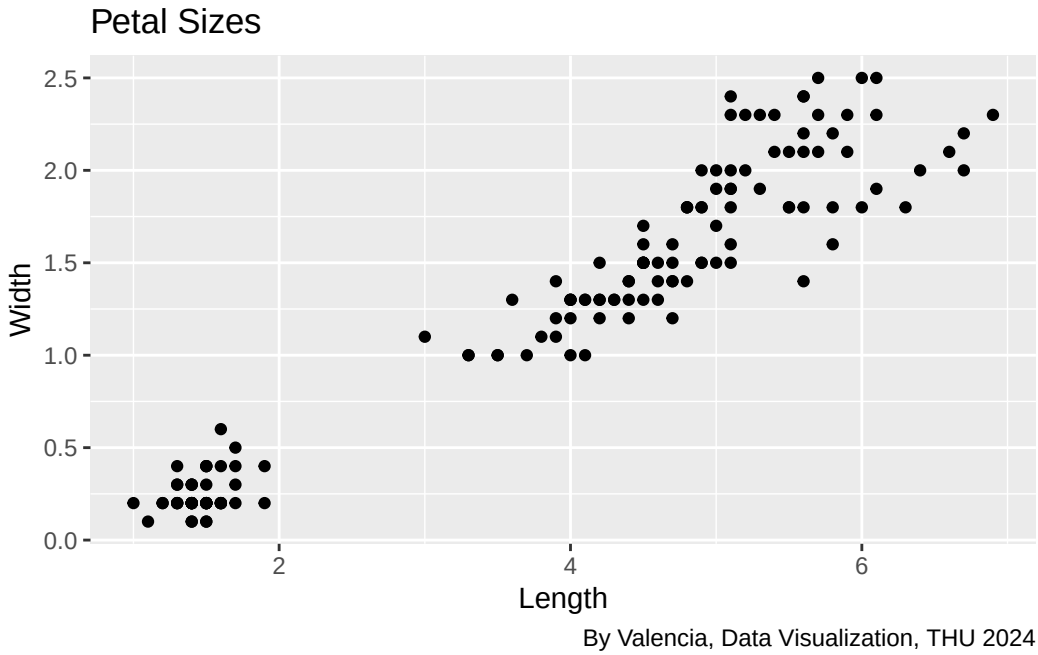


---

8 Correlation chart

In this section, we will learn on how to create Correlation chart or Scatter plot using `geom_point`. `iris` dataset provided by `ggplot2` package will be used in this section

```
ggplot(iris, aes(x = Petal.Length, y = Petal.Width )) +
  geom_point()+
  labs(title="Petal Sizes",
       x= 'Length',
       y= 'Width',
       caption = 'By Valencia, Data Visualization, THU 2024')
```



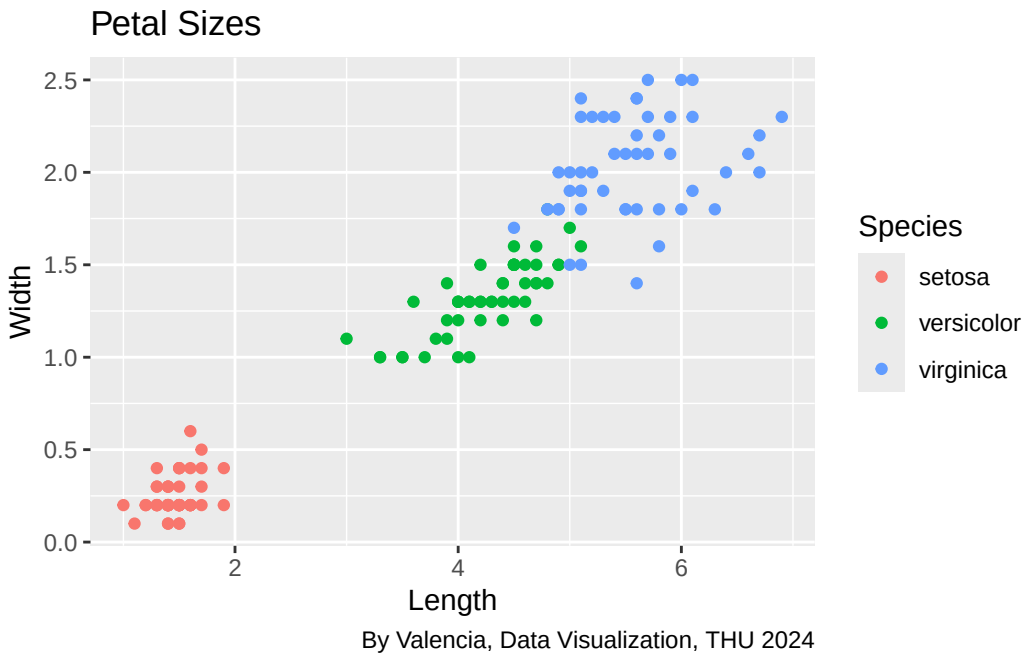
---

9 Correlation chart: Color by group

In the `iris` dataset, the column `species` is provided, which sepeartes it into 3 groups (`setosa`, `versicolor`, and `virginica`), a better method to visualize the `species` is by grouping them with `colors`. we use `color=species` to color the Scatter plot and categorizing by the `species` variable

```
#|message: false
#|warning: false

ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color= Species )) +
  geom_point()+
  labs(title="Petal Sizes",
       x= 'Length',
       y= 'Width',
       caption = 'By Valencia, Data Visualization, THU 2024')
```



As you can see, now we can clearly see the distribution of the `iris`'s `species` Petal Sizes

---

10 Multigroup histogram

In this section we will make multigroup histogram. As the name suggest, multigroup histogram is multiple overlayed with each other. We will use the `birthwt` dataset provided by the `MASS` in this section.

Specifically for this section, we will need extra function and `dataset`, from the `dplyr` and `MASS` packages

```
#|message: false
#|warning: false

library(dplyr)
```

```
Attaching package: 'dplyr'
```

```
The following object is masked from 'package:MASS':

    select

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(MASS)
```

We first need to load `dplyr` package in order to use the `recode_factor` function and `MASS` in order to use the `birthwt` dataset

```
#|message: false
#|warning: false

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

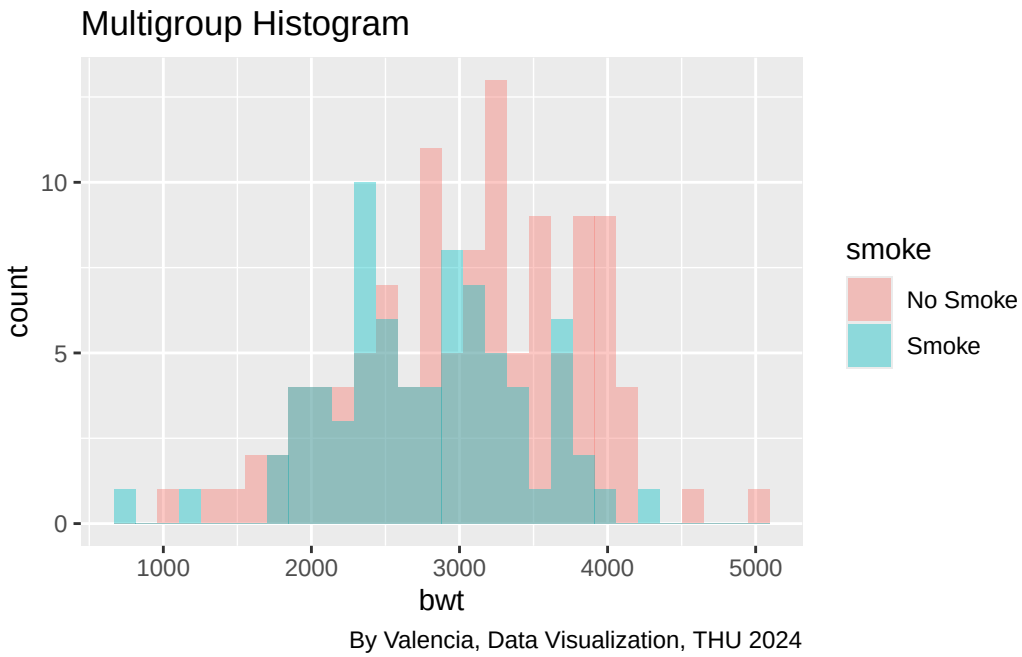
Temporary dataset will be made so that the original dataset will not be affected. The `recode_factor` function is used to change binary data into variables, in this case `0` and `1` is changed into `No Smoke` and `Smoke`.

In order to let R know where the data is, we will need to write `birthwt_mod$smoke` to indicate which dataset and column to recode

```
#|message: false
#|warning: false

ggplot(birthwt_mod, aes(x = bwt, fill = smoke)) +
  geom_histogram(position = "identity", alpha = 0.4) +
  labs(title="Multigroup Histogram",
       caption = 'By Valencia, Data Visualization, THU 2024')
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Once it's done, we will use `geom_histogram()` to create the histogram. And by using `position = identity`, it means that we are able to make the histogram overlay with each other, `Smoke` `No smoke`

`alpha` is used to change the transparency of the chart and `fill = smoke` is the grouped variables

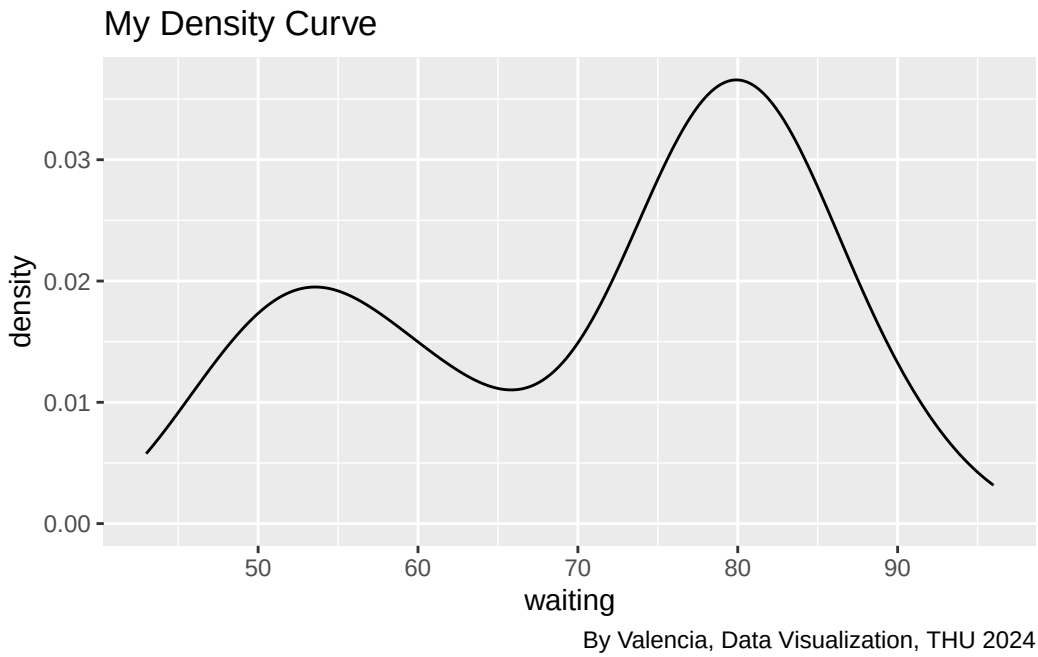
---

11 Density chart

In this section, we will make Density chart. Density Chart is a geometric provided by `ggplot2` package that displays the distribution like `histogram` but in a smooth line

```
#|message: false
#|warning: false

ggplot(faithful, aes(x=waiting)) +
   geom_density() +
  labs(title="My Density Curve",
       caption = 'By Valencia, Data Visualization, THU 2024')
```



We use`geom_density()` as a function to create Density Chart provided by `ggplot2` package.

---

12 Histogram and Density chart

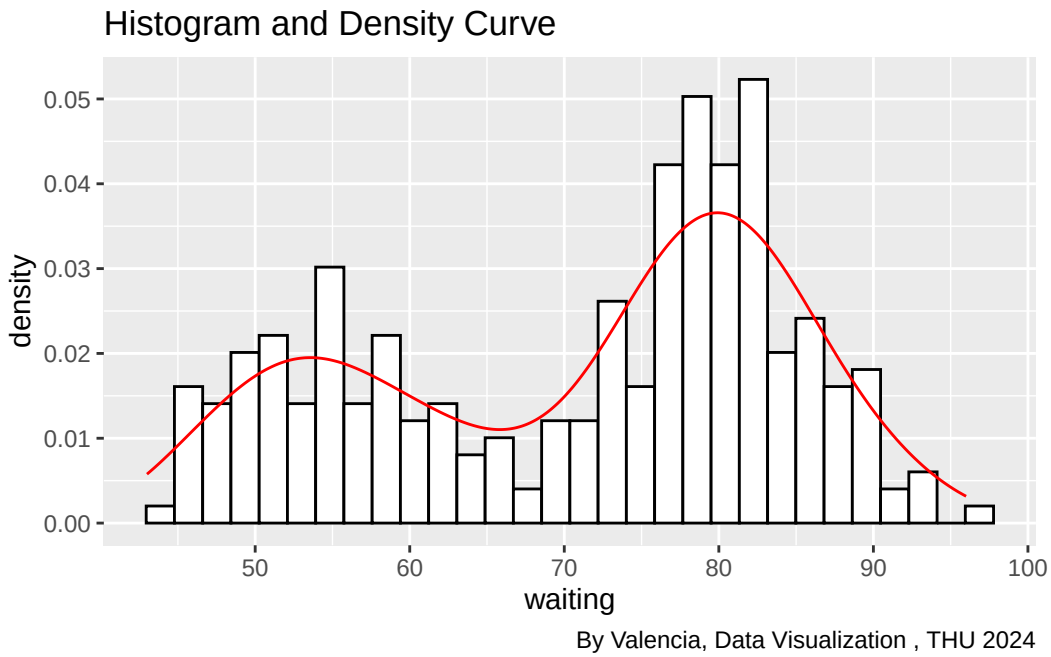
We can combine Histogram and Density Chart using the `geom_histogram` and `geom_density` for better visualization. The histogram will serve as a raw visualization of the data and the Density chart will visualize the smooth approximations

```
#|message: false
#|warning: false

ggplot(faithful, aes(x=waiting, y=..density..)) +
  geom_histogram(color='black', fill='white') +
  geom_density(color='red') +
  labs(title="Histogram and Density Curve",
       caption = 'By Valencia, Data Visualization , THU 2024')
```

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

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



In order to create the combined chart, we must first place the appropriate aesthetics (`aes()`). Althought the, `geom_density` does not require a y-axis `variable`, the `geom_histogram` needs the y-axis `variable`. So in this demonstration we added `y=..density..` so it ensures that it will represent the density data points

For better visualization, `color` and `fill` have been added to the appropriate geometrics

It's also very important to where to place the `geom_histogram`, `geom_density` and other geometrics when you want to display multiple chart. If you place the `geom_histogram` first and then `geom_density`, this means that the density curves will overlay the histogram.

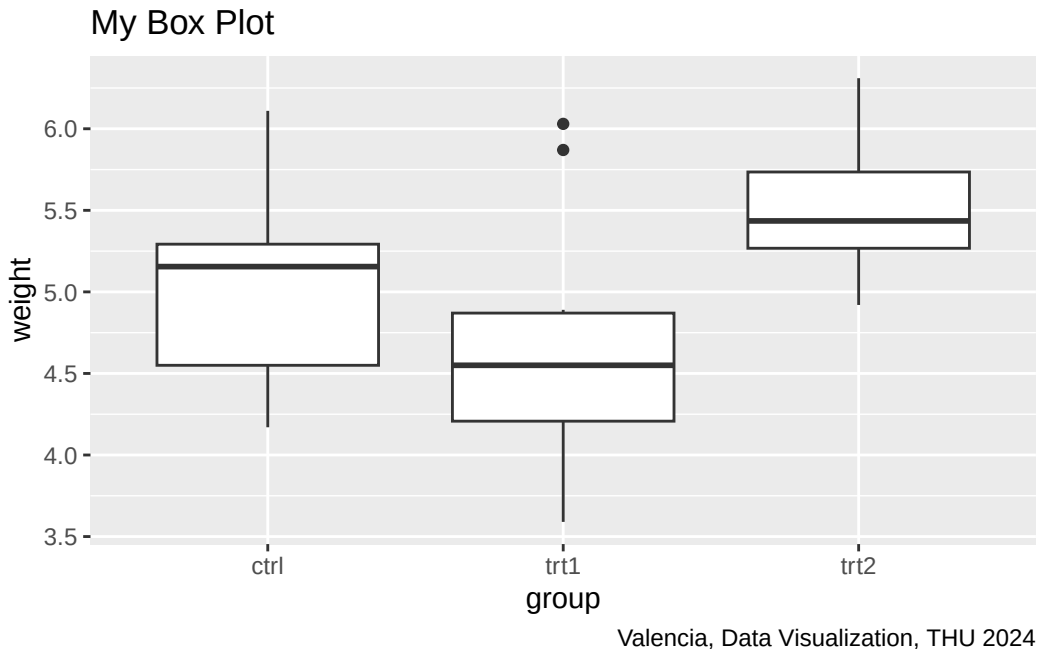
---

13 Box plot

The last section of charts that will be discussed in this tutorial is Box Plot, we use `geom_boxplot()` to plot the box plot. `PlantGrowth` dataset provided by the `ggplot2` packaage will be used in this section

```
#|message: false
#|warning: false

ggplot(PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot() +
  labs(title = 'My Box Plot',
       caption = 'Valencia, Data Visualization, THU 2024')
```



To make the box plot, we use `geom_boxplot()`. The following explains the plot's aesthetic

- `x = group`: Plot the `experimental treatment`.
- `y = weight`: Plot the `weight` of the plant

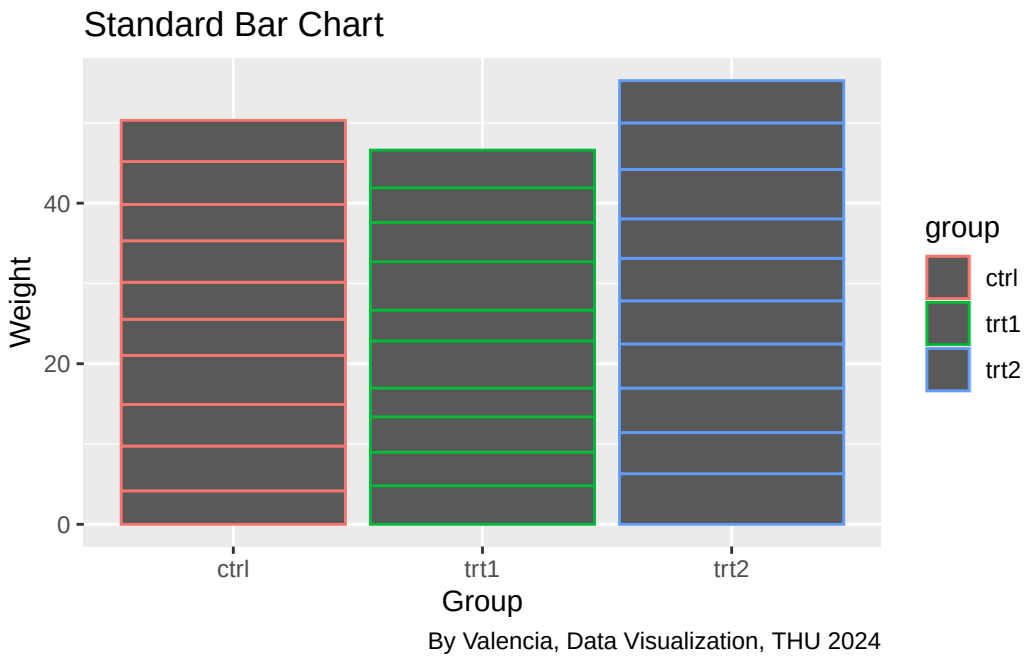
---

14 Customizing your Chart with Labels

When creating our chart or visualization, it is important to add labels. Labels serve as a function to further enhance visualization by labeling plots which gives the reader a better understanding and direction. Moreover, it's also used to claim credits of your work.

```
#|message: false
#|warning: false

ggplot(PlantGrowth, aes(x = group, y = weight, color = group)) +
  geom_col()+
  labs(title = 'Standard Bar Chart',
      x= 'Group',
      y= 'Weight',
      captions= 'By Valencia, Data Visualization, THU 2024')
```



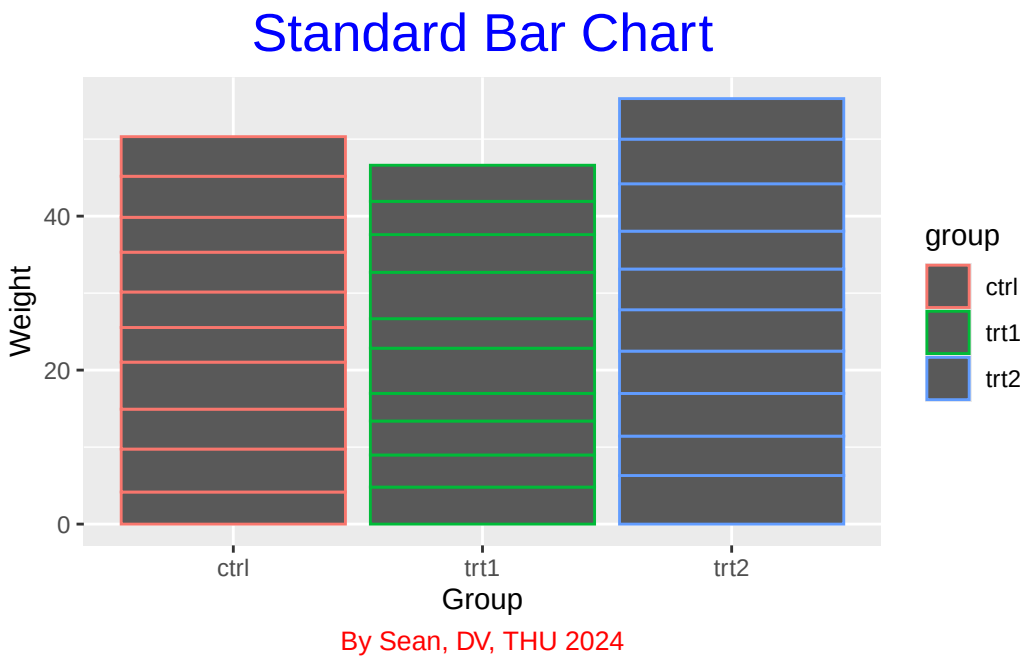
We can use `labs()` to change add title, axis labels and captions:

- `labs(title=` : Is a function to add `'text'` to the title of the chart
- `labs(x=`: Is a function to change the x-axis label
- `labs(y-`: Is a function to change the y-axis label
- `labs(captions=`: Is a function to add `'text'` on the bottom right of the chart

When adding `'text'` in R, it is important to use `' '` between the text, otherwhise it will be recognize as a function, instead of a `'text'`

```
ggplot(PlantGrowth, aes(x = group, y = weight, color = group)) +
  geom_col()+
  labs(title = 'Standard Bar Chart',
```

```
    x= 'Group',
    y= 'Weight',
    captions= 'By Sean, DV, THU 2024') +
theme(plot.title = (element_text(size=20, hjust= 0.5, color = 'blue')),
      plot.caption = (element_text(size=10, hjust= 0.5, color = 'red')))
```



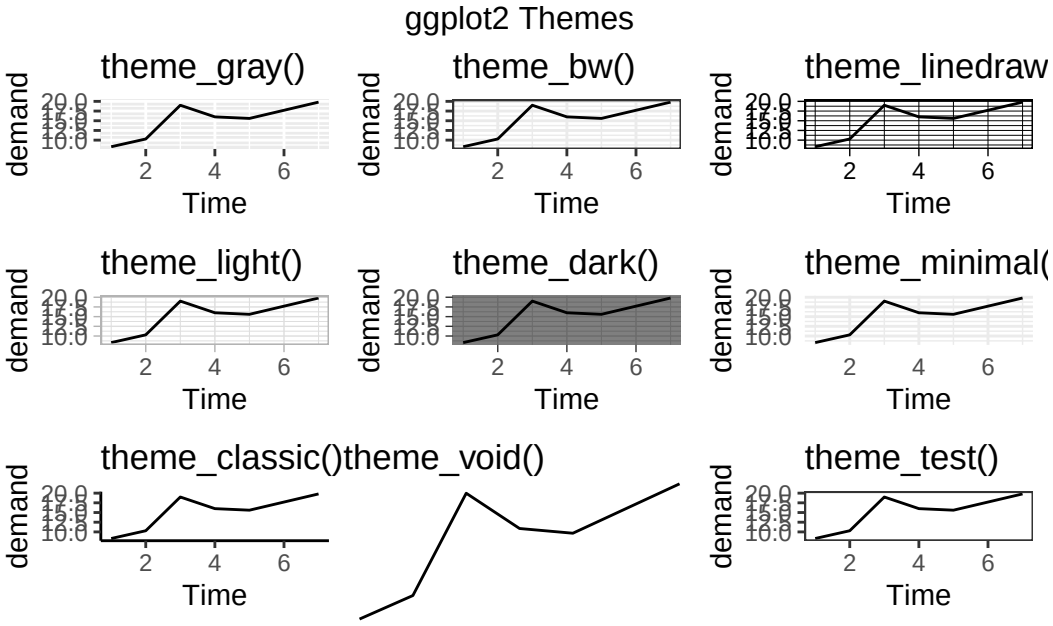
Furthermore, we can also change the appearance of the `'text'` in the `theme()` which we will discuss in more detail in the next section

- `theme(plot.title`: A function to let the R know to change the `title`'s element
- `element_text`: Which element to change the appearance
- `size`: change the font size
- `hjust`: changes the position of the text
- `theme(plot.caption`: A function to let R know to change the `caption`'s element

---

15 Theme

`theme()` is a feature that is provided by the `ggplot2` packages



- `theme_gray()` (default):
  - The signature `ggplot2` theme with a grey background and white gridlines, designed to put the data forward yet make comparisons easy.
- `theme_bw()` (black and white):
  - The classic dark-on-light `ggplot2` theme. May work better for presentations displayed with a projector.
- `theme_linedraw()`:
  - A theme with only black lines of various widths on white backgrounds, reminiscent of a line drawing. Serves a purpose similar to `theme_bw()`. Note that this theme has some very thin lines (`<< 1 pt`) which some journals may refuse.
- `theme_light()`:
  - A theme similar to `theme_linedraw()` but with light grey lines and axes, to direct more attention towards the data.
- `theme_dark()`:
  - The dark cousin of `theme_light()`, with similar line sizes but a dark background. Useful to make thin coloured lines pop out.
- `theme_minimal()`:
  - A minimalistic theme with no background annotations.

- `theme_classic()`:
  - A classic-looking theme, with x and y axis lines and no gridlines.
- `theme_void()`:
  - A completely empty theme.
- `theme_test()`:
  - A theme for visual unit tests. It should ideally never change except for new features.

---

16 Adding and Removing Legend

To add Legend:

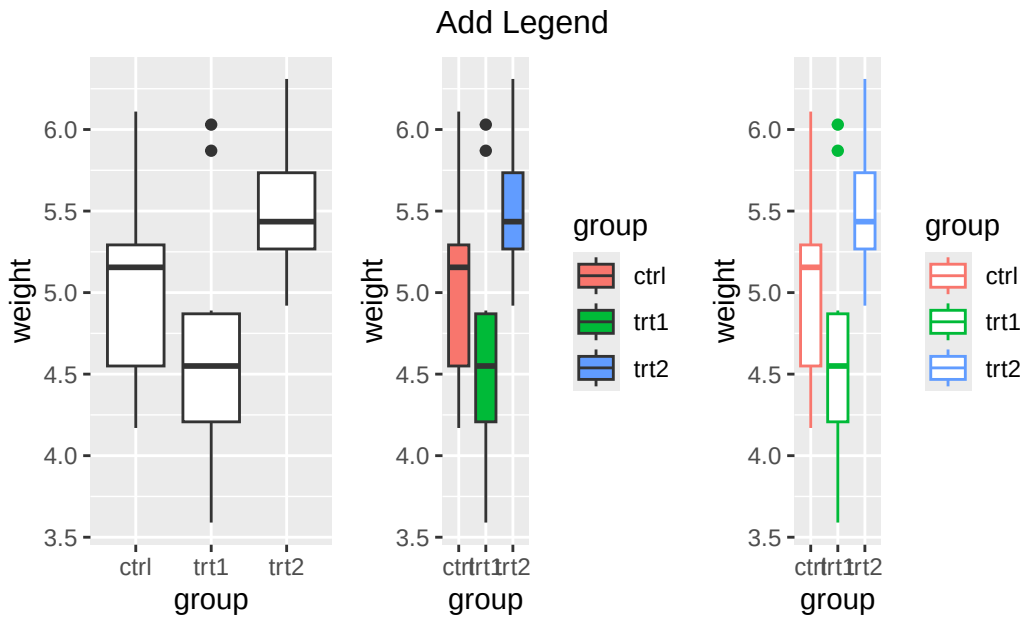
By adding `fill` or `color` options to `aes()`, a legend is created automatically on the right side of the plot.

- `library(gridExtra)`: This loads the `gridExtra` library to use `grid.arrange`
- `fill`: Fill in the `colors` inside the chart
- `color`: Change the `color` of the outlines of the chart
- `grid.arrange`: Arrange the order of the chart
- `top`: Add `text` on the top of the chart
- `bottom`: Add `text` on the bottom of the chart

```
#|message: false
#|warning: false

library(gridExtra)

p1 <- ggplot(PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot()
p2 <- ggplot(PlantGrowth, aes(x = group, y = weight, fill = group)) +
  geom_boxplot()
p3 <- ggplot(PlantGrowth, aes(x = group, y = weight, color = group)) +
  geom_boxplot()
grid.arrange(p1,p2,p3,ncol=3,top='Add Legend', bottom = 'By Valencia, Data Visualization, THU
```



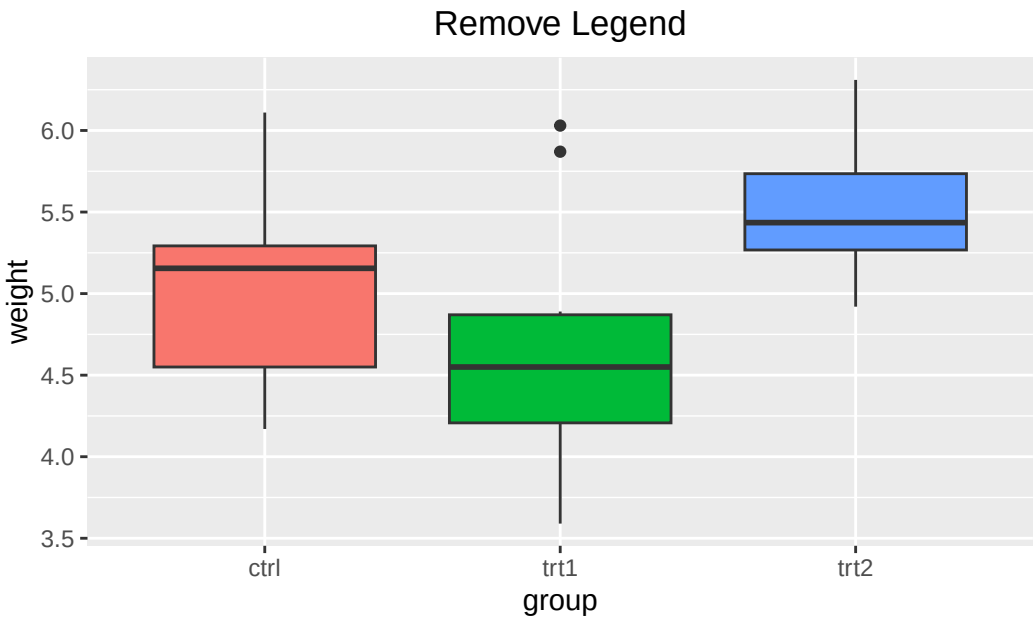
To Remove Legend

The legend is removed by `guides(fill = FALSE)`

```
#|message: false
#|warning: false

ggplot(PlantGrowth, aes(x = group, y = weight, fill = group)) +
  geom_boxplot() +
  guides(fill = FALSE) +
  ggtitle('Remove Legend') +
  labs(caption = 'By Valencia, Data Visualization, THU 2024') +
  theme(plot.title = element_text(hjust=0.5))
```

```
Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
of ggplot2 3.3.4.
```


Remove Legend
weight
6.0
5.5
5.0
4.5
4.0
3.5
ctrl
trt1
trt2
group
By Valencia, Data Visualization, THU 2024