Grammar of Graphics

Brian Perron, Ph.D.

November 12, 2015

CHILD & ADOLESCENT DATA LAB

Key points

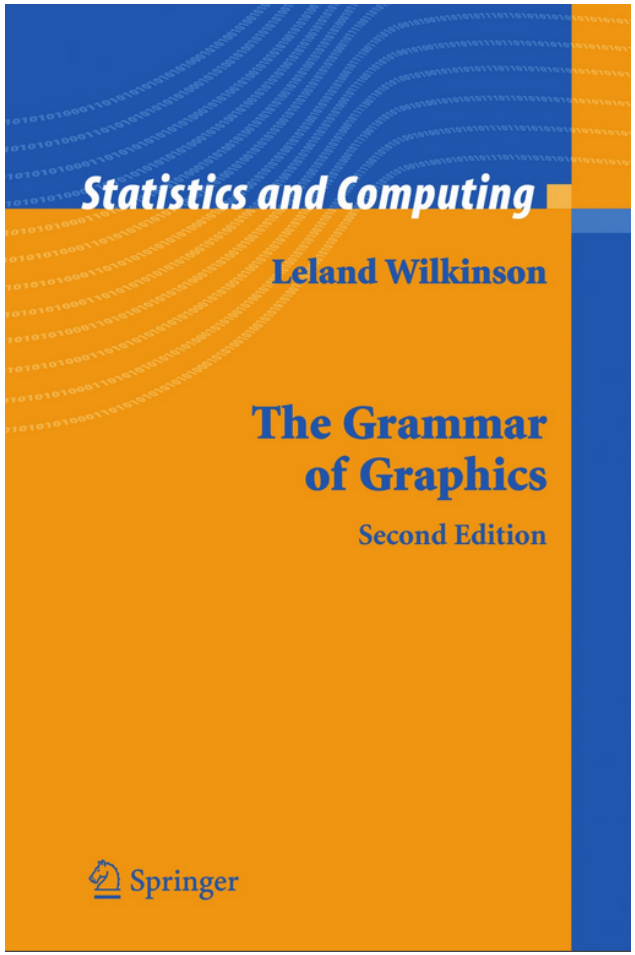
1. Conceptual overview of the grammar of graphics
2. Essential elements of the graphical language
3. Grammar applied to the workflow
4. Relate graphical language to code

Conceptual overview of the grammar of graphics

The original grammar

Leland Wilkinson

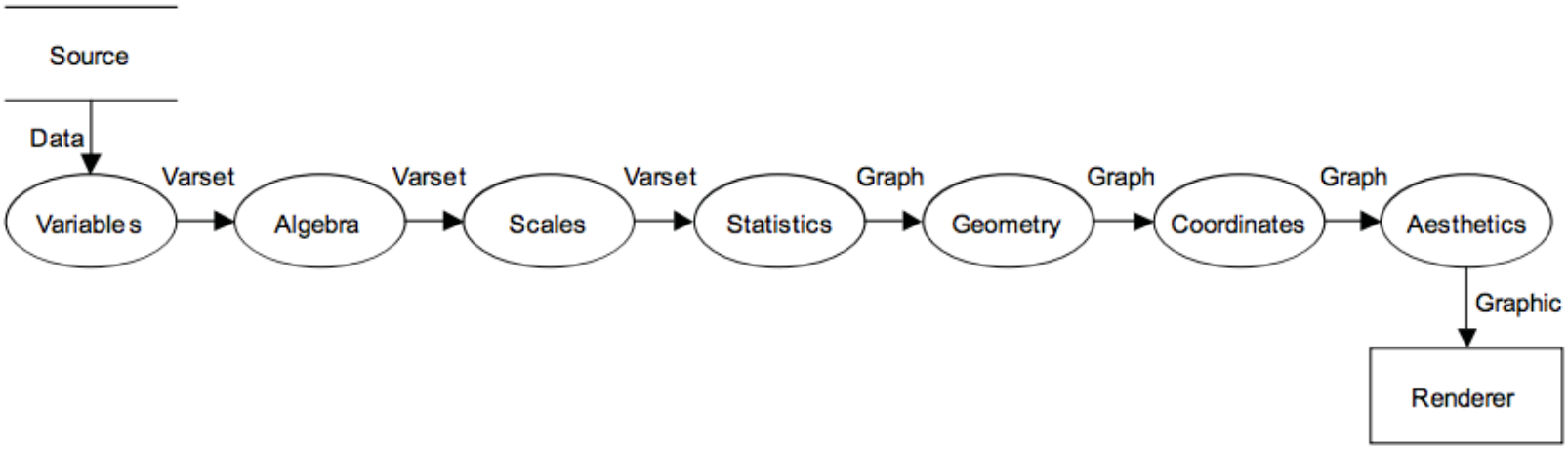
- Bertin's Semiology of Graphics (1967)
- Graphics Production Library (GPL)
- SYSTAT, SPSS, Tableau



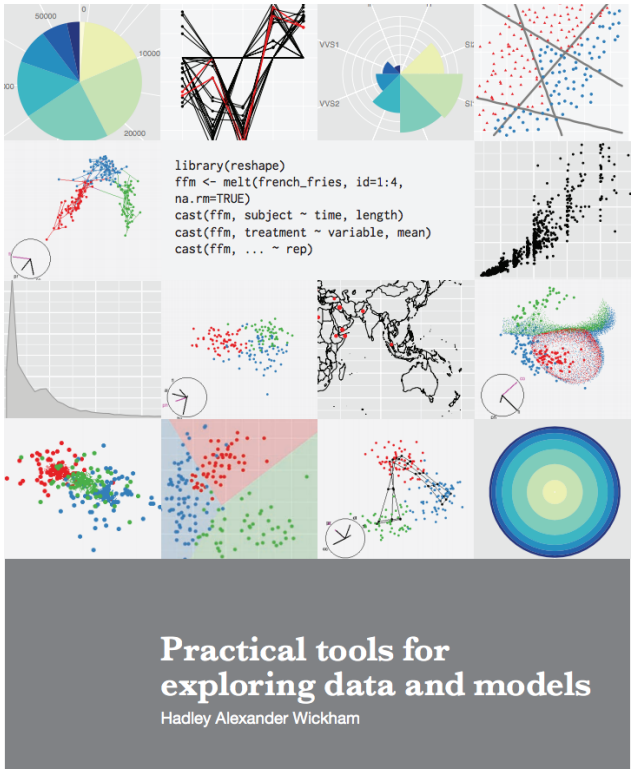
Motivations for a grammar

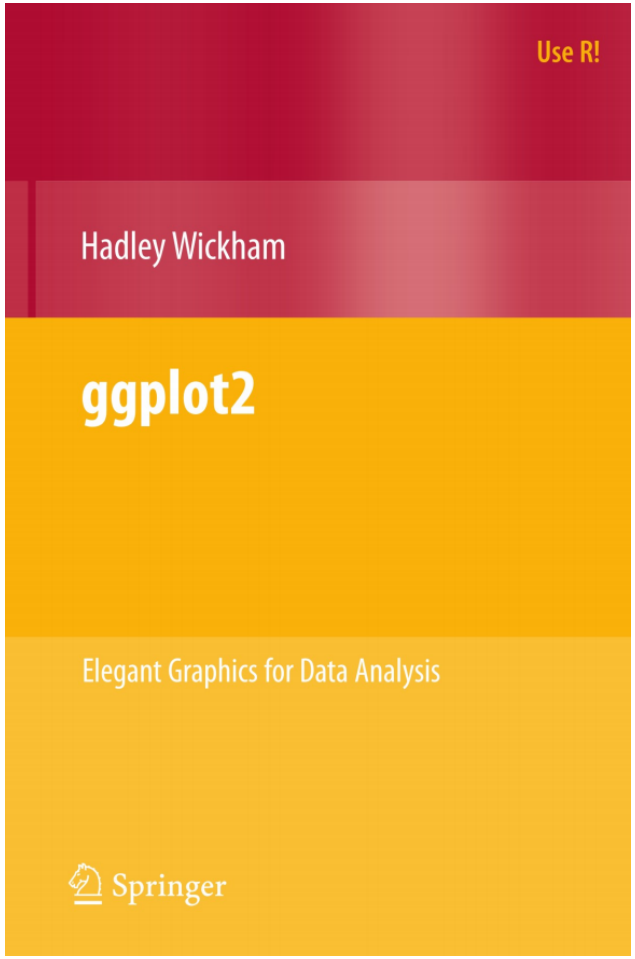
- To describe deep features that underlie all graphics
- Provides language-based rules
- Beyond chart typologies to unlimited graphical forms
- Emphasis on effective display of data

The graphical pipeline



The grammar revisioned





The grammar revisioned

What is maintained?

- Original graphical language
- Conceptual framework of graphics
- Motivations for describing deep structure

What is different?

- Primacy on layered development
- Graphical pipeline

Plotting functions

qplot

- Quick plot
- Similar to `plot` in base R
- Rapid exploration of data
- Limited use of the grammar

ggplot

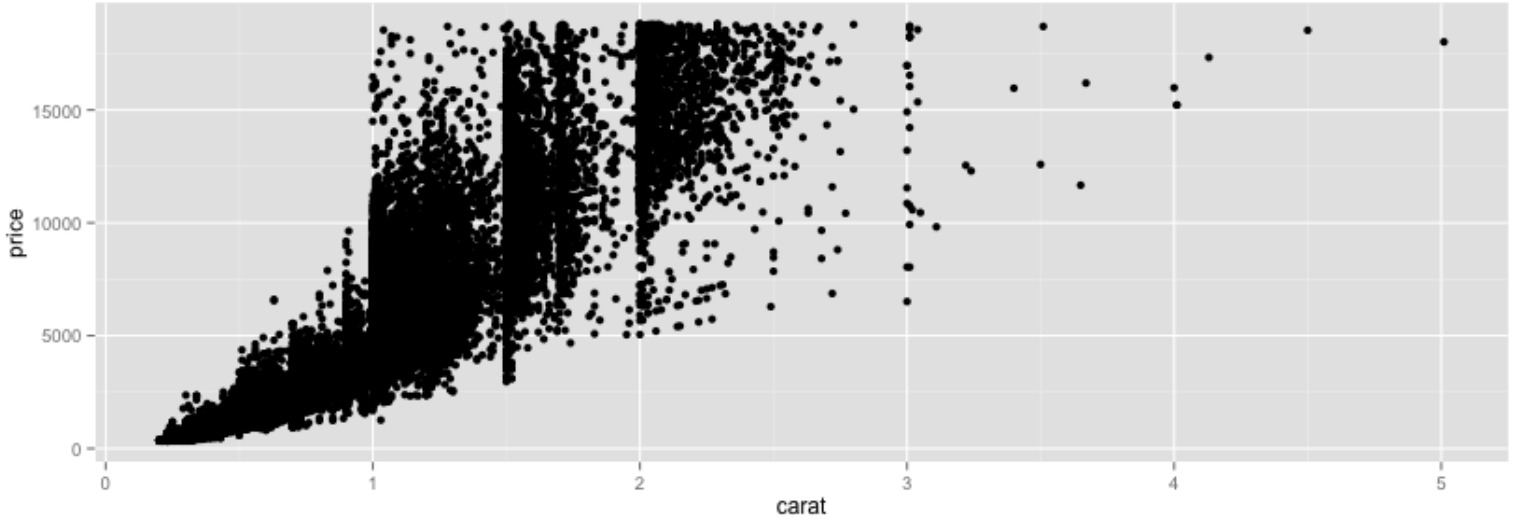
- Infinite number of graphical options
- Full control over the grammar
- Extensible
- Limited scalability

qplot vs. ggplot

Minimal example

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

p <- ggplot(data=diamonds, aes(carat, price)) + geom_point()
```



Essential elements of the graphical language

Graphical language

Parts of speech

1. Data (variables and algebra)
2. Transformations (linear, log)
3. Geometry
4. Scales (aesthetics)
5. Statistics (summarized vs. unsummarized data)
6. Coordinate system (cartesian, polar, `facet`)
7. Guides (axes, legends, annotations, etc.)

Geometry

Graphical language

geom

- Things that you see
- Lines, points, bars, polygons, area, etc.
- Defines the range of aesthetic attributes
- Constitutes a single layer
- Each unique layer contains a data set
- Painter's model

Aesthetics

Graphical language

aes

- Mapping of aesthetic properties to a layer
- Position within coordinate system
- Attributes of geometry
- Color, shape, size, group

Data

Graphical language

data

- Derive variables from the data
- Each aesthetic must have a variable mapping
- Each geometry has a corresponding dataframe
- Data must be an R dataframe
- Variable(s) for faceting
- Summarized versus unsummarized data (`stat`)

I ❤ melted data . . .

Grammar applied to the workflow

The old way of working

1. Have idea
2. Start hacking
3. Google search
4. Swear, get coffee, and return to hacking
5. Search StackOverflow
6. Everything but one thing working
7. Swear, get coffee, and return to hacking
8. Swear
9. Post on StackOverflow
10. Post closed, redirected to unhelpful post
11. Hacking finally works but not sure why
12. Mediocre graphic . . . repeat workflow

Where to begin?

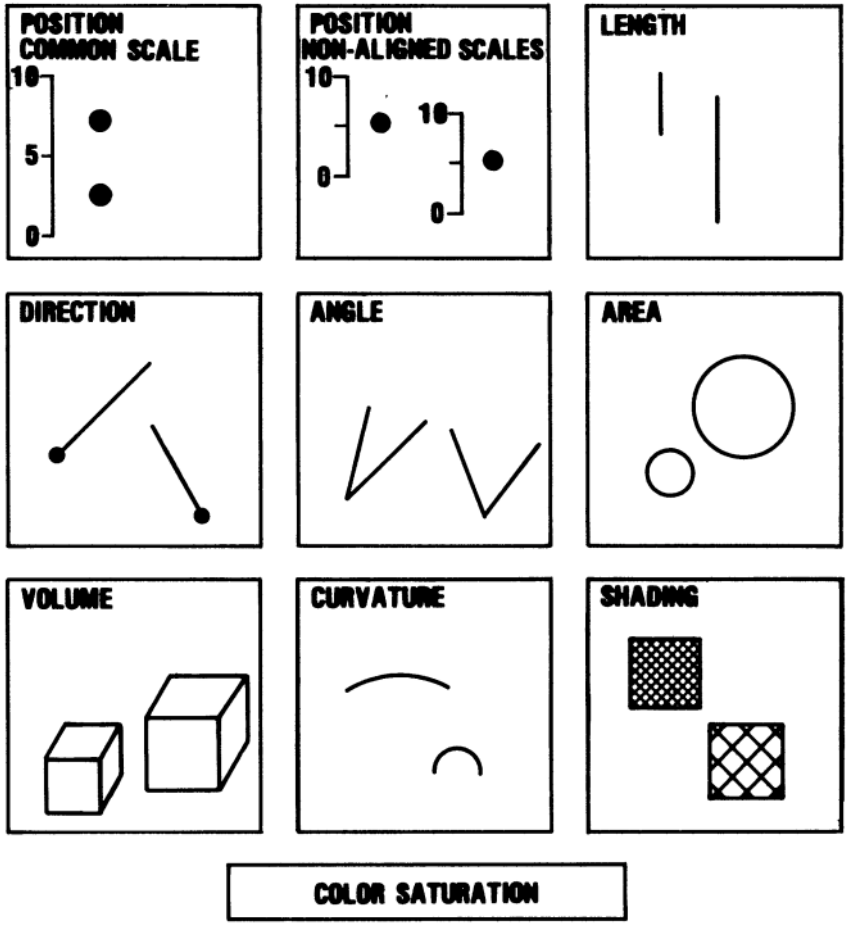
Example workflow

1. Establish the storyline
2. Determine best way to visually encode
3. Design the graphic (outside of R)
4. Describe the graphic with the grammar
5. Shape the data
6. Execute in `ggplot`
7. Tufte-fy

Visual encoding & decoding principles

Theory of Graphical Perception - Cleveland & McGill

Journal of the American Statistical Association, September 1984



Visual encoding & decoding principles

Pre-attentive processing

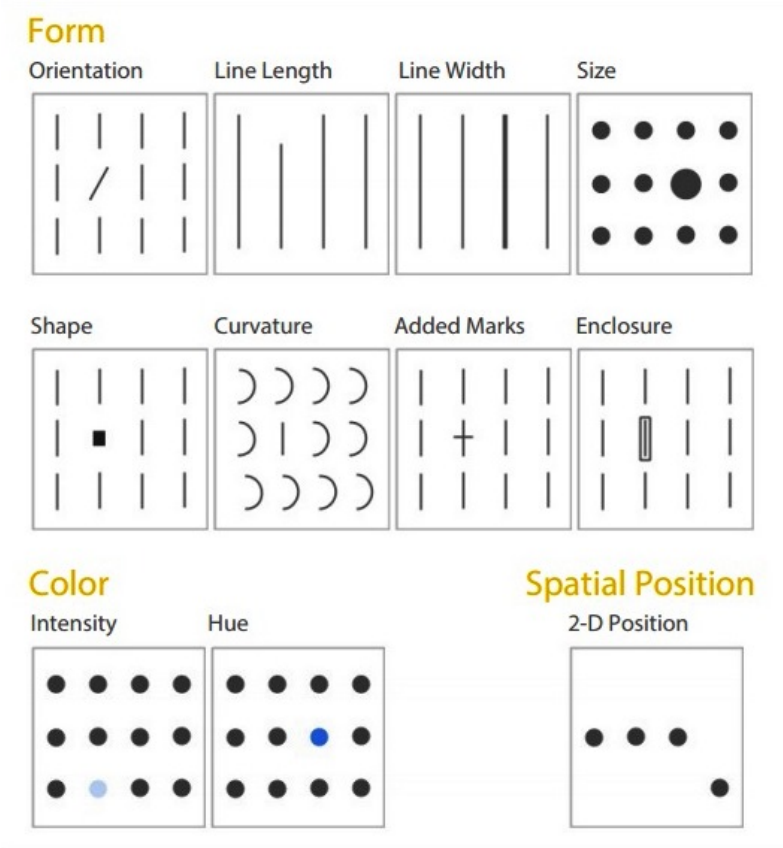


Image credit: Colin Ware, Information Visualization: Perception for Design

Visual encoding & decoding principles

Analytic patterns from attributes

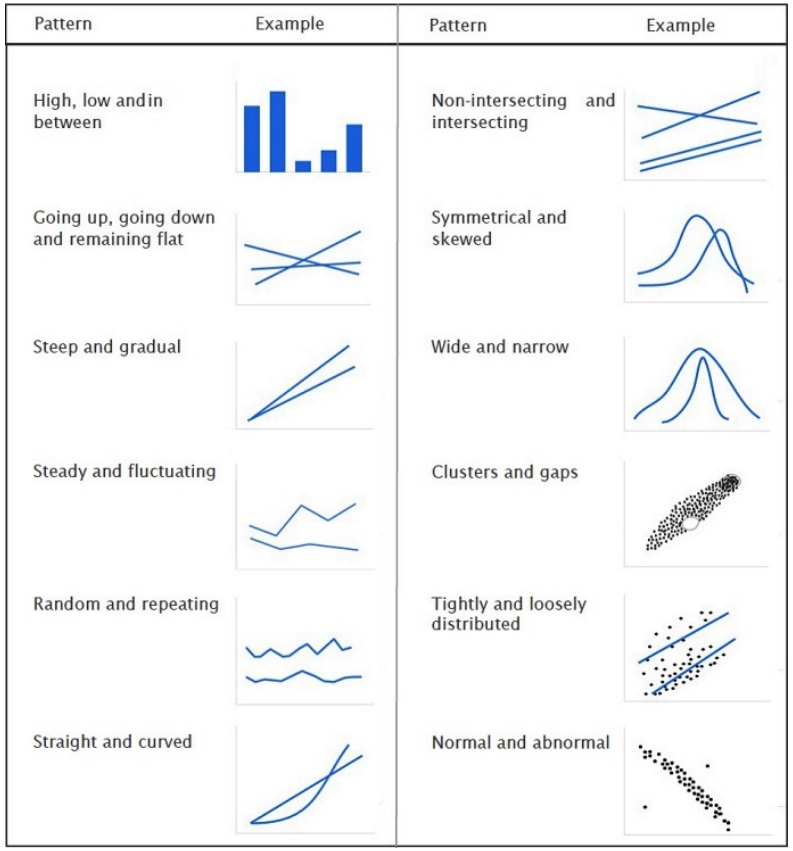
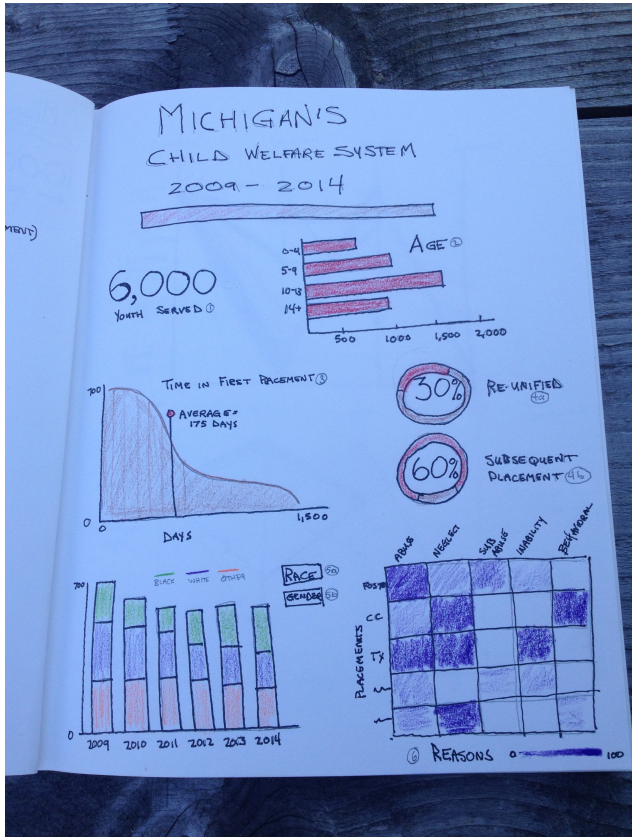
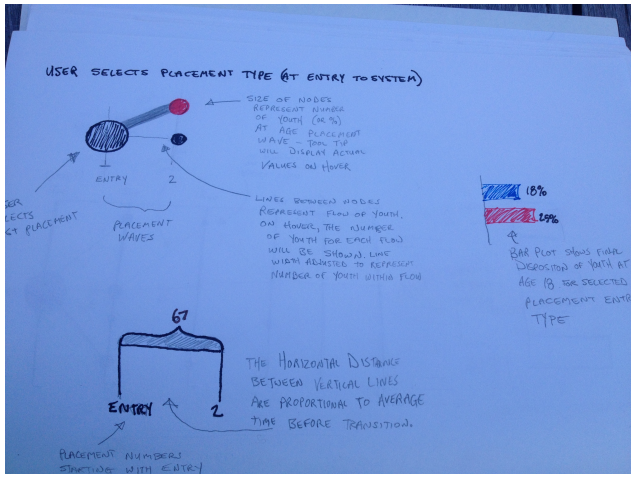


Image credit: Stephen Few, Now you see it: Simple visualization techniques for quantitative analysis

Design your graphic

Analog approach





Describe the graphic with the grammar

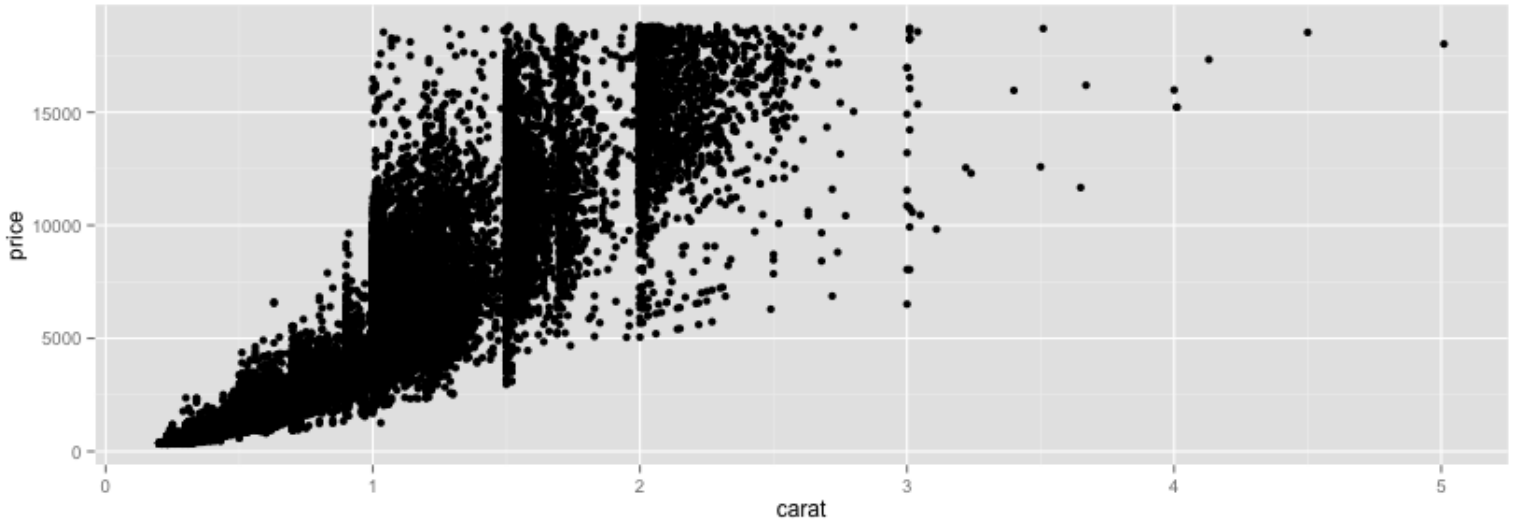
Applying the graphical language

- Data
- Transformations (outside of ggplot if possible)
- Geometry
- Scales & aesthetics
- Statistics (summarized vs. unsummarized data)
- Coordinate system (facet)
- Guides (Final touches)

Relate graphical language to code

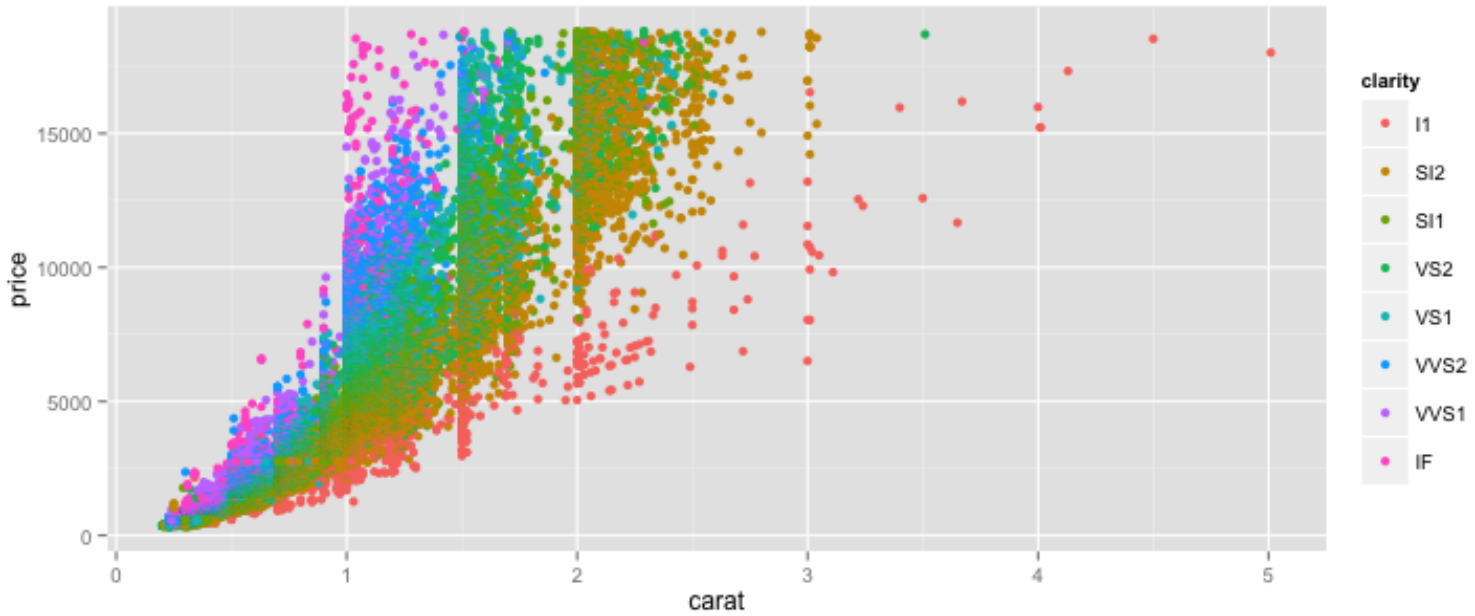
Remember that ugly scatterplot?

```
p <- ggplot(data=diamonds, aes(carat, price)) + geom_point()
```



Remember that ugly scatterplot?

```
p <- ggplot(data=diamonds, aes(carat, price, colour = clarity)) +
geom_point()
```

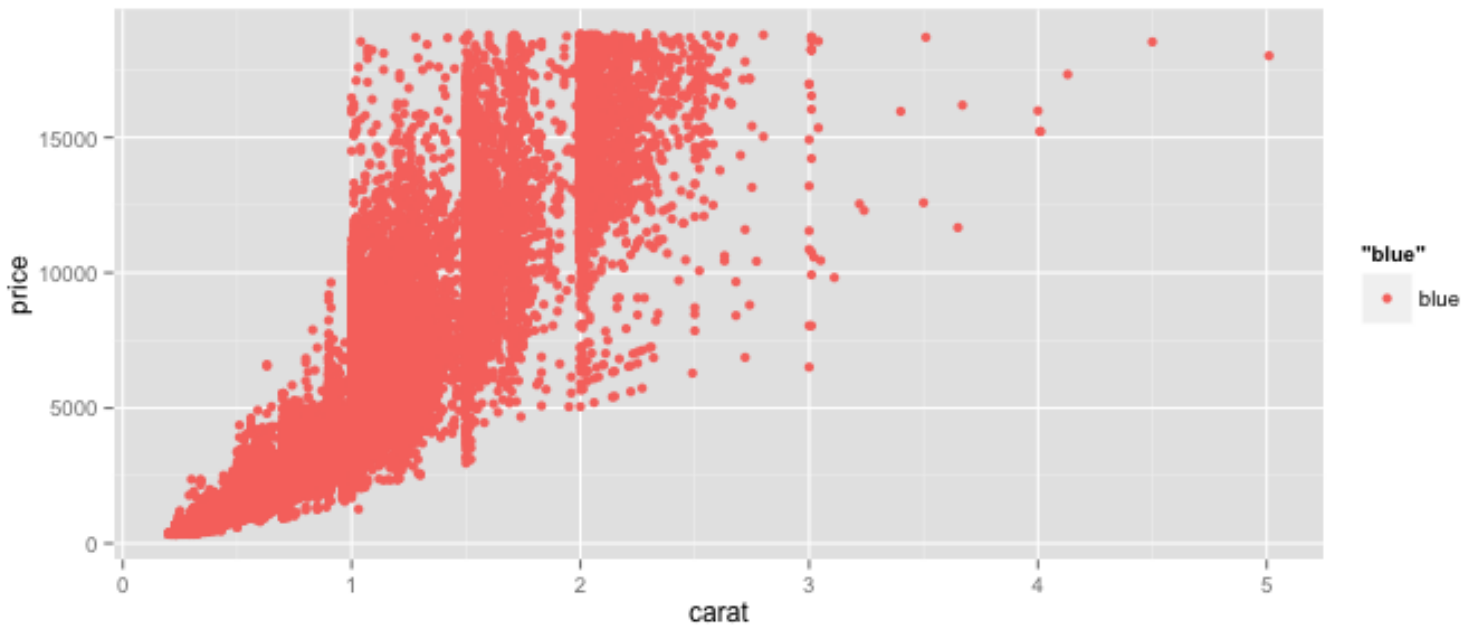


Under the hood

```
> p <- ggplot(diamonds, aes(carat, price, colour = clarity)) + geom_point()
> p
> summary(p)
data: carat, cut, color, clarity, depth, table, price, x, y, z [53940x10]
mapping:  x = carat, y = price, colour = clarity
faceting: facet_null()
-----------------------------------
geom_point: na.rm = FALSE
stat_identity:
position_identity: (width = NULL, height = NULL)
```

How about blue points?

```
p <- ggplot(data=diamonds, aes(carat, price, colour = "blue")) +
geom_point()
```



WTF?

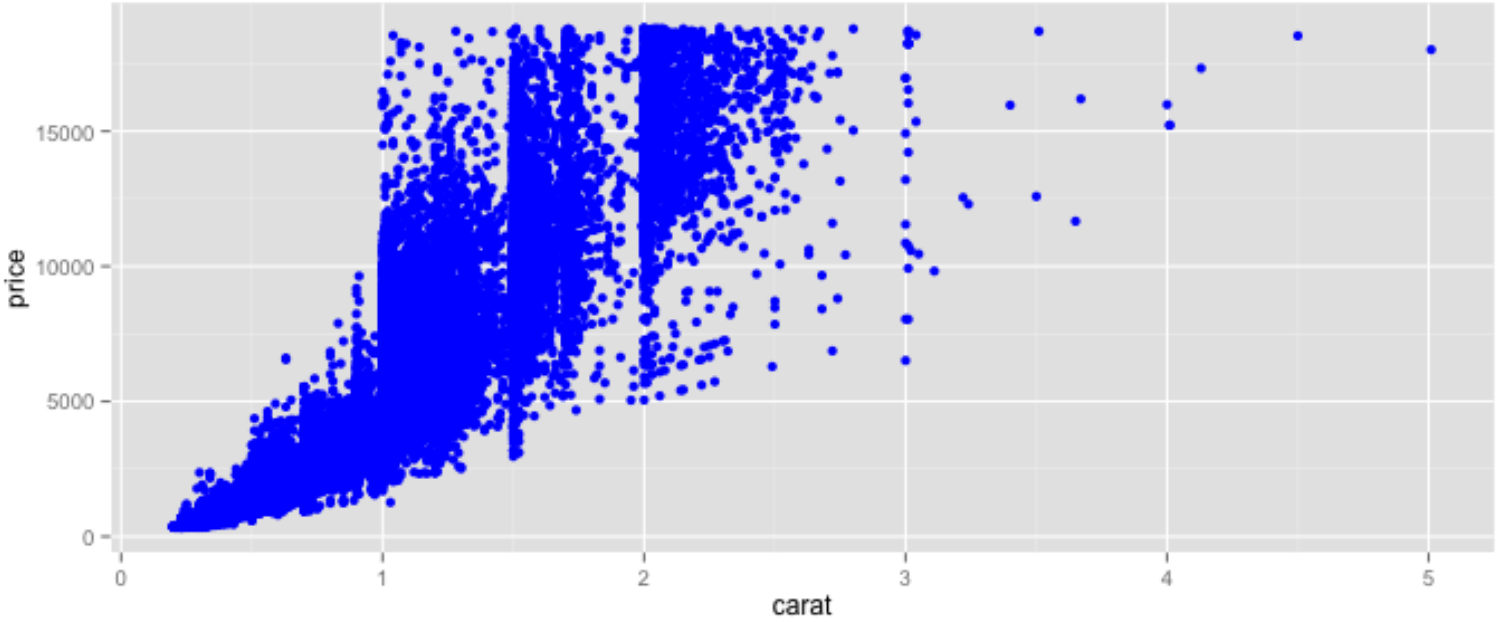
Under the hood

```
data: carat, cut, color, clarity, depth, table, price,
  x, y, z [53940x10]
mapping:  x = carat, y = price, colour = blue
faceting: facet_null()
-----------------------------------
geom_point: na.rm = FALSE
stat_identity:
position_identity: (width = NULL, height = NULL)
```

How about blue points?

Ahhhhh . . .

```
p <- ggplot(data=diamonds, aes(carat, price)) + geom_point(color
= "blue")
```

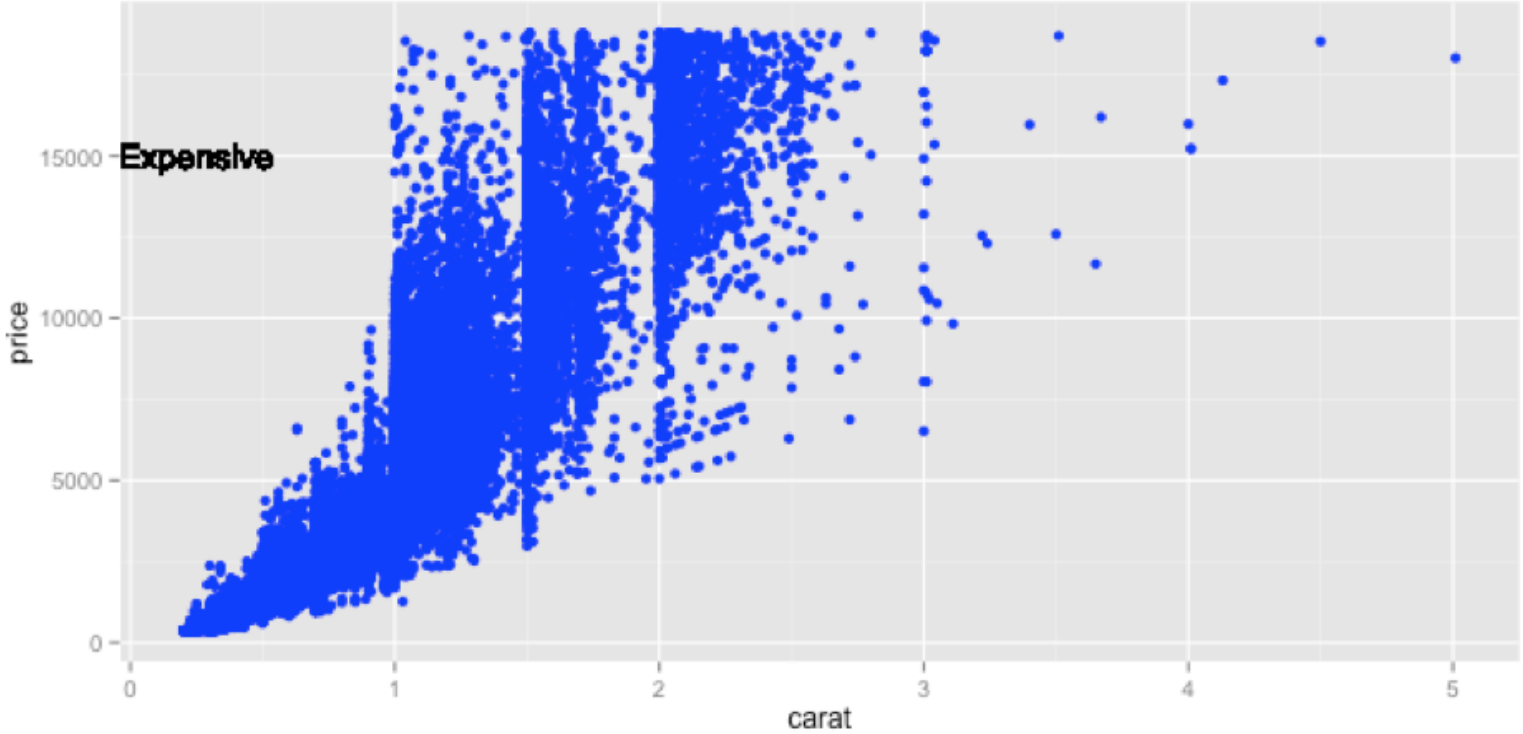


Under the hood

```
data: carat, cut, color, clarity, depth, table, price,
  x, y, z [53940x10]
mapping:  x = carat, y = price
faceting: facet_null()
-----------------------------------
geom_point: na.rm = FALSE, colour = blue
stat_identity:
position_identity: (width = NULL, height = NULL)
```

How about a text annotation?

```
p <- ggplot(data=diamonds, aes(carat, price)) + geom_point(color
= "blue") + geom_text(data = diamonds, aes(.75, 15000), label =
"Expensive")
```

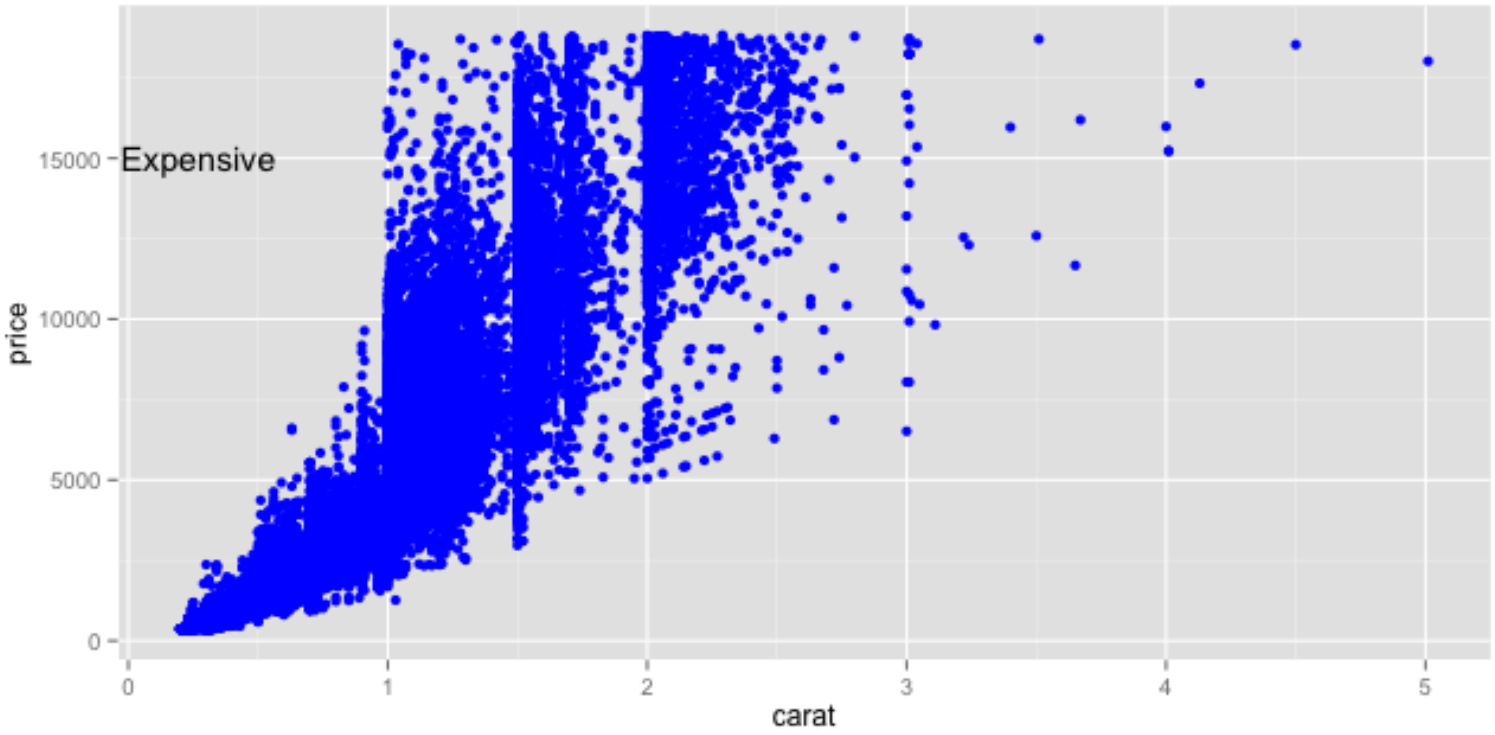


WTF?

How about a text annotation?

Ahhhhh . . .

```
p <- ggplot(data=diamonds, aes(carat, price)) + geom_point(color
= "blue") + annotate("text", x = .27, y = 15000,label =
"Expensive")
```



Under the hood

```
> summary(p)
data: carat, cut, color, clarity, depth, table, price, x,
  y, z [53940x10]
mapping:  x = carat, y = price
faceting: facet_null()
-----------------------------------
geom_point: na.rm = FALSE, colour = blue
stat_identity:
position_identity: (width = NULL, height = NULL)

mapping: x = x, y = y
geom_text: label = Expensive
stat_identity:
position_identity: (width = NULL, height = NULL)
```

A few more examples . . .

Worked example

Overview of data

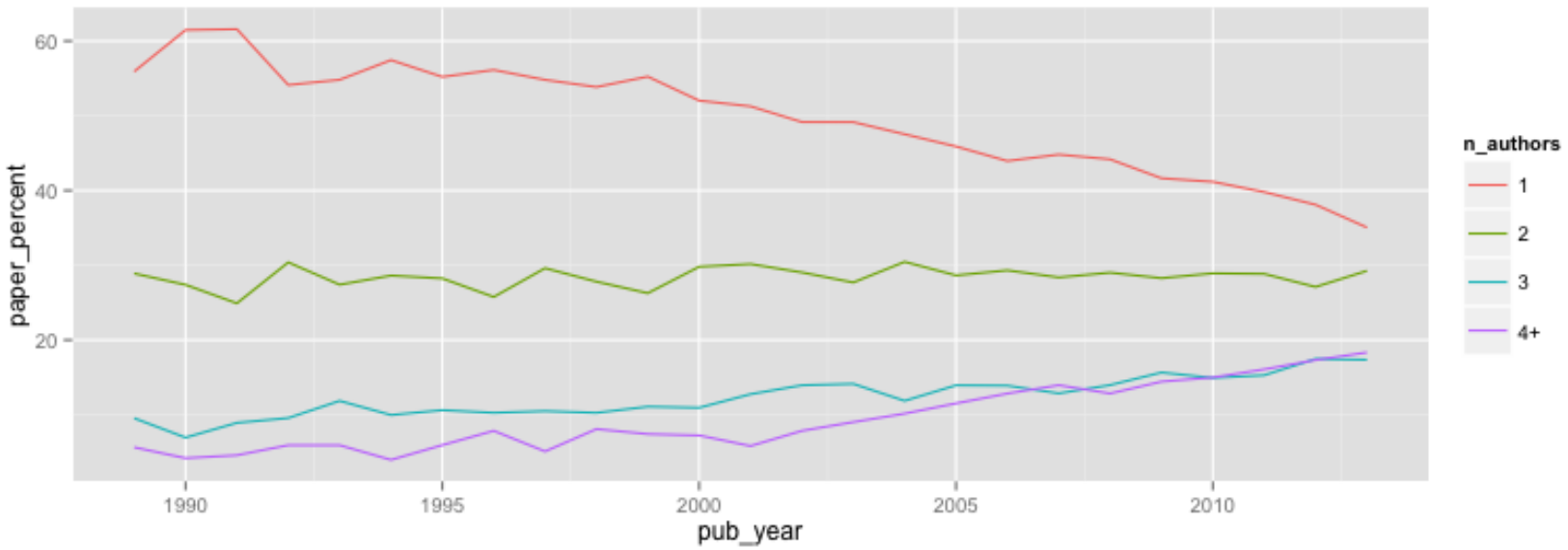
Science of science

- Measurement of scientific growth
- Co-authorship and network analysis
- Topic analysis
- Citation analysis

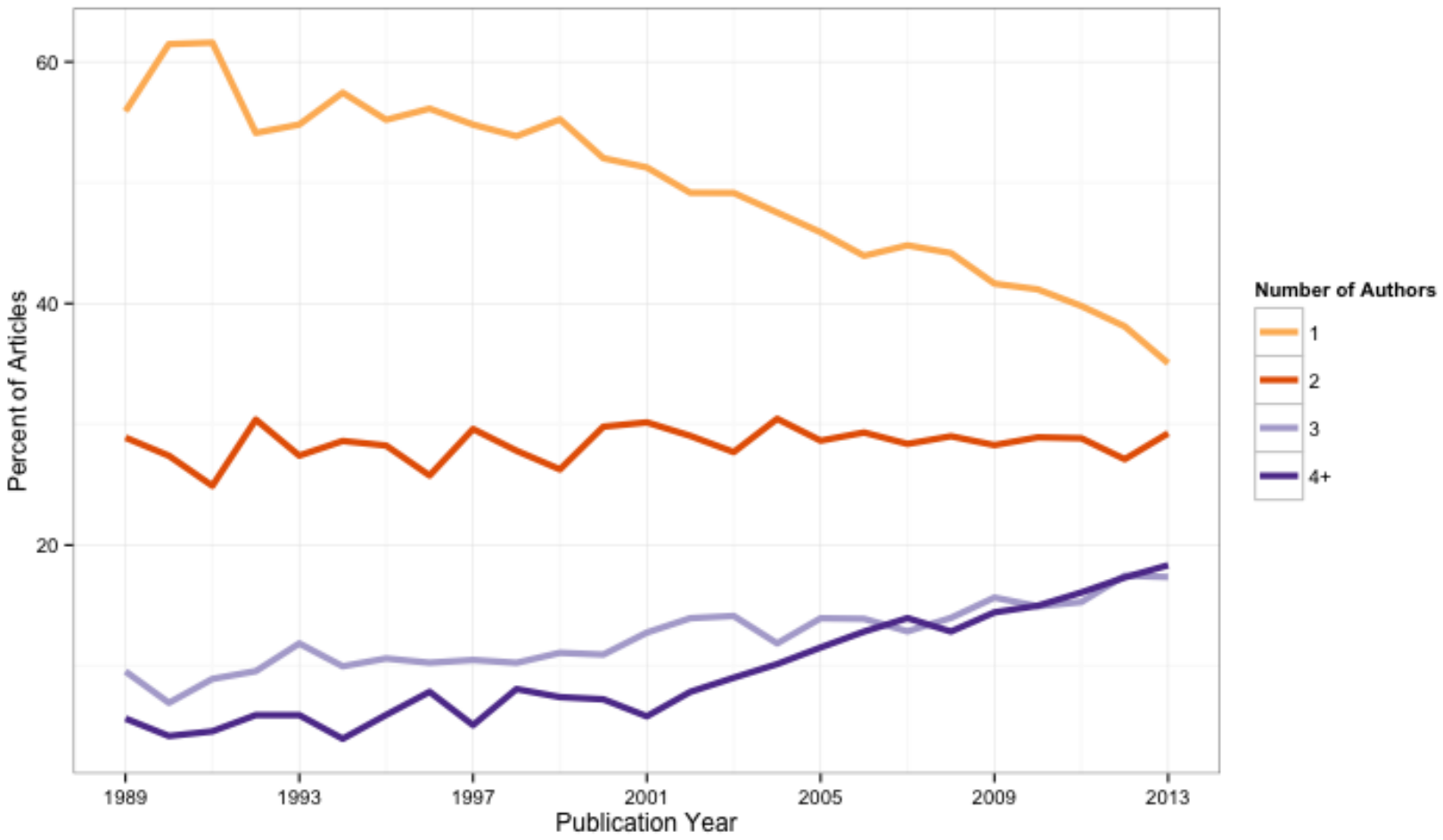
Overview of data

- Population set of social work journals ($N = 88$)
- Search of unique social science databases ($N = 35$)
- Retrieve all available article records in past 25 years ($N = 35k$)

```
minimal <- ggplot(author_count, aes(pub_year, paper_percent,
colour=n_authors)) + geom_line()
```



```
enhanced <- ggplot(author_count, aes(pub_year, paper_percent, colour=n_authors)) +
  geom_line(size = 1.5) +
  scale_colour_manual(values = cp) +
  labs(colour = "Number of Authors") +
  scale_x_continuous(breaks = c(seq(1989, 2013, 4))) +
  theme(axis.title.x = element_text(vjust=-0.5),
        axis.title.y = element_text(vjust=.75),
        axis.text.x = element_text(size=8),
        title = element_text(size = 10)) +
  theme_bw() +
  xlab("Publication Year") +
  ylab("Percent of Articles")
```


60
40
20
Percent of Articles
1989
1993
1997
2001
2005
2009
2013
Publication Year
Number of Authors
1
2
3
4+

```
> summary(minimal)
data: pubYear, pub_year, paper_percent, n_authors [100x4]
mapping:  x = pub_year, y = paper_percent, colour = n_authors
faceting: facet_null()
-----------------------------------
geom_line:
stat_identity:
position_identity: (width = NULL, height = NULL)

> summary(enhanced)
data: pubYear, pub_year, paper_percent, n_authors [100x4]
mapping:  x = pub_year, y = paper_percent, colour = n_authors
scales:   colour, x, xmin, xmax, xend, xintercept
faceting: facet_null()
-----------------------------------
geom_line: size = 1.5
stat_identity:
position_identity: (width = NULL, height = NULL)
```

```
facet_plot <- ggplot(author_count, aes(pub_year, paper_percent,
colour=n_authors)) + geom_line() + facet_wrap(~ journal)
```