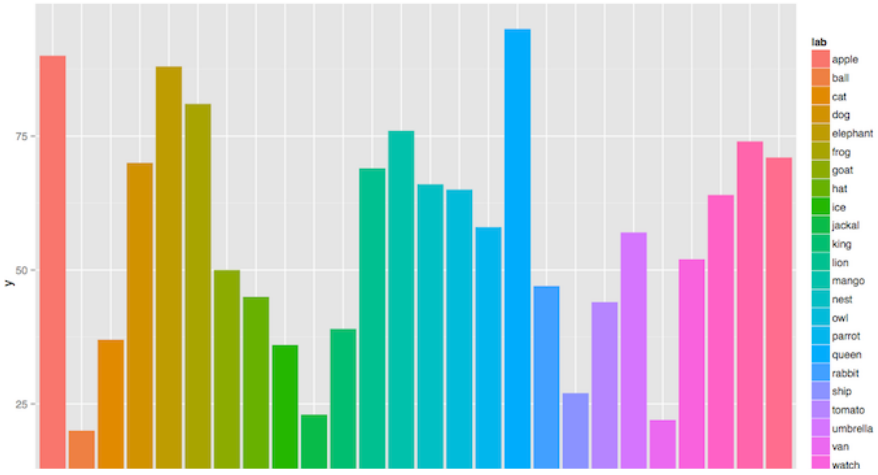
Randomising qualitative colours for large sets in ggplot

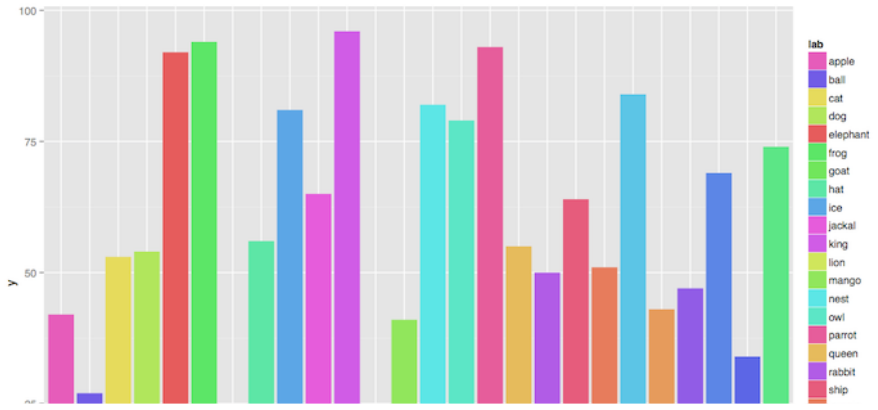
If you're mapping qualitatively to colour for a large number of groups, ggplot's automatic colour assignment plots very similar colours adjacently, making it hard to see which refer to which key etc. To illustrate:

```
require(ggplot2); require(stringr)
df = data.frame(x = letters, y = sample(20:100,26), lab=word("apple ball cat dog elephant
frog goat hat ice jackal king lion mango nest owl parrot queen rabbit ship tomato umbrella
van watch xylophone yatch zebra", 1:26))
p = ggplot(df, aes(x, y, fill=lab)) + geom_bar(stat="identity")
p + scale_fill_discrete()
```



Its possible to mix up some random colours manually:

```
cols = rainbow(26, s=.6, v=.9)[sample(1:26,26)]
p + scale_fill_manual(values=cols)
```



.. resulting in more useful breakup of the rainbow, but this seems clumsy, still leaves some colours clumped together and is generally not ideal. Does ggplot have a native method to achieve something like this (but hopefully better)?

r ggplot2

edited Jan 26 '14 at 16:18
Julius
11.1k 5 35 51

asked Jan 25 '14 at 15:50

geotheory
3,120 2 21 46

2 My advice, don't. It's far too much information in one plot and the colours are not significantly different enough to translate that information usefully for the viewer. Do your best to separate or group the results into meaningful or otherwise functional breaks - your readers will thank you. Notwithstanding the 10% that are colorblind anyways. – Brandon Bertelsen Jan 27 '14 at 6:32

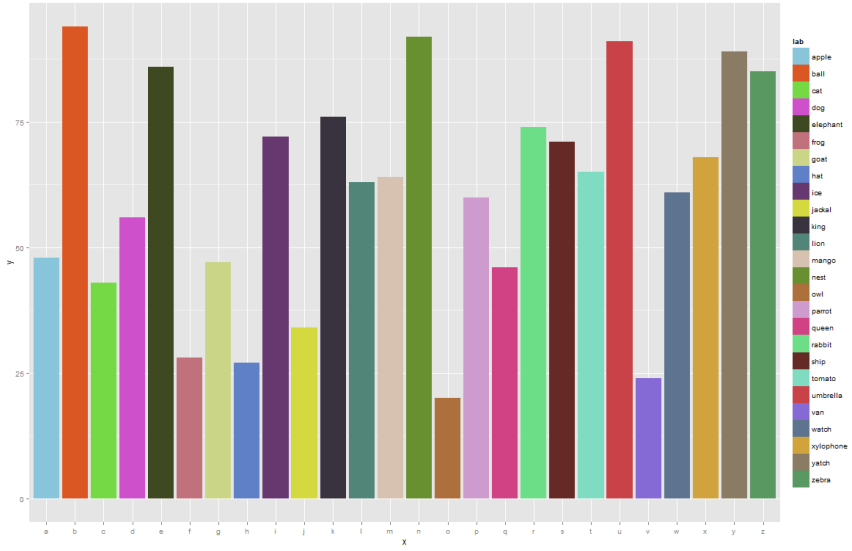
Speaking directly from experience here: stackoverflow.com/questions/13616515/… – Brandon Bertelsen Jan 27 '14 at 6:40

Brandon, I appreciate the perceptual issue you raise and normally I'd agree. My issue is I'm dealing with 30+ categories of non-groupable data with quite lengthy titles (too big for x axis), and I want to avoid a 90 degree rotation. – geotheory Jan 27 '14 at 12:33

1 Answer

Producing a good palette for that many colours is indeed a difficult task. However, there is one solution which may be helpful. Some time ago I forked this repo and found a reference to iWantHue. As far as I can see, the resulting palette is already mixed, so that neighbouring colours look distinguishable.

For instance, for your example I have



Just in case, the palette is

```
"#89C5DA", "#DA5724", "#74D944", "#CE50CA", "#3F4921", "#C0717C", "#CBD588", "#5F7FC7",
"#673770", "#D3D93E", "#38333E", "#508578", "#D7C1B1", "#689030", "#AD6F3B", "#CD9BCD",
"#D14285", "#6DDE88", "#652926", "#7FDCC0", "#C84248", "#8569D5", "#5E738F", "#D1A33D",
"#8A7C64", "#599861"
```

answered Jan 27 '14 at 6:25

tonytonov
9,267 10 22 43

Thanks tonyronov this is useful, although I think in essence this variablisation of saturation and value is achievable with a tweek to my code: `col.par = function(n) sample(seq(0.3, 1, length.out=50),n); cols = rainbow(26, s=col.par(26), v=col.par(26))[sample(1:26,26)]` . – geotheory Jan 27 '14 at 12:42

It is. However, when you use `sample` , two or more close-looking colours may fall close to each other randomly. I think the proposed palette is already shuffled properly. – tonytonov Jan 27 '14 at 12:45

Fair point. What is the key R code you're referring to? I can see a reference to iWantHue in ggplot.Rnw but I don't recognise the code format. – geotheory Jan 27 '14 at 12:59

Well, seems iWantHue doest not provide any sort of API, so I just took a JSON from there. BTW, more ideas here. – tonytonov Jan 27 '14 at 13:04

If I understand the question correctly, the presentation is compiled by latex and knitr. – tonytonov Jan 27 '14 at 13:10