Making plots with ggplot2: histograms, boxplots, line graphs

Emily Malcolm-White

```
# load packages
library(tidyverse)
```

`possum` data

The possum data frame consists of nine morphometric measurements on each of 104 mountain brushtail possums, trapped at seven Australian sites from Southern Victoria to central Queensland.

There are two different populations (`pop`): `Vic` (Victoria) and `other` (New South Wales or Queensland)

```
#Be sure to install the DAAG package if you've never used it before...
library(DAAG)
data("possum")
```

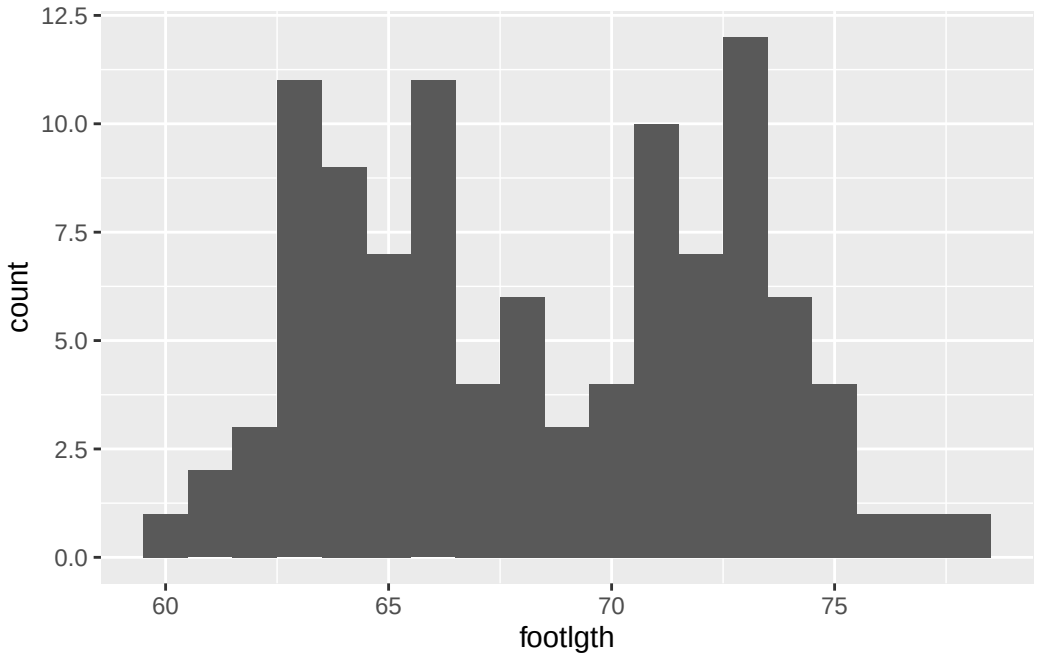
Histograms

> Tip
>
> Histograms are great for looking at the distributions of numeric variables

A boxplot for the footlength (`footlgth`) of all possums in this dataset:

```
ggplot(possum, aes(x=footlgth)) + #<1>
  geom_histogram(binwidth=1)  #<2>
```

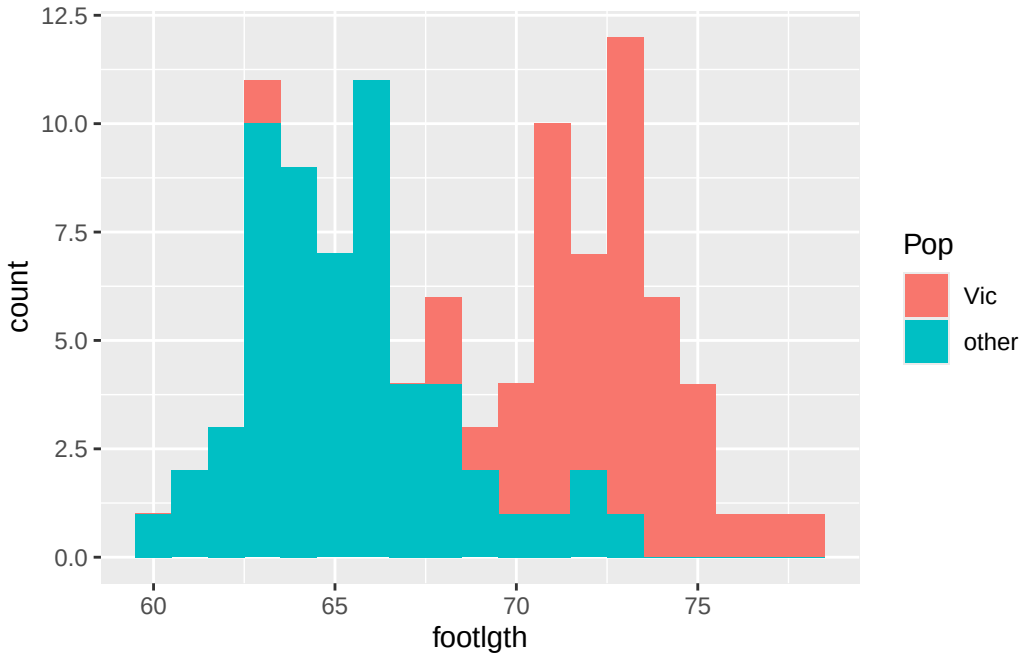
① only one x variable is needed
② you can adjust the binwidth, as needed



You can also customize by color:

```
ggplot(possum, aes(x=footlgth, fill=Pop)) + #<1>
  geom_histogram(binwidth=1)
```

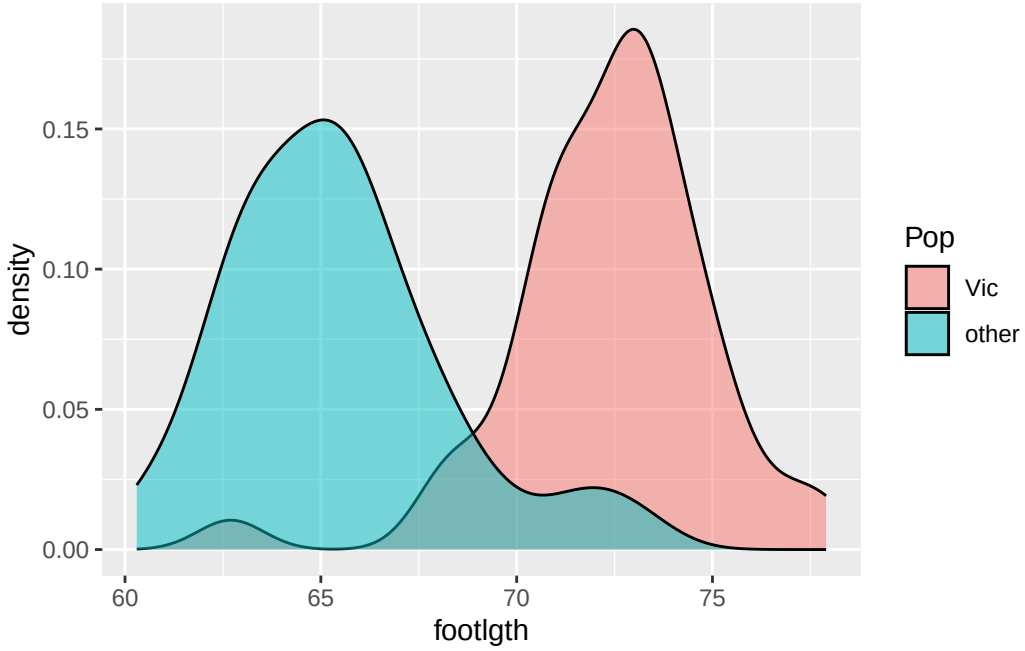
① adding `fill=Pop` customizes the colors based on which population the possums come from



Some people prefer to use `geom_density` for a smoother effect:

```
ggplot(possum, aes(x=footlgth, fill=Pop)) +
  geom_density(alpha=0.5)  #<1>
```

① Personally, I prefer density plots with slight transparency (`alpha=0.5`) so that you can fully see both plots



Boxplots

> **Tip**
>
> Boxplots are good for displaying the spread, central tendency, and distribution of one numeric variable.

A lone box-plot for one numeric variable (foot length) with some custom colors:

```
ggplot(possum, aes(y=footlgth)) +
  geom_boxplot(color="blue", fill="lightblue")  #<1>
```

① The `geom` for boxplot. The `color` arguments makes the outline of the boxplot blue and the the `fill` argument shades the inside of the inner quartile range.

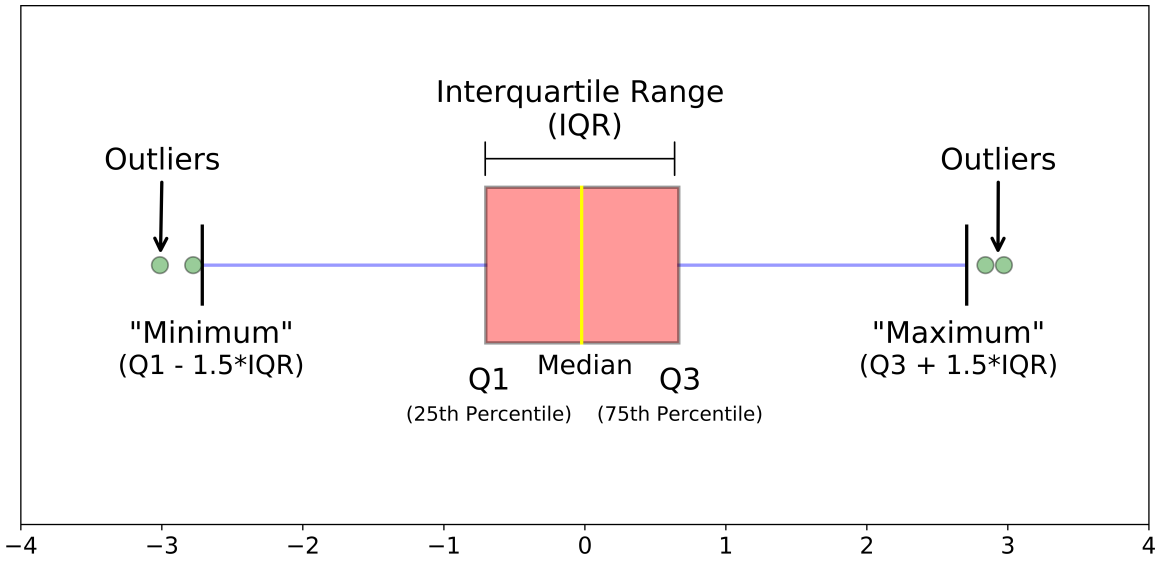
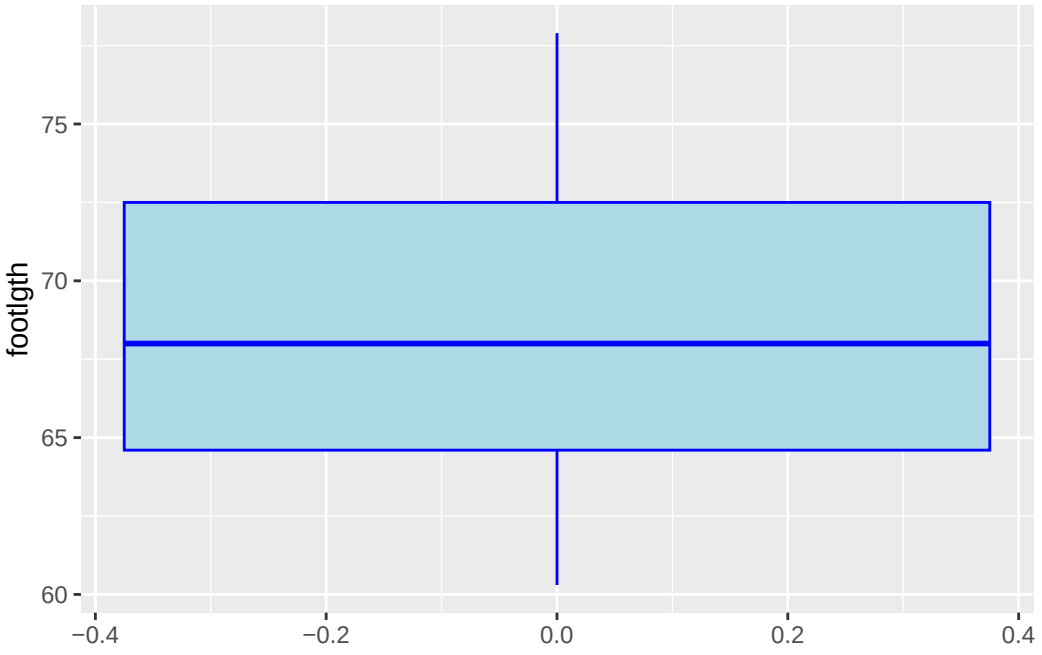
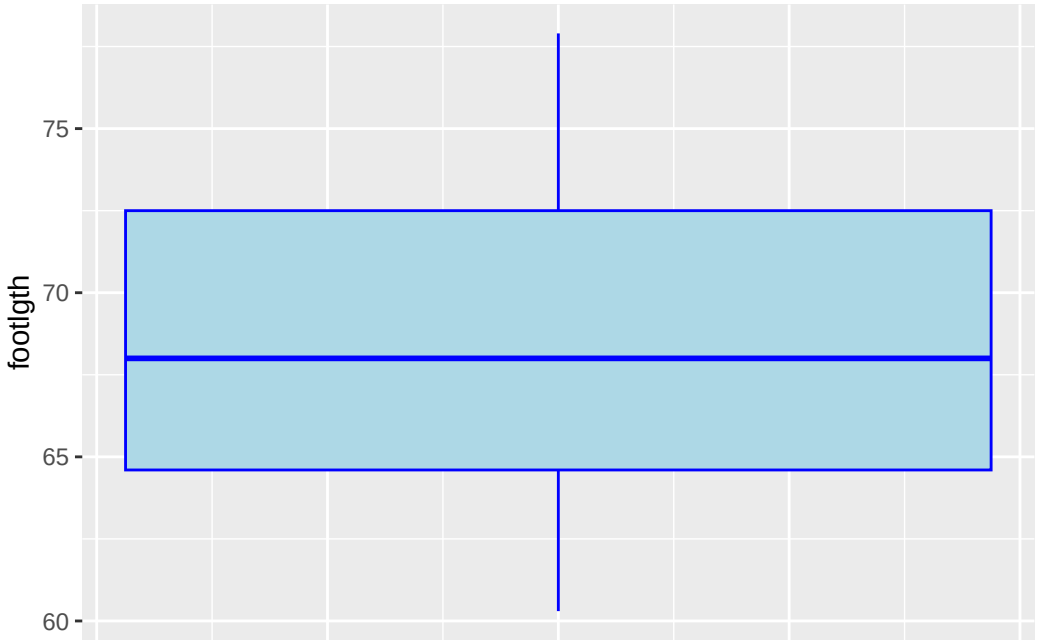


Figure 1: Credit: Michael Galarnyk



If only one boxplot, it puts weird numbers on the x axis, you may want to use the theme to hide these numbers.

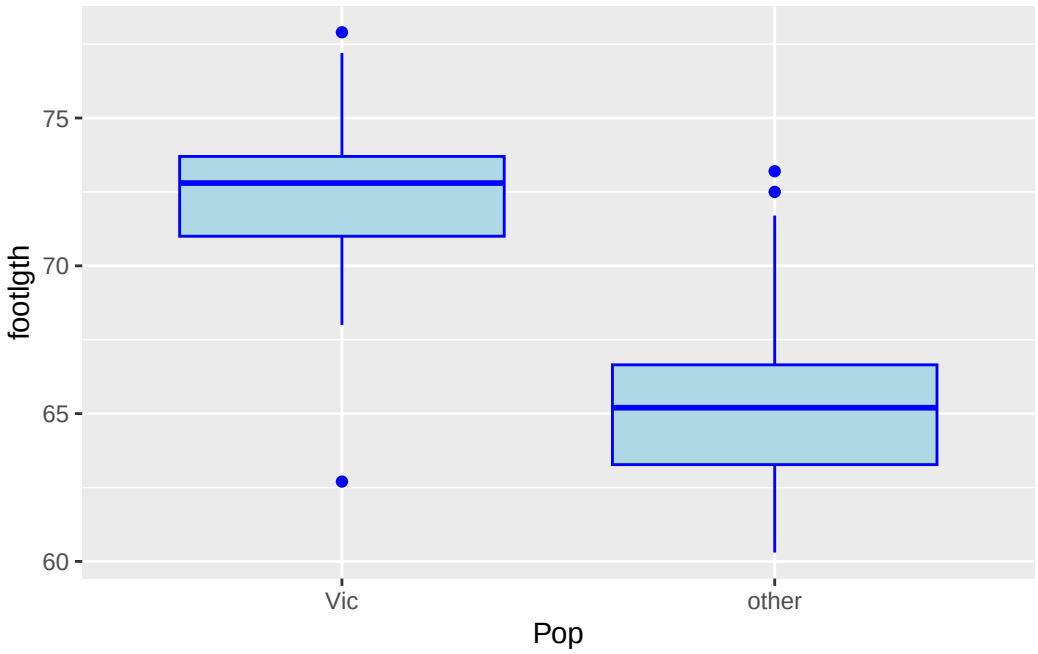
```r
ggplot(possum, aes(y = footlgth)) +
  geom_boxplot(color = "blue", fill = "lightblue") +
  theme(
    axis.text.x = element_blank(),  # Hide text
    axis.ticks.x = element_blank(), # Hide tick marks
    axis.title.x = element_blank()  # Hide axis title
  )
```



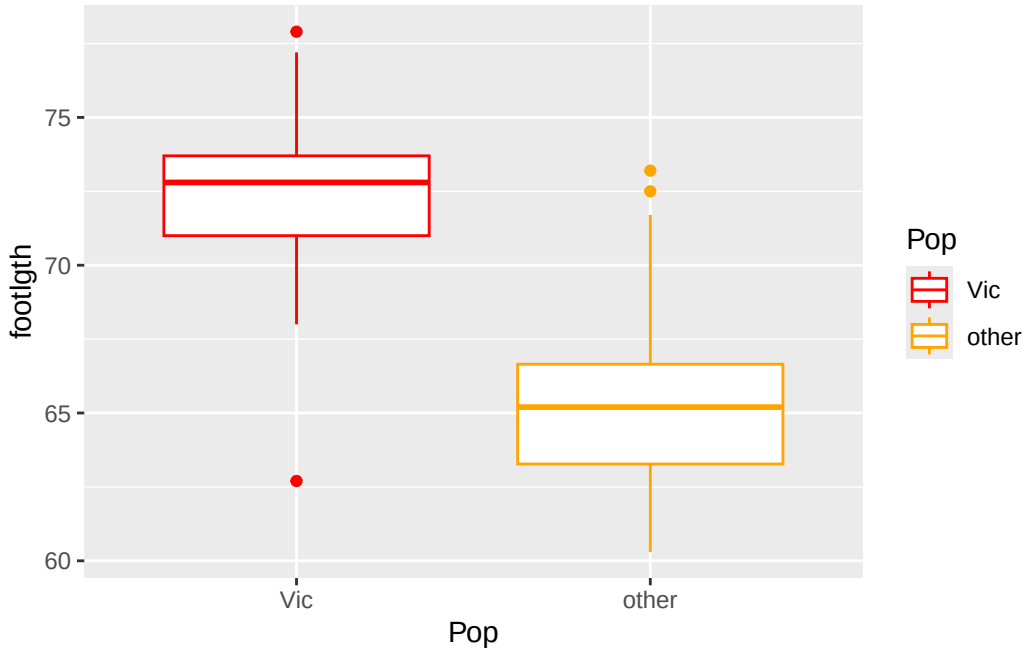
> **Tip**
>
> Side-by-side boxplots are good for displaying one categorical variable and one numeric variable. One advantage of boxplots over bar plots is that they are able to show a bit about the spread and distribution of the numeric variable!

A side-by-side boxplot to compare the foot lengths between the two populations of possums:

```r
# same color for both boxplots
ggplot(possum, aes(y=footlgth, x=Pop)) +
  geom_boxplot(color="blue", fill="lightblue")
```




```
# different color for both boxplots
ggplot(possum, aes(y=footlgth, x=Pop, color=Pop)) +
  geom_boxplot() +
  scale_color_manual(values=c("red", "orange"))
```



Line Graph

This dataset was produced from US economic time series data available from https://fred.stlouisfed.org/. Type `? economics` to learn more.

```
data("economics")
```

Create a line plot with the unemployment rate of the US over time:

```
# Create a line plot of unemployment over time
ggplot(economics, aes(x = date, y = unemploy)) +
  geom_line(color = "darkblue", size = 1) +
  labs(
    title = "U.S. Unemployment Over Time",
    x = "Year",
    y = "Number of Unemployed (in thousands)"
  )
```

> **Tip**
>
> In ggplot2, you use the **`group`** aesthetic in a `geom_line()` plot when you need to explicitly tell R how to group data points together into lines.

Spicy alert: I am creating a dataset here using some techniques that might be new to you. Don't worry so much about *how* I created the dataset, you should focus on what the dataset looks like.

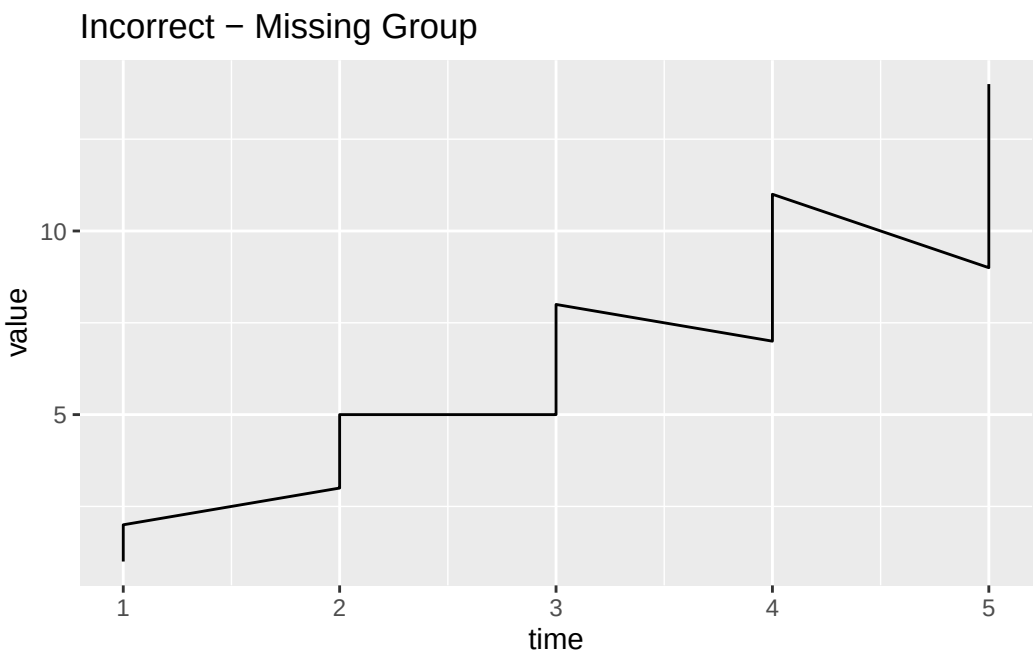
```r
# Example dataset
df <- data.frame(
  time = rep(1:5, 2),
  value = c(1, 3, 5, 7, 9, 2, 5, 8, 11, 14),
  category = rep(c("A", "B"), each = 5)
)

df
```

```
   time value category
1     1     1        A
2     2     3        A
3     3     5        A
4     4     7        A
5     5     9        A
6     1     2        B
7     2     5        B
8     3     8        B
9     4    11        B
10    5    14        B
```

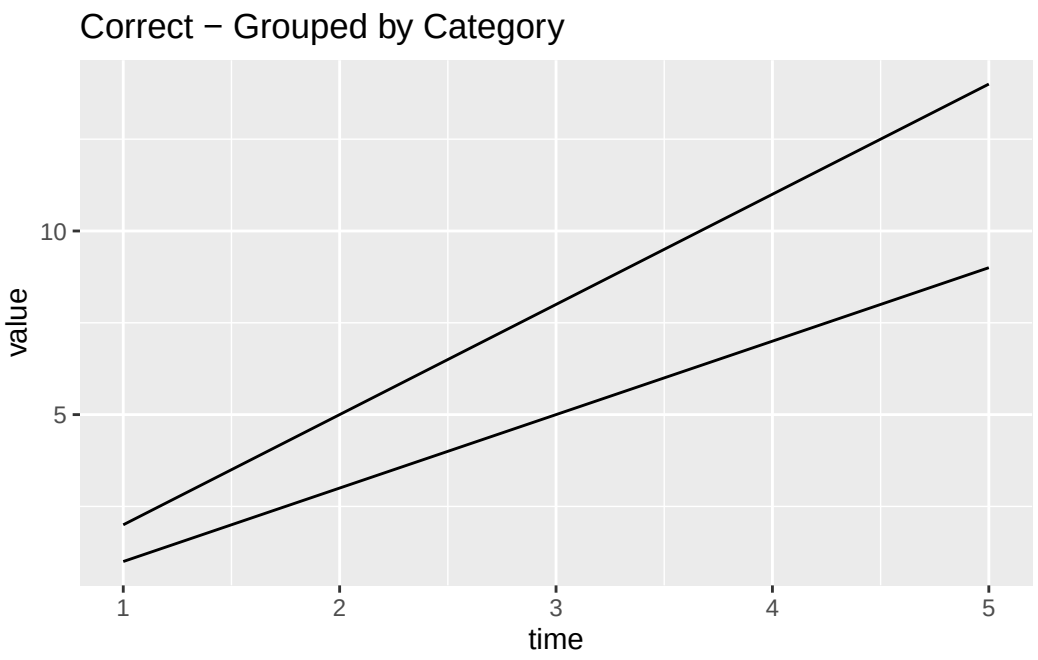
Without the needed `group` command

```r
# Incorrect: Only one line drawn without group
ggplot(df, aes(x = time, y = value)) +
  geom_line() +
  ggtitle("Incorrect - Missing Group")
```



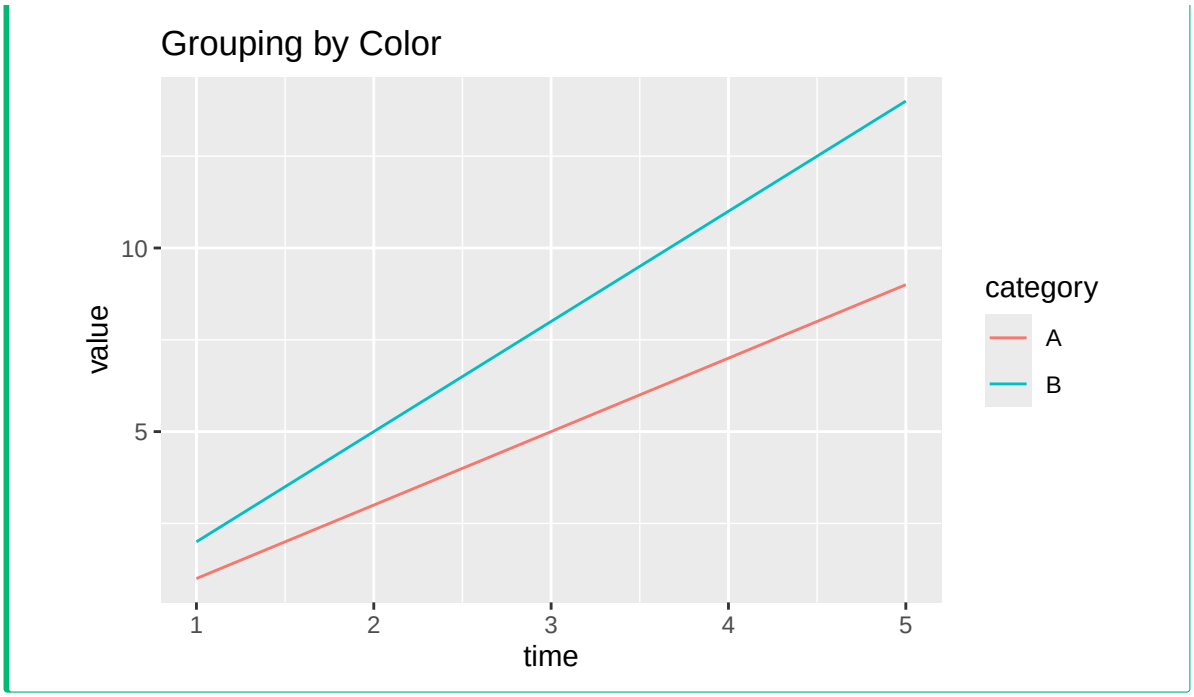
With the `group` command

```
# Correct: Separate lines for each category using group
ggplot(df, aes(x = time, y = value, group = category)) +
  geom_line() +
  ggtitle("Correct - Grouped by Category")
```



Using `color` (or `linetype`) instead

```
# Automatically groups by color
ggplot(df, aes(x = time, y = value, color = category)) +
  geom_line() +
  ggtitle("Grouping by Color")
```


Grouping by Color
value
10
5
1
2
3
4
5
time
category
A
B