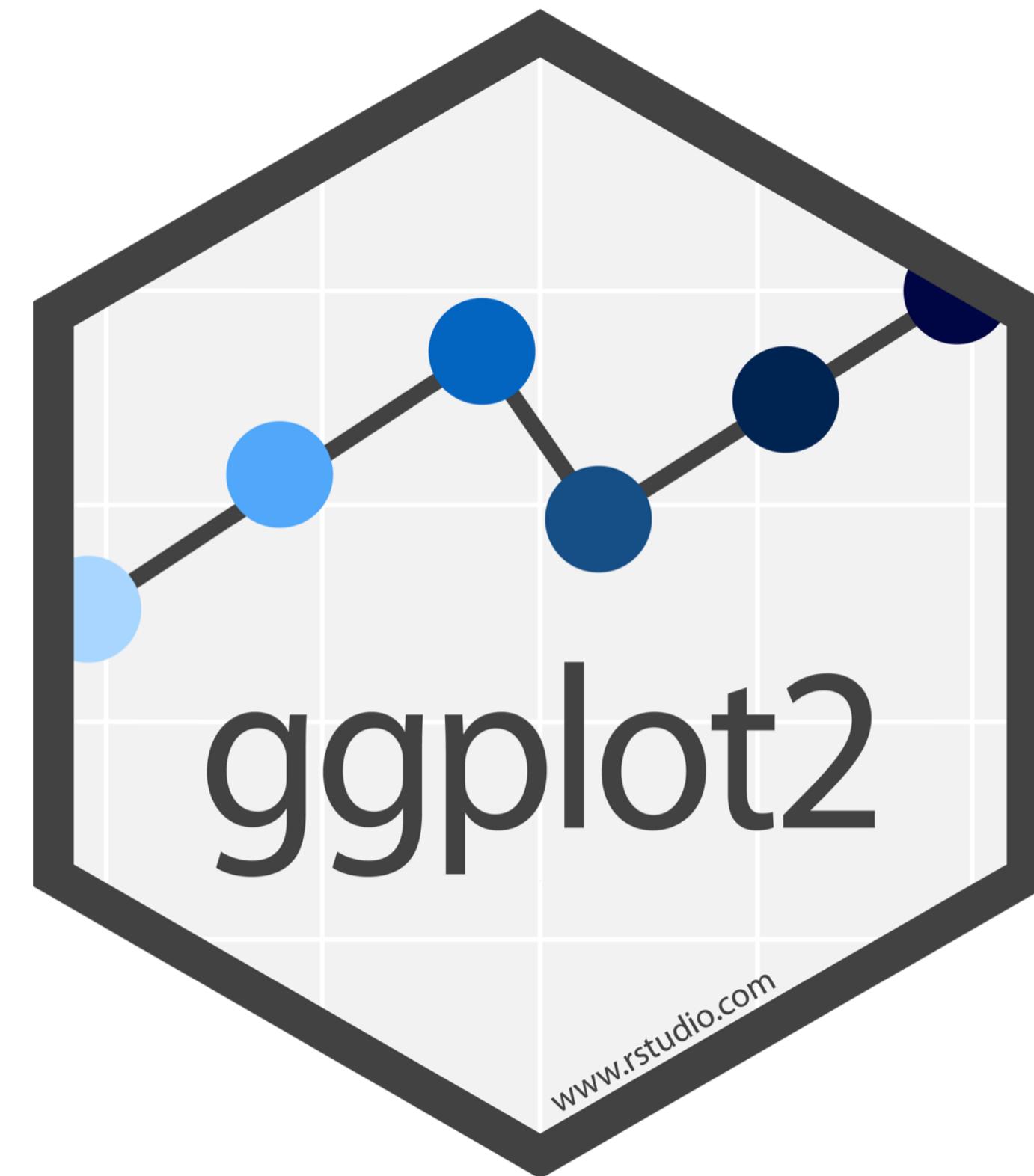


Visualize Data - Extras

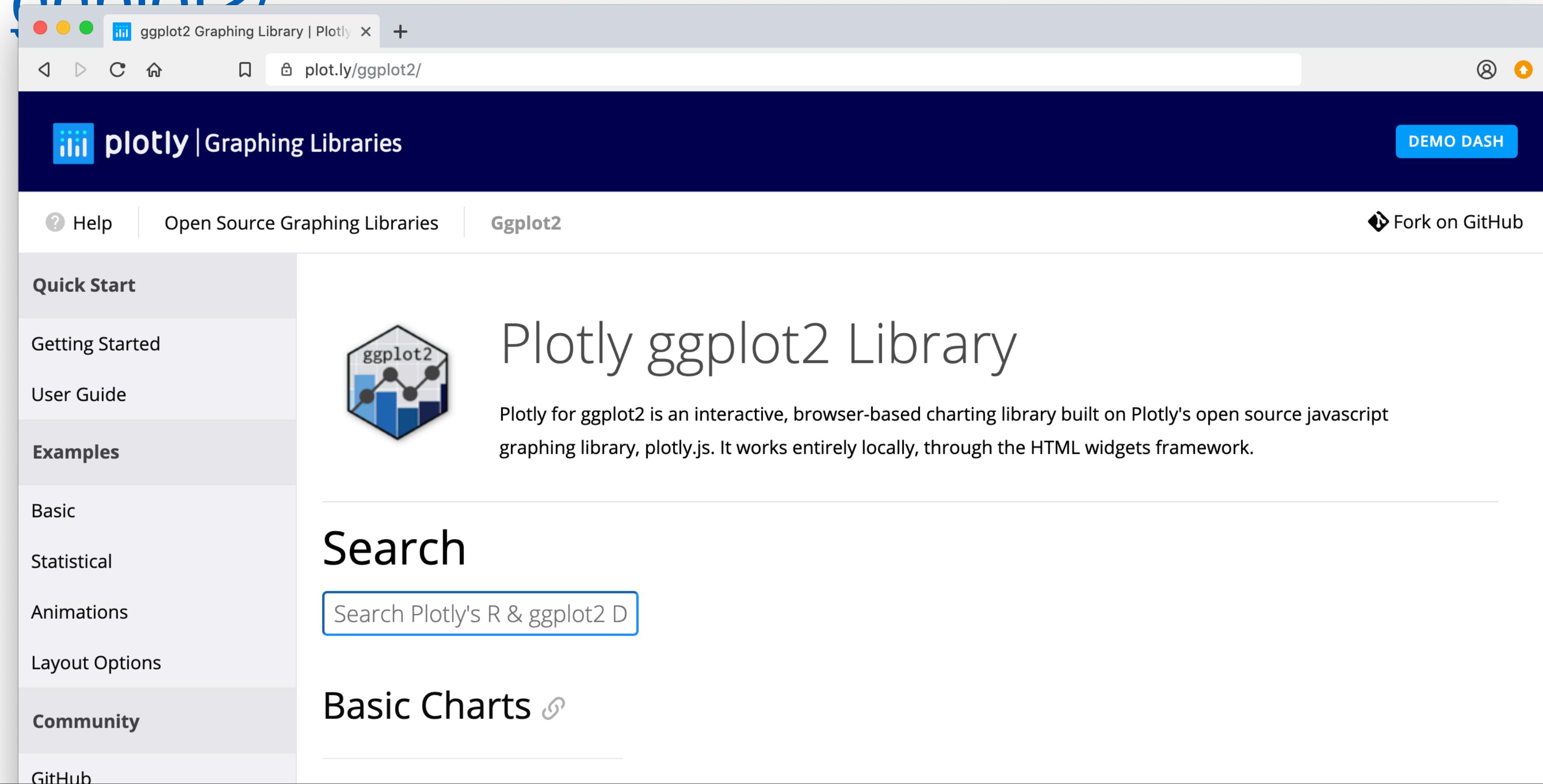


interactivity



Plotly

Tools for making interactive plots. plot.ly/ggplot2/



A screenshot of a web browser displaying the Plotly ggplot2 library page. The browser window has a title bar "ggplot2 Graphing Library | Plotly" and a URL bar showing "plot.ly/ggplot2/". The main content area features a dark header with the Plotly logo and "Graphing Libraries". Below the header is a navigation bar with links for "Help", "Open Source Graphing Libraries", "Ggplot2", and "Fork on GitHub". A sidebar on the left lists categories: "Quick Start" (Getting Started, User Guide), "Examples" (Basic, Statistical, Animations, Layout Options), "Community" (GitHub). The main content area is titled "Plotly ggplot2 Library" and contains a description: "Plotly for ggplot2 is an interactive, browser-based charting library built on Plotly's open source javascript graphing library, plotly.js. It works entirely locally, through the HTML widgets framework." It also includes a "Search" section with a search bar and a "Basic Charts" section.

ggplot2 Graphing Library | Plotly

plot.ly/ggplot2/

plotly | Graphing Libraries

DEMO DASH

Help Open Source Graphing Libraries Ggplot2 Fork on GitHub

Quick Start

Getting Started User Guide

Examples

Basic Statistical Animations Layout Options

Community GitHub

Plotly ggplot2 Library

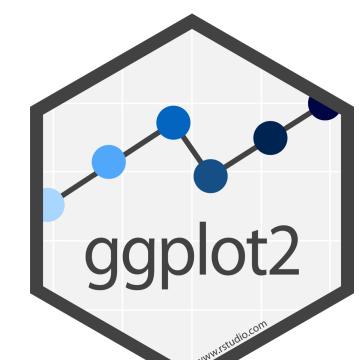
Plotly for ggplot2 is an interactive, browser-based charting library built on Plotly's open source javascript graphing library, plotly.js. It works entirely locally, through the HTML widgets framework.

Search

Search Plotly's R & ggplot2 D

Basic Charts

www.plotly.com



Your Turn

Create an interactive graph. Can you add an extra information to on-hover info? Can you restrict it to just 'class' info?

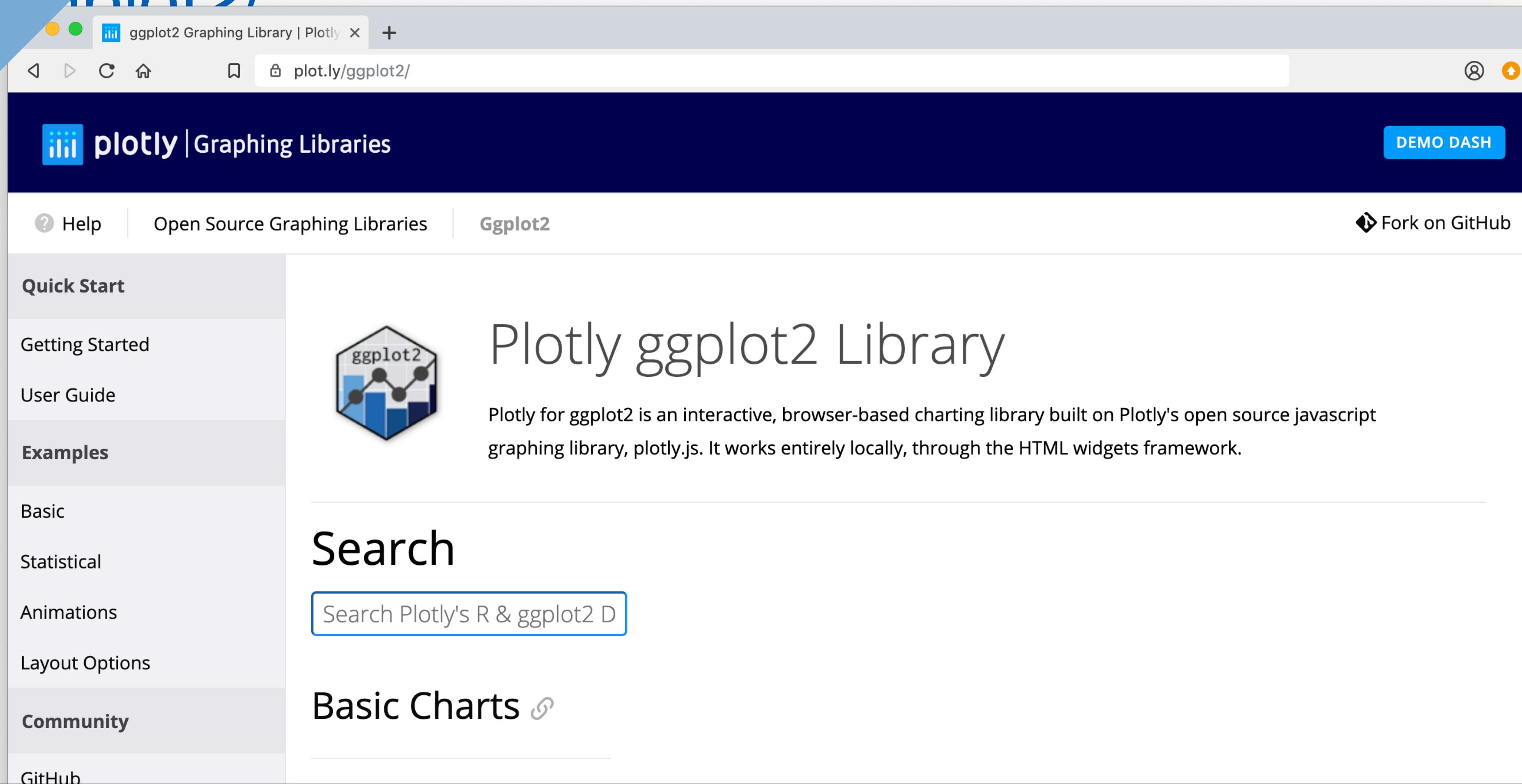
```
p <- ggplot(mpg) +  
  geom_point(aes(displ, hwy)) +  
  geom_smooth(aes(displ, hwy))
```

```
library(plotly)  
ggplotly(p)
```

htmlwidgets::
Save with
htmlwidgets::saveWidget()

Plotly

Tools for making interactive plots. plot.ly/



The screenshot shows a web browser window displaying the "ggplot2 Graphing Library | Plotly" page at plot.ly/ggplot2/. The page has a dark blue header with the Plotly logo and a "DEMO DASH" button. Below the header, there are navigation links for "Help", "Open Source Graphing Libraries", "Ggplot2", and "Fork on GitHub". A sidebar on the left lists categories: "Quick Start" (Getting Started, User Guide), "Examples" (Basic, Statistical, Animations, Layout Options), "Community" (GitHub), and "Search". The main content area features a hexagonal icon with a line plot and text about the "Plotly ggplot2 Library". It explains that Plotly for ggplot2 is an interactive, browser-based charting library built on Plotly's open source javascript graphing library, plotly.js. It works entirely locally, through the HTML widgets framework. There is also a search bar and a "Basic Charts" section.

ggplot2 Graphing Library | Plotly

plot.ly/ggplot2/

plotly | Graphing Libraries

DEMO DASH

Help Open Source Graphing Libraries Ggplot2 Fork on GitHub

Quick Start

Getting Started User Guide

Examples

Basic Statistical Animations Layout Options

Community GitHub

Search

Search Plotly's R & ggplot2 D

Plotly ggplot2 Library

Plotly for ggplot2 is an interactive, browser-based charting library built on Plotly's open source javascript graphing library, plotly.js. It works entirely locally, through the HTML widgets framework.

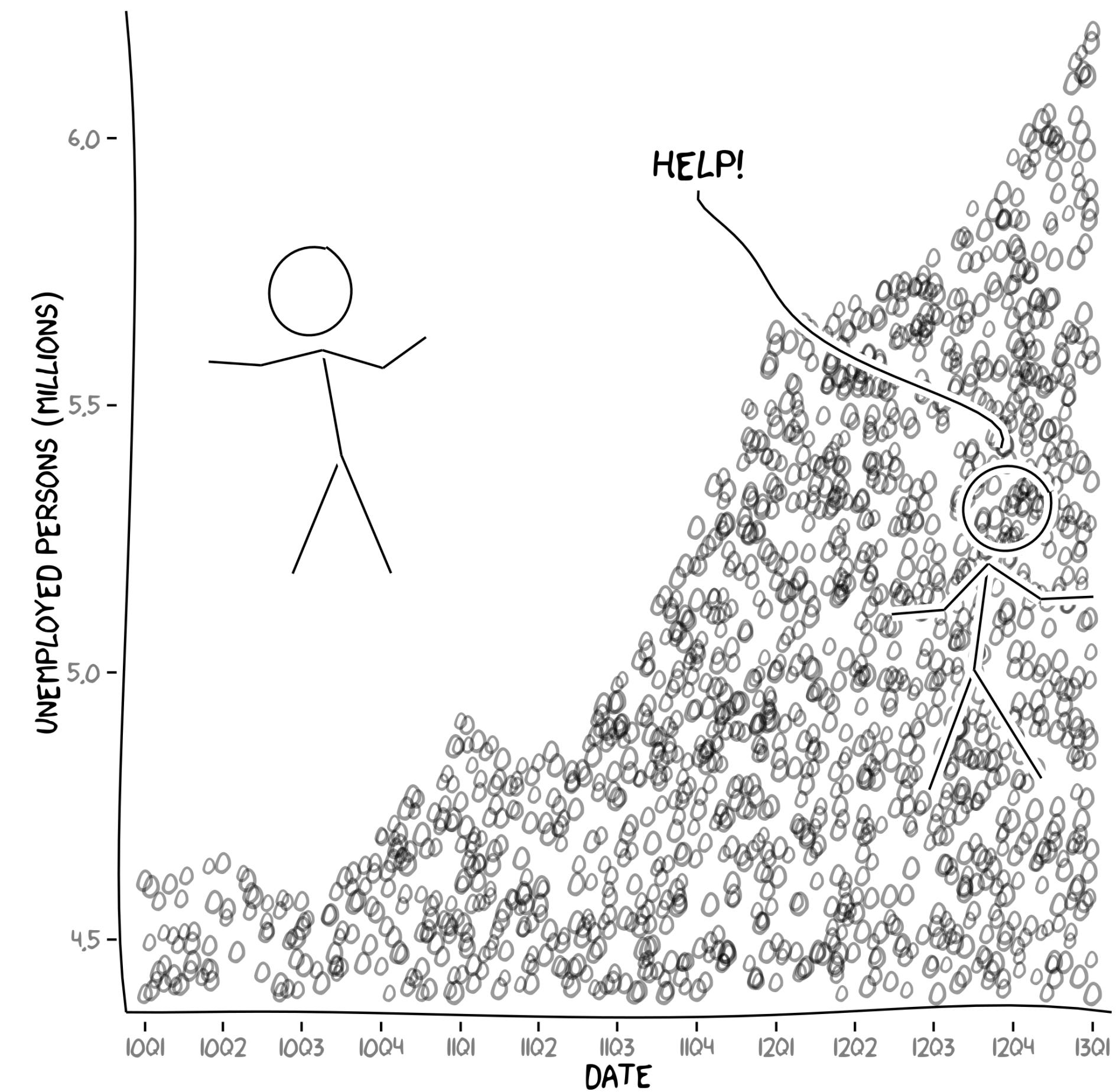
Basic Charts

ggplot2

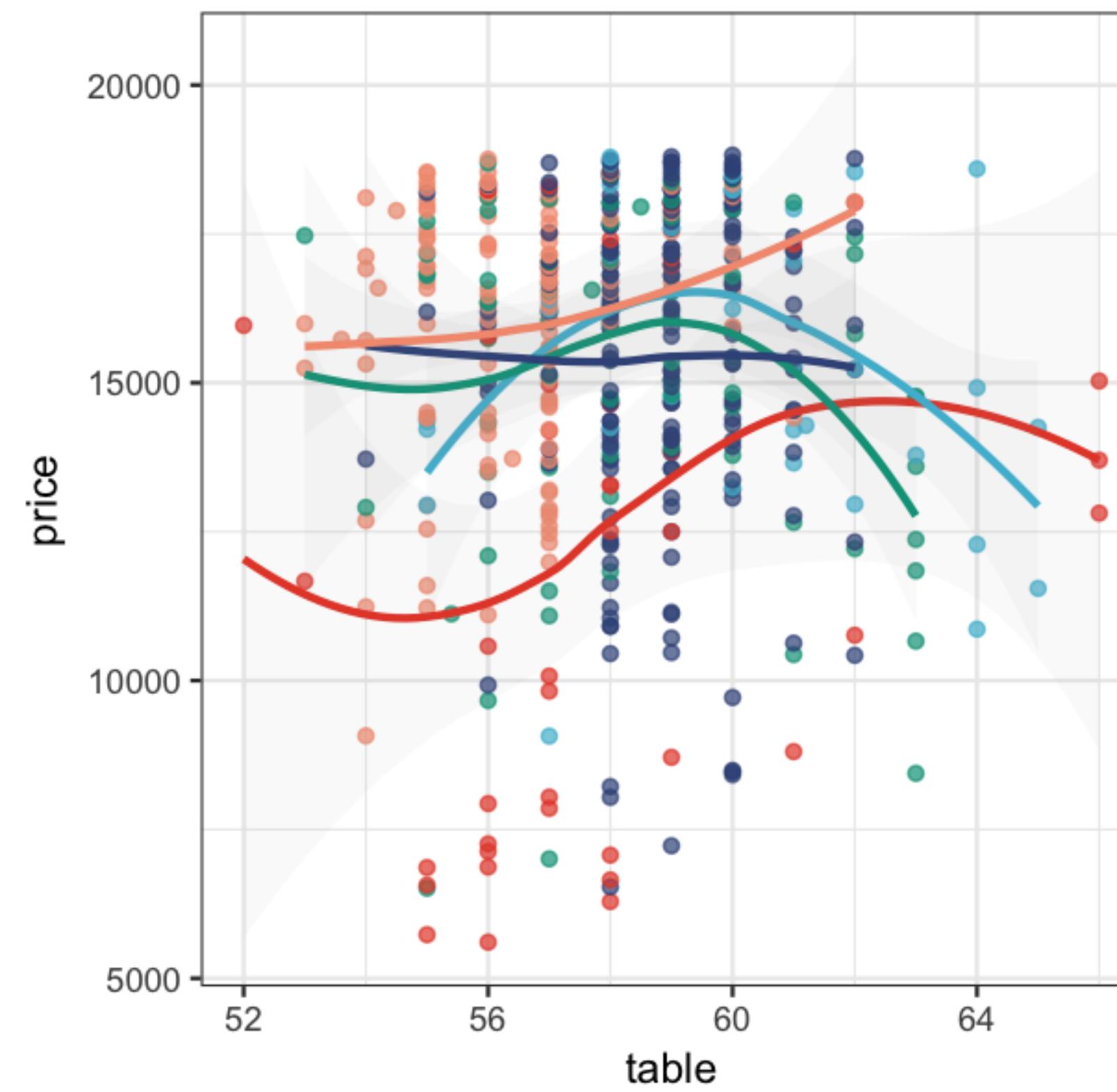
pallets & themes

TR
R

xkcd theme

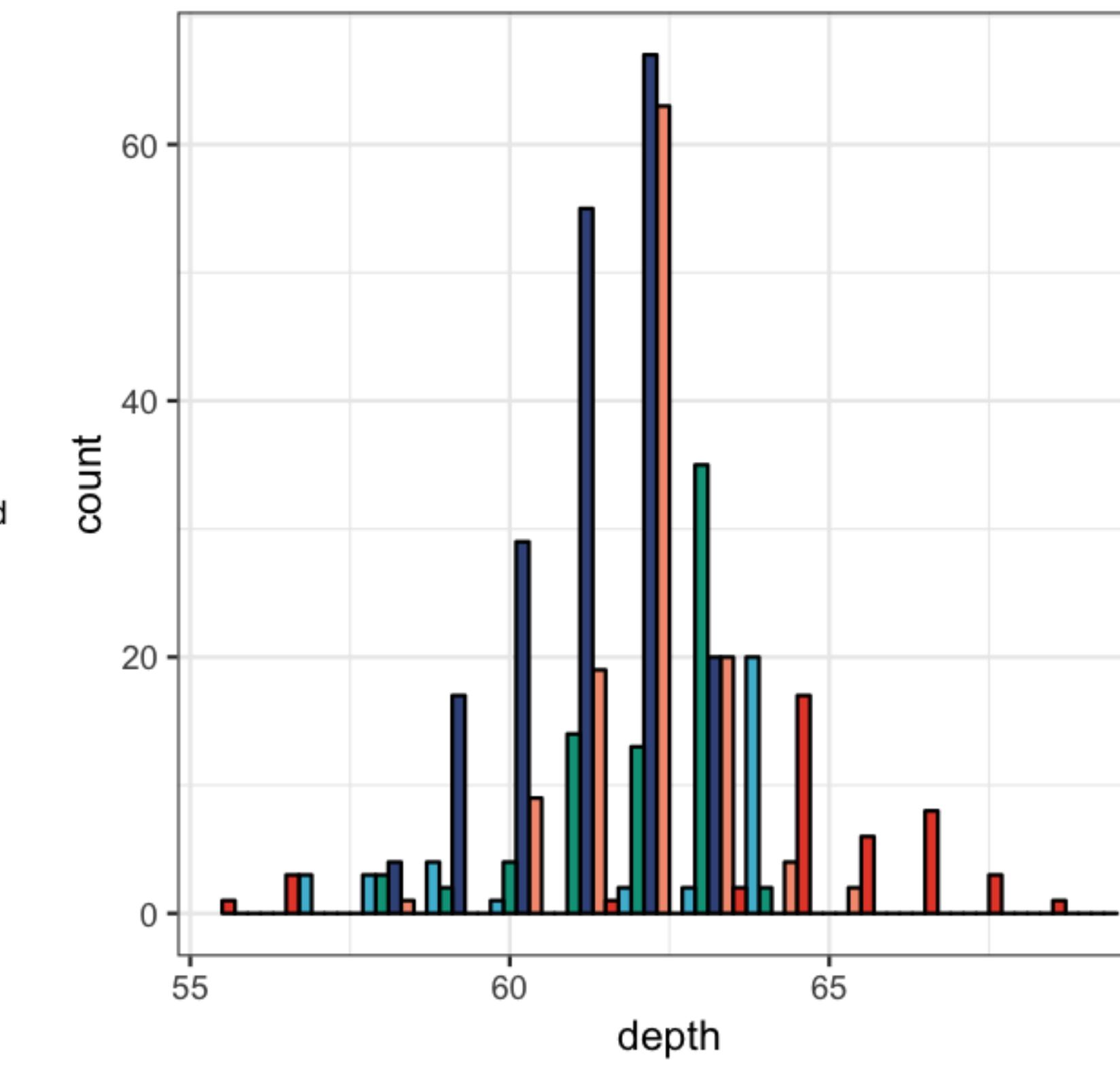


ggsci: scientific journals pallettes



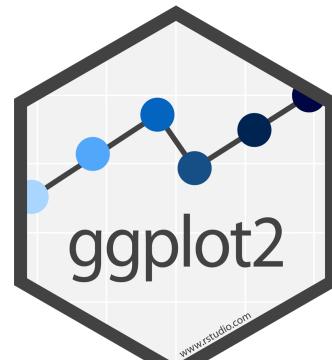
cut

- Fair
- Good
- Very Good
- Premium
- Ideal

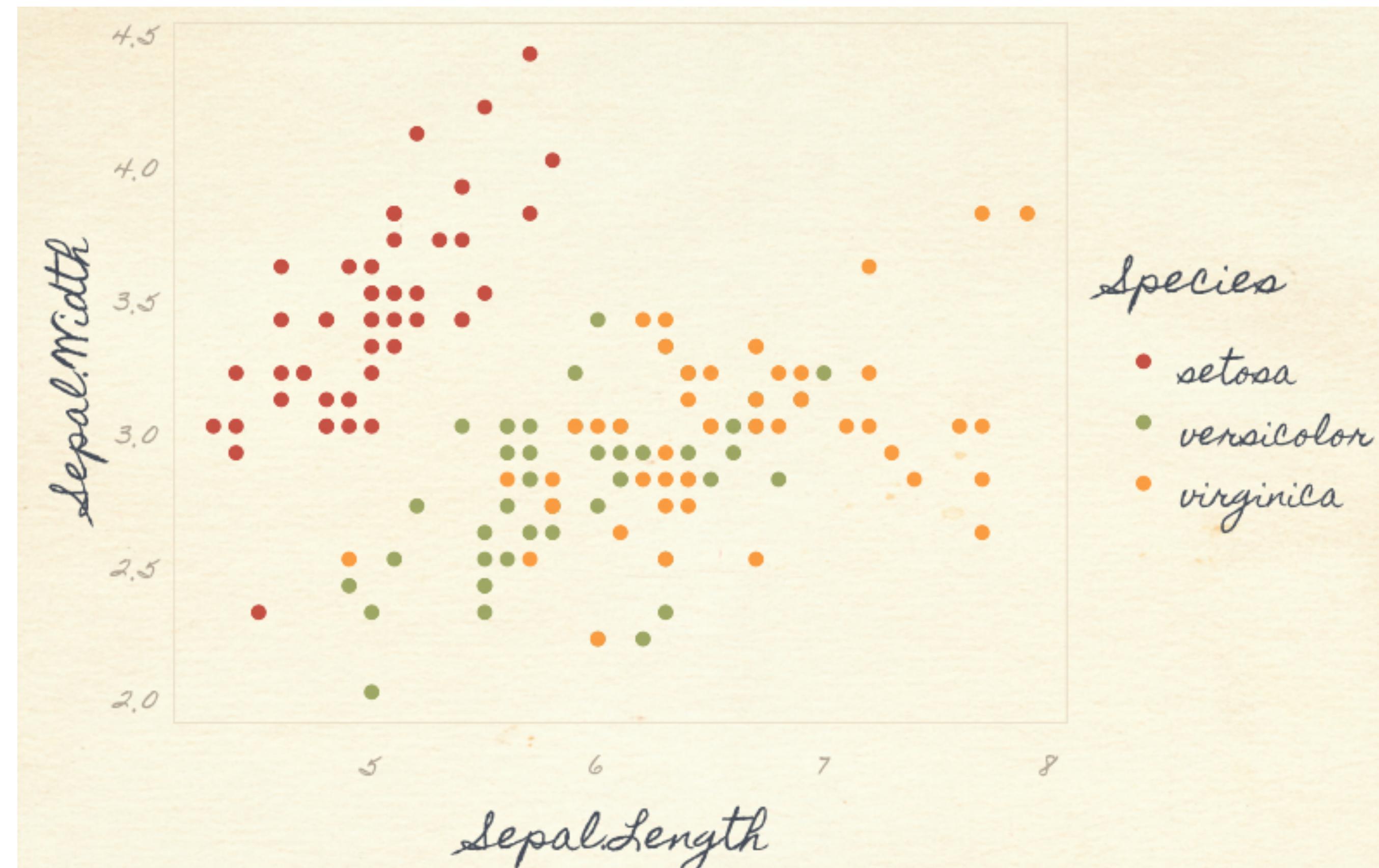


cut

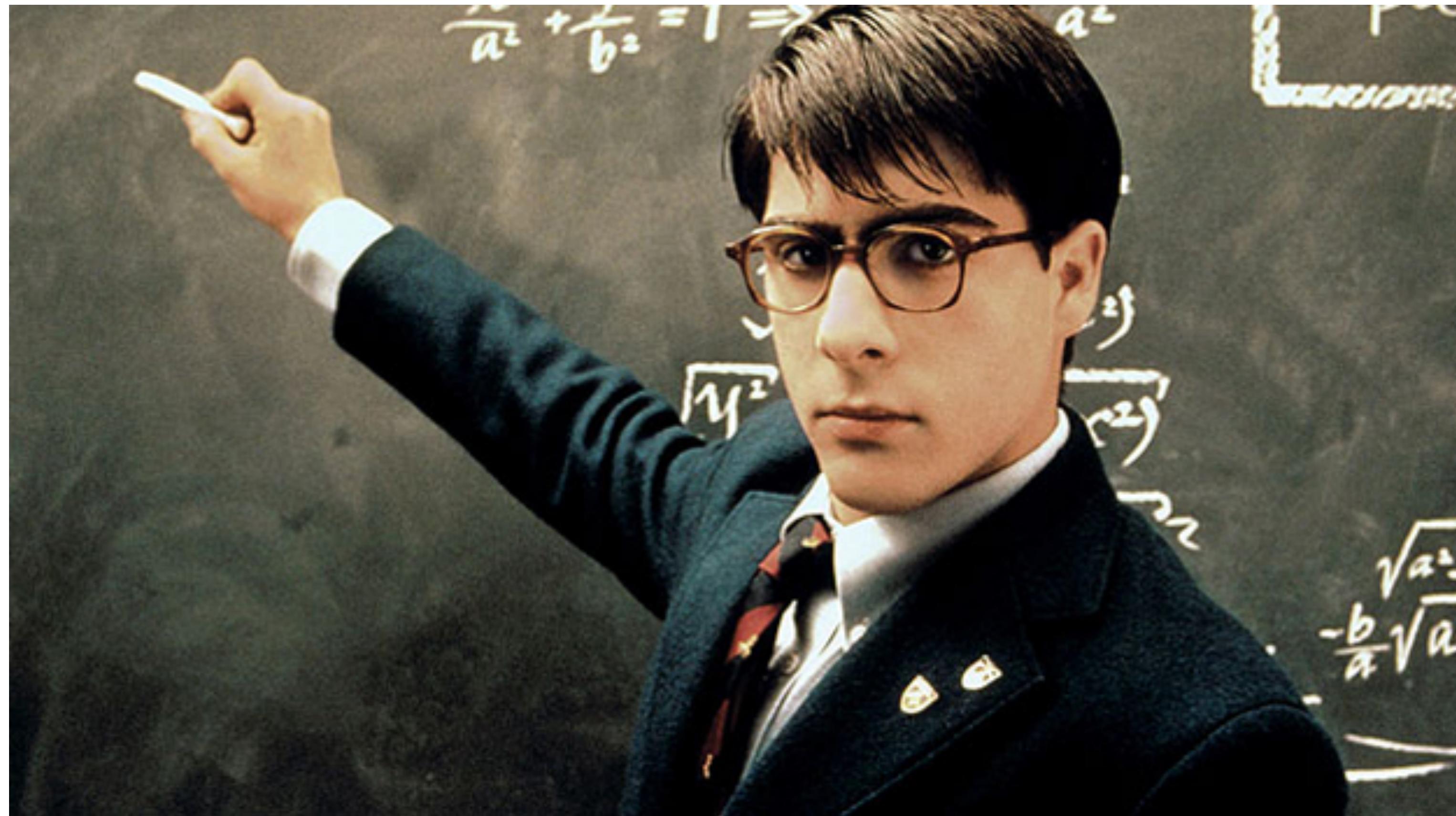
- Fair
- Good
- Very Good
- Premium
- Ideal



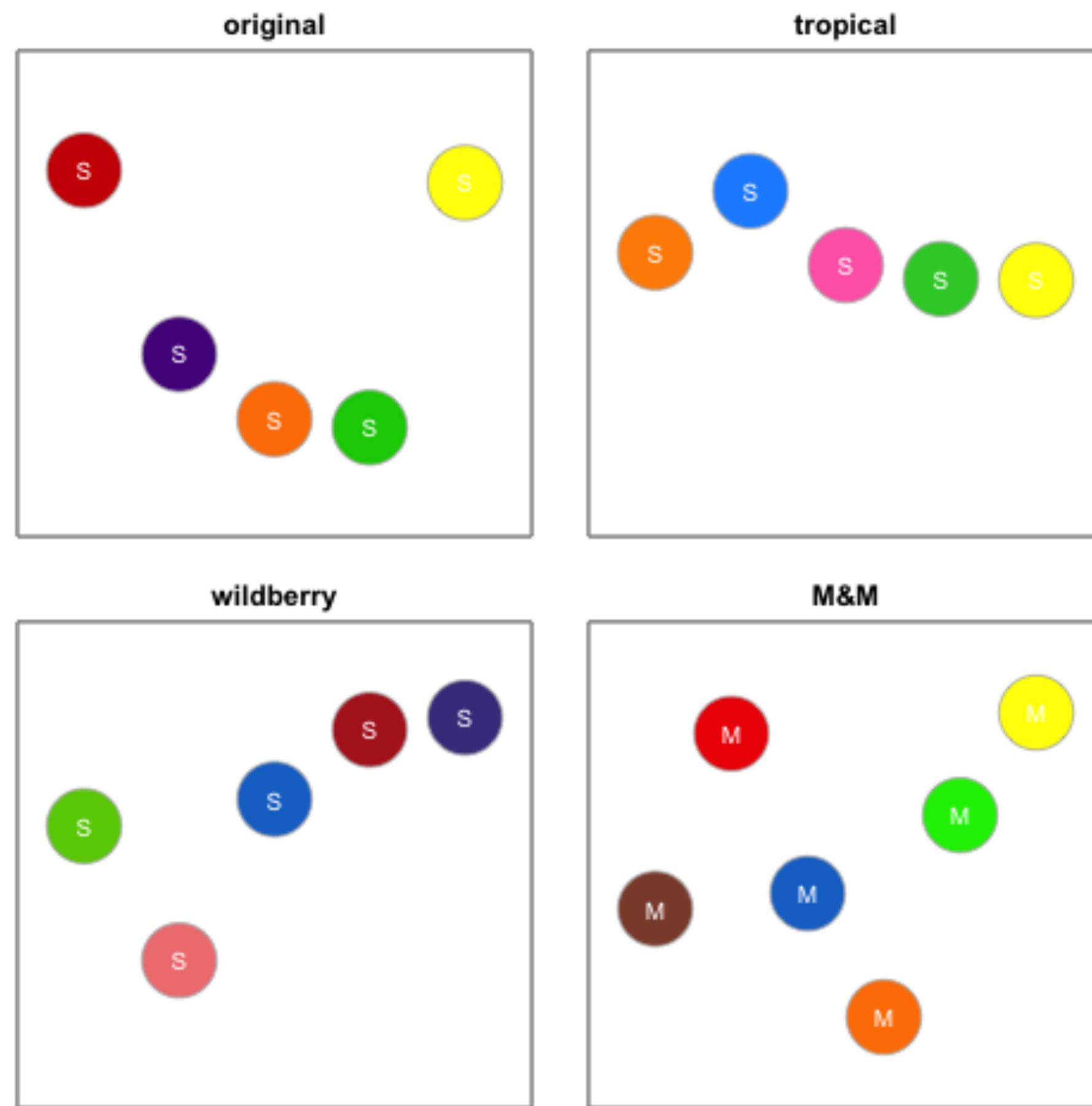
USDA Pomological Watercolors collection



Pallets of Wess Anderson Movies



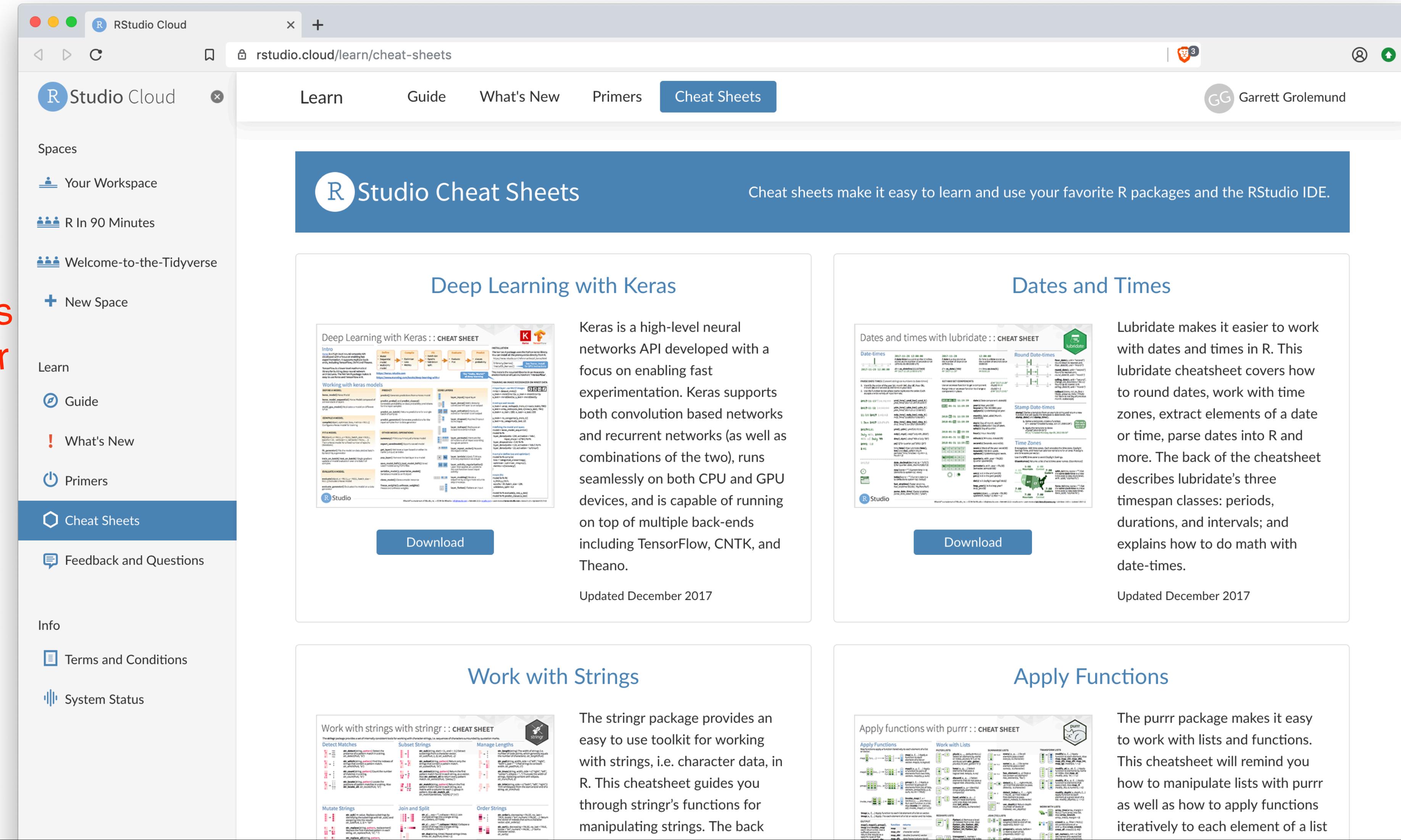
skittles & gummy bears



Additional Information

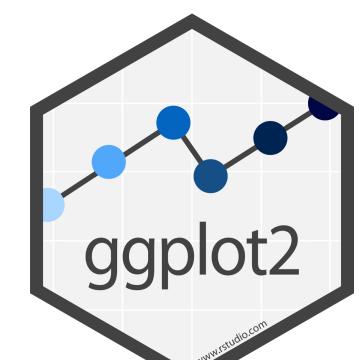
rstudio.cloud/learn/cheat-sheets

Click
CheAT Sheets
in the sidebar



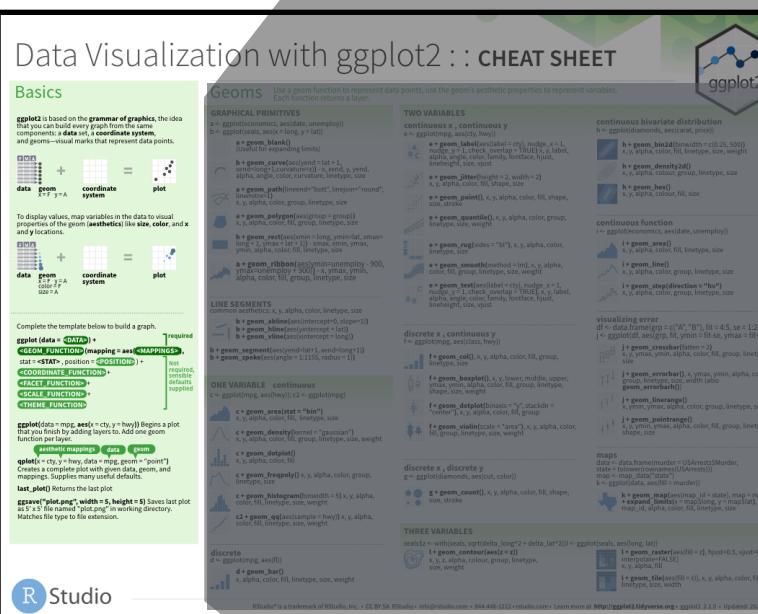
The screenshot shows the RStudio Cloud interface with the 'Cheat Sheets' section highlighted in the sidebar. The main content area displays four cheat sheets: 'Deep Learning with Keras', 'Dates and Times', 'Work with Strings', and 'Apply Functions'. Each section includes a preview of the cheat sheet, a 'Download' button, and a note indicating it was updated in December 2017.

- Deep Learning with Keras**
Keras is a high-level neural networks API developed with a focus on enabling fast experimentation. Keras supports both convolution based networks and recurrent networks (as well as combinations of the two), runs seamlessly on both CPU and GPU devices, and is capable of running on top of multiple back-ends including TensorFlow, CNTK, and Theano.
Updated December 2017
- Dates and Times**
Lubridate makes it easier to work with dates and times in R. This lubridate cheatsheet covers how to round dates, work with time zones, extract elements of a date or time, parse dates into R and more. The back of the cheatsheet describes lubridate's three timespan classes: periods, durations, and intervals; and explains how to do math with date-times.
Updated December 2017
- Work with Strings**
The stringr package provides an easy to use toolkit for working with strings, i.e. character data, in R. This cheatsheet guides you through stringr's functions for manipulating strings. The back
- Apply Functions**
The purrr package makes it easy to work with lists and functions. This cheatsheet will remind you how to manipulate lists with purrr as well as how to apply functions iteratively to each element of a list



geom_ functions

Each requires a mapping argument.



Geoms Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

ggplot2

GRAPHICAL PRIMITIVES

```
a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))
```

- a + geom_blank()** (Useful for expanding limits)
- b + geom_curve(aes(yend = lat + 1, xend = long + 1, curvature = z))** - x, yend, y, yend, alpha, angle, color, curvature, linetype, size
- a + geom_path(lineend = "butt", linejoin = "round", linemetre = 1)** x, y, alpha, color, group, linetype, size
- a + geom_polygon(aes(group = group))** x, y, alpha, color, fill, group, linetype, size
- b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1))** - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size
- a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900))** - x, ymax, ymin, alpha, color, fill, group, linetype, size

TWO VARIABLES

continuous x , continuous y

```
e <- ggplot(mpg, aes(cty, hwy))
```

- e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)** x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust
- A B C**
- e + geom_jitter(height = 2, width = 2)** x, y, alpha, color, fill, shape, size
- e + geom_point()** x, y, alpha, color, fill, shape, size, stroke
- e + geom_quantile()** x, y, alpha, color, group, linetype, size, weight
- B**
- e + geom_rug(sides = "bl")** x, y, alpha, color, linetype, size
- e + geom_smooth(method = lm)** x, y, alpha, color, fill, group, linetype, size, weight
- C A B**
- e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)** x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

continuous bivariate distribution

```
h <- ggplot(diamonds, aes(carat, price))
```

- h + geom_bin2d(binwidth = c(0.25, 500))** x, y, alpha, color, fill, linetype, size, weight
- h + geom_density2d()** x, y, alpha, colour, group, linetype, size
- h + geom_hex()** x, y, alpha, colour, fill, size

continuous function

```
i <- ggplot(economics, aes(date, unemploy))
```

- i + geom_area()** x, y, alpha, color, fill, linetype, size
- i + geom_line()** x, y, alpha, color, group, linetype, size
- i + geom_step(direction = "hv")** x, y, alpha, color, group, linetype, size

visualizing error

```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))
```

- j + geom_crossbar(fatten = 2)** x, y, ymax, ymin, alpha, color, fill, group, linetype, size
- J I I**
- j + geom_errorbar()**, x, y, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh()**)
- J I I**
- j + geom_linerange()** x, ymin, ymax, alpha, color, group, linetype, size
- J I I**
- j + geom_pointrange()** x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

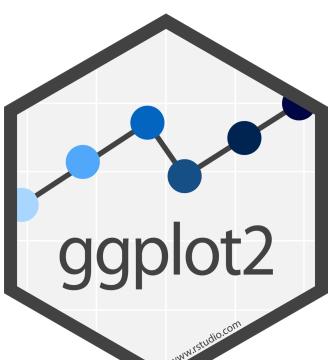
```
data <- data.frame(murder = USArrests$Murder,
state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))
```

- k + geom_map(aes(map_id = state), map = map) + expand_limits(x = map\$long, y = map\$lat), map_id, alpha, color, fill, linetype, size**
- K**

THREE VARIABLES

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))
l <- ggplot(seals, aes(long, lat))
```

- l + geom_contour(aes(z = z))** x, y, z, alpha, colour, group, linetype, size, weight
- L**
- l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)** x, y, alpha, fill
- L**
- l + geom_tile(aes(fill = z))** x, y, alpha, color, fill, linetype, size, width
- L**



ggplot2.tidyverse.org

The screenshot shows a web browser displaying the ggplot2.tidyverse.org website. The page title is "Create Elegant Data Visualisation" and the user is logged in as "Garrett". The URL in the address bar is "ggplot2.tidyverse.org". The main content area features the ggplot2 logo and the text "part of the tidyverse". A large section titled "Usage" contains a paragraph explaining the philosophy of ggplot2 and a code snippet demonstrating its usage with the mpg dataset. Below the code is a scatter plot showing the relationship between engine displacement (displ) and fuel economy (hwy), with points colored by vehicle class. The right sidebar contains links for "Links", "License", and "Developers". A small ggplot2 logo is in the bottom right corner.

Usage

It's hard to succinctly describe how ggplot2 works because it embodies a deep philosophy of visualisation. However, in most cases you start with `ggplot()`, supply a dataset and aesthetic mapping (with `aes()`). You then add on layers (like `geom_point()` or `geom_histogram()`), scales (like `scale_colour_brewer()`), faceting specifications (like `facet_wrap()`) and coordinate systems (like `coord_flip()`).

```
library(ggplot2)

ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()
```

A scatter plot with "displ" on the x-axis and "hwy" on the y-axis. The x-axis ranges from approximately 100 to 400, and the y-axis ranges from 10 to 50. Points are colored by "class": "2seater" (red), "4cyl" (blue), "6cyl" (green), "8cyl" (yellow), and "10cyl" (purple). A legend on the right identifies the colors for each class. A single black point is located at the bottom center of the plot area.

Links

- Download from CRAN at <https://cran.r-project.org/package=ggplot2>
- Browse source code at <https://github.com/tidyverse/ggplot2>
- Report a bug at <https://github.com/tidyverse/ggplot2/issues>
- Learn more at <http://r4ds.had.co.nz/data-visualisation.html>

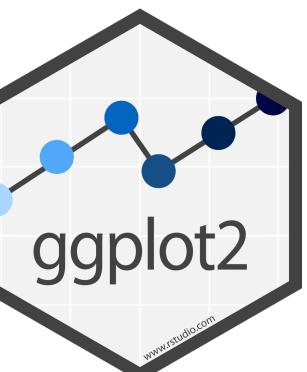
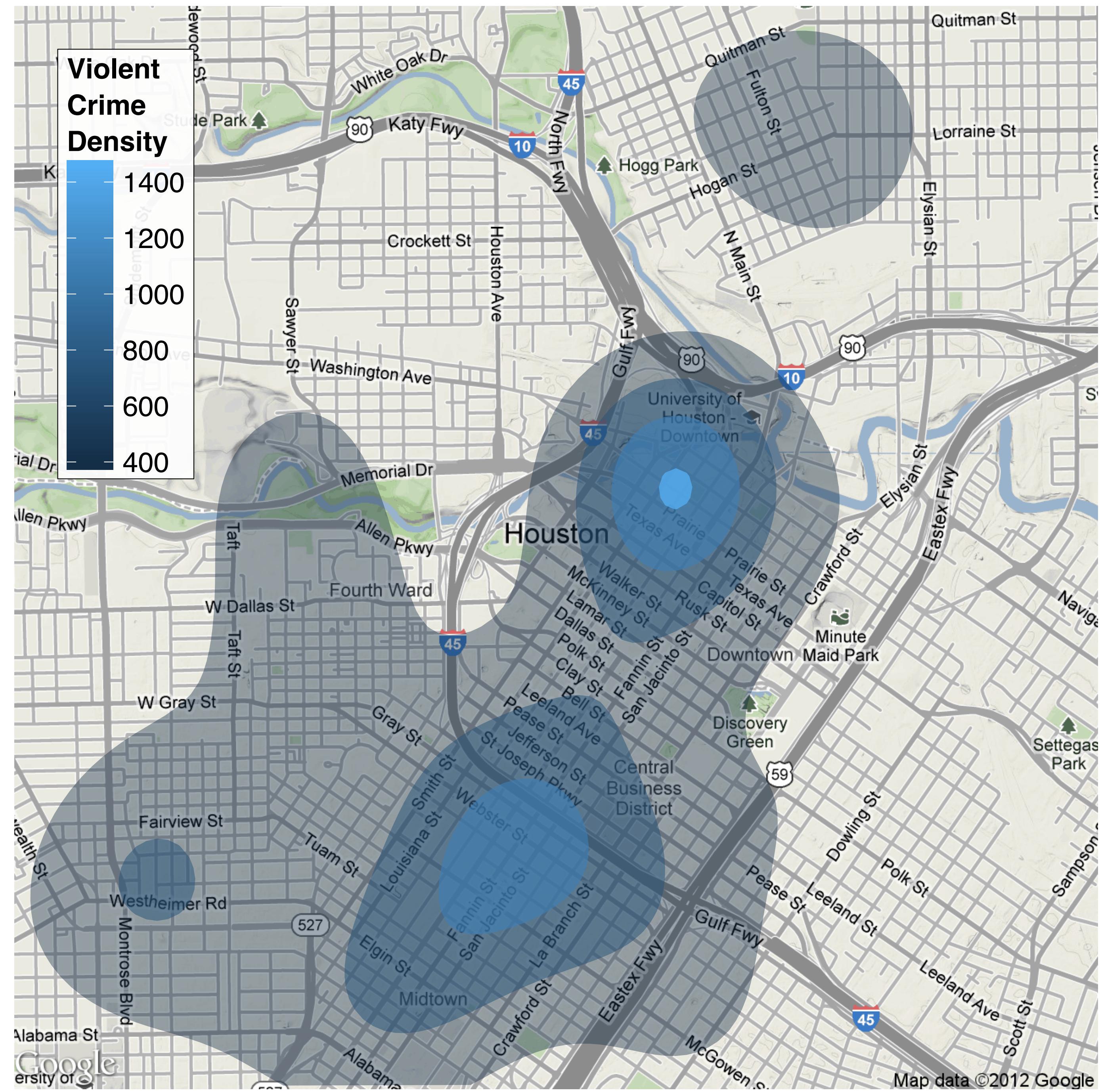
License

GPL-2 | file [LICENSE](#)

Developers

Hadley Wickham
Author maintainer

A small hexagonal logo for ggplot2, featuring a blue line graph icon and the text "ggplot2" below it.



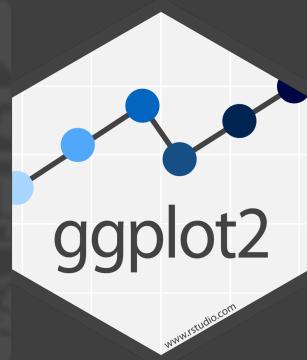
London Cycle Hire Journeys

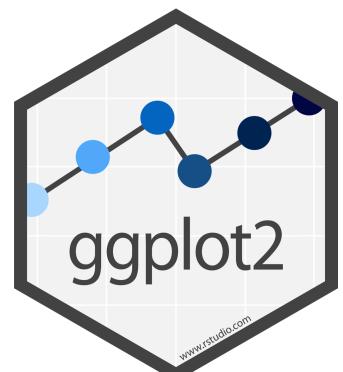
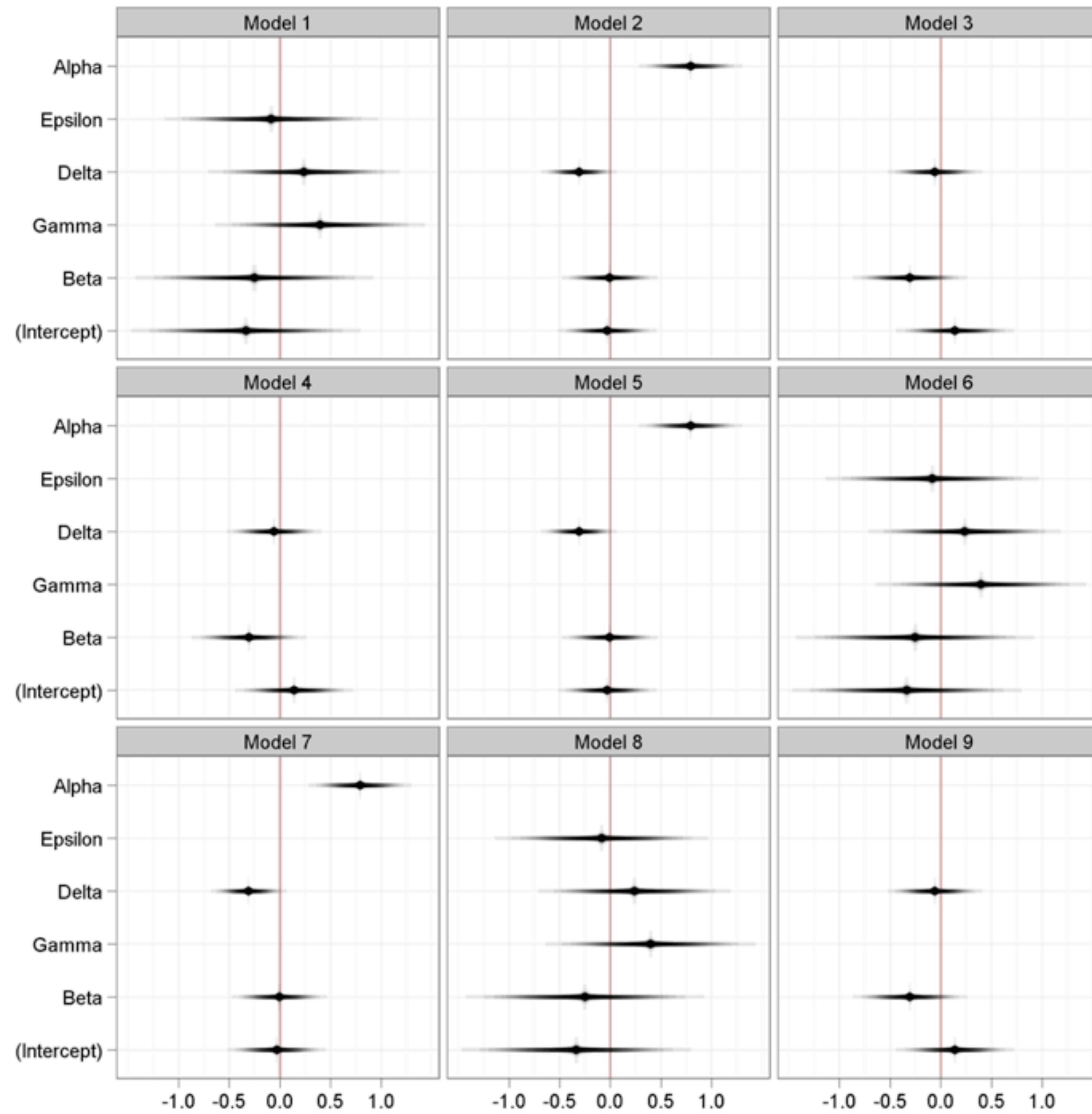
Thicker, yellower lines mean more journeys



Data: 3.2 Million Journeys (from TfL)
Routing: Ollie O'Brien (@oobr) + OpenStreetMap cc-by-sa
Buildings: OS OpenData Crown Copyright 2011
Map: James Cheshire (@spatialanalysis)

James Cheshire, <http://bit.ly/xqHhAs>





from Data to Viz



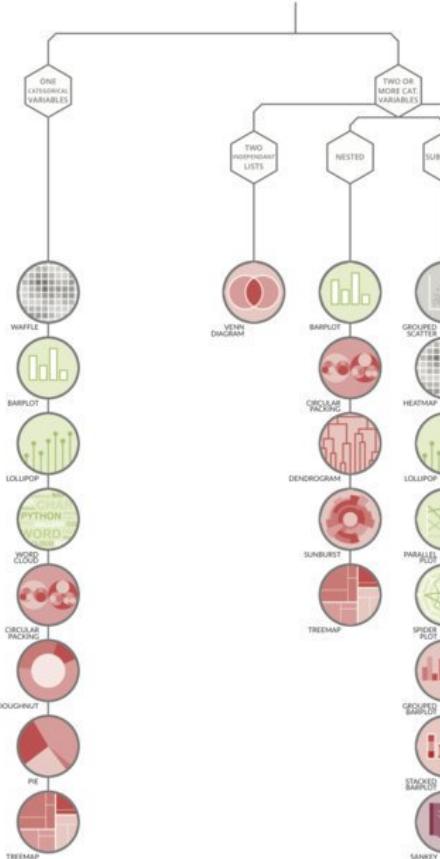
'From Data to Viz' is a classification of chart types based on input data format. It will help you find the perfect chart in three simple steps:

- ① Identify what type of data you have.
- ② Go to the corresponding decision tree and follow it down to a set of possible charts.
- ③ Choose the chart from the set that will suit your data and your needs best.

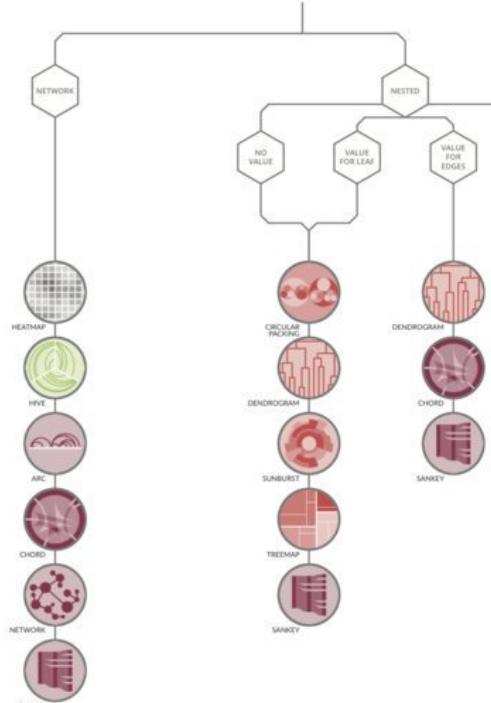
Dataviz is a world with endless possibilities and this project does not claim to be exhaustive. However it should provide you with a good starting point. For an interactive version and much more, visit:

data-to-viz.com

CATEGORIC



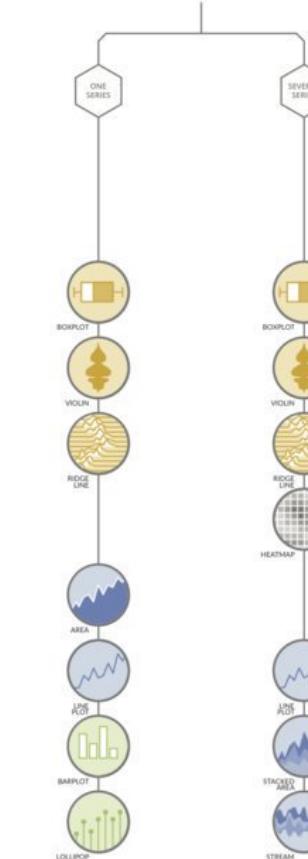
RELATIONAL



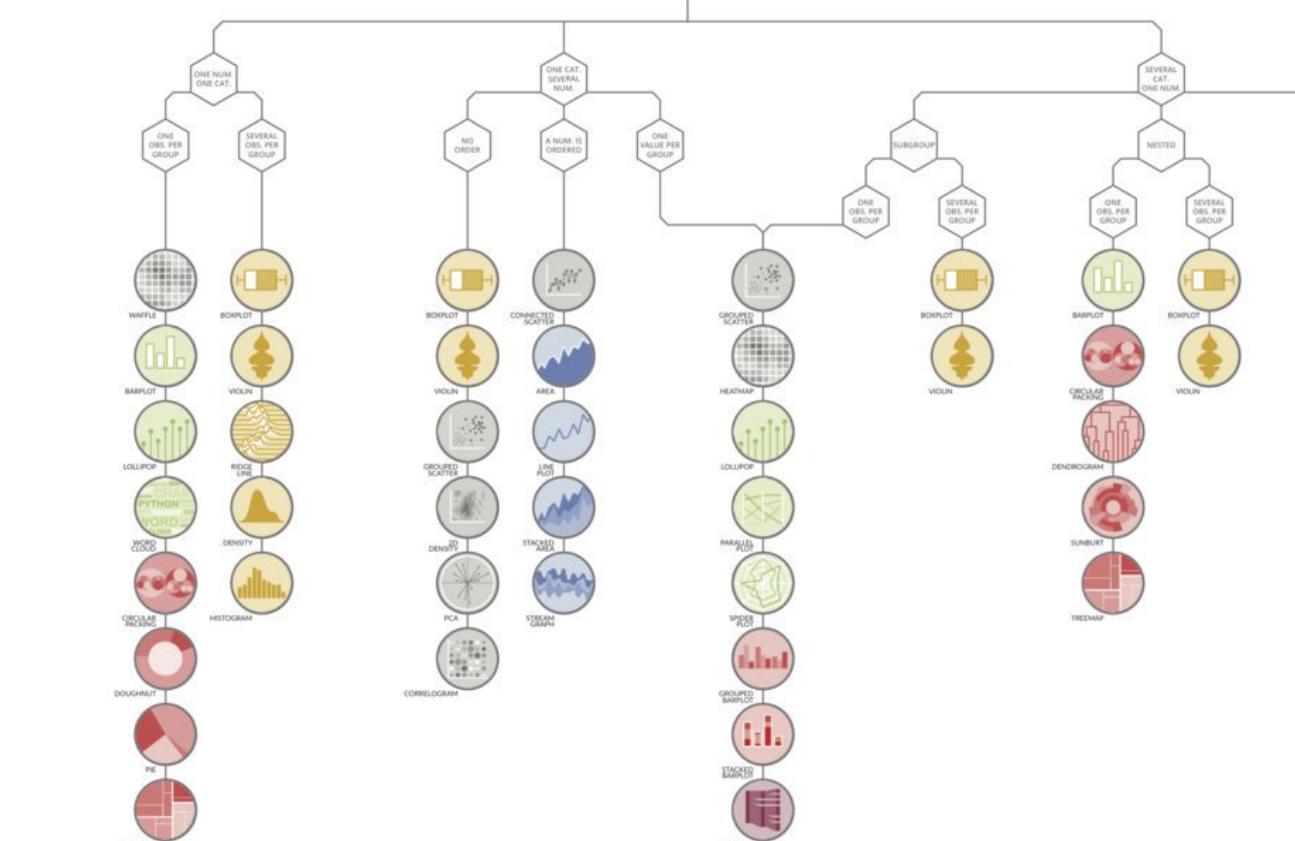
MAP



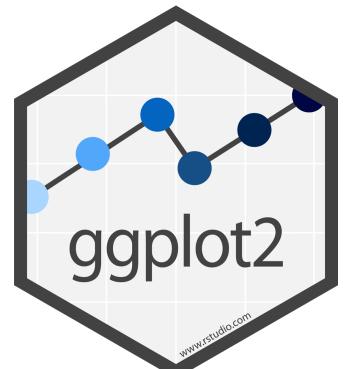
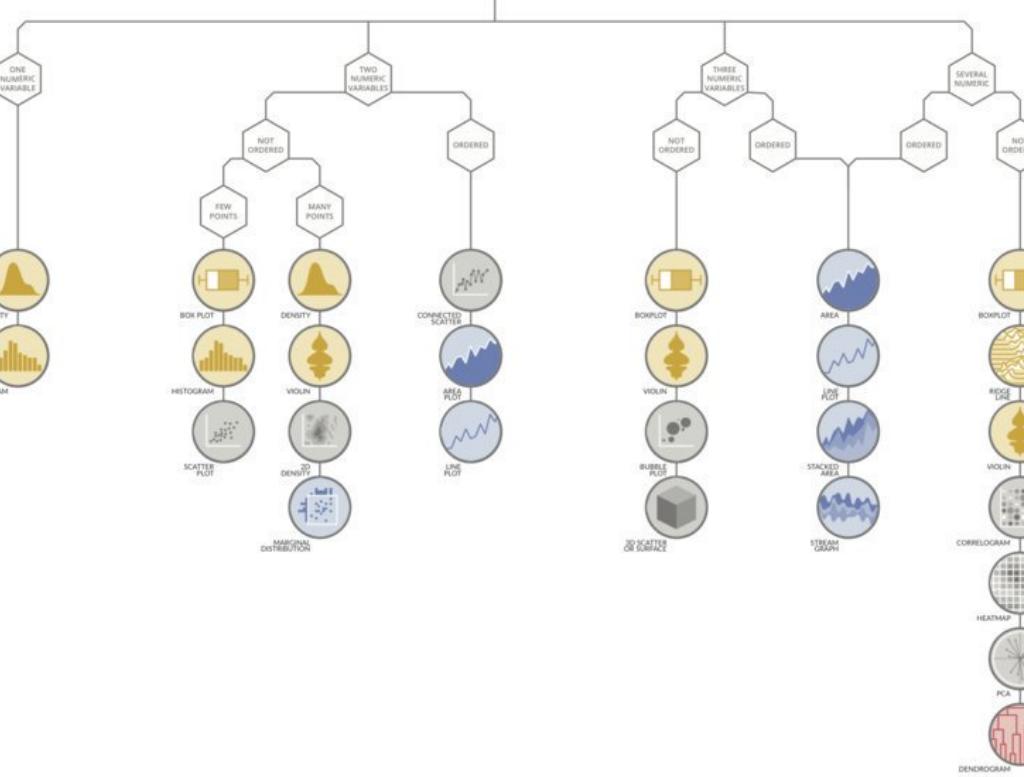
TIME SERIES



CATEGORIC AND NUMERIC



NUMERIC



The R Graph Gallery

A SET OF PRE-BUILT THEMES

It is possible to customize any part of a `ggplot2` chart thanks to the `theme()` function. Fortunately, heaps of pre-built themes are available, allowing to get a good style with one more line of code only. Here is a glimpse of the available themes. [See code](#)

→ `ggplot2`

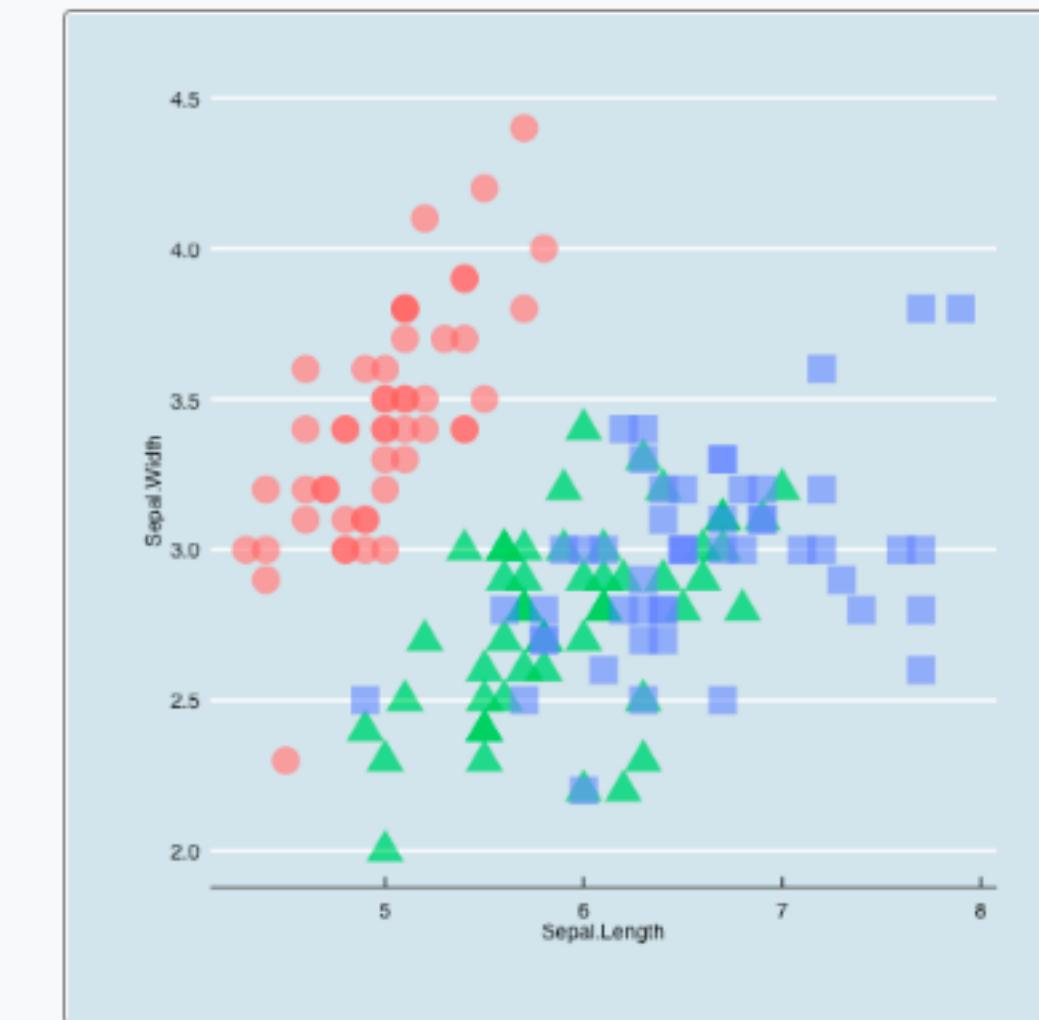
`default` `theme_bw()` `theme_minimal()`
`theme_classic()` `theme_gray()`

→ `ggthemes`

`theme_excel()` `theme_economist()`
`theme_fivethirtyeight()` `theme_tufte()`
`theme_gdocs()` `theme_wsj()` `theme_calc()`
`theme_hc()`

→ `other`

`theme_article()` `theme_pubclean()`
`theme_bigstatsr()` `theme_ipsum()`



r4ds.had.co.nz

Complete and
free online

The ultimate
resource for all of
today

ggplot2

RStudio Cloud

R for Data Science

r4ds.had.co.nz

Welcome

1 Introduction

I Explore

2 Introduction

3 Data visualisation

4 Workflow: basics

5 Data transformation

6 Workflow: scripts

7 Exploratory Data Analysis

8 Workflow: projects

II Wrangle

9 Introduction

10 Tibbles

11 Data import

12 Tidy data

13 Relational data

14 Strings

15 Factors

R for Data Science

Garrett Grolemund

Hadley Wickham

Welcome

This is the website for “**R for Data Science**”. This book will teach you how to do data science with R: You’ll learn how to get your data into R, get it into the most useful structure, transform it, visualise it and model it. In this book, you will find a practicum of skills for data science. Just as a chemist learns how to clean test tubes and stock a lab, you’ll learn how to clean data and draw plots—and many other things besides. These are the skills that allow data science to happen, and here you will find the best practices for doing each of these things with R. You’ll learn how to use the grammar of graphics, literate programming, and reproducible research to save time. You’ll also learn how to

Visualize Data with

