Recently one of my coworkers showed me a ggplot and although it is not wrong, it is also not ideal. Here is the TL:DR :

Whenever you find yourself adding multiple geom\_\* to show different groups, reshape your data

In software engineering there are things called [antipatterns](https://en.wikipedia.org/wiki/Anti-pattern#Software_engineering), ways of programming  
that lead you into potential trouble. This is one of them.

I’m not saying it is incorrect, but it might lead you into trouble.

example: we have some data, some different calculations and we want to plot that.

**I load tidyverse and create a modified mtcars set in this hidden part,  
but if you don’t care you can leave it unopened**

*Cool how this folds away right? It even works on github markdown, if you want to know how I did this, I explain it below*

library(tidyverse) # I started loading magrittr, ggplot2 and tidyr, and realised

## ── Attaching packages ─────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0 ✔ purrr 0.3.0

## ✔ tibble 2.0.1 ✔ dplyr 0.7.8

## ✔ tidyr 0.8.2 ✔ stringr 1.4.0

## ✔ readr 1.3.1 ✔ forcats 0.3.0

## ── Conflicts ────────────────────────────────────────────────── tidyverse\_conflicts() ──

## ✖ dplyr::filter() masks stats::filter()

## ✖ dplyr::lag() masks stats::lag()

# I needed dplyr too, at some point loading tidyverse is simply easiest.

very\_serious\_data <-

mtcars %>%

as\_tibble(rownames = "carname") %>%

group\_by(cyl) %>%

mutate(

mpg\_hp = mpg/hp,

first\_letter = str\_extract(carname, "^[A-z]"),

mpg\_hp\_c = mpg\_hp/mean(mpg\_hp),# grouped mean

mpg\_hp\_am = mpg\_hp+ am

)

Now the data (mtcars) and calculations don’t really make sense but they are here to show you the  
antipattern. I created 3 variants of dividing mpg (miles per gallon) by hp (horse power)

**The antipattern**

We have a dataset with multiple variables (columns) and want to plot  
one against the other, so far so good.

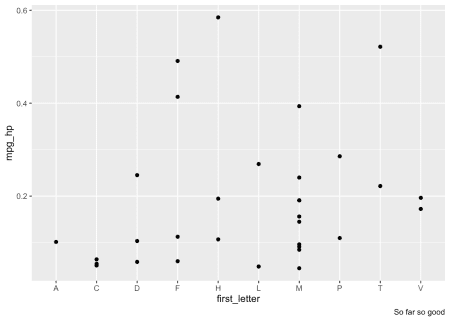
What is the effect of mpg\_hp for every first letter of the cars?

very\_serious\_data %>%

ggplot(aes(first\_letter, mpg\_hp))+

geom\_point()+

labs(caption = "So far so good")



But you might wonder what the other transformations of that variable do?  
You can just add a new geom\_point, but maybe with a different color?  
And to see the dots that overlap you might make them a little opaque.

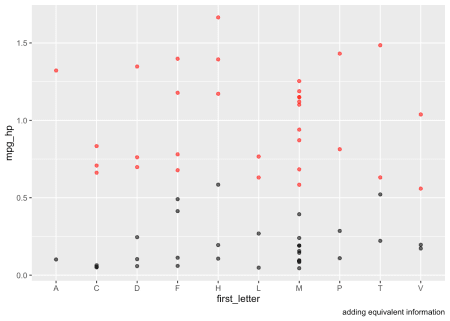
very\_serious\_data %>%

ggplot(aes(first\_letter, mpg\_hp))+

geom\_point(alpha = 2/3)+

geom\_point(aes(y = mpg\_hp\_c), color = "red", alpha = 2/3)+

labs(caption = "adding equivalent information")



And maybe the third one too?

very\_serious\_data %>%

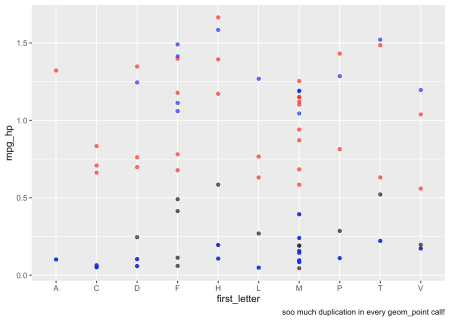
ggplot(aes(first\_letter, mpg\_hp))+

geom\_point(alpha = 2/3)+

geom\_point(aes(y = mpg\_hp\_c), color = "red", alpha = 2/3)+

geom\_point(aes(y = mpg\_hp\_am), color = "blue", alpha = 2/3)+

labs(caption = "soo much duplication in every geom\_point call!")



This results in lots of code duplication for specifying what is essentially  
the same for every geom\_point() call. It’s also really hard to add a legend  
now.

**What is the alternative?**

Whenever you find yourself adding multiple geom\_\* to show different groups, reshape your data

Gather the columns that are essentially representing the group and reshape  
the data into a format more suitable for plotting. Bonus: automatic correct labeling.

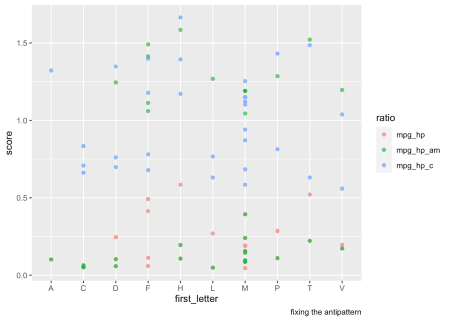
very\_serious\_data %>%

gather(key = "ratio", value = "score", mpg\_hp, mpg\_hp\_c, mpg\_hp\_am ) %>%

ggplot(aes(first\_letter, score, color = ratio))+

geom\_point(alpha = 2/3)+

labs(caption = "fixing the antipattern")



And that’s it.

Mari also tells you it will work

**State of the machine**

At the moment of creation (when I knitted this document ) this was the state of my machine: **click here to expand**

sessioninfo::session\_info()

## ─ Session info ──────────────────────────────────────────────────────────

## setting value

## version R version 3.5.2 (2018-12-20)

## os Ubuntu 16.04.5 LTS

## system x86\_64, linux-gnu

## ui X11

## language en\_US

## collate en\_US.UTF-8

## ctype en\_US.UTF-8

## tz Europe/Amsterdam

## date 2019-03-07

##