

Stat405

Polishing, continued

Hadley Wickham

1. Coming up next.

2. Project.

3. Colour scales

4. Themes.

	Monday	Wednesday
Nov 9	Polishing	Spam
Nov 16	Posters / Project	Spam
Nov 23	Debugging	Thanksgiving
Nov 30	Professional development	Poster pres

Project 3

Your turn to find an interesting data source and analyse it. Any data is acceptable but it needs to be large (>1,000 points).

If you don't have any ideas, try looking at <http://delicious.com/hadley/data>

Deliverables

Poster.

Single double-sided page handout, summarising the major findings on the poster.

The R code used to clean the data and produce the plots. (**No manual steps!**)

Timeline

Nov 17-19. Meet to discuss choice of data and initial analysis ideas. Data preparation.

Nov 23-30. Review of draft poster (optional, but highly recommended).

Dec 2. Final poster presentation. Turn in electronic copy of handout & code.

Polishing

Tools

Scales. Used to override default perceptual mappings, and tune parameters of axes and legends.

Themes: control presentation of non-data elements.

Your turn

Recall the four key parameters that all scales have in common. What are their names and what do they do?

```
p <- qplot(carat, price, data = diamonds, geom = "hex")

# First argument (name) controls legend title
p + scale_fill_continuous("Count")

# Breaks and labels control legend keys
p + scale_fill_continuous(breaks = c(1000, 3500, 7000))
p + scale_fill_continuous(breaks = c(0, 4000, 8000))

# Why don't 0 and 8000 have colours?
p + scale_fill_continuous(breaks = c(0, 4000, 8000),
  limits = c(0, 8000))

# Can use labels to make more human readable
breaks <- c(0, 2000, 4000, 6000, 8000)
labels <- format(breaks, big.mark = ",")
p + scale_fill_continuous(breaks = breaks, labels = labels,
  limits = c(0, 8000))
```

Default colour scales

Discrete: evenly spaced hues of equal chroma and luminance. No colour appears more important than any other. Does not imply order.

Continuous: evenly spaced hues between two colours.

Alternatives

Discrete: **hue**, brewer, manual

Continuous: **gradient**, gradient2,
gradientn

Color brewer

Cynthia Brewer applied the basic principles and then rigorously tested the results to produce a selection of good palettes, particularly tailored for maps:
<http://colorbrewer2.org/>

Can use `cut_interval()` or `cut_number()` to convert continuous to discrete.

```
# Fancy looking trigonometric function
vals <- seq(-4 * pi, 4 * pi, len = 50)
df <- expand.grid(x = vals, y = vals)
df$r <- with(df, sqrt(x ^ 2 + y ^ 2))
df$z <- with(df, cos(r ^ 2) * exp(- r / 6))
df$z_cut <- cut_interval(df$z, 9)
```

```
(p1 <- qplot(x, y, data = df, fill = z,
  geom = "tile"))
(p2 <- qplot(x, y, data = df, fill = z_cut,
  geom = "tile"))
```

```
p1 + scale_fill_gradient(low = "white",  
  high = "black")  
  
# Highlight deviations  
p1 + scale_fill_gradient2()  
p1 + scale_fill_gradient2(breaks = seq(-1, 1,  
  by = 0.25), limits = c(-1, 1))  
p1 + scale_fill_gradient2(mid = "white",  
  low = "black", high = "black")  
  
p2 + scale_fill_brewer(pal = "Blues")
```

Your turn

Read through the examples for
`scale_colour_brewer`,
`scale_colour_gradient2` and
`scale_colour_gradientn`.

Experiment!

Colour blindness

7-10% of men are red-green colour “blind”. (Many other rarer types of colour blindness)

Solutions: avoid red-green contrasts; use redundant mappings; **test**. I like color oracle: <http://colororacle.cartography.ch>

Themes

Visual appearance

So far have only discussed how to get the data displayed the way you want, focussing on the essence of the plot.

Themes give you a huge amount of control over the appearance of the plot, the choice of background colours, fonts and so on.

```
# Two built in themes. The default:  
qplot(carat, price, data = diamonds)
```

```
# And a theme with a white background:  
qplot(carat, price, data = diamonds) + theme_bw()
```

```
# Use theme_set if you want it to apply to every  
# future plot.  
theme_set(theme_bw())
```

```
# This is the best way of seeing all the default  
# options  
theme_bw()  
theme_grey()
```

Elements

You can also make your own theme, or modify an existing one.

Themes are made up of elements which can be one of: `theme_line`, `theme_segment`, `theme_text`, `theme_rect`, `theme_blank`

Gives you a lot of control over plot appearance.

Elements

Axis: axis.line, axis.text.x, axis.text.y,
axis.ticks, axis.title.x, axis.title.y

Legend: legend.background, legend.key,
legend.text, legend.title

Panel: panel.background, panel.border,
panel.grid.major, panel.grid.minor

Strip: strip.background, strip.text.x,
strip.text.y

```
p <- qplot(displ, hwy, data = mpg) +  
  opts(title = "Bigger engines are less efficient")  
  
# To modify a plot  
p  
p + opts(plot.title =  
  theme_text(size = 12, face = "bold"))  
p + opts(plot.title = theme_text(colour = "red"))  
p + opts(plot.title = theme_text(angle = 45))  
p + opts(plot.title = theme_text(hjust = 1))
```

Your turn

Fix the overlapping y labels on this plot:

```
qplot(reorder(model, hwy), hwy, data =  
mpg)
```

Rotate the labels on these strips so they are easier to read.

```
qplot(hwy, reorder(model, hwy), data =  
mpg) + facet_grid(manufacturer ~ .,  
scales = "free", space = "free")
```