UNIVERSITY OF SUSSEX | BUSINESS SCHOOL | SCIENCE POLICY RESEARCH UNIT

Data visualisation

Daniele Rotolo

Introductory Data Science for Innovation (995N1) – Weeks 3, 11 October 2021

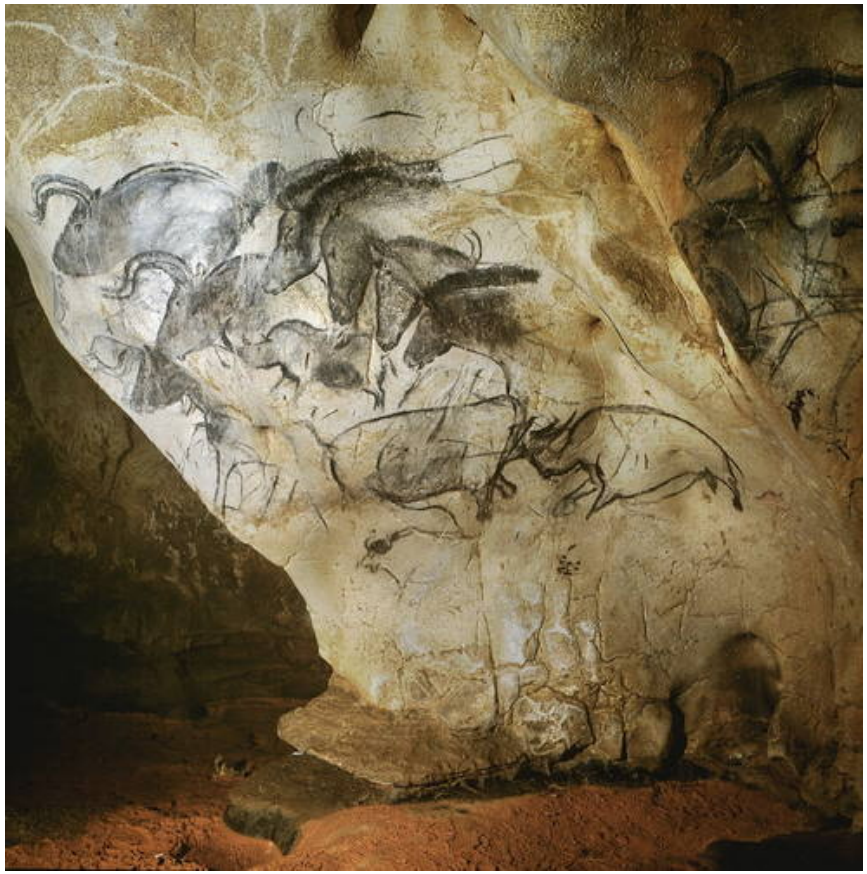
Outline

Outline

- Why visualising data?
- Visualisation types
- Beyond statistics and visualisation
- Data visualisation in R

Why visualising data?

Why visualising data?



Source: Figurative drawings of the Grotte Chauvet-Pont d'Arc, 30,000-32,000 BP

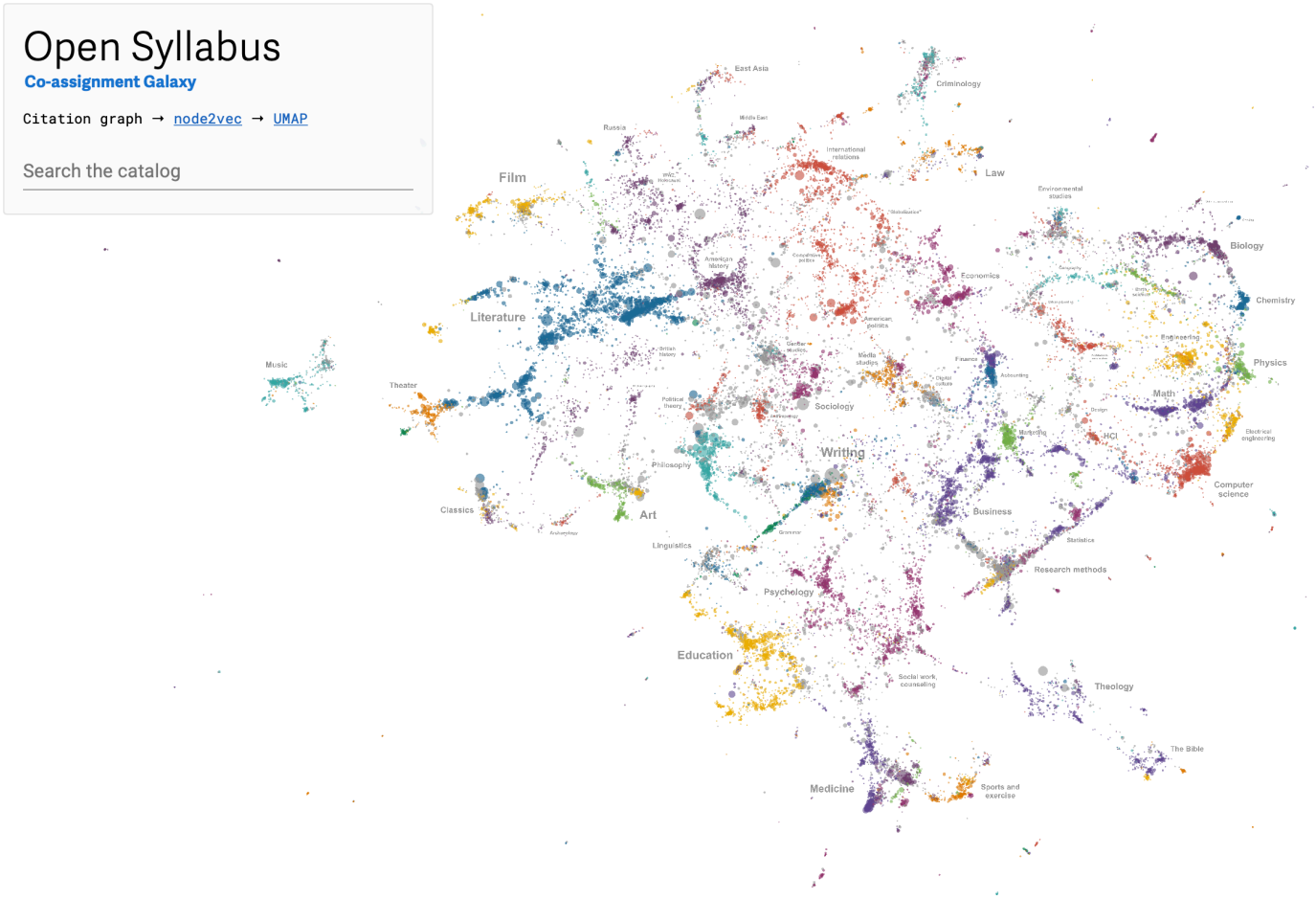
(http://whc.unesco.org/)

- Half of our brain is dedicated to processing **visual signals**
- We can process images **60,000 times faster** than text

Why visualising data?

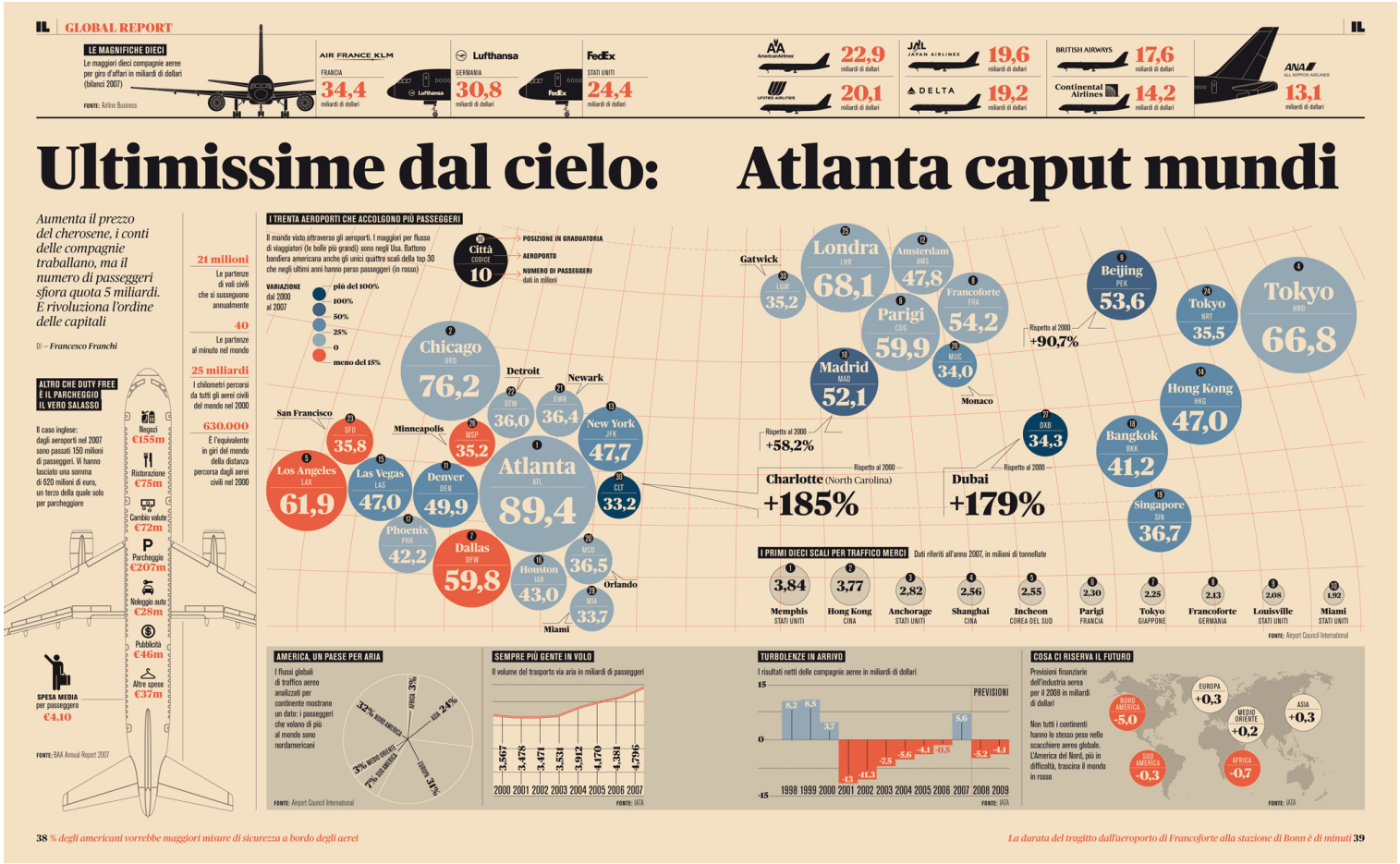
- Increasing capabilities to collect, store and analyse data
  - government data
  - APIs
  - …
- Large amount of real time data
- Risk of **information overload**
- **Data visualisation**
  - to simplify the complexity
  - to identify actual and misleading patterns

… to simplify the complexity



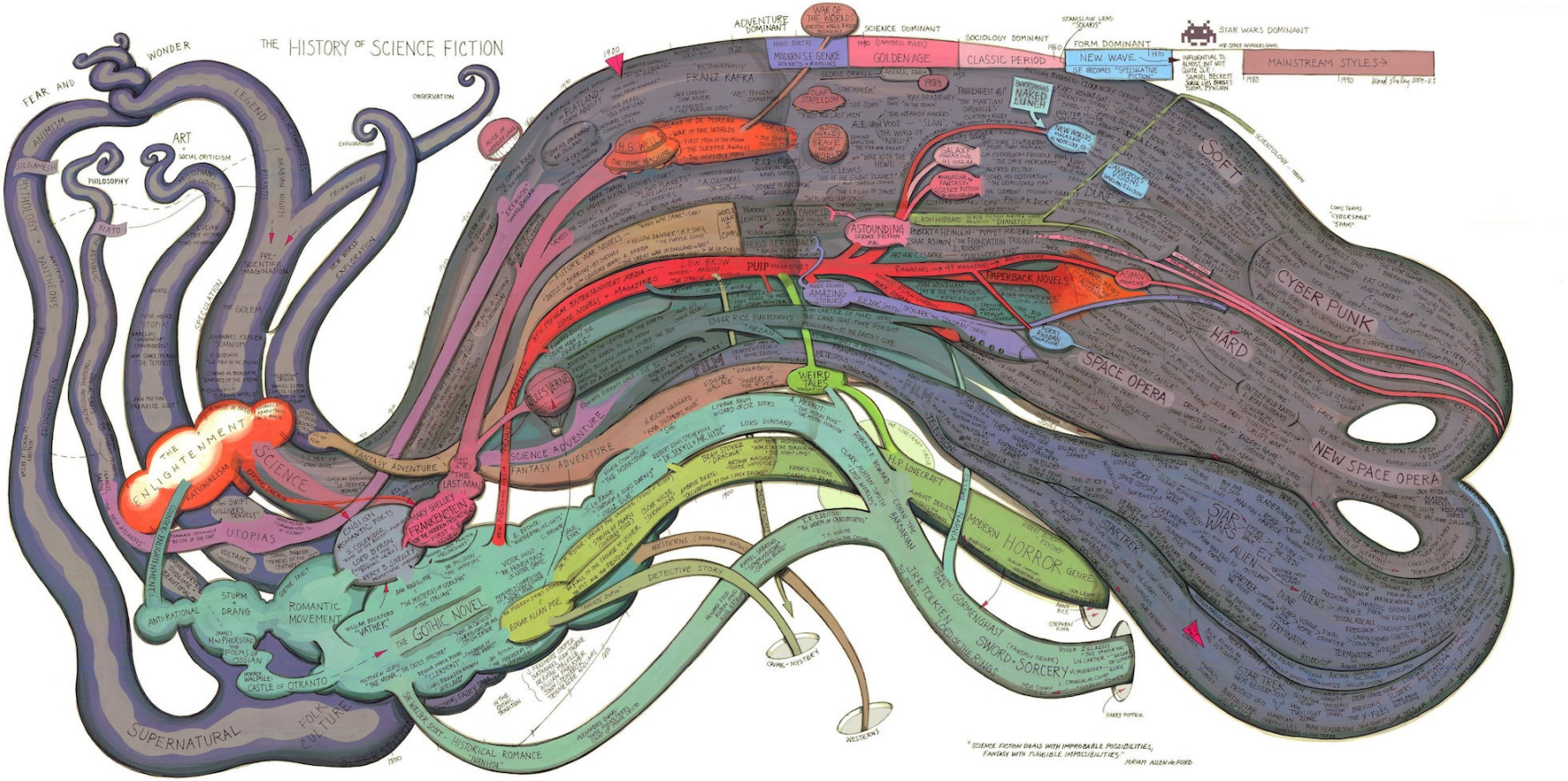
Source: https://opensyllabus.org

… to simplify the complexity



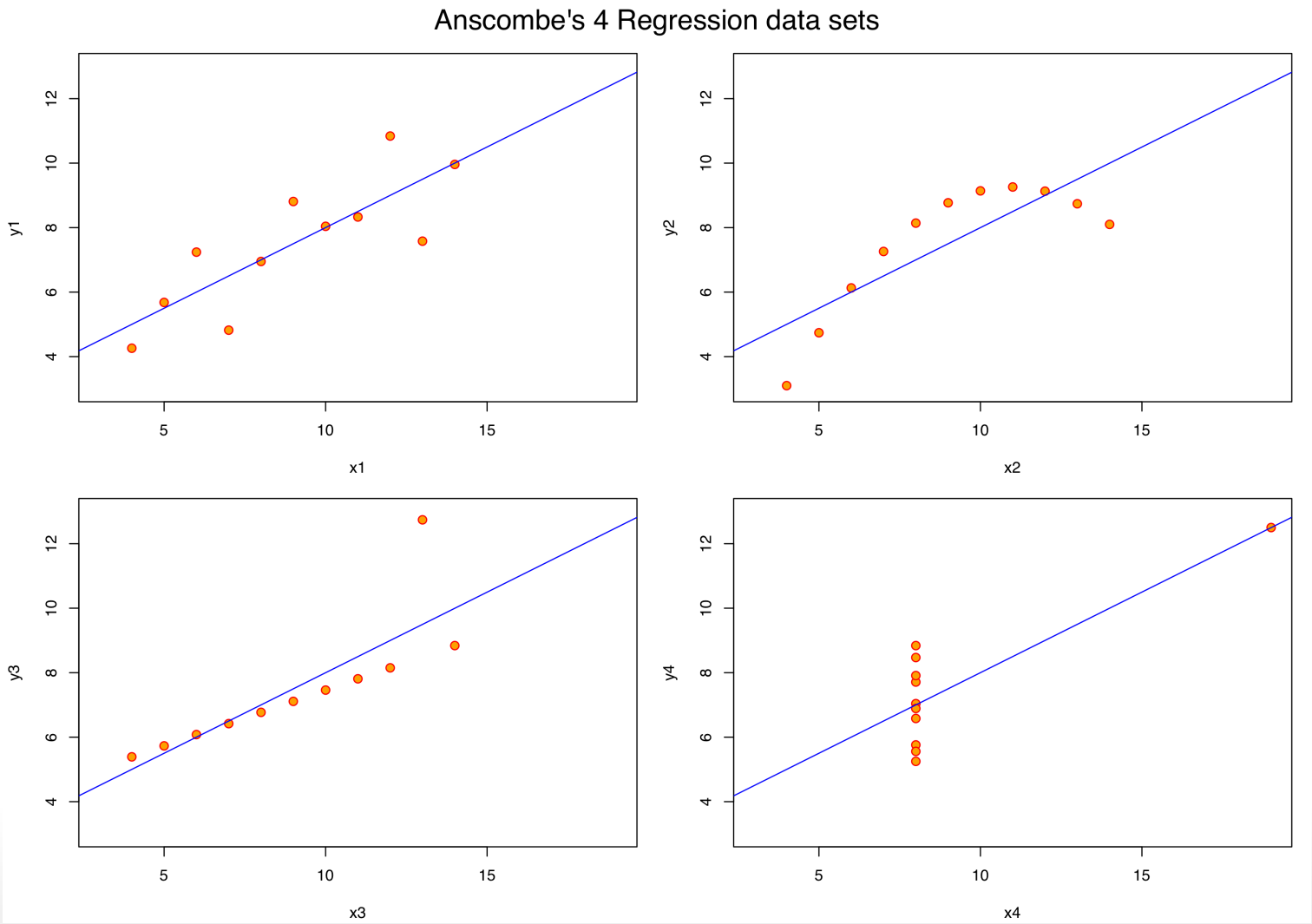
Source: Il Sole 24 Ore, November 2008

… to simplify the complexity

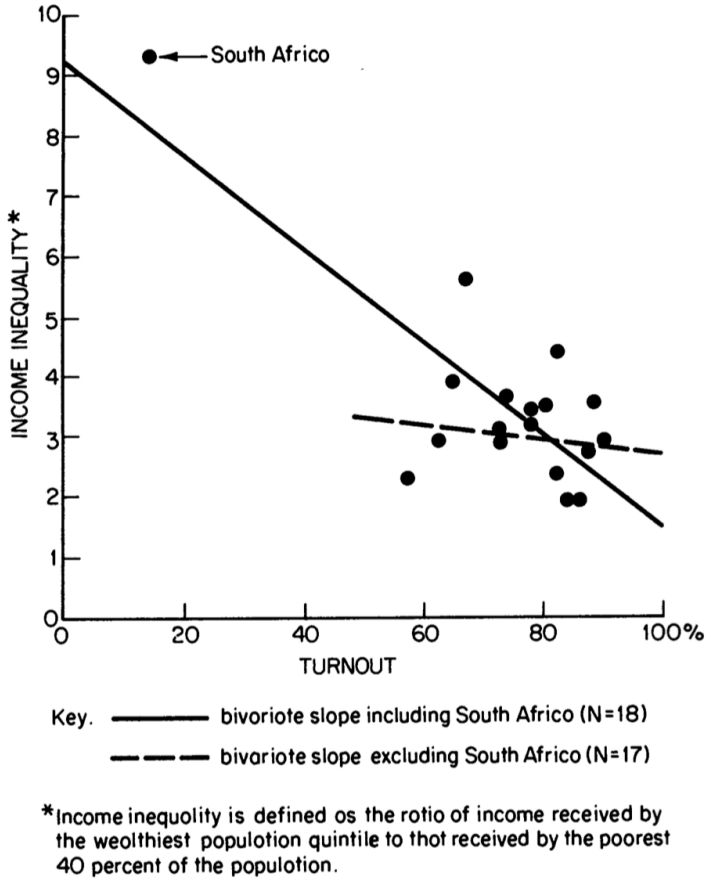


Source: VanderMeer (2013)

… to identify actual and misleading patterns



… to identify actual and misleading patterns



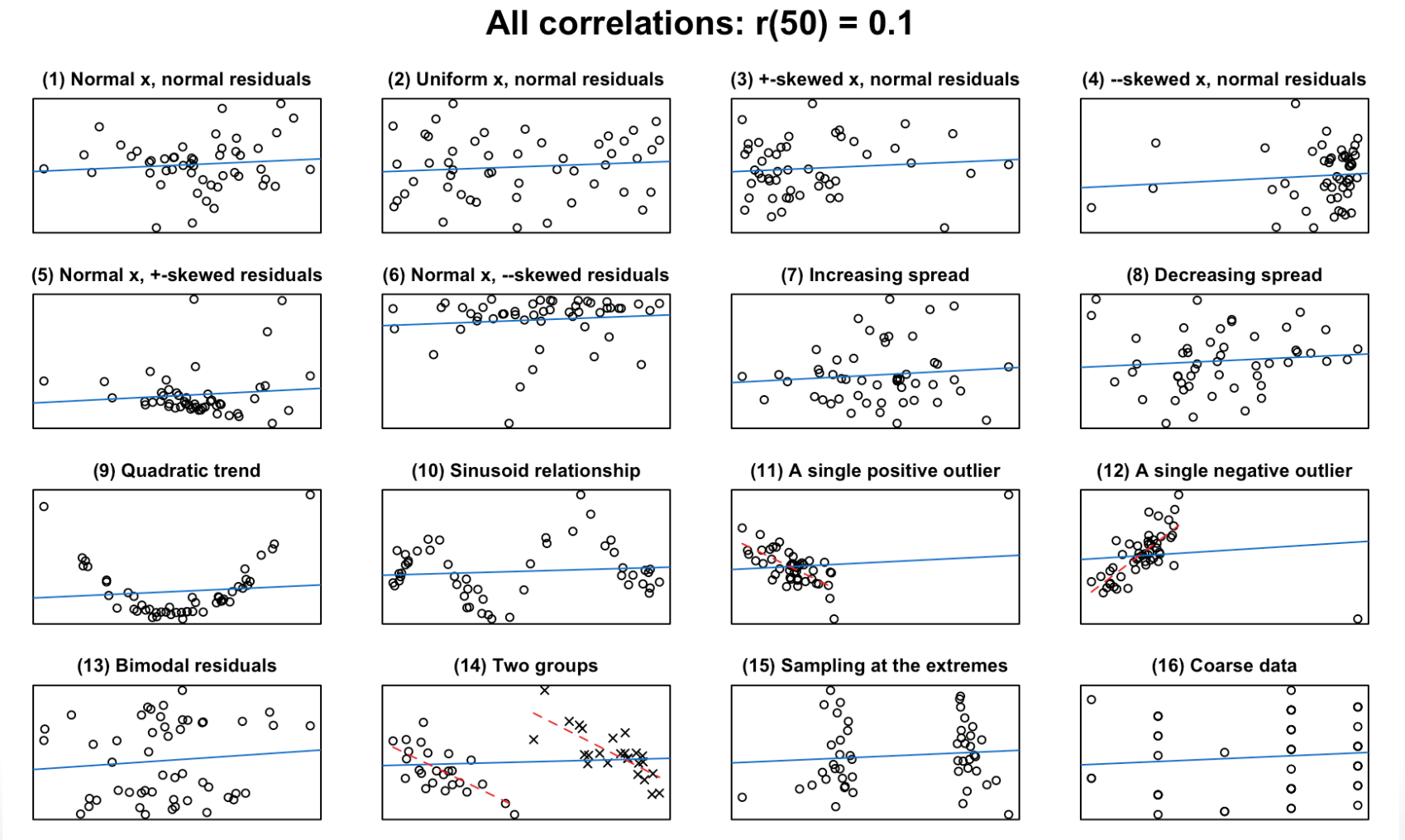
Source: Jackman (1980)

… to identify actual and misleading patterns

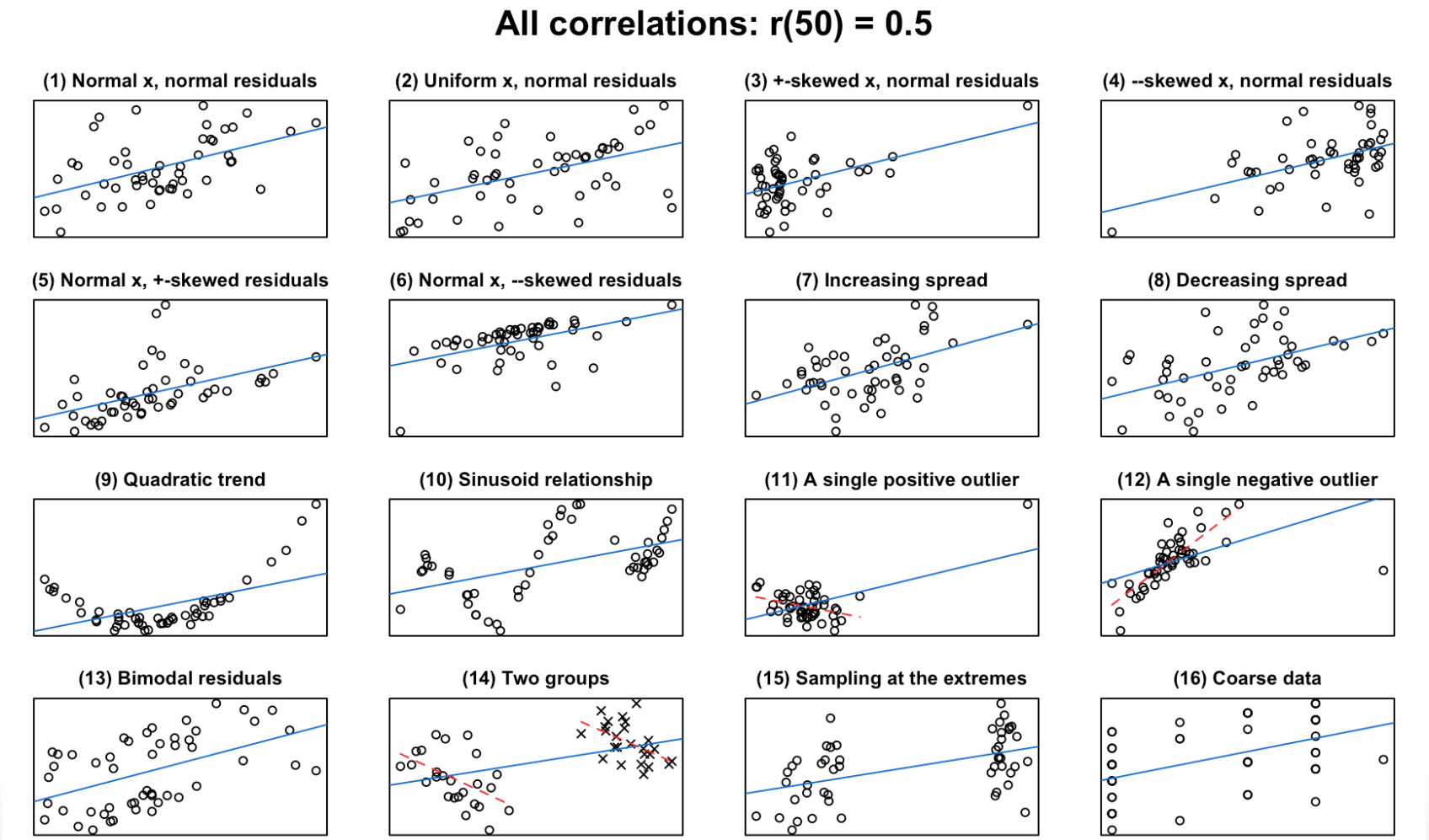
- Jan Vanhove has provided a script in R to explore the diversity of patterns that may lie behind a simple correlation measure
- You can specify the correlation (`r`) and the number of observations (`n`)

```
source("http://janhove.github.io/RCode/plot_r.R")
plot_r(r = 0.1, n = 50)
```

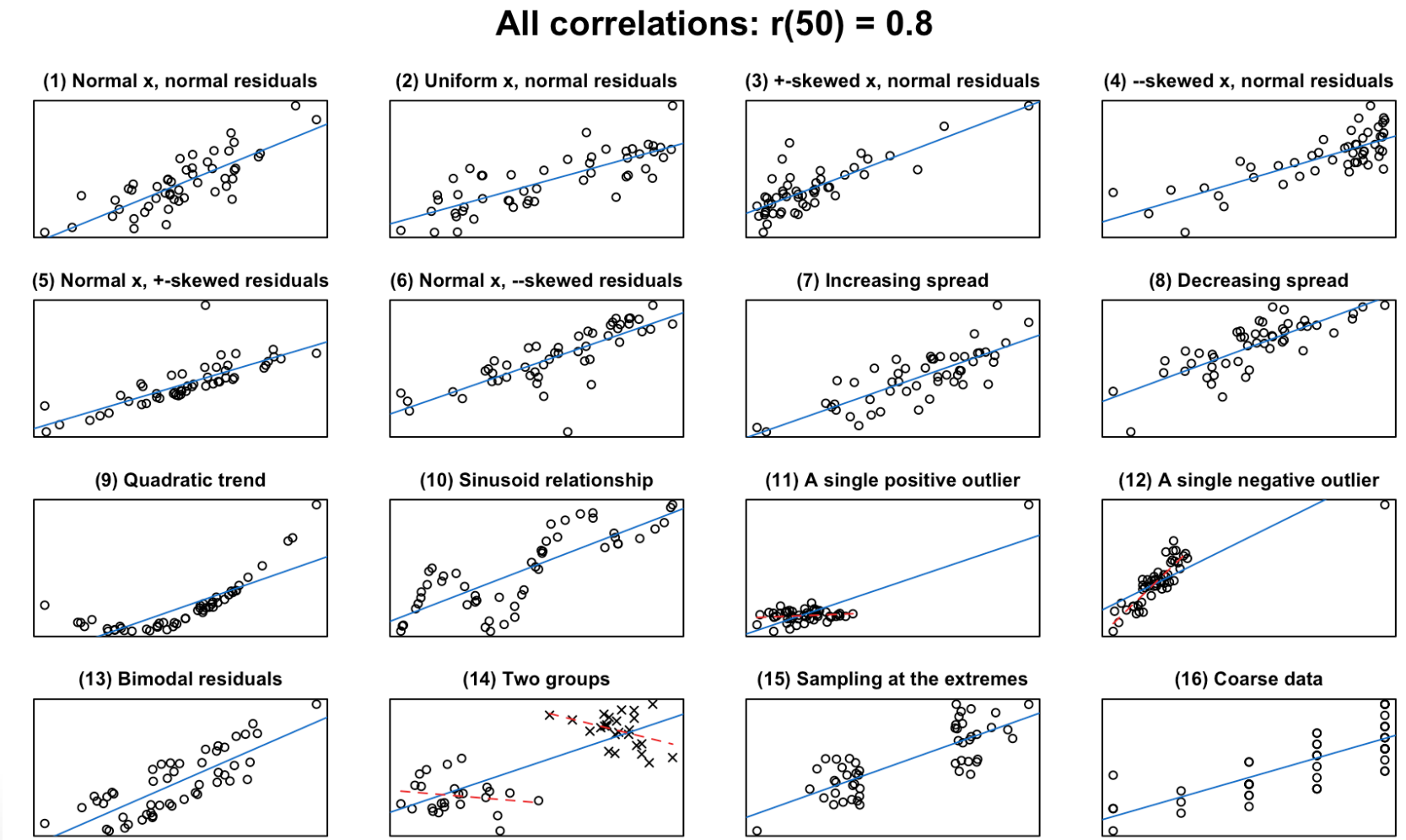
… to identify actual and misleading patterns



… to identify actual and misleading patterns



… to identify actual and misleading patterns



Visualisation types

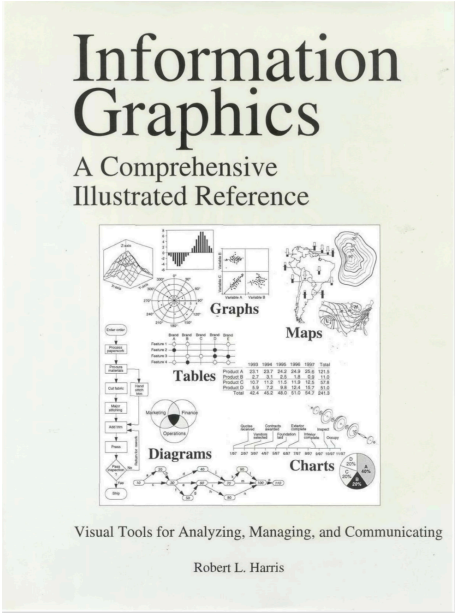
Visualisation types

Visualisation types by Börner (2016)

- **Tables:** values arranged in rows and columns (e.g. summary tables)
- **Charts:** qualitative and quantitative representations of data without a reference system (e.g. pie charts)
- **Graphs:** qualitative and quantitative representations of data with a reference system (e.g. scatterplot)
- **Maps:** geographical distribution of data (e.g. geographical maps)
- **Networks layouts:** data represented as sets of nodes and links between these, i.e. relationships between nodes (e.g. co-occurrence networks)

Visualisation types

- One type of visualisation can be **converted** into another (e.g. a pie chart can be transformed into a bar chart)
- Different visualisation types can be **combined** to examine data from different a number of angles
- See Harris (1996) for a comprehensive list of **visualisation types**



Tables: *One way*

Name	Age
Alice	6
Bill	7
Harry	4
Sue	5

Two variables along the vertical axis without rules

Name	Age
Alice	6
Bill	7
Harry	4
Sue	5

Two variables along vertical axis with rules

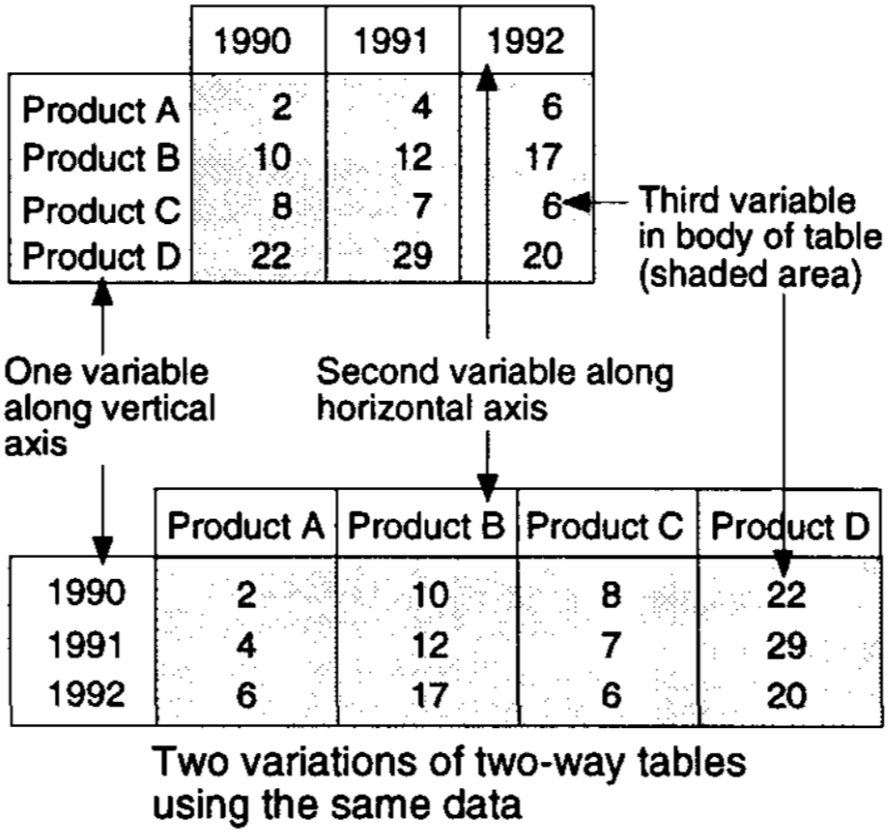
Two variables along horizontal axis with rules

Name	Alice	Bill	Harry	Sue
Age	6	7	4	5

Three variations of one-way tables

Source: Harris (1996)

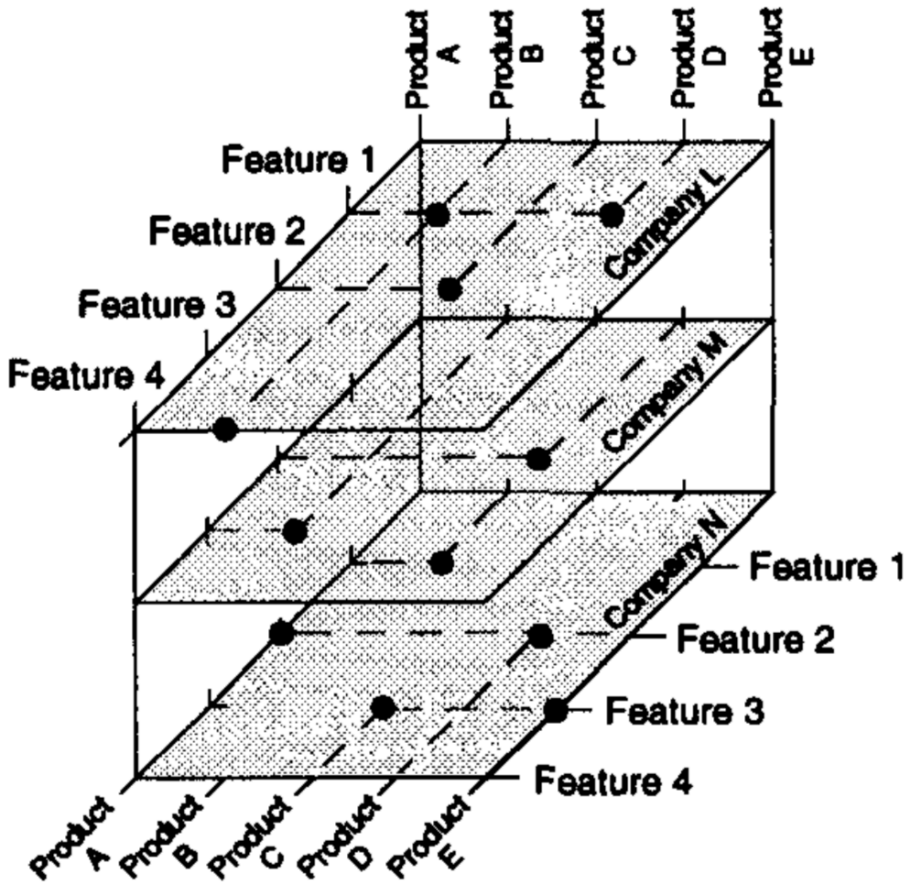
Tables: *Two way*



Two variations of two-way tables using the same data

Source: Harris (1996)

Tables: *3D*



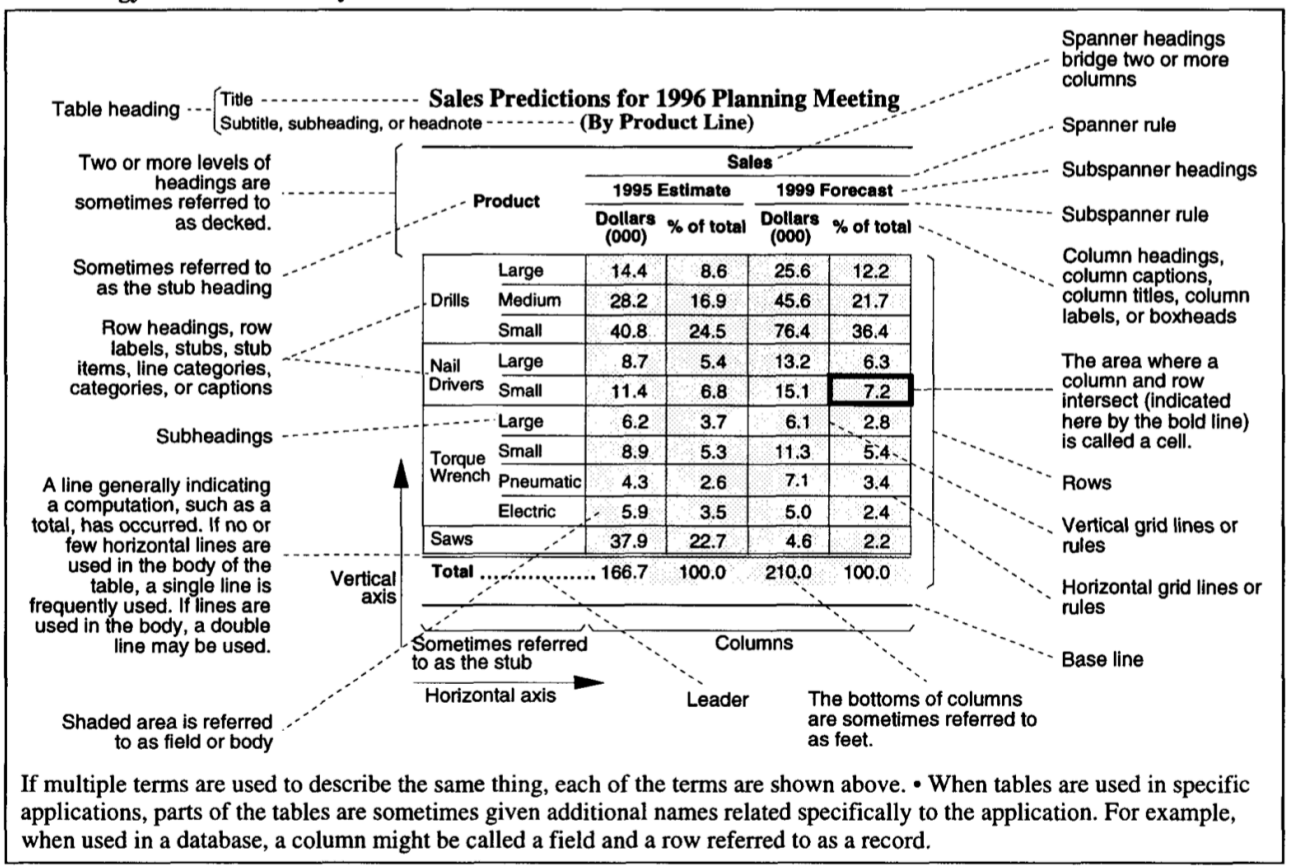
Source: Harris (1996)

Tables: …

… many other variations

Tables: *Terminology*

Terminology and location of key elements

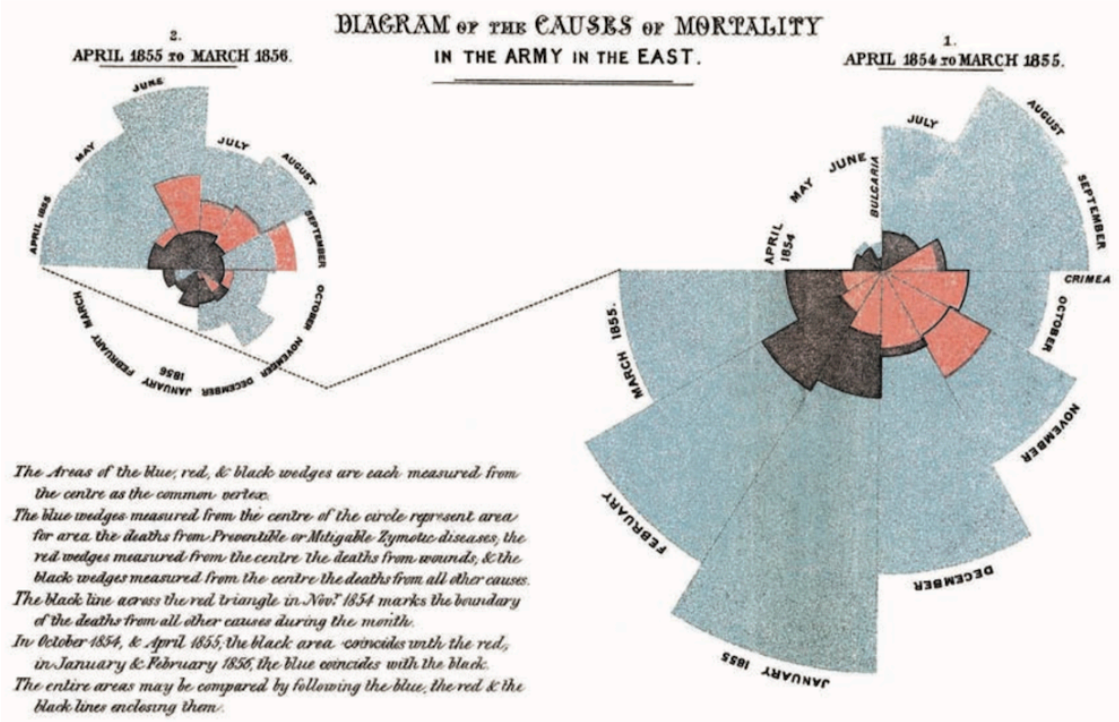


If multiple terms are used to describe the same thing, each of the terms are shown above. • When tables are used in specific applications, parts of the tables are sometimes given additional names related specifically to the application. For example, when used in a database, a column might be called a field and a row referred to as a record.

Source: Harris (1996)

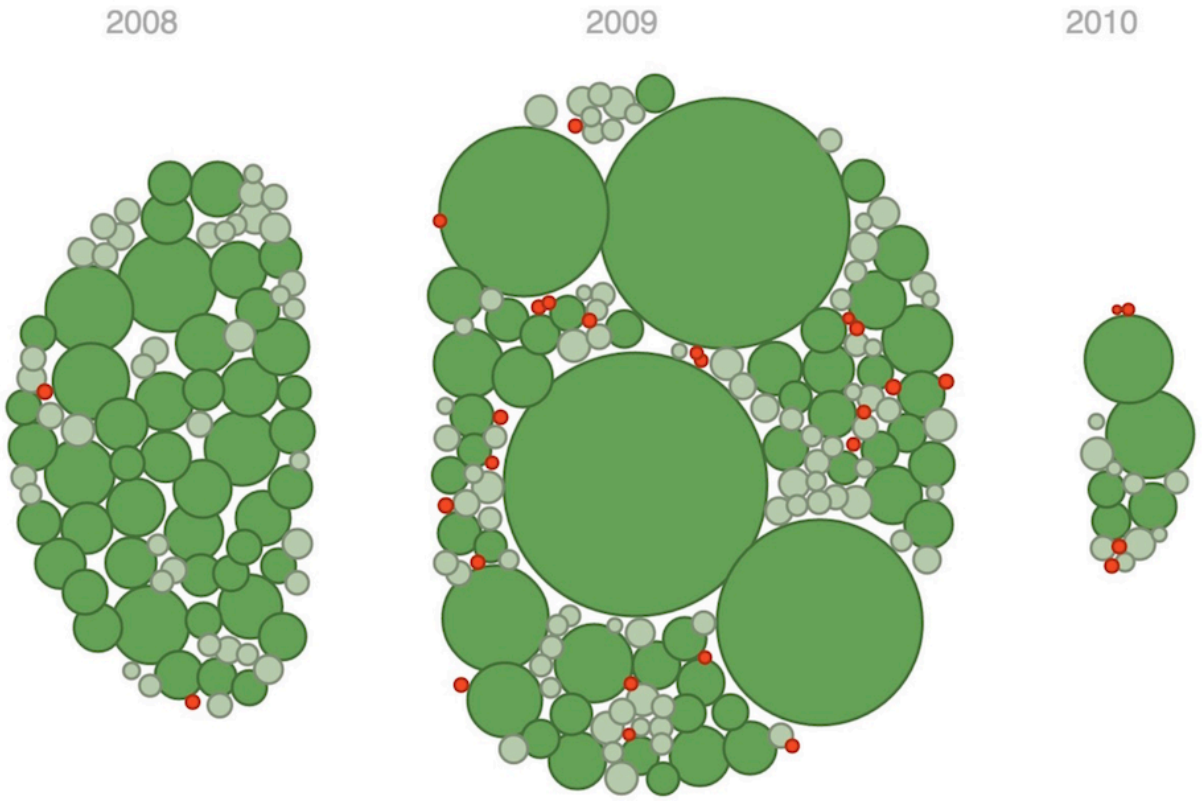
Charts: *Pie charts*

Causes of Mortality in the British Military during the Crimean War



Source: Börner (2016)

Charts: *Bubble charts*



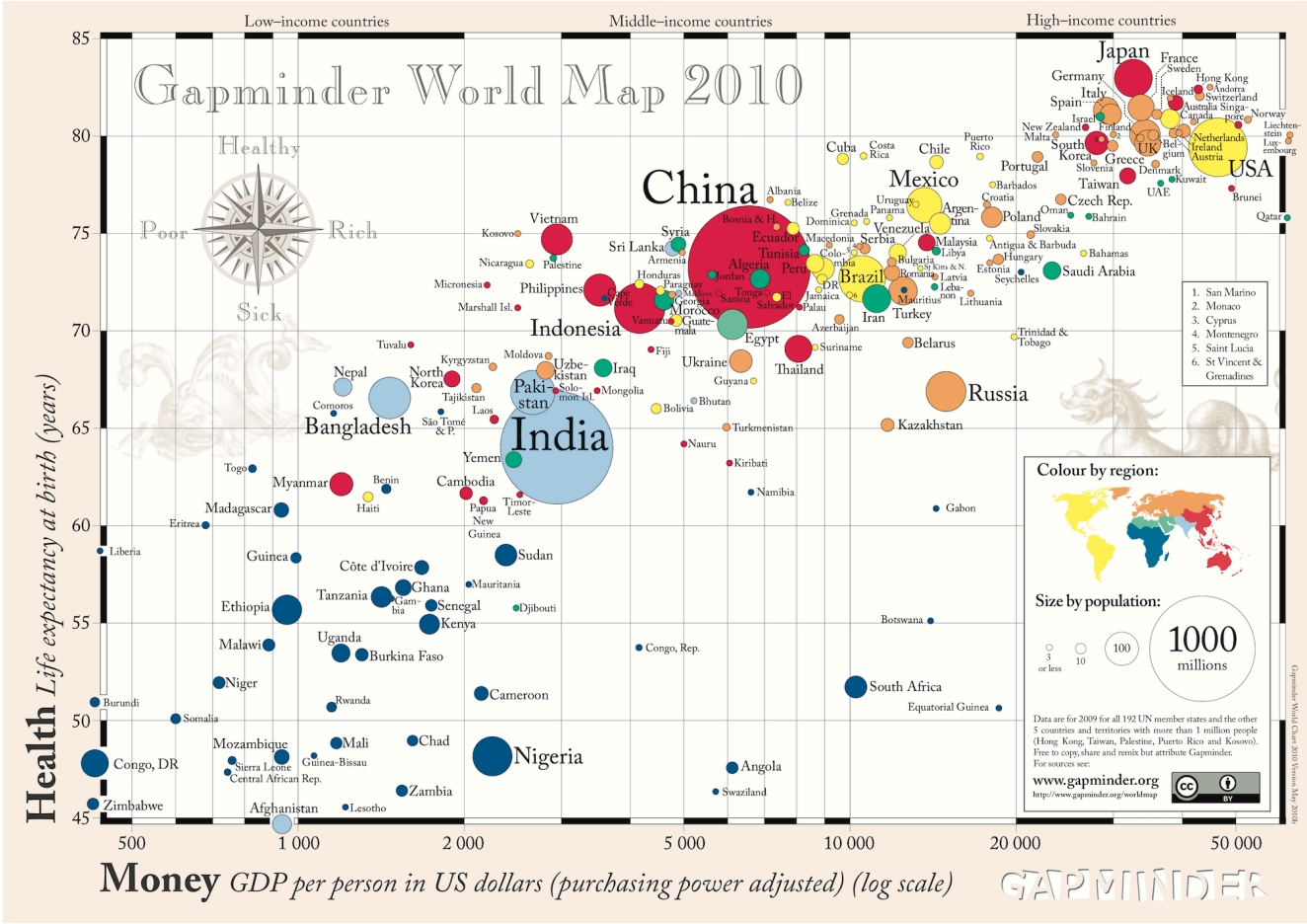
Source: Gates Foundation's education-based donations (color and size represent grant amount)

Charts: *Word clouds*

Charts: …

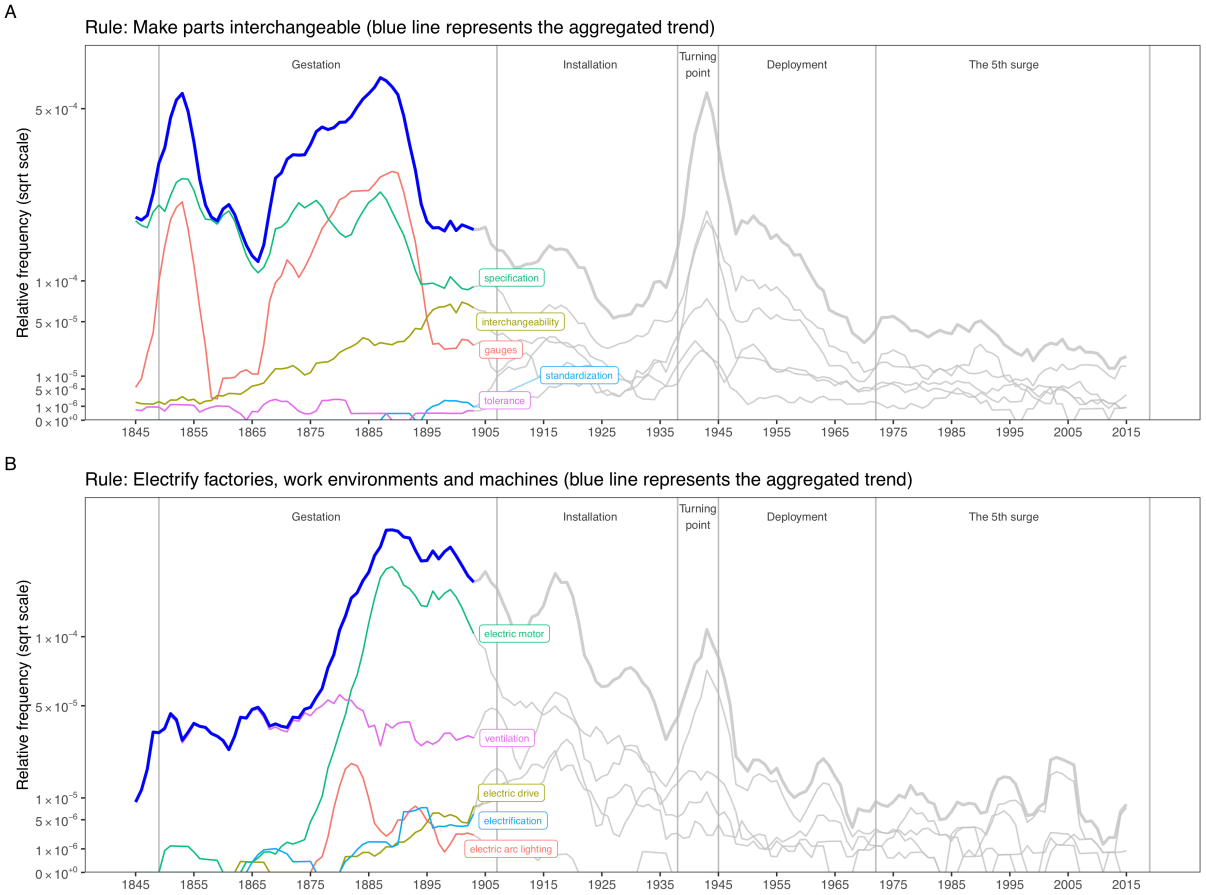
… many other variations

Graphs: *Bubble graphs*



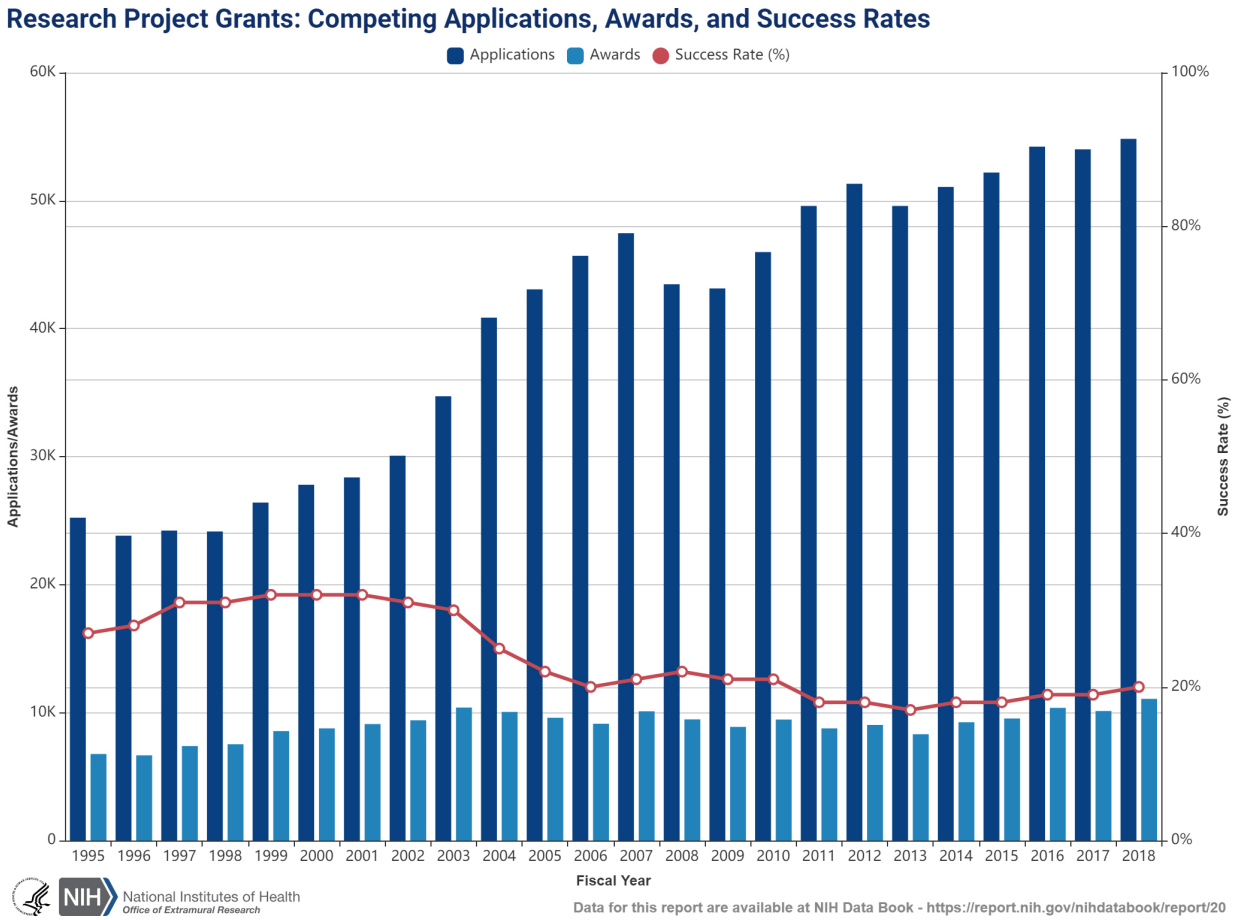
Source: https://www.gapminder.org

Graphs: *Line graphs*



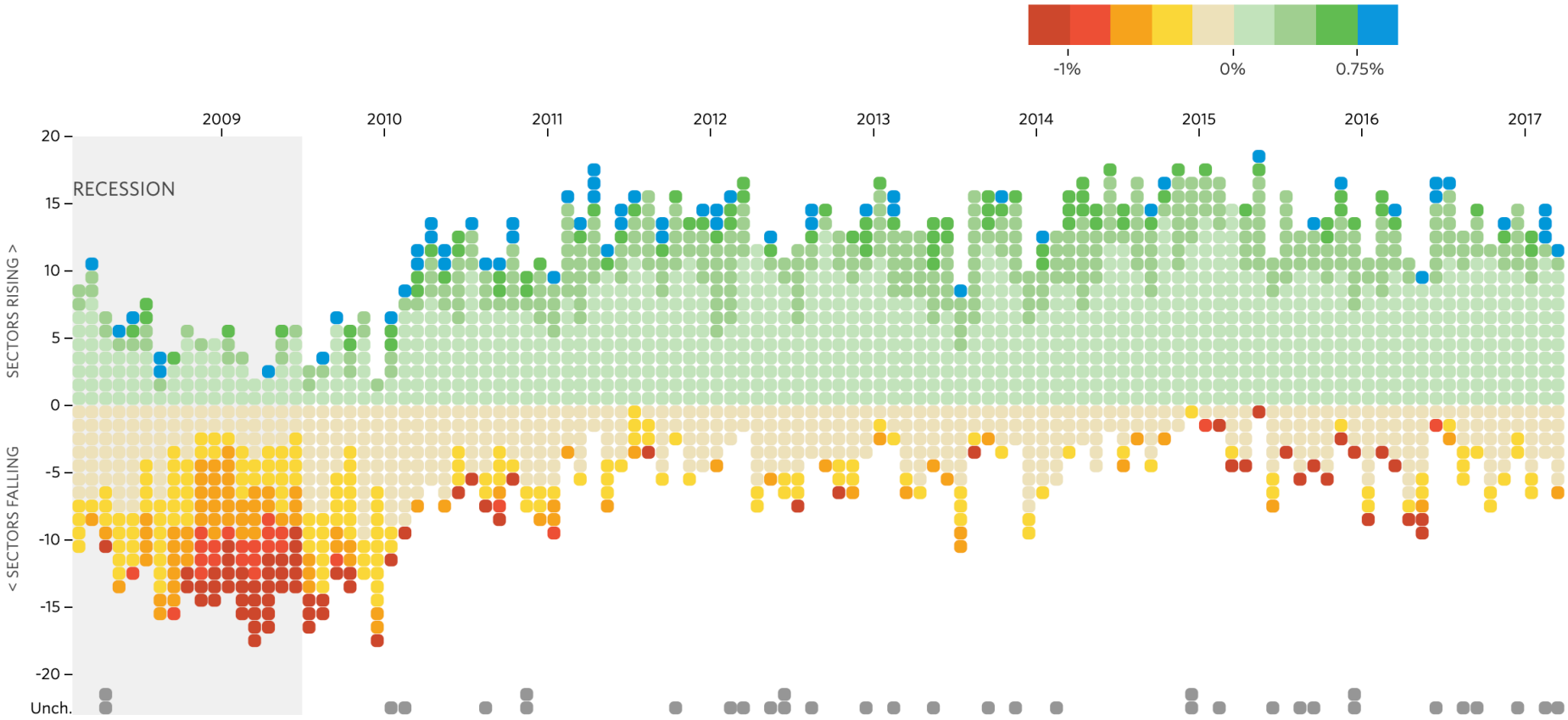
Source: Bone and Rotolo (2020)

Graphs: *Line and bar graphs*



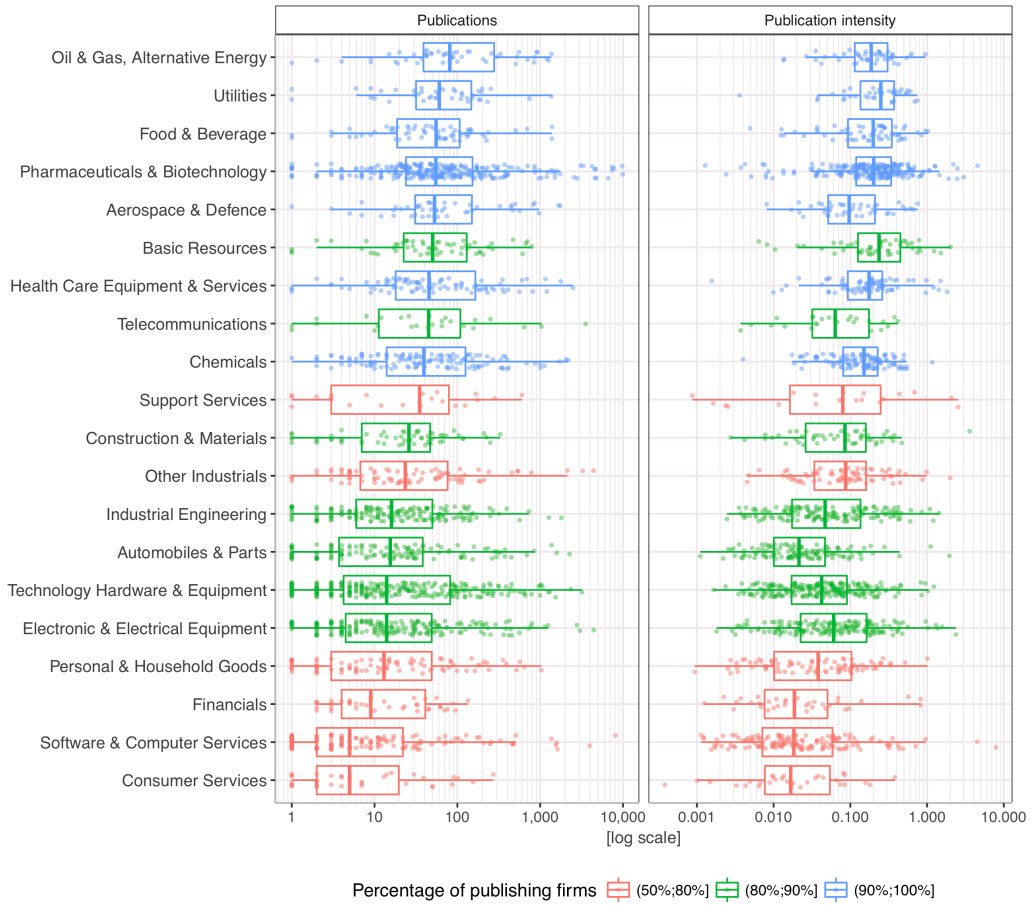
Source: https://www.nih.gov/news-events/nih-research-grants-digital-press-kit

Graphs: *Interactive bar graphs*



Source: Winners and Losers: Job Gains and Losses, Wall Street Journal

Graphs: *Box plot*



Source: Camerani, Rotolo, and Grassano (2018)

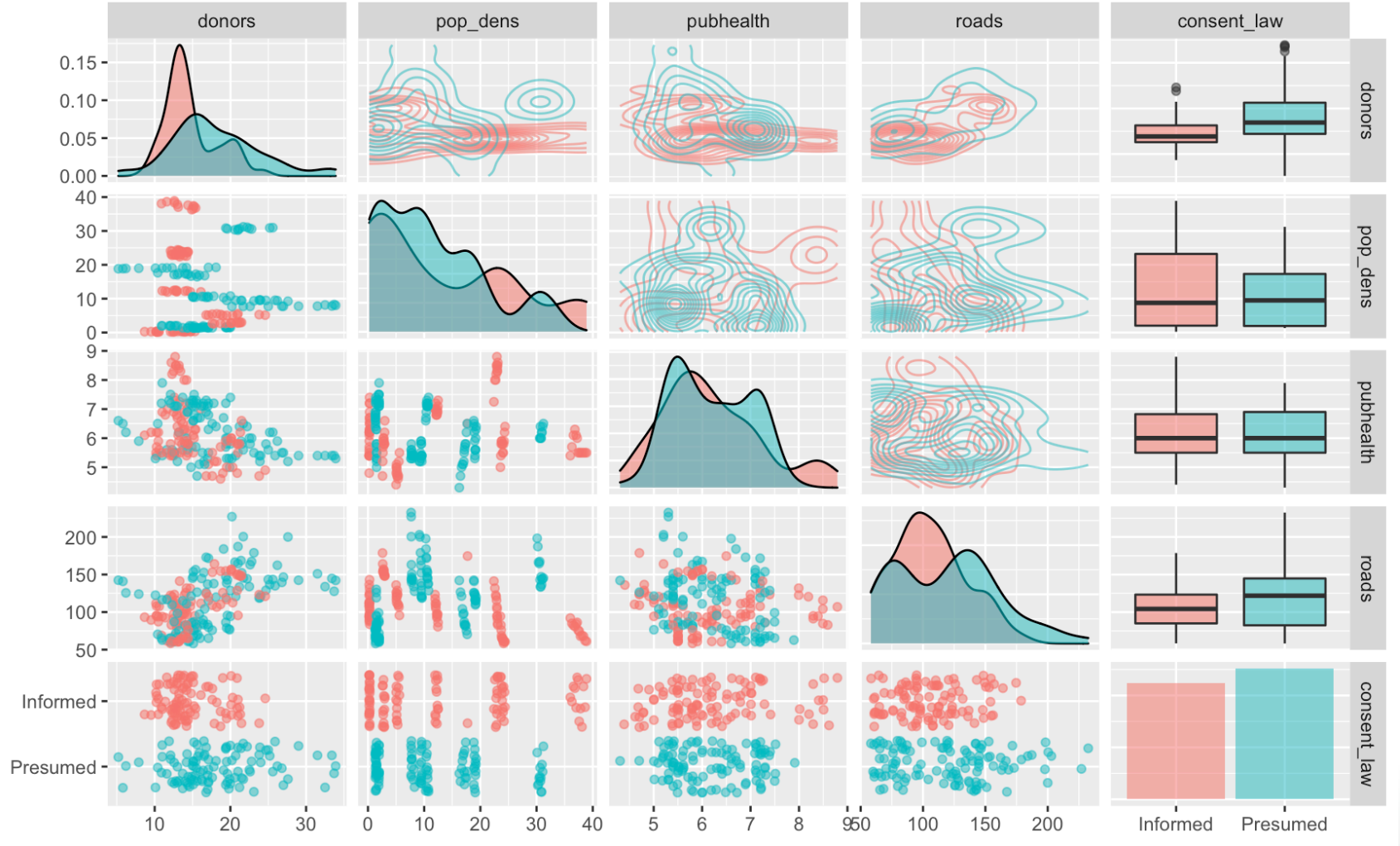
Graphs: *Statistical visualisations*

```
library(tidyverse)
library(GGally)
load(file = "organdata.rda")

organdata_sm <- organdata %>%
  select(donors, pop_dens, pubhealth,
         roads, consent_law)

g <- ggpairs(data = organdata_sm,
             mapping = aes(color = consent_law, alpha = 0.5),
             upper = list(continuous = wrap("density"), combo = "box_no_facet"),
             lower = list(continuous = wrap("points"), combo = wrap("dot_no_facet")))
```

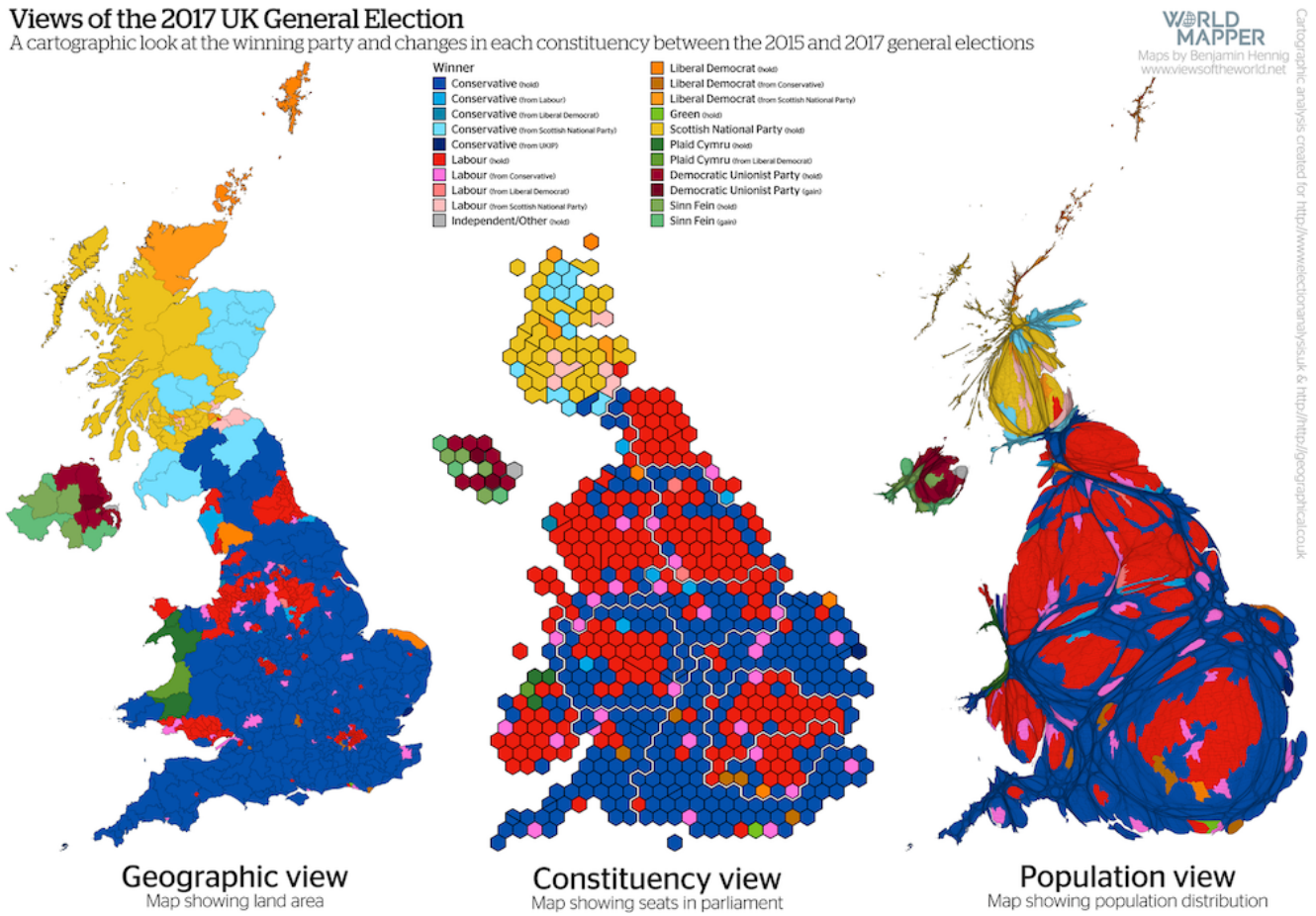
Graphs: *Statistical visualisations*



Graphs: …

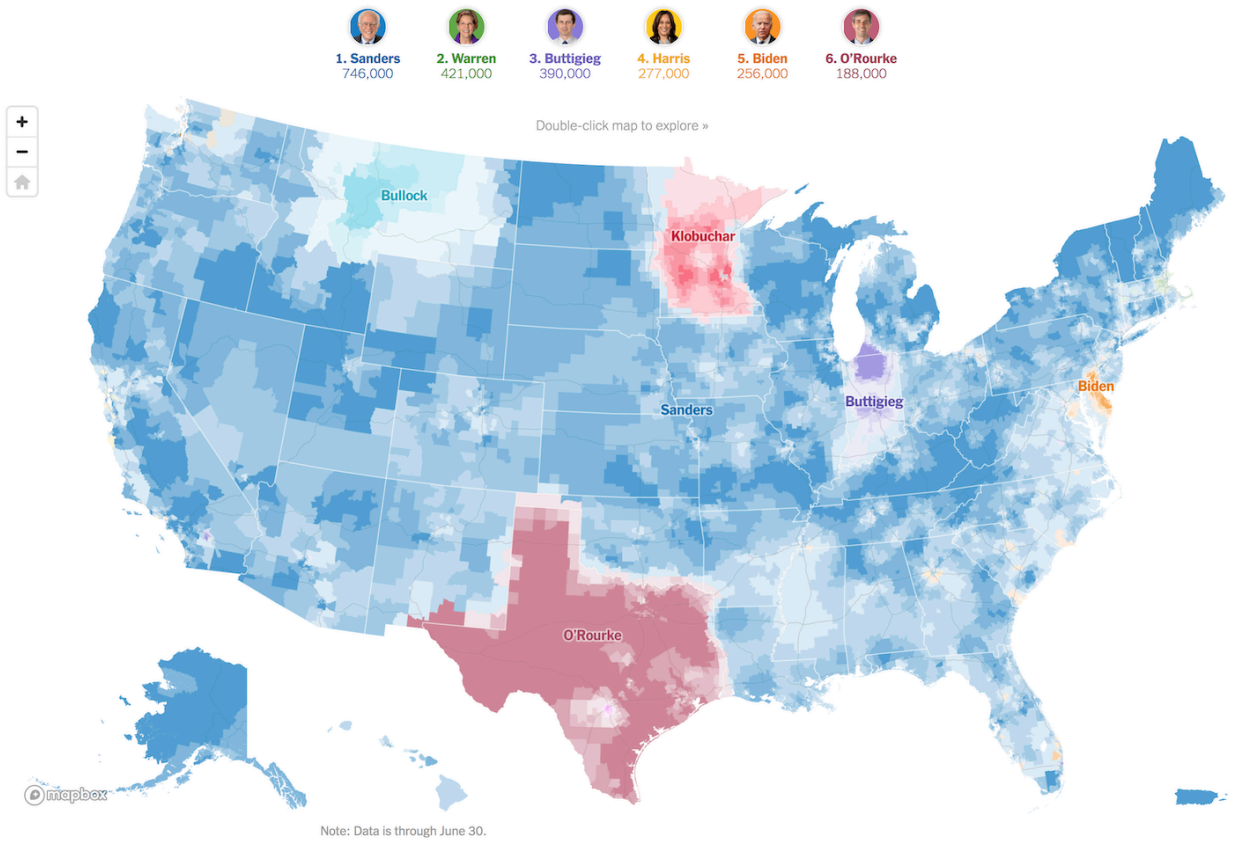
… many other variations

Maps: *Cartograms*



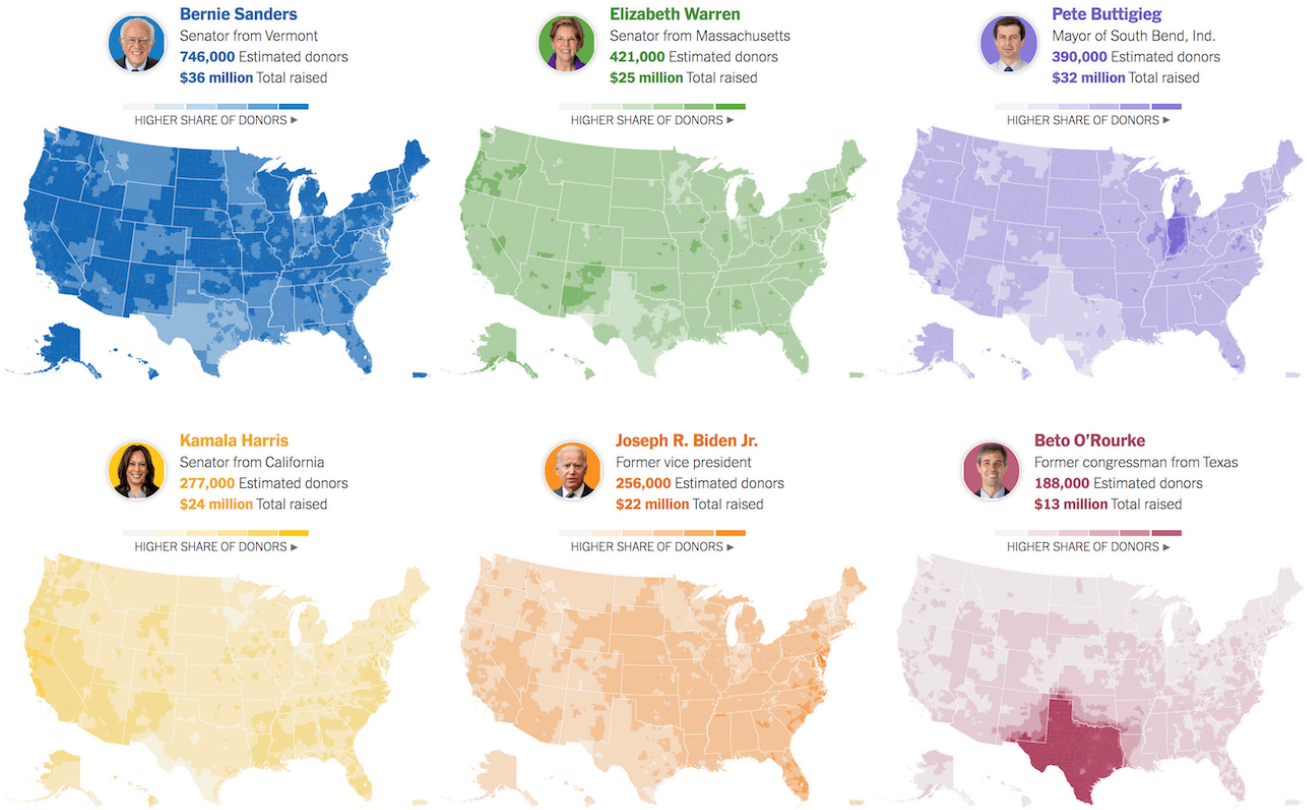
Source: https://geographical.co.uk/places/mapping/item/2276-mapping-the-2017-general-election

Maps: *Choropleth maps*



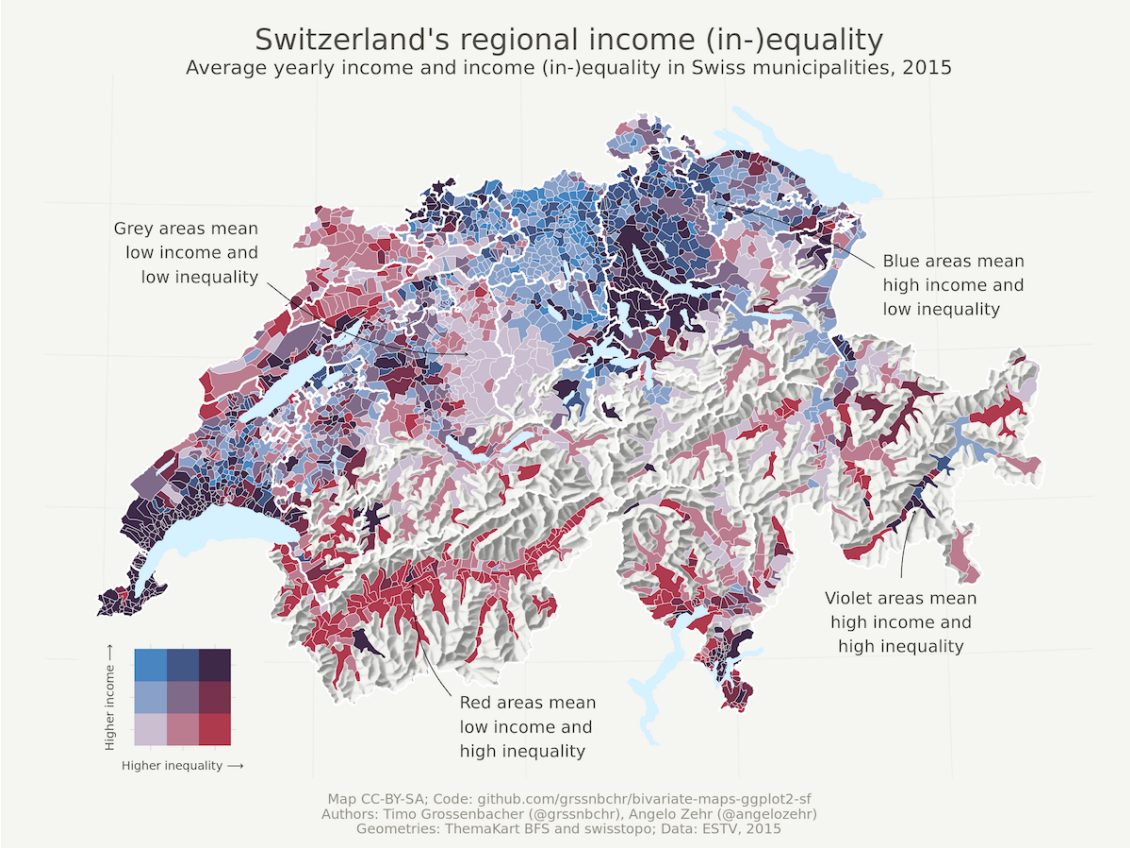
Source: https://www.nytimes.com/interactive/2019/08/02/us/politics/2020-democratic-fundraising.html

Maps: *Choropleth maps*



Source: https://www.nytimes.com/interactive/2019/08/02/us/politics/2020-democratic-fundraising.html

Maps: *Choropleth maps*

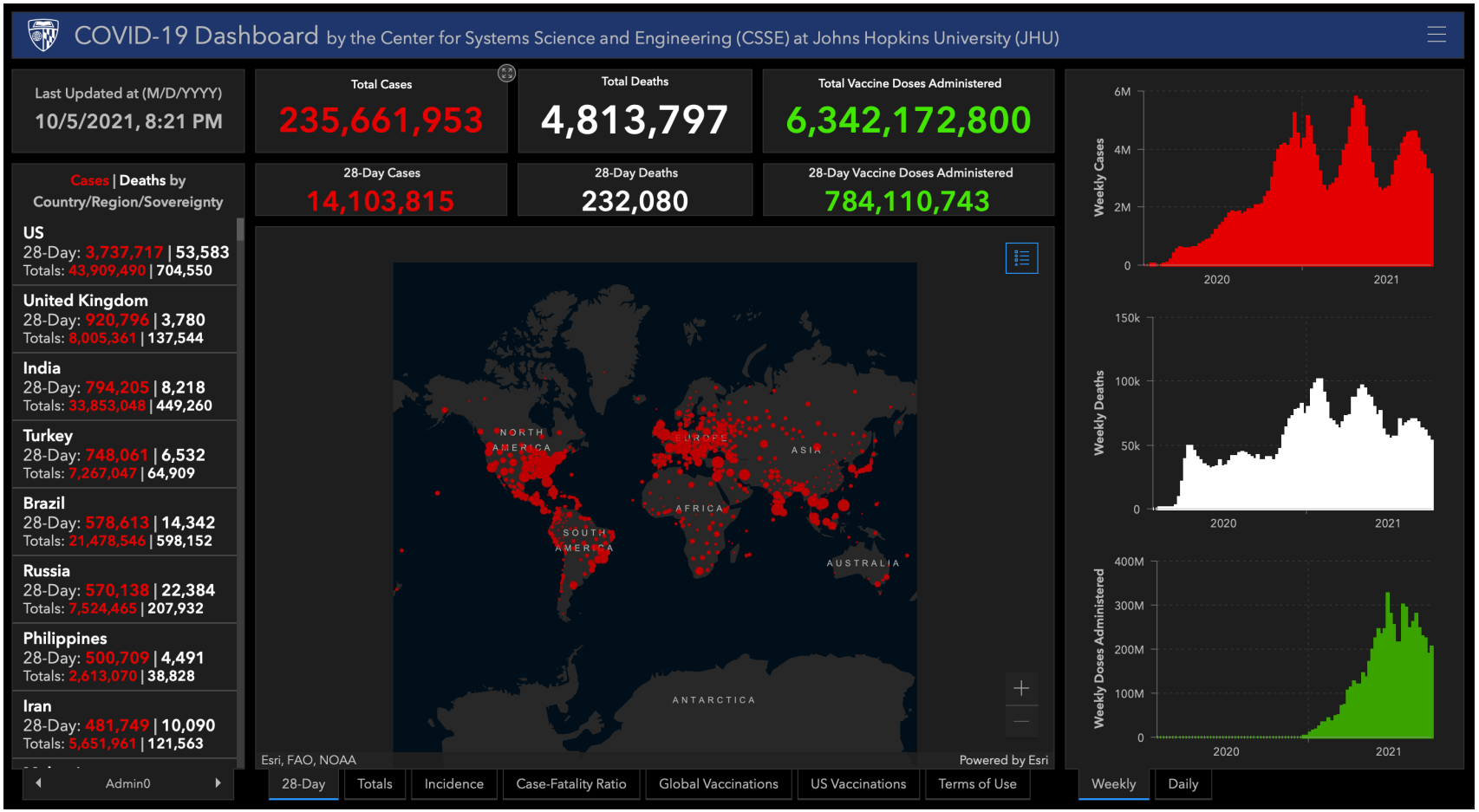


Source: https://timogrossenbacher.ch/2019/04/bivariate-maps-with-ggplot2-and-sf/

Maps: *Choropleth maps*

Source: https://centerforcollectivelearning.org/urbanperception

Maps: *Proportional symbol maps*

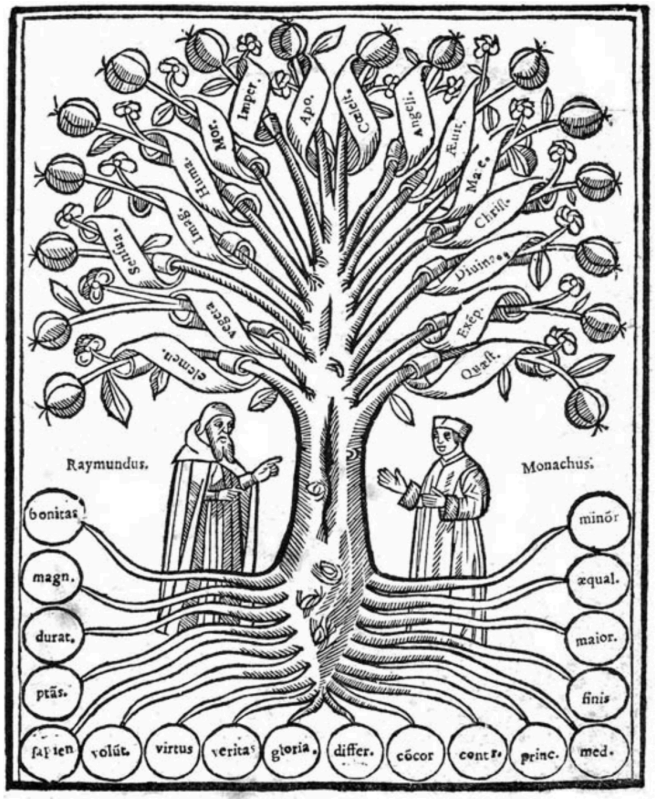


Source: Active COVID-19 cases on 5 Ocotber 2021 (https://www.gisaid.org/epiflu-applications/global-cases-covid-19/)

Maps: …

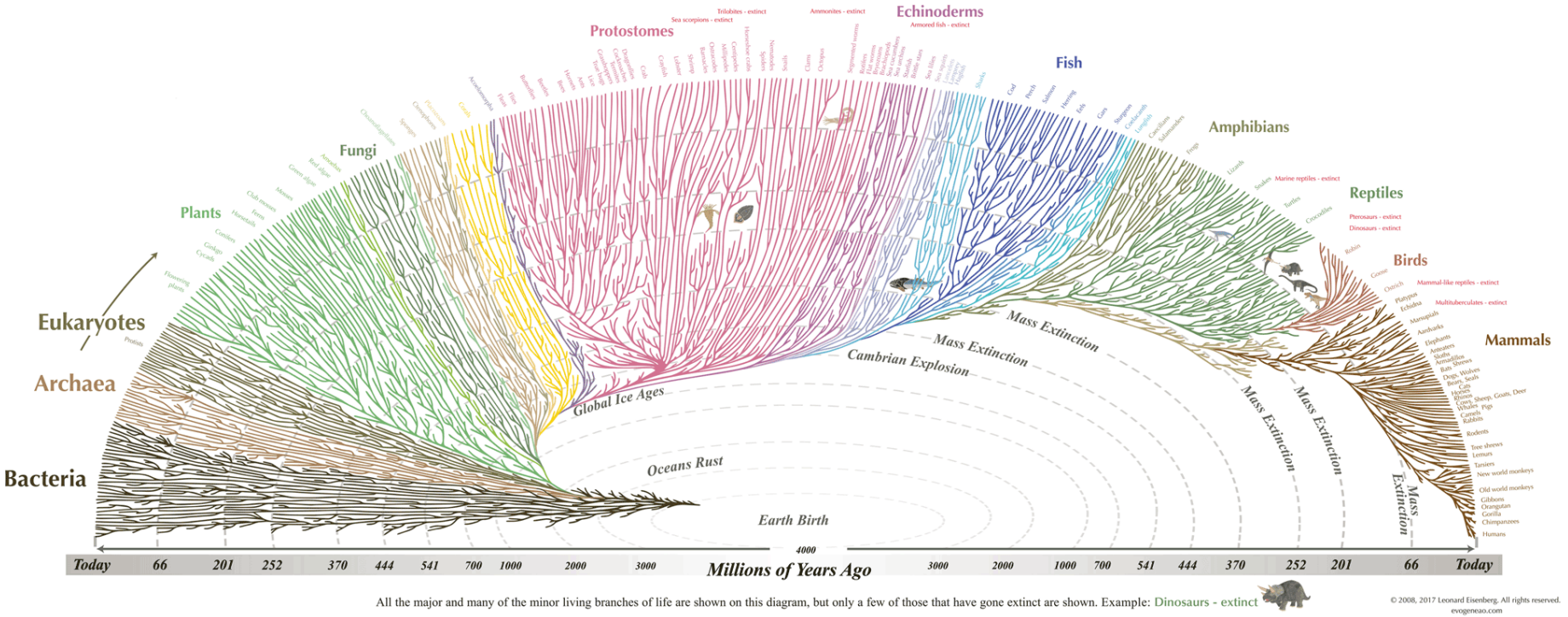
… many other variations

Network layouts: *Trees*



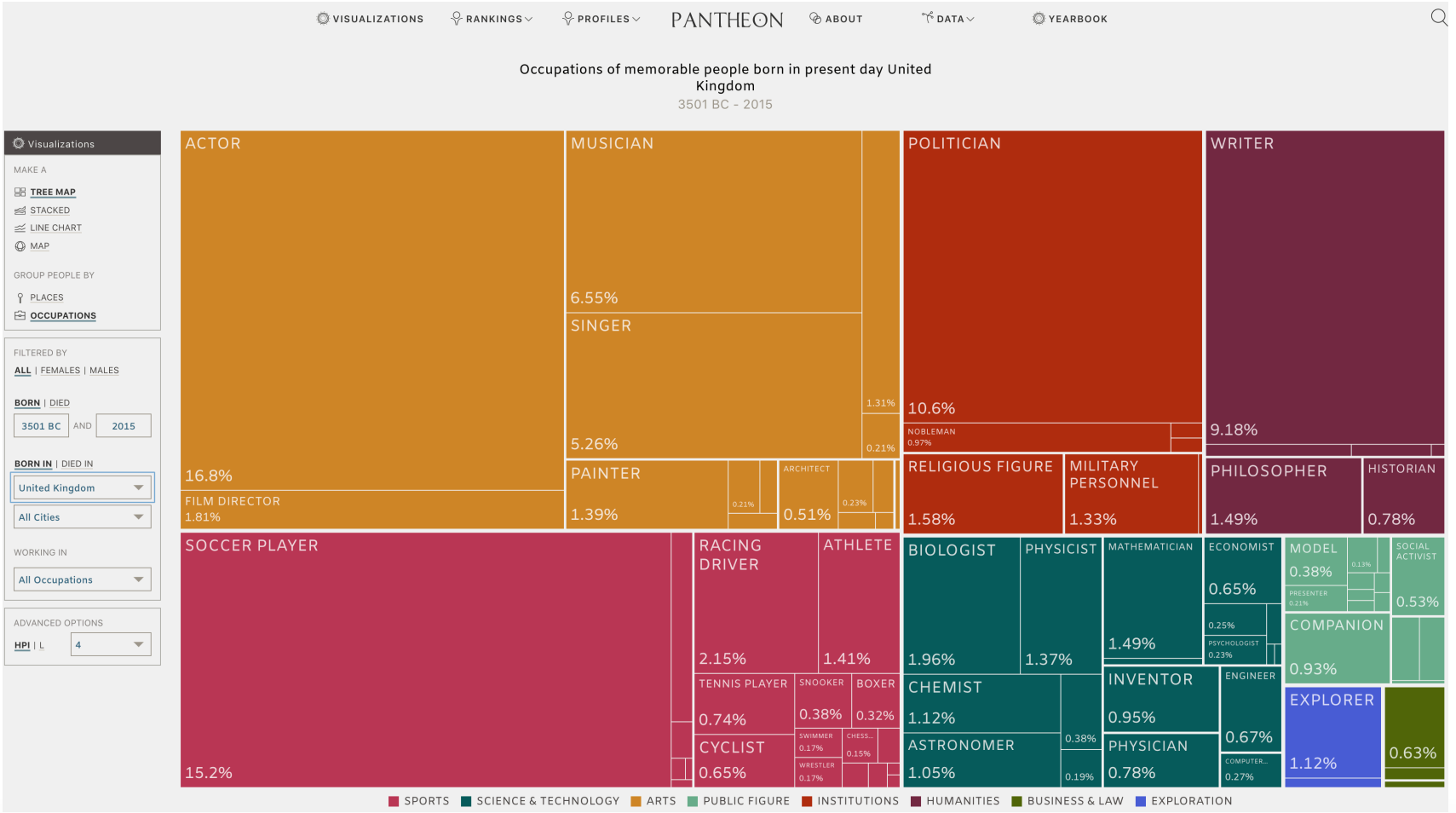
Source: 1296 Ramon Llull's **arbor scientiae** (tree of science)

Network layouts: *Trees*



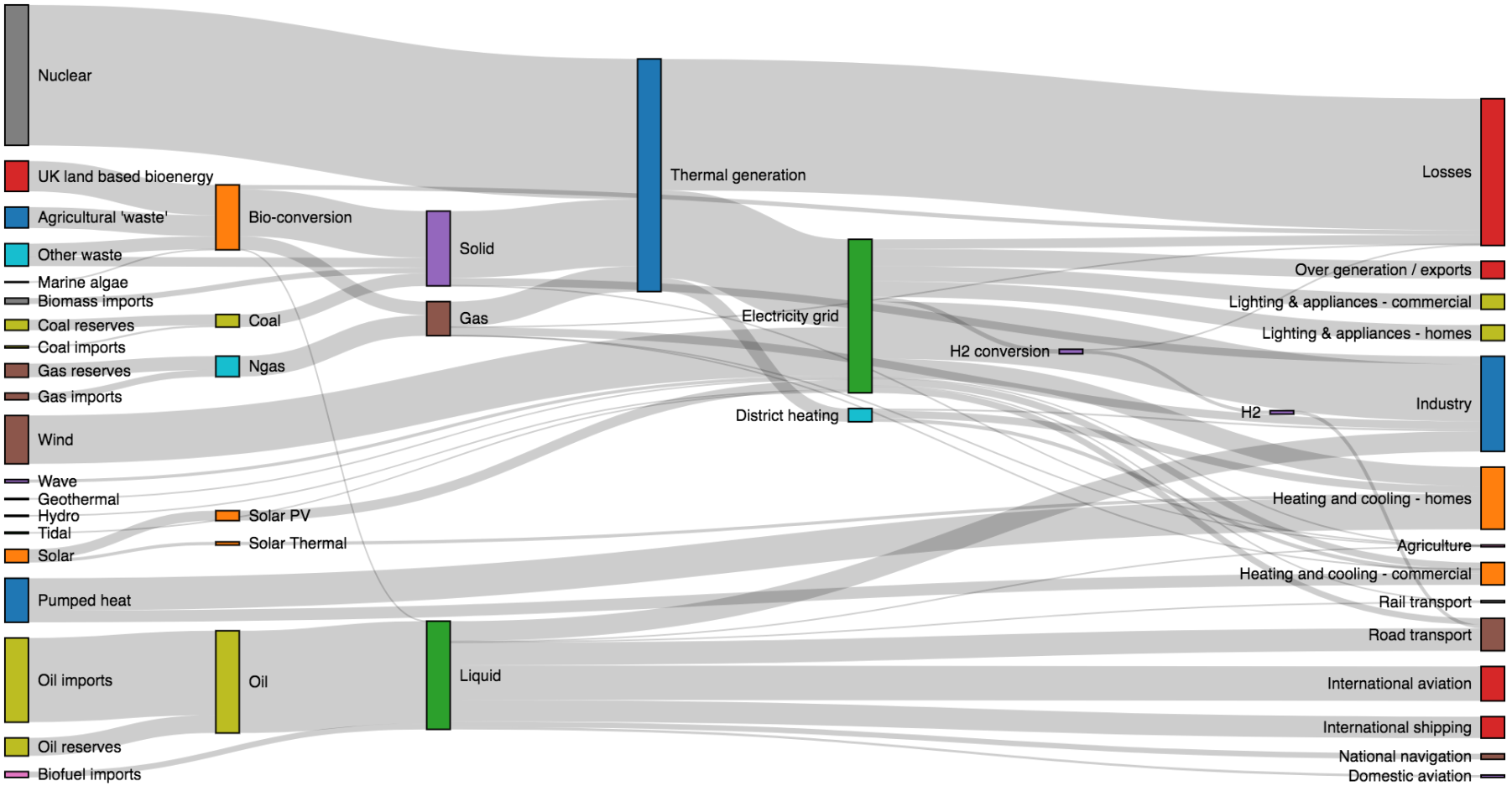
Source: Tree of life (www.evogeneao.com)

Network layouts: *Tree maps*



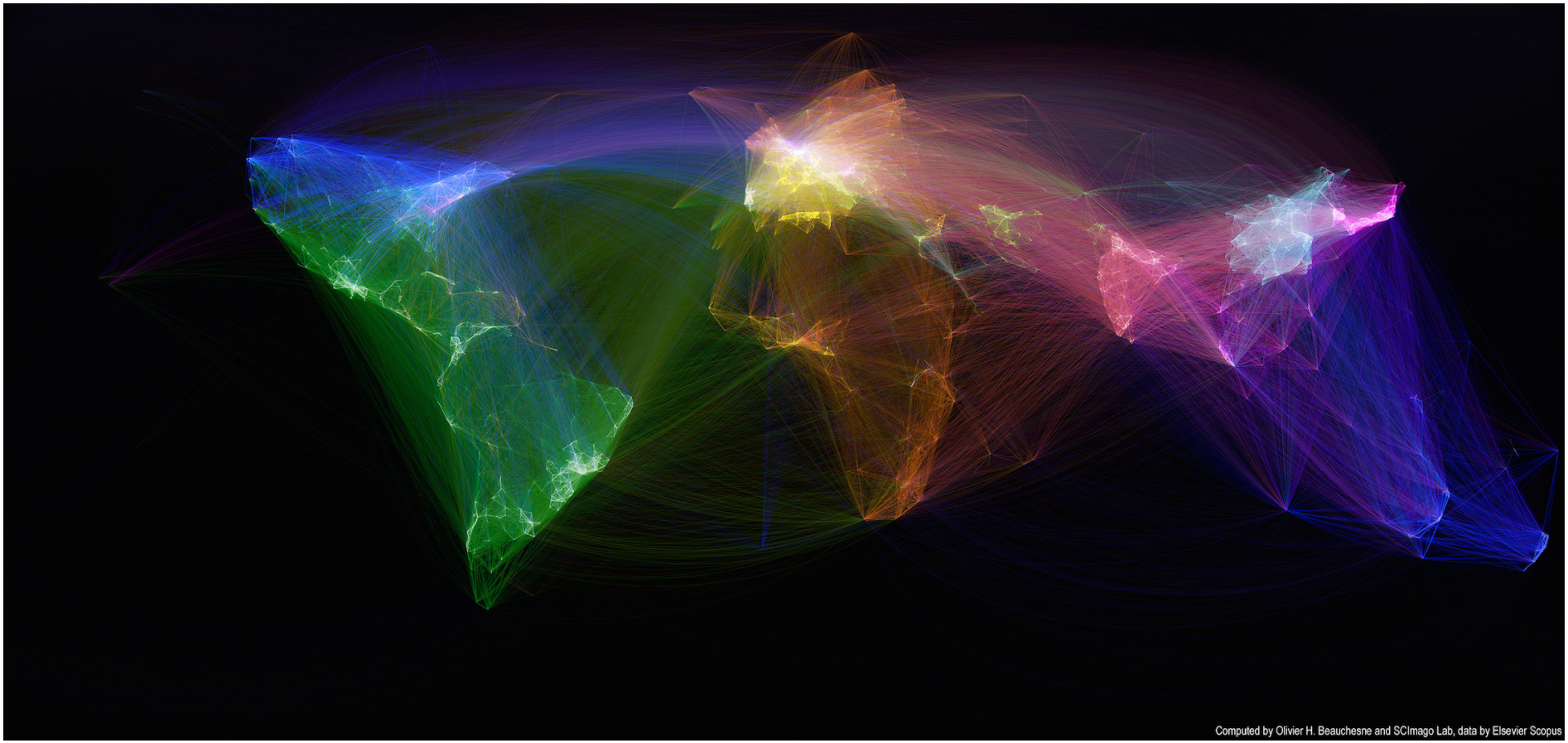
Source: https://pantheon.world

Network layouts: *Sankey diagrams*



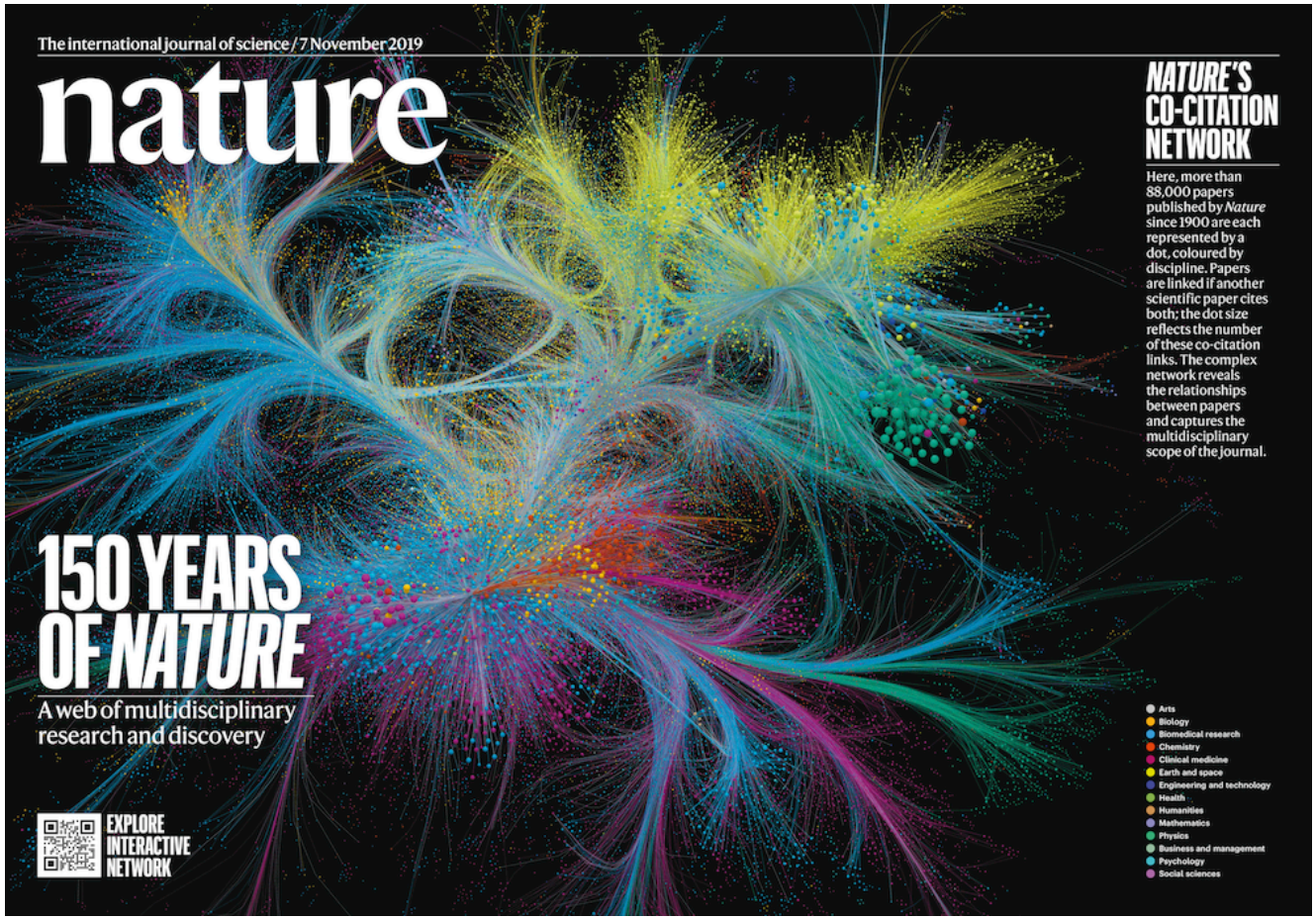
Source: UK energy production and consumption in 2050 (Department of Energy & Climate Change)

Network layouts: *Geographical networks*



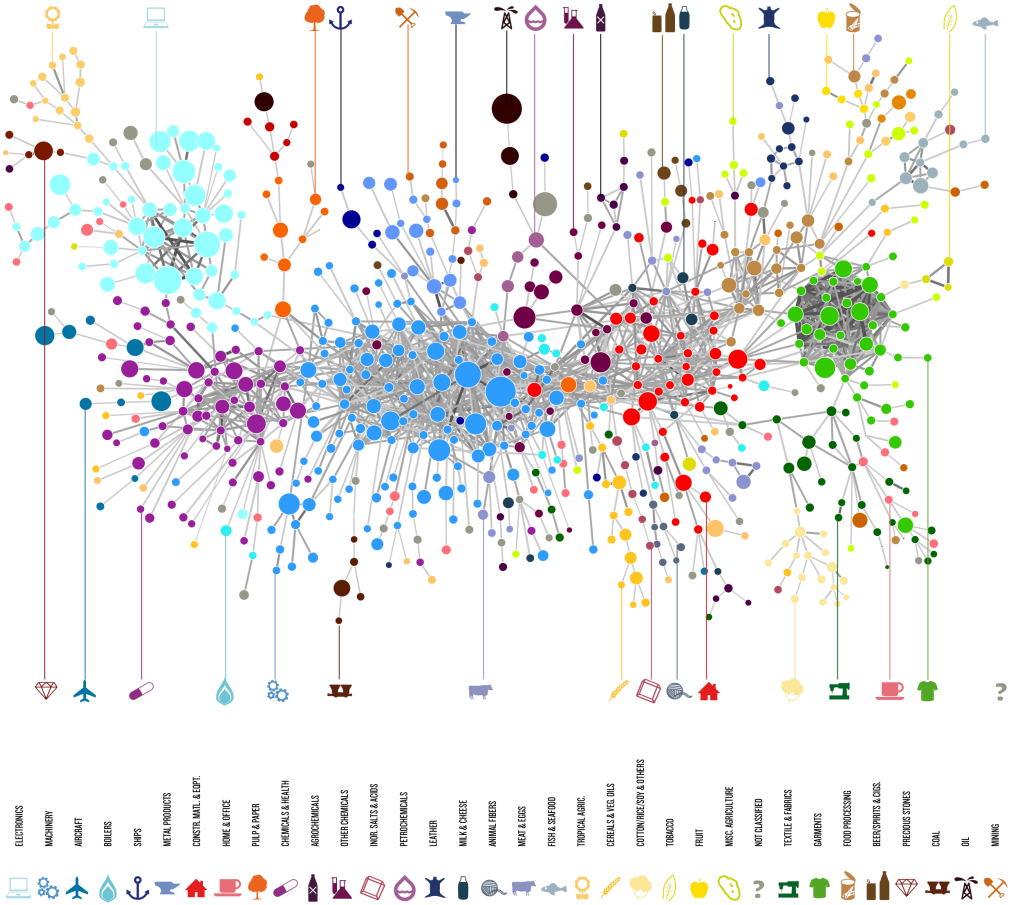
Source: Co-authorship at the city level (SCOPUS 2008-2012) (http://olihb.com)

Network layouts: *Citation networks*



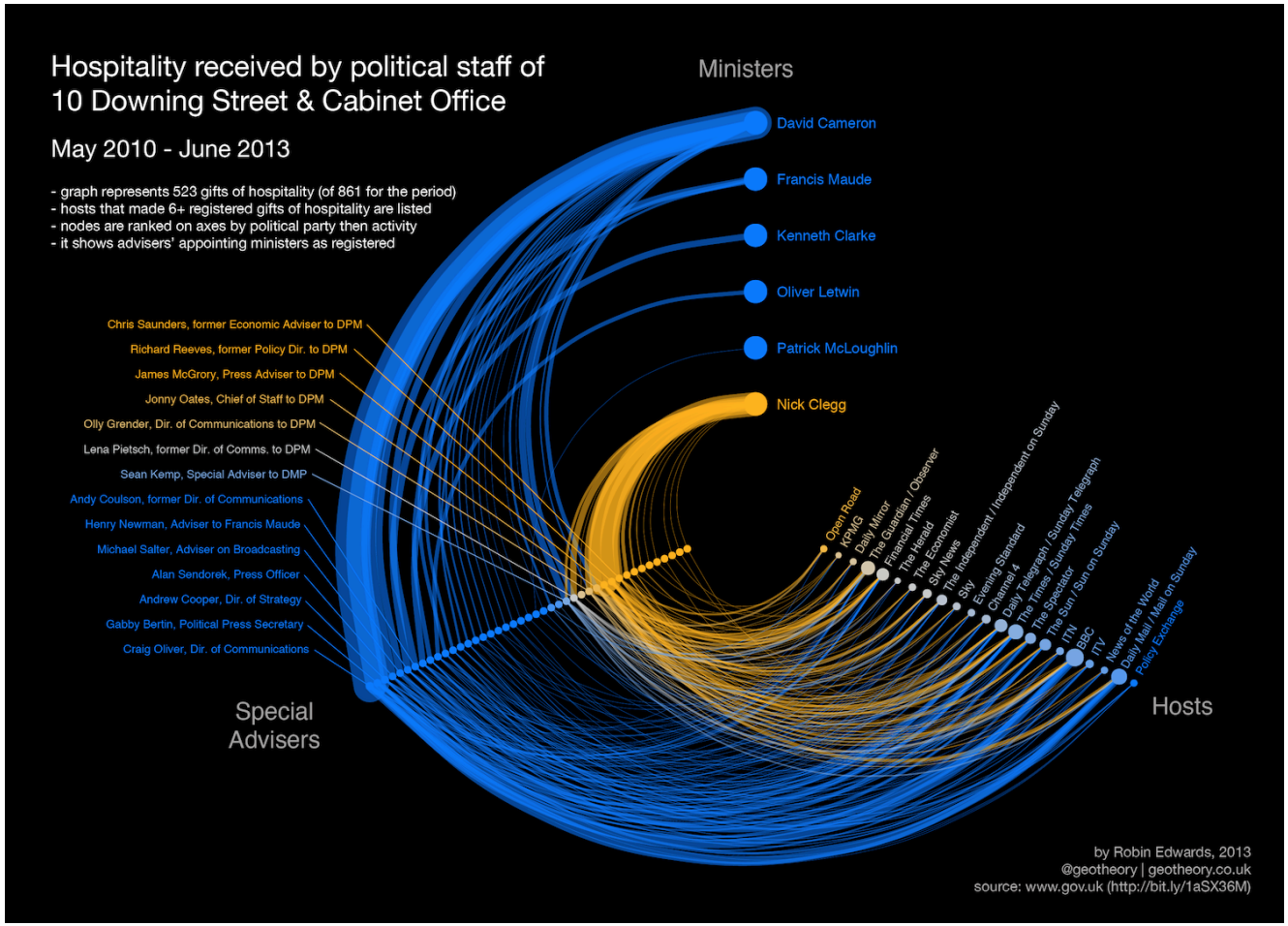
Source: Nature, November 2019

Network layouts: *Networks*



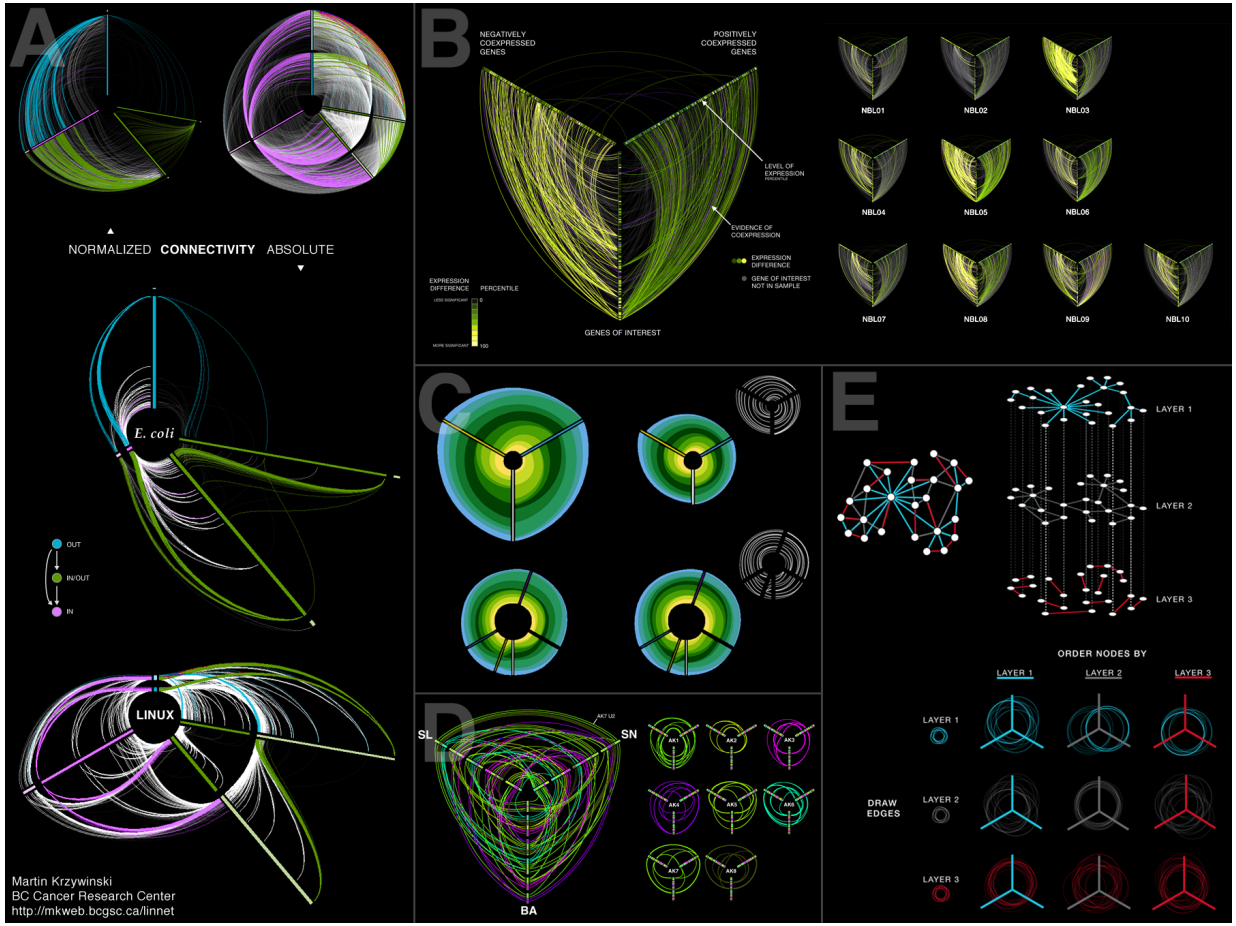
Source: Product space (Hausmann et al. 2013)

Network layouts: *Hive plots*



Source: https://www.gov.uk/government/collections/special-advisers-transparency-publications

Network layouts: *Hive plots*



Source: http://hiveplot.com

Network layouts: …

… many other variations

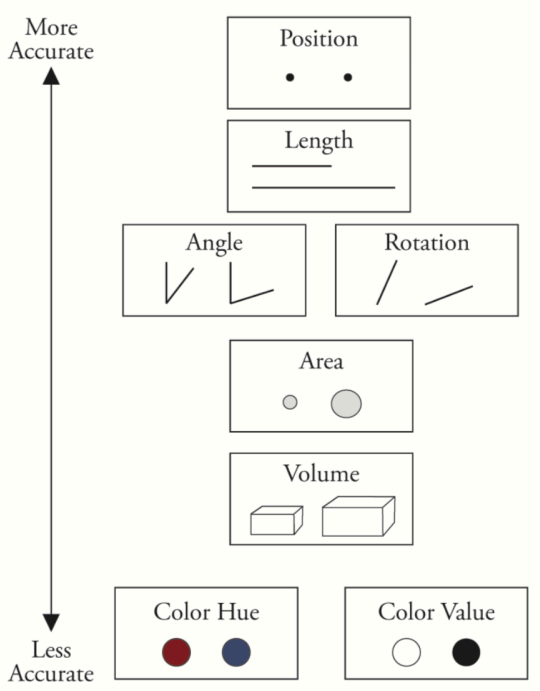
Graphical symbols and variables

Graphical symbols

- **Graphical symbols** are graphical representations of data information
- **Types**
  - *Geometric*: point, line, area, surface, volume
  - *Linguistic*: text, numerals, punctuation marks
  - *Pictorial*: images, icons, statistical glyph

Graphical variables

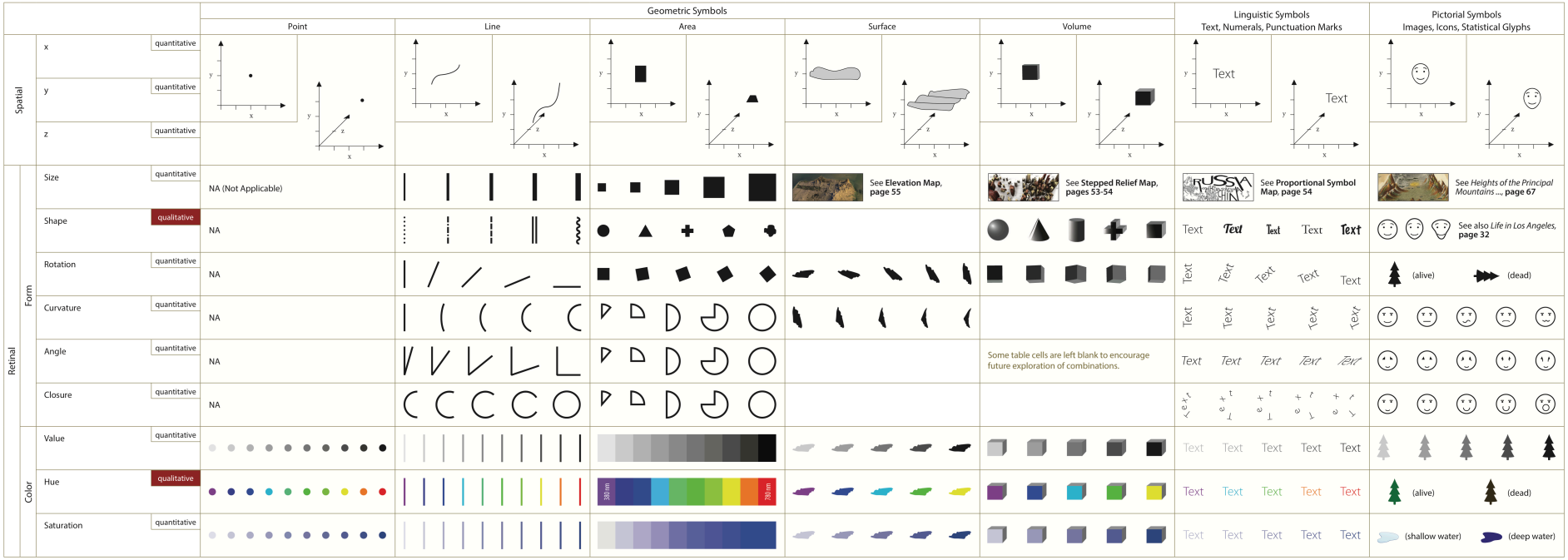
- **Graphic variables** are used to encode additional data
- **Types**
  - **Spatial:** Position (x, y , z)
  - **Retinal:**
    - *Form*: size, shape, rotation, curvature, angle, closure,
    - *Color*: value, hue, saturation (colorbrewer)
    - *Texture*: spacing, pattern, granularity, gradient
    - *Optics*: blur, transparency, shading
    - *Motion*: speed, velocity, rhythm



Source: Perception Accuracy (Börner 2016)

Graphical symbols and variables

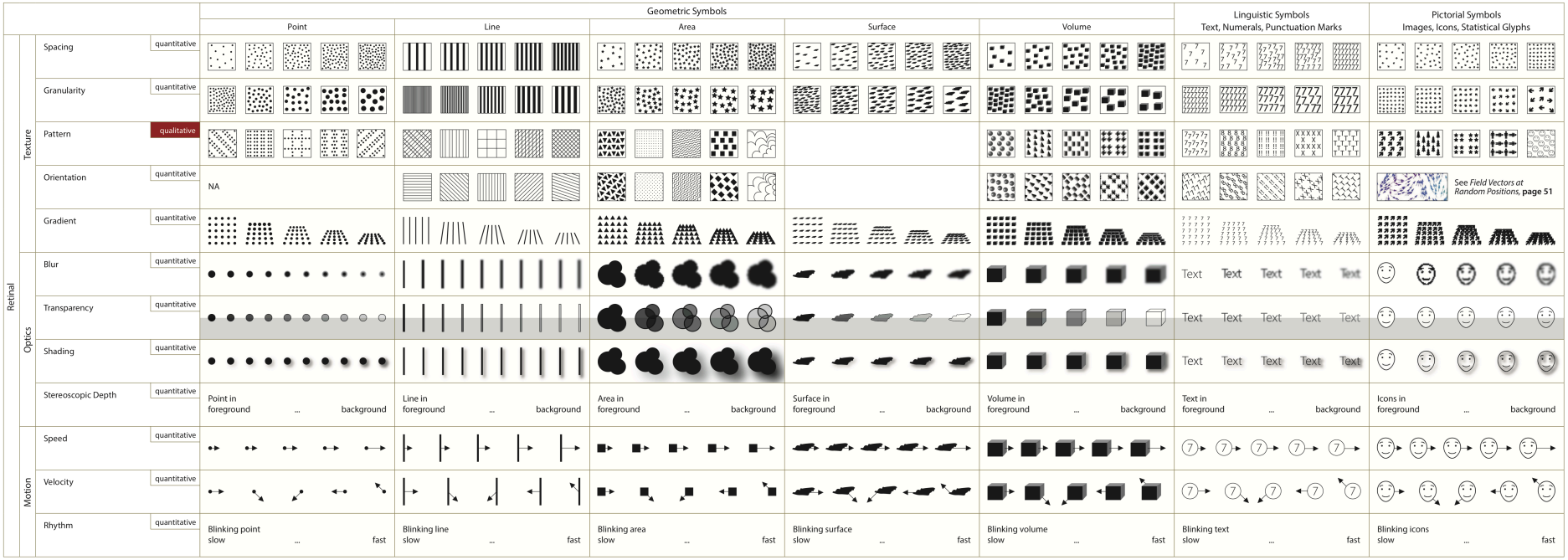
Graphic Variable Types Versus Graphic Symbol Types



Source: Atlas of Science (Börner 2016)

Graphical symbols and variables

Graphic Variable Types Versus Graphic Symbol Types (continued)

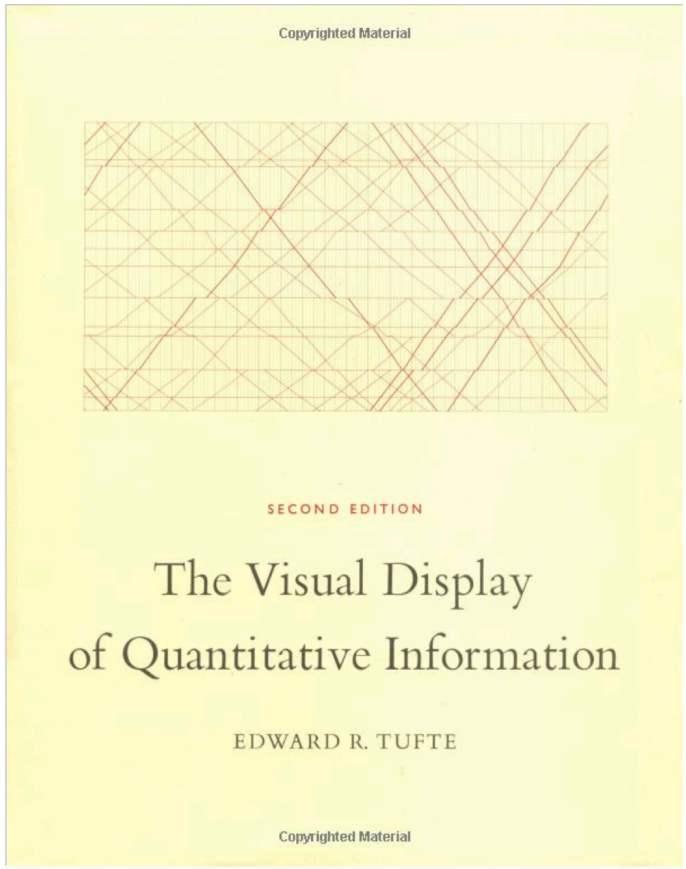


Source: Atlas of Science (Börner 2016)

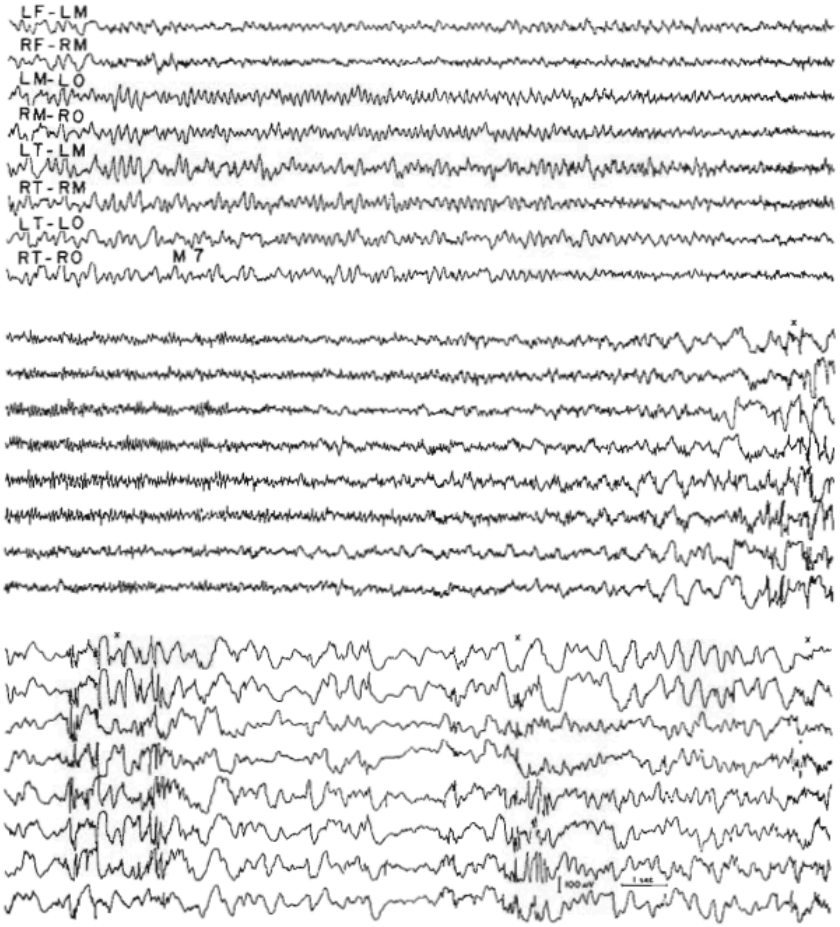
Data-ink ratio

The concept of **data-ink ration** has been introduced by Tufte (2001)

- Most of the 'ink' used to print a graphic should represent data-information
- **Data-ink** is defined as the part of the graphic that we cannot erase without losing information
- **Data-ink ratio** is defined as the proportion of a graphic's ink devoted to the display of non-redundant information (data-ink)

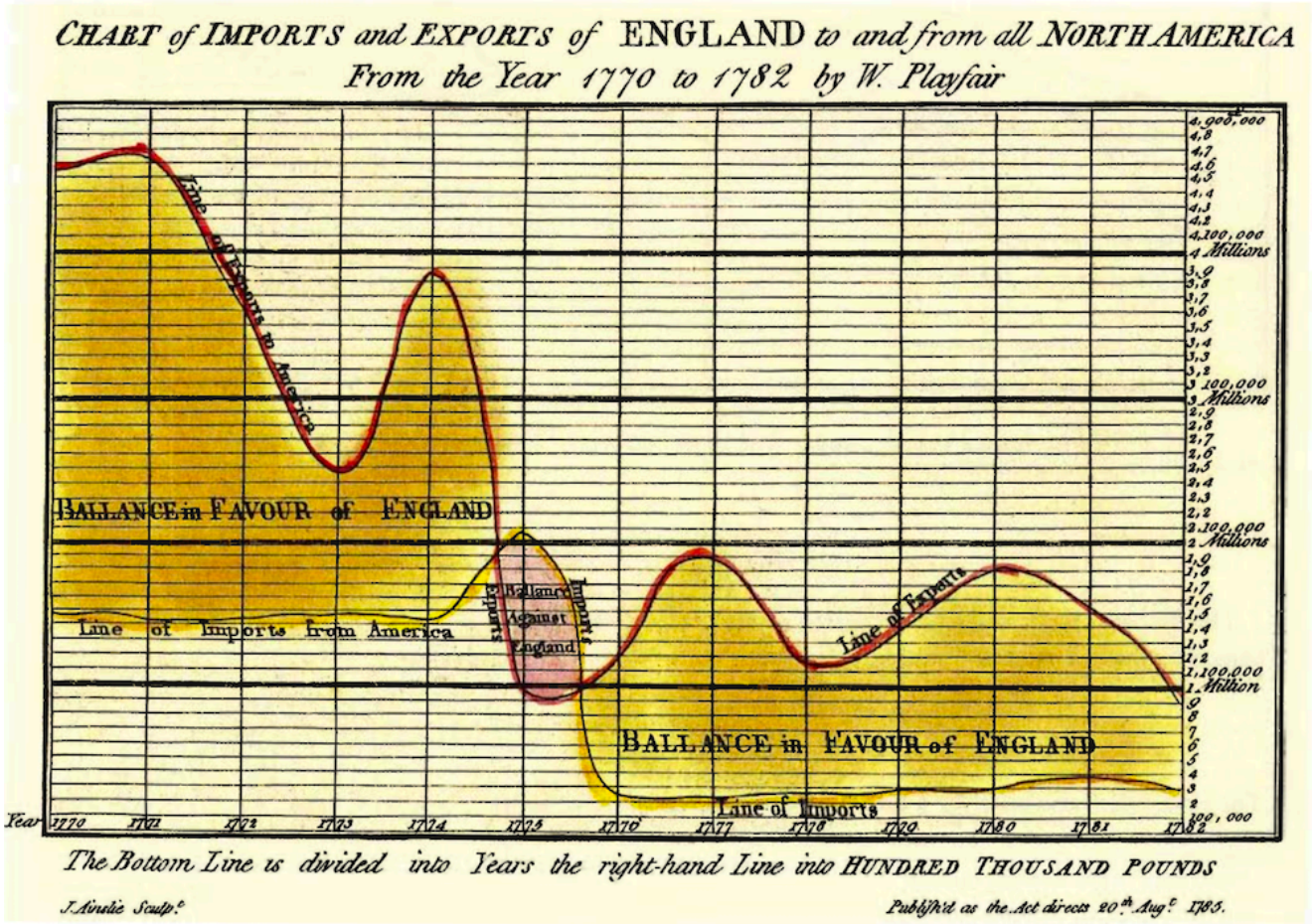


Data-ink ratio



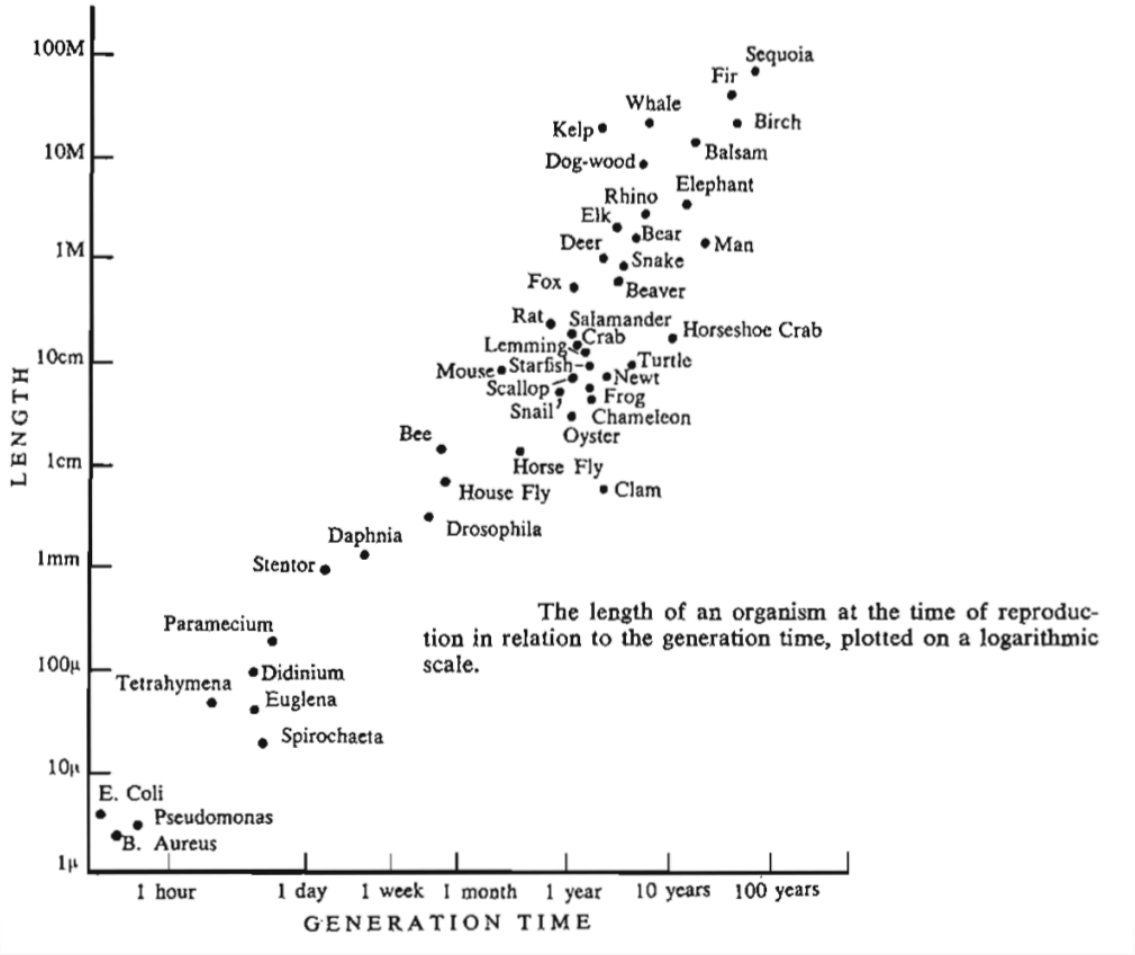
Source: Fundamentals of electroencephalography (Tufte 2001)

Data-ink ratio



Source: England's import/export from and to North America from 1770 to 1782 (Tufte 2001)

Data-ink ratio



The length of an organism at the time of reproduction in relation to the generation time, plotted on a logarithmic scale.

Source: John Tyler Bonner. 1975. Size and Cycle: An Essay on the Structure of Biology, Princeton Legacy Library (Tufte 2001)

Beyond statistics and visualisation

Triggering emotions

- **Raw data** do not tell as much
- Statistics and visualisation help us to go **beyond raw data** to see patterns, and relationships between variables
- **Data visualisation** helps us to communicate this into a story and to trigger emotions and actions
- It is crucial to consider your audience

Triggering emotions

- Watch **Hans Rosling**'s videos
- BBC: 200 years in 4 minutes
- TED talk: Global Population Growth, Box by Box
- TED talk: Asia's Rise = How and When

Source: New York Times, 9 February 2017

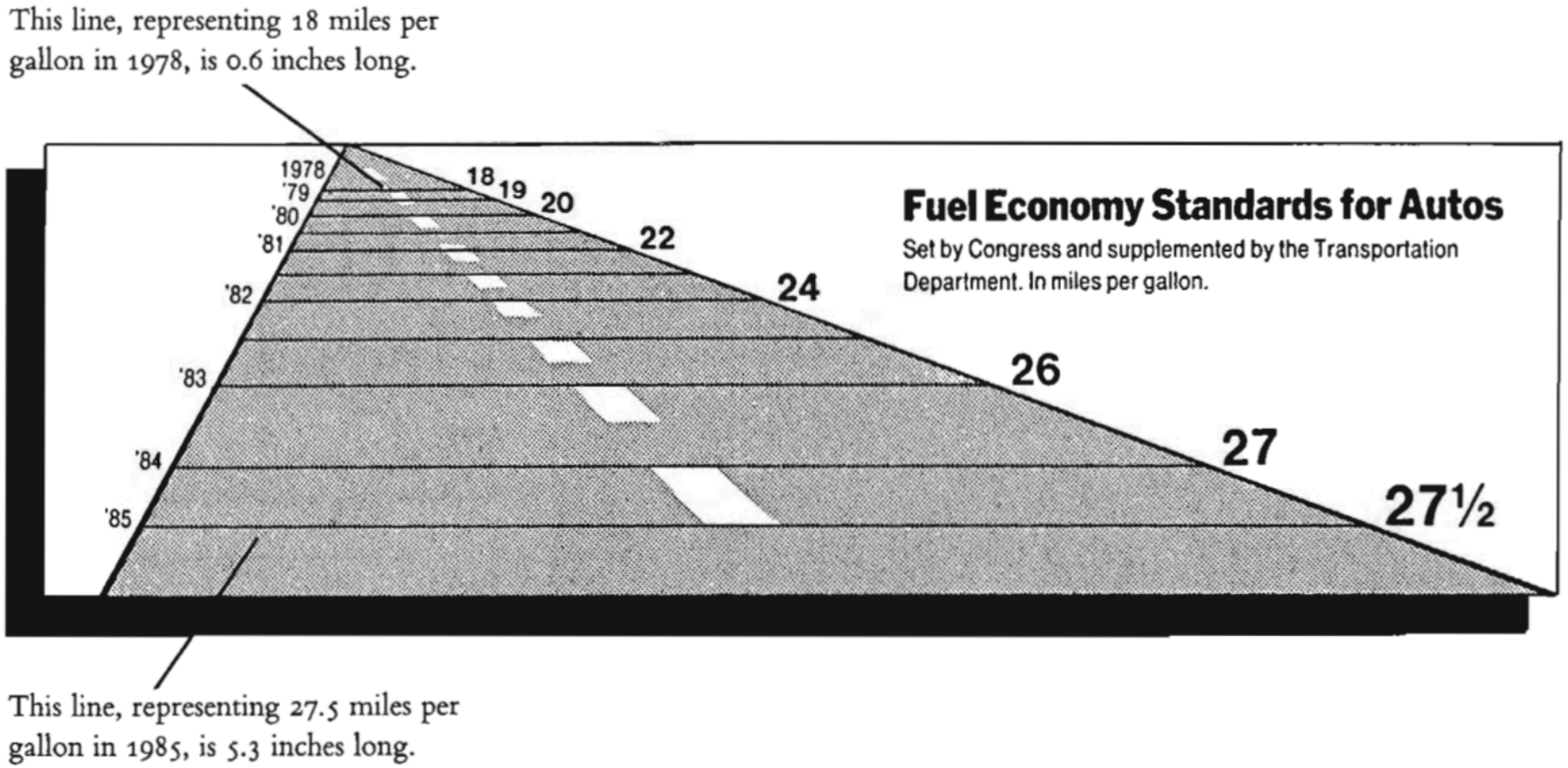
Graphical integrity

- Graphics are a form of **data communication**
- Graphics may **(intentionally or unintentionally) distort** this communication and therefore the underlying data
- The **visual perception** has to be assessed carefully

$$\text{Lie factor} = \frac{\text{Size of effect shown in graphic}}{\text{Size of effect in data}}$$

- A **lie factor** greater than 1.05 or smaller than 0.95 introduces a significant distortion

Graphical integrity



Source: New York Times, August 9, 1978 (Tufte 2001)

Graphical integrity

$$\text{Size of effect in data} = (27.5 - 18.0)/18.0 = 0.53$$

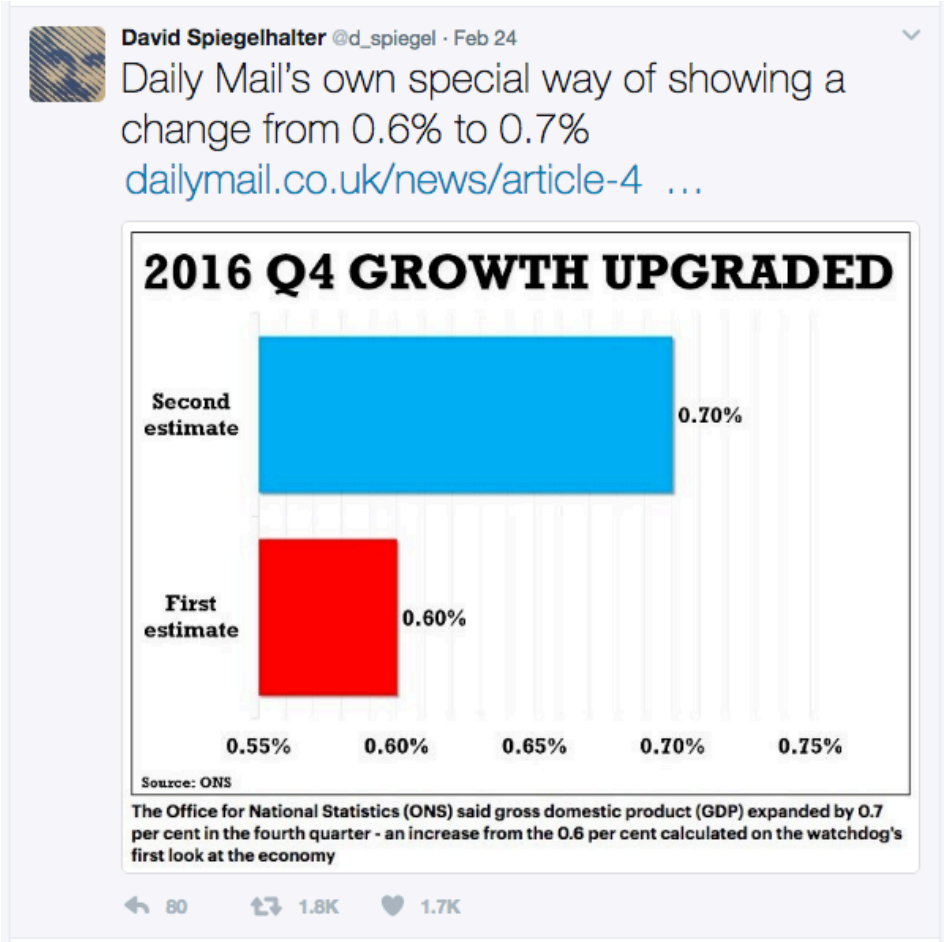
$$\text{Size of effect shown in graphic} = (5.3 - 0.6)/0.6 = 0.783$$

$$\text{Lie factor} = 0.783/0.53 = 14.8$$

Graphical integrity

Source: Campaign ad in Islington South (https://electionleaflets.org/brb.html)

Graphical integrity



Source: Twitter (March, 2017)

Data visualisation in R

Data visualisation in R

- The most popular package is `ggplot2`
- Developed by **Hadley Wickham** and colleagues
- It is based on the **Grammar of Graphics**: components that can be composed in different manners (like a LEGO)
- It works in **layers of graphics**
- Layers map data into aesthetic attributes (e.g. color, shape, size) of geometric objects (e.g. points, lines, bars), or in other words **graphical symbols and variables**

Elements of `ggplot2`

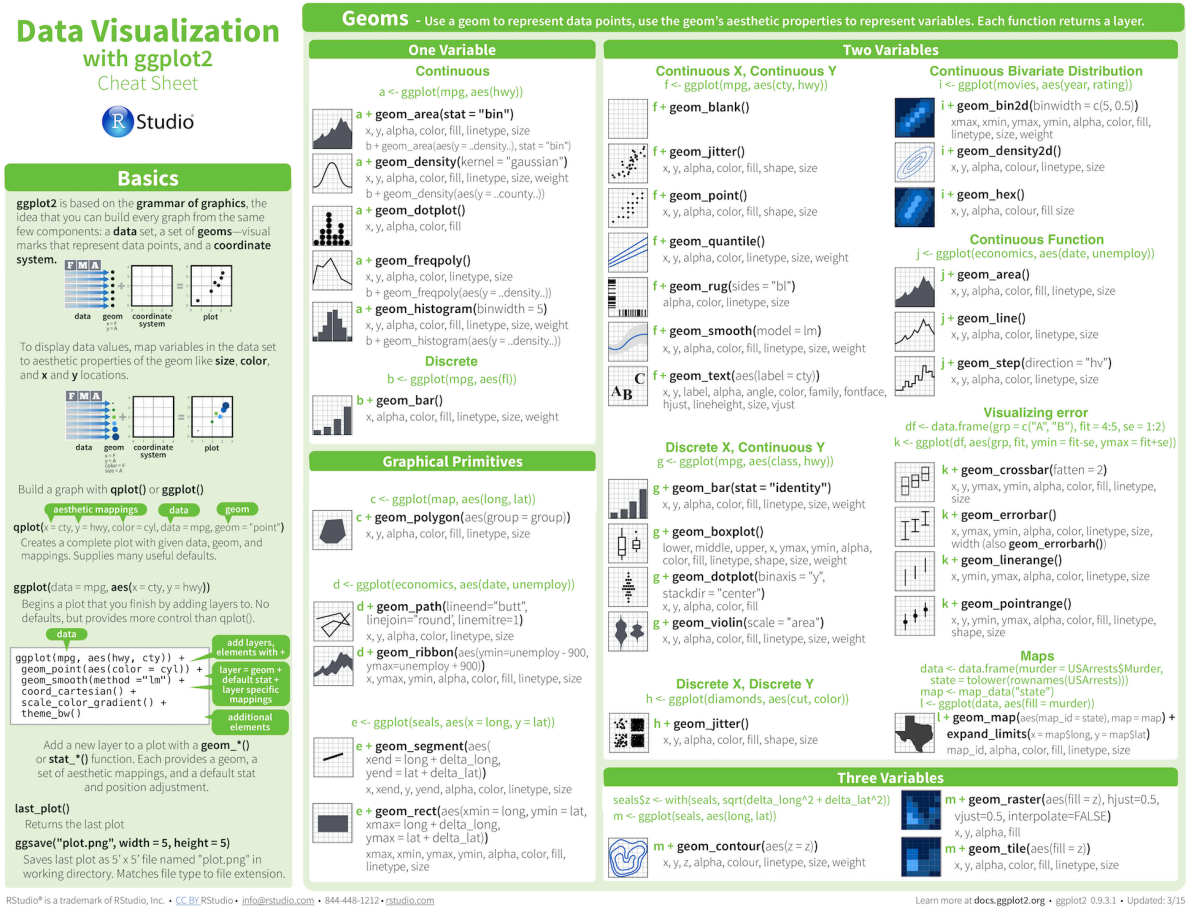
- **Data** to visualise and a set of aesthetic (`aes`) mappings (e.g. x, y, group)
- **Layers** of geometric elements (`geom`) and statistical transformation (`stat`)
- **Scales** which allow us to read the data from the plot (legend and axis)
- A system of **coordinates** (e.g. coordinates of observations, gridlines)
- **Faceting** to describe how data are separated in different subsets
- **Theme** (e.g. size, background colors)

Elements of `ggplot2`

Describes all the non-data ink	Theme
Plotting space for the data	Coordinates
Statistical models & summaries	Statistics
Rows and columns of sub-plots	Facets
Shapes used to represent the data	Geometries
Scales onto which data is mapped	Aesthetics
The actual variables to be plotted	Data

Source: https://api.rpubs.com/NemiVoraR/511391

ggplot2 cheat sheet



Data Visualization with ggplot2 Cheat Sheet

RStudio

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data** set, a set of **geoms**—visual marks that represent data points, and a **coordinate system**.

data · geom · coordinate system · plot

To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.

data · geom · coordinate system · plot

Build a graph with **qplot()** or **ggplot()**

aesthetic mappings · data · geom

```
qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point")
```

Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

```
ggplot(data = mpg, aes(x = cty, y = hwy))
```

Begins a plot that you finish by adding layers to. No defaults, but provides more control than qplot().

data · add layers, elements with + · layer = geom + default stat + layer specific mappings · additional elements

```
ggplot(mpg, aes(hwy, cty)) +
 geom_point(aes(color = cyl)) +
 geom_smooth(method ="lm") +
 coord_cartesian() +
 scale_color_gradient() +
 theme_bw()
```

Add a new layer to a plot with a **geom_*()** or **stat_*()** function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.

```
last_plot()
```

Returns the last plot

```
ggsave("plot.png", width = 5, height = 5)
```

Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

One Variable

Continuous

```
a <- ggplot(mpg, aes(hwy))
```

- a + **geom_area(stat = "bin")** x, y, alpha, color, fill, linetype, size — b + geom_area(aes(y = ..density..), stat = "bin")
- a + **geom_density(kernel = "gaussian")** x, y, alpha, color, fill, linetype, size, weight — b + geom_density(aes(y = ..county..))
- a + **geom_dotplot()** x, y, alpha, color, fill
- a + **geom_freqpoly()** x, y, alpha, color, linetype, size — b + geom_freqpoly(aes(y = ..density..))
- a + **geom_histogram(binwidth = 5)** x, y, alpha, color, fill, linetype, size, weight — b + geom_histogram(aes(y = ..density..))

Discrete

```
b <- ggplot(mpg, aes(fl))
```

- b + **geom_bar()** x, alpha, color, fill, linetype, size, weight

Graphical Primitives

```
c <- ggplot(map, aes(long, lat))
```

- c + **geom_polygon**(aes(group = group)) x, y, alpha, color, fill, linetype, size

```
d <- ggplot(economics, aes(date, unemploy))
```

- d + **geom_path**(lineend="butt", linejoin="round", linemitre=1) x, y, alpha, color, linetype, size
- d + **geom_ribbon**(aes(ymin=unemploy - 900, ymax=unemploy + 900)) x, ymax, ymin, alpha, color, fill, linetype, size

```
e <- ggplot(seals, aes(x = long, y = lat))
```

- e + **geom_segment**(aes(xend = long + delta_long, yend = lat + delta_lat)) x, xend, y, yend, alpha, color, linetype, size
- e + **geom_rect**(aes(xmin = long, ymin = lat, xmax= long + delta_long, ymax = lat + delta_lat)) xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

Two Variables

Continuous X, Continuous Y

```
f <- ggplot(mpg, aes(cty, hwy))
```

- f + **geom_blank()**
- f + **geom_jitter()** x, y, alpha, color, fill, shape, size
- f + **geom_point()** x, y, alpha, color, fill, shape, size
- f + **geom_quantile()** x, y, alpha, color, linetype, size, weight
- f + **geom_rug(sides = "bl")** alpha, color, linetype, size
- f + **geom_smooth(model = lm)** x, y, alpha, color, fill, linetype, size, weight
- f + **geom_text(aes(label = cty))** x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Discrete X, Continuous Y

```
g <- ggplot(mpg, aes(class, hwy))
```

- g + **geom_bar(stat = "identity")** x, y, alpha, color, fill, linetype, size, weight
- g + **geom_boxplot()** lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight
- g + **geom_dotplot(binaxis = "y", stackdir = "center")** x, y, alpha, color, fill
- g + **geom_violin(scale = "area")** x, y, alpha, color, fill, linetype, size, weight

Discrete X, Discrete Y

```
h <- ggplot(diamonds, aes(cut, color))
```

- h + **geom_jitter()** x, y, alpha, color, fill, shape, size

Continuous Bivariate Distribution

```
i <- ggplot(movies, aes(year, rating))
```

- i + **geom_bin2d(binwidth = c(5, 0.5))** xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight
- i + **geom_density2d()** x, y, alpha, colour, linetype, size
- i + **geom_hex()** x, y, alpha, colour, fill size

Continuous Function

```
j <- ggplot(economics, aes(date, unemploy))
```

- j + **geom_area()** x, y, alpha, color, fill, linetype, size
- j + **geom_line()** x, y, alpha, color, linetype, size
- j + **geom_step(direction = "hv")** x, y, alpha, color, linetype, size

Visualizing error

```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))
```

- k + **geom_crossbar(fatten = 2)** x, y, ymax, ymin, alpha, color, fill, linetype, size
- k + **geom_errorbar()** x, ymax, ymin, alpha, color, linetype, size, width (also **geom_errorbarh()**)
- k + **geom_linerange()** x, ymin, ymax, alpha, color, linetype, size
- k + **geom_pointrange()** x, y, ymin, ymax, alpha, color, fill, linetype, shape, size

Maps

```
data <- data.frame(murder = USArrests$Murder,
state = tolower(rownames(USArrests)))
map <- map_data("state")
l <- ggplot(data, aes(fill = murder))
```

- l + **geom_map**(aes(map_id = state), map = map) + **expand_limits**(x = map$long, y = map$lat) map_id, alpha, color, fill, linetype, size

Three Variables

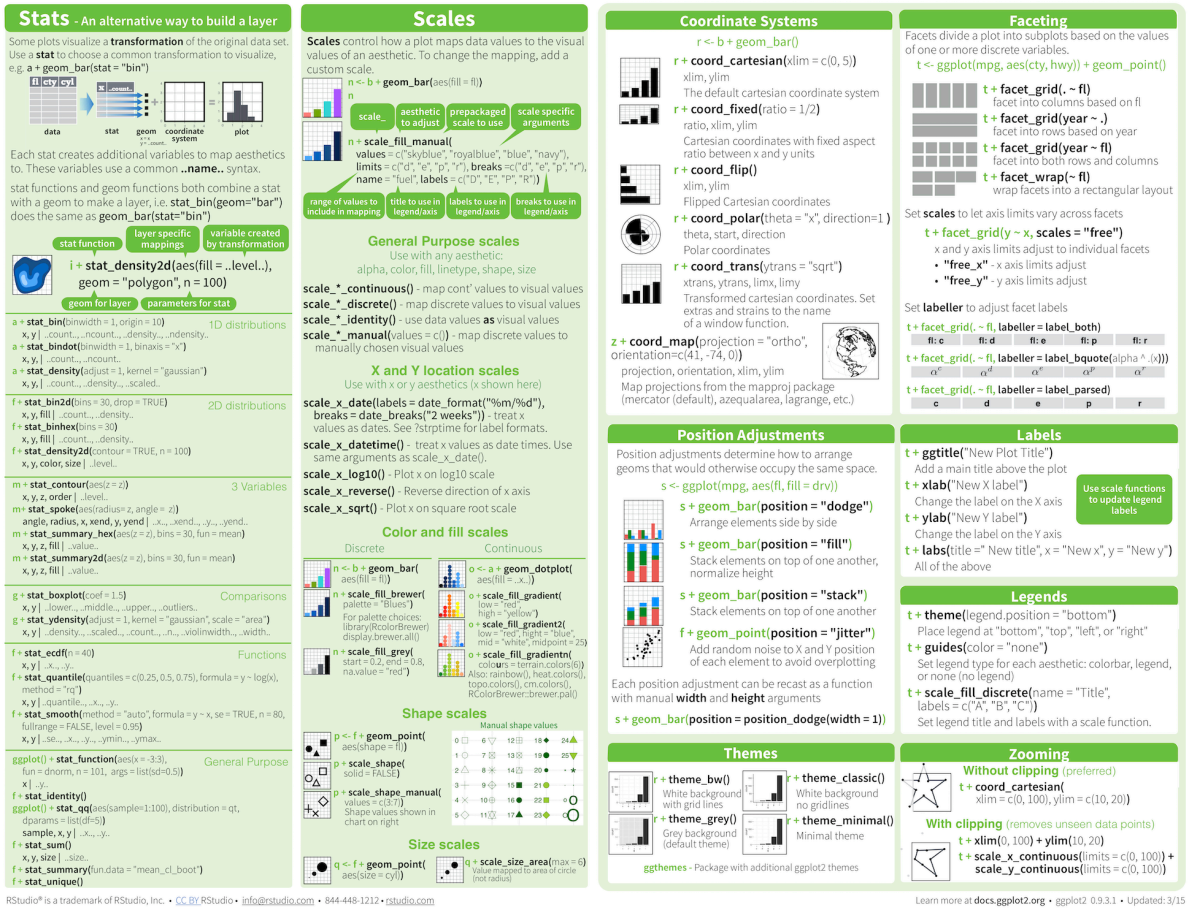
```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))
m <- ggplot(seals, aes(long, lat))
```

- m + **geom_contour**(aes(z = z)) x, y, z, alpha, colour, linetype, size, weight
- m + **geom_raster**(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE) x, y, alpha, fill
- m + **geom_tile**(aes(fill = z)) x, y, alpha, color, fill, linetype, size

Learn more at docs.ggplot2.org • ggplot2 0.9.3.1 • Updated: 3/15

Source: https://rstudio.com/resources/cheatsheets

ggplot2 cheat sheet



Source: https://rstudio.com/resources/cheatsheets

`ggplot2` in practice

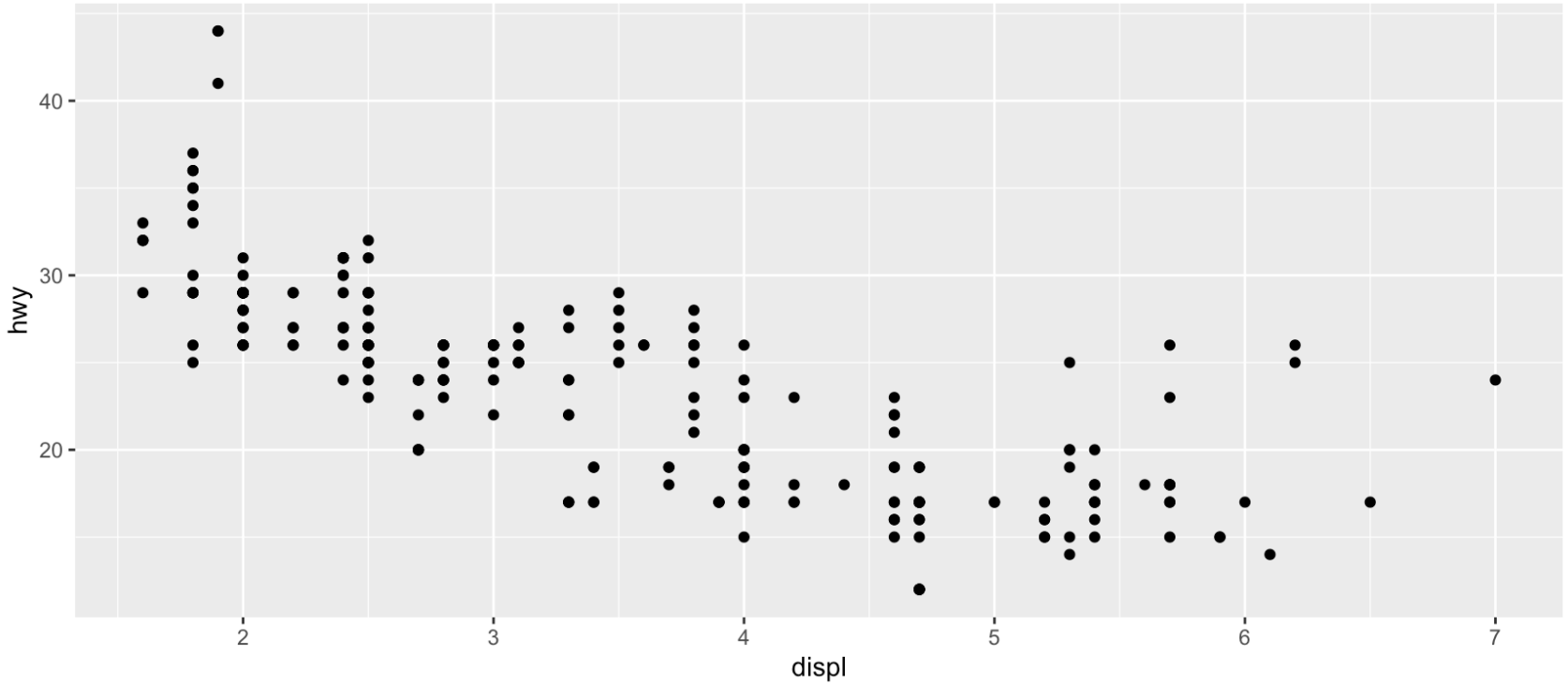
- We focus on the `mpg` dataset: `cty` (mpg for driving), `hwy` (mpg for highway driving), `displ` (engine displacement in liters), `drv` (drivetrain: front wheel [f], rear wheel [r] or four wheel [4]), `model` (car model), `class` (two seater, SUV, compact, etc.)

```
library(ggplot2)
print(mpg, n = 4)
```

```
## # A tibble: 234 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa…
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa…
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa…
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa…
## # … with 230 more rows
```

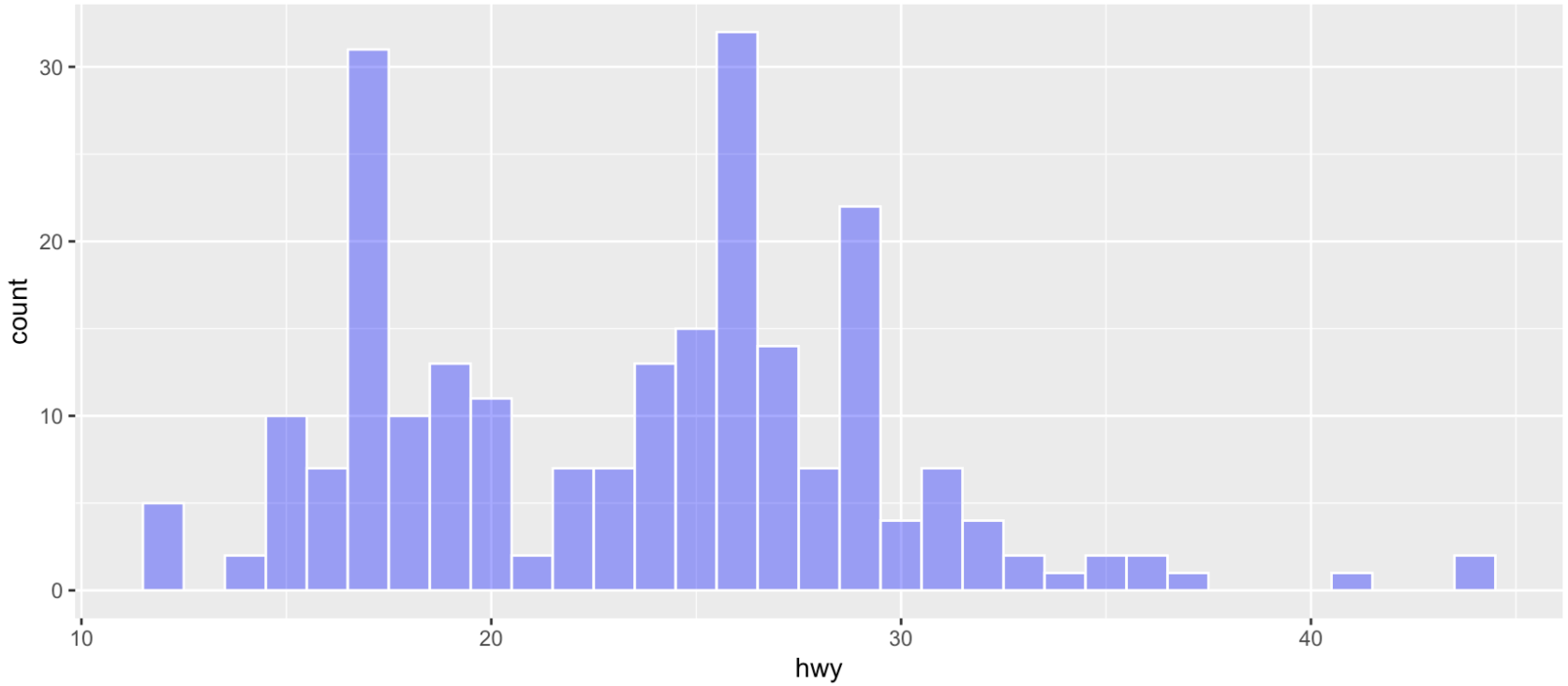
ggplot2 in practice: one variable

```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point()
```



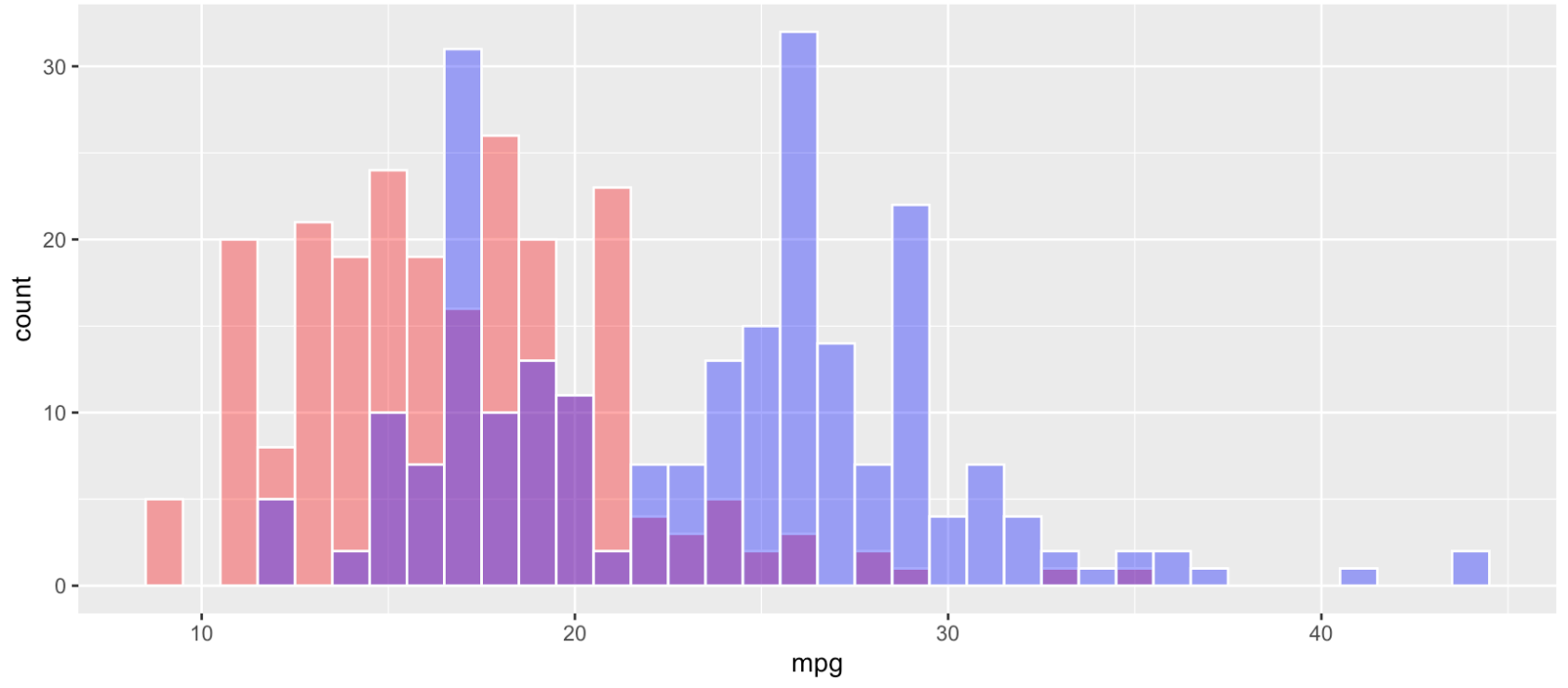
`ggplot2` in practice: one variable

```
ggplot(data = mpg, aes(hwy)) +
  geom_histogram(binwidth = 1, color = "white", fill = "blue", alpha = 0.4)
```



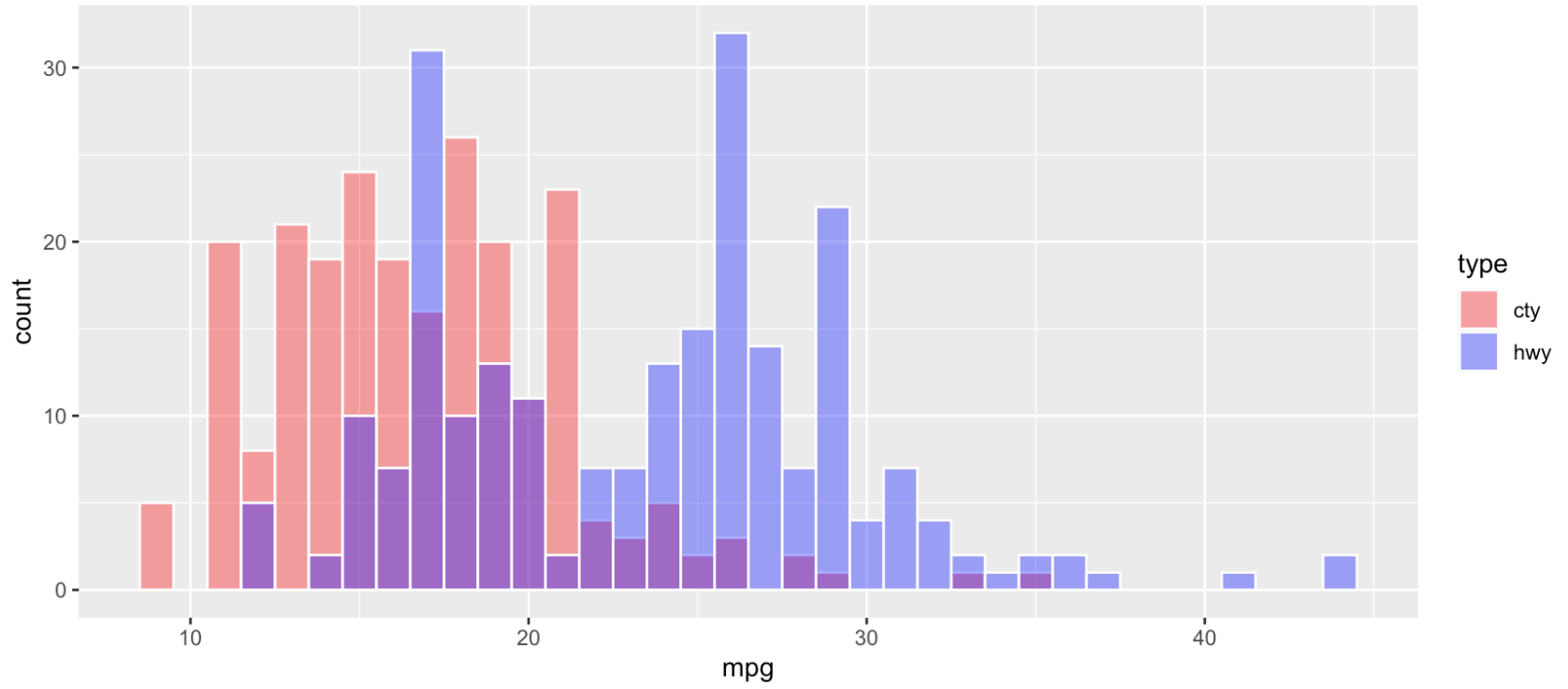
ggplot2 in practice: one variable

```
ggplot(data = mpg) +
  geom_histogram(aes(cty), binwidth = 1, fill = "red", color = "white", alpha = 0.4) +
  geom_histogram(aes(hwy), binwidth = 1, fill = "blue", color = "white", alpha = 0.4) +
  xlab("mpg")
```



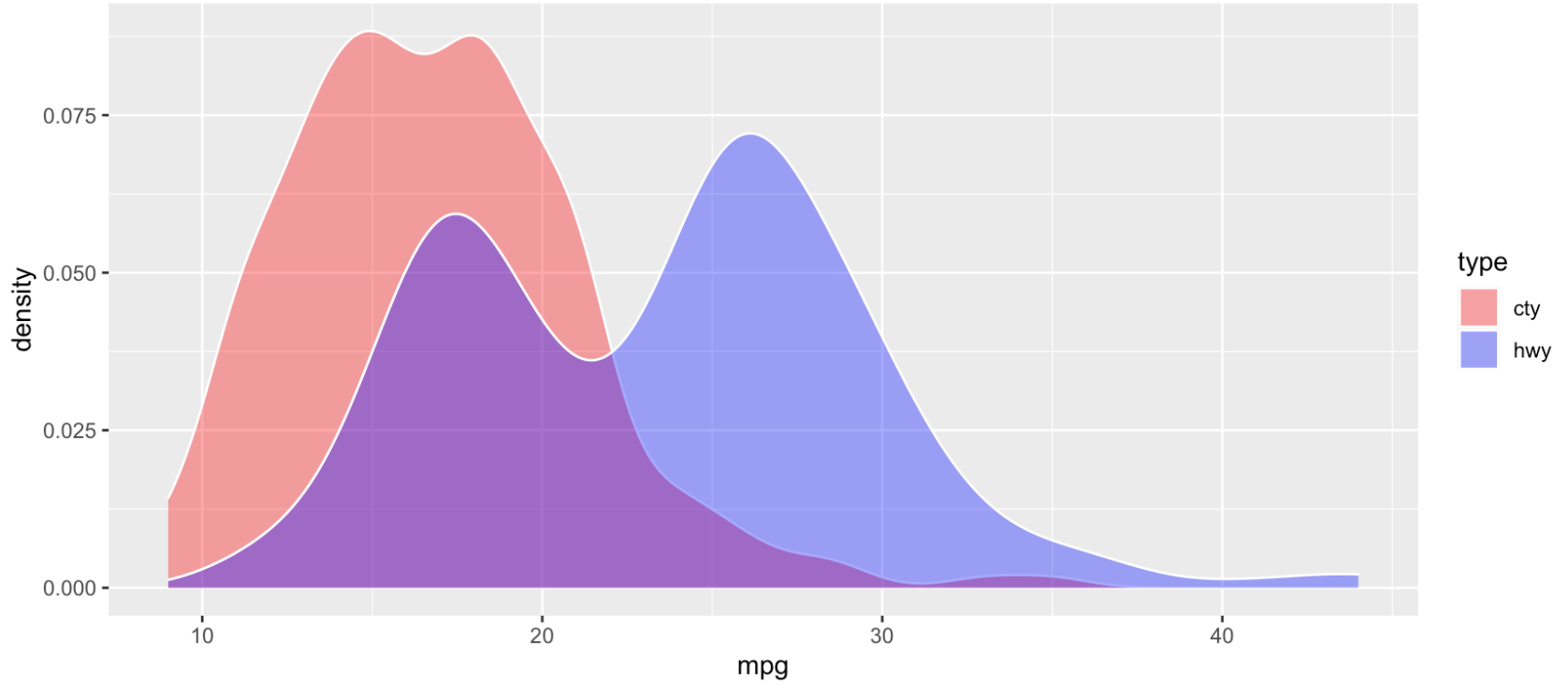
ggplot2 in practice: one variable

```
mpg %>%
  select(model, hwy, cty) %>%  pivot_longer(-model, names_to = "type", values_to = "mpg") %>%
  ggplot() +
  geom_histogram(aes(mpg, fill = type), color = "white", position = "identity", binwidth = 1, alpha = 0.4) +
  scale_fill_manual(values=c("red", "blue"))
```



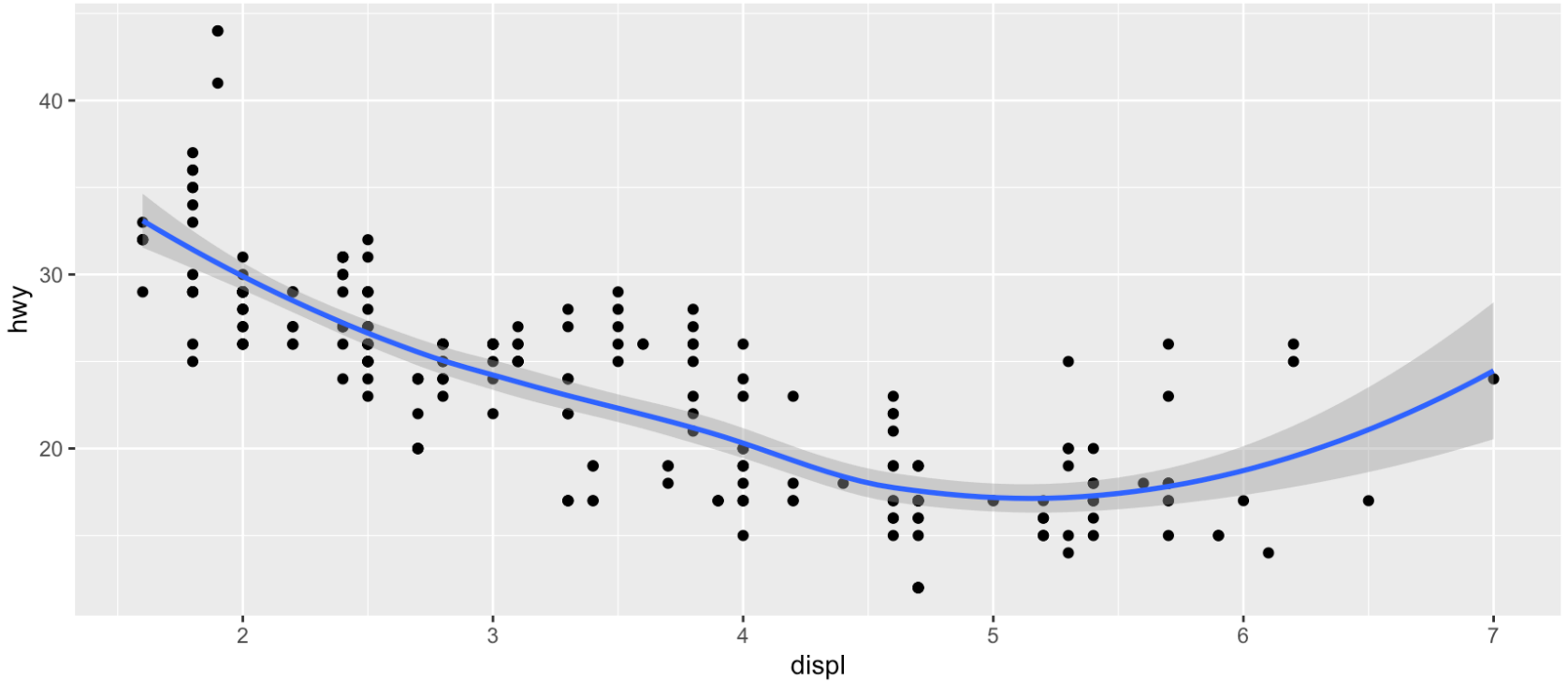
`ggplot2` in practice: one variable

```
mpg %>%
  select(model, hwy, cty) %>%  pivot_longer(-model, names_to = "type", values_to = "mpg") %>%
  ggplot() +
  geom_density(aes(mpg, fill = type), color = "white", position = "identity", alpha = 0.4) +
  scale_fill_manual(values=c("red", "blue"))
```



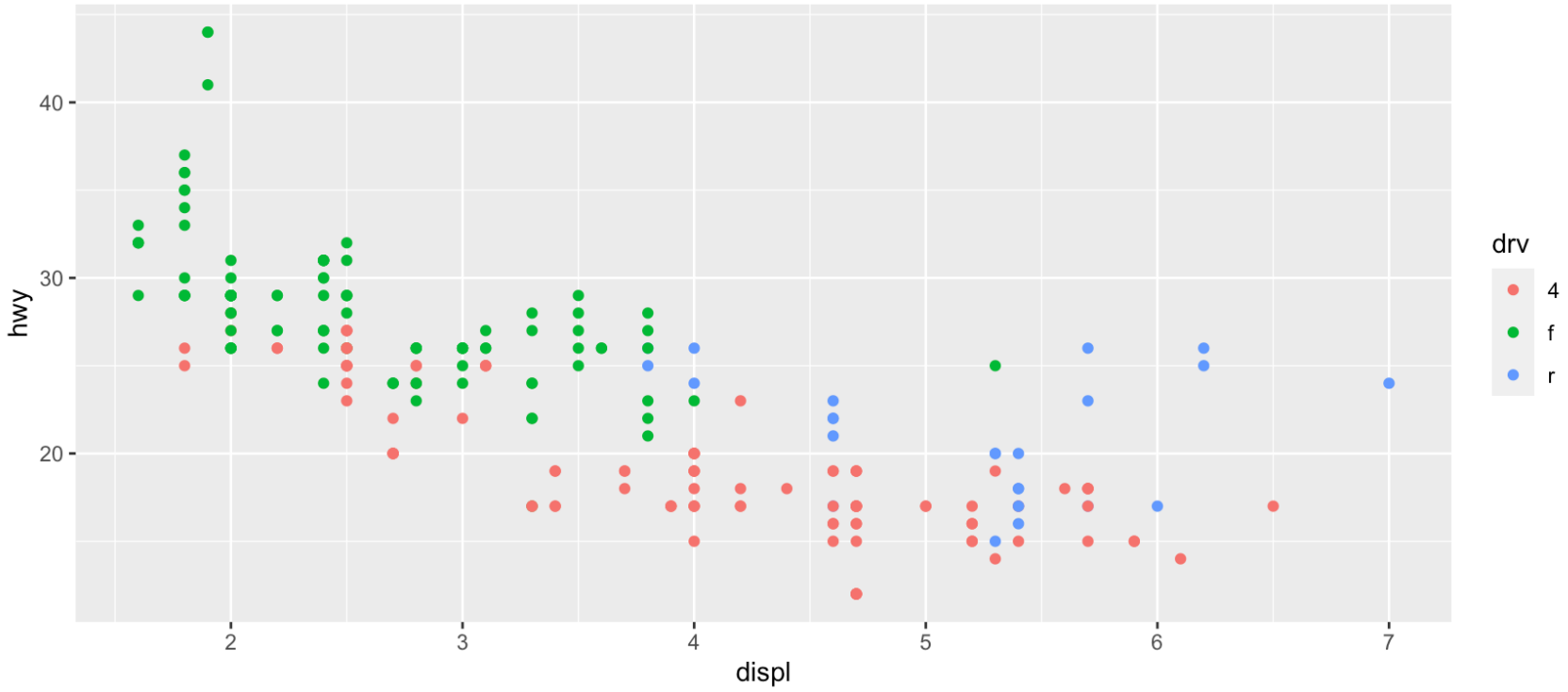
`ggplot2` in practice: two variables

```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
```



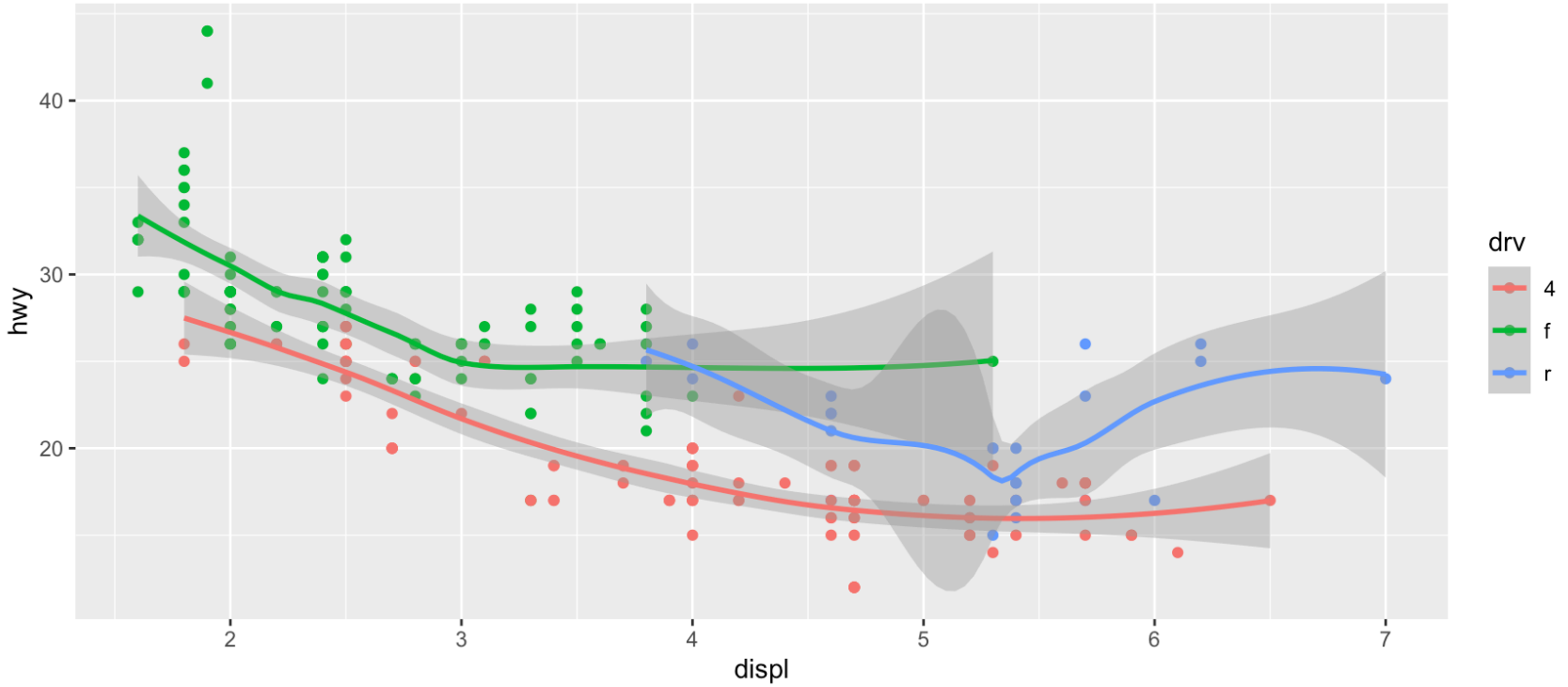
`ggplot2` in practice: two variables

```
ggplot(data = mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point()
```



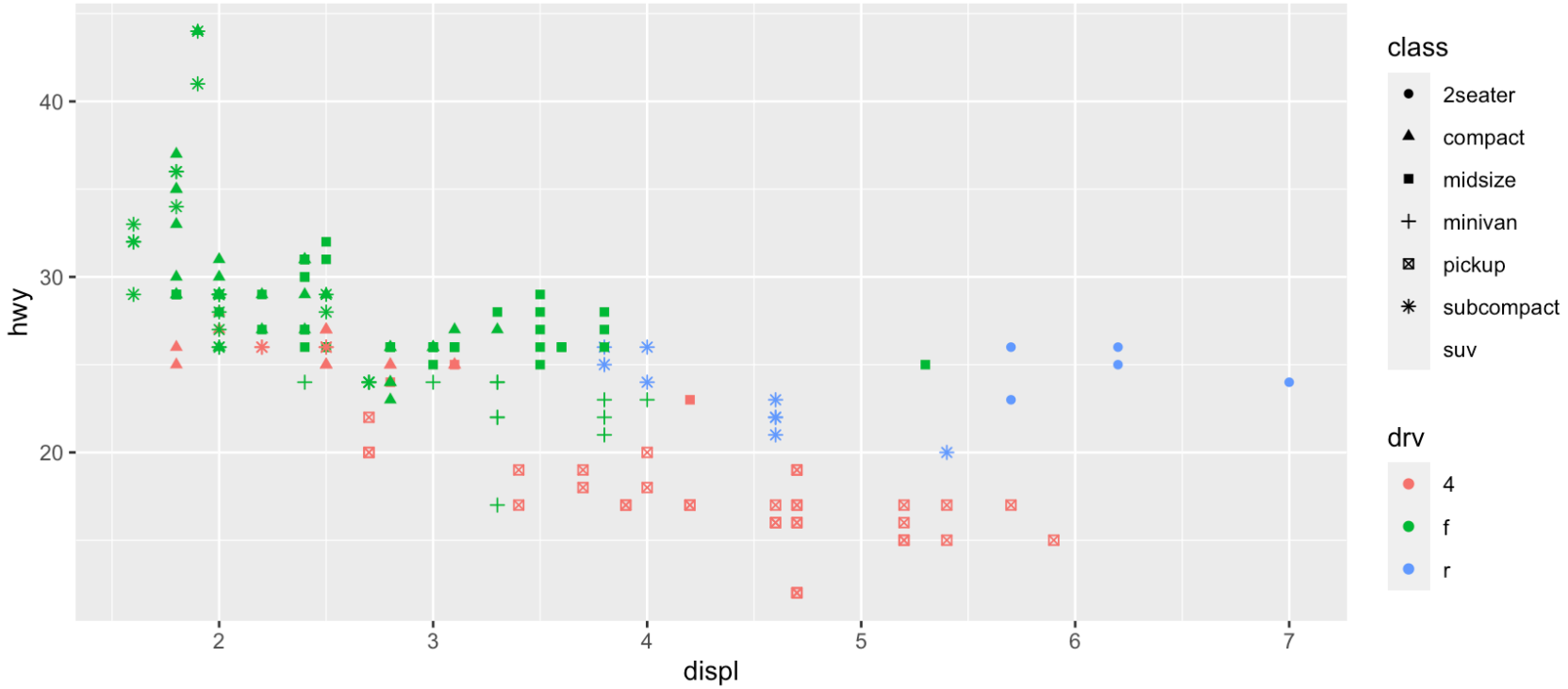
`ggplot2` in practice: two variables

```
ggplot(data = mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point() +
  geom_smooth()
```



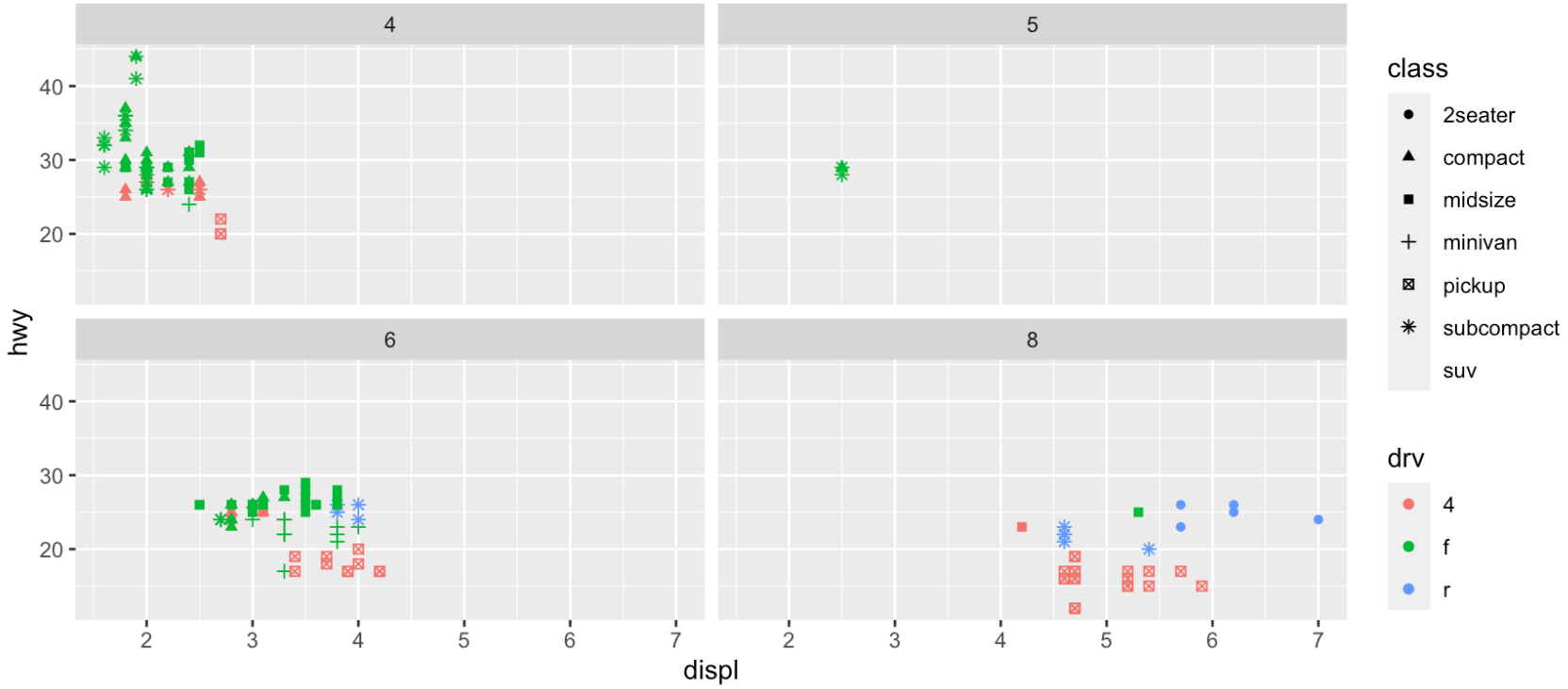
`ggplot2` in practice: two variables

```
ggplot(data = mpg, aes(x = displ, y = hwy, color = drv, shape = class)) +
  geom_point()
```



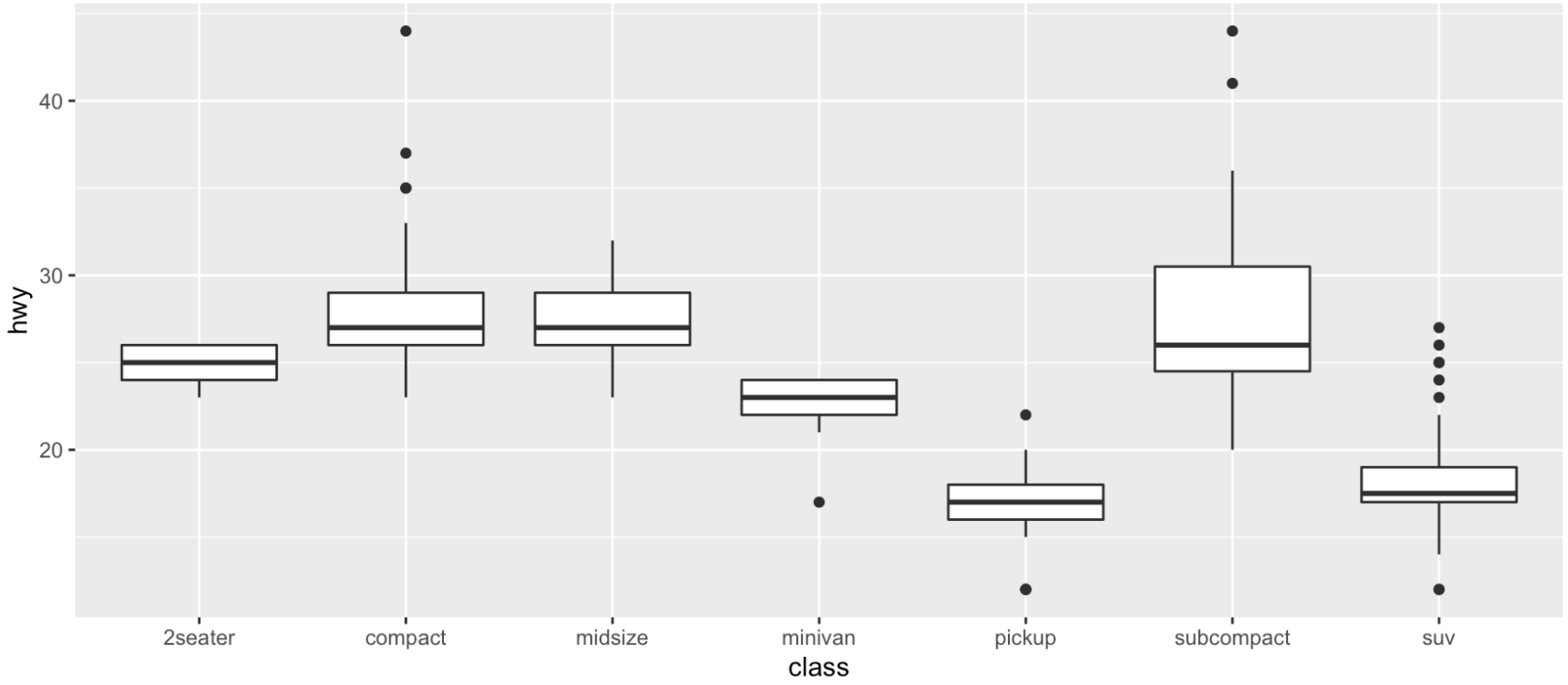
`ggplot2` in practice: two variables

```
ggplot(data = mpg, aes(x = displ, y = hwy, color = drv, shape = class)) +
  geom_point() +
  facet_wrap(~cyl)
```



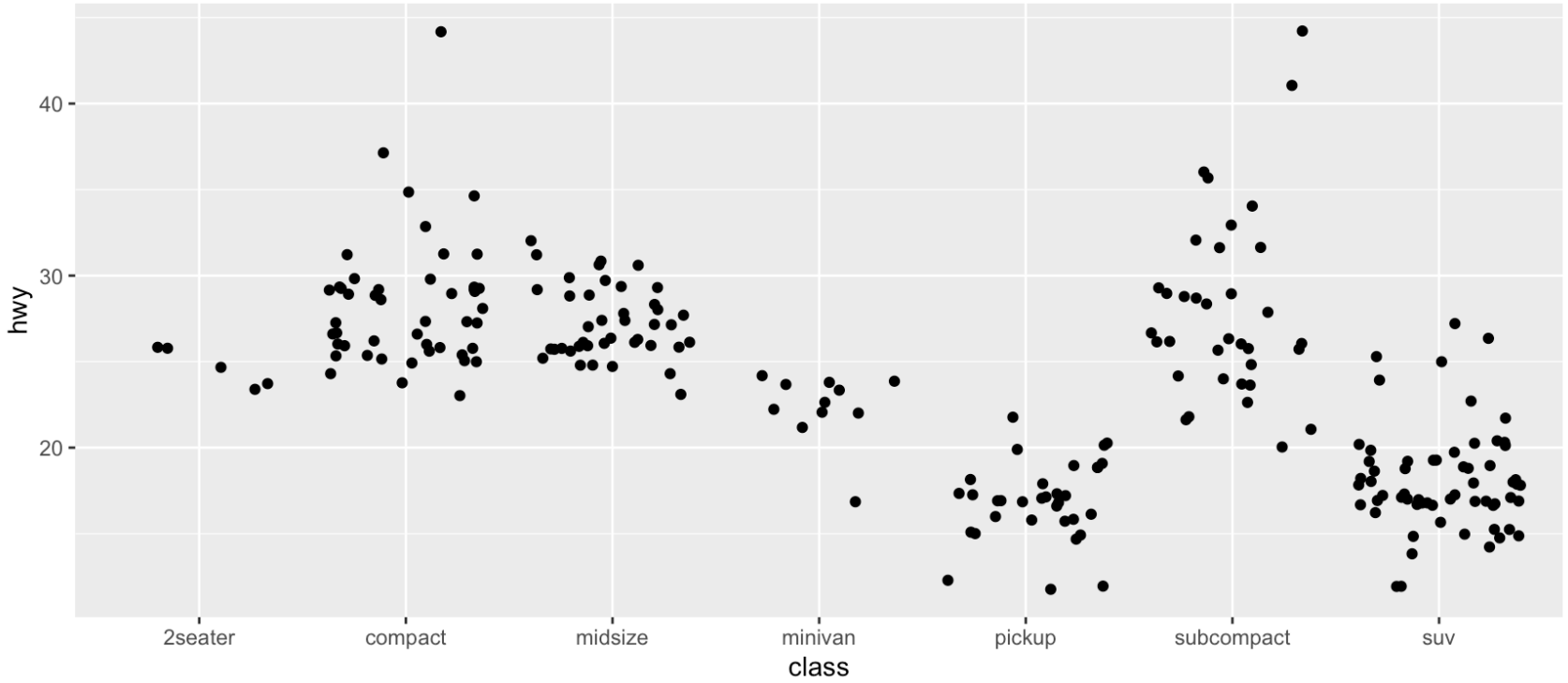
`ggplot2` in practice: two variables

```
ggplot(data = mpg, aes(x = class, y = hwy)) +
  geom_boxplot()
```



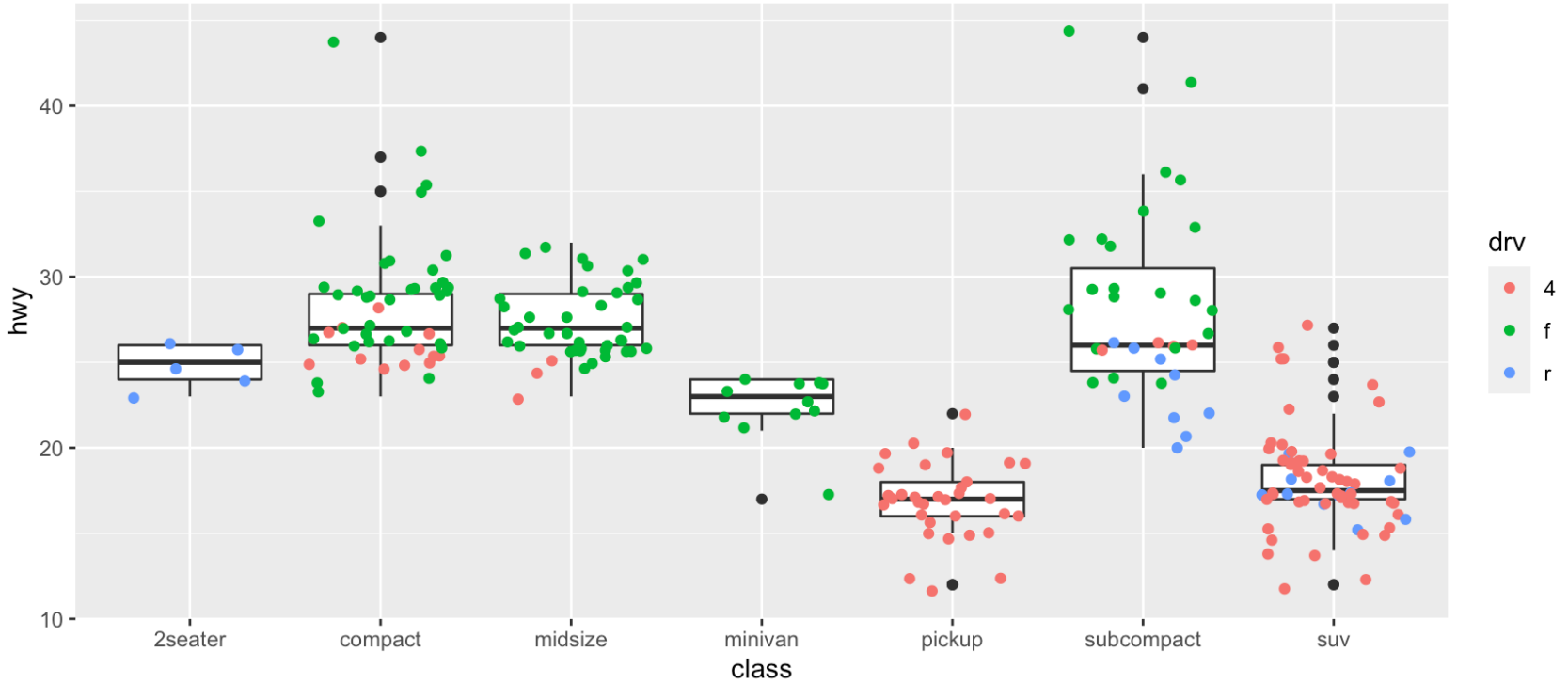
`ggplot2` in practice: two variables

```
ggplot(data = mpg, aes(x = class, y = hwy)) +
  geom_jitter()
```



`ggplot2` in practice: two variables

```
ggplot(data = mpg, aes(x = class, y = hwy)) +
  geom_boxplot() +
  geom_jitter(aes(color = drv))
```



`ggplot2` in practice: other layers

- `theme()`
- `scale_x_continuous()`, `scale_y_continuous()`
- `scale_x_log10`, `scale_y_log10`
- `coord_trans()`
- `annotate()`

`ggplot2` and other packages

- `patchwork`
- `gghighlight`

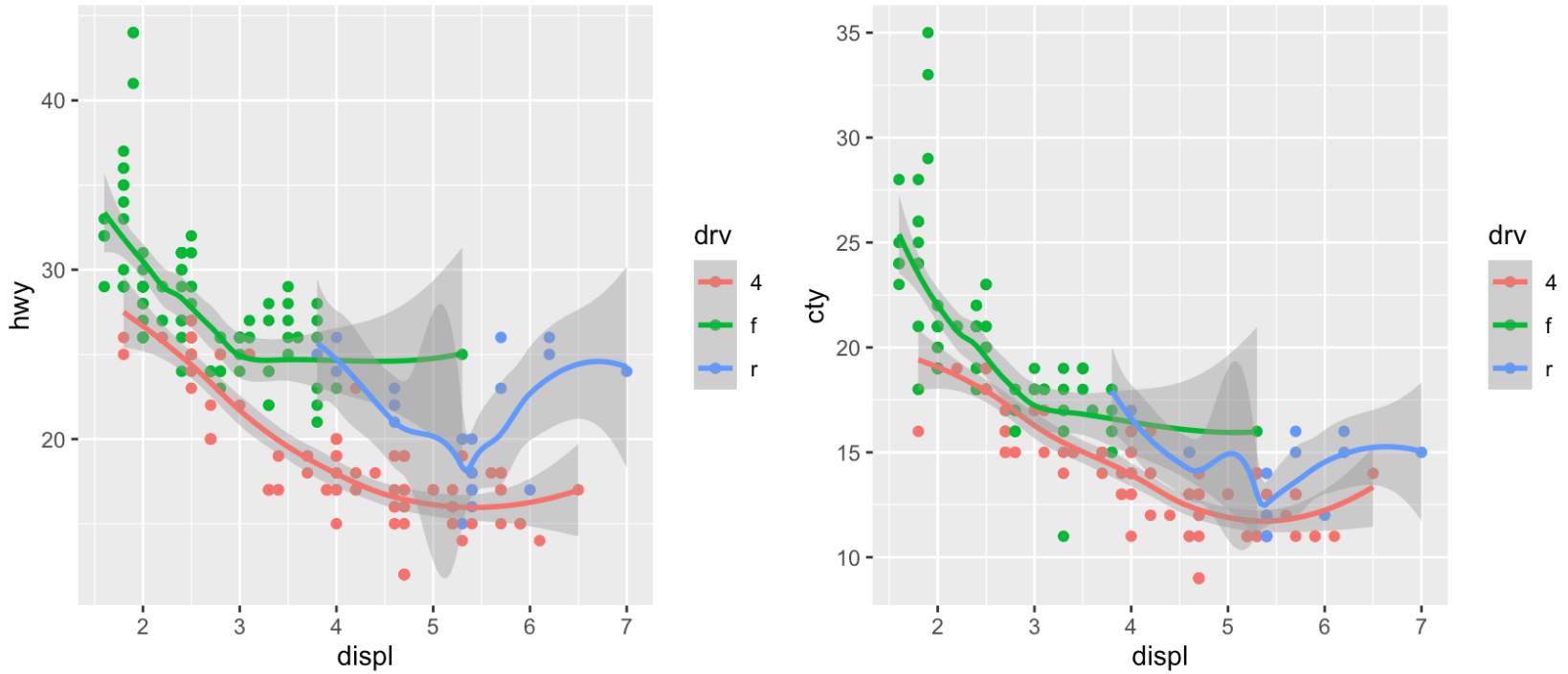
ggplot2 in practice: **patchwork**

```
g_hwy <- ggplot(data = mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point() +
  geom_smooth()

g_cty <- ggplot(data = mpg, aes(x = displ, y = cty, color = drv)) +
  geom_point() +
  geom_smooth()
```

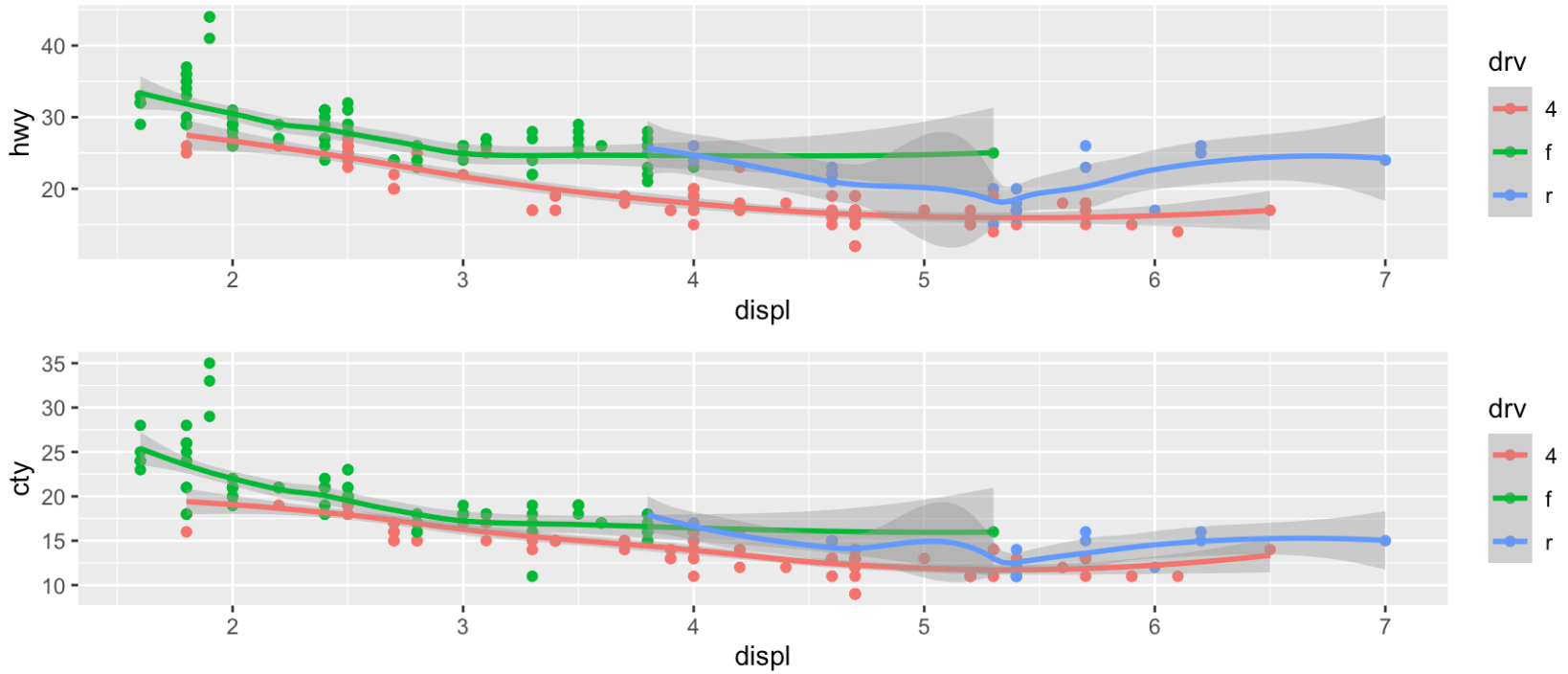
ggplot2 in practice: patchwork

```
library(patchwork)
g_hwy + g_cty
```



ggplot2 in practice: patchwork

```
library(patchwork)
g_hwy / g_cty
```

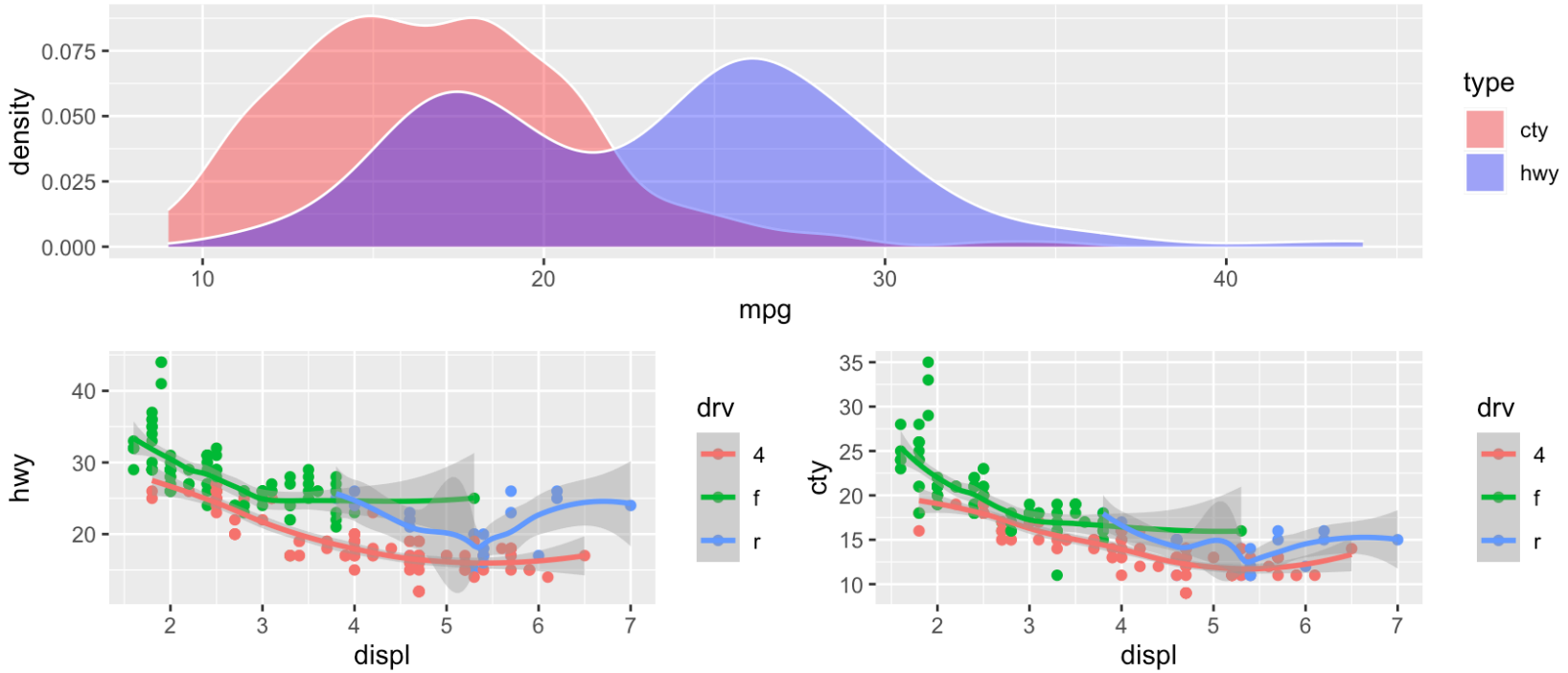


ggplot2 in practice: **patchwork**

```
g_den <- mpg %>%
  select(model, hwy, cty) %>%  pivot_longer(-model, names_to = "type", values_to = "mpg") %>%
  ggplot() +
  geom_density(aes(mpg, fill = type), color = "white", position = "identity", alpha = 0.4) +
  scale_fill_manual(values=c("red", "blue"))
```

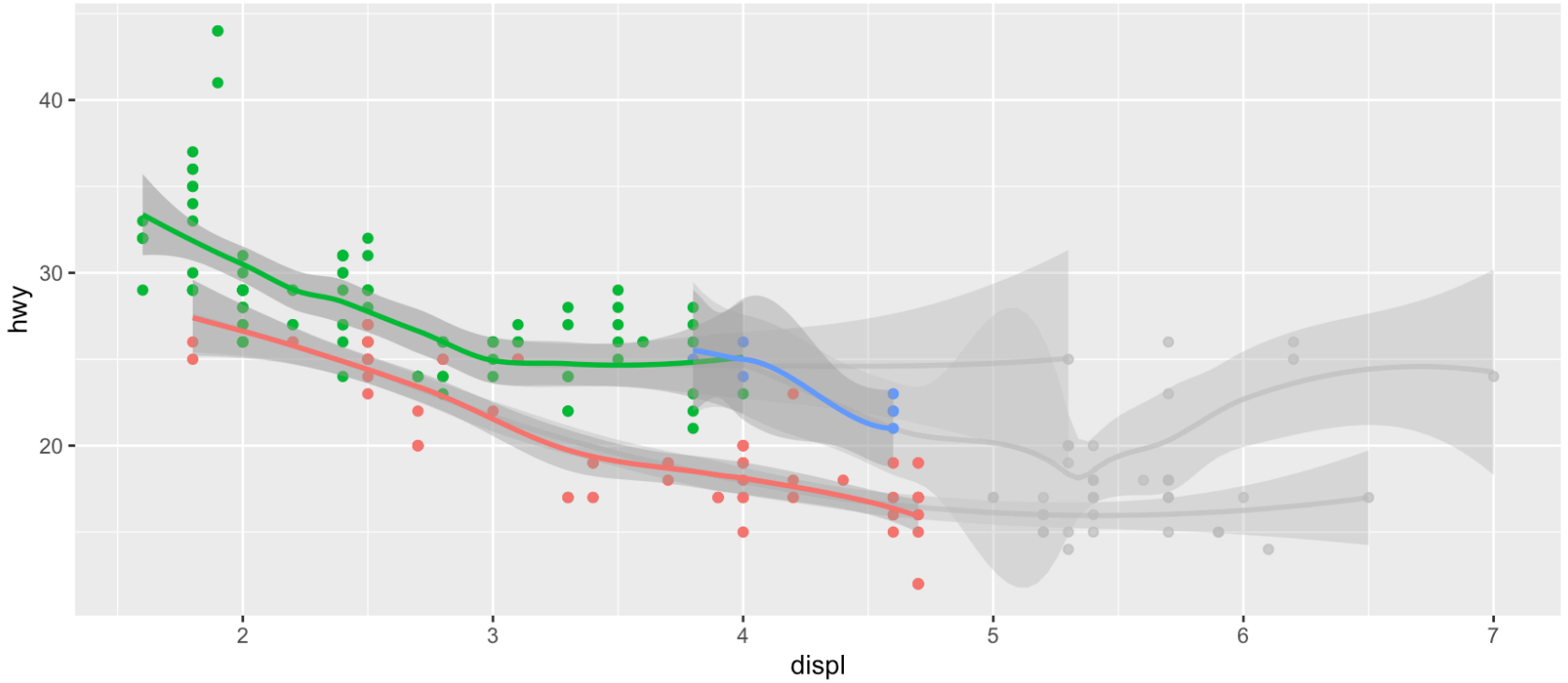
ggplot2 in practice: **patchwork**

```
g_den / (g_hwy + g_cty)
```



ggplot2 in practice: gghighlight

```
library(gghighlight)
g_hwy +
  gghighlight(displ < 5)
```



Questions

References

References

Bone, Frederique, and Daniele Rotolo. 2020. “Text mining historical sources to trace technological change. The case of mass production.” Working Paper.

Börner, Katy. 2016. *Atlas of Science*. Cambridge, Massachusetts: The MIT Press.

Camerani, Roberto, Daniele Rotolo, and Nicola Grassano. 2018. “Do firms publish? A Multi-Sectoral Analysis.” *SPRU Working Paper Series* 21: 1–38. https://www.sussex.ac.uk/webteam/gateway/file.php?name=2018-21-swps-camerani-et-al.pdf{\&}site=25.

Harris, Robert L. 1996. *Information Graphics: A Comprehensive Illustrated Reference*. Oxford; New York: Oxfor University Press.

Hausmann, Ricardo, César A Hidalgo, Sebastián Bustos, Michele Coscia, Alexander Simoes, and Muhammed A Yıldırım. 2013. *The Atlas of Economic Complexity*. MIT Press. http://www.jstor.org/stable/j.ctt9qf8jp.

Jackman, Robert W. 1980. “The Impact of Outliers on Income Inequality.” *American Sociological Review* 45 (2): 344–47. https://doi.org/10.2307/2095134.

Tufte, Edward R. 2001. *The Visual Display of Quantitative Information*. Vol. 4. Cheshire, Connecticut: Graphic Press.

VanderMeer, Jeff. 2013. *Wonderbook: The Illustrated Guide to Creating Imaginative Fiction*. New York, NY, USA: Abrams Image.