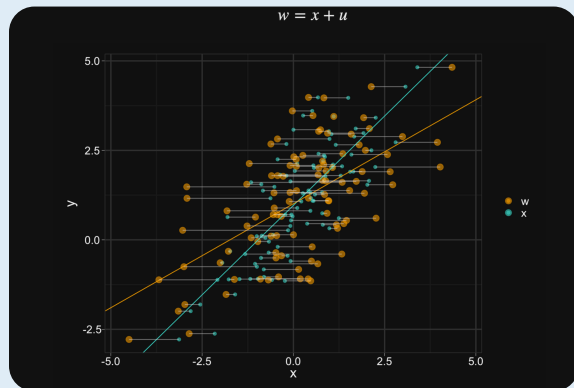


# Moving beyond theme\_gray CHEAT SHEET

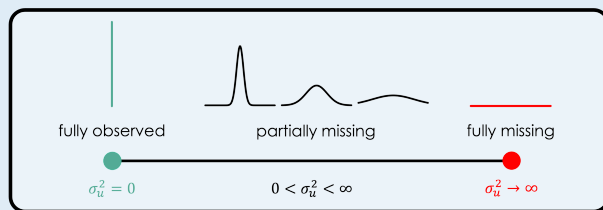
Exploring customized themes for improved visualizations in ggplot2

## Why modify the theme?

- Having a consistent graphical expression throughout a paper or thesis (possibility of matching fonts and colors).
- Modifying the background color of a plot to match (or contrast with) the background color of a presentation or poster.



- Removing distracting and unimportant elements of a plot for presentations or posters, creating more minimal visualizations.



## Organizing your themes

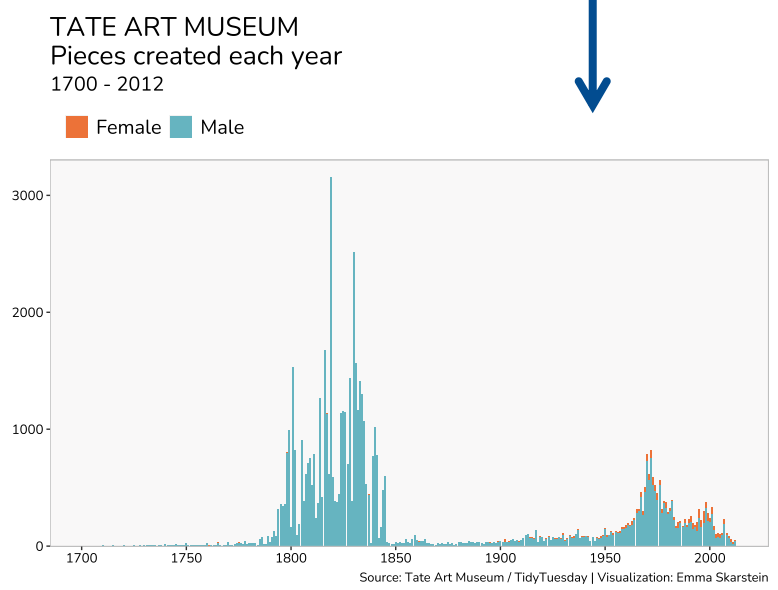
- **IN THE PLOT-CODE:** If your theme is specific to a single plot you are making, there is little to gain from specifying the theme outside of the plotting call itself.
- **IN A THEME-FUNCTION:** However, if you are going to apply the theme to two or more plots, you should specify it in it's own function.
- **IN A PACKAGE:** If there are specific theme changes you find yourself doing consistently (*like font, background colors, line preferences, etc.*) you can make a graphics-package with your favorite custom themes.

```
# Example of a custom theme
my_basic_theme <- function(base_size = 12,
                           base_family = 'sans'){
  return_theme <- theme(
    text = element_text(family = base_family,
                        size = base_size),
    axis.text = element_text(size = 10,
                             color = "black"),
    plot.title = element_text(size = 20,
                              margin = margin(b = 5)),
    plot.subtitle = element_text(size = 15,
                                  margin = margin(b = 10)),
    axis.title = element_blank(),
    legend.position = "top",
    legend.title = element_blank(),
    legend.justification = 0,
    legend.text = element_text(size = 15,
                                family = base_family),
    panel.grid = element_blank(),
    panel.background = element_rect(fill = "#F9F8F8"),
    panel.border = element_rect(color = "gray",
                                fill = NA))
  return(return_theme)
}
```

- Defaults
- Specifying font and size for various text elements
- Removing axis labels
- Various legend specifications
- Remove background grid, set a nice gray color, and set a slightly darker gray border

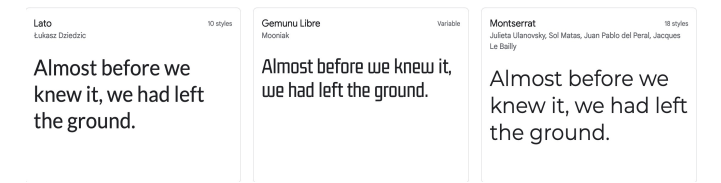
```
# This is how the theme is used in a plotting script:
ggplot(art_artist, aes(x = year, fill = gender)) +
  geom_bar() +
  scale_fill_manual(values = c("#EC7238", "#66B4C0", "#1E4A76")) +
  scale_x_continuous(breaks = seq(1700, 2012, by = 50),
                    expand = c(0.05, 0.05)) +
  scale_y_continuous(limits = c(0, 3300),
                    expand = c(0.001, 0)) +
  labs(title = "TATE ART MUSEUM\nPieces created each year",
       subtitle = "1700 - 2012",
       caption = "Source: Tate Art Museum / TidyTuesday |
               Visualization: Emma Skarstein") +
  my_basic_theme(base_family = "Nunito")
```

- The basic bar-plot
- Custom colors
- Scale axes, let y-axis start at 0
- Title, subtitle, and caption
- Finally, adding my theme!

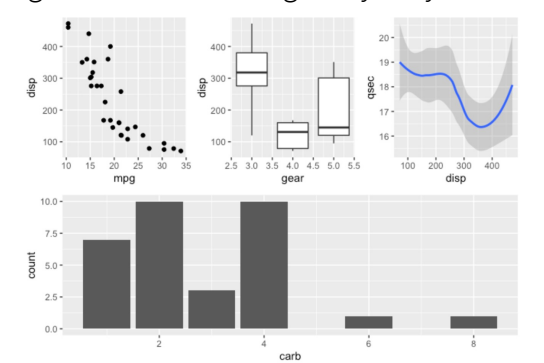


## Helpful packages/websites

- **FONTS:** [showtext](#) lets you easily set the font to any of the ones you find in the [Google font library](#).



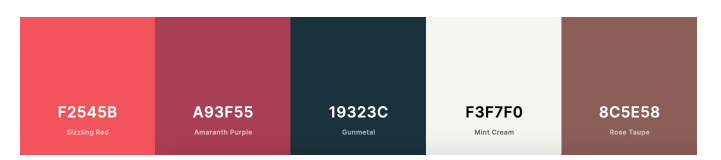
- **COMBINING PLOTS:** [patchwork](#) for combining plots. Themes can be set on the combined plot object, making consistent theming really easy.



- **TRANSPARENT BACKGROUND:** [ggpubr](#) has (among other things) a [transparent background theme](#).



- **COLORS:** [This article](#) by Lisa Charlotte Muth gives a thorough introduction to general principles in choosing colors. There are many good color palette generators out there, <https://colors.co/> is one example, if you just want a ready made palette, check [this list of ggplot2 color palettes](#).



- **EXAMPLE THEMES:** BBC has a [really well-documented theme](#) in their [bbplot](#) package. My personal theme is public on Github [here](#).