# Using grattantheme

## Contents

This vignette explains how to use grattantheme to quickly and consistently apply Grattan chart formatting to charts made in R using ggplot.

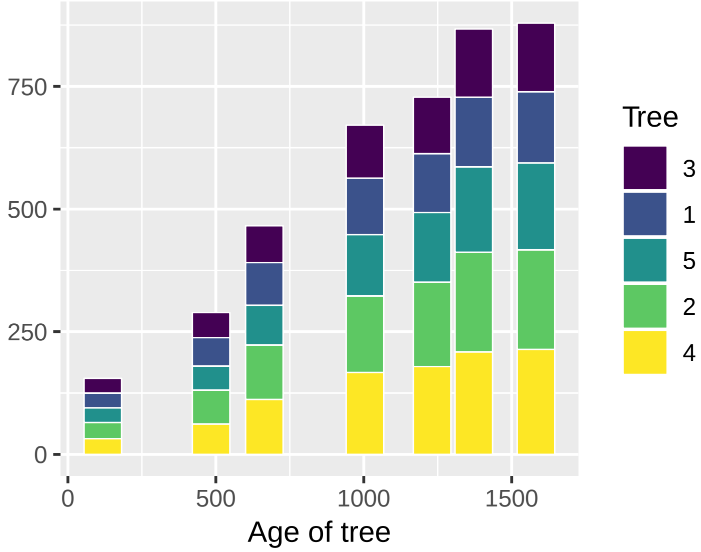
When creating a chart using ggplot we have to:

- 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,
                  fill = Tree)) +
        geom_bar(stat = "identity") +
        labs(x = "Age of tree",
             y = "",
             colour = "Tree")
```

This successfully plots the data we want to plot:

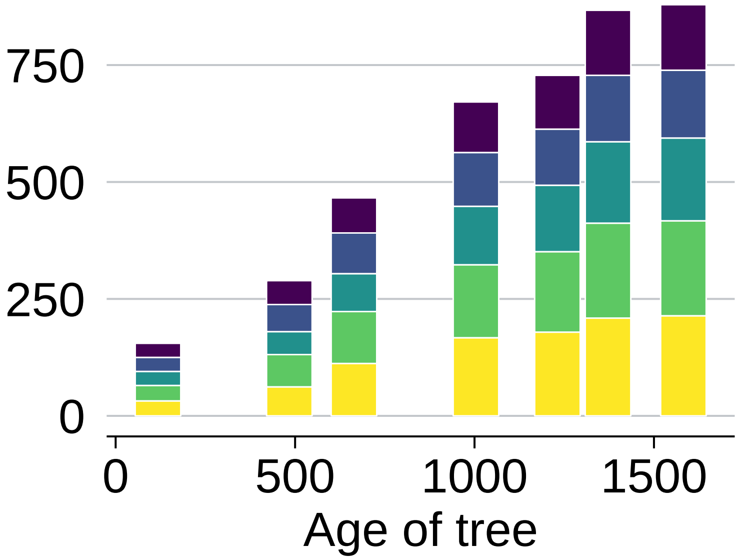


But it doesn't yet *look* like a Grattan chart. To adjust the *look* we adjust 'theme' elements, like axis.ticks.x = element_line(colour = "black") to adjust the axis tickmarks on the x axis; panel.grid.major.x = element_blank() to turn off vertical gridlines; and so on; and on; and on. We also need to adjust aesthetic colours to the Grattan palette; setting, for example, fill = "#F68B33". The grattantheme package contains tools and shortcuts to simplify this process.

### Formatting theme elements with `theme_grattan()`

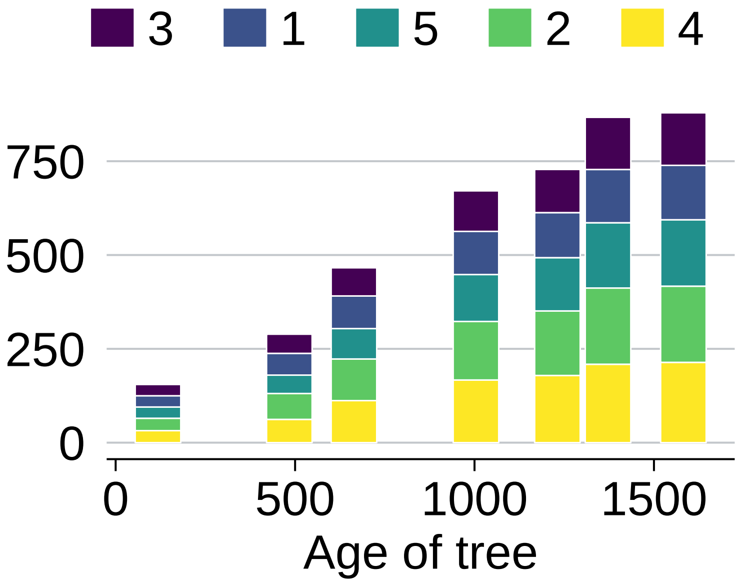
The function `theme_grattan()` contains all of the Grattan `theme` adjustments in one handy command. Combined with `grattan_colour_manual`, which easily changes colours of aesthetics, your `R` chart will be ready for a report or a slide in no time.

```
plot +
  theme_grattan()
```



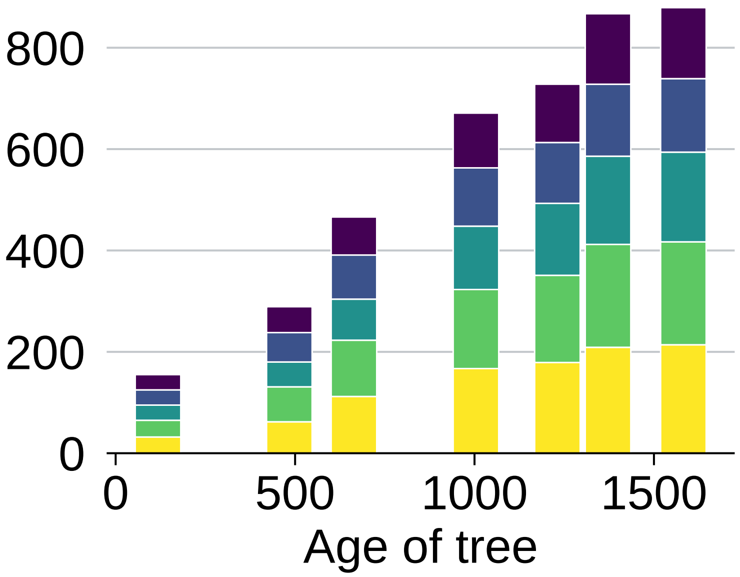
By default, `theme_grattan()` supresses the legend to allow for clearer on-chart labelling. We can include the legend with the `legend` argument, which takes `"off"`, `"top"`, `"bottom"`, `"left"` or `"right"`:

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



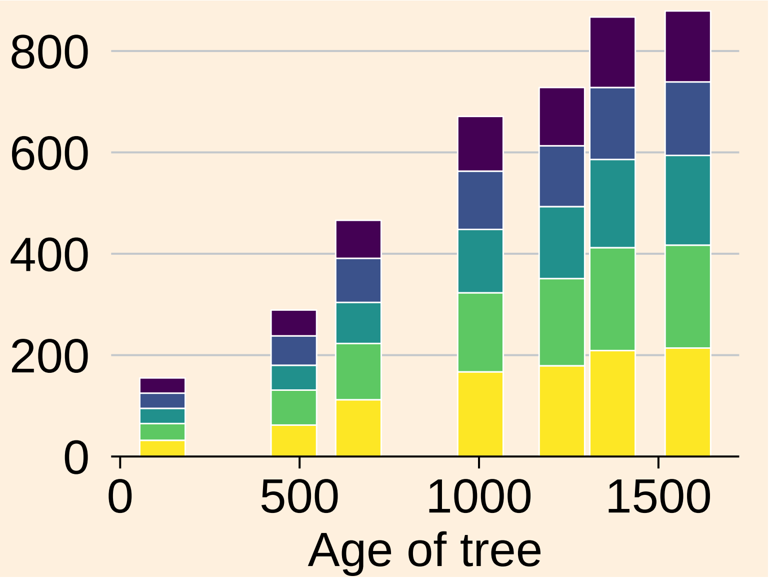
To align the y-axis with zero, change the y scale with `grattan_y_continuous()`:

```
plot +
  theme_grattan() +
  grattan_y_continuous()
```



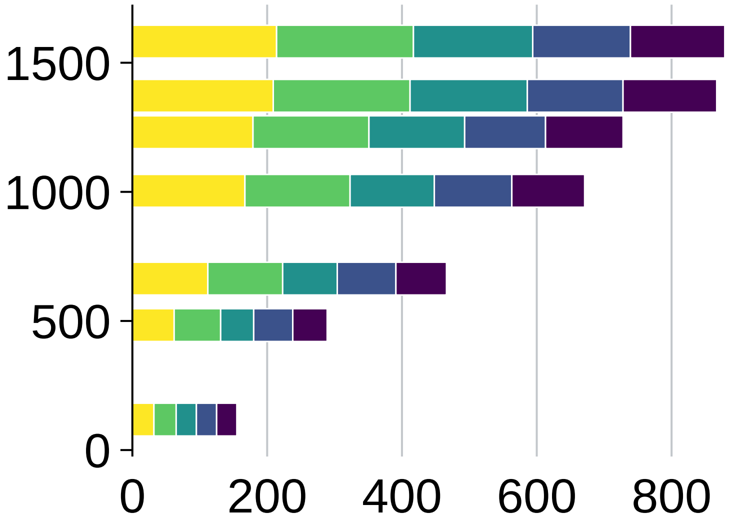
Sometimes we'll want a chart for a box in a report. We can change the background colour with the `background` argument:

```
plot +
  theme_grattan(background = "box") +
  grattan_y_continuous()
```



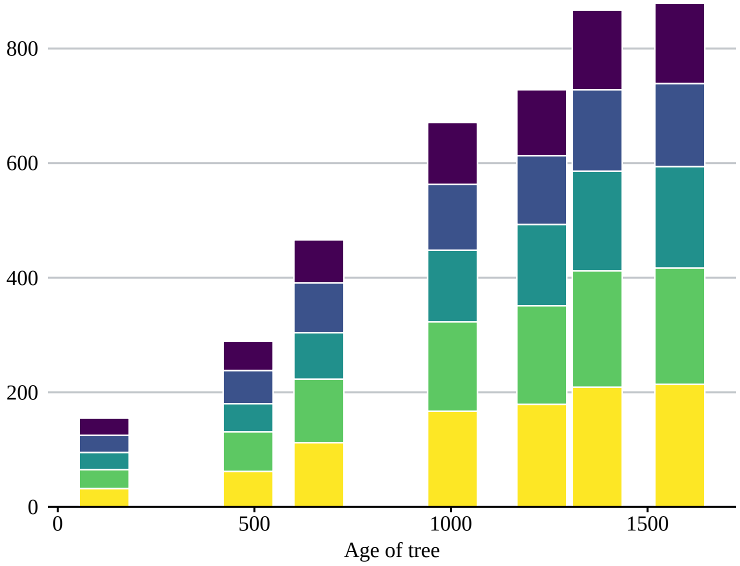
The standard Grattan rules for `x` and `y` axes flip if the chart is a horizontal bar chart. The `x` axis then follows the rules of the `y` axis, and vice-versa. If we are using a 'flipped' chart (imlemented with `coord_flipped()`), we can tell `theme_grattan` this is the case using the argument `fillped` set to `TRUE`.

```
plot +
  coord_flip() +
  theme_grattan(flipped = TRUE) +
  grattan_y_continuous()
```



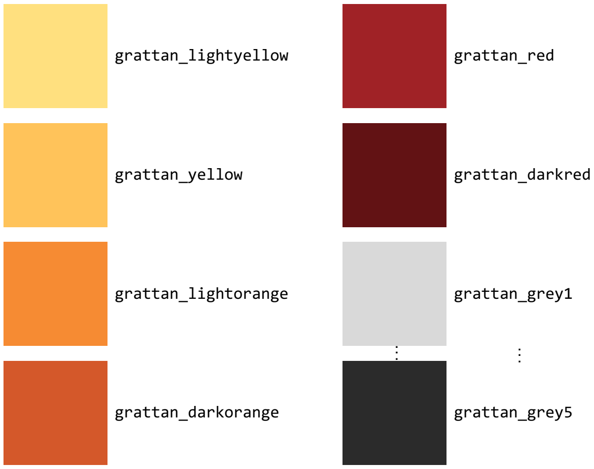
The final adjustments we can specify with `theme_grattan` are the font size and font family. The defaults meet Grattan formatting requirements, but if we do need to change them we can:

```
plot +
  theme_grattan(base_size = 8, base_family = "serif") +
  grattan_y_continuous()
```



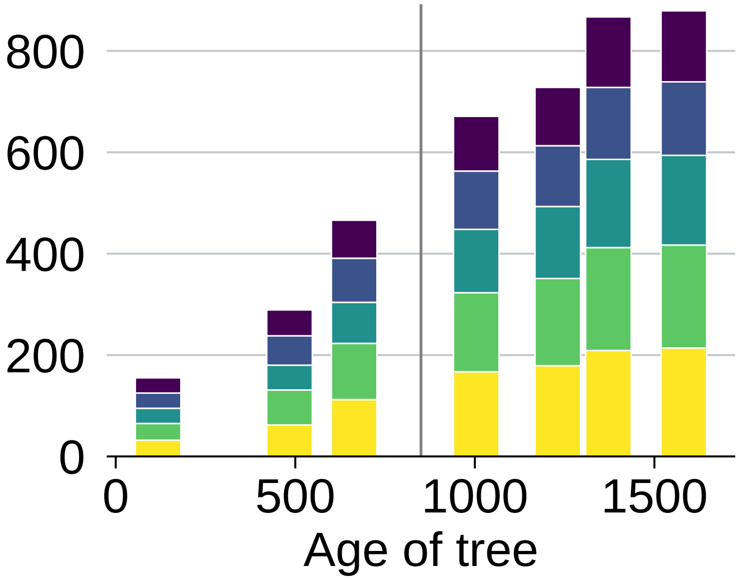
## Using Grattan colours

Grattan's colours are loaded with `grattantheme`. The HEX codes for individual Grattan colours can be called using `grattan_[colourname]`, eg `grattan_lightorange`. Colours names are taken from the chart-guide and are:



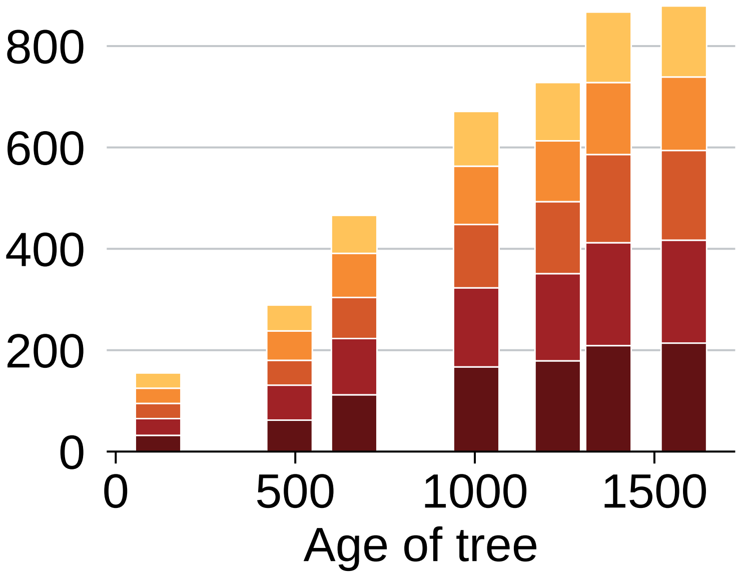
We can call a single colour:

```r
plot +
  geom_vline(xintercept = 850, colour = grattan_grey3) +
  theme_grattan() +
  grattan_y_continuous()
```



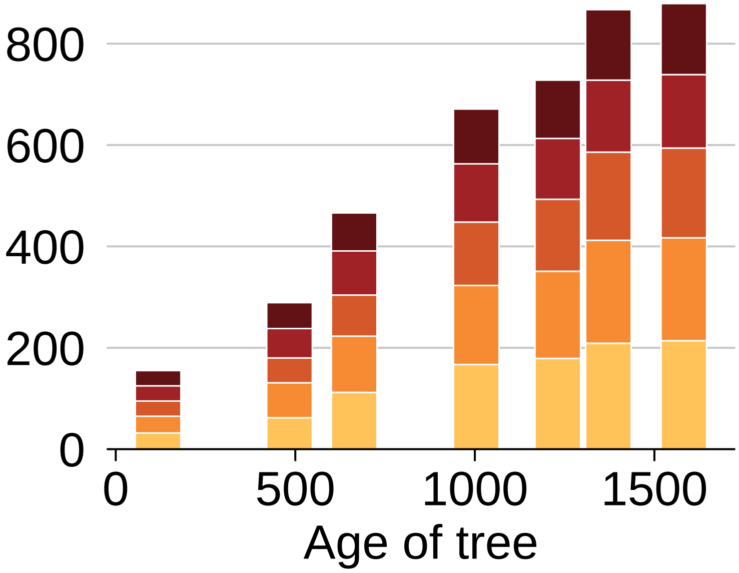
We can also use the `grattan_fill_manual()` or `grattan_colour_manual()` functions to change the colours of our fill or colour *aesthetics*. In our example, we have five different trees each represented by a colour, so we set the number of colours to five: `grattan_fill_manual(n = 5)`:

```r
plot +
  theme_grattan() +
  grattan_y_continuous() +
  grattan_fill_manual(n = 5)
```



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

```r
plot +
  theme_grattan() +
  grattan_y_continuous() +
  grattan_fill_manual(n = 5, reverse = TRUE)
```



Note that if you do not specify *enough* colours, will receive an error:

```r
plot +
  theme_grattan() +
  grattan_y_continuous() +
  grattan_fill_manual(n = 3)
#> Error: Insufficient values in manual scale. 5 needed but only 3 provided.
```

You won't receive an error if you specify *too many* colours. `ggplot` will choose the first `n` colours it needs and ignore the rest.

## Saving plots with grattan_save

The `grattan_save()` function implements `ggsave()` with Grattan defaults for different occasions. We specify the occasion with the `type` argument that can take the arguments:

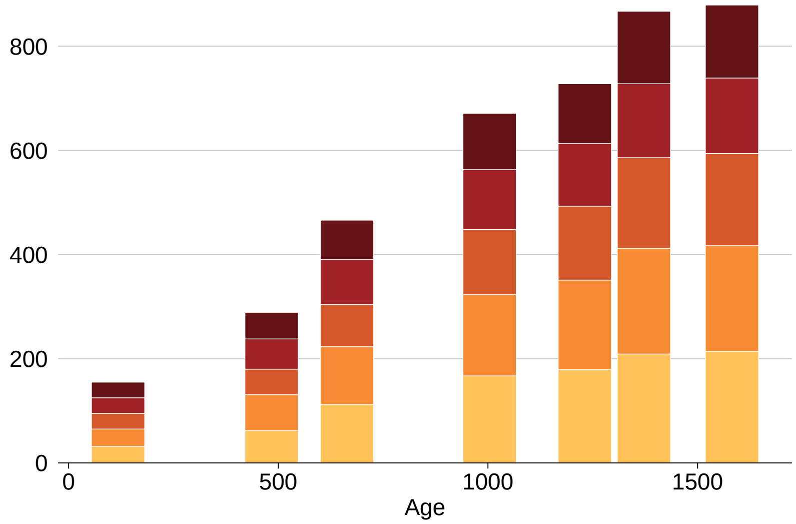
- `"normal"`: a standard report chart size, and the default. [**height** = 14.5cm x **width** = 22.16cm]
- `"wholecolumn`: a taller whole-column chart for reports. [22.16 x 22.16]
- `"fullpage"`: a full-page chart for reports. [22.16 x 44.32]
- `"tiny"`: a rarely-used short chart for reports. [11.08 x 22.16]
- `"fullslide"` : to produce charts on a full slide with the Grattan logo—to be used in articles or blogposts—use `"type = fullslide"`. This option allows notes and sources to accompany the saved figure.

The argument `filename` is required with an extension (`.pdf` is standard for Grattan charts). Like `ggsave`, the `ggplot2` object to be saved will default to the last plot, but you can specify something else with `object`.

Now we can save our Grattan-formatted graph as a normally-formatted report chart:

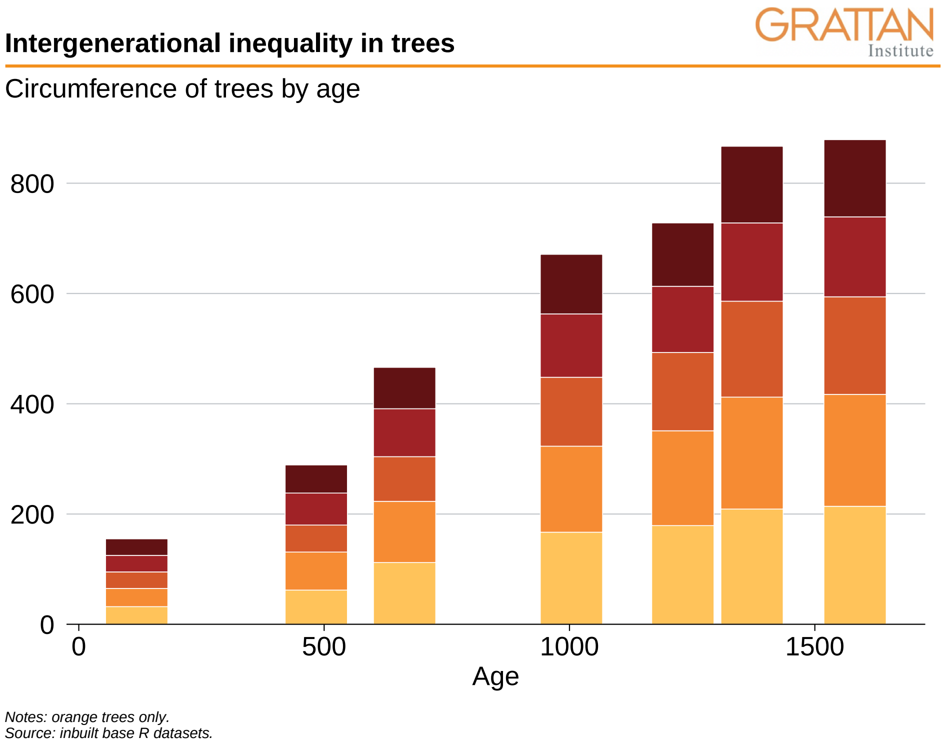
```
# Create and store the plot:
plot_final <- plot +
              theme_grattan() +
              grattan_y_continuous() +
              grattan_fill_manual(n = 5, reverse = TRUE) +
              labs(title = "Intergenerational inequality in trees",
                 subtitle = "Circumference of trees by age",
                 x = "Age",
                 y = "",
                 caption = "Notes: orange trees only. Source: inbuilt base R datasets.")

# Save the plot
grattan_save(filename = "tree_plot.pdf",
             object = plot_final,
             type = "normal")
```

Which produces a chart that can fit into a report. Note that the title, subtitle and notes/sources have been removed.



If we wanted to save a full-slide chart, complete with a title, subtitle, notes/sources and the Grattan slide format, we specify `type = "fullslide"`:

```
# Save the plot
grattan_save(filename = "tree_plot_fullslide.pdf",
             object = plot_final,
             type = "fullslide")
```



Intergenerational inequality in trees

Circumference of trees by age

*Notes: orange trees only.*
*Source: inbuilt base R datasets.*