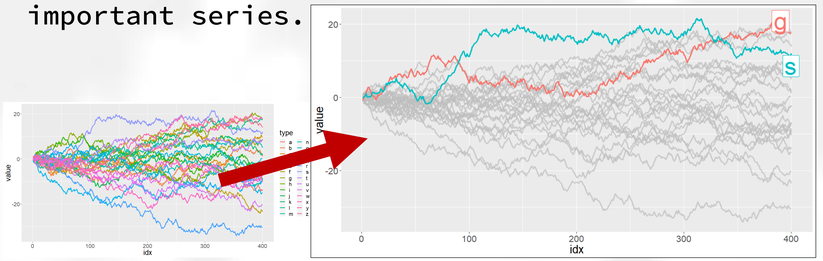
gghighlight: Highlight ggplot2 with Predicates

Hiroaki Yutani
(@yutannihilat_en)

Why highlight?

Visualization is hard when there are so many data series that you can't distinguish them by their colors. So, here arises the need for "highlighting," setting colors only on a few important series.



Usage

gghighlight is a package to add highlighting functionality to ggplot2. It's just as easy as adding gghighlight() to an usual ggplot object.

```
ggplot(d, aes(idx, value)) +
  geom_line(aes(colour = type)) +

  gghighlight(max(value) > 20)
```

Usual ggplot2 code

gghighlight

predicate

gghighlight() takes predicates, or conditions, to determine which data to highlight. In the predicates, the column names of the data can be referred directly just like with dplyr's filter().

Predicates can be more than one. It is useful to add/modify conditions interactively for EDA.

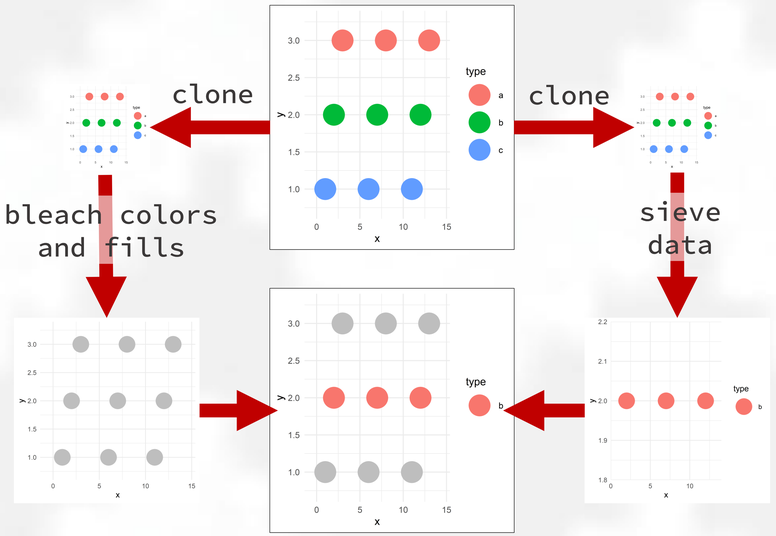
```
gghighlight(max(value) > 15,
            mean(flag) > 0.55,
            ...)
```

Under the hood of gghighlight

Highlighting is basically a combination of:

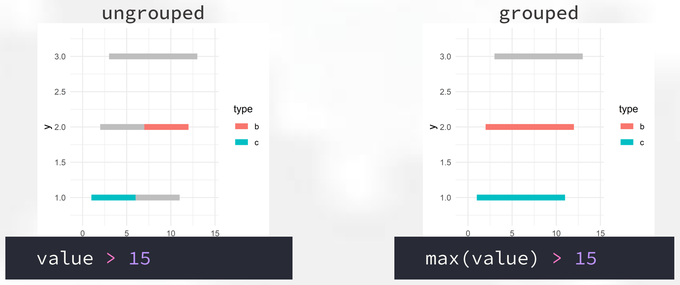
- bleaching, removing the color-related aesthetics (*colour* and *fill*) and
- sieving, choosing the data points/series where conditions are true.

Both can be done with bare ggplot2; gghighlight is just a shortcut of this powered by tidy eval.



Ungrouped vs grouped

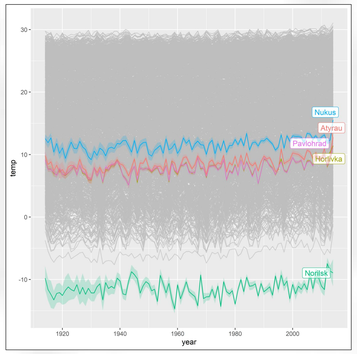
A visualization might be done with the groups. For example, lines are drawn for each group, which requires the predicate to be group-wise.



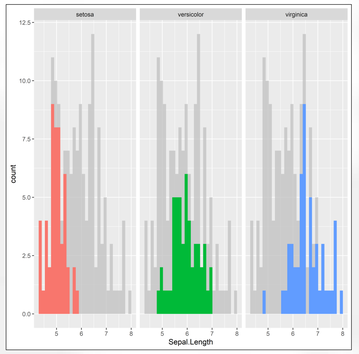
Examples

gghighlight can highlight almost any type of visualizations. Besides, as the result is also an usual ggplot object, it's fully customizable.

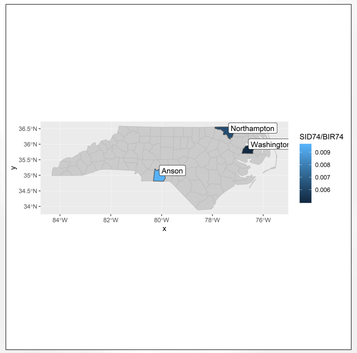
geom_line() + geom_ribbon()



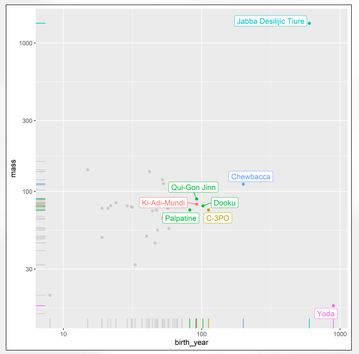
geom_bar() + facet



geom_sf()



geom_point() + geom_rug()



Limitations

gghighlight assumes that the sieved layers exactly overwrap the bleached layers, but there are some cases this assumption is false:

- the *Position* locates the series relatively (e.g. *position_dodge()*)
- the *Stat* calculates relative values from the whole data (e.g. *stat_density_ridges()*)