Stat405

Grammar of graphics

Hadley Wickham

1. Grammar of graphics
2. Communication graphics
3. Scales
4. Themes

Grammar of graphics

“If any number of magnitudes are each the same multiple of the same number of other magnitudes, then the sum is that multiple of the sum.”

Euclid, ~300 BC

"If any number of magnitudes are each the same multiple of the same number of other magnitudes, then the sum is that multiple of the sum."
Euclid, ~300 BC

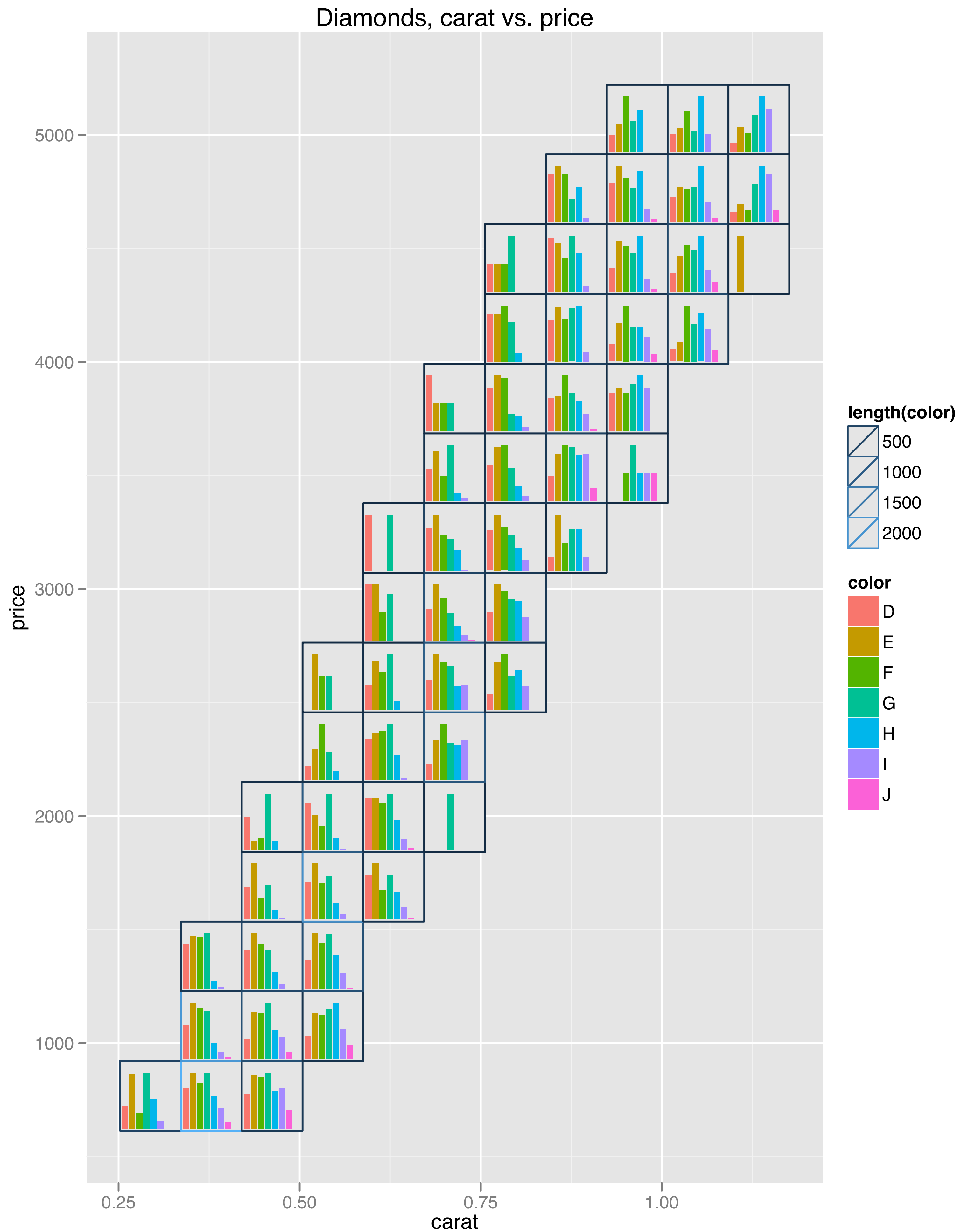
$$m(\Sigma x) = \Sigma(mx)$$

The grammar of graphics

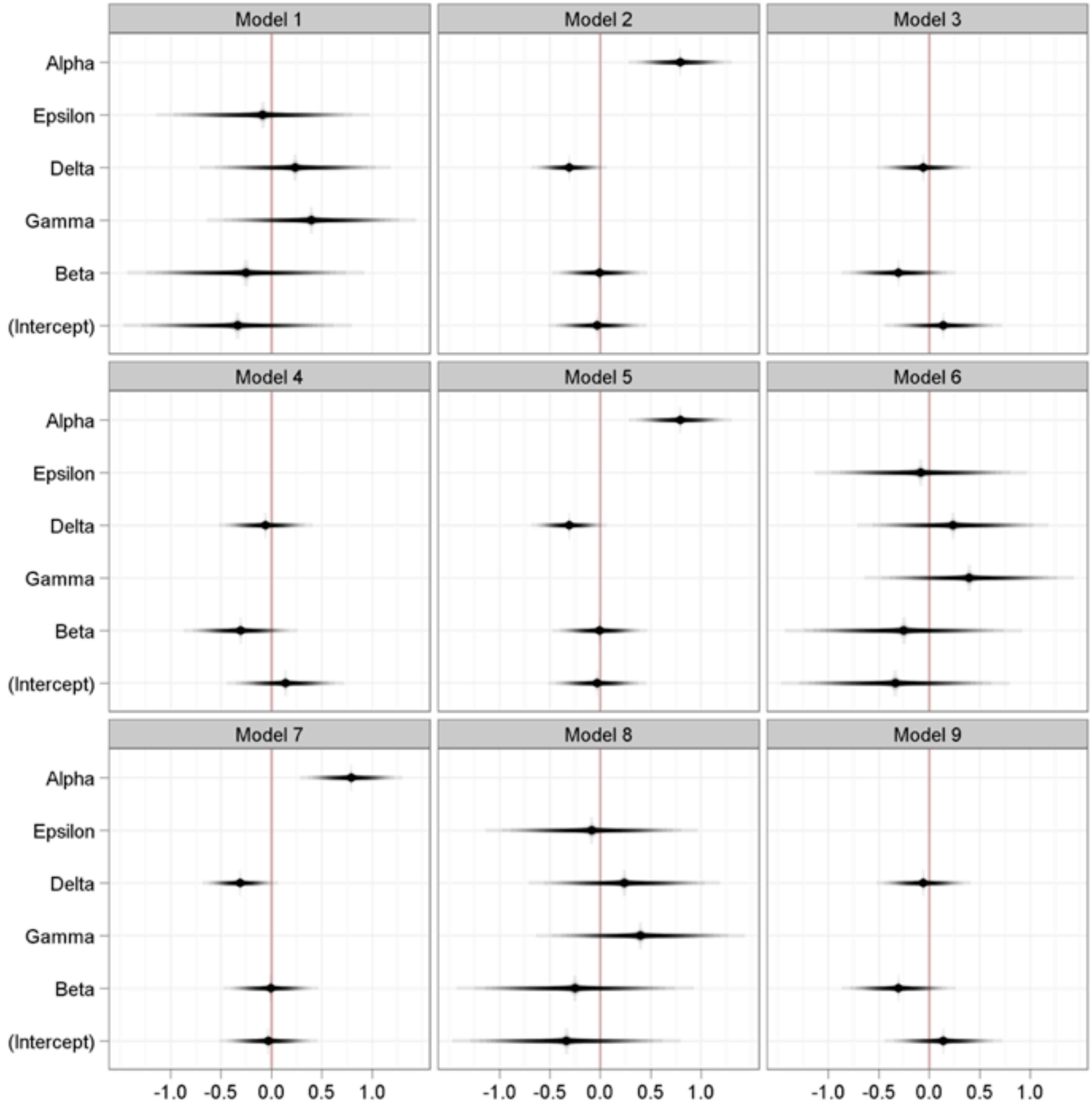
An abstraction which makes thinking about, reasoning about and communicating graphics easier.

Developed by Leland Wilkinson, particularly in “The Grammar of Graphics” 1999/2005

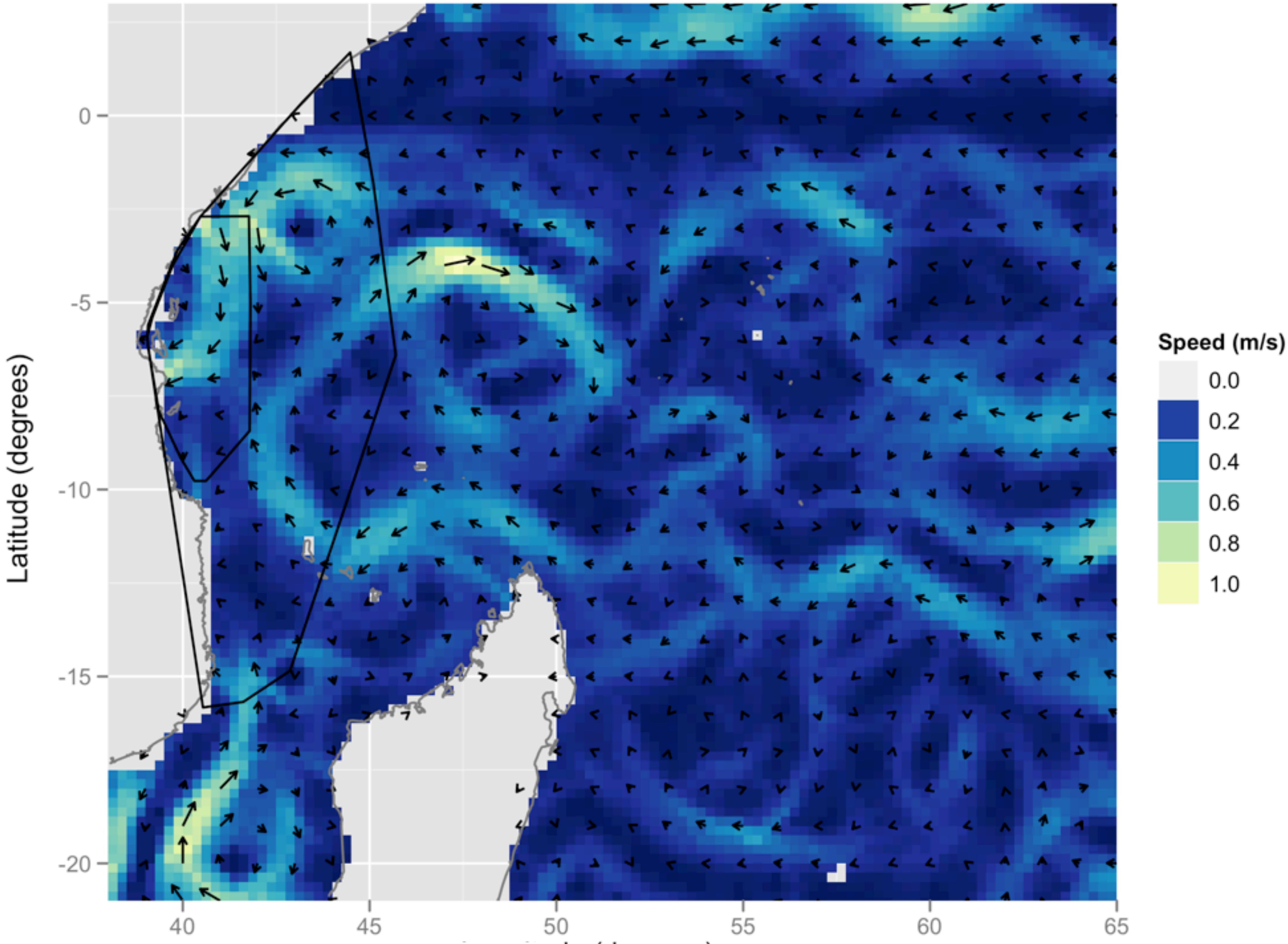
You’ve been using it in ggplot2 without knowing it! But to do more, you need to learn more about the theory.



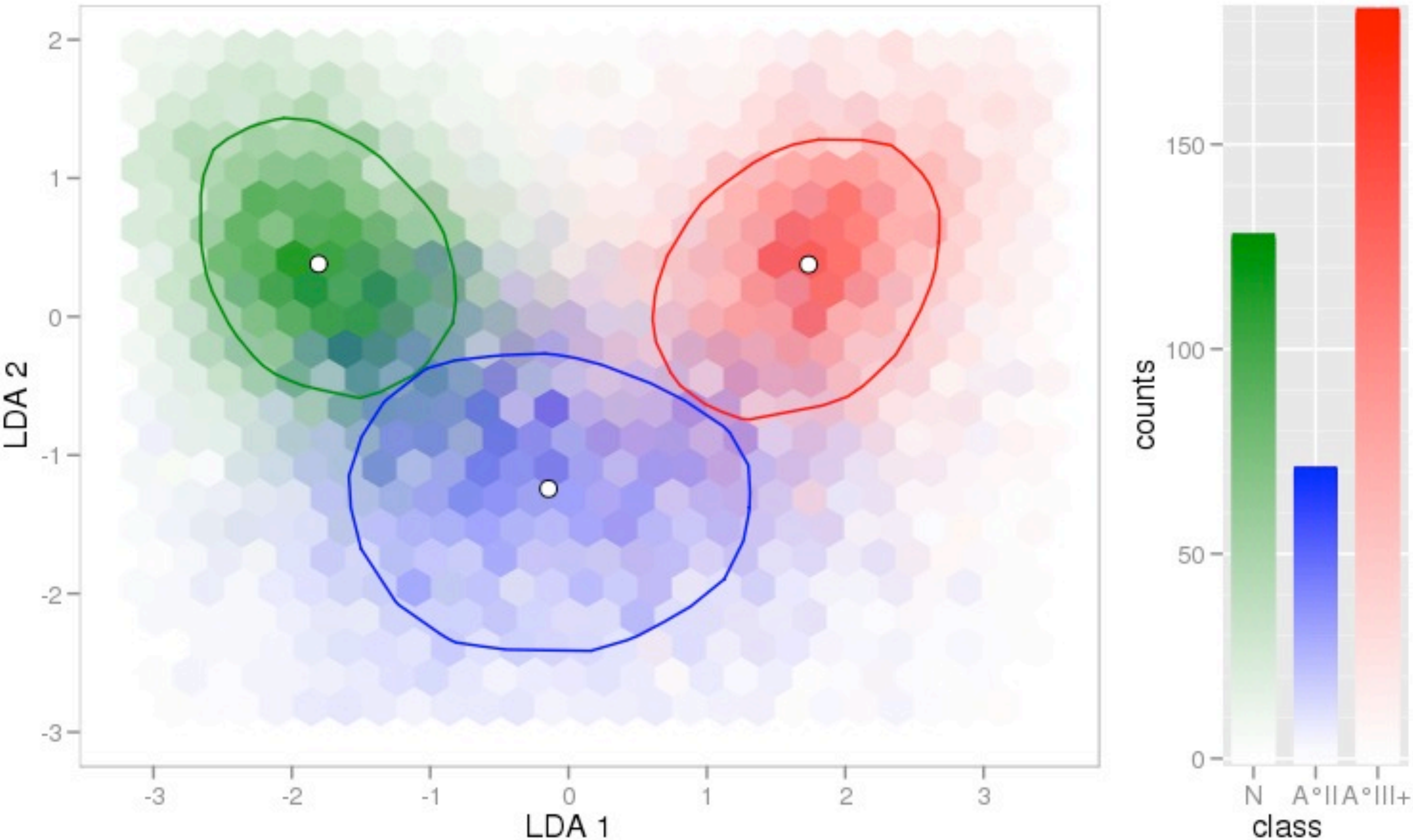
Garrett Grolemund, http://blog.revolutionanalytics.com/2012/09/visualize-complex-data-with-subplots.html



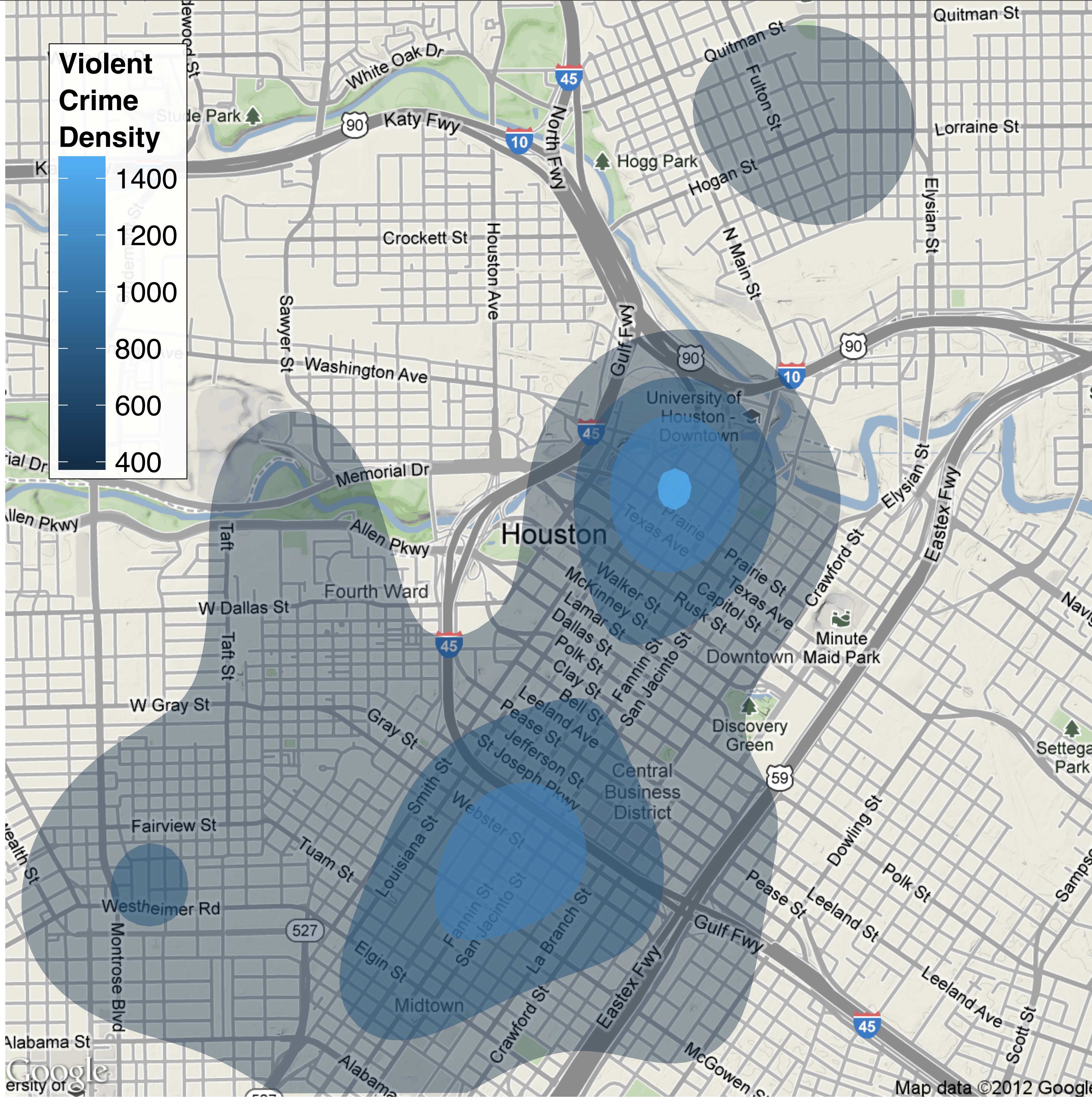
David B Sparks, http://bit.ly/hn54NW



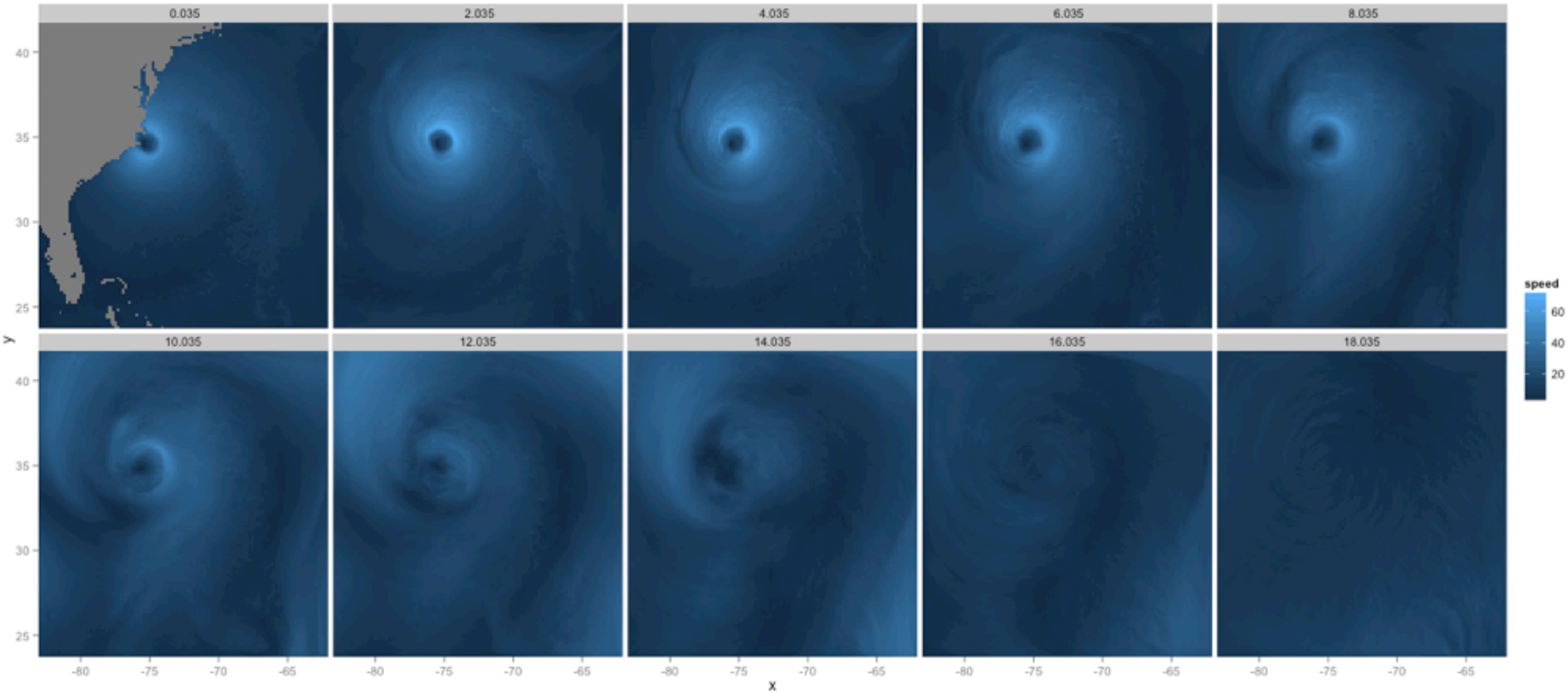
Charlotte Wickham, http://cwick.co.nz/



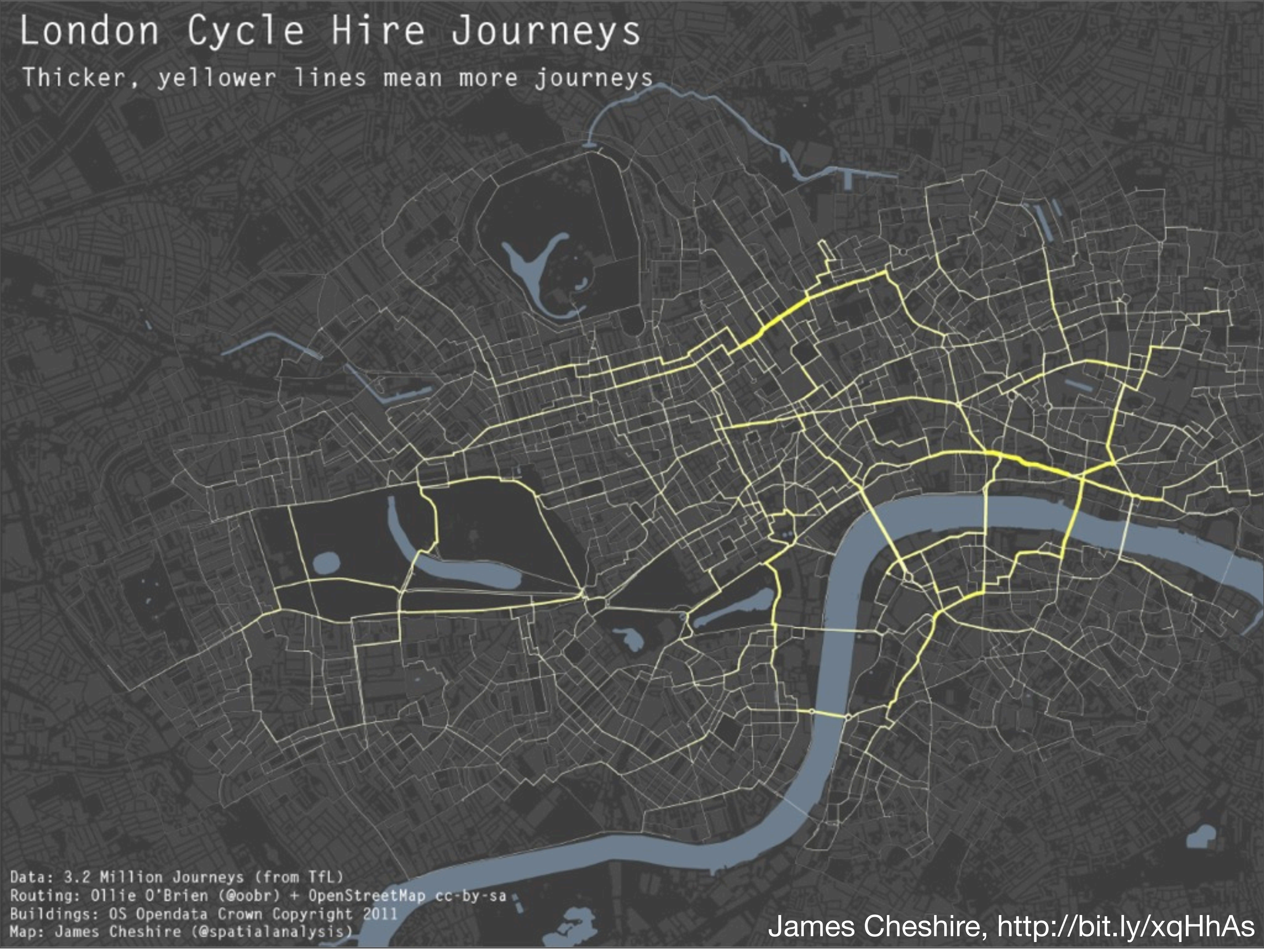
Claudia Beleites, http://bit.ly/yNqlpz



David Kahle, https://dl.dropbox.com/u/24648660/ggmap%20useR%202012.pdf



Winston Chang, http://shop.oreilly.com/product/0636920023135.do


London Cycle Hire Journeys
Thicker, yellower lines mean more journeys
Data: 3.2 Million Journeys (from TfL)
Routing: Ollie O'Brien (@oobr) + OpenStreetMap cc-by-sa
Buildings: OS Opendata Crown Copyright 2011
Map: James Cheshire (@spatialanalysis)
James Cheshire, http://bit.ly/xqHhAs

What is a plot?

A set of **layers**

A set of scales

A coordinate system

A facetting specification

What is a layer?

- Data
- Aesthetic mappings (**aes**)
- A geometric object (**geom**)
- A statistical transformation (**stat**)
- A position adjustment (**position**)

```
layer(geom, stat, position, data, mapping, ...)

layer(
  data = mpg,
  mapping = aes(x = displ, y = hwy),
  geom = "point",
  stat = "identity",
  position = "identity"
)

layer(
  data = diamonds,
  mapping = aes(x = carat),
  geom = "bar",
  stat = "bin",
  position = "stack"
)
```

```
# A lot of typing!

layer(
  data = mpg,
  mapping = aes(x = displ, y = hwy),
  geom = "point",
  stat = "identity",
  position = "identity"
)

# Every geom has an associated default statistic
# (and vice versa), and position adjustment.

geom_point(aes(displ, hwy), data = mpg)
geom_histogram(aes(carat), data = diamonds)
```

```
# To actually create the plot
ggplot() +
  geom_point(aes(displ, hwy), data = mpg)

ggplot() +
  geom_histogram(aes(carat), data = diamonds)
```

```
# Multiple layers
ggplot() +
  geom_point(data = mpg, aes(displ, hwy)) +
  geom_smooth(data = mpg, aes(displ, hwy))

# Avoid redundancy:
ggplot(aes(displ, hwy), data = mpg) +
  geom_point() +
  geom_smooth()
```

```
# Different layers can have different aesthetics
ggplot(mpg, aes(displ, hwy)) +
  geom_smooth() +
  geom_point(aes(colour = class))

ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point() +
  geom_smooth(method = "lm", se = F)

ggplot(mpg, aes(displ, hwy, group = class)) +
  geom_point(aes(colour = class)) +
  geom_smooth(method = "lm", se = F)

ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(colour = class)) +
  geom_line(aes(group = class), stat = "smooth",
    method = "lm", se = F)
```

```
# ggplot doesn't stop you from doing dumb things

ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  geom_point(aes(cyl, displ))
```

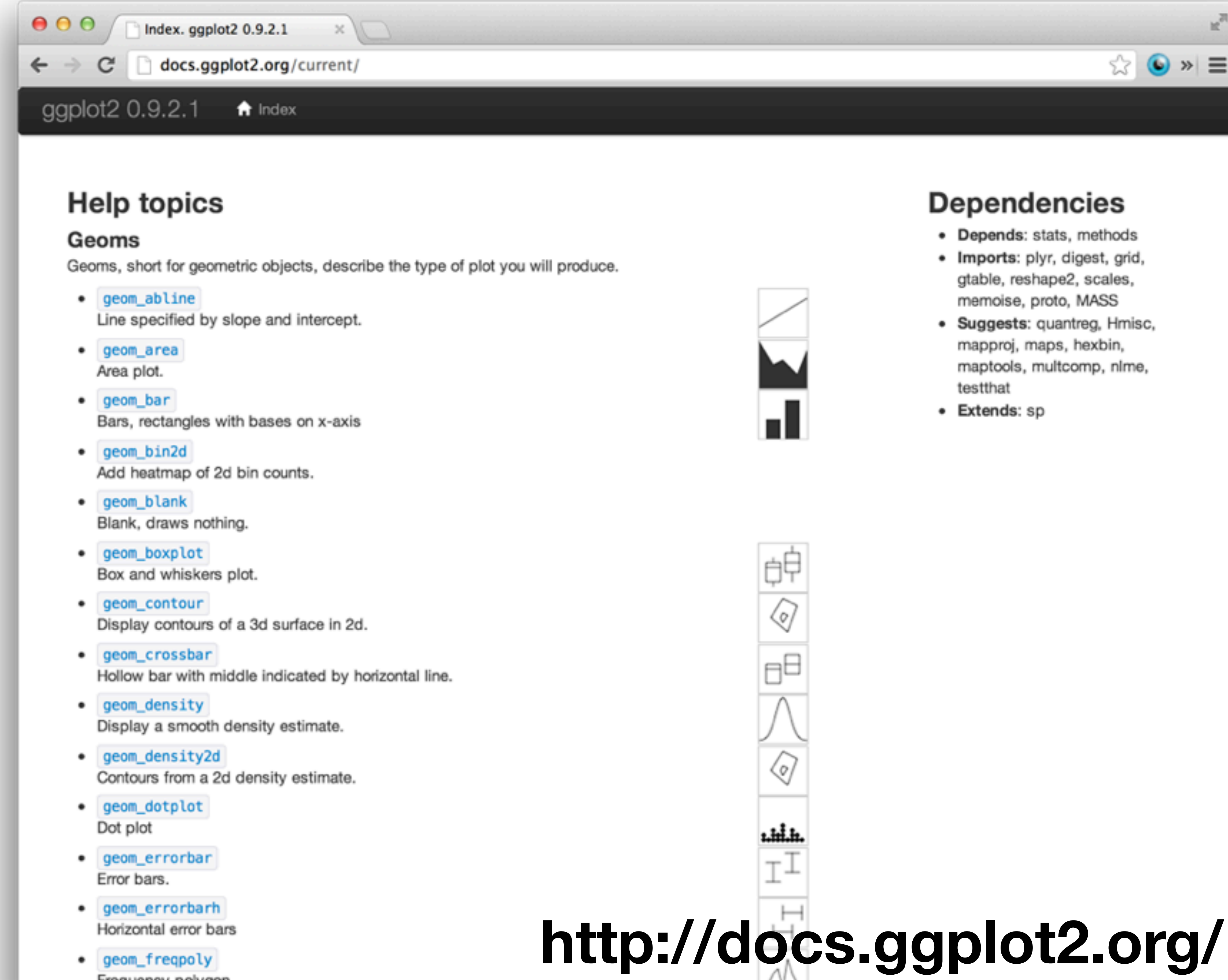
	stat	geom
histogram	bin	bar
smooth	smooth	line
boxplot	boxplot	boxplot
density	density	line
freqpoly	bin	line

Your turn

For each of the following plots created with qplot, recreate the equivalent ggplot code.

```
qplot(carat, price, data = diamonds)

qplot(hwy, cty, data = mpg, geom = "jitter")

qplot(reorder(class, hwy), hwy, data = mpg,
  geom = c("jitter", "boxplot"))

qplot(log10(carat), log10(price),
data = diamonds, colour = color) +
geom_smooth(method = "lm")
```

```
ggplot(diamonds, aes(carat, price)) +
  geom_point()

ggplot(mpg, aes(hwy, cty)) +
  geom_jitter()

ggplot(mpg, aes(reorder(class, hwy), hwy)) +
  geom_jitter() +
  geom_boxplot()

ggplot(diamonds, aes(log10(carat), log10(price),
  colour = color)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Index. ggplot2 0.9.2.1

docs.ggplot2.org/current/

ggplot2 0.9.2.1 Index

Help topics

Geoms

Geoms, short for geometric objects, describe the type of plot you will produce.

- `geom_abline`
  Line specified by slope and intercept.
- `geom_area`
  Area plot.
- `geom_bar`
  Bars, rectangles with bases on x-axis
- `geom_bin2d`
  Add heatmap of 2d bin counts.
- `geom_blank`
  Blank, draws nothing.
- `geom_boxplot`
  Box and whiskers plot.
- `geom_contour`
  Display contours of a 3d surface in 2d.
- `geom_crossbar`
  Hollow bar with middle indicated by horizontal line.
- `geom_density`
  Display a smooth density estimate.
- `geom_density2d`
  Contours from a 2d density estimate.
- `geom_dotplot`
  Dot plot
- `geom_errorbar`
  Error bars.
- `geom_errorbarh`
  Horizontal error bars
- `geom_freqpoly`
  Frequency polygon.

Dependencies

- **Depends**: stats, methods
- **Imports**: plyr, digest, grid, gtable, reshape2, scales, memoise, proto, MASS
- **Suggests**: quantreg, Hmisc, mapproj, maps, hexbin, maptools, multcomp, nlme, testthat
- **Extends**: sp

http://docs.ggplot2.org/

Learning ggplot2

ggplot2 mailing list
http://groups.google.com/group/ggplot2

stackoverflow
http://stackoverflow.com/tags/ggplot2

Cookbook for common graphics
http://wiki.stdout.org/rcookbook/Graphs/

ggplot2 book
http://www.springerlink.com/content/978-0-387-98140-6/contents/

Communication

Exploratory graphics

Are for **you** (not others). Need to be able to create rapidly because your first attempt will never be the most revealing.

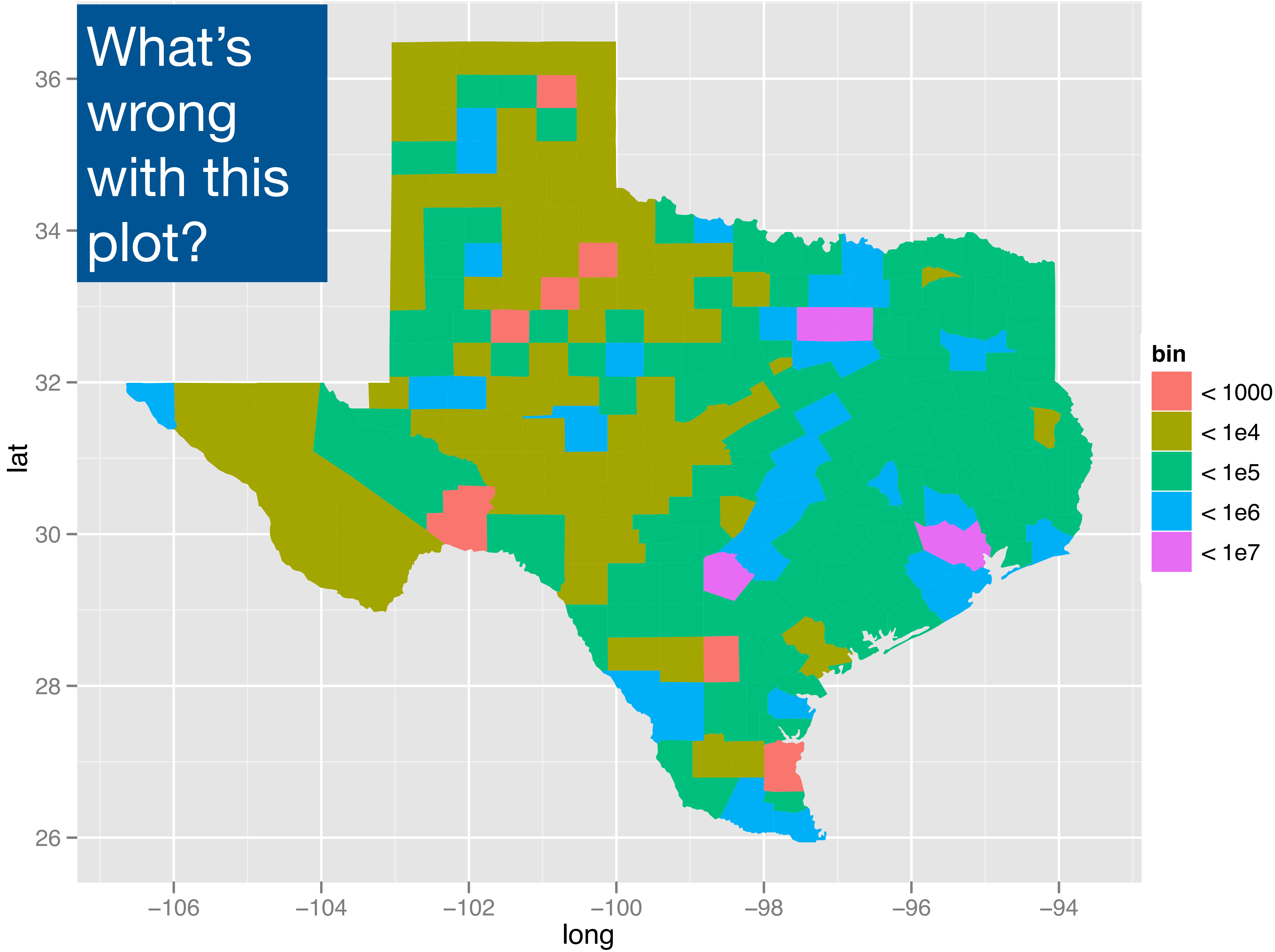
Iteration is crucial for developing the best display of your data.

Communication graphics

When you **communicate** your findings, you need to spend a lot of time polishing your graphics to eliminate distractions and focus on the story.

Iteration is crucial to ensure all the the small stuff works well: labels, color choices, tick marks...

What's wrong with this plot?



Some problems

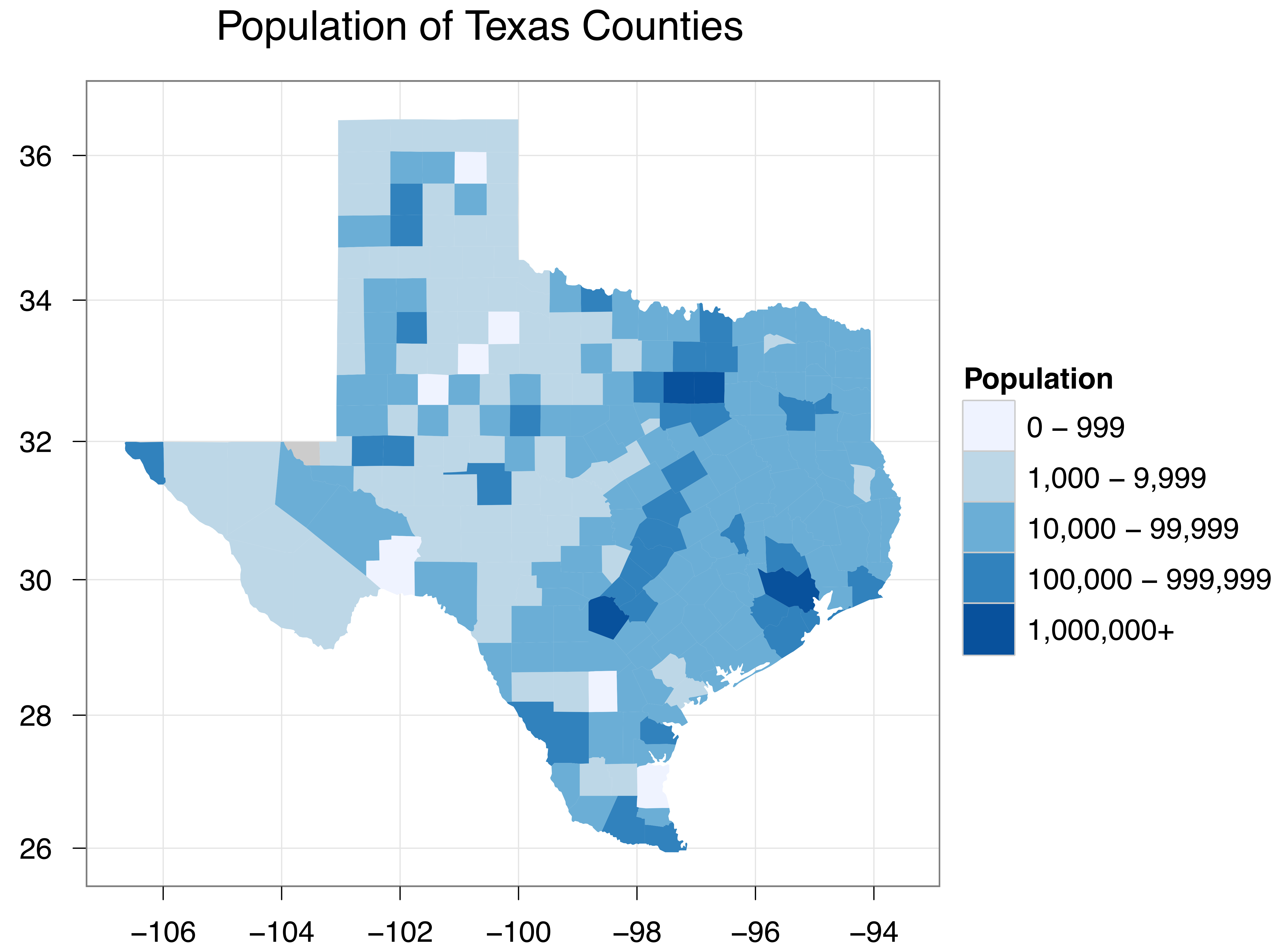
Bad colour scheme

Unnecessary axis labels

Legend needs improvement: better title and better key labels

No title

Incorrect aspect ratio


Population of Texas Counties
36
34
32
30
28
26
−106
−104
−102
−100
−98
−96
−94
Population
0 – 999
1,000 – 9,999
10,000 – 99,999
100,000 – 999,999
1,000,000+

Scales

Scales

Control how data is mapped to perceptual properties, and produce **guides** (axes and legends) which allow us to read the plot.

Important parameters: **name**, **breaks** & **labels**, **limits**.

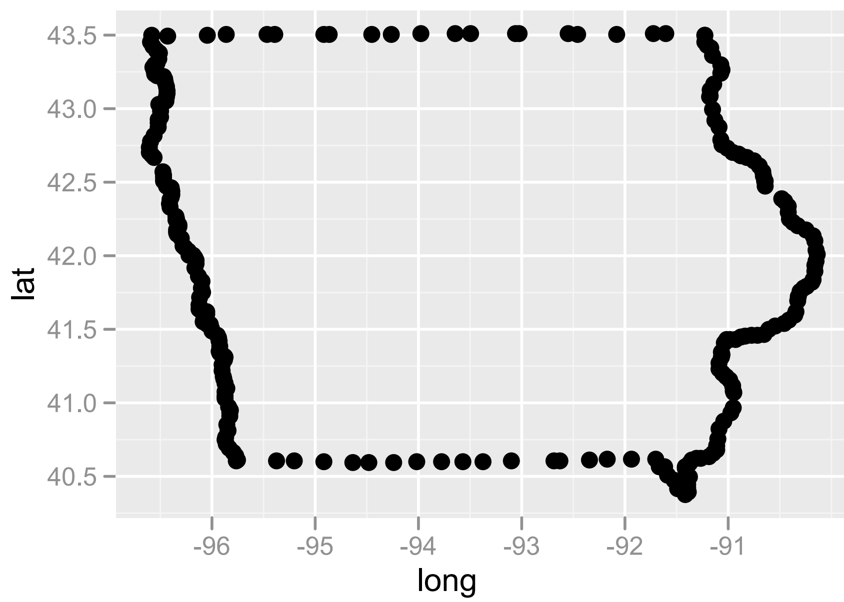
Naming scheme: scale_*aesthetic*_*name*. All default scales have name continuous or discrete.

```
# Default scales
scale_x_continuous()
scale_y_discrete()
scale_colour_discrete()

# Custom scales
scale_colour_hue()
scale_x_log10()
scale_fill_brewer()

# Scales with parameters
scale_x_continuous("X Label", limits = c(15, 30))
scale_colour_gradient(low = "blue", high = "red")
```

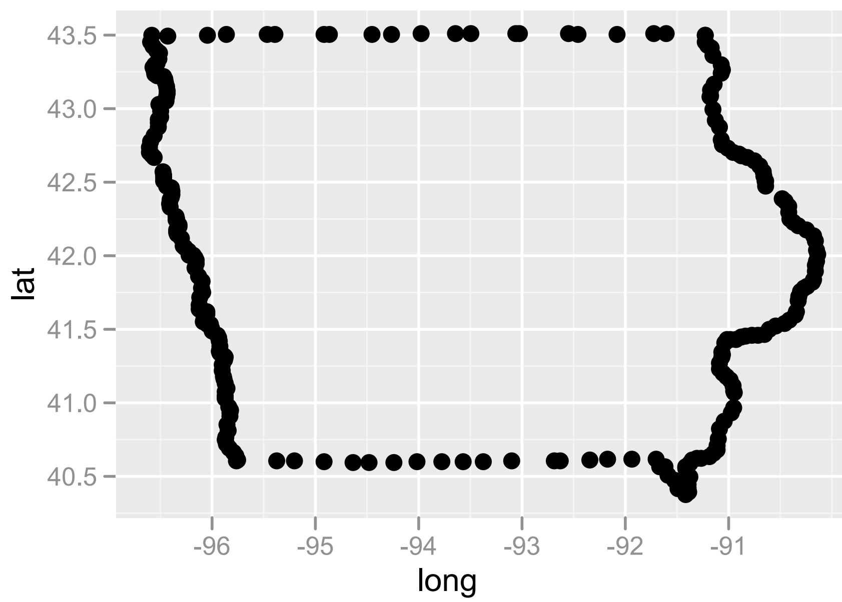
```
# First argument (name) controls axis label
scale_y_continuous("Latitude")
scale_x_continuous("")

# Breaks and labels control tick marks
scale_x_continuous(breaks = -c(106,100,94))
scale_fill_discrete("Population", labels =
  c("< 1000" = "0 - 999" , "< 1e4" = "1,000 - 9,999",
    "< 1e5" = "10,000 - 99,999",  "< 1e6" = "100,000 -
    999,999", "< 1e7" = "1,000,000+"))
scale_y_continuous(breaks = NA)

# Limits control range of data
scale_y_continuous(limits = c(26, 32))
# same as:
p + ylim(26, 32)
```

What is a map?

What is a map?

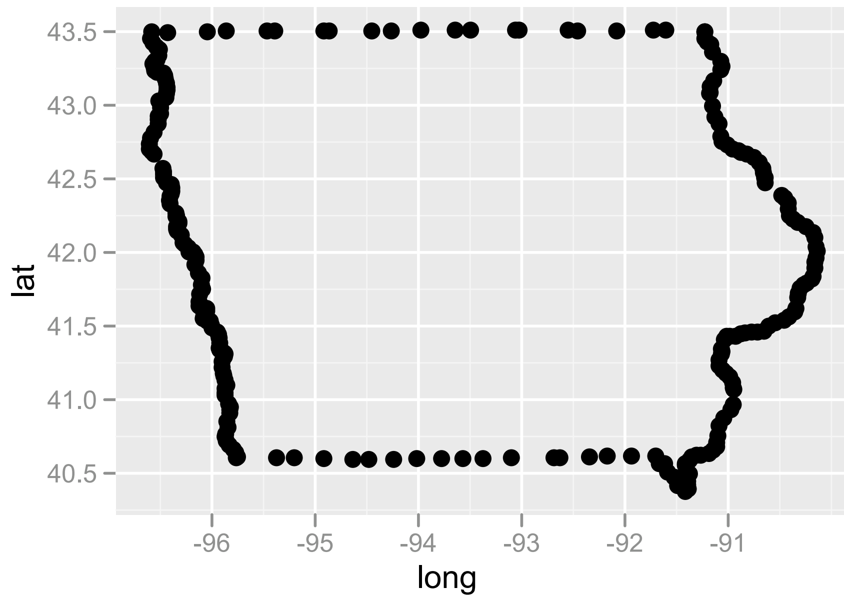


What is a map?

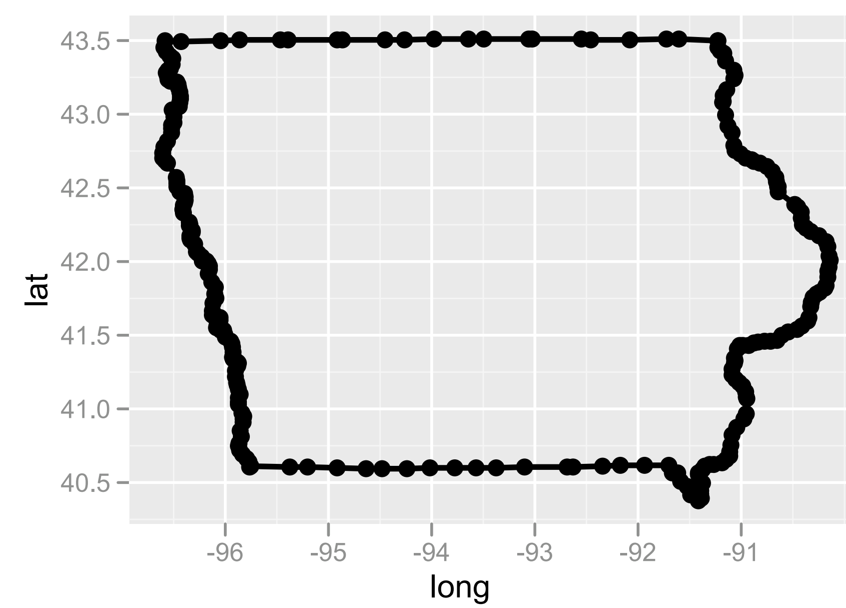


Set of **points** specifying latitude and longitude

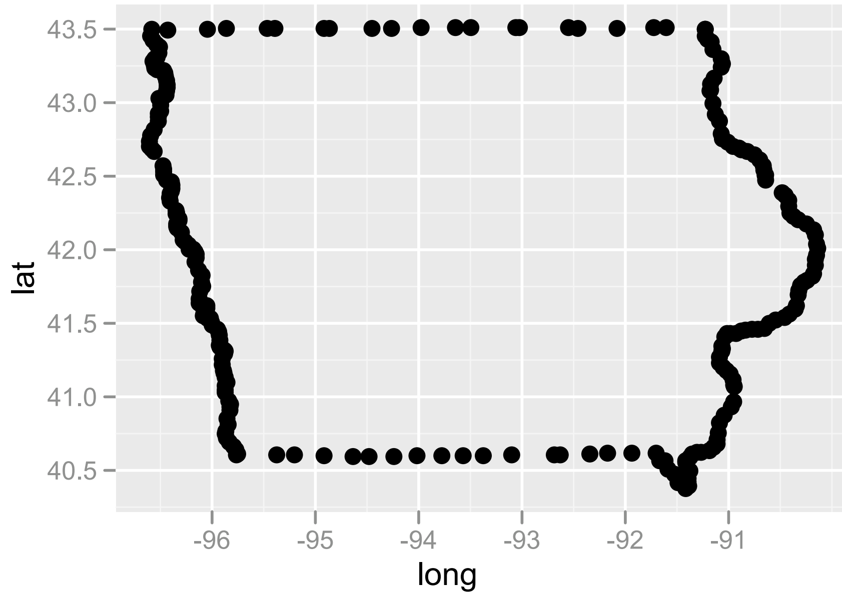
What is a map?



Set of **points** specifying latitude and longitude

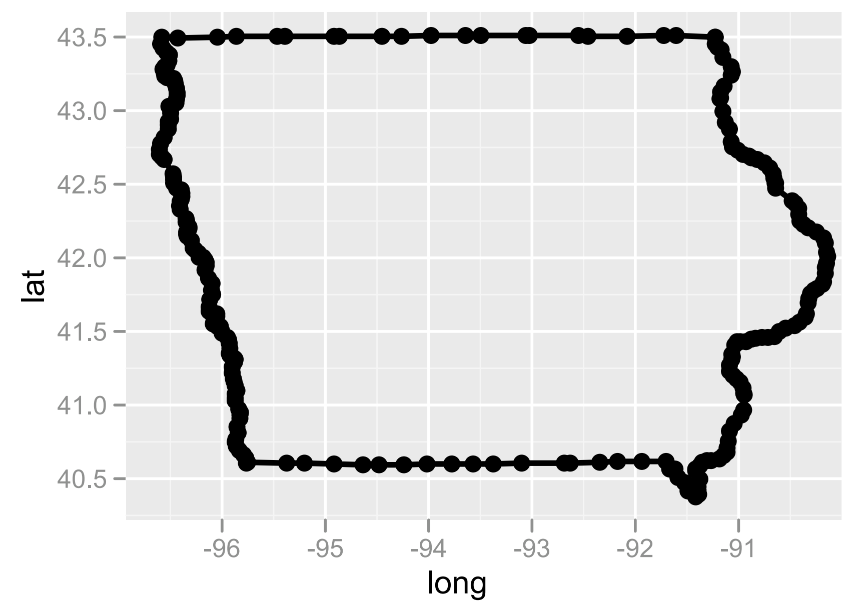


What is a map?



Set of **points** specifying latitude and longitude

Polygon: connect dots in correct order

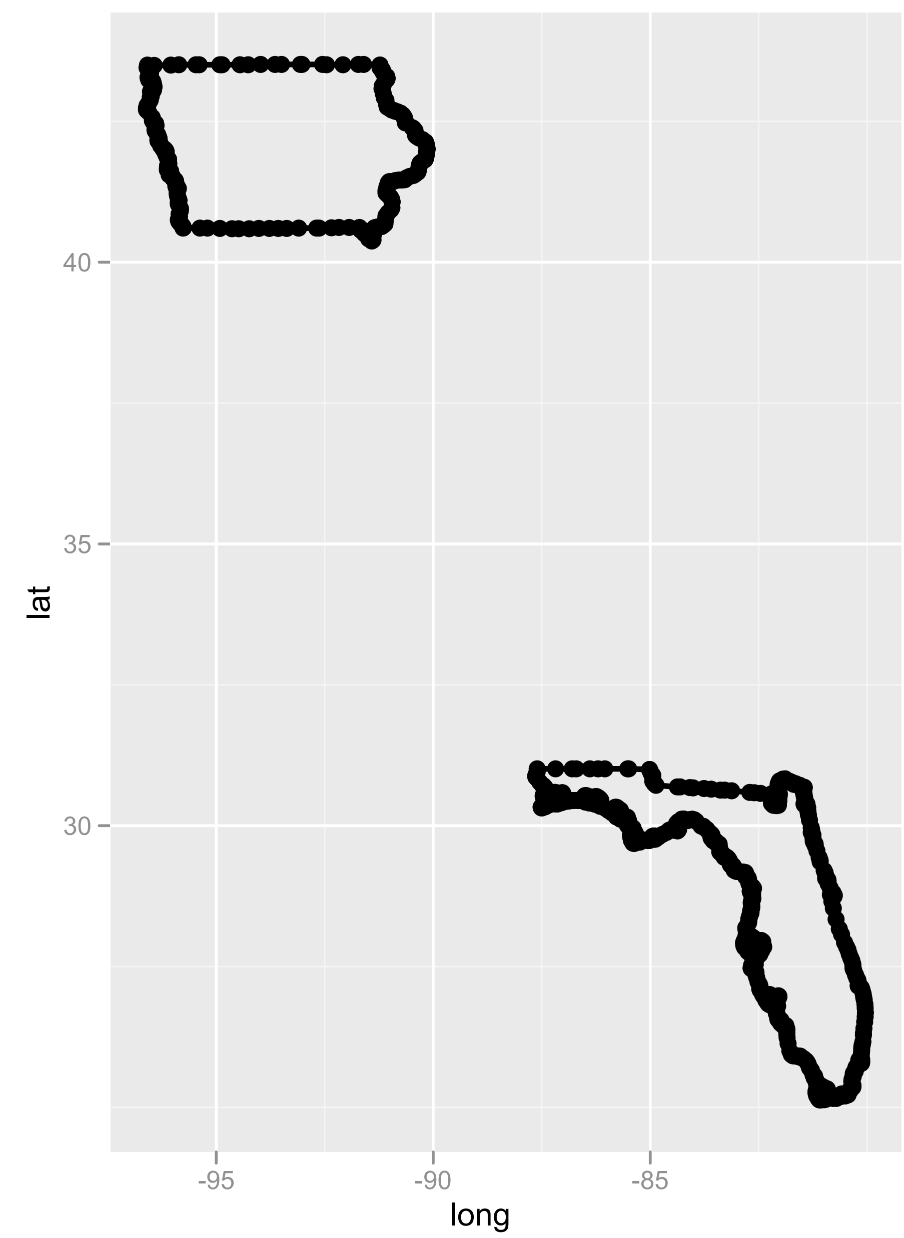


What is a map?

Polygon: connect only the correct dots

What is a map?

Polygon: connect only the correct dots



What is a map?

Polygon: connect only the correct dots

This is **group**ing (again)

```
# To draw the graph
options(stringsAsFactors = FALSE)
pop <- read.csv("tx-pop.csv")
pop$bin <- cut(log10(pop$pop), breaks = 2:7,
  labels = c("< 1000", "< 1e4", "< 1e5",
  "< 1e6", "< 1e7"))

borders <- read.csv("tx-borders.csv")
choro <- join(borders, pop)

qplot(long, lat, data = choro,
  geom = "polygon", group = group,
  fill = bin) + coord_map()
```

Your turn

Fix the axis and legend related problems we identified. You'll need to add multiple scales on to the original plot.

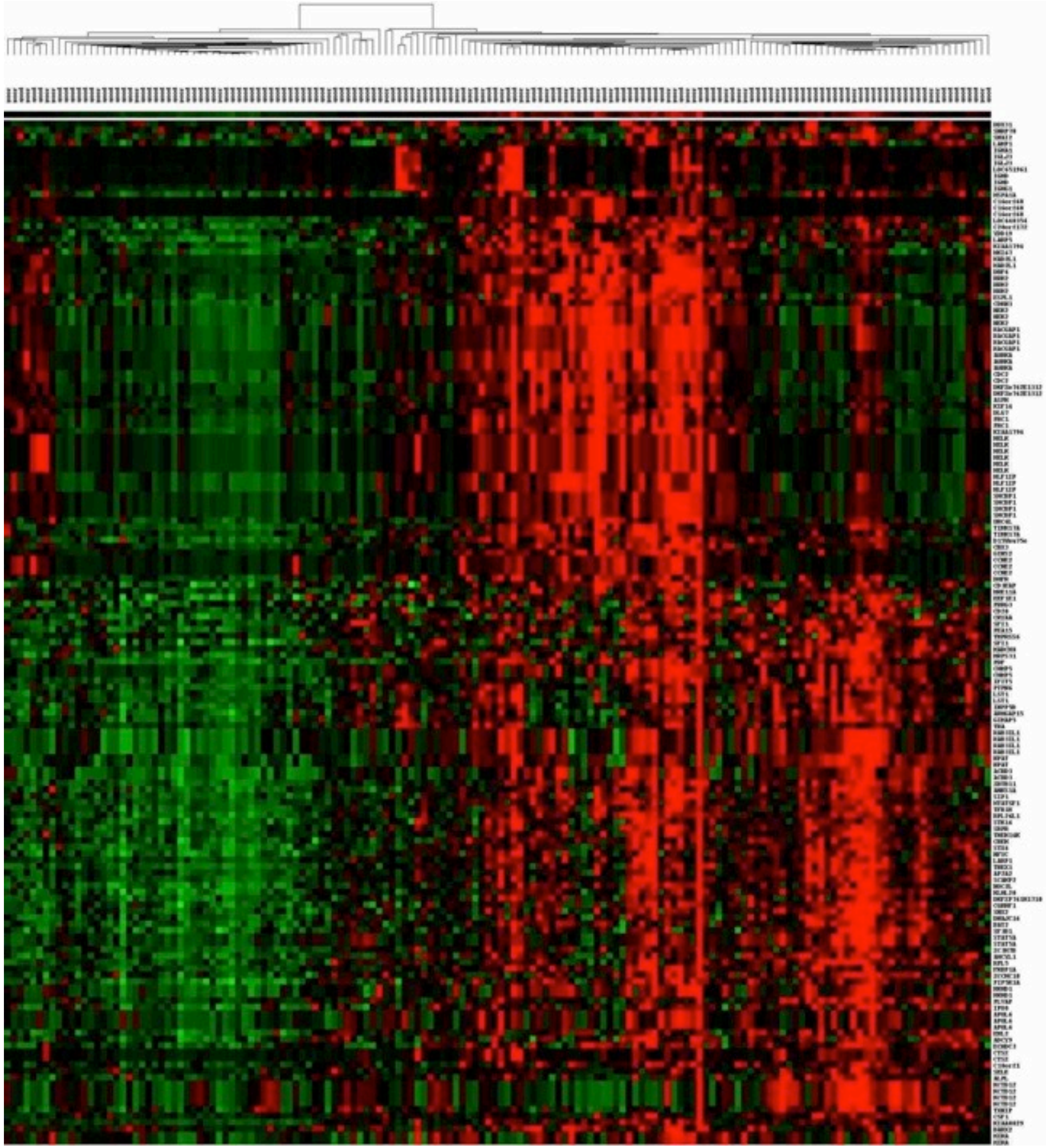
```r
qplot(long, lat, data = choro, geom = "polygon", group = group, fill = bin) +
  scale_fill_discrete("Population", labels =
    c("< 1000" = "0 - 999" , "< 1e4" = "1,000 - 9,999", "< 1e5" =
       "10,000 - 99,999",  "< 1e6" = "100,000 - 999,999", "< 1e7" =
       "1,000,000+")) +
  scale_x_continuous("") +
  scale_y_continuous("") +
  coord_map()
```

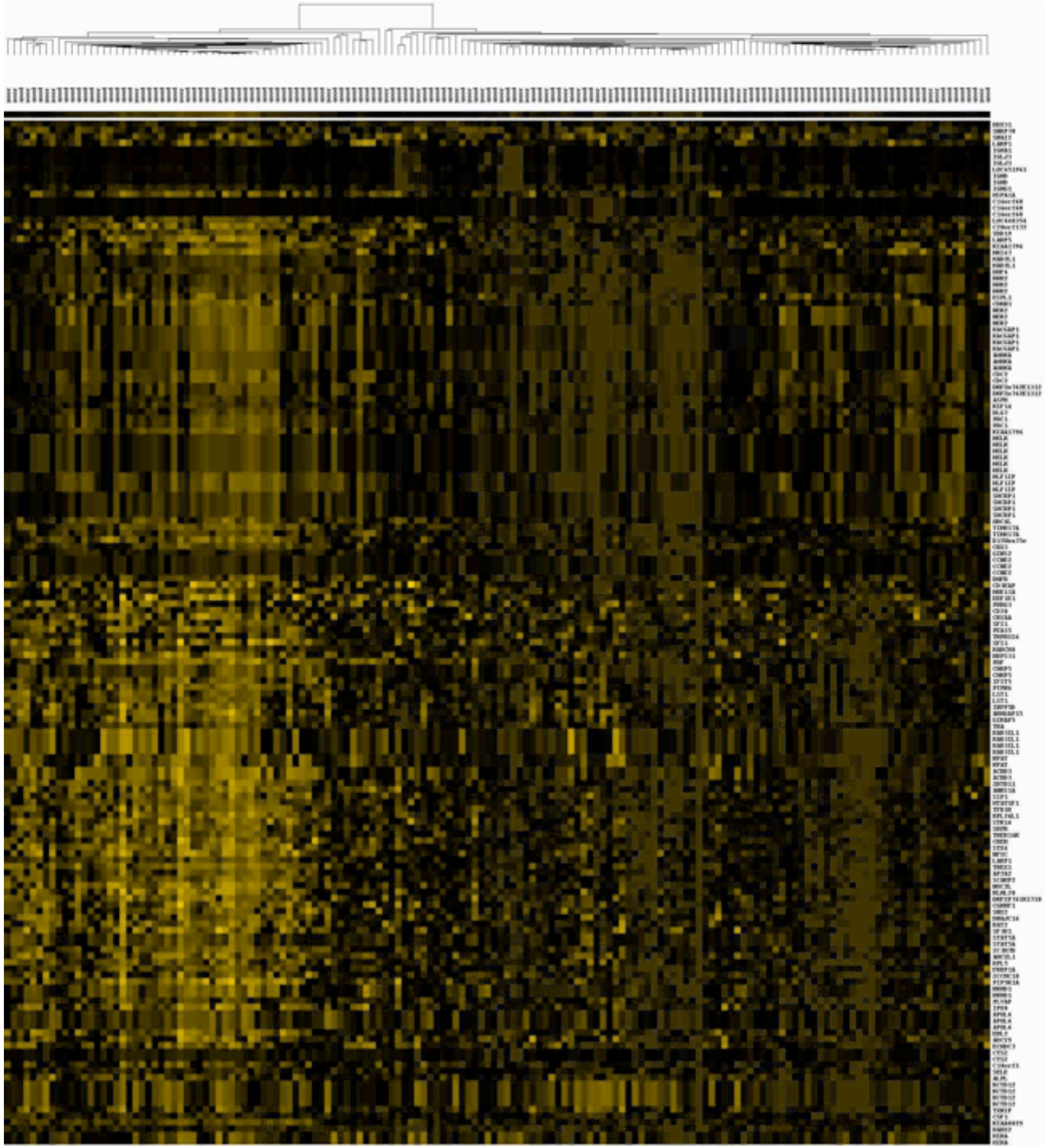
Colour

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





Alternatives

Discrete: brewer, grey, manual

Continuous: gradient2, gradientn

Your turn

Modify the fill scale to use a Brewer colour palette of your choice. (Hint: you will need to change the name of the scale)

Use `RColorBrewer::display.brewer.all` to list all palettes.

```
ggplot(choro, aes(long, lat)) +
  geom_polygon(aes(group = group, fill = bin)) +
  scale_fill_brewer("Population", labels = c("< 1000" = "0 - 999" , "< 1e4" =
    "1,000 - 9,999", "< 1e5" = "10,000 - 99,999", "< 1e6" = "100,000 -
    999,999", "< 1e7" = "1,000,000+"), palette = "Blues") +
  scale_x_continuous("") +
  scale_y_continuous("") +
  coord_map()
```

Themes

```
# Lots to learn, but the most important things
# are:

qplot(mpg, wt, data = mtcars) + theme_bw()

qplot(mpg, wt, data = mtcars) +
  theme(title = "My awesome title")
```

```
ggplot(choro, aes(long, lat)) +
  geom_polygon(aes(group = group, fill = bin)) +
  scale_fill_brewer("Population", labels = c("< 1000" = "0 - 999" , "< 1e4" =
    "1,000 - 9,999", "< 1e5" = "10,000 - 99,999", "< 1e6" = "100,000 -
    999,999", "< 1e7" = "1,000,000+"), palette = "Blues") +
  scale_x_continuous("") +
  scale_y_continuous("") +
  coord_map() +
  theme_bw() +
  theme(title = "Population of Texas Counties")
```