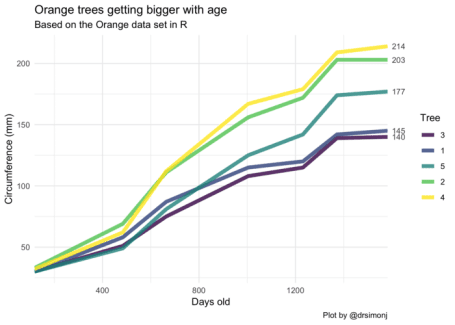
Here with a quick share on making great use of the secondary y axis with ggplot2 – super helpful if you’re plotting groups of time series!

Here’s an example of what I want to show you how to create (pay attention to the numbers of the right):

[](https://i0.wp.com/svbtleusercontent.com/a3LT3yKxA29K3Vc1aXnDsA0xspap.png?ssl=1)

**Setup**

To setup we’ll need the tidyverse package and the Orange data set that comes with R. This tracks the circumference growth of five orange trees over time.

library(tidyverse)

d <- Orange

head(d)

#> Grouped Data: circumference ~ age | Tree

#> Tree age circumference

#> 1 1 118 30

#> 2 1 484 58

#> 3 1 664 87

#> 4 1 1004 115

#> 5 1 1231 120

#> 6 1 1372 142

**Template code**

To create the basic case where the numbers appear at the end of your time series lines, your code might look something like this:

# You have a data set with:

# - GROUP colum

# - X colum (say time)

# - Y column (the values of interest)

DATA\_SET

# Create a vector of the last (furthest right) y-axis values for each group

DATA\_SET\_ENDS <- DATA\_SET %>%

group\_by(GROUP) %>%

top\_n(1, X) %>%

pull(Y)

# Create plot with `sec.axis`

ggplot(DATA\_SET, aes(X, Y, color = GROUP)) +

geom\_line() +

scale\_x\_continuous(expand = c(0, 0)) +

scale\_y\_continuous(sec.axis = sec\_axis(~ ., breaks = DATA\_SET\_ENDS))

**Let’s see it!**

Let’s break it down a bit. We already have our data set where the group colum is Tree, the X value is age, and the Y value is circumference.

So first get a vector of the last (furthest right) values for each group:

d\_ends <- d %>%

group\_by(Tree) %>%

top\_n(1, age) %>%

pull(circumference)

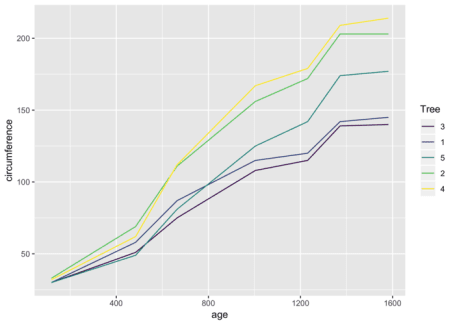
d\_ends

#> [1] 145 203 140 214 177

Next, let’s set up the basic plot without the numbers to see how each layer adds up.

ggplot(d, aes(age, circumference, color = Tree)) +

geom\_line()

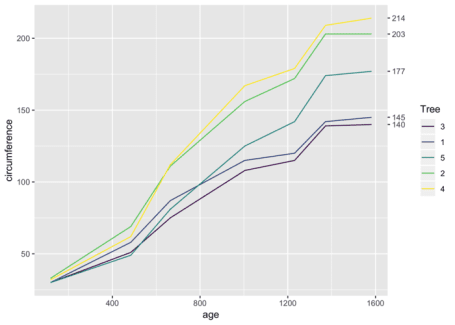
[](https://i1.wp.com/svbtleusercontent.com/4zKSnhW3JDUzwHbmKGBMAo0xspap.png?ssl=1)

Now we can use scale\_y\_\*, with the argument sec.axis to create a second axis on the right, with numbers to be displayed at breaks, defined by our vector of line ends:

ggplot(d, aes(age, circumference, color = Tree)) +

geom\_line() +

scale\_y\_continuous(sec.axis = sec\_axis(~ ., breaks = d\_ends))

[](https://i0.wp.com/svbtleusercontent.com/xfnR83uEfJP9nEz7ZToAkt0xspap.png?ssl=1)

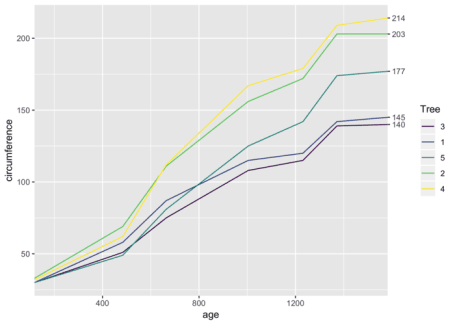
This is a great start, The only major addition I suggest is expanding the margins of the x-axis so the gap disappears. You do this with scale\_x\_\* and the expand argument:

ggplot(d, aes(age, circumference, color = Tree)) +

geom\_line() +

scale\_y\_continuous(sec.axis = sec\_axis(~ ., breaks = d\_ends)) +

scale\_x\_continuous(expand = c(0, 0))

[](https://i0.wp.com/svbtleusercontent.com/65yGdKt3cRfnBpJegdYRC0xspap.png?ssl=1)

**Polishing it up**

Like it? Here’s the code to recreate the first polished plot:

library(tidyverse)

d <- Orange %>%

as\_tibble()

d\_ends <- d %>%

group\_by(Tree) %>%

top\_n(1, age) %>%

pull(circumference)

d %>%

ggplot(aes(age, circumference, color = Tree)) +

geom\_line(size = 2, alpha = .8) +

theme\_minimal() +

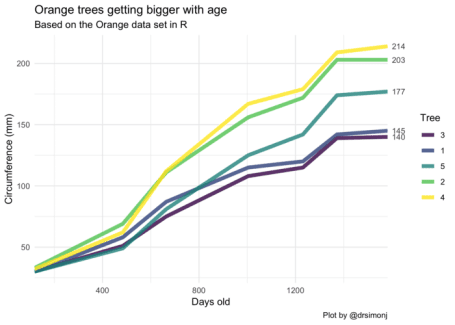
scale\_x\_continuous(expand = c(0, 0)) +

scale\_y\_continuous(sec.axis = sec\_axis(~ ., breaks = d\_ends)) +

ggtitle("Orange trees getting bigger with age",

subtitle = "Based on the Orange data set in R") +

labs(x = "Days old", y = "Circumference (mm)", caption = "Plot by @drsimonj")

[](https://i0.wp.com/svbtleusercontent.com/a3LT3yKxA29K3Vc1aXnDsA0xspap.png?ssl=1)