What is grafify?

grafify is a new R package for making great-looking ggplot2 graphs quickly in R. It has 19 plotting functions that simplify common ggplot graphs and provide color-blind friendly themes.

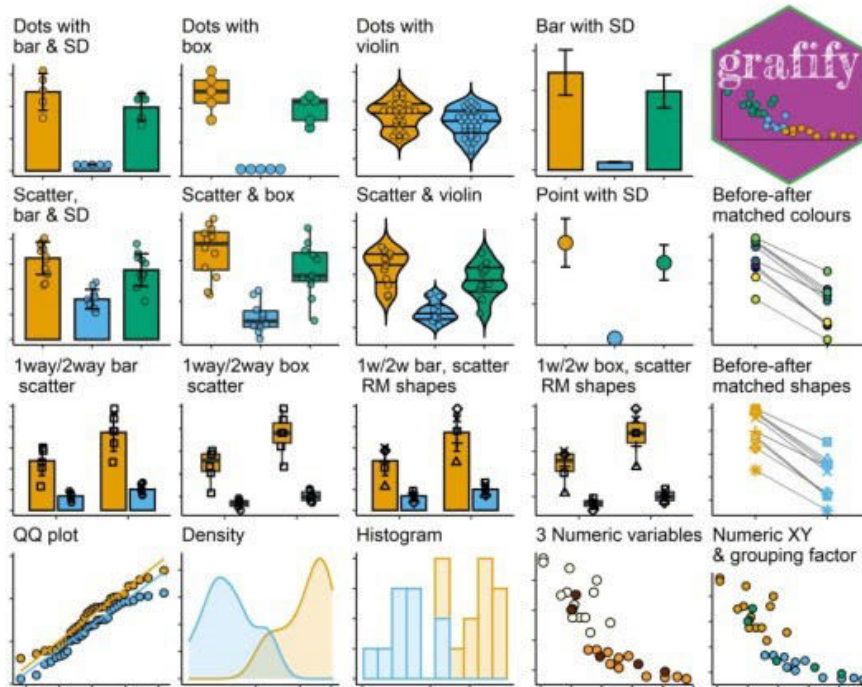Image Credit: grafify package

We'll go through a short tutorial to get you up and running with grafify.

Before we get started, get the R Cheat Sheet

grafify is great for making quick ggplot2 plots. But, you'll still need to learn how to wrangle data with dplyr and visualize data with ggplot2. For those topics, I'll use the Ultimate R Cheat Sheet to refer to dplyr and ggplot2 code in my workflow.

Quick Example:
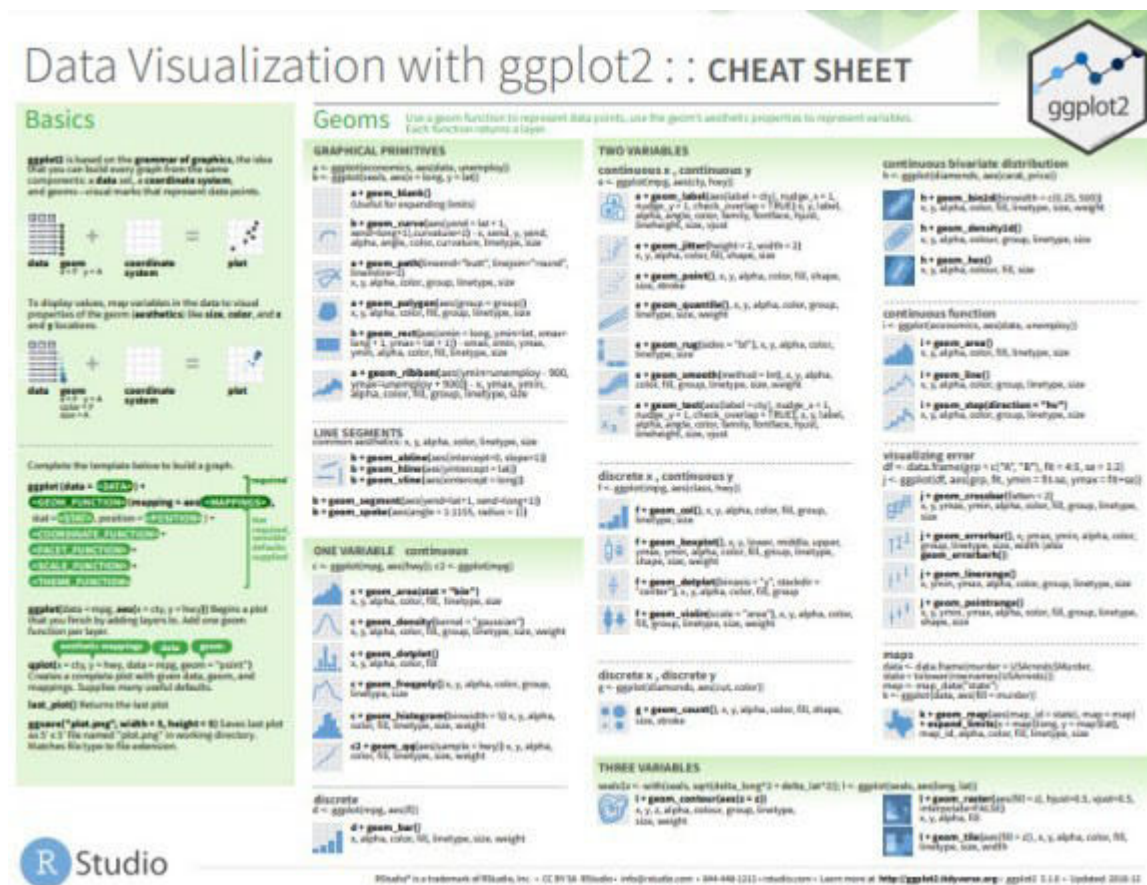
Download the Ultimate R Cheat Sheet. **Then Click the "CS" next to "ggplot2"** opens the Data Visualization with ggplot2 Cheat Sheet.
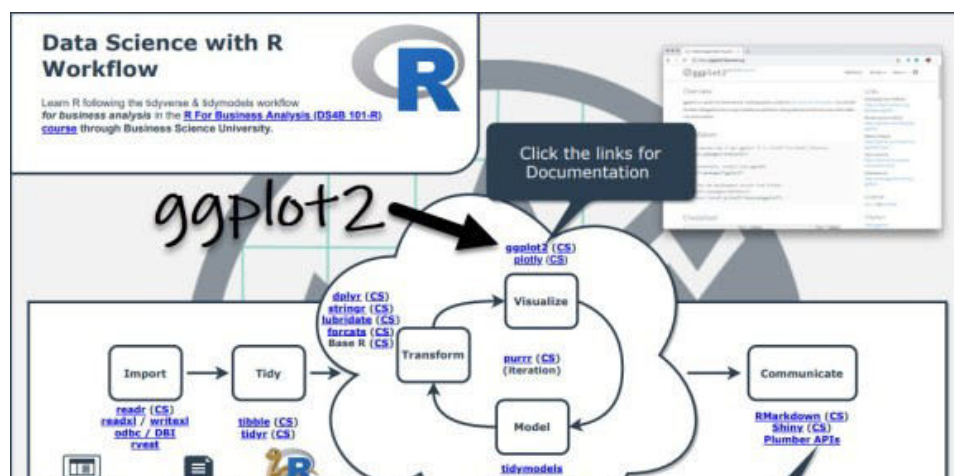
Now you're ready to quickly reference `ggplot2` functions.

Onto the tutorial.

How grafify works

The `grafify` package extends `ggplot2` by adding several simplified plotting functions. In this tutorial, we'll cover:

- **2-Variable Functions:** `plot_scatterbar_sd()`, `plot_scatterbox()`, and `plot_dotviolin()`

- **3-Variable Functions:** `plot_3d_scatterbox()`

- **Before-After Functions:** `plot_befafter_colors()`

Load the Libraries and Data

First, run this code to:

1. **Load Libraries:** Load `grafify` and `tidyverse`.
2. **Import Data:** We're using the `mpg` dataset that comes with `ggplot2`.

```
 7 ▾ # LIBRARIES ----
 8
 9   # remotes::install_github("ashenoy-cmbi/grafify@*release", dependencies = T)
10
11   library(tidyverse)
12   library(grafify)
13   |
14 ▾ # DATA ----
15   mpg
```

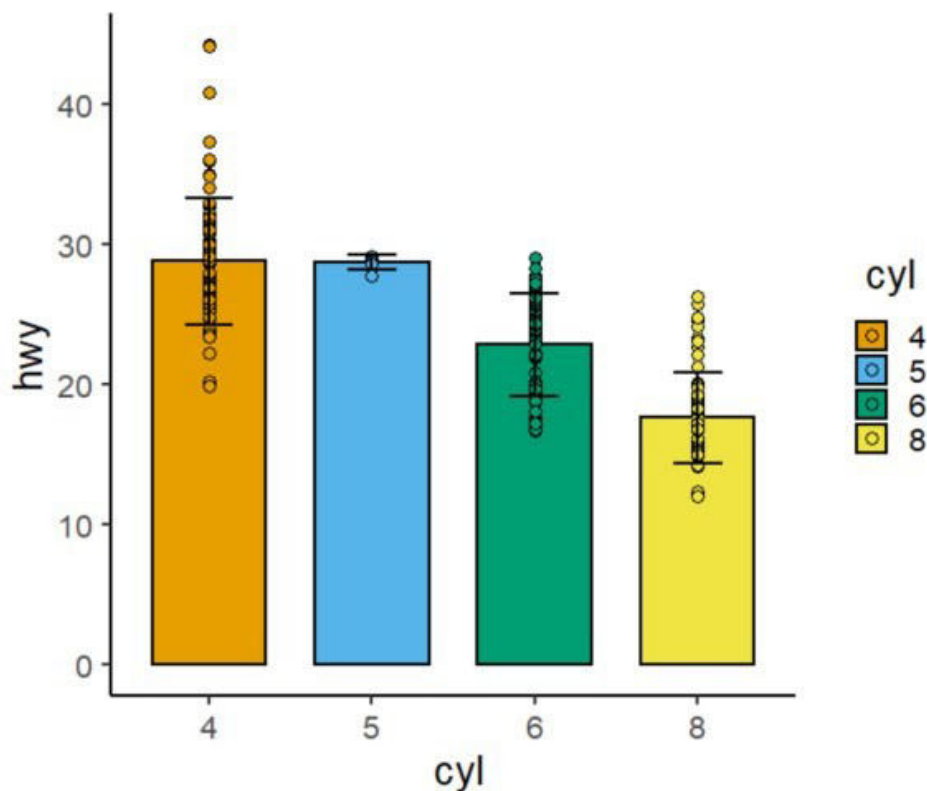Get the code.

Scatterbar SD Plot

First, we can make a Scatterbar Plot that shows the data points along with error bars at a standard deviation. Simply use `plot_scatterbar_sd()`.

```
19 ▾ # 1.1 Scatterbar SD ----
20   mpg %>%
21       plot_scatterbar_sd(cyl, hwy)
22
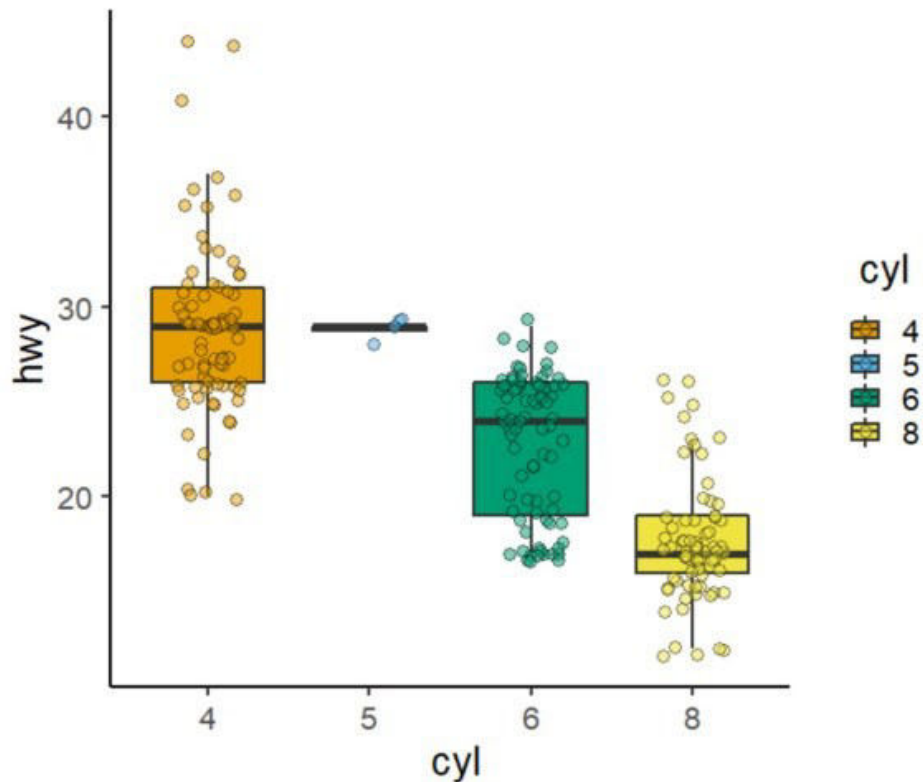```

Get the code.



# Scatterbox Plot

Next, we can make a Scatterbox Plot that shows a custom boxplot / jitter plot combination. I've added a jitter point to show the distribution. Simply use `plot_scatterbox()`.

```
22
23 ▾ # 1.2 Scatterbox ----
24 mpg %>%
25   plot_scatterbox(cyl, hwy, jitter = 0.2, s_alpha = 0.5)
26
```
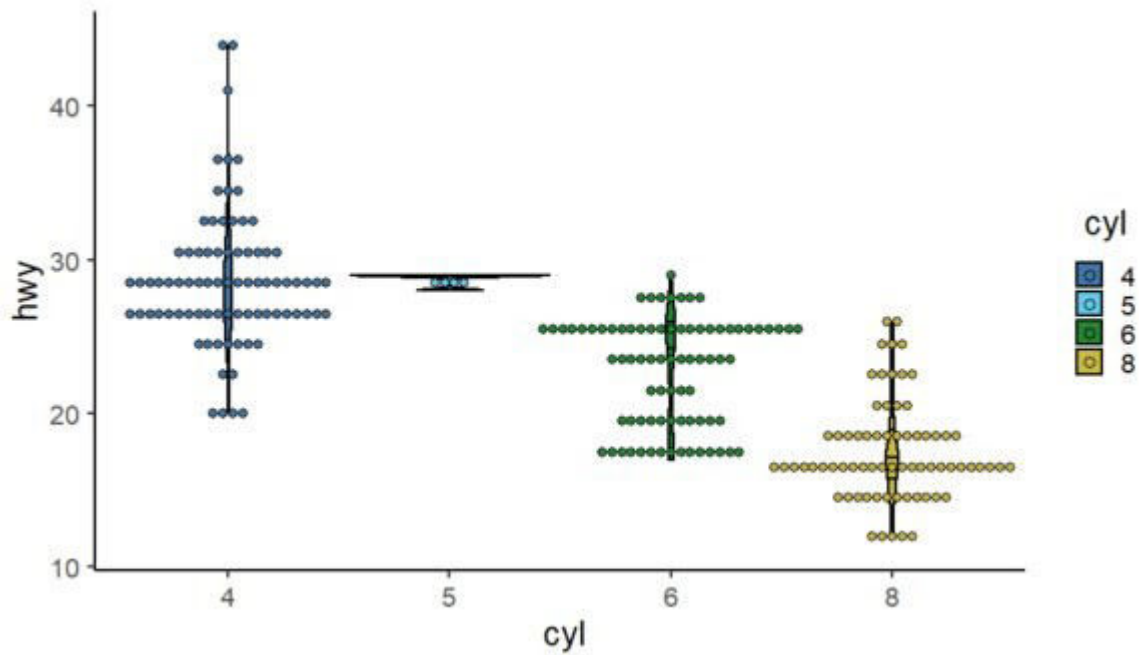
Get the code.



## Dotviolin Plot

Next, we can make a Dotviolin Plot that shows a custom violin plot / dotplot combination. Simply use `plot_dotviolin()`.

```
27 ▾ # 1.3 Dotviolin ----
28 mpg %>%
29   plot_dotviolin(cyl, hwy, dotsize = 0.6, ColPal = "bright")
30
```
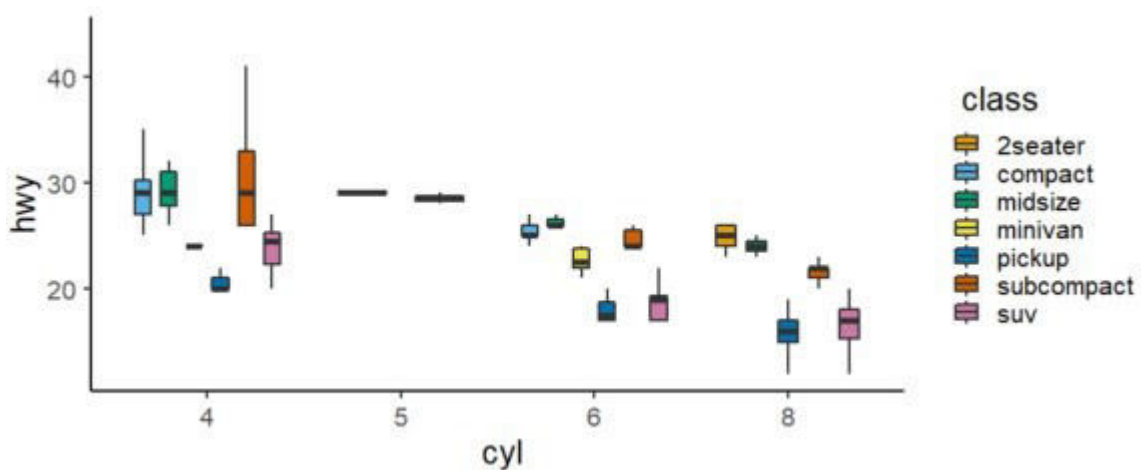
Get the code.

## Scatterbox 3D Plot

Next, we can make a 3D Scatterbox Plot that shows three variables using boxplot / jitter plot combination. This is great for drilling into multiple categories. Simply use `plot_3d_scatterbox()`.

```
31 ▾ # 2.0 GRAPHING 3-VARIABLES ----
32
33   mpg %>%
34   plot_3d_scatterbox(cyl, hwy, class, s_alpha = 0)
```

Get the code.



## Before-After Plot

Finally, we can make a Before-After Plot that shows changes between two states (in this case how various models changed in MPG Fuel Efficiency from 1999 to 2008). This is great for comparing two states. Simply use `plot_befafter_colors()`.

```
36  # 3.0 BEFORE-AFTER PLOTS ----
37
38  mpg %>%
39      group_by(model, year) %>%
40      summarize(mean_hwy = mean(hwy)) %>%
41      ungroup() %>%
42      plot_befafter_colors(year, mean_hwy, model)
43
44
```

Get the code.
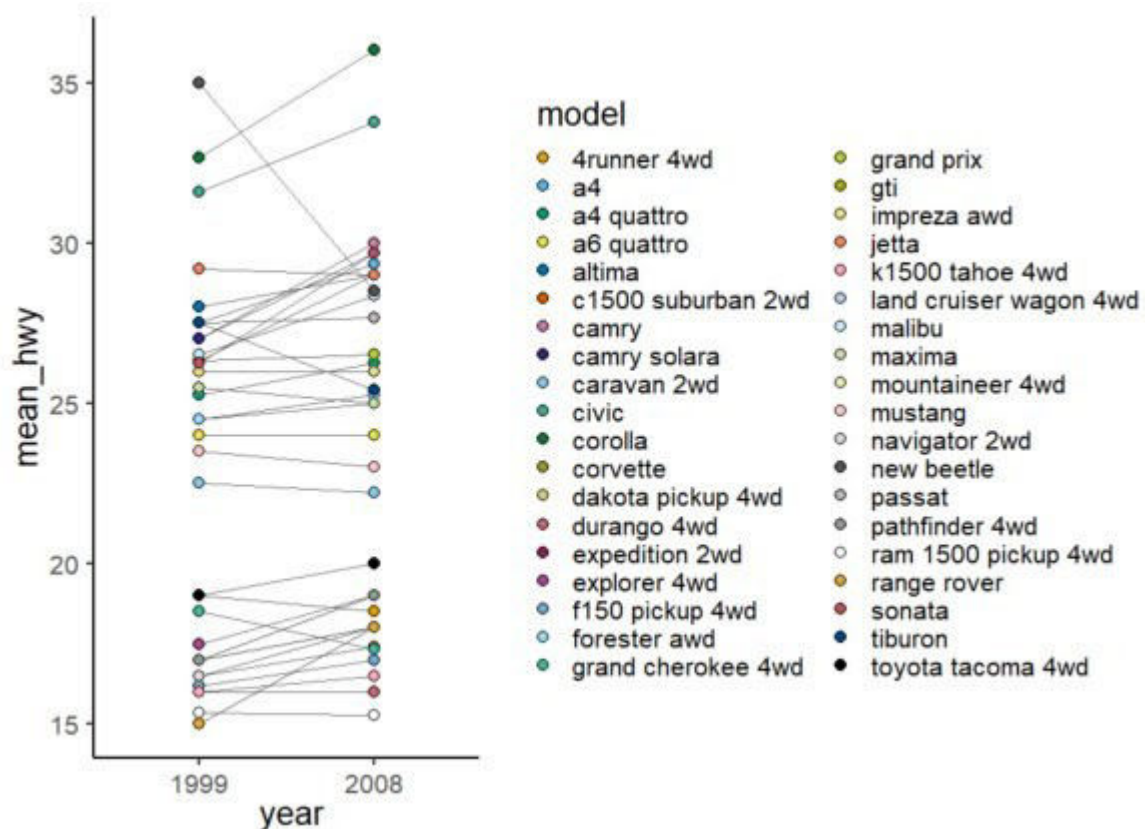
Summary

With 19 plotting functions, the `grafify` package makes it quick and easy to make custom `ggplot2` visualizations that are easy to visualize and explore data. With that said, it's critical to learn `ggplot2` for plots beyond what `grafify` offers.