Axes

Daven

2024-11-25

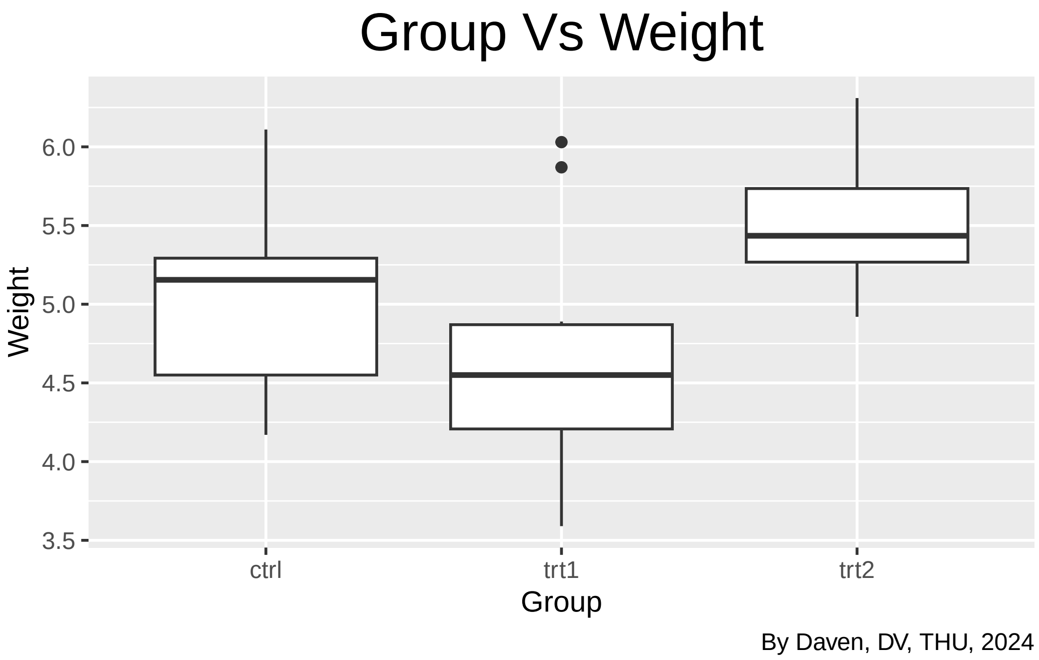
Table of contents

1. Box Plot

- Draw a boxplot using `PlantGrowth` dataset.
- x axis represents `group`, y axis represents `weight`.
- Use `geom_boxplot()` function to draw boxplot.
- `labs()` function is to add title, axis lables, and caption.
- Use `theme()` function to center the title.

```r
library(ggplot2)

ggplot(PlantGrowth, aes (x = group, y = weight)) +
  geom_boxplot() +
  labs(title = "Group Vs Weight",
       x = "Group",
       y = "Weight",
       caption = "By Daven, DV, THU, 2024")+
  theme(plot.title = element_text(hjust = 0.5, size = 20))
```

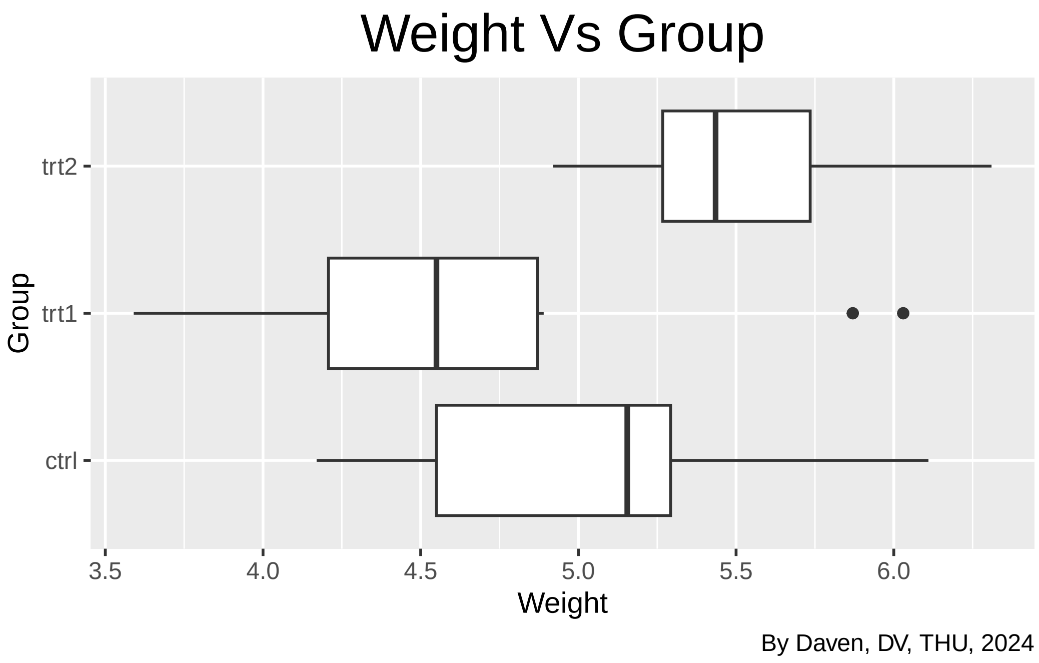


2. Swapped X and Y Axis

- Use `coord_flip()` to swapped the X and Y Axis lable

```
library(ggplot2)

ggplot(PlantGrowth, aes (x = group, y = weight)) +
  geom_boxplot() +
  coord_flip() +
  labs(title = "Weight Vs Group",
       x = 'Group',
       y = 'Weight',
       caption = "By Daven, DV, THU, 2024")+
  theme(plot.title = element_text(hjust = 0.5, size = 20))
```



3. Arrange 2 plots side by side

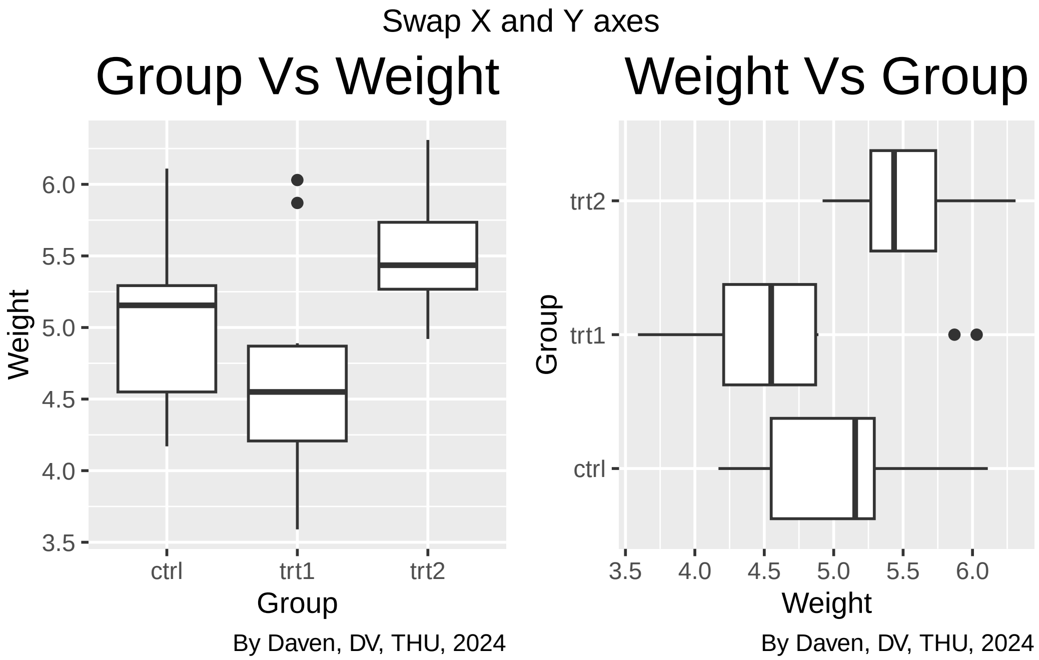
- Install `gridExtra` library.
- Prepare two plots: `p1`, `p2`.
- Use `grid.arrange()` function to arrange plots and add title.

```
library(ggplot2)
library(gridExtra)

p1 <- ggplot(PlantGrowth, aes (x = group, y = weight)) +
  geom_boxplot() +
  labs(title = "Group Vs Weight",
       x = "Group",
       y = "Weight",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

p2 <- ggplot(PlantGrowth, aes (x = group, y = weight)) +
  geom_boxplot() +
  coord_flip() +
  labs(title = "Weight Vs Group",
       x = 'Group',
       y = 'Weight',
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

grid.arrange(p1, p2, ncol = 2,
             top = 'Swap X and Y axes')
```



4. Setting the Range of a Continuous Axis

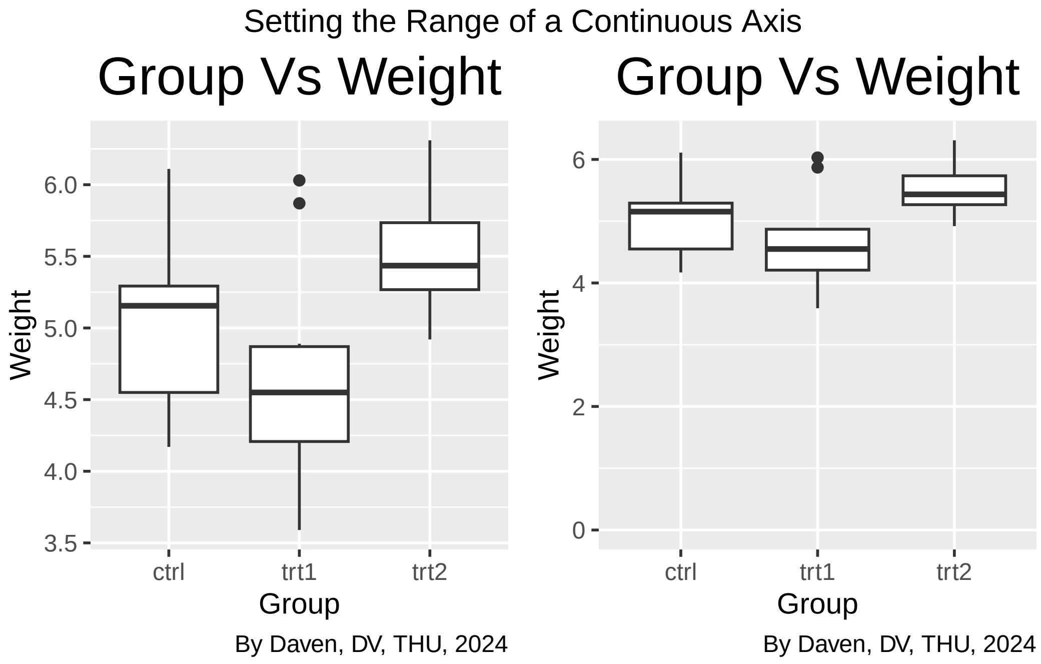
- `ylim()`: Sets the limits of the y-axis.

```
library(ggplot2)
library(gridExtra)

p1 <- ggplot(PlantGrowth, aes (x = group, y = weight)) +
  geom_boxplot() +
  labs(title = "Group Vs Weight",
       x = "Group",
       y = "Weight",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))
```

```r
p2 <- ggplot(PlantGrowth, aes (x = group, y = weight)) +
  geom_boxplot() +
  ylim(0, max(PlantGrowth$weight)) +
  labs(title = "Group Vs Weight",
       x = 'Group',
       y = 'Weight',
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

grid.arrange(p1, p2, ncol = 2,
             top = 'Setting the Range of a Continuous Axis')
```



5. Reverse a Continuous Axis

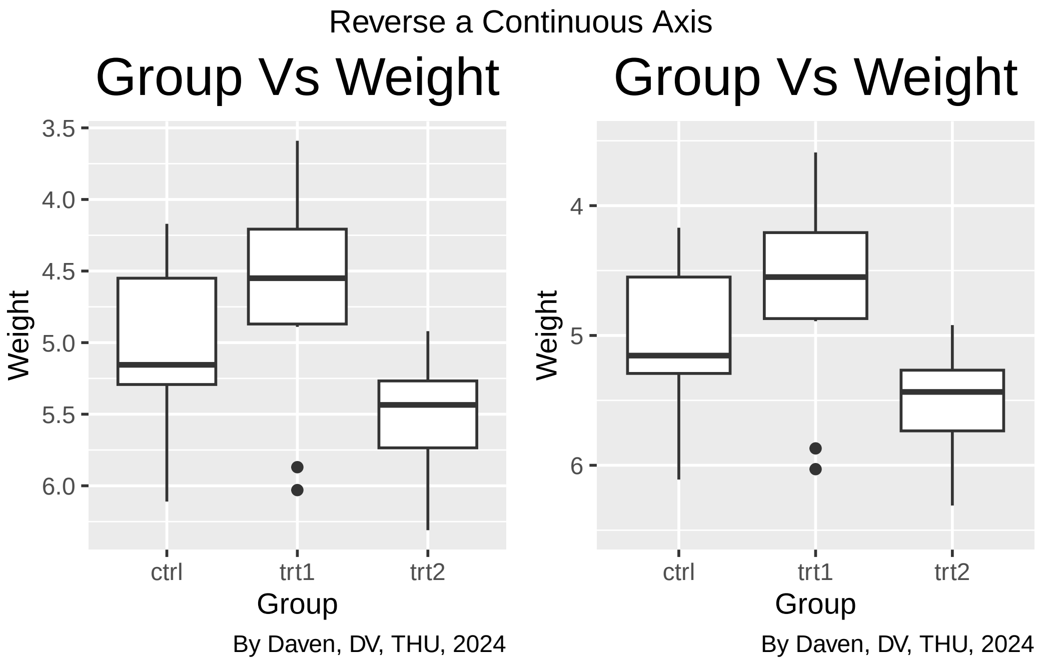
- `scale_y_reverse()` reverses the direction of the y-axis on a plot. So, what was previously the highest value on the y-axis becomes the lowest, and vice-versa.

```r
library(ggplot2)
library(gridExtra)

p1 <- ggplot(PlantGrowth, aes (x = group, y = weight)) +
  geom_boxplot() +
  scale_y_reverse() +
  labs(title = "Group Vs Weight",
       x = "Group",
       y = "Weight",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

p2 <- ggplot(PlantGrowth, aes (x = group, y = weight)) +
  geom_boxplot() +
  ylim(6.5, 3.5) +
  labs(title = "Group Vs Weight",
       x = 'Group',
       y = 'Weight',
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

grid.arrange(p1, p2, ncol = 2,
               top = 'Reverse a Continuous Axis')
```



6. Changing the Order of Items on a Categorial Axis

- `grid.arrange()` function from the gridExtra package in R to arrange four plots (named plot1, plot2, plot3, and plot4) into a single figure.

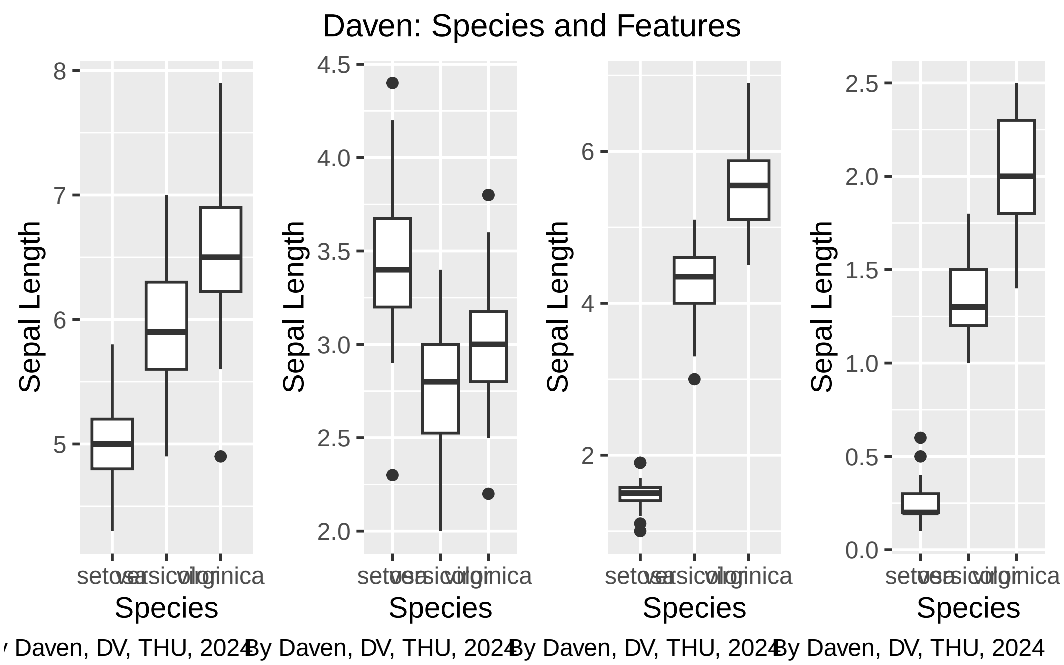
```r
p1 <- ggplot(iris, aes(x = Species, y = Sepal.Length)) +
  geom_boxplot() +
  labs(x = "Species",
       y = "Sepal Length",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

p2 <- ggplot(iris, aes(x = Species, y = Sepal.Width)) +
  geom_boxplot() +
  labs(x = "Species",
       y = "Sepal Length",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

p3 <- ggplot(iris, aes(x = Species, y = Petal.Length)) +
  geom_boxplot() +
  labs(x = "Species",
       y = "Sepal Length",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

p4 <- ggplot(iris, aes(x = Species, y = Petal.Width)) +
  geom_boxplot() +
  labs(x = "Species",
       y = "Sepal Length",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

grid.arrange(p1, p2, p3, p4, ncol=4,
             top = 'Daven: Species and Features')
```



7. Setting the Scalling Ratio of the X- and Y- Axes

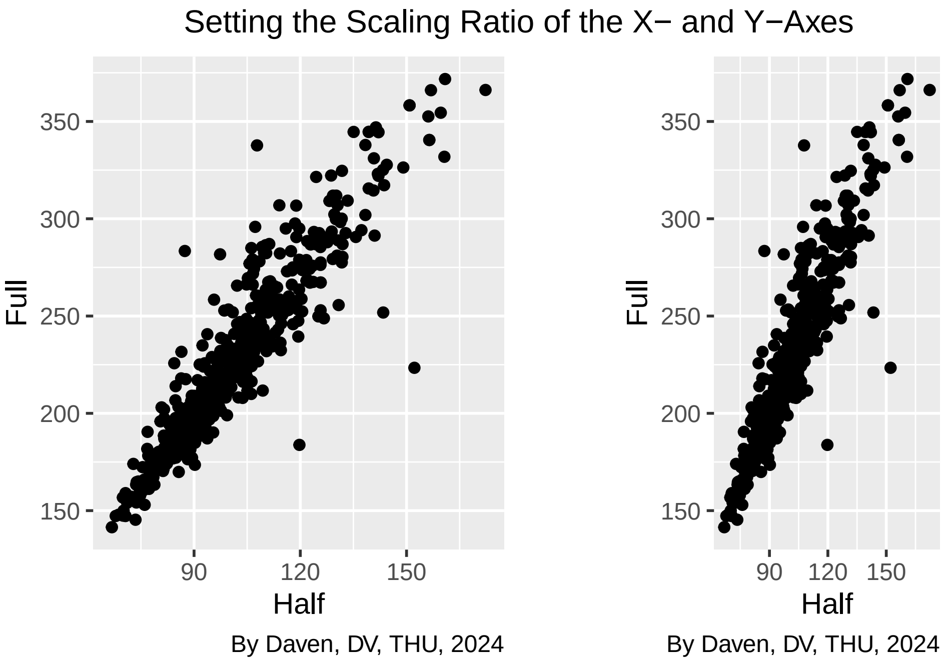
- `coord_fixed()` is a function that fixes the aspect ratio of the plot. This means that the units on the x-axis and y-axis will be of equal length.

```
library(gcookbook)

plot1 <- ggplot(marathon, aes(x = Half, y = Full)) +
  geom_point() +
  labs(x = "Half",
       y = "Full",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

plot2 <- ggplot(marathon, aes(x = Half, y = Full)) +
  geom_point() +
  coord_fixed() +
  labs(x = "Half",
       y = "Full",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))
```

```
grid.arrange(plot1, plot2, ncol=2,
             top = 'Setting the Scaling Ratio of the X- and Y-Axes')
```



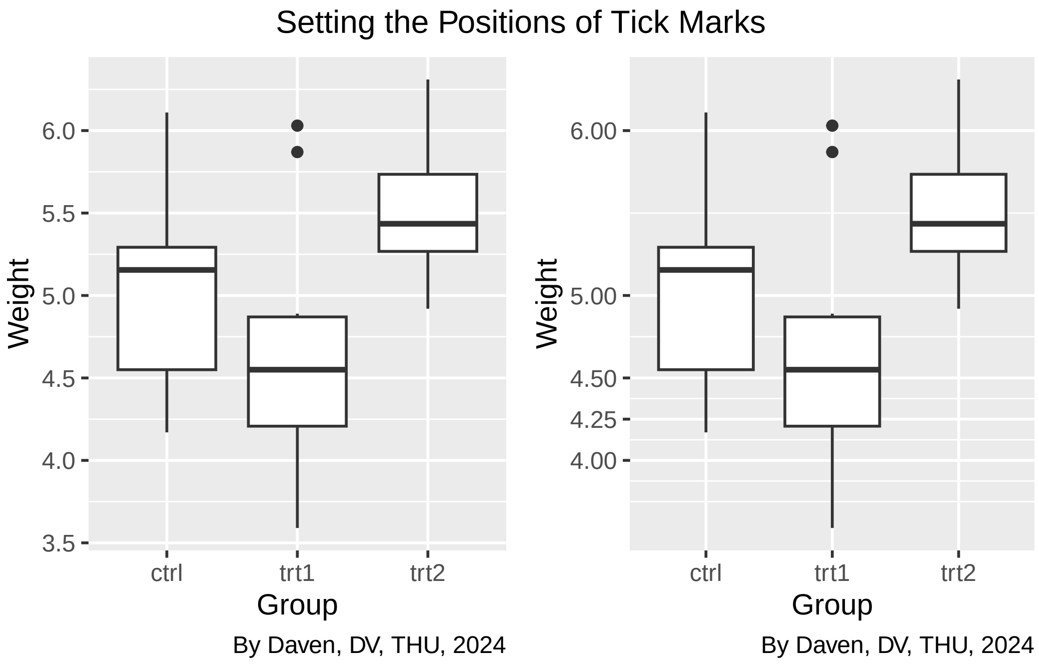
8. Setting the Positions of Tick Marks

- `scale_y_continuous()` is a function used to control the y-axis of a plot when the y-axis variable is continuous (meaning it can take on any value within a range, like temperature, weight, or time).

```
p1 <- ggplot(PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot() +
  labs(x = "Group",
       y = "Weight",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

p2 <- ggplot(PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot() +
  labs(x = "Group",
       y = "Weight",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  scale_y_continuous(breaks = c(4, 4.25, 4.5, 5, 6, 8))

grid.arrange(p1, p2, ncol = 2,
             top = 'Setting the Positions of Tick Marks')
```



9. Removing Tick Marks and Labels

- Use `grid.arrange()` from the gridExtra package to arrange the four plots (p1, p2, p3, p4) in a single row (ncol = 4) with a title at the top (top = 'Removing Tick Marks and Labels').

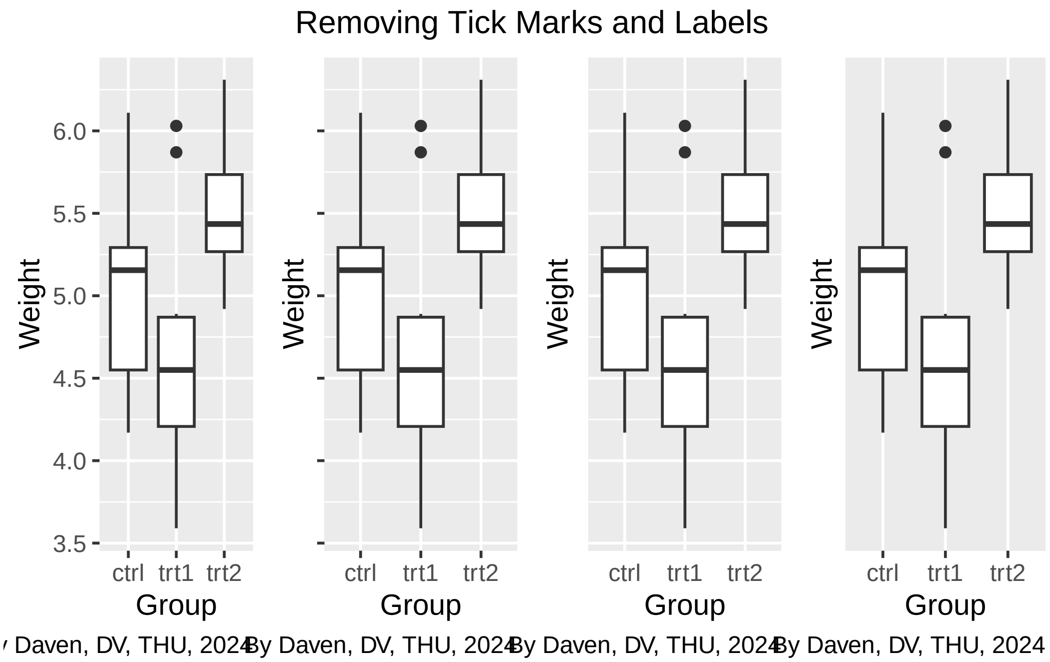
```
p1 <- ggplot(PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot() +
  labs(x = "Group",
       y = "Weight",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20))

p2 <- ggplot(PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot() +
  labs(x = "Group",
       y = "Weight",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  theme(axis.text.y = element_blank())

p3 <- ggplot(PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot() +
  labs(x = "Group",
       y = "Weight",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  theme(axis.ticks = element_blank(), axis.text.y = element_blank())

p4 <- ggplot(PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot() +
  labs(x = "Group",
       y = "Weight",
       caption = "By Daven, DV, THU, 2024") +
  theme(plot.title = element_text(hjust = 0.5, size = 20)) +
  scale_y_continuous(breaks = NULL)

grid.arrange(p1, p2, p3, p4, ncol = 4,
             top = 'Removing Tick Marks and Labels')
```



10. Changing the Text of Tick Labels

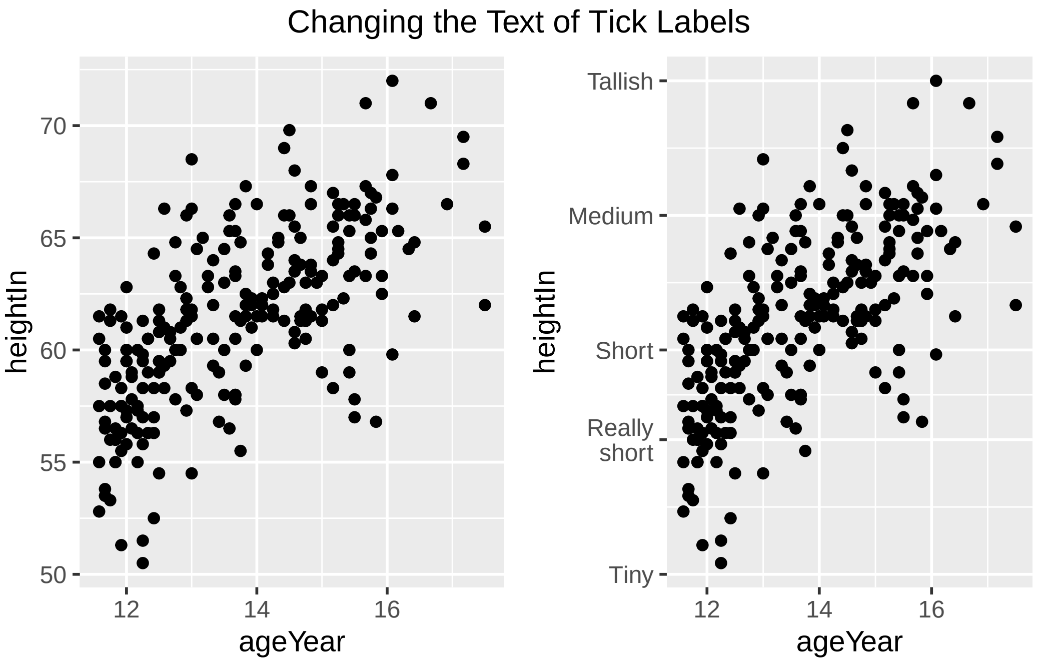
- `breaks`: Specifying the locations of the tick marks and labels on the y-axis (breaks = seq(min, max, by = interval)). This gives you more control over the spacing and readability of the axis labels.
- `labels`: Customizing the text of the axis labels (labels = function(x) …). This allows for formatting,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

```
library(gcookbook)

p1 <- ggplot(heightweight, aes(x = ageYear, y = heightIn)) +
  geom_point()

p2 <- 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")
  )

grid.arrange(p1, p2, ncol = 2,
             top = 'Changing the Text of Tick Labels')
```



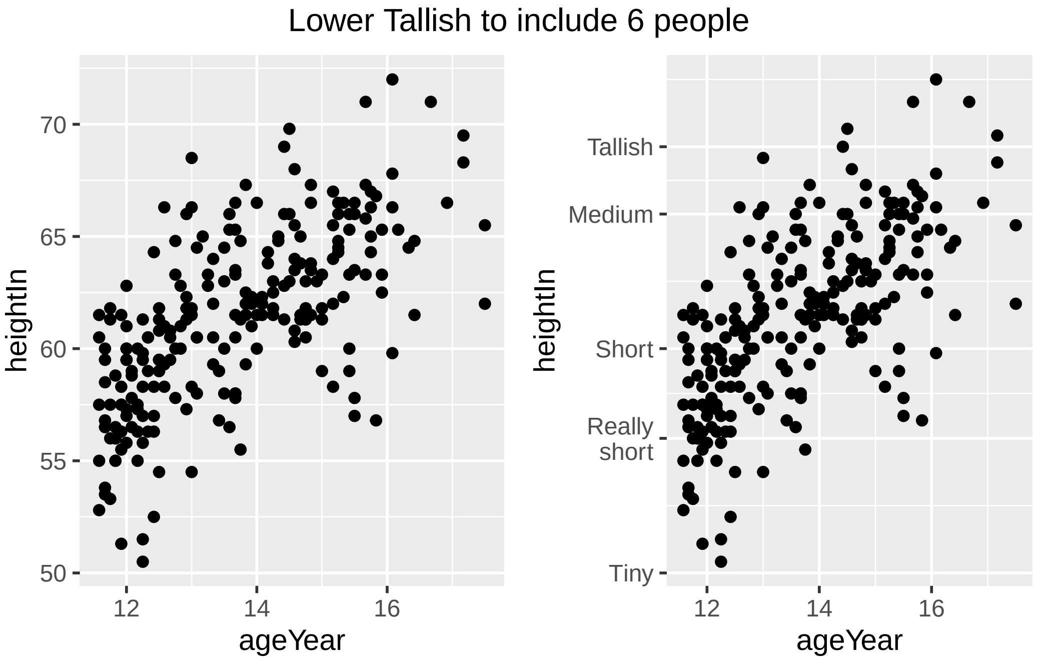
11. Lower 'Tallish' to include 6 people

- Use `grid.arrange()` from the gridExtra package to display both plots side by side (ncol = 2).
- Add a title to the grid using the `top` parameter: "Lower Tallish to include 6 people"

```r
p1 <- ggplot(heightweight, aes(x = ageYear, y = heightIn)) +
  geom_point()

p2 <- ggplot(heightweight, aes(x = ageYear, y = heightIn)) +
  geom_point() +
  scale_y_continuous(
    breaks = c(50, 56, 60, 66, 69),
    labels = c("Tiny", "Really\nshort", "Short", "Medium", "Tallish")
  )

grid.arrange(p1, p2, ncol = 2,
             top = 'Lower Tallish to include 6 people')
```

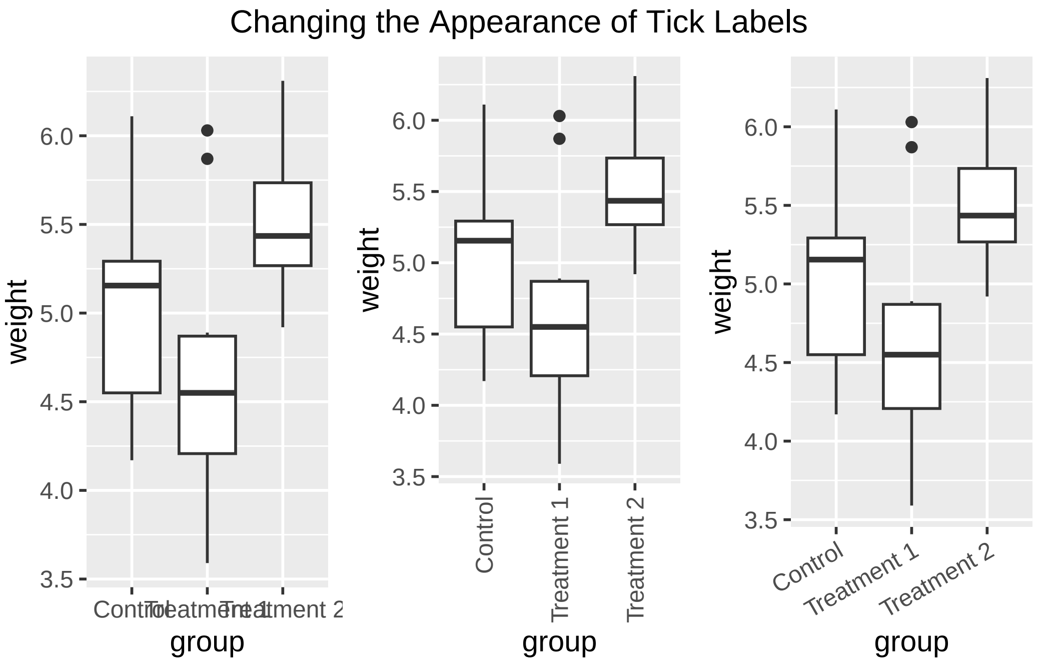


12. Changing the Appearance of Tick Labels

- `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")
  )
p1 <- pg_plot
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 = 'Changing the Appearance of Tick Labels')
```


Changing the Appearance of Tick Labels
weight
6.0
5.5
5.0
4.5
4.0
3.5
Control
Treatment 1
Treatment 2
group