

Modifying ggplot2 themes

`theme()` with options to override default

Overall	
<code>line</code>	all line elements (<code>element_line</code>)
<code>rect</code>	all rectangular elements (<code>element_rect</code>)
<code>text</code>	all text elements (<code>element_text</code>)
<code>title</code>	all title elements: plot, axes, legends (<code>element_text</code> ; inherits from <code>text</code>)
<code>aspect.ratio</code>	aspect ratio of the panel
Axes	
<code>axis.title</code>	label of axes (<code>element_text</code> ; inherits from <code>text</code>)
<code>axis.title.x</code>	x axis label (<code>element_text</code> ; inherits from <code>axis.title</code>)
<code>axis.title.x.top</code>	x axis label on top axis (<code>element_text</code> ; inherits from <code>axis.title.x</code>)
<code>axis.title.y</code>	y axis label (<code>element_text</code> ; inherits from <code>axis.title</code>)
<code>axis.title.y.right</code>	y axis label on right axis (<code>element_text</code> ; inherits from <code>axis.title.y</code>)
<code>axis.text</code>	tick labels along axes (<code>element_text</code> ; inherits from <code>text</code>)
<code>axis.text.x</code>	x axis tick labels (<code>element_text</code> ; inherits from <code>axis.text</code>)
<code>axis.text.x.top</code>	x axis tick labels on top axis (<code>element_text</code> ; inherits from <code>axis.text.x</code>)
<code>axis.text.y</code>	y axis tick labels (<code>element_text</code> ; inherits from <code>axis.text</code>)
<code>axis.text.y.right</code>	y axis tick labels on right axis (<code>element_text</code> ; inherits from <code>axis.text.y</code>)
<code>axis.ticks</code>	tick marks along axes (<code>element_line</code> ; inherits from <code>line</code>)
<code>axis.ticks.x</code>	x axis tick marks (<code>element_line</code> ; inherits from <code>axis.ticks</code>)
<code>axis.ticks.y</code>	y axis tick marks (<code>element_line</code> ; inherits from <code>axis.ticks</code>)
<code>axis.ticks.length</code>	length of tick marks (unit)
<code>axis.line</code>	lines along axes (<code>element_line</code> ; inherits from <code>line</code>)
<code>axis.line.x</code>	line along x axis (<code>element_line</code> ; inherits from <code>axis.line</code>)
<code>axis.line.y</code>	line along y axis (<code>element_line</code> ; inherits from <code>axis.line</code>)
Legend	
<code>legend.background</code>	background of legend (<code>element_rect</code> ; inherits from <code>rect</code>)
<code>legend.margin</code>	the margin around each legend (margin)
<code>legend.spacing</code>	the spacing between legends (unit)
<code>legend.spacing.x</code>	the horizontal spacing between legends (unit); inherits from <code>legend.spacing</code>
<code>legend.spacing.y</code>	the horizontal spacing between legends (unit); inherits from <code>legend.spacing</code>
<code>legend.key</code>	background underneath legend keys (<code>element_rect</code> ; inherits from <code>rect</code>)
<code>legend.key.size</code>	size of legend keys (unit)
<code>legend.key.height</code>	key background height (unit; inherits from <code>legend.key.size</code>)
<code>legend.key.width</code>	key background width (unit; inherits from <code>legend.key.size</code>)
<code>legend.text</code>	legend item labels (<code>element_text</code> ; inherits from <code>text</code>)
<code>legend.text.align</code>	alignment of legend labels (number from 0 (left) to 1 (right))

legend.title	title of legend (element_text; inherits from title)
legend.title.align	alignment of legend title (number from 0 (left) to 1 (right))
legend.position	the position of legends ("none", "left", "right", "bottom", "top", or two-element numeric vector)
legend.direction	layout of items in legends ("horizontal" or "vertical")
legend.justification	anchor point for positioning legend inside plot ("center" or two-element numeric vector) or the justification according to the plot area when positioned outside the plot
legend.box	arrangement of multiple legends ("horizontal" or "vertical")
legend.box.just	justification of each legend within the overall bounding box, when there are multiple legends ("top", "bottom", "left", or "right")
legend.box.margin	margins around the full legend area, as specified using margin
legend.box.background	background of legend area (element_rect; inherits from rect)
legend.box.spacing	The spacing between the plotting area and the legend box (unit)
Panel	
panel.background	background of plotting area, drawn underneath plot (element_rect; inherits from rect)
panel.border	border around plotting area, drawn on top of plot so that it covers tick marks and grid lines. This should be used with fill=NA (element_rect; inherits from rect)
panel.spacing	spacing between facet panels (unit)
panel.spacing.x	horizontal spacing between facet panels (unit; inherits from panel.spacing)
panel.spacing.y	vertical spacing between facet panels (unit; inherits from panel.spacing)
panel.grid	grid lines (element_line; inherits from line)
panel.grid.major	major grid lines (element_line; inherits from panel.grid)
panel.grid.minor	minor grid lines (element_line; inherits from panel.grid)
panel.grid.major.x	vertical major grid lines (element_line; inherits from panel.grid.major)
panel.grid.major.y	horizontal major grid lines (element_line; inherits from panel.grid.major)
panel.grid.minor.x	vertical minor grid lines (element_line; inherits from panel.grid.minor)
panel.grid.minor.y	horizontal minor grid lines (element_line; inherits from panel.grid.minor)
panel.ontop	option to place the panel (background, gridlines) over the data layers. Usually used with a transparent or blank panel.background. (logical)
Plot	
plot.background	background of the entire plot (element_rect; inherits from rect)
plot.title	plot title (text appearance) (element_text; inherits from title) left-aligned by default
plot.subtitle	plot subtitle (text appearance) (element_text; inherits from title) left-aligned by default
plot.caption	caption below the plot (text appearance) (element_text; inherits from title) right-aligned by default
plot.margin	margin around entire plot (unit with the sizes of the top, right, bottom, and left margins)
Strips	
strip.background	background of facet labels (element_rect; inherits from rect)

<code>strip.placement</code>	placement of strip with respect to axes, either "inside" or "outside". Only important when axes and strips are on the same side of the plot.
<code>strip.text</code>	facet labels (<code>element_text</code> ; inherits from <code>text</code>)
<code>strip.text.x</code>	facet labels along horizontal direction (<code>element_text</code> ; inherits from <code>strip.text</code>)
<code>strip.text.y</code>	facet labels along vertical direction (<code>element_text</code> ; inherits from <code>strip.text</code>)
<code>strip.switch.pad.grid</code>	space between strips and axes when strips are switched (unit)
<code>strip.switch.pad.wrap</code>	space between strips and axes when strips are switched (unit)
Other	
<code>...</code>	additional element specifications not part of base <code>ggplot2</code> . If supplied <code>validate</code> needs to be set to <code>FALSE</code> .
<code>complete</code>	set this to <code>TRUE</code> if this is a complete theme, such as the one returned by <code>theme_grey()</code> . Complete themes behave differently when added to a <code>ggplot</code> object. Also, when setting <code>complete = TRUE</code> all elements will be set to inherit from blank elements.
<code>validate</code>	<code>TRUE</code> to run <code>validate_element</code> , <code>FALSE</code> to bypass checks.

In conjunction with the theme system, the `element_` functions specify the display of how non-data components of the plot are drawn.

- `element_blank()`: draws nothing, and assigns no space.
- `element_rect()`: borders and backgrounds.
- `element_line()`: lines.
- `element_text()`: text.
- `unit()`: size. for example `unit(3, "cm")`

Usage

```
margin(t = 0, r = 0, b = 0, l = 0, unit = "pt")
```

```
element_blank()
```

```
element_rect(fill = NULL, color = NULL, size = NULL, linetype = NULL,
             color = NULL, inherit.blank = FALSE)
```

```
element_line(color = NULL, size = NULL, linetype = NULL,
             lineend = NULL, color = NULL, arrow = NULL, inherit.blank = FALSE)
```

```
element_text(family = NULL, face = NULL, color = NULL, size = NULL,
             hjust = NULL, vjust = NULL, angle = NULL, lineheight = NULL,
             color = NULL, margin = NULL, debug = NULL, inherit.blank = FALSE)
```

rel(x)

t, r, b, l	Dimensions of each margin. (To remember order, think trouble).
unit	Default units of dimensions. Defaults to "pt" so it can be most easily scaled with the text.
fill	Fill color.
color, color	Line/border color. Color is an alias for color.
size	Line/border size in mm; text size in pts.
linetype	Line type. An integer (0:8), a name (blank, solid, dashed, dotted, dotdash, longdash, twodash), or a string with an even number (up to eight) of hexadecimal digits which give the lengths in consecutive positions in the string.
inherit.blank	Should this element inherit the existence of an element_blank among its parents? If TRUE the existence of a blank element among its parents will cause this element to be blank as well. If FALSE any blank parent element will be ignored when calculating final element state.
lineend	Line end Line end style (round, butt, square)
arrow	Arrow specification, as created by arrow
family	Font family
face	Font face ("plain", "italic", "bold", "bold.italic")
hjust	Horizontal justification (in [0, 1])
vjust	Vertical justification (in [0, 1])
angle	Angle (in [0, 360])
lineheight	Line height
margin	Margins around the text. See margin for more details. When creating a theme, the margins should be placed on the side of the text facing towards the center of the plot.
debug	If TRUE, aids visual debugging by drawing a solid rectangle behind the complete text area, and a point where each label is anchored.
x	A single number specifying size relative to parent element.

Using System Fonts (On Windows)

```
install.packages("extrafont")
```

```
library(extrafont)
```

```
font_import() # import system fonts - only supports TrueType fonts  
# this can take a while
```

You may see some warnings, but you should be able to ignore them. After the fonts are imported, you can view the available fonts by running `fonts()` or `fonttable()`:

```
fonts()  
## [1] "Andale Mono"           "AppleMyungjo"  
## [3] "Arial Black"          "Arial"  
## [5] "Arial Narrow"         "Arial Rounded MT Bold"  
## [7] "Arial Unicode MS"     "Bangla Sangam MN"  
## [9] "Brush Script MT"      "Comic Sans MS"  
## [11] "Courier New"          "Georgia"  
## [13] "Gujarati Sangam MN"   "Impact"  
## ...
```

```
# This will show more detailed information about fonts  
fonttable()
```

```
# access system fonts
```

```
# This registers fonts so that they can be used with the pdf, postscript,  
# or Windows bitmap output device. It must be run once in each R session.  
loadfonts(device="win") # can be "pdf", "postscript", or "win"
```

example

```
ggplot(data=mtcars, aes(x=wt, y=mpg, color=factor(am))) +  
  geom_point() +  
  facet_wrap(~am) +  
  theme(strip.background=element_rect(fill="skyblue", color="white"),  
        strip.text=element_text(color="white", face="bold", hjust=.1, family="Rockwell"))
```