

# Using grattantheme

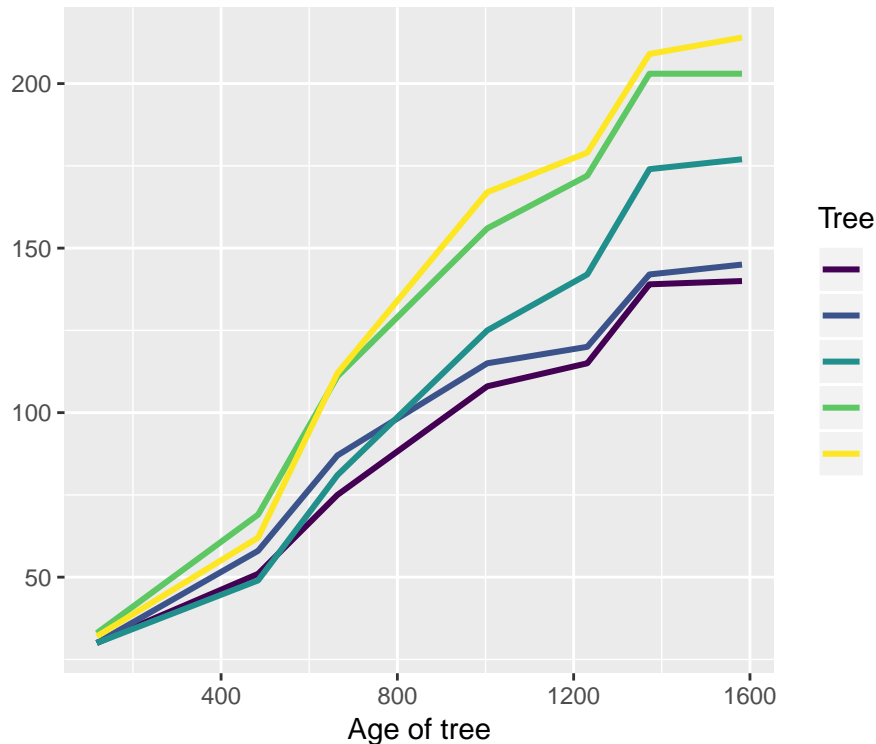
When creating a chart in R you have to:

- Choose a dataset;
- Map variables to chart aesthetics `aes()`;
- Choose a `geom_`.

For example, using the `Orange` dataset tracking the growth of five orange trees by age:

```
plot <- ggplot(Orange,
  aes(x = age,
      y = circumference,
      colour = Tree)) +
  geom_line() +
  labs(x = "Age of tree",
      y = "")
```

This successfully plots the data we want to plot:

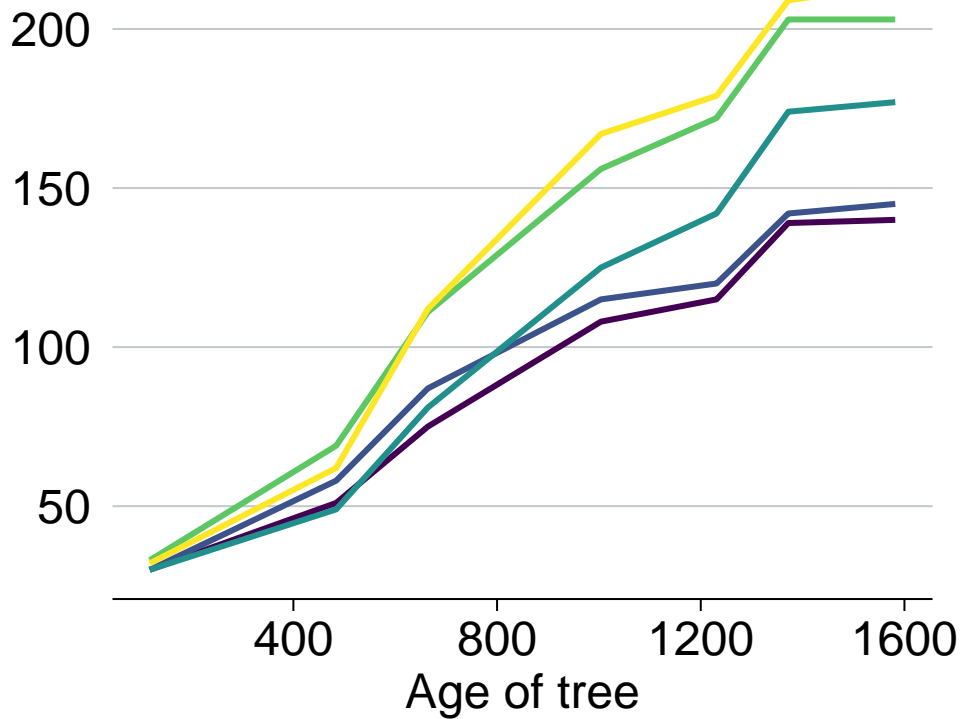


But it doesn't *look* like a Grattan chart. To adjust the *look* we use `theme` elements, like `axis.ticks.x = element_line(colour = "black")` to adjust the axis tickmarks on the x axis, and `panel.grid.major.x = element_blank()` to turn off vertical gridlines, and so on, and on, and on.

The function `theme_grattan()` contains all of the Grattan style-guide elements in one handy command. Combined with `grattan_colour_manual` to easily change colours of aesthetics, your R chart will be ready for a report or a slide in no time.

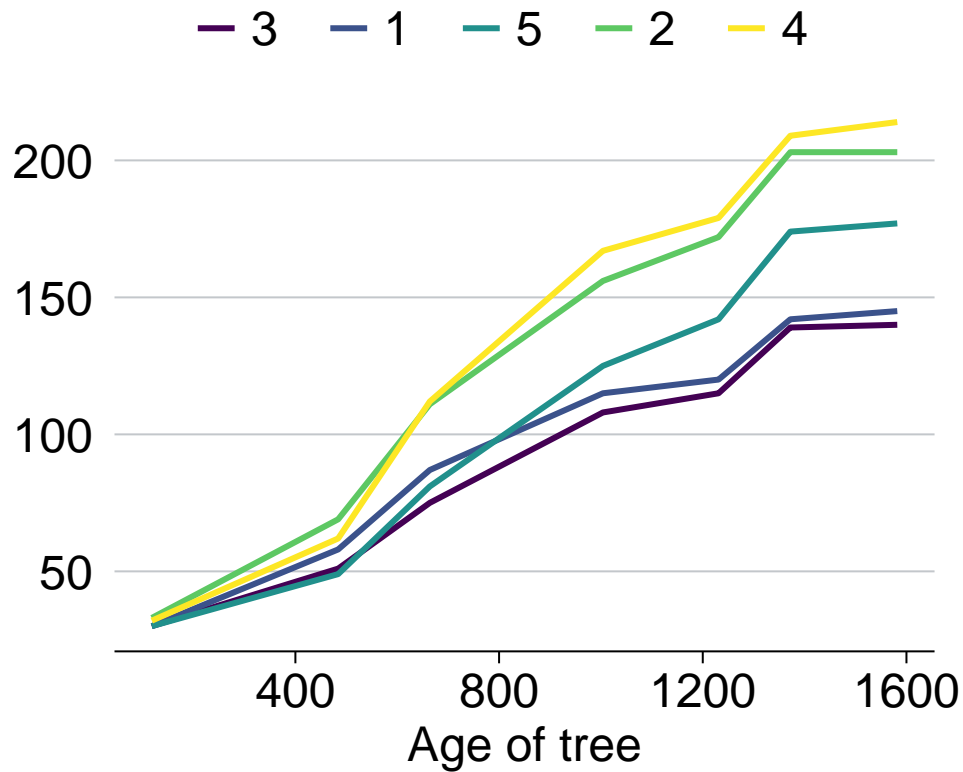
## Formatting theme elements with `theme_grattan()`

```
plot +  
  theme_grattan()
```



By default, `theme_grattan()` suppresses the legend to allow for clearer on-chart labelling. We can include the legend with the `legend` argument, which takes "off", "top", "bottom", "left" or "right":

```
plot +  
  theme_grattan(legend = "top")
```



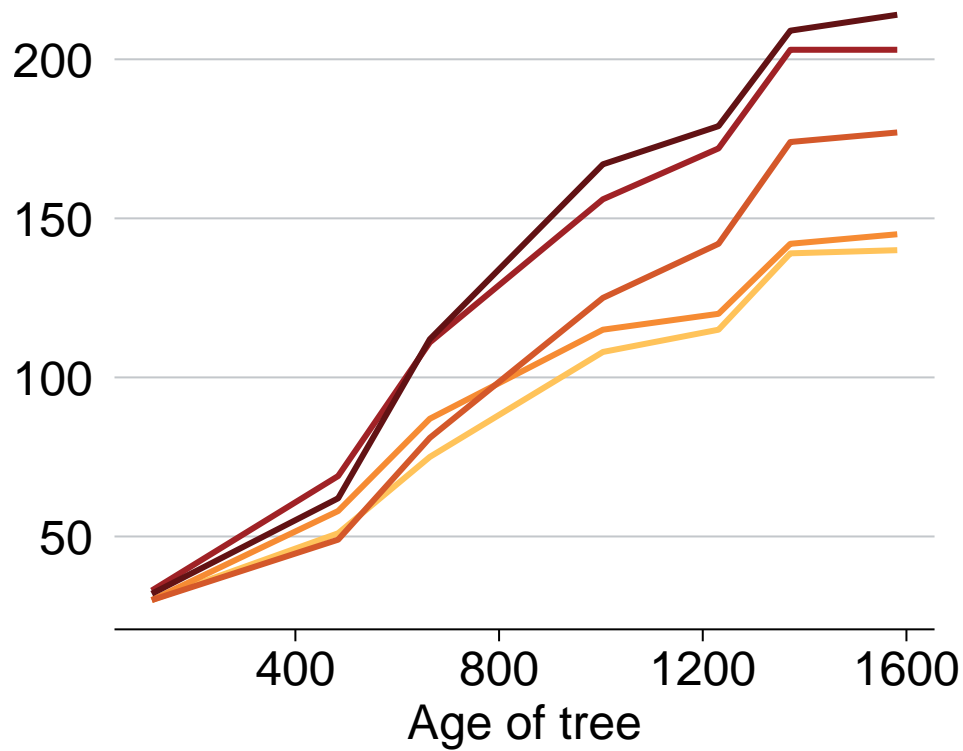
## Using Grattan colours

Grattan's colours are loaded with `grattantheme`. It has all the classics: ...

We can use the `grattan_colour_manual` function to change the colours of our colour aesthetic. In our example, we have five different trees each represented by a colour, so we set the number of colours to five

```
grattan_colour_manual(n = 5):
```

```
plot +  
  theme_grattan() +  
  grattan_colour_manual(n = 5)
```



We can reverse the order of the colours using the `reverse` argument:

```
plot +  
  theme_grattan() +  
  grattan_colour_manual(n = 5, reverse = TRUE)
```

