

ggplot2 grammar cheat sheet

Evangeline Reynolds

11/12/2019

“Hacker News” exchange about ggplot2:

sk5t 8 months ago [-]

I can't help but wonder—as someone who has flirted with R but certainly not committed—how much time and practice it takes to acquire decent fluency with these half-magical incantations to become productive.

jhbadger 8 months ago [-]

Not that long. The trick is to really understand what each clause is doing. It's easy to just find some code on the Web for a graph similar to what you want and tweak it for your data, and I did that myself in the beginning, but then they will remain “half-magical incantations” as you put it. The thing about ggplot is nothing is there arbitrarily – it is really describing how the plot is put together.

ggplot2 grammar:

1. **The Declarative Mood**, declaring data, `ggplot(data = gapminder)`
2. **The Interogative Mood**, aesthetic mapping, `aes(color = pop)`
 - a. **Modifiers I**, changing aes scales, `scale__color__viridis__d()`
 - b. **Modifiers II**, changing aes and plot labels, `labs(color = “continent”)`
 - c. **Modifiers III**, changing the coordinate system, `coord_polar()`
3. **Nouns**, geometric objects, `geom_point()`
4. **The Conditional Mood** aesthetics tied to specific “geoms”
 - a. data, `geom_point(data = gapminder)`
 - b. mapped aesthetic, `geom_point(aes(color = continent))`
 - c. unmapped aesthetics (**i.e. The Imperative Mood**), `geom_point(color = “blue”)`
5. **Interjections**, annotation - geoms not connected to data, `annotate(geom = “point”, ...)`
6. **Punctuation**, small multiple plots, `facet_wrap(~continent)`
7. **Greetings**, changing plot look and feel, `theme_minimal()`
8. **The Written Language**, saving plots, `ggsave(file = “plot.png”, plot = g)`

```

# some data wrangling
library(tidyverse)
library(gapminder)
gapminder %>%
  filter(year == 2002) ->
gapminder_2002
gapminder_2002 %>%
  filter(continent == "Europe") ->
gapminder_2002_europe

# Declarative mood - declare data
ggplot(data = gapminder_2002) +
# Interogative mood - ask aesthetic for representation
  # x, y, color, fill, linetype, linewidth, size, alpha, shape
  aes(x = gdpPercap) + # x position
# Noun - declare geometric object
  geom_rug() + # above aes are required for point
# Modifiers I - changing default aesthetic scales
  scale_x_log10() + # also scale_y_*, scale_color_*, scale_size_*
# Modifiers II - changing default labels
  labs(x = "GDP per cap") + # also y axis and color, size etc. legend labels
# Modifiers IIb. Note that labs is also used to adjust *plot* labels()
  labs(title = "GDP in 2002") + # also subtitle, caption, tag
# Modifiers III - changing default coordinate system (cartesian)
  coord_flip() + # flip x and y coordinates, also coord_equal, coord_polar ...
# Conditional Mood - for specific geoms, locally defining:
  # 1) data, 2) aesthetic representation, 3) *unmapped* aesthetics (the Imperative mood)
  geom_rug(
    data = gapminder_2002_europe, # local data
    aes(color = pop), # local aesthetics representation
    size = .5 # local across-the-board aesthetics
  ) +
# Interjections - annotation - geometric objects not tied to data
  annotate(geom = "rug", # define which geometric shape
    x = 10000, # define required aesthetic for geom
    color = "red" # define optional aesthetic
  ) +
# Idioms (don't quite fit the grammar) annotation geoms...
  geom_abline(slope = .01, intercept = 0) + # line equation
  geom_hline(yintercept = 70, # horizontal line
    linetype = "dotted") +
  geom_vline(xintercept = c(1000, 10000), # vertical line
    linetype = "dashed") +
# Punctuation - small multiples with facet statements
  facet_wrap(~ continent, ncol = 2) + # small multiples by one variable
# Greetings - what meets the eye - changing default plot look and feel
  theme_minimal() -> # also theme_bw, theme_classic, theme_grey
g
# Written Language
ggsave(filename = "my_plot.png", plot = g)

```