Graphical communication and plotting

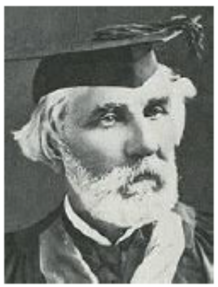
Stat 528, Winter 2015

w/ thanks to Ken Rice for most of these slides

Obligatory quotations

One picture is worth 10,000 words

Fred Barnard (in a fake Chinese proverb)
Printer's Ink 1927



A picture shows me at a glance what it takes dozens of pages of a book to expound.

Ivan Turgenev (Russian Novelist), 1862

Un bon croquis vaut mieux qu'un long discours
(A good sketch is better than a long speech)

attr. Napoleon Bonaparte

1001 words are worth more than a picture
John McCarthy, computer scientist

Why communicate graphically?

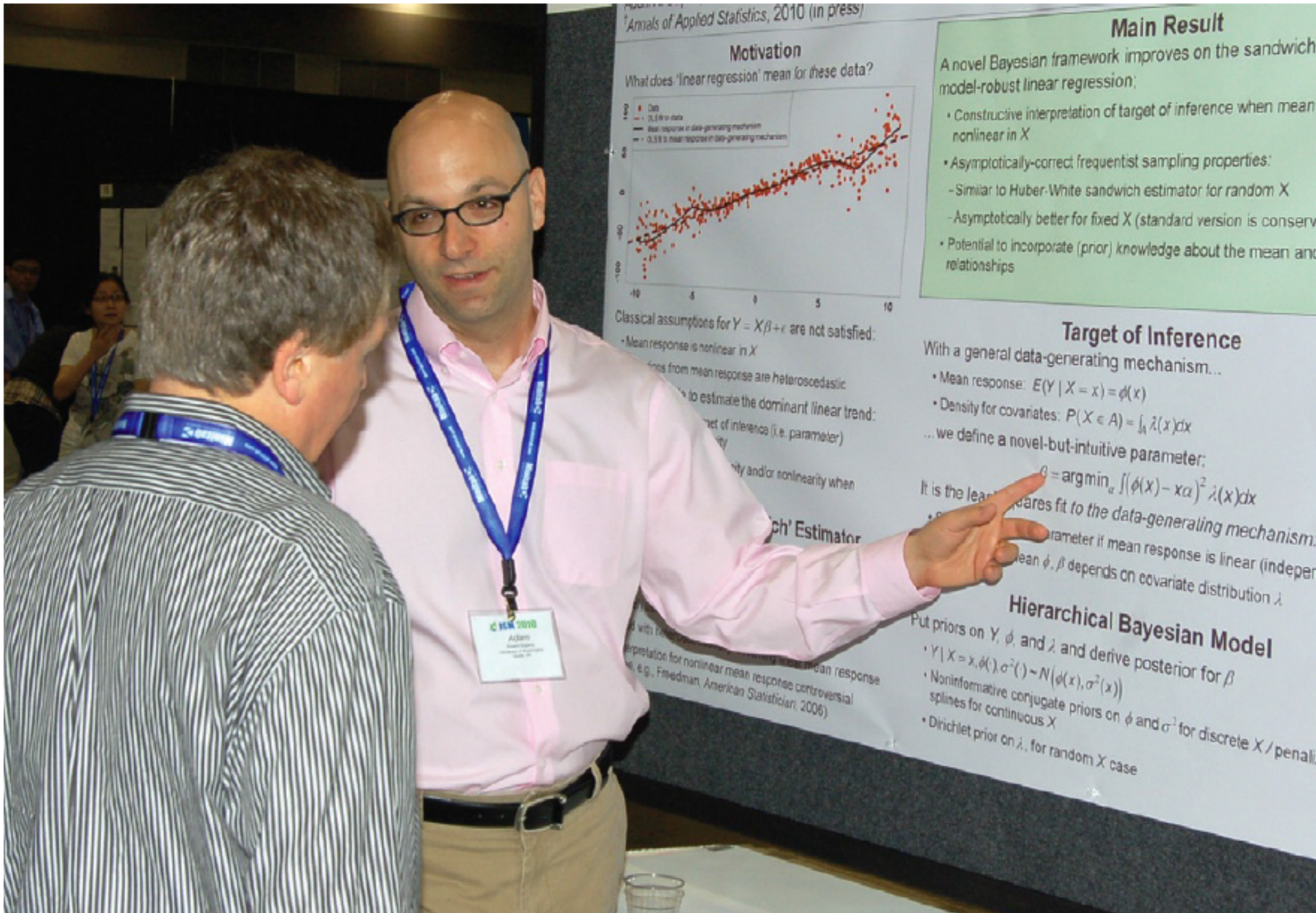
This is a poster session;

Your presentation of information must be;

- Comprehensible – easily/quickly
- Captivating (or near the bar)

Why communicate graphically?

This is a poster session in Amstat News;



Some Recommended Reading

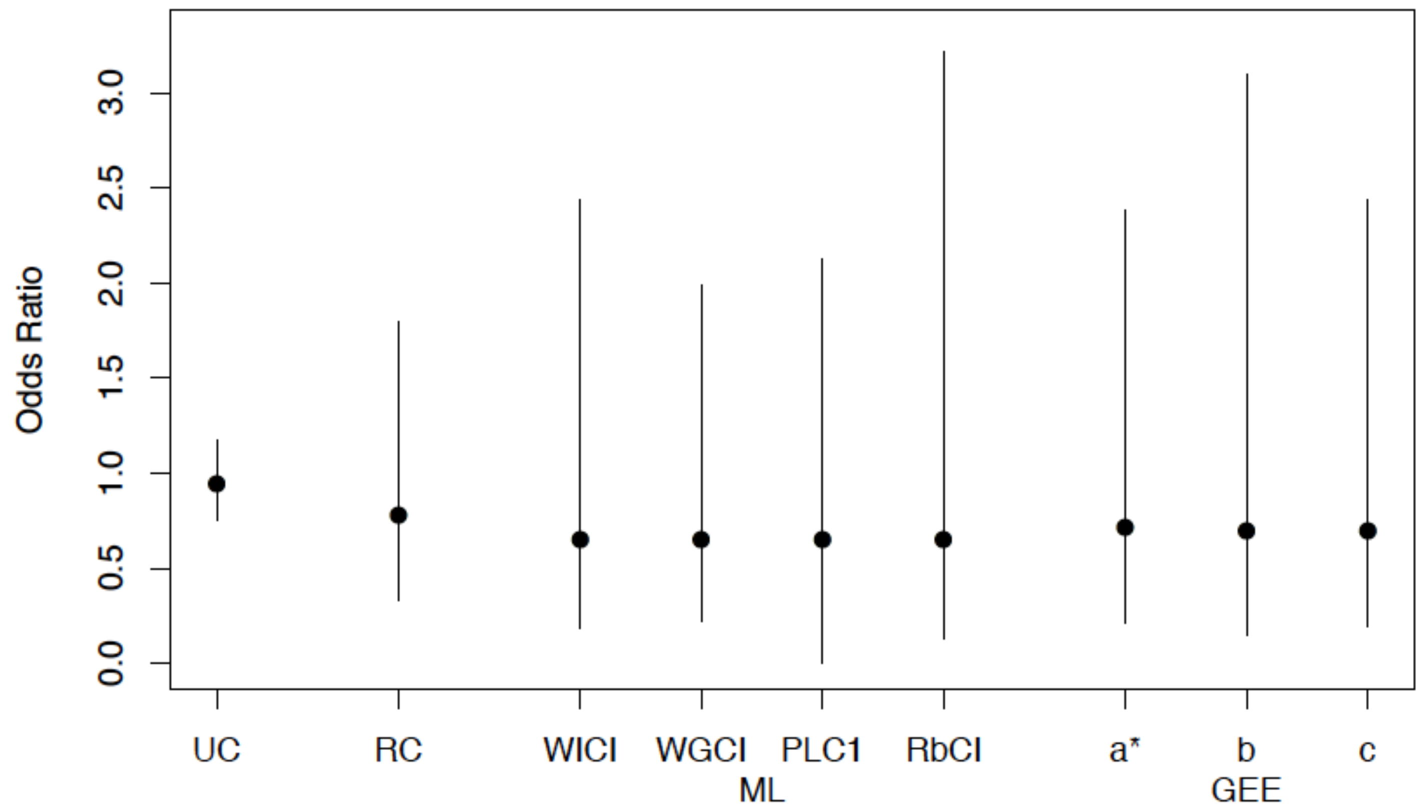
Broadly, statisticians like precision. So why not display your numeric results, precisely, as tables?

In *Let's Practice What We Preach: Turning Tables Into Graphs*, Gelman *et al* (2002) compare tables for **lookup** ...

Method	$\widehat{OR}$	95% Interval
UC	0.94	0.75–1.17
RC	0.78	0.33–1.80
ML-WICI	0.65	0.18–2.44
ML-WGCI	0.65	0.22–1.99
ML-PLCI	0.65	0.00–2.13
ML-RbCI	0.65	0.13–3.22
GEEa*-RBCI	0.71	0.21–2.38
GEEb-RbCI	0.69	0.15–3.10
GEEc-RbCI	0.69	0.19–2.44

Some Recommended Reading

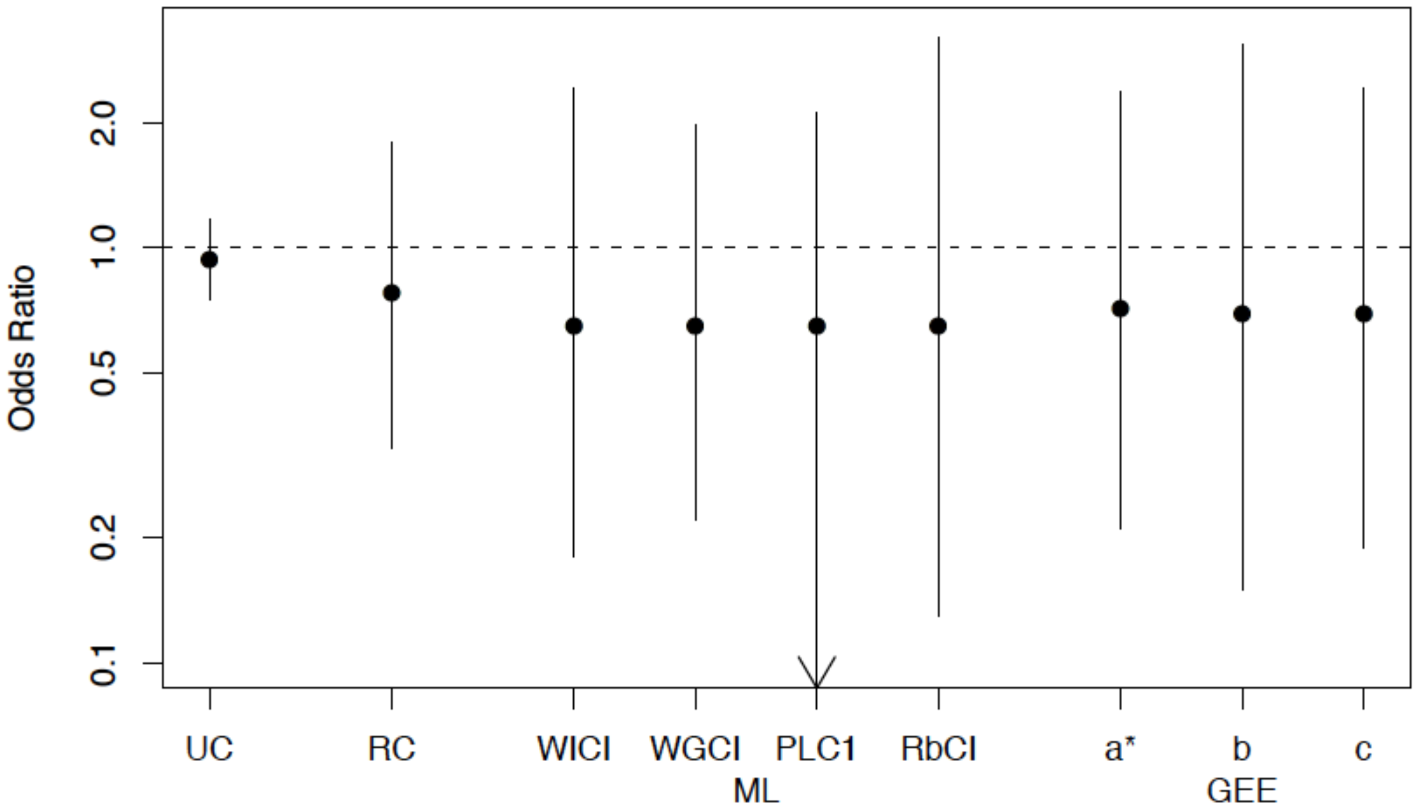
... to graphs, for **comparison**;



- Grouping helps (can also do in tables)
- Comparisons are far easier, faster than in tables

Some Recommended Reading

... to graphs, for **comparison**;



- Log-scale helps compare estimates *and* Std Errs, in this case
- ... but zeroes require extra work

From Gelman et al, 2002, The American Statistician

Method	*Saturated fat* *(≤ 30 g/day; ≥ 30 g/day)*	*Total energy intake* *(1,000 kcal/day)*	*Alcohol* *(20 g/day; ≥ 20 g/day)*	*Age* *(10 years)*
UC	.94 (.75–1.17)	.95 (.78–1.16)	1.35 (1.05–1.74)	1.60 (1.43–1.87)
RC	.78 (.33–1.80)	.80 (.27–2.38)	1.57 (1.09–2.28)	1.54 (1.42–1.67)
ML-WICI	.65 (.18–2.44)	.89 (.26–3.05)	1.55 (1.06–2.27)	1.53 (1.33–1.76)
ML-WGCI	.65 (.22–1.99)	.89 (.26–3.01)	1.55 (1.05–2.27)	1.53 (1.33–1.77)
ML-PLCI	.65 (0–2.13)	.89 (.19–3.61)	1.55 (1.07–2.60)	1.53 (1.25–1.75)
ML-RbCI	.65 (.13–3.22)	.89 (.24–3.30)	1.55 (1.05–2.28)	1.53 (1.34–1.75)
GEE a*-RBCI	.71 (.21–2.38)	.88 (.25–3.15)	1.53 (1.05–2.25)	1.53 (1.38–1.75)
GEE b-RbCI	.69 (.15–3.10)	.93 (.20–4.42)	1.52 (1.01–2.30)	1.55 (1.37–1.76)
GEE c-RbCI	.69 (.19–2.44)	.94 (.24–3.71)	1.52 (1.03–2.23)	1.55 (1.38–1.75)

Figure 6a. Table from Spiegelman et al. (1990) compares several methods of estimating odds ratios and 95% intervals in a logistic regression.

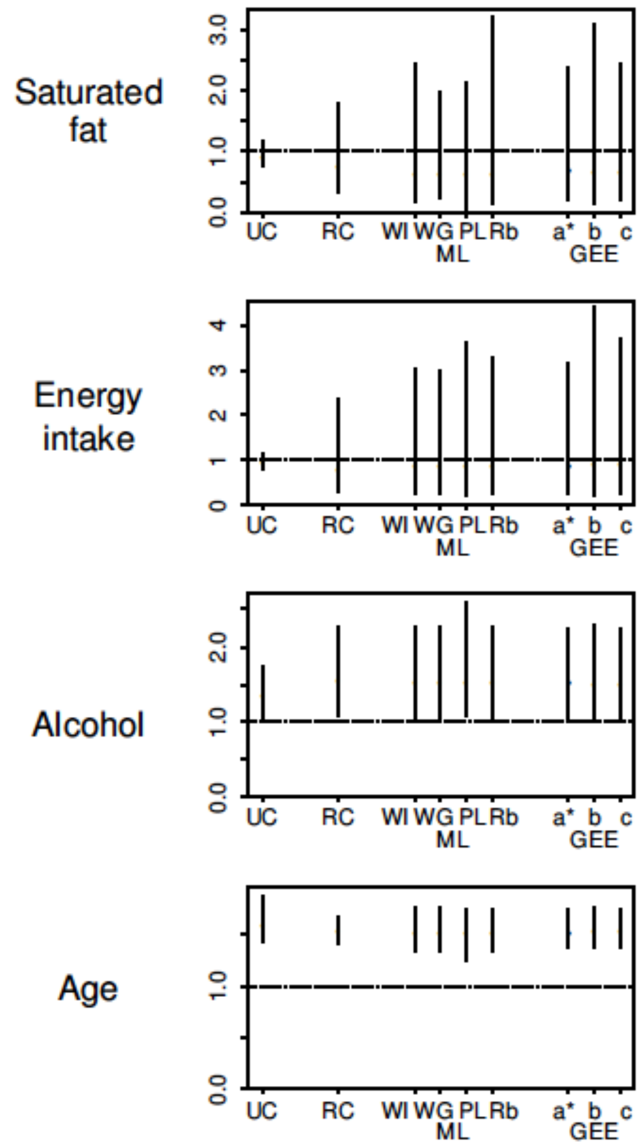
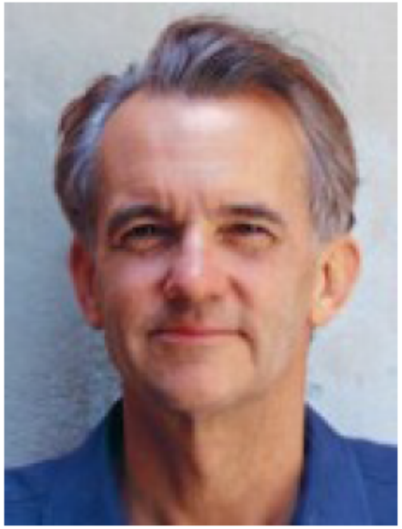
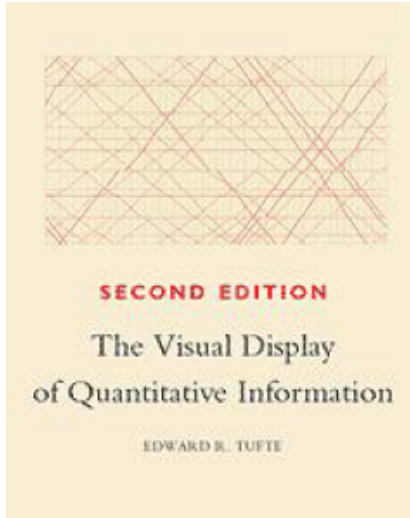
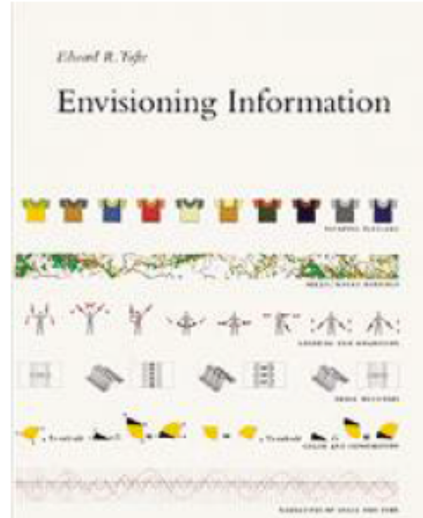


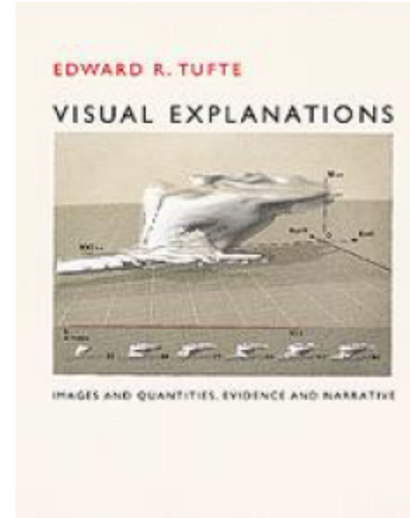
Figure 6b. Graphical display of Figure 6a separates the comparisons for the four odds ratios. For each, the estimates and standard errors under the different methods are displayed, with the different forms of ML and GEE estimates grouped. The y-axes of the graphs are on different scales, but they all include zero at the bottom and have dotted lines showing the reference odds ratio of 1.

More Recommended Reading









Principles from Tufte;

- Serve a reasonably clear purpose
- Show the data
- Avoid distorting what the data have to say
- Encourage the eye to compare different pieces of data

Later work (Stephen Few) adds practical ideas from visual perception research, to help us do this. [Stephen Few website]

Following Tufte's principles (or not)

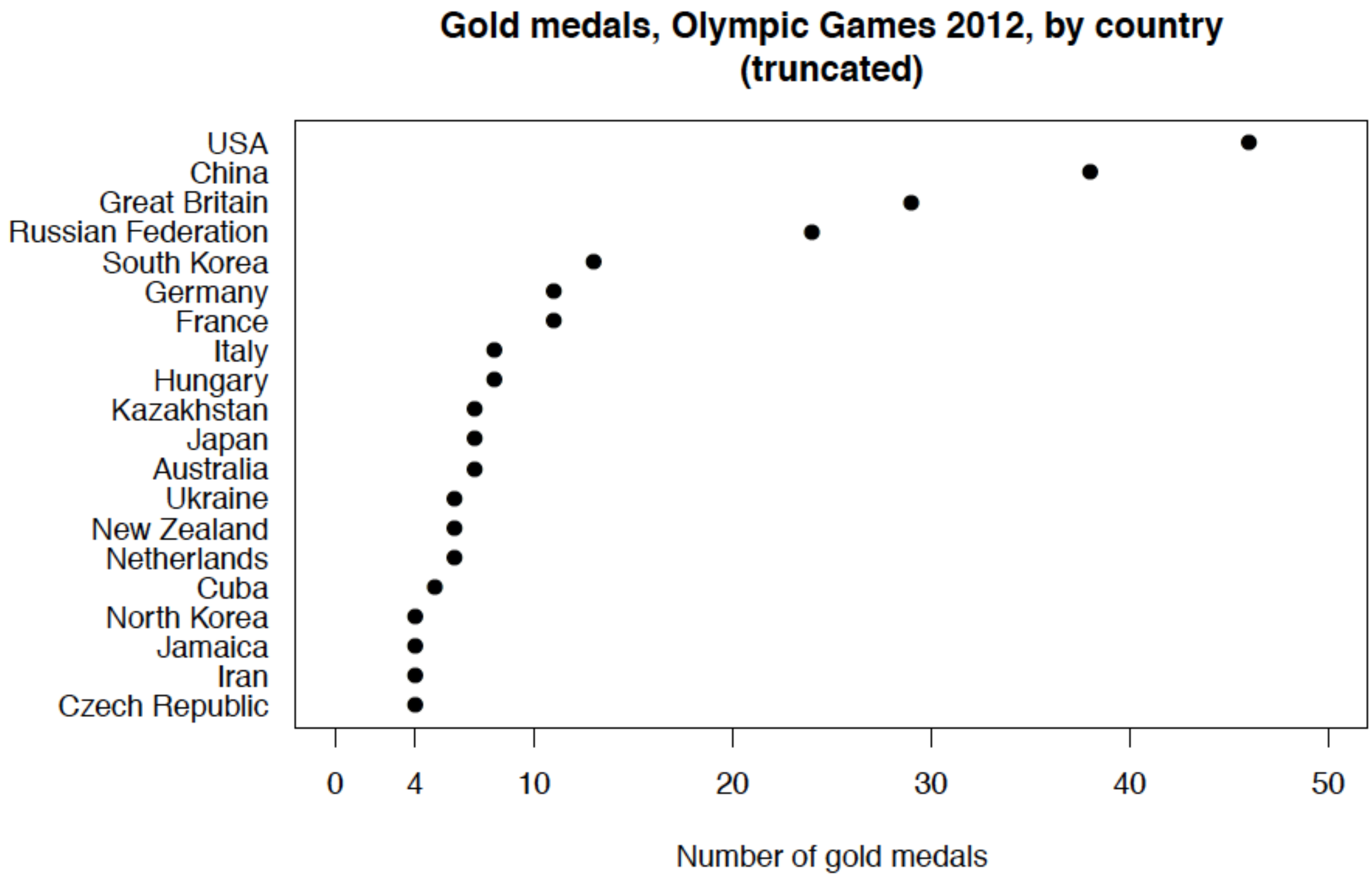
What *is* the graph's purpose?

- Histogram/Dotchart: *summarize* one-dimensional continuous data
- Barchart: *compare* one-dimensional categorical data
- Scatterplots: *show association* of continuous Y and X (or lack of association)
- Mosaic plots: *show association* of categorical Y and X (or lack of association)
- Boxplots: *show association* of continuous Y and categorical X (or lack of association), emphasizing 25/50/75 %iles
- QQ plots: *compare* two continuous distributions; talk about the shift, spread, heavy tails, light tails etc

Note the close connections to 'Table 1', t-tests, regression etc

Following Tufte's principles (or not)

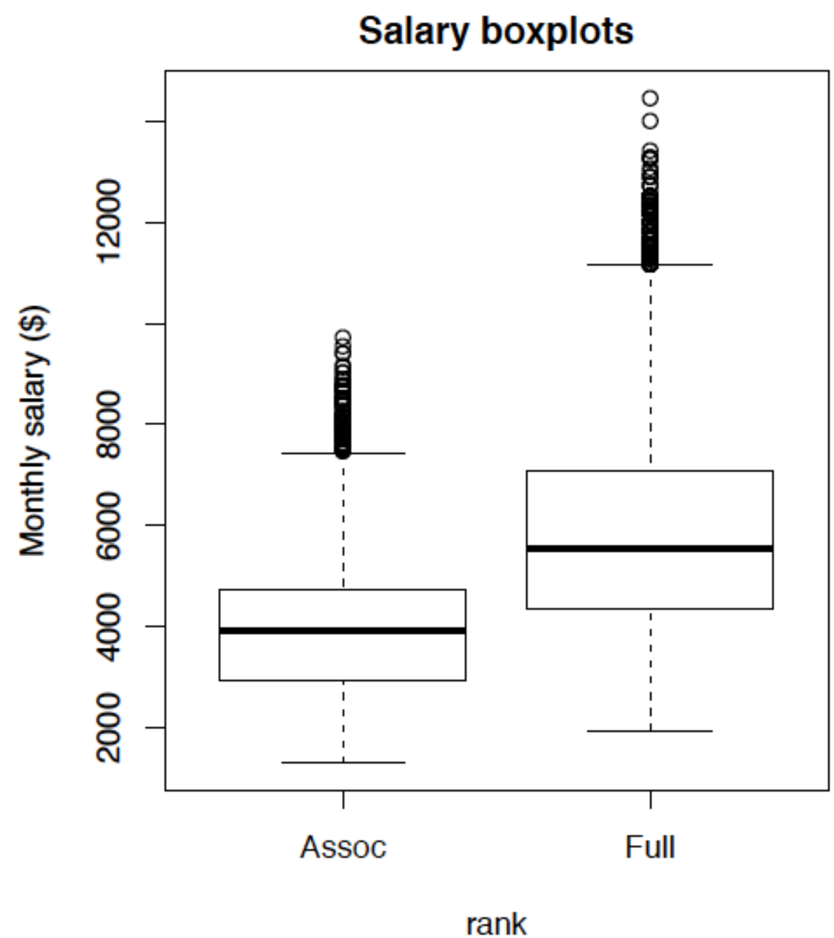
A dotchart – see `dotchart()`, `stripchart()`

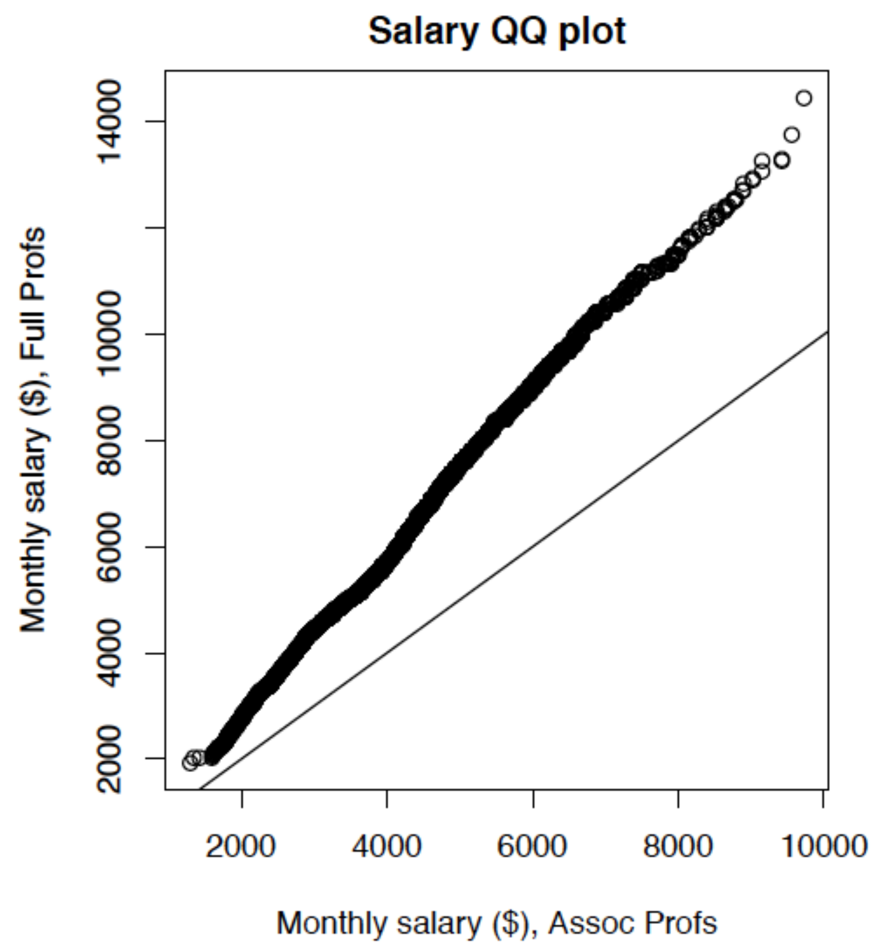


– good for up to ≈20 lines, can use > 1 point per line.

Following Tufte's principles (or not)

Boxplot or QQ plot? It depends what you want to compare;

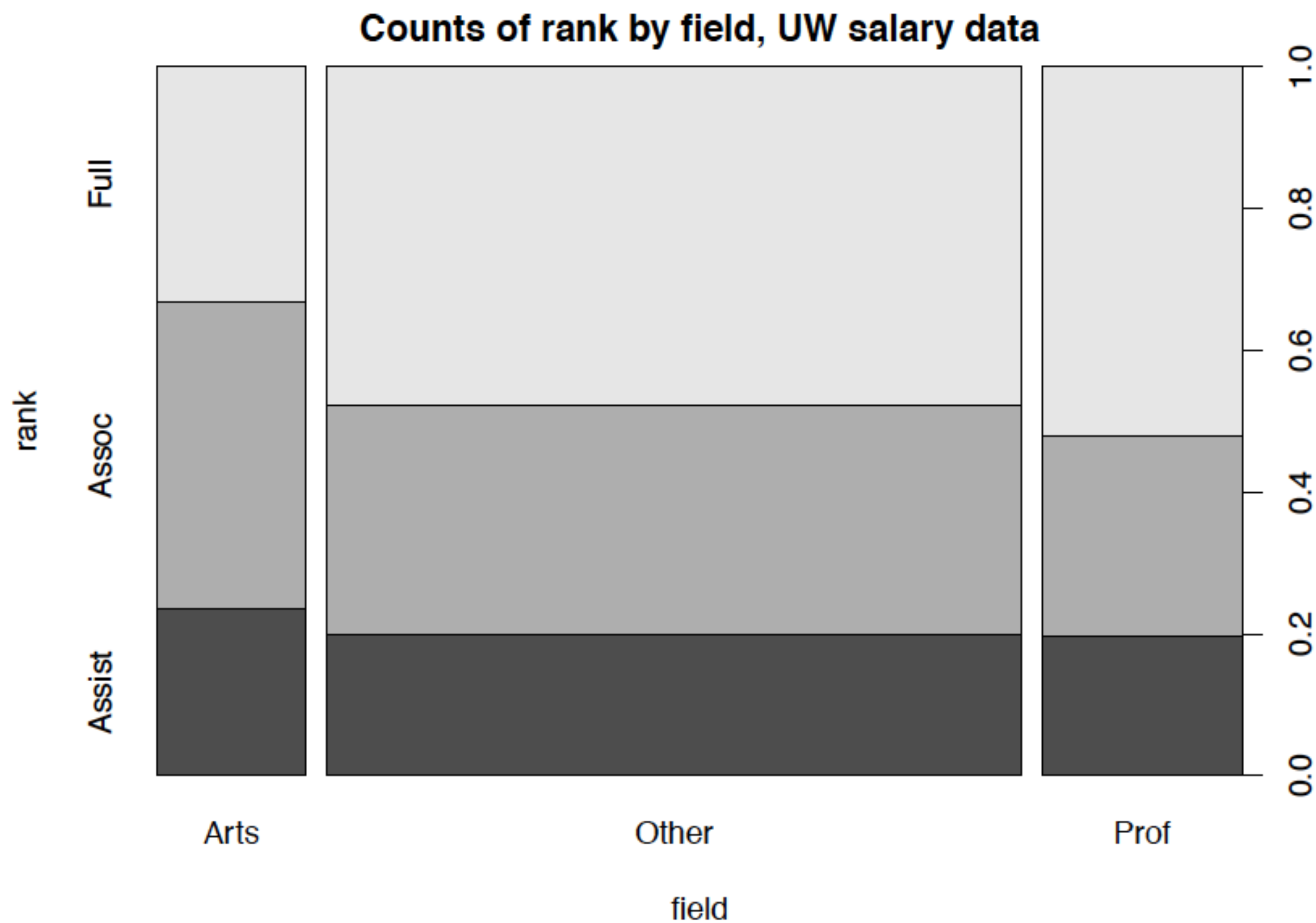




The 'whiskers' go to most extreme data that's $\leq 1.5\times$box length away from the box... don't expect readers to remember this.

Following Tufte's principles (or not)

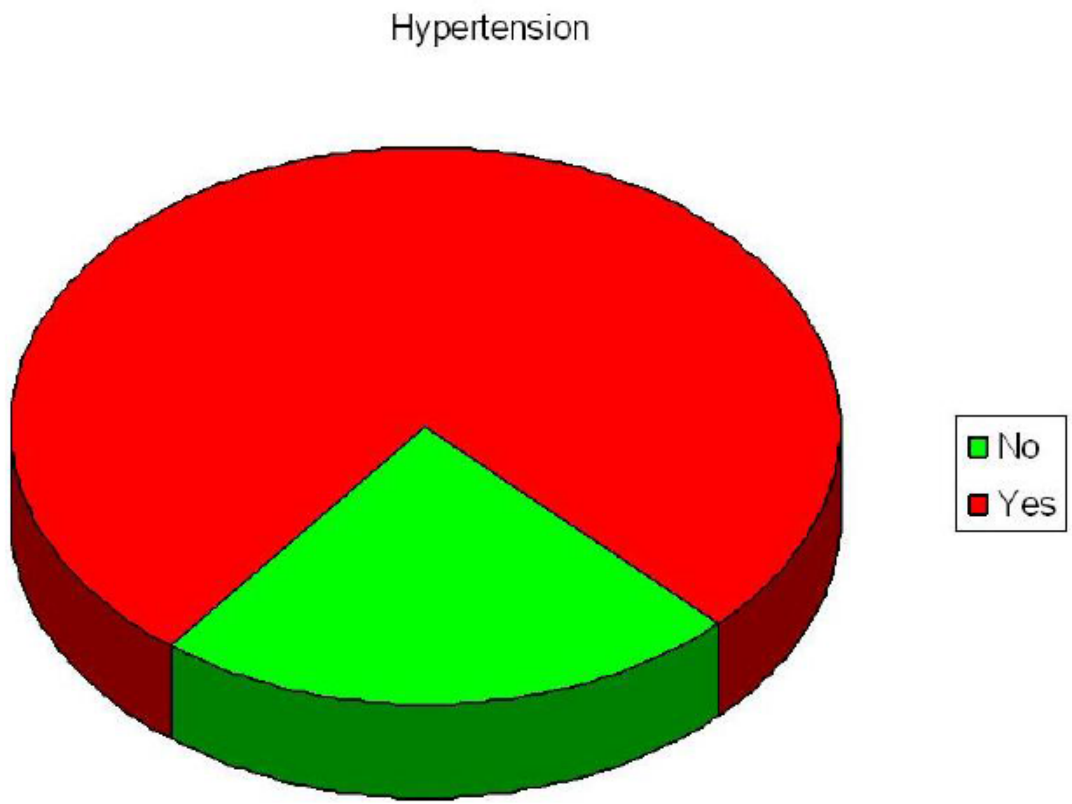
A mosaic plot – areas indicate counts, use `plot(y~x)` in R



... have to condition on one variable (just like regression)

Following Tufte's principles (or not)

Show the data ?



From a real poster; (American Heart Association); three of these (percentages Yes, Female, Yes & Female) were worth a 2×2 table ... more examples later!

Following Tufte's principles (or not)

Avoid distorting what the data have to say ?

Some "Fair and Balanced" Fox News reporting – "We Report. You Decide", in 2010

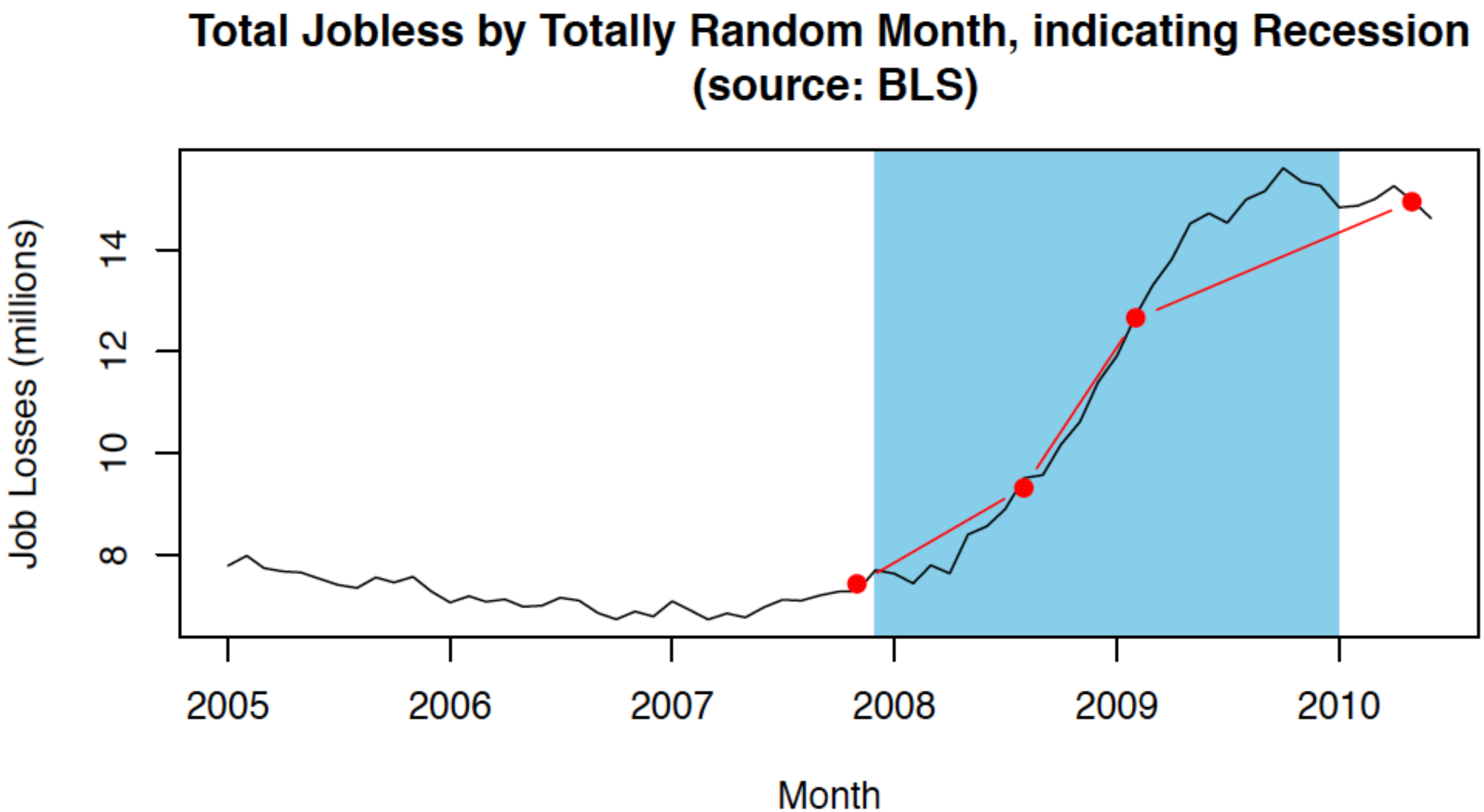
Following Tufte's principles (or not)

They did correct the wildly-wrong title;

... but are *still* distorting the BLS's hard work;

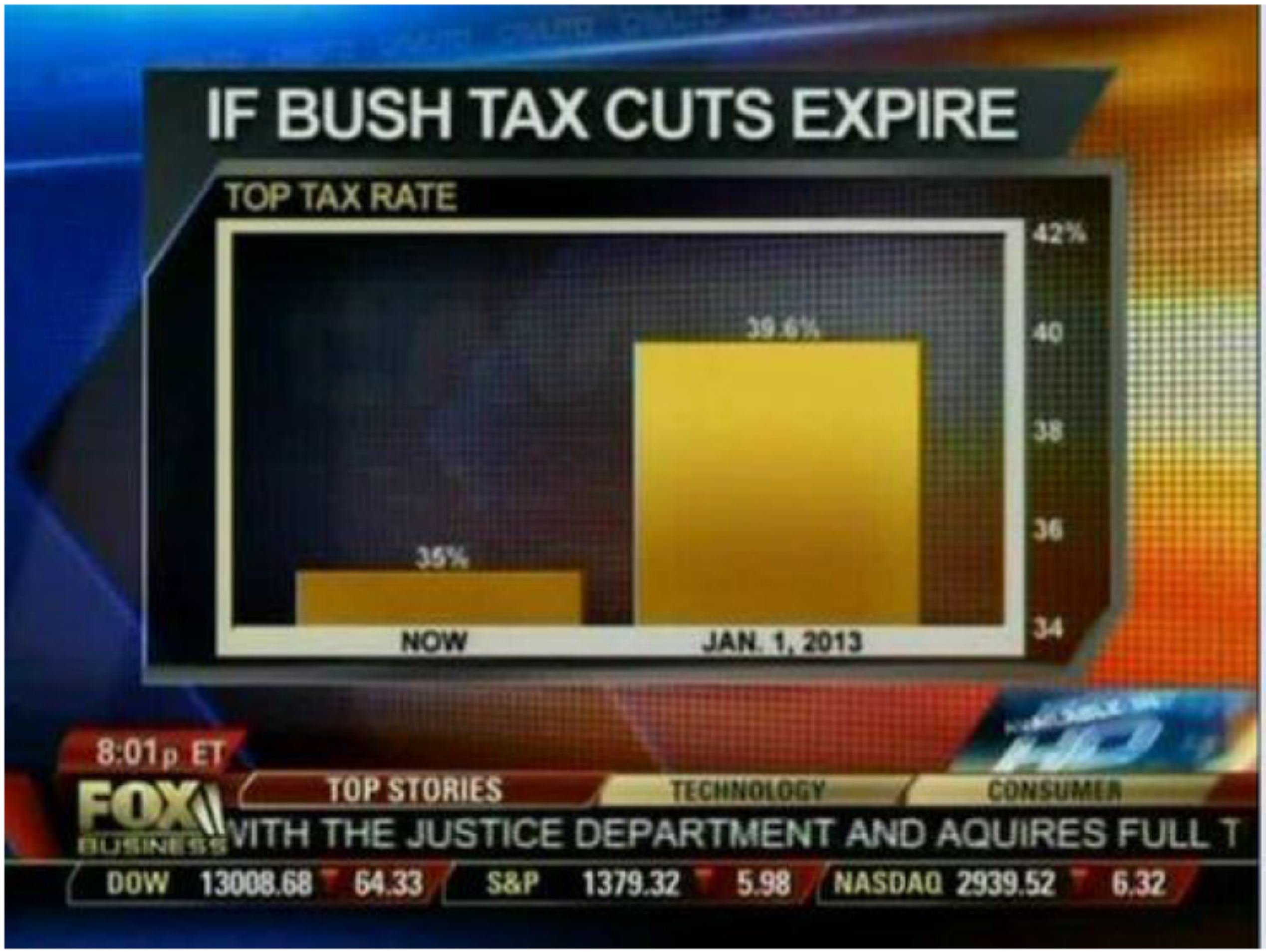
Following Tufte's principles (or not)

Distorted? Yes – as we see, using *actual* BLS data;



Following Tufte's principles (or not)

The August 2012 vintage – actually an old trick;



Daryl Huff exposed this in *How To Lie With Statistics* (1954).

Following Tufte's principles (or not)

Encourage the eye to compare different pieces of data ?

This is easier said than done. Good graphs, like good statistical analysis, should help your reader *accurately* assess whether;

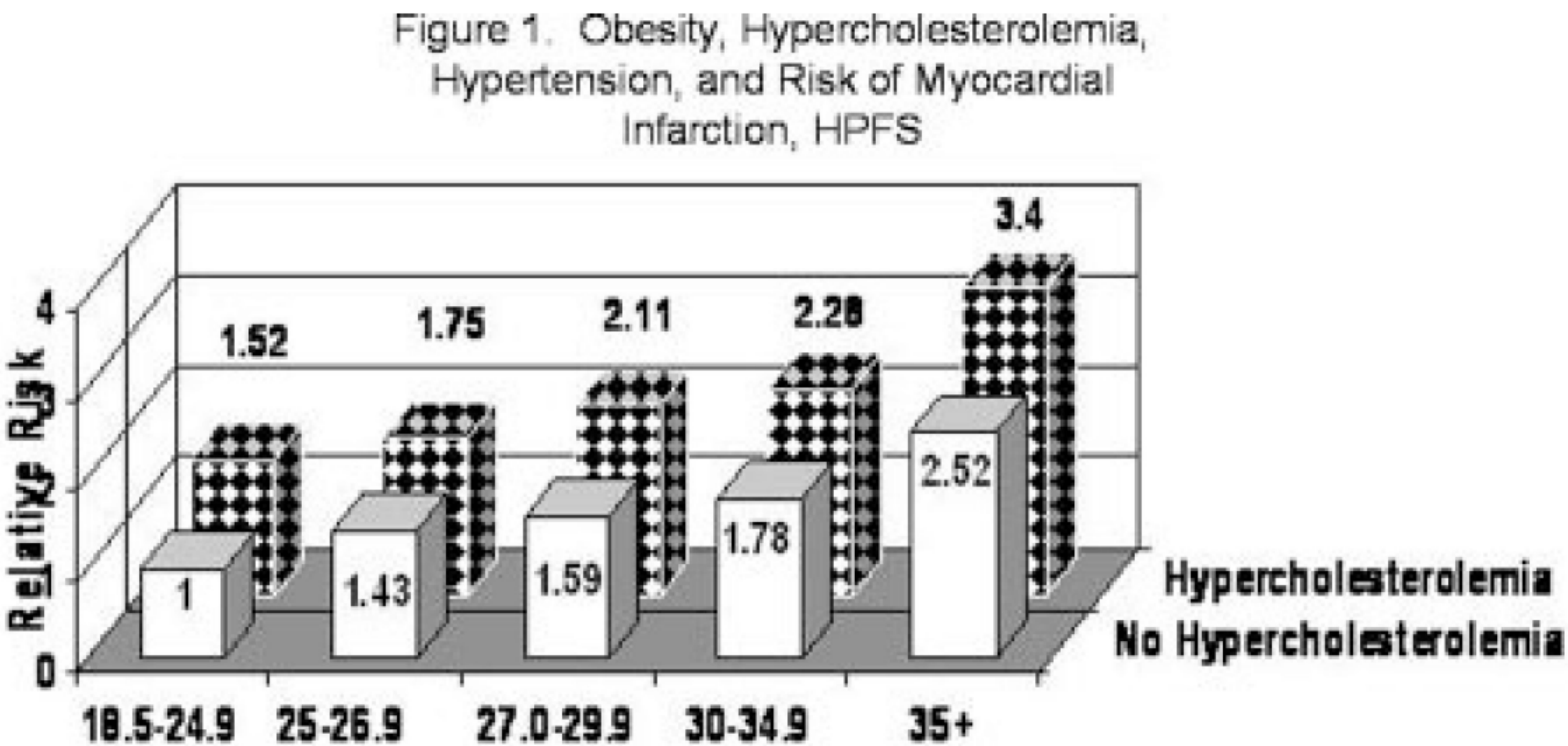
- The effect is there
- The effect is not there
- The data are so uninformative that no-one can tell

To make a graph that does this, a good starting point is 'use the R defaults' – these are based on work at Bell Labs in the 1970s, on early graphics systems, making use of research into how visual perception actually works.

They are a *little* out of date – particularly for plots of 'big data' – but still out-perform defaults from elsewhere.

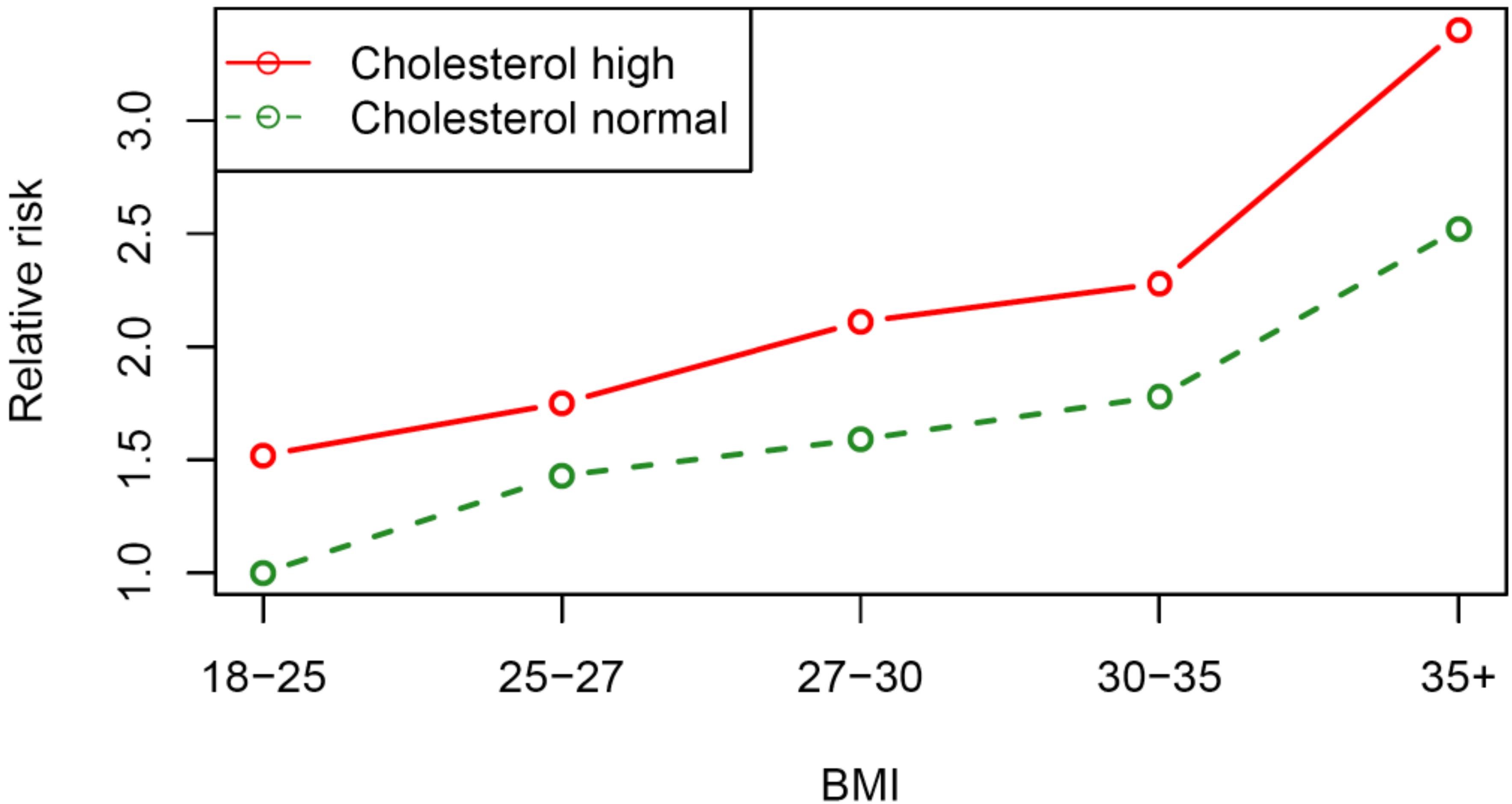
Effective Comparisons

Some actual science – back to the AHA Epi conference;



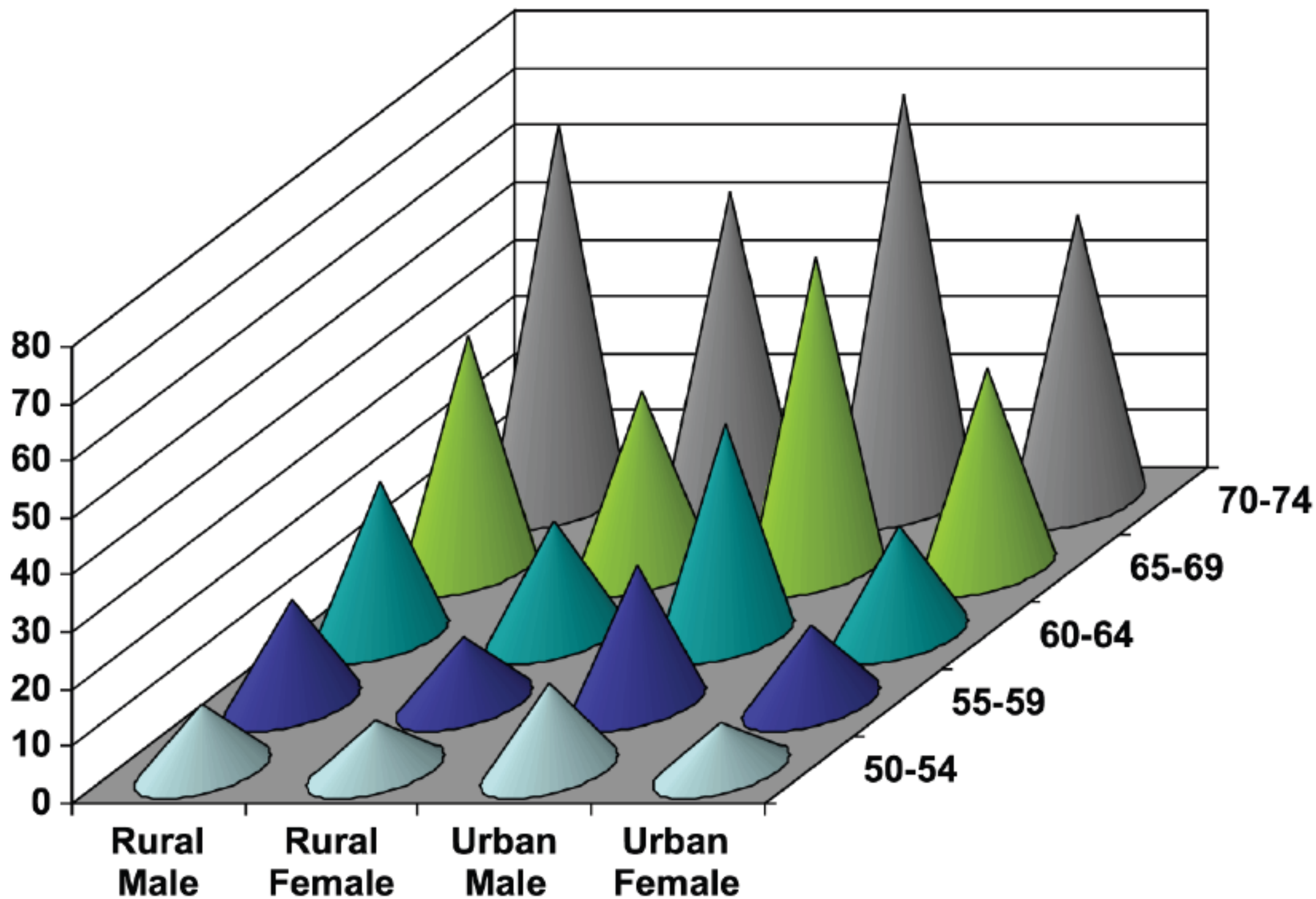
Effective Comparisons

Re-imagined; (confidence intervals would help too)



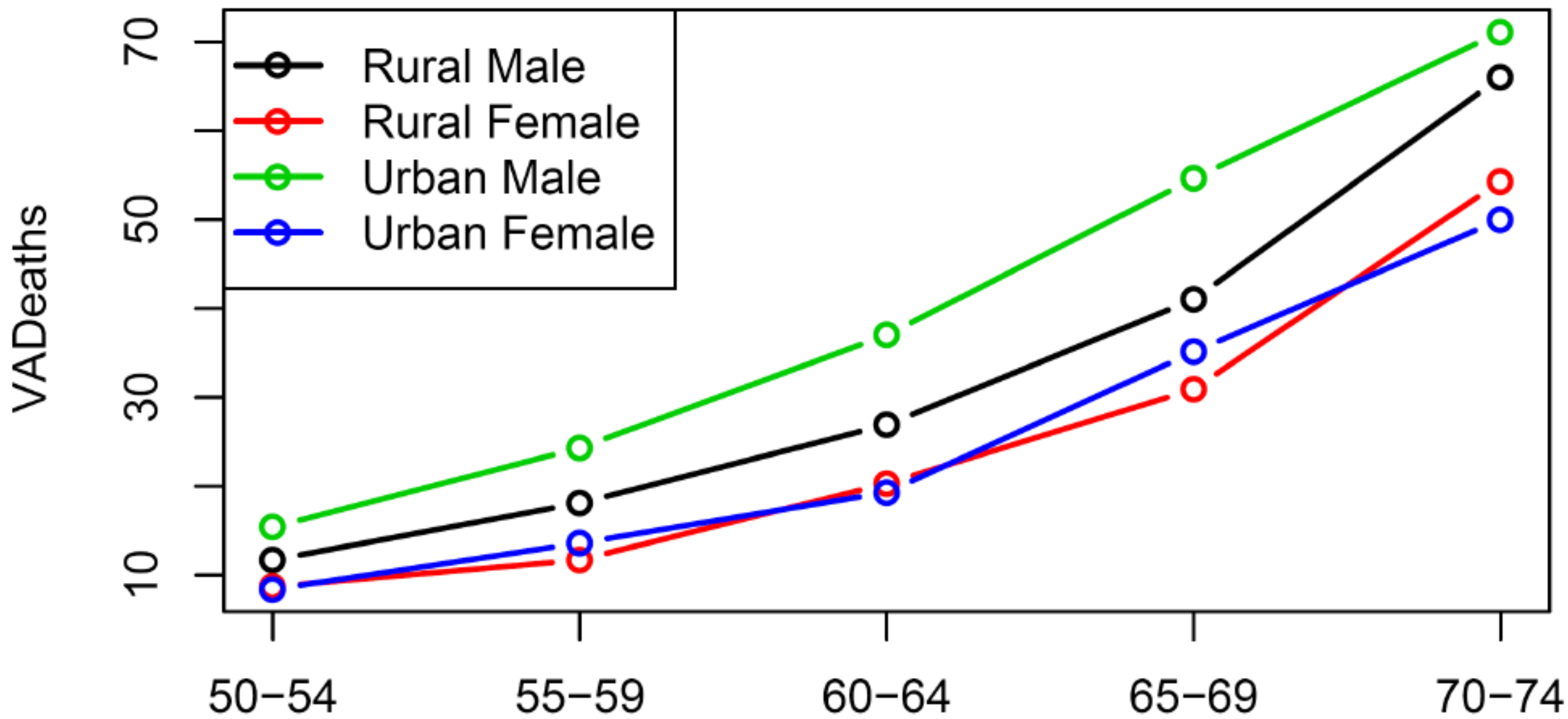
Effective Comparisons

The ‘Bed of Nails’, from an AHA poster;



Effective Comparisons

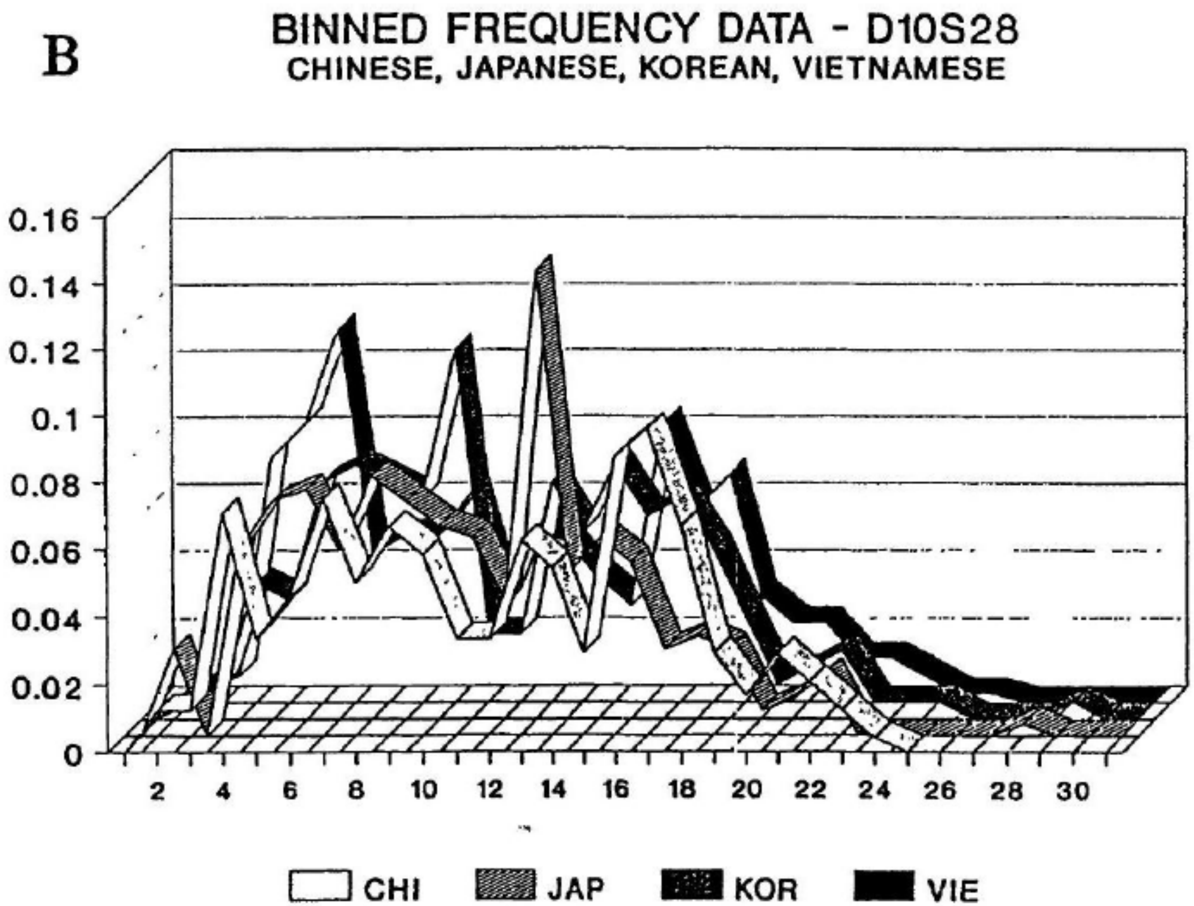
A comprehensible version – note overplotting/empty circles;



Note both graphs used `type="b"`; this is okay as there *is* an 'underlying continuum'.

Effective Comparisons

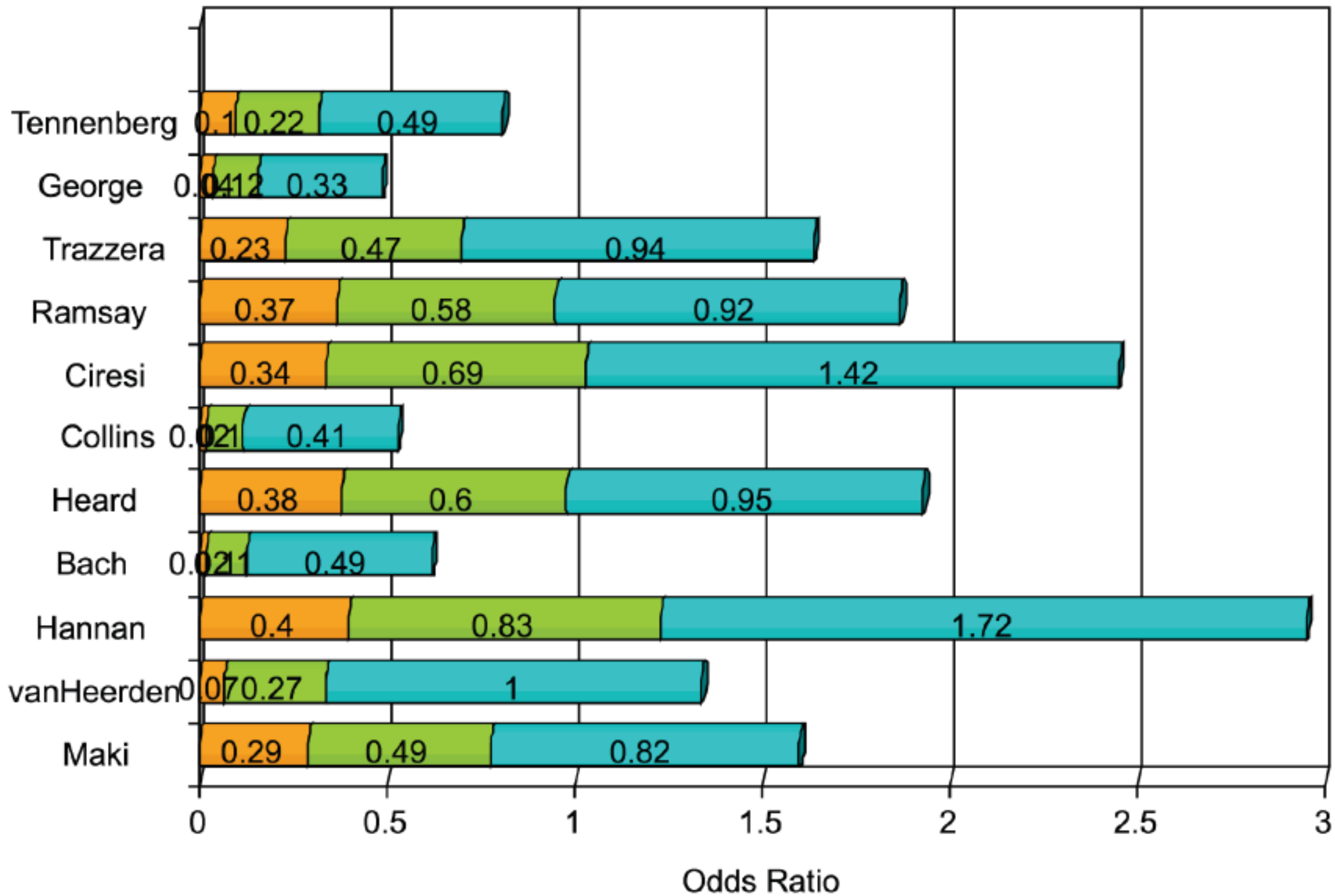
A good statistician making a very bad graph;



Roeder K (1994) DNA fingerprinting: A review of the controversy (with discussion). *Statistical Science* 9:222-278, Figure 4

Effective Comparisons

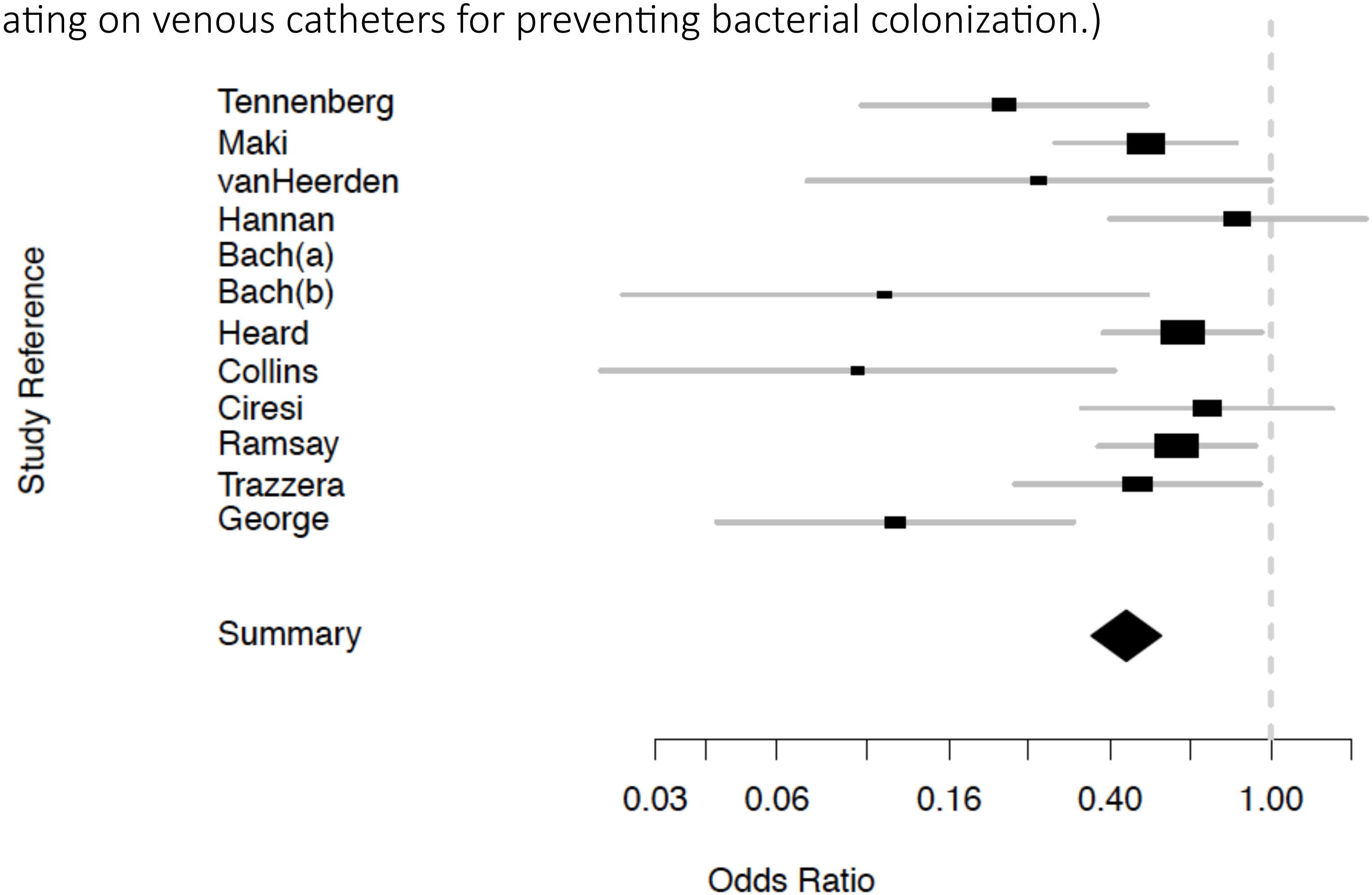
We often want to compare intervals. Here's the *worst* way;



Effective Comparisons

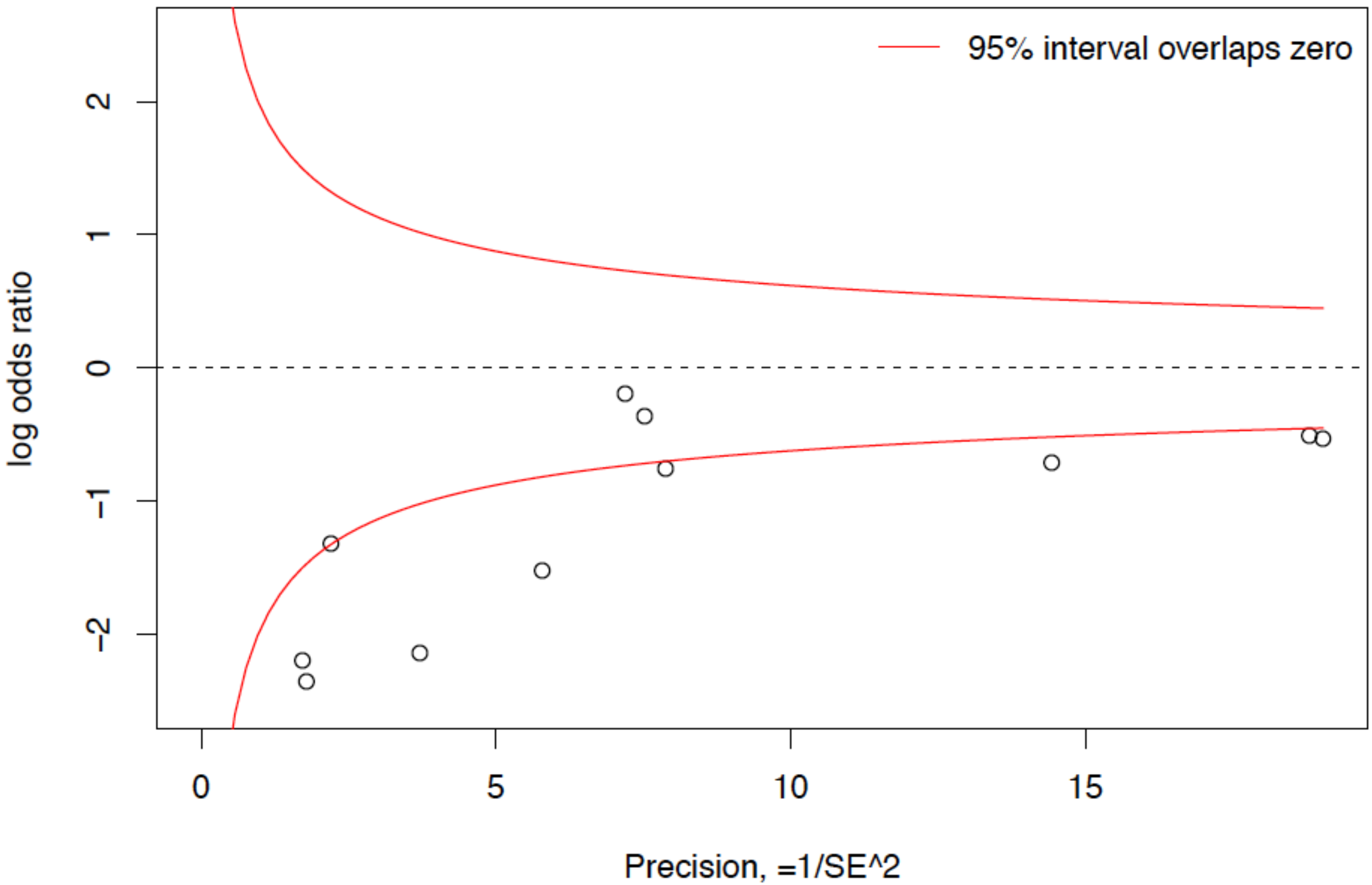
The standard method is a 'forest' plot; (see `rmeta` package)

(These are from a meta analysis, reporting multiple studies on the effectiveness of a coating on venous catheters for preventing bacterial colonization.)



Effective Comparisons

'Funnel plots' are better, can help show publication bias;



What works well/not so well?

The comparisons that *actually work well* may not be the ones that look coolest. Based on perception research, here are the best methods for visual comparison of numeric data;

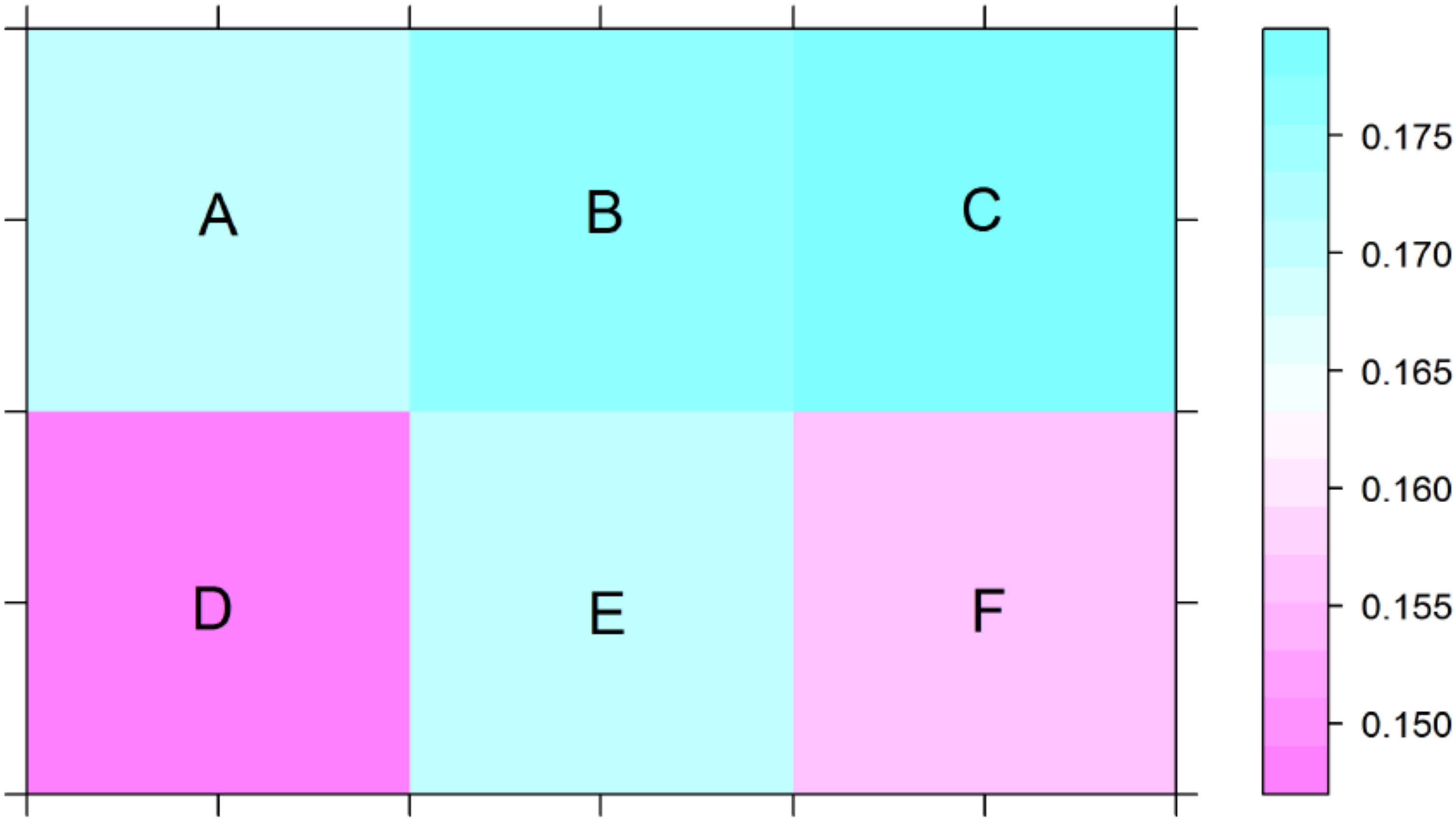
Metric	Usage	Accuracy
Position on common scale	Dot Plot	Best
Length	Bar chart	
Angle/Slope	Pie chart	
Area	Bubble Plot	
Volume/Curvature	Fake3D	
Color hue, density	e.g. Heat map	Worst

- For more see e.g. Cleveland and McGill (JASA 1984, JRSSA 1987), and Few's books
- Let's illustrate these; try to order the 6 numbers A,B,...,F

Stephen Few website for his books, articles and more on visual/graphical display.

What works well/not so well?

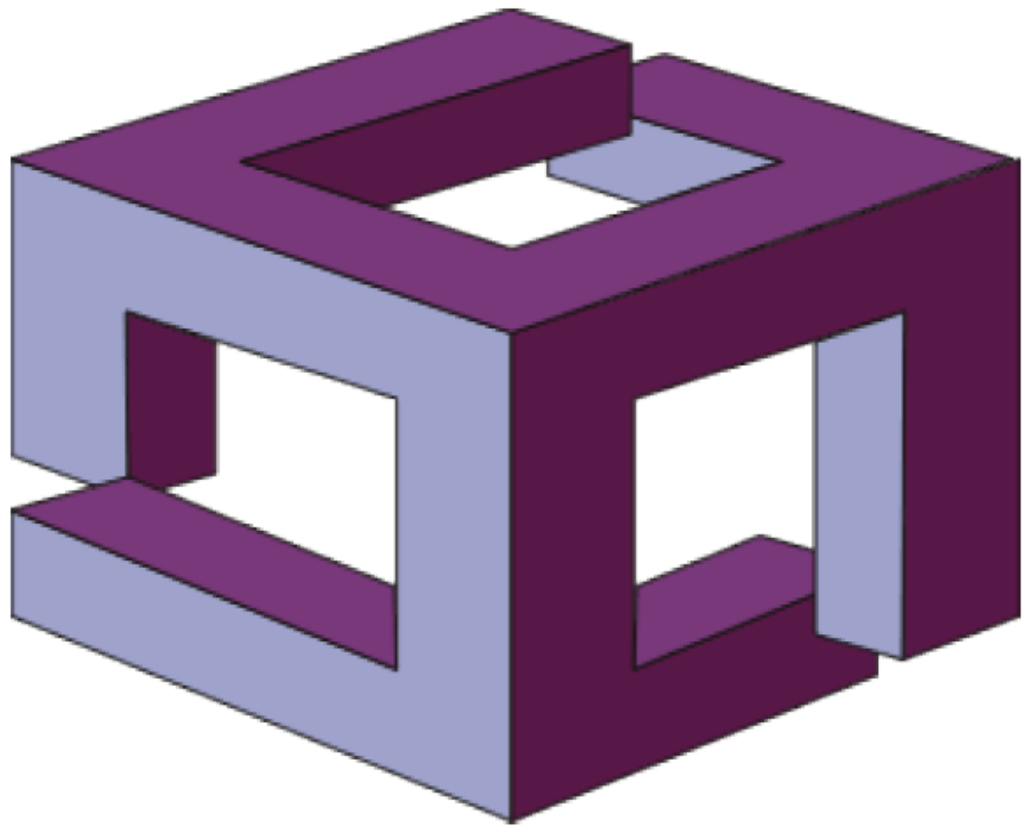
Using `image()` or `levelplot()`;



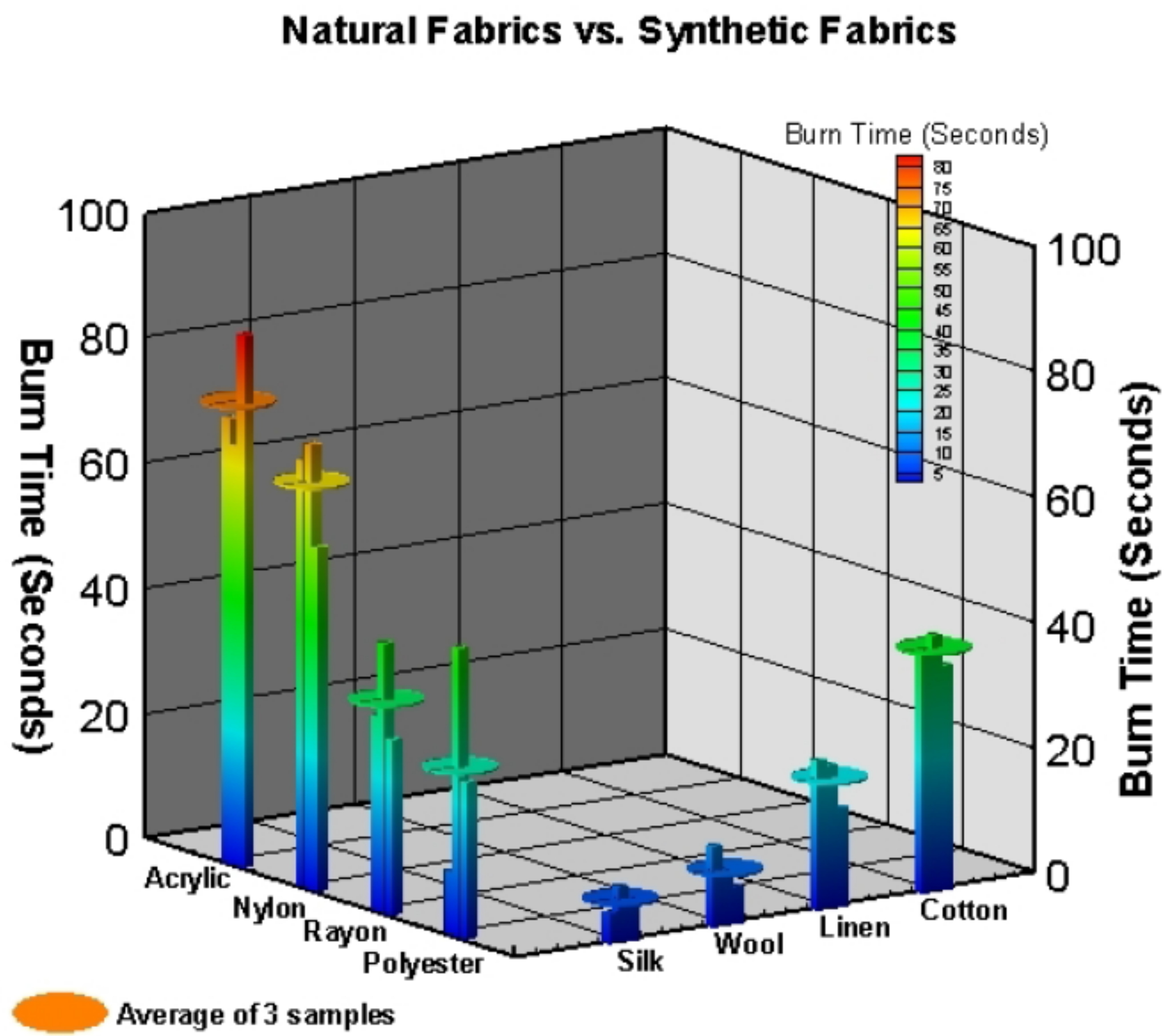
... gets really messy with larger plots

What works well/not so well?

Your brain (and everyone else's) is really poor at unpicking 3D information from 'flat' pictures;



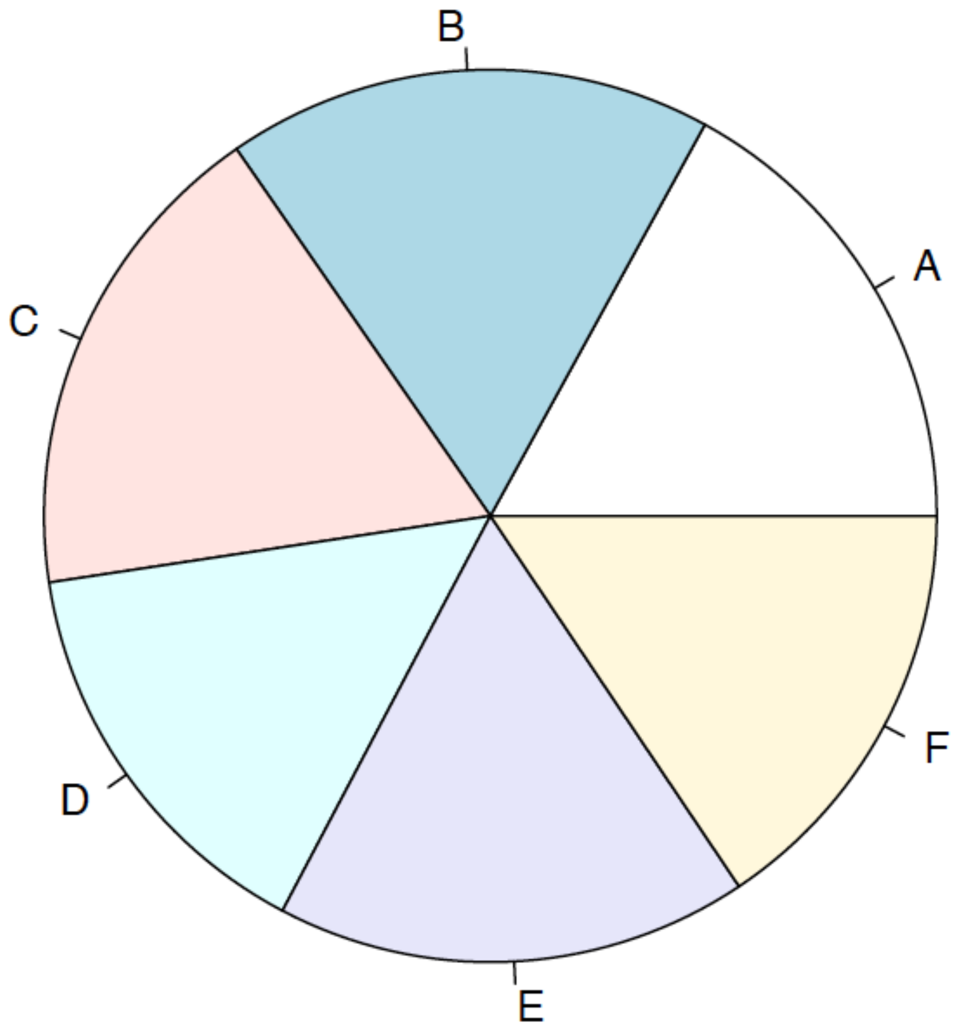
What works well/not so well?



Spiffy 3D plot in GAUSS – it'd take *ages* to be *so* bad in R.

What works well/not so well?

Back to our 6 numbers; `pie()` is available, but seldom useful;

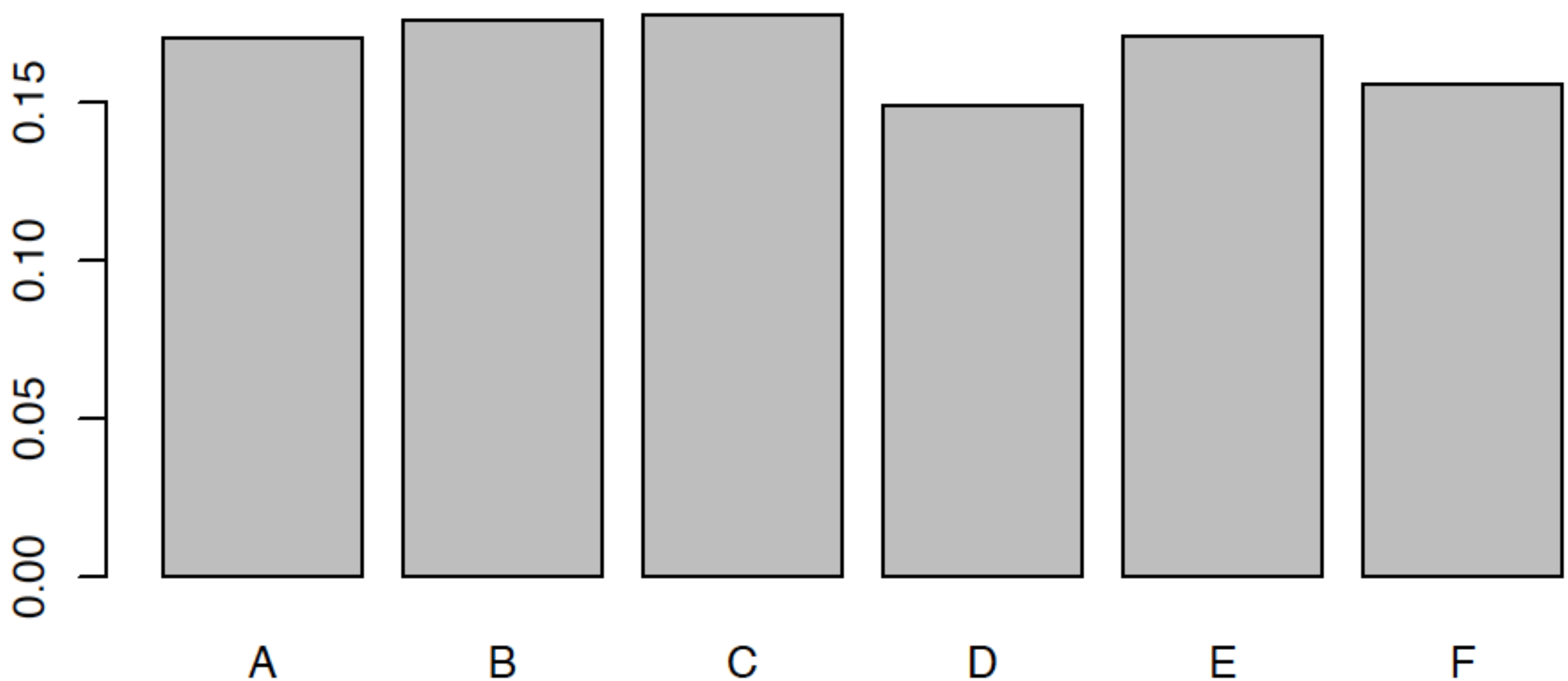


What works well/not so well?

One actually-acceptable pie;

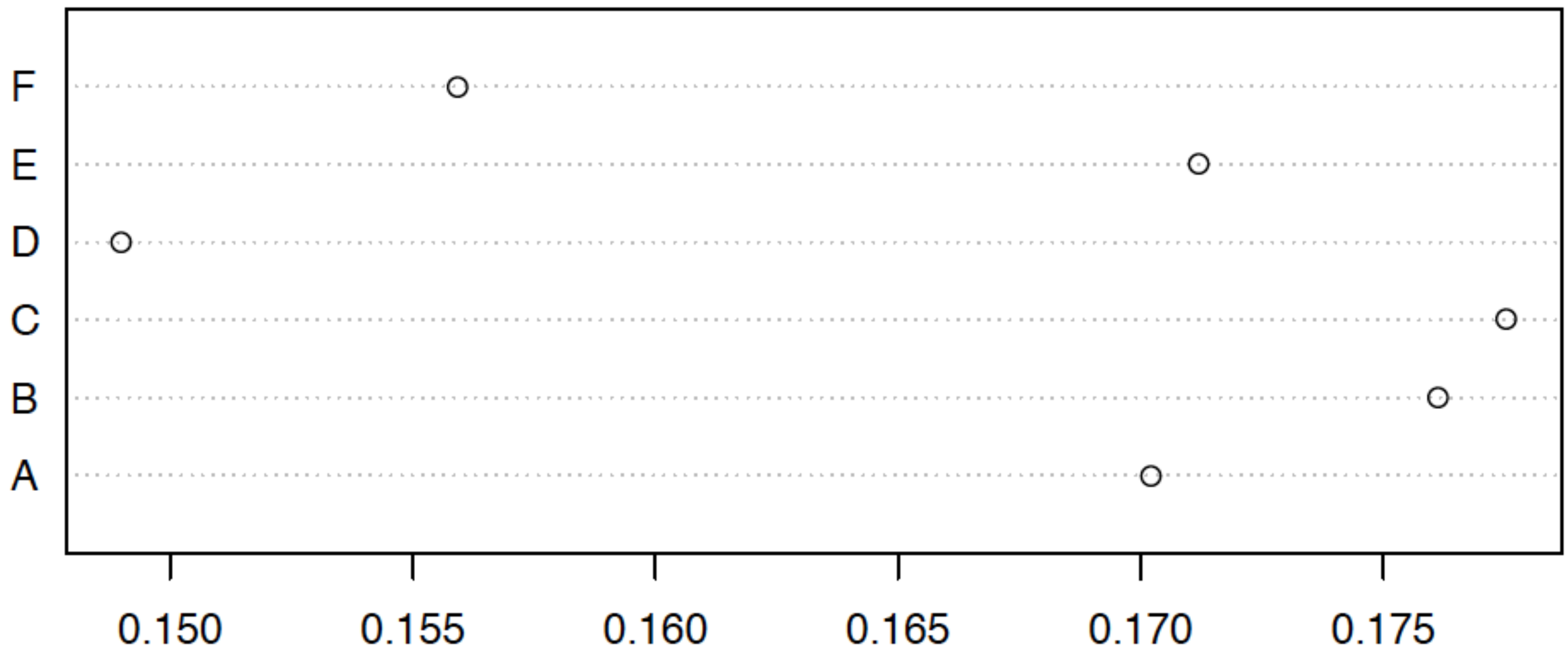
What works well/not so well?

Using area works – reliably, if your `barplot()` starts at zero;



What works well/not so well?

'Position on a common scale' works best



Why does it work well/not so well?

Your vision evolved, primarily, to avoid predators and find food – not to read scientific data.

How many 5's in this list?

08601023903452120458251O
11945492118776654388369 5
937945255947375722930620

- This task requires your conscious *attention*
- Your reader has a limited attention span, and memory – here, the bad presentation will distract/annoy them

Why does it work well/not so well?

Your vision evolved, primarily, to avoid predators and find food – not to read scientific data.

How many 5's in this list?

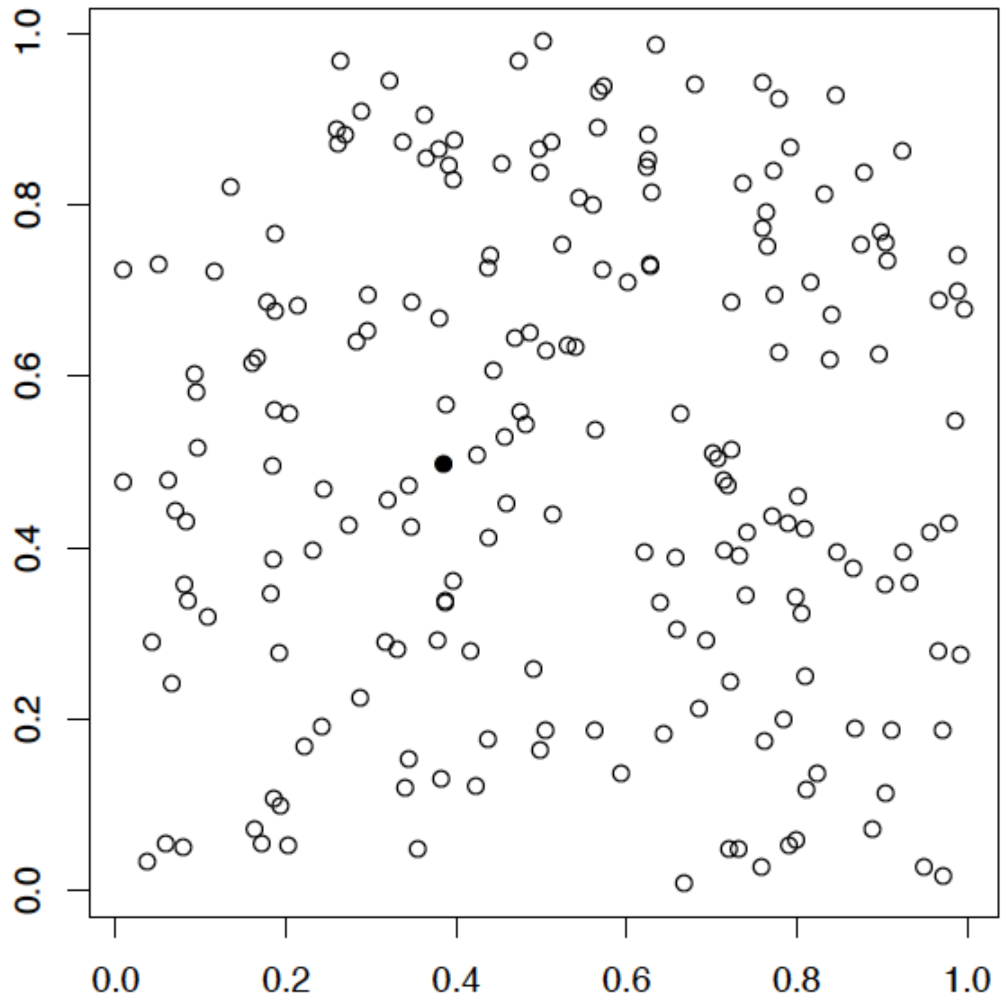
086010239034521204682510
119454921187766543883695
937948255947374722930620

- Some visual signals (e.g. color) are processed *pre-attentively*
- By using these signals, you make comparisons easy, and avoid distracting/annoying your reader with trivialities

Why does it work well/not so well?

Simple color/shading

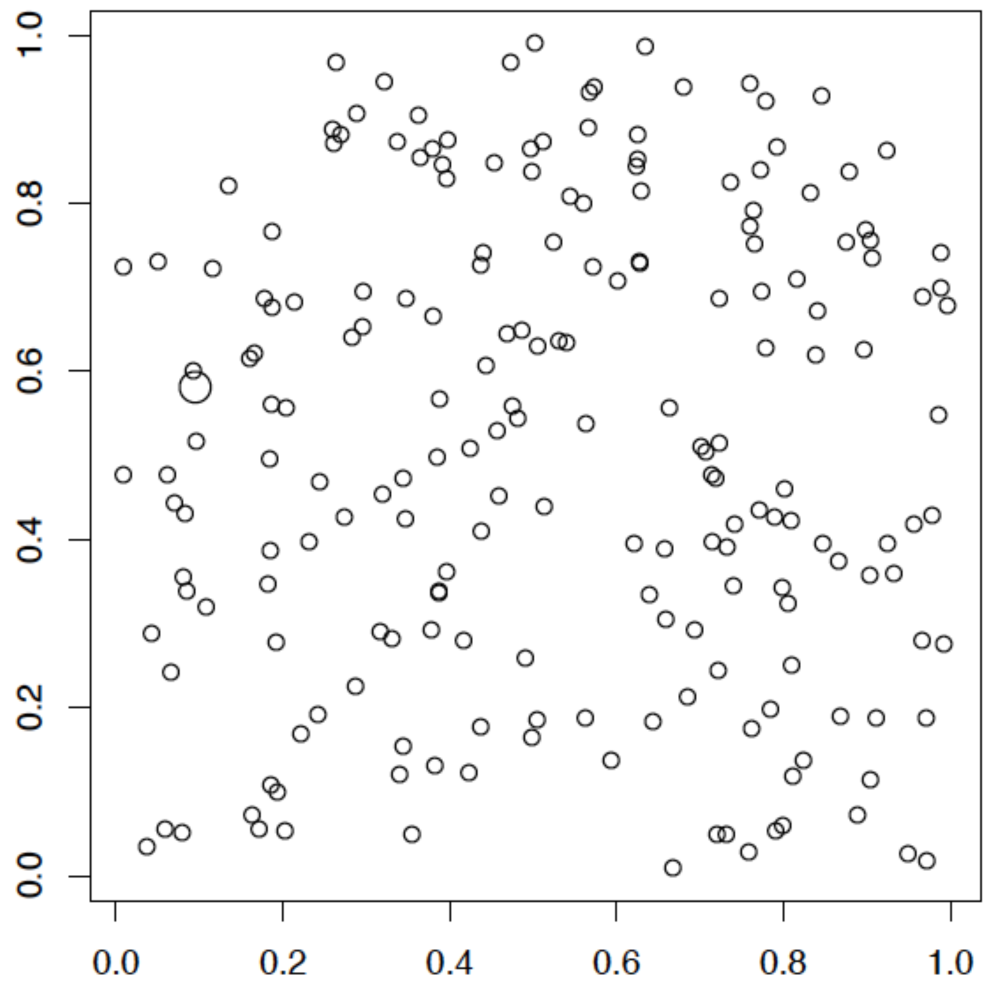
~~Size~~ is also pre-attentive; which point is not like the others?



Why does it work well/not so well?

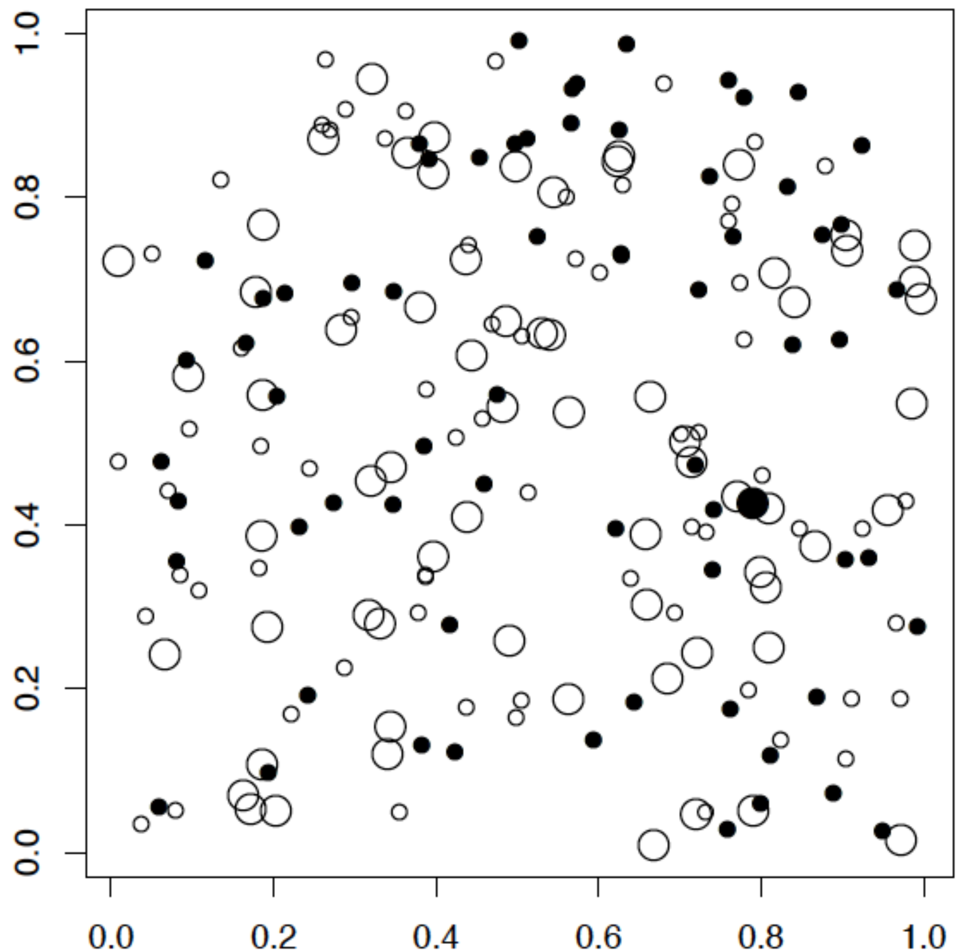
Size

~~Simple color/shading~~ is pre-attentive;



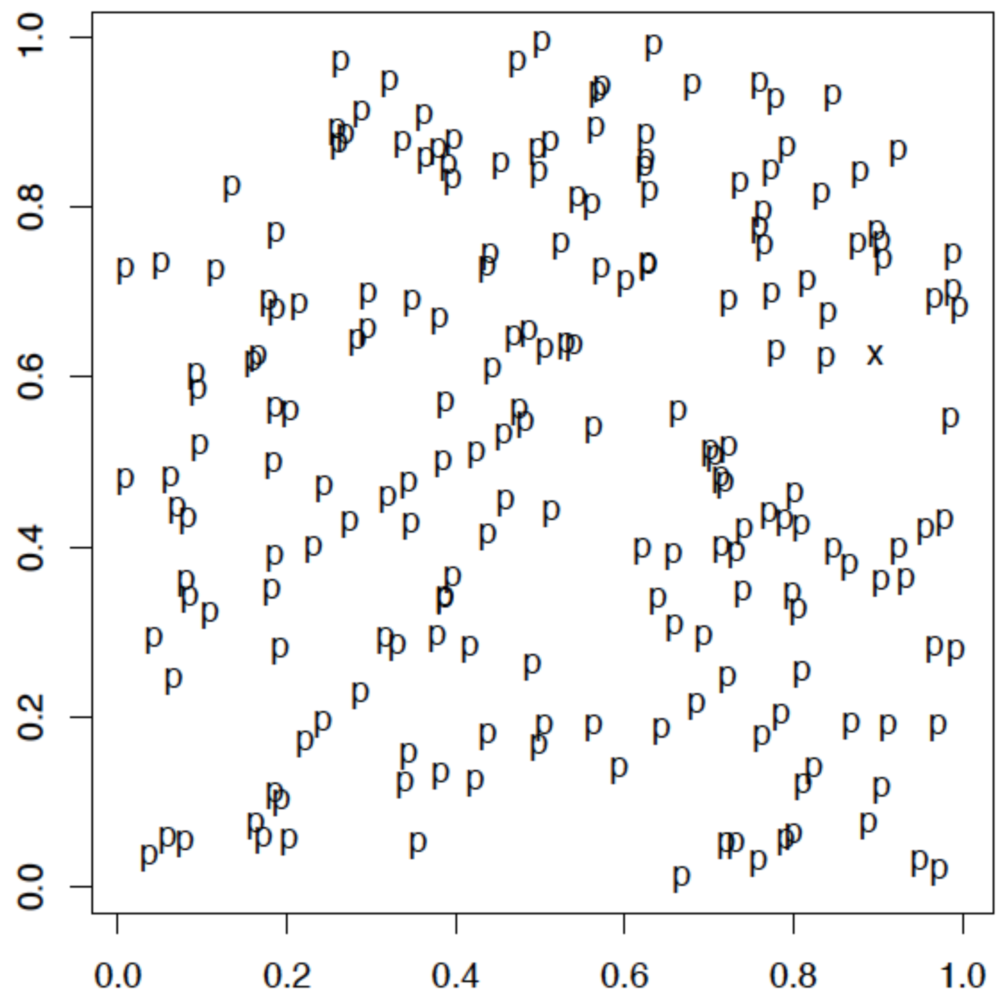
Why does it work well/not so well?

Combining size:color requires a lot more attention; which point is not like the others?



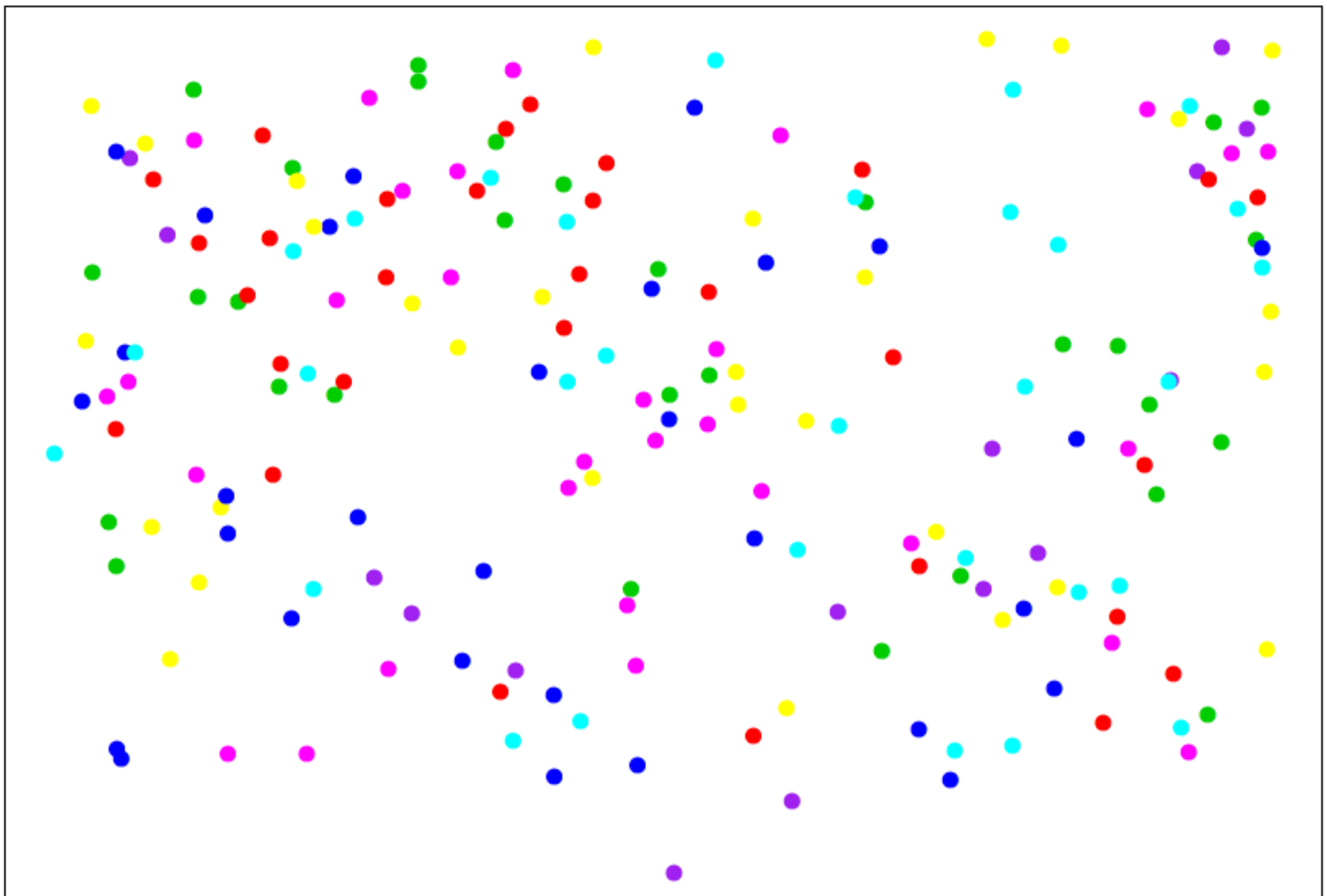
Why does it work well/not so well?

Shapes require lots of attention, unless *very* different;



Why does it work well/not so well?

Too many colors (> 4, say) requires too much attention; what pattern is illustrated here?



Designing graphs: an algorithm

If/when choosing a graph gets difficult;

1. Think, fairly hard, about what you want to illustrate
2. Pick a graph you think codes it appropriately
3. *Explain it to someone* – yourself, at first – like you will do with your poster
4. Iterate steps 2 & 3 until it works

- At step 2, borrowing ideas from other people is **just fine** – there are no prizes for originality
- If you get stuck, ask for help
- Some high-dimensional patterns are *just too complex* for 2D paper... but we rarely have enough data to say much about them

Effective Comparisons: Examples

Some (published!) data on favorite color;

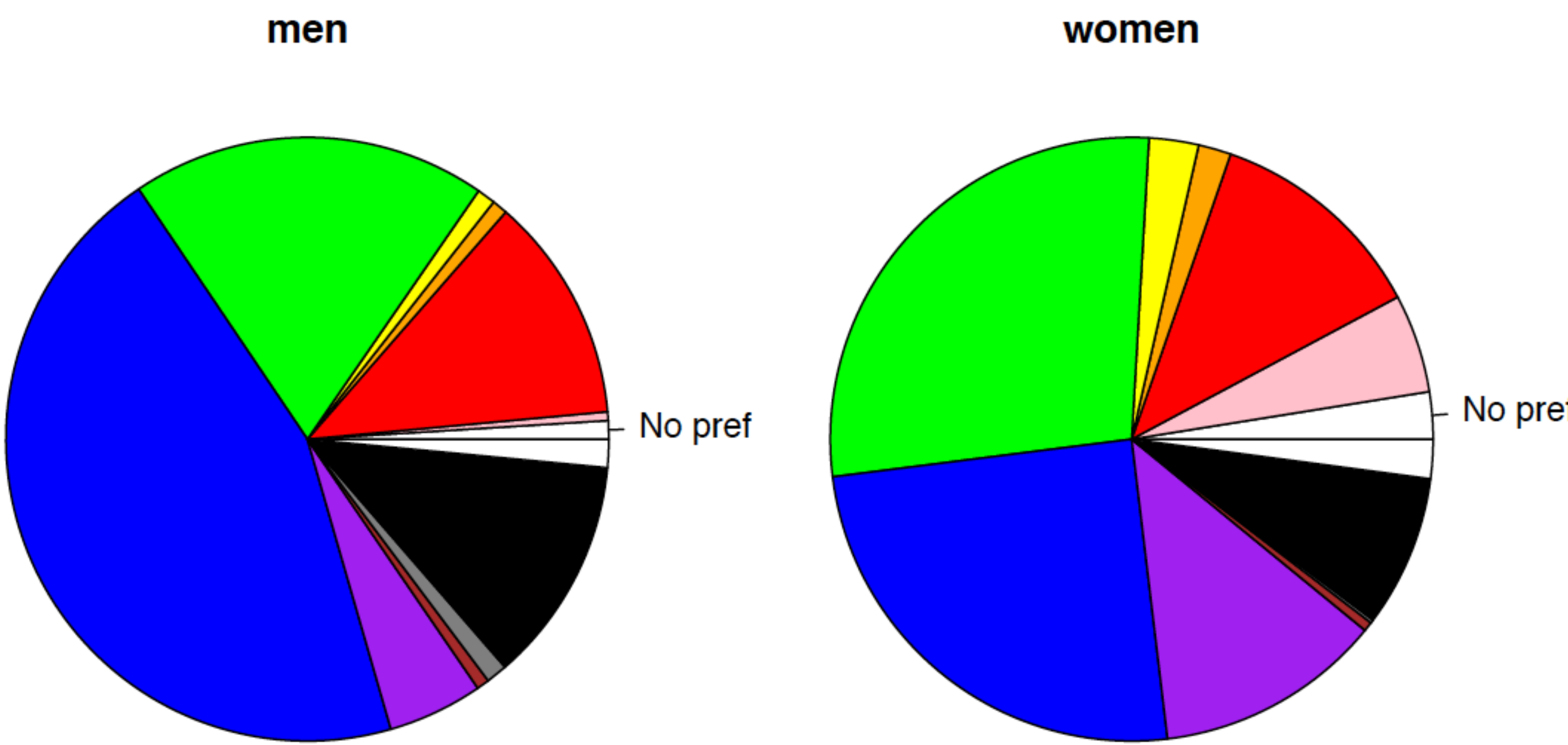
color	M	F	color	M	F
No pref	19	95	blue	866	938
pink	9	199	purple	98	459
red	233	447	brown	13	19
orange	16	66	grey	22	7
yellow	19	100	black	233	306
green	367	1051	white	29	79

Source: Ellis and Ficek, Color preferences according to gender and sexual orientation *Personality and Individual Differences* (2001) 31:8

– who suggest differences are "inclined to suspect the involvement of neurohormonal factors" noting there are "sex differences in retinal biochemistry and in how the brain processes color information".

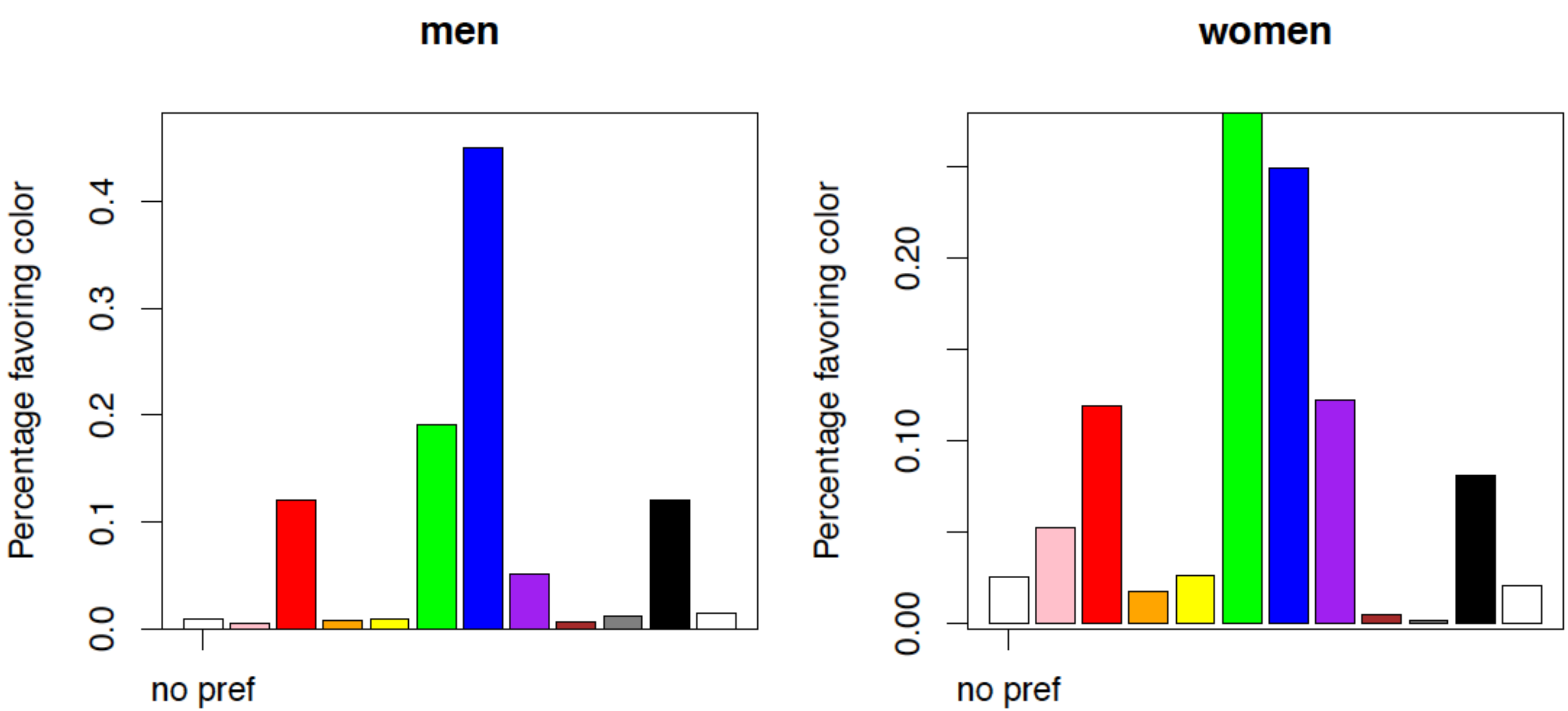
Effective Comparisons: Examples

A first attempt; no intervals, comparisons hard



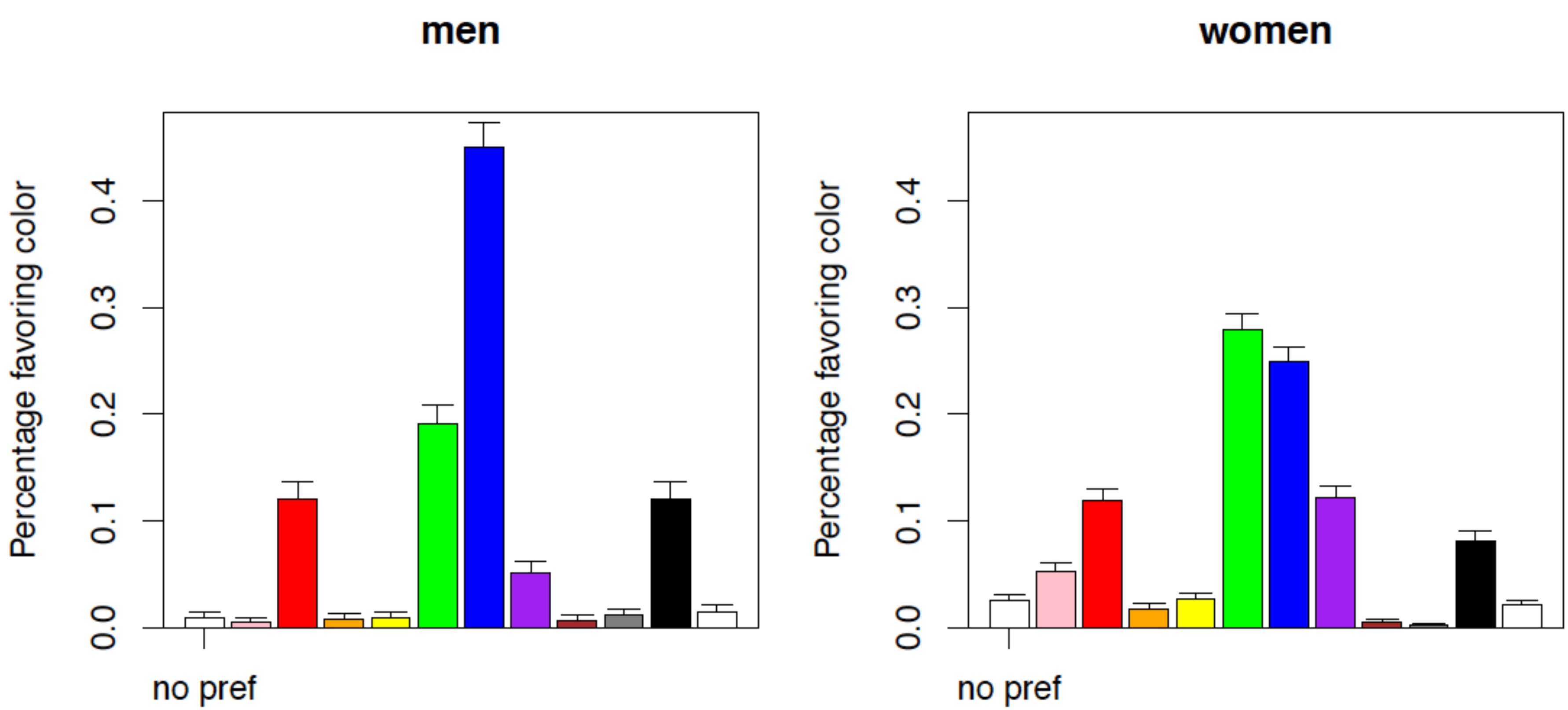
Effective Comparisons: Examples

Using `barplot()` ;



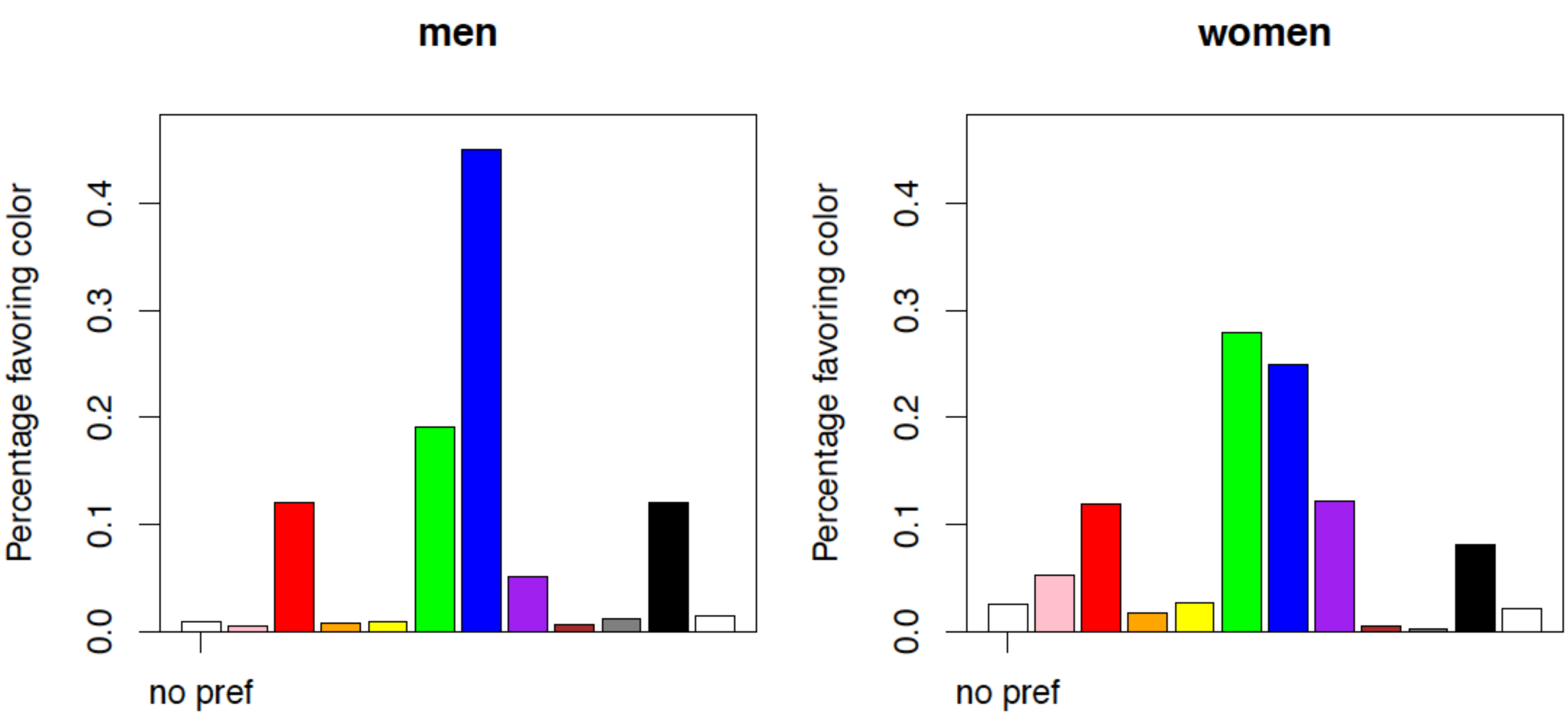
Effective Comparisons: Examples

Tower blocks with antennae – use `segments()`



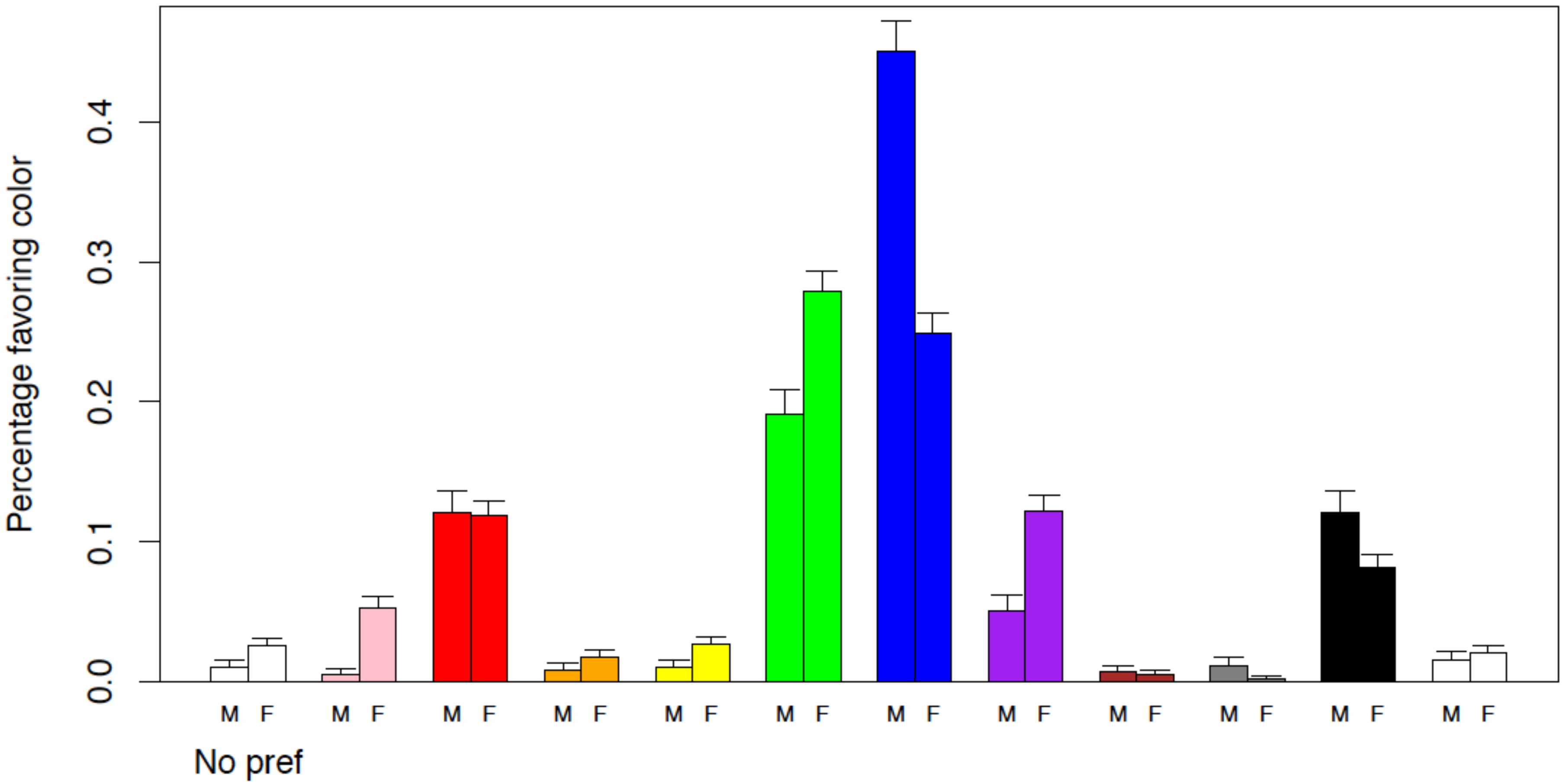
Effective Comparisons: Examples

Using `barplot()` and a common scale;



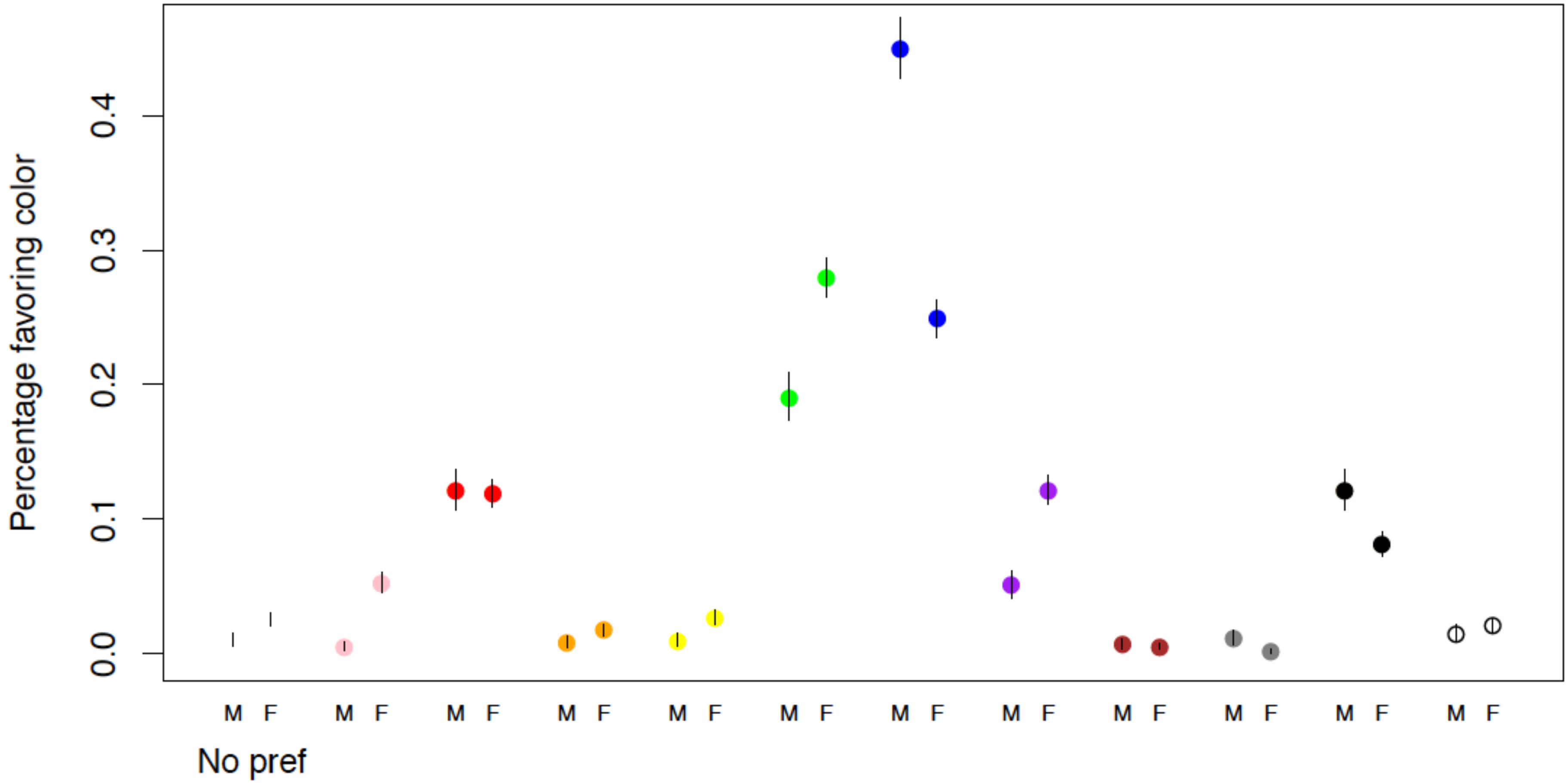
Effective Comparisons: Examples

Tower blocks with antennae; what do we compare?



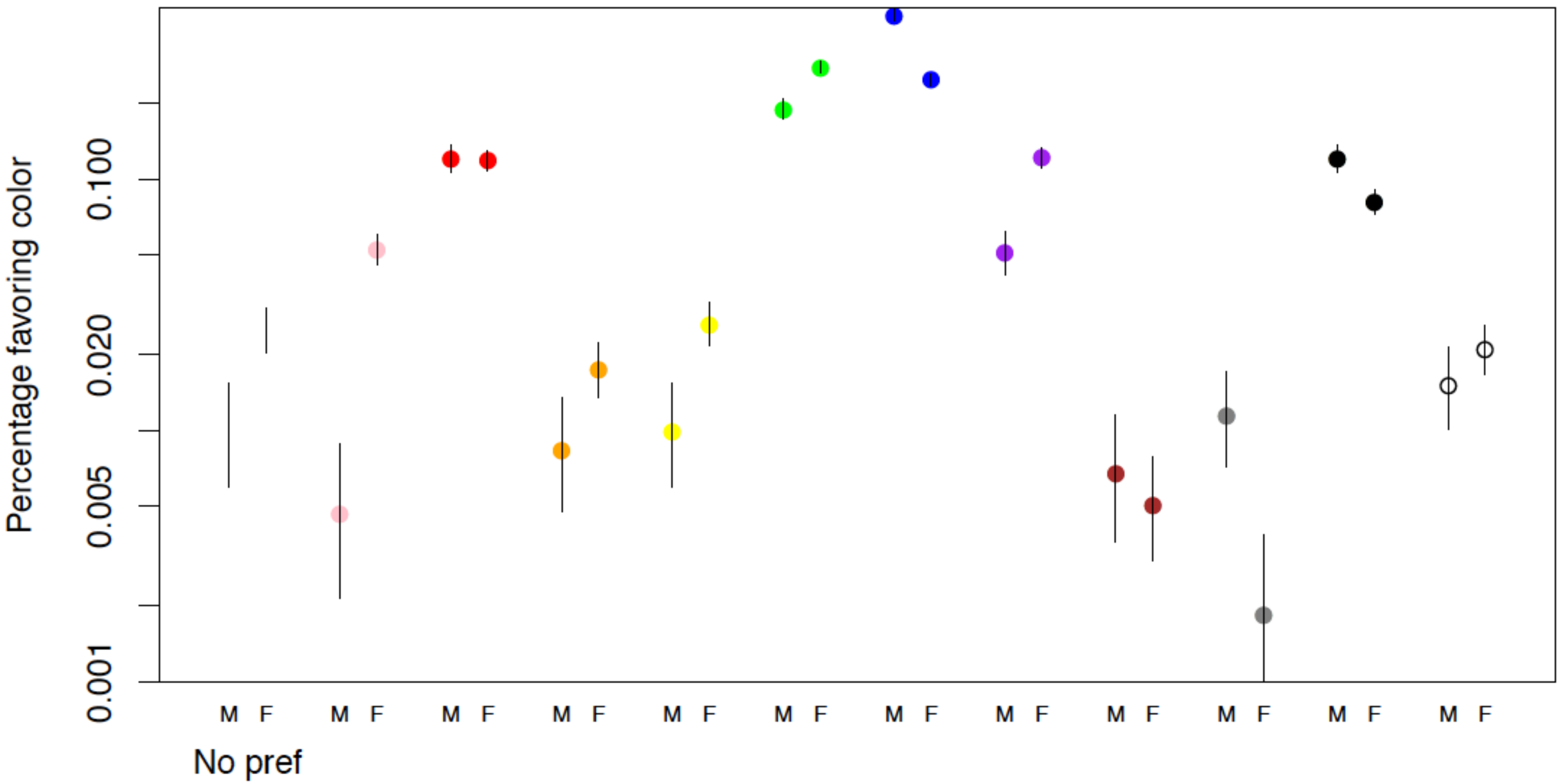
Effective Comparisons: Examples

Dump the blocks; ‘position on common scale’



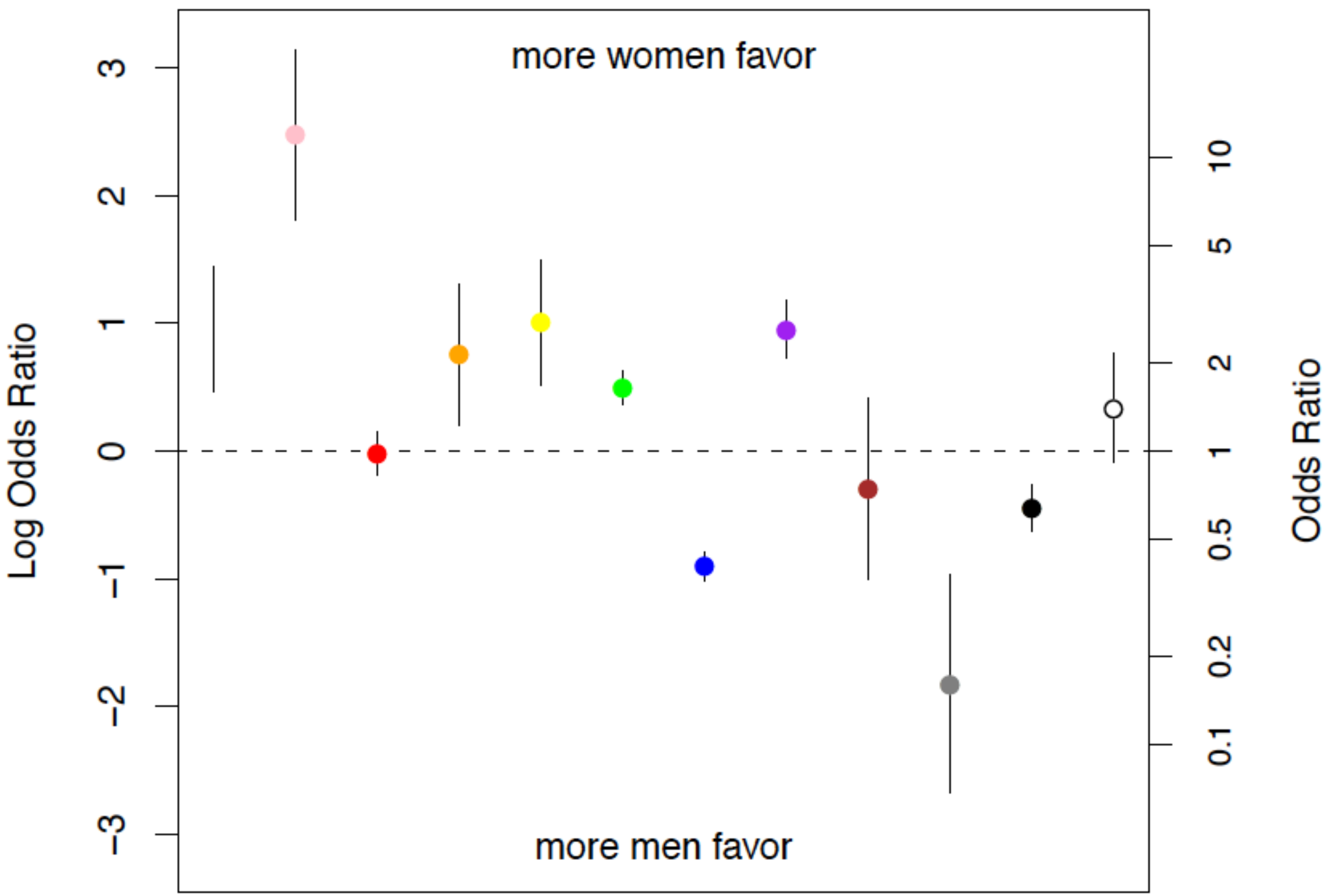
Effective Comparisons: Examples

Stresses unpopular colors;



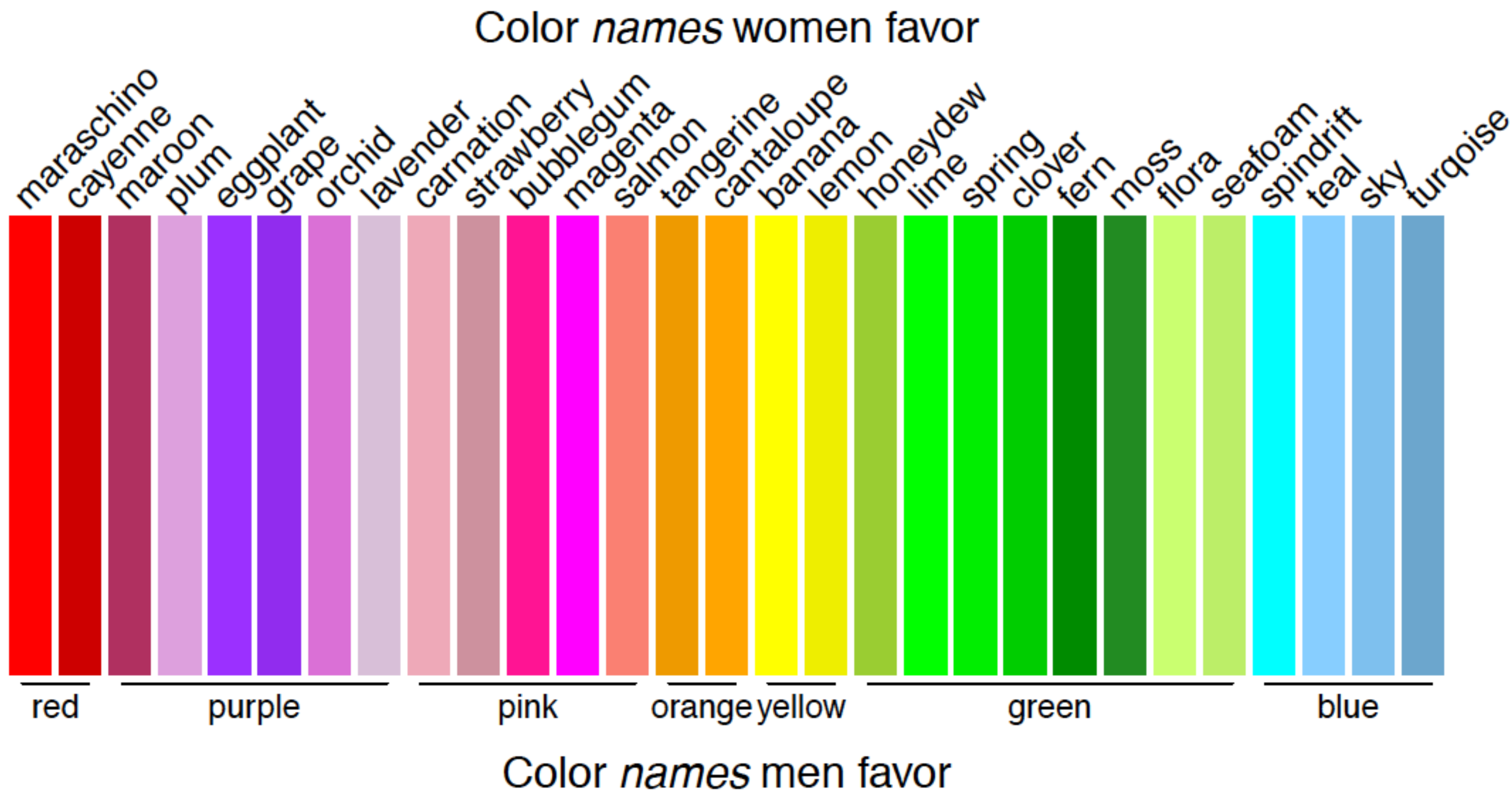
Effective Comparisons: Examples

Stresses only differences; (baseline group irrelevant)



Effective Comparisons: Examples

And finally;



Effective Comparisons: Examples

Some lessons from all that;

- Decide what you want to compare; differences or absolute values?
- Often it will be differences – recall plotting residuals for model-checking, not data
- If you want to compare items, put them beside each other
- Minimalist representations (e.g. use of points not areas) are aesthetically 'clean' – and permit e.g. confidence intervals
- Plots will/should evolve, as you decide to stress different results
- Pie charts are rarely useful

Tufte's theory of data graphics

The 'minimalism' approach is formalized by Tufte, in his principles* for better graphics;

- Above all else, show the data

- Maximize the data-ink ratio (i.e. data ink / total ink)
- Erase non-data-ink (*chartjunk*)
- Erase redundant data-ink
- Revise and edit

Let's apply these, for another small dataset;

* … they are a *little* controversial – so think of them as 'guidelines'

Tufte theory: Berkeley 1973 data

In 1973, sex discrimination was suspected in admission to Berkeley;

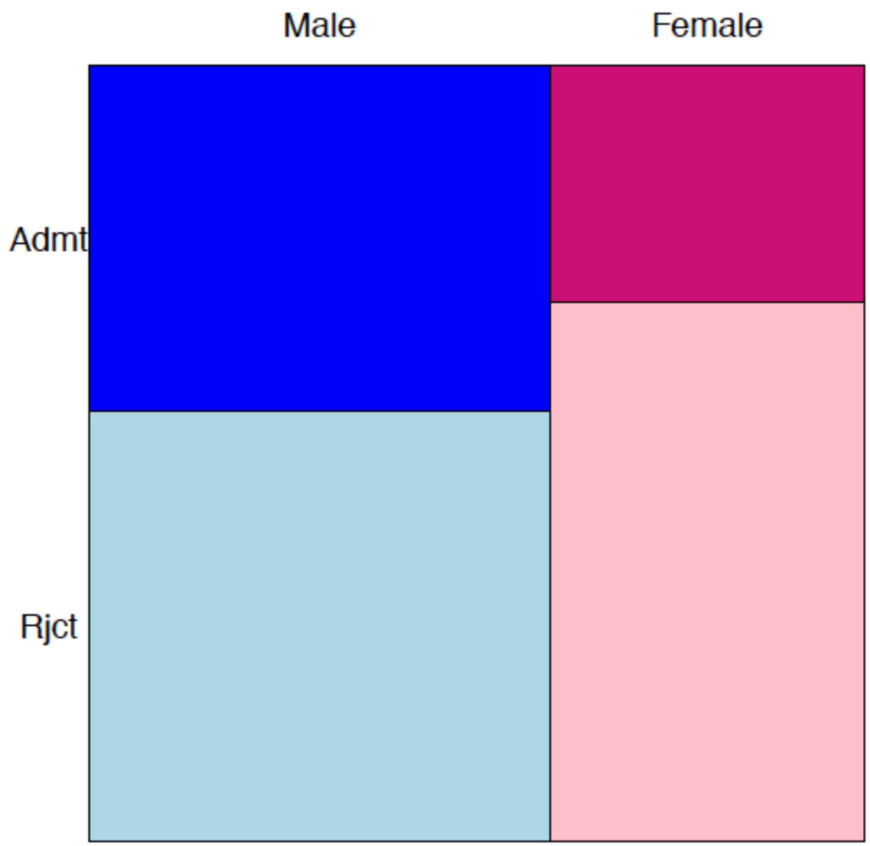
Dept	Men		Women	
	n	Admit	n	Admit
A	825	0.62	108	0.82
B	560	0.63	25	0.68
C	325	0.37	593	0.34
D	417	0.33	375	0.35
E	191	0.28	393	0.24
F	373	0.06	341	0.07
Total	2691	0.45	1835	0.30

– the ‘headlines’ compared 45% to 30%.

How can we turn this table into a graph?

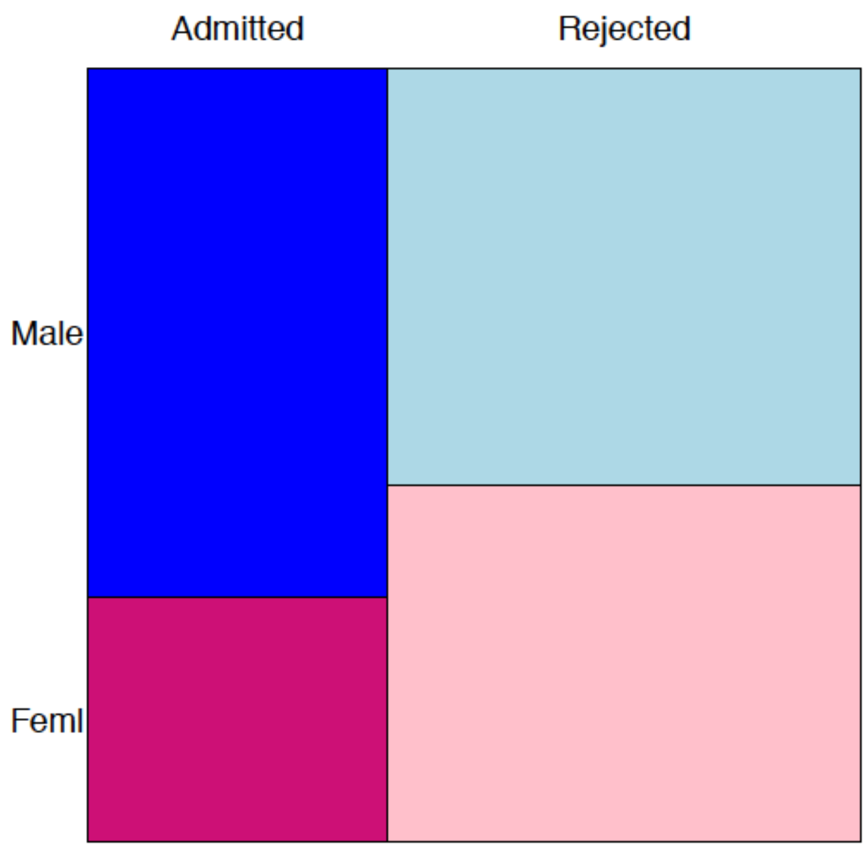
Tufte theory: Berkeley 1973 data

Mosaic plots are a fairly ‘old school’ method...



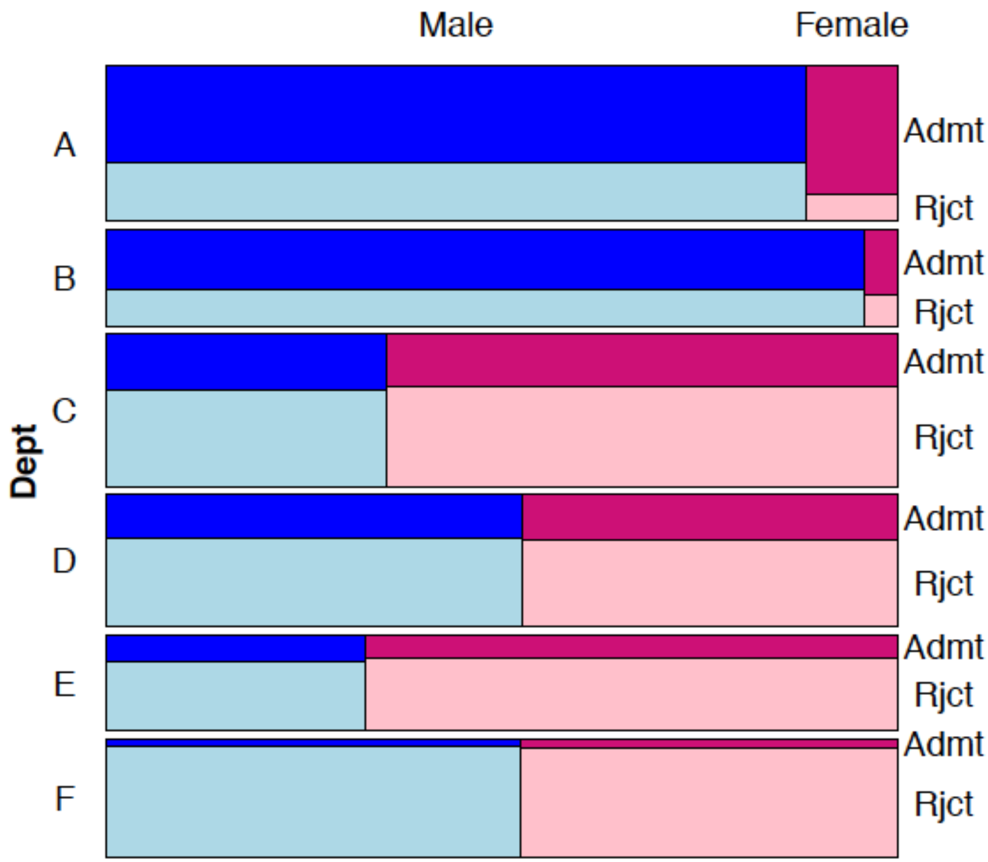
Tufte theory: Berkeley 1973 data

... where conditioning matters;



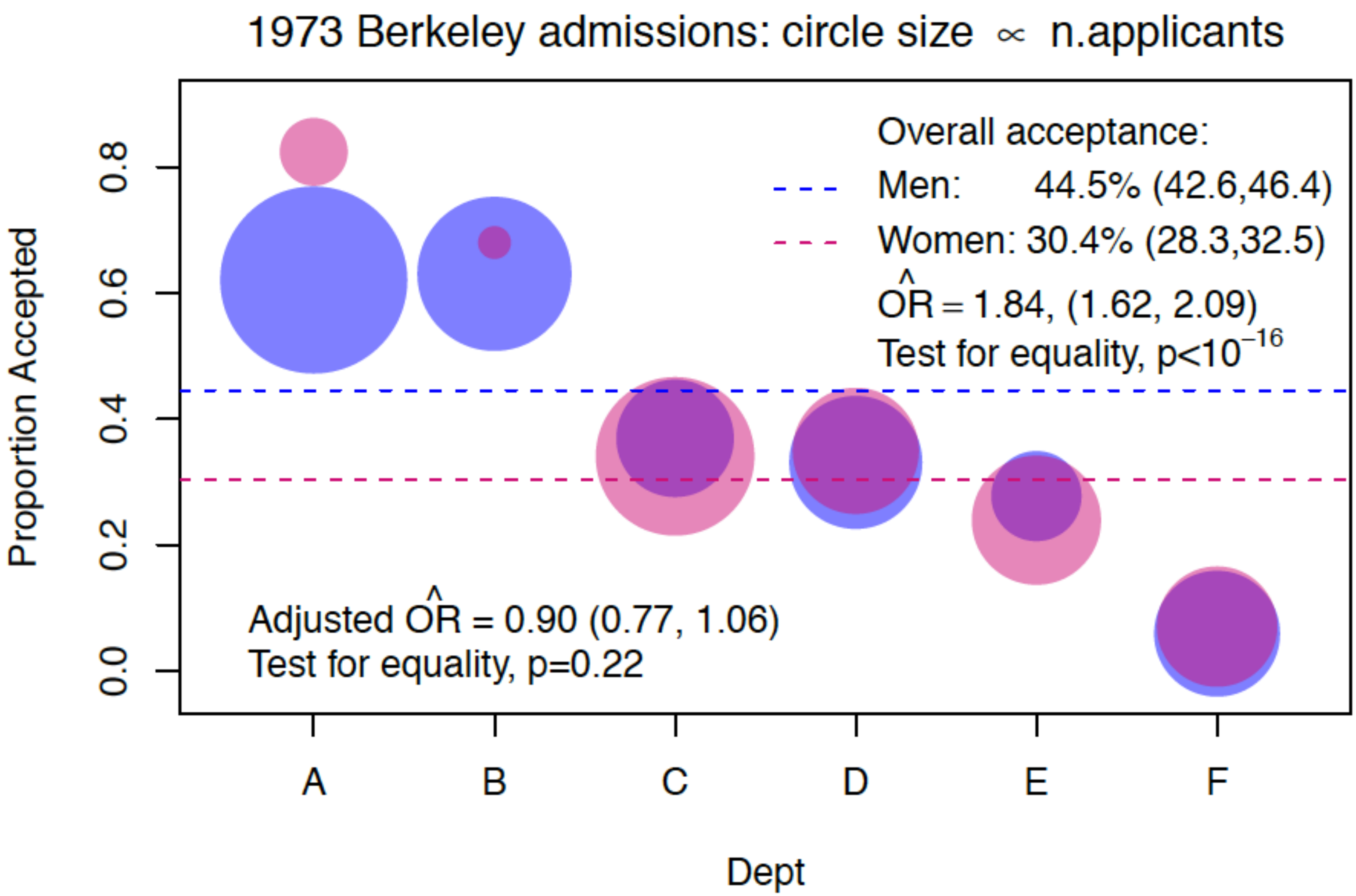
Tufte theory: Berkeley 1973 data

Broken down by department...



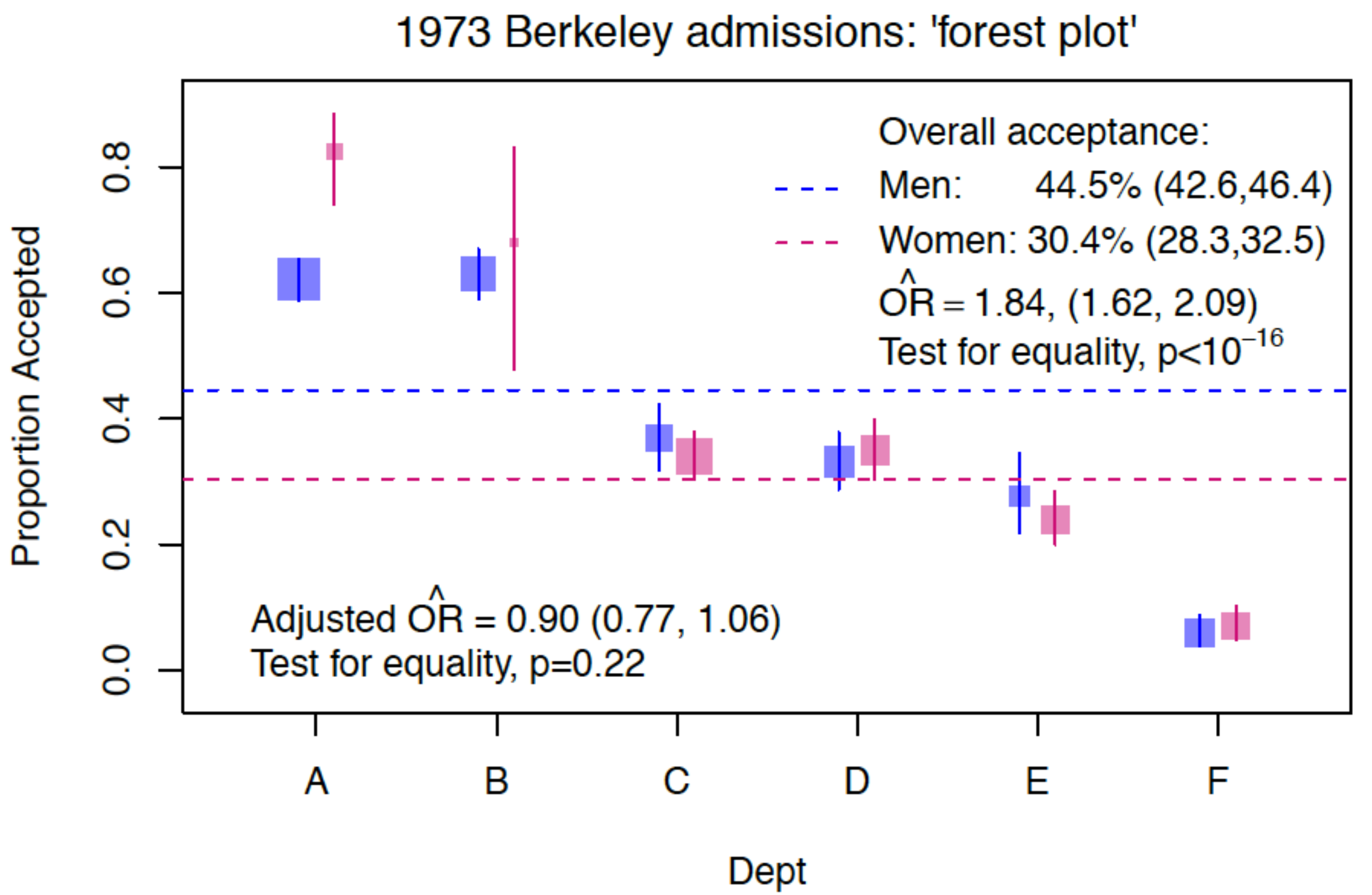
Tufte theory: Berkeley 1973 data

Recall 'position on a common scale'/Tufte;



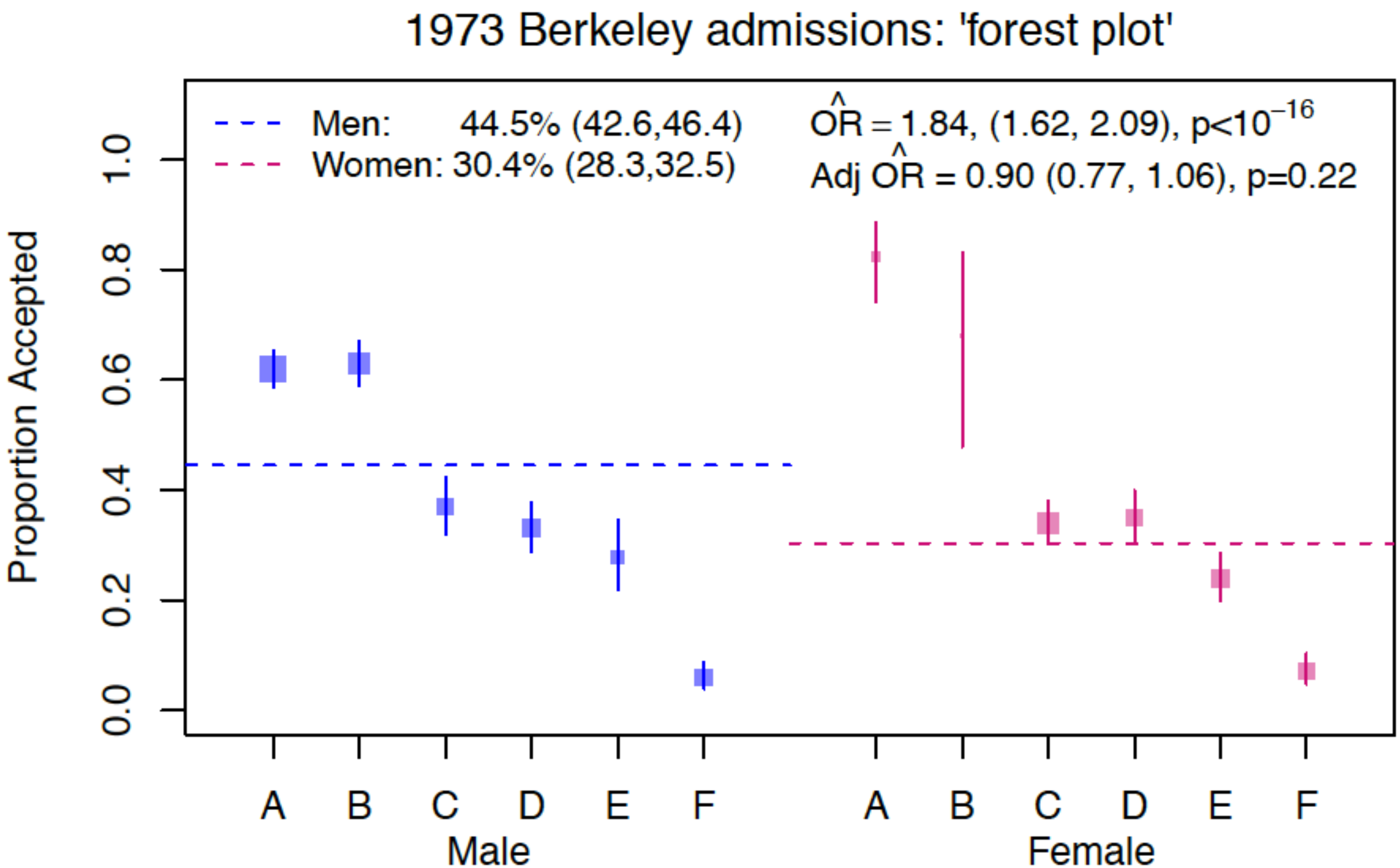
Tufte theory: Berkeley 1973 data

Less ink – confounding less obvious



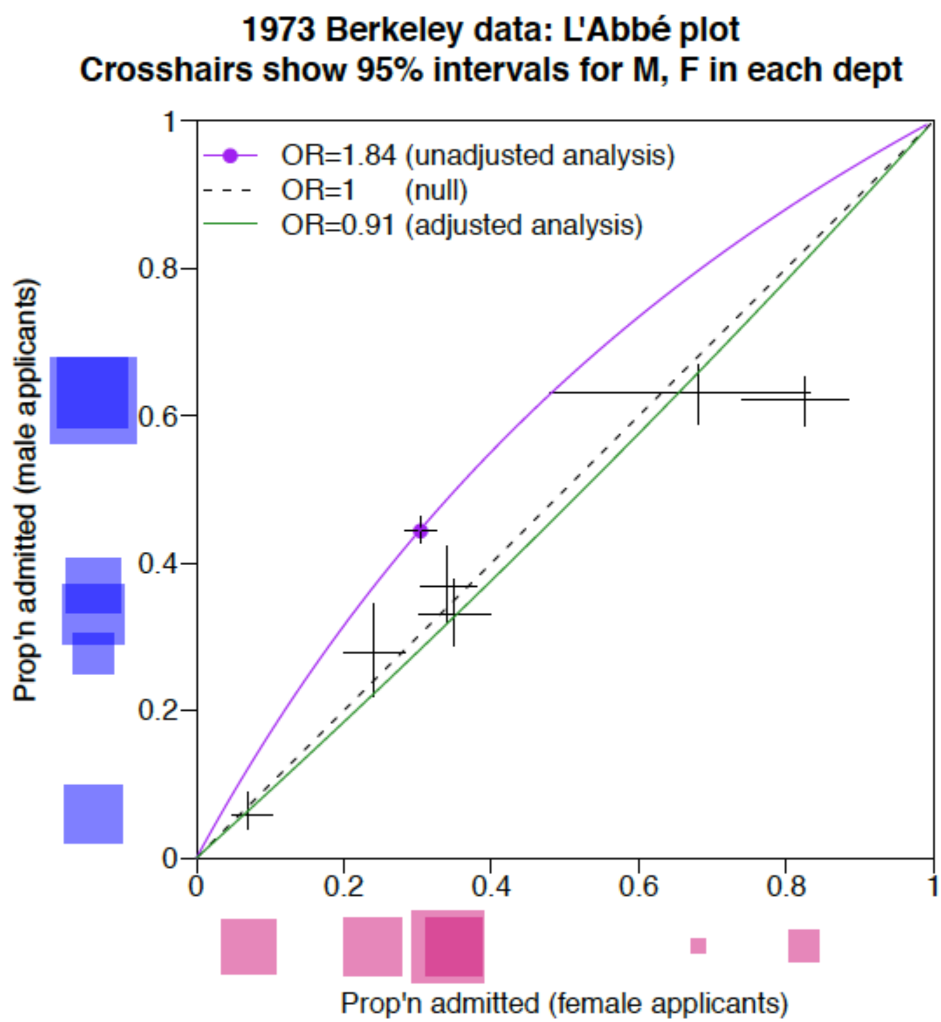
Tufte theory: Berkeley 1973 data

Berkeley-wide comparison of admittance;



Tufte theory: Berkeley 1973 data

Remove irrelevant A/B/C ordering;



Tufte theory: Berkeley 1973 data

In summary;

- Counts may not be helpful, if you want to compare proportions
- L'Abbé plots (the last version) are a good way to illustrate *just* the proportions, in two groups – although they are unfamiliar to some audiences
- Non-collapsibility was for *decades* viewed as weird and non-intuitive – see "Simpson's paradox". With the right graph it's straightforward to see it happening

Large datasets

You have to deal with "Big Data" – Bell Labs in the 70s did not

- Consider conditioning plots for (up to) 4-dimensional relationships – the `lattice` package has many options of 'small multiples'
- Transparent colors and/or hexagonal binning are useful
- Parallel co-ordinate plots make sense in higher dimensions, when the axes can be ordered in some sensible manner.

For higher dimensions, you can also 'fly through' projections of the data onto 2D, but this requires animation; see the `ggobi` software.

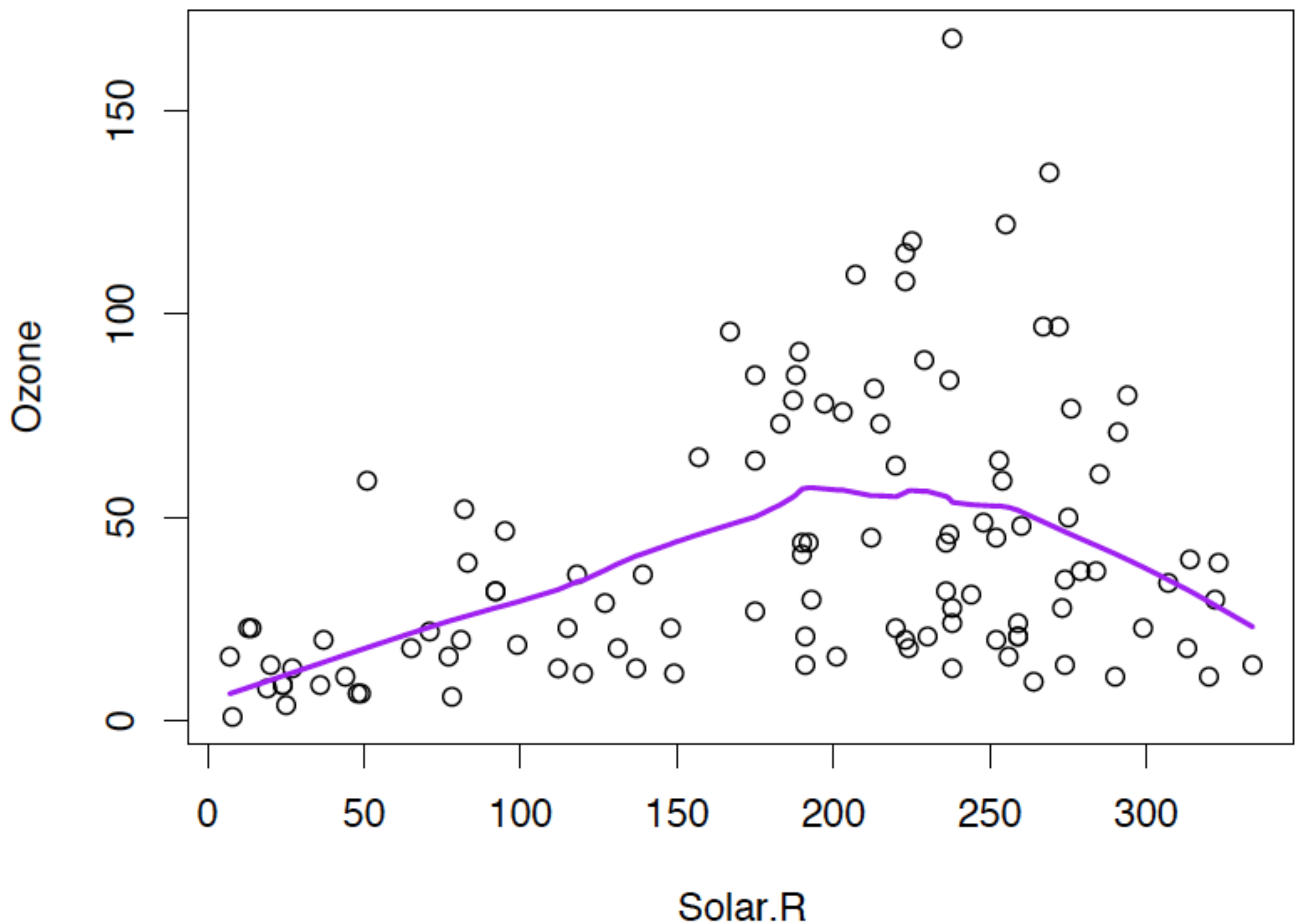
Conditioning plots

A typical goal for measuring Z is to see whether the $Y-X$ relationship changes at different values of Z. For example, we might want to see if a Blood Pressure/genotype association varies by Body Mass Index (weight/height^2)

In this case, it's useful to show plots of Y against X conditioned on the value of Z, i.e. Y versus X for all data with Z in a small range. This is known as a *conditioning plot*, and can be produced with `coplot()`.

Ozone is a *secondary pollutant*, it is produced from organic compounds and atmostpheric oxygen in reactions catalyzed by nitrogen oxides and powered by sunlight. But looking at ozone concentrations in NY in summer (Y) we see a non-monotone relationship with sunlight (X) ...

Conditioning plots

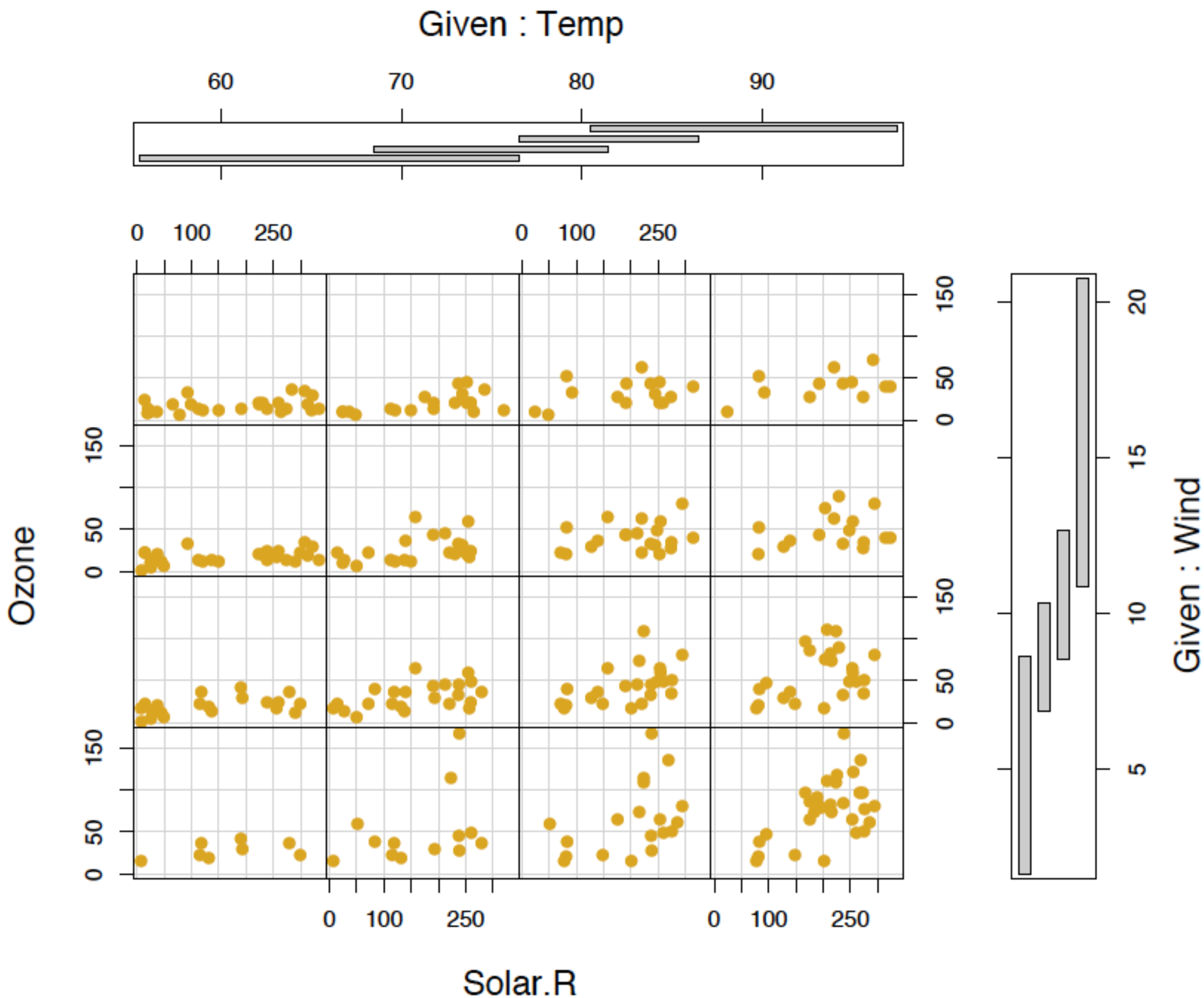


Conditioning plots

Now draw a scatterplot of `Ozone` vs `Solar.R` for various subranges of `Temp` and `Wind`. (For more examples like this, see the commands in the `lattice` package.)

```
data(airquality)
coplot(Ozone ~ Solar.R | Temp * Wind, number = c(4, 4),
     data = airquality,
    pch = 21, col = "goldenrod", bg = "goldenrod")
```

Conditioning plots

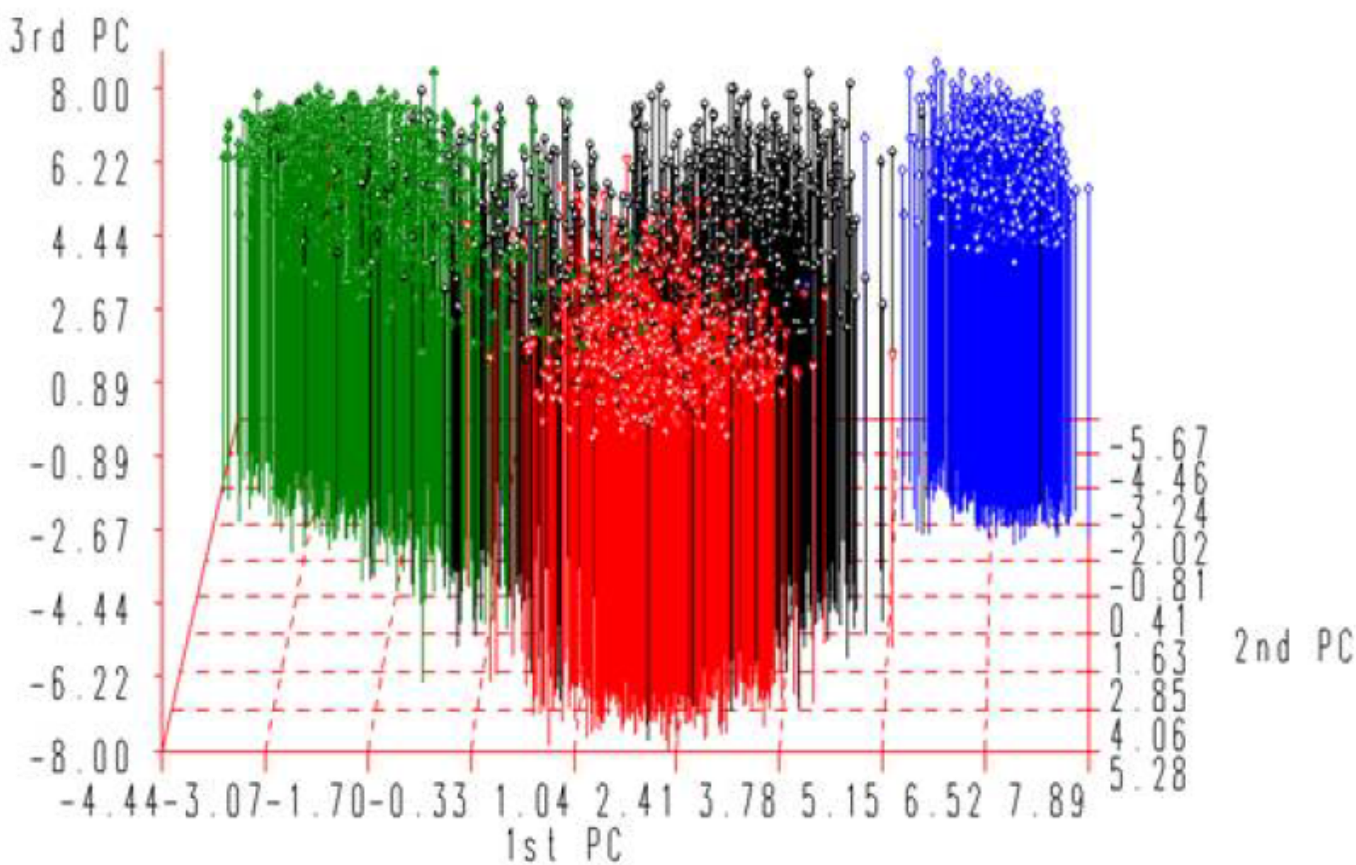


Conditioning plots

- A 4-D relationship is illustrated; the Ozone/sunlight relationship changes in strength depending on both the Temperature and Wind
- The horizontal/vertical 'shingles' tell you which data appear in which plot. The overlap can be set to zero, if preferred
- `coplot()`'s default layout is a bit odd; try setting `rows`, `columns` to different values
- For more plotting commands that support conditioning, see `library(help="lattice")`

More than 3,4 dimensions

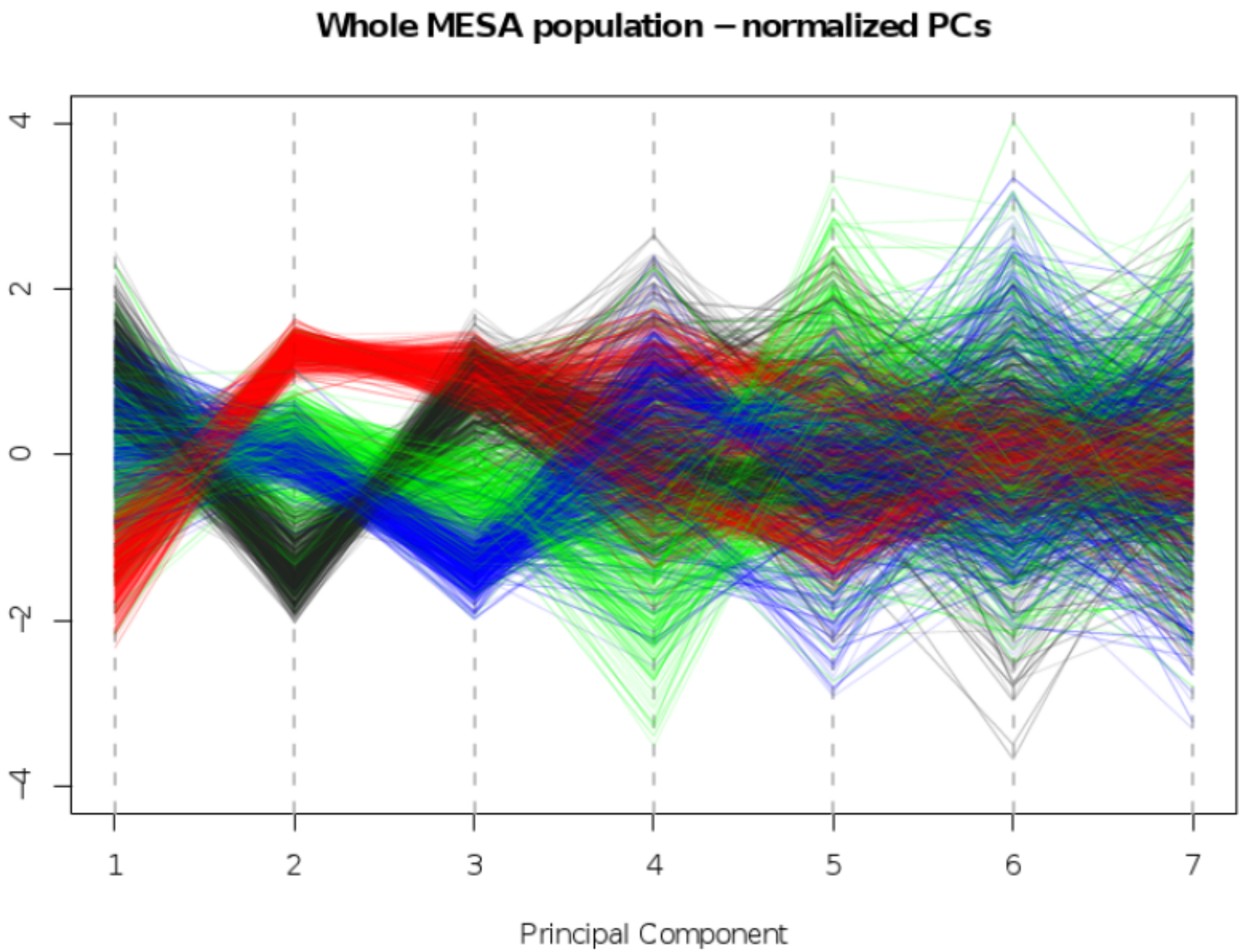
Even 3 dimensions comes out pretty badly in fake 3D;



Self–reported ancestry: Hispanic–American ▽ European–American * Chinese–American ♁ African–American ♁

More than 3,4 dimensions

Parallel co-ordinates plots show multiple *ordered* values; each data point is represented by a line;



Transparency

The colors in the last examples were *transparent*. As well as specifiying e.g. `col=2` or `col="red"`, you can also specify

`col="#FF000033"`

– coded as RRGGBB in hexadecimal, with transparency 33 (also hexadecimal). This is a 'pale' red – 33/FF $\approx$ 20%.

Get from color names to RGB with `col2rgb()`, and from base 10 to base 16 using `format(as.hexmode(11), width=2)`

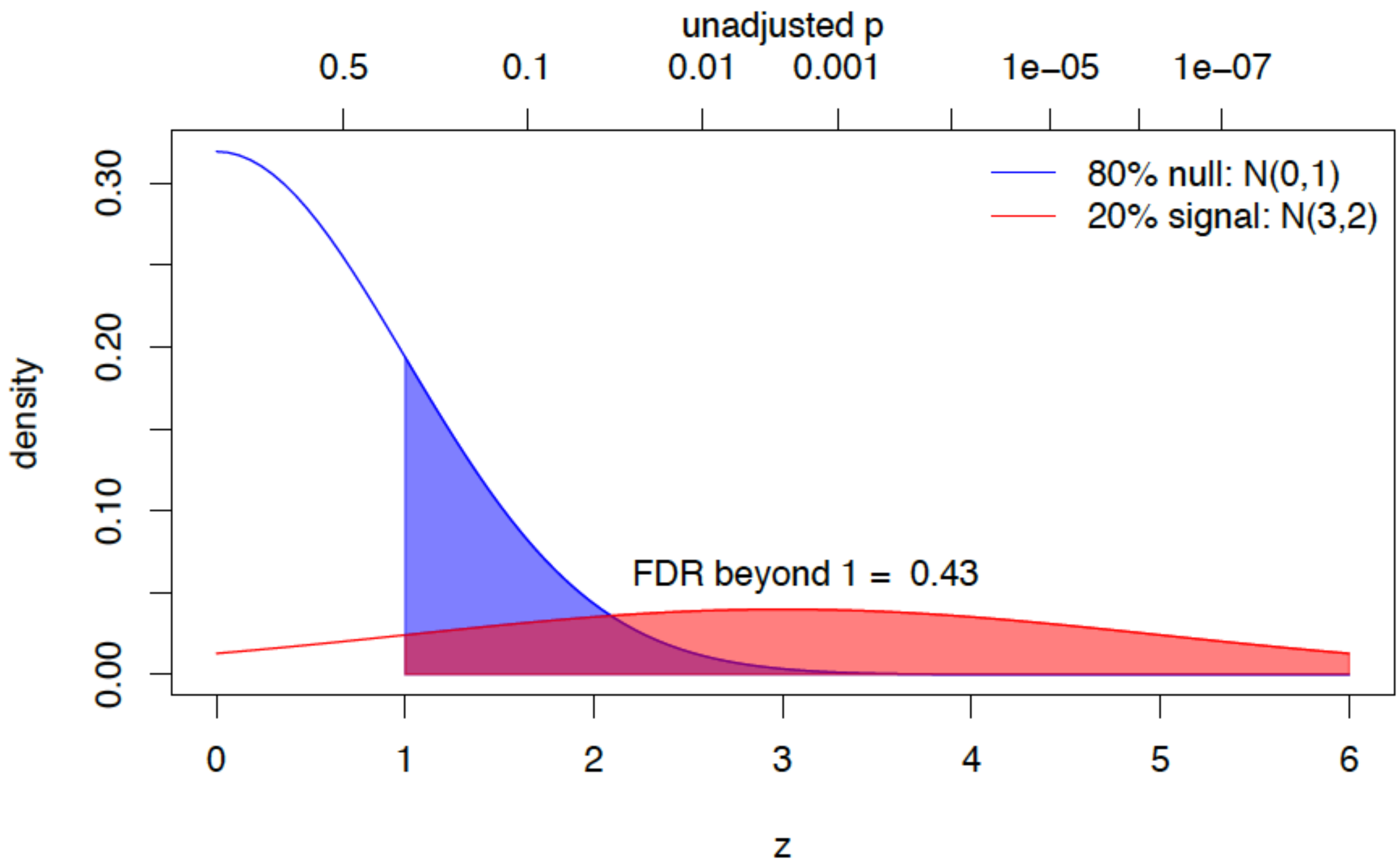
Transparency

R code for another; (also shows other graphics commands)

```
curve(0.8*dnorm(x), 0, 6, col="blue", ylab="density", xlab="z")
curve(0.2*dnorm(x,3,2), 0, 6, col="red", add=T)

xvals <- seq(1, 6, l=101)
polygon(
c(xvals,6,1), c(0.8*dnorm(xvals), 0,0),
density=NA, col="#0000FF80" ) # transparent blue
polygon(
c(xvals,6,1), c(0.2*dnorm(xvals,3,2), 0,0),
density=NA, col="#FF000080" ) # transparent red

legend("topright", bty="n", lty=1, col=c("blue","red"),
c("80% null: N(0,1)", "20% signal: N(3,2)"))
axis(3, at=qnorm(c(0.25, 0.5*10^(-1:-7)), lower=F), c(0.5, 10^(-1:-7)) )
mtext(side=3, line=2, "unadjusted p")

text(2.2, 0.07, adj=c(0,1), paste("FDR beyond 1 = ",
round(0.8*pnorm(1,lower=F)/(0.8*pnorm(1,lower=F) + 0.2*pnorm(1,3,2,lower=F)),3)))
```

Transparency

Here's the output;



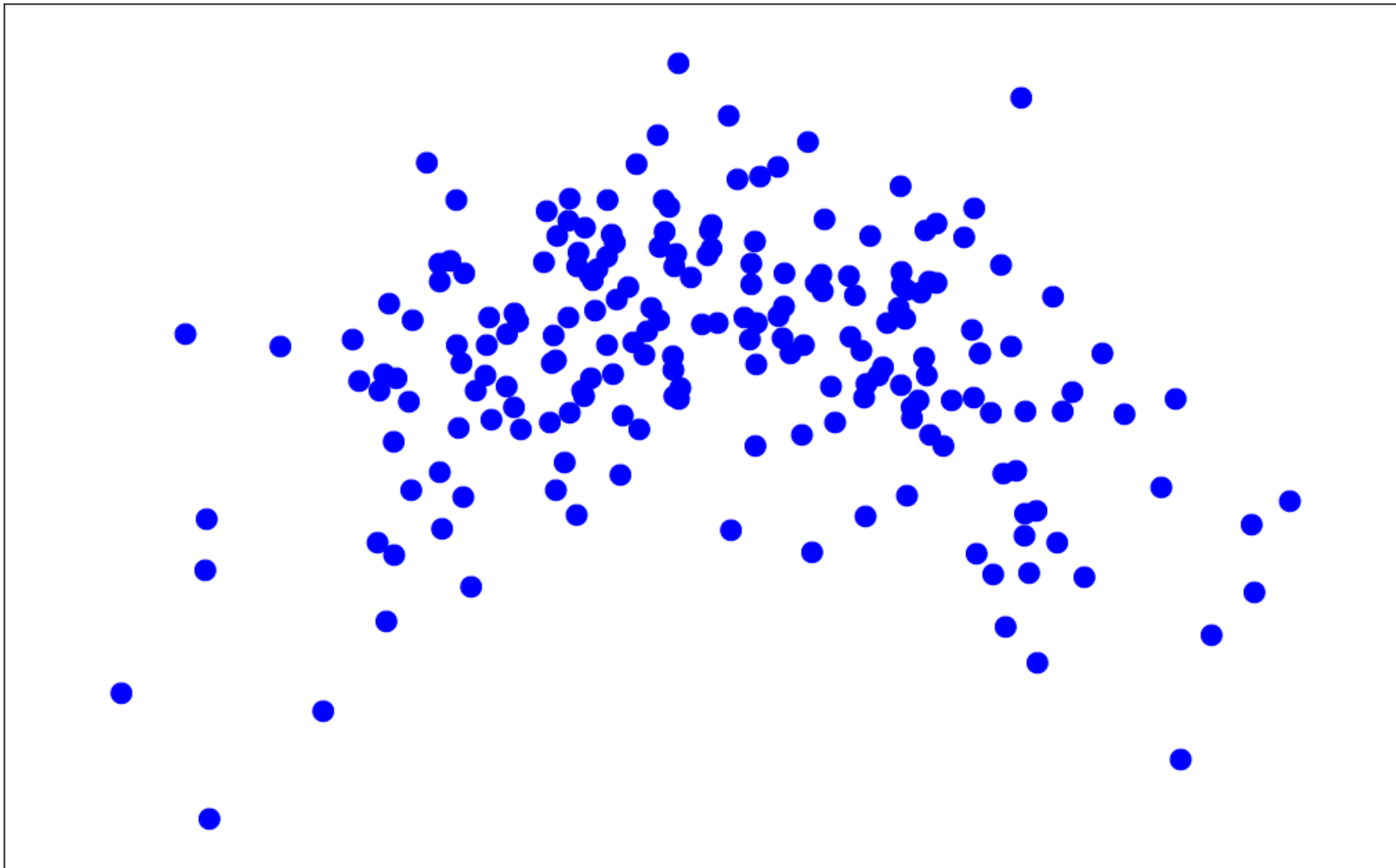
Hexagonal binning

Using transparent plotting symbols is a quick-and-dirty way to adapt scatterplots for use with large datasets.

A better method is 'hexagonal binning'; this is a 2D analog of a histogram – where you would count the number of data in one area, and then draw a bar with height proportional to that count.

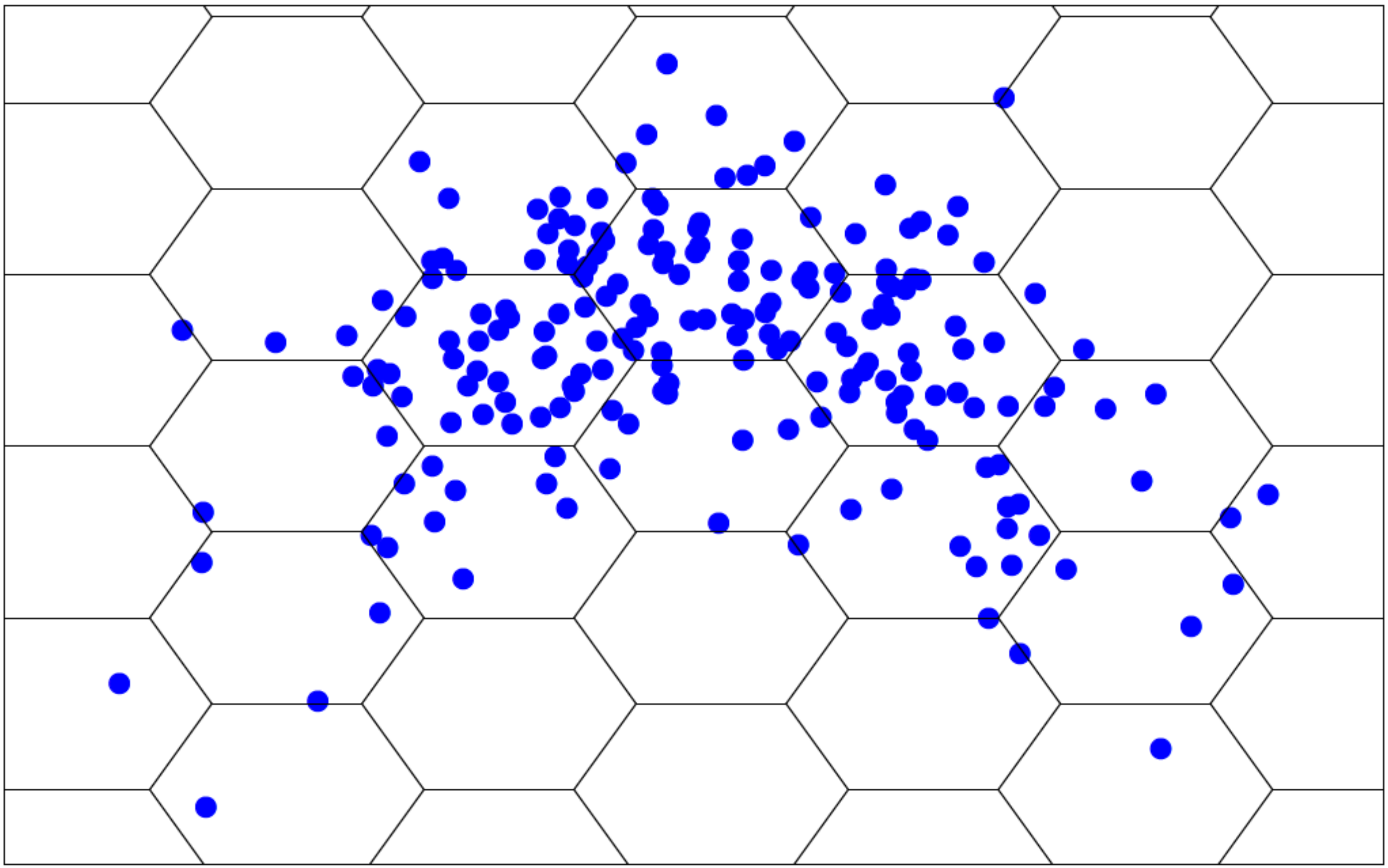
Hexagonal binning

Binning in two dimensions;



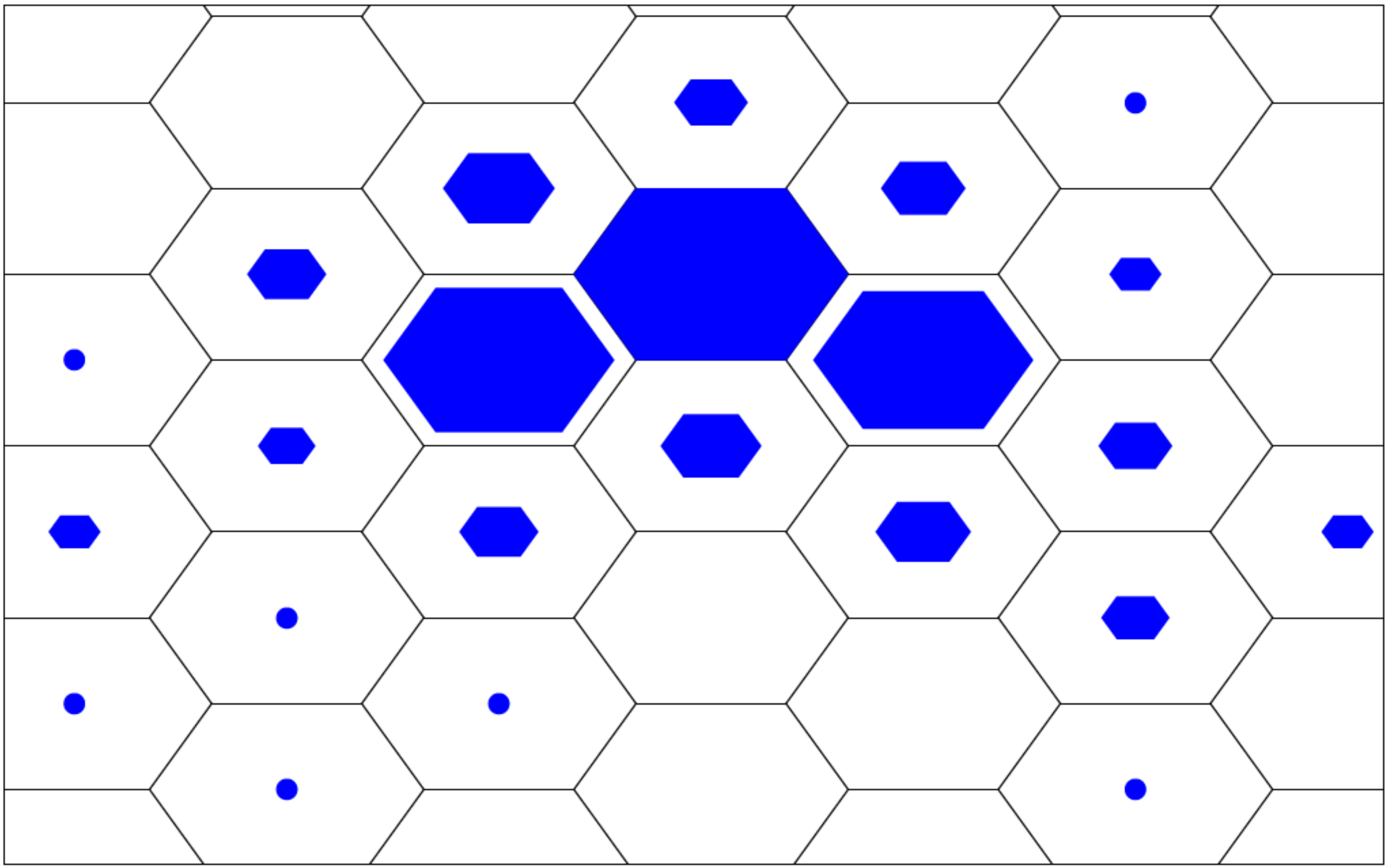
Hexagonal binning

Binning in two dimensions;



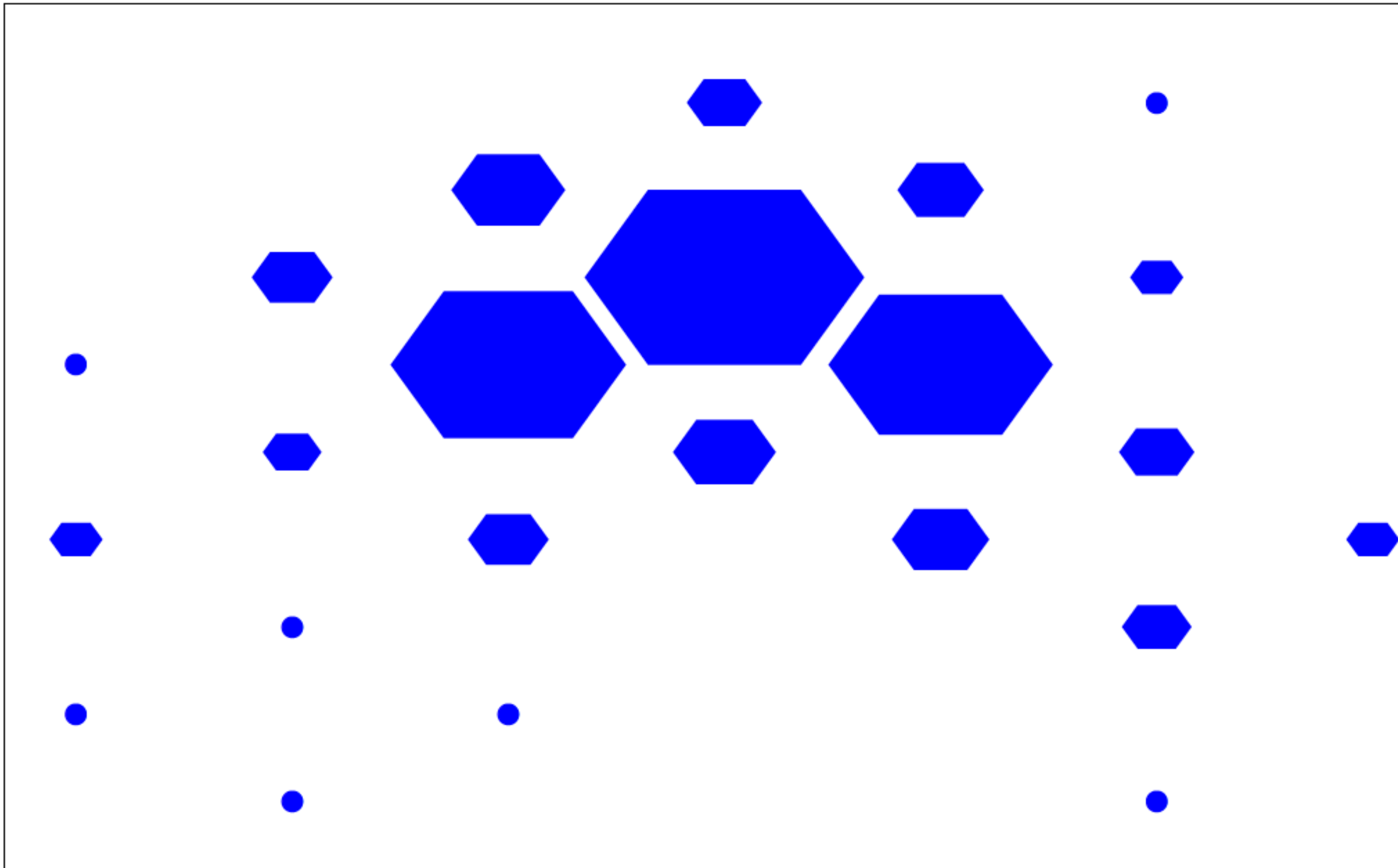
Hexagonal binning

Binning in two dimensions;



Hexagonal binning

Binning in two dimensions;



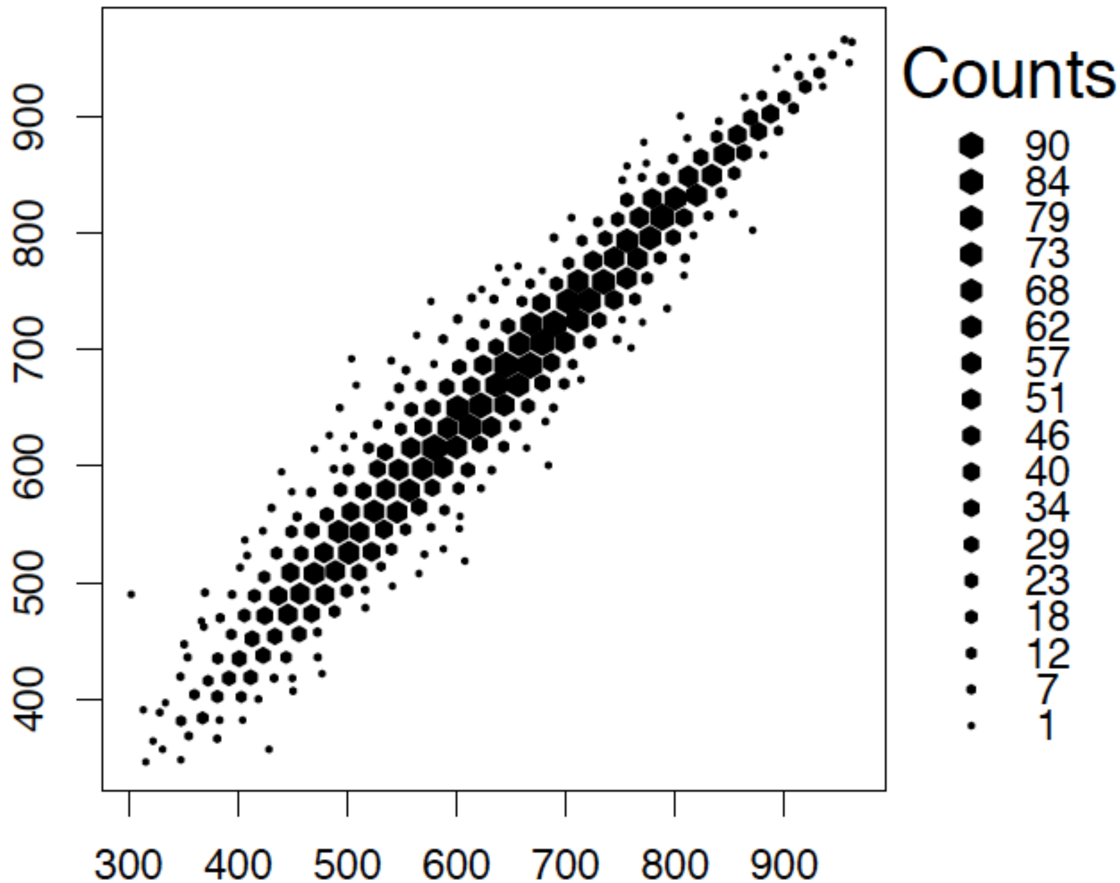
Hexagonal binning

The `hexbin()` package does all the bin construction, and counting. It has a `plot` method for its `hexbin` objects;

```
install.packages(c("hexbin","survey"))
library("hexbin")
library("survey")# for apipop data frame
data(api)
with(apipop, plot(hexbin(api99,api00), style="centroids"))
```

This is a dataset of Academic Performance indices for all schools in California. There are 6194 rows in the apipop dataframe. Api99 and api00 are the indices for 1999 and 2000.

Hexagonal binning



Hexagonal binning

Hexbin is used when you don't *really* care about the exact location of every single point

- Singleton points are plotted 'as usual'; you do (perhaps) care about them

- `hexbin` centers the 'ink' at the cell data's 'center of gravity'

- `style="centroids"` gives the center-of-gravity version; the default `style` is `colorscale` – usually grayscale. See `?gplot.hexagons` for more options

Hexagonal binning

For keen people: the **`hexbin`** package doesn't use the standard R graphics plotting devices; instead, it operates through the **`Grid`** system (in the **`grid`** package) which defines rectangular regions on a graphics device; these **`viewport`** regions can have a number of coordinate systems. To add lines to a hexbin plot, the options are;

- Use **`hexVP.abline()`** to add these directly
- Move everything into 'standard' graphics – not **`Grid`** graphics (see **`?Grid`**. This system lets you alter graphics *after* plotting them
- Write your own plot method for hexbin objects, with standard R graphics commands

But you will find things easier with ggplot

Description

Bin 2d plane into hexagons.

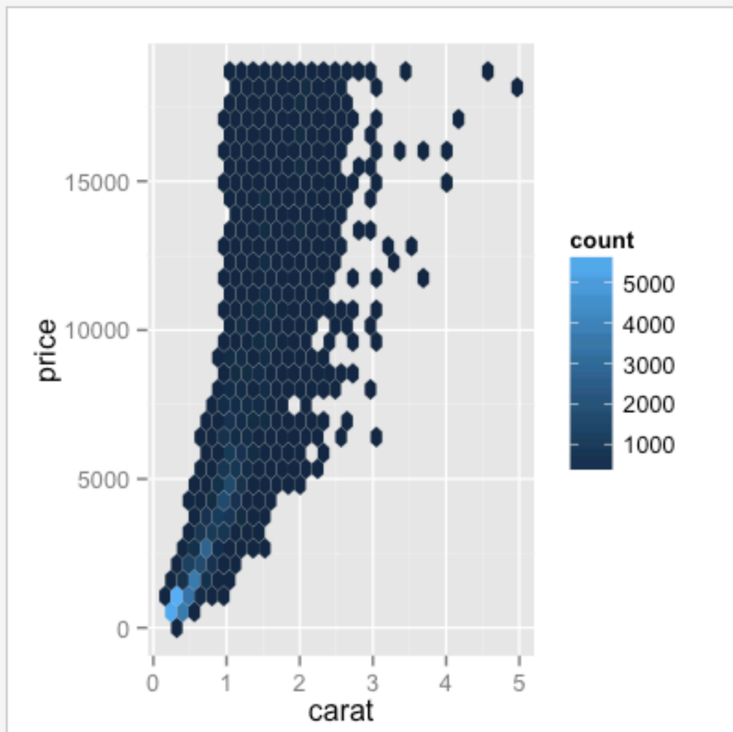
Aesthetics

`stat_binhex` understands the following aesthetics (required aesthetics are in bold):

- `x`
- `y`
- `fill`

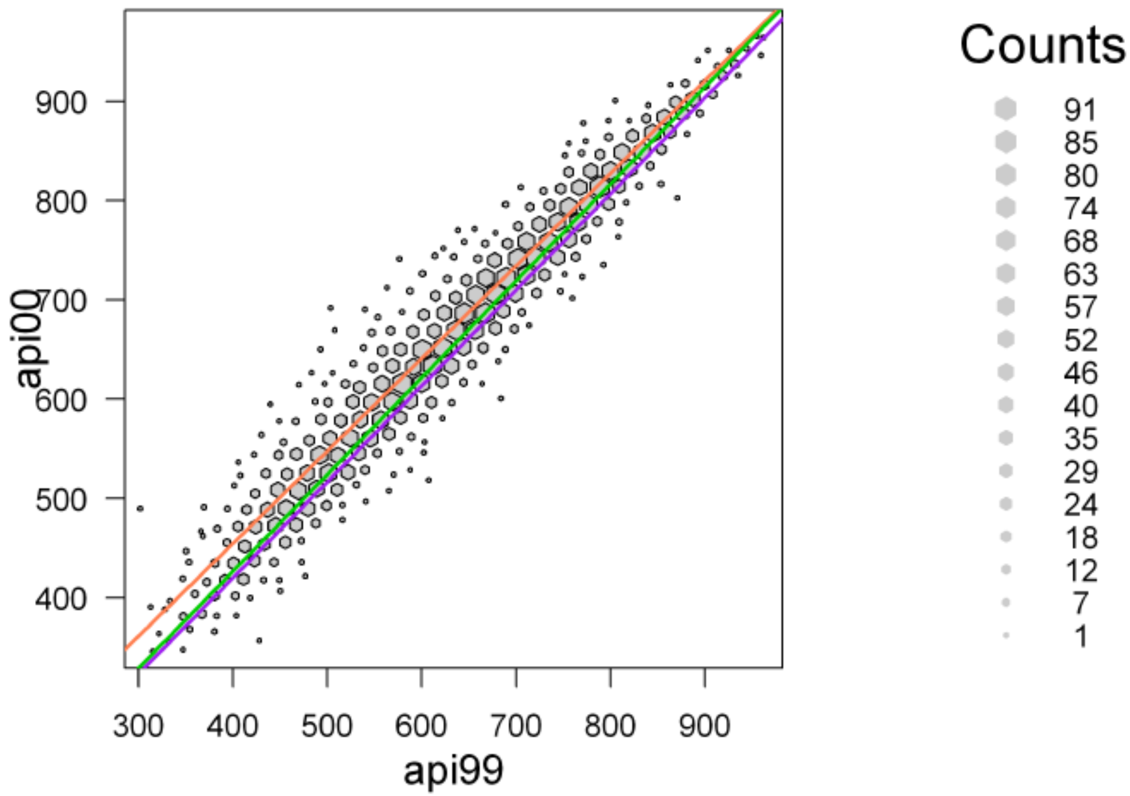
Examples

```
d <- ggplot(diamonds, aes(carat, price))
d + stat_binhex()
```



Hexagonal binning

An example; color-coded lines of best fit, by school type;




```
lm.e <- coef(lm(api00~api99, data=apipop, subset=stype=="E"))
lm.m <- coef(lm(api00~api99, data=apipop, subset=stype=="M"))
lm.h <- coef(lm(api00~api99, data=apipop, subset=stype=="H"))

hexVP.abline(vp1$plot.vp, lm.e[1], lm.e[2], col="coral")
```

Files and formats

When making a graphic file (a picture) a common problem is making the fonts too small; this happens when you use the default font sizes, on a letter-size graphic, which then gets printed much smaller.

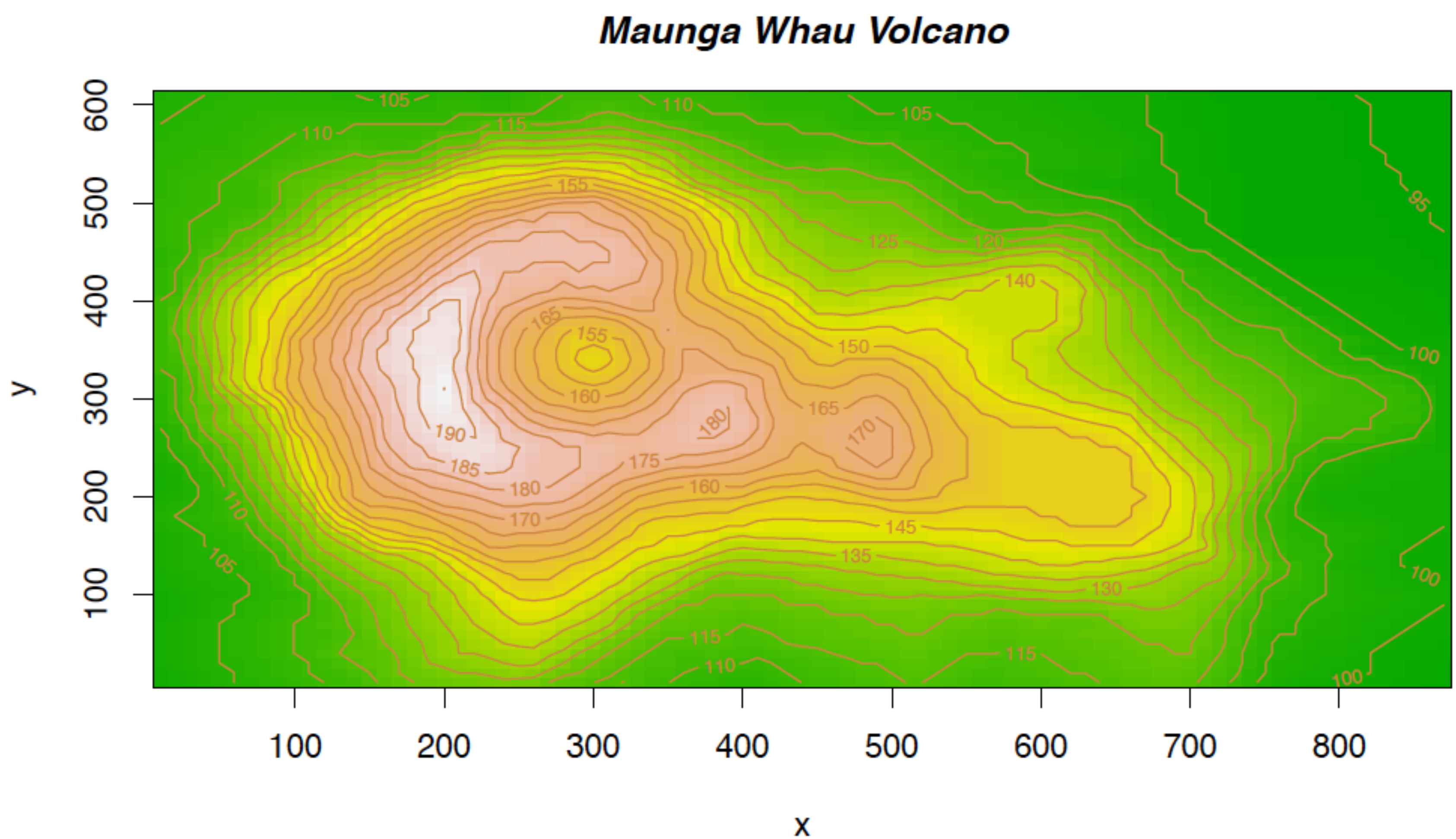
The solution is to let R pick the resolution, after you specify how big the picture will be ;

```
## on Windows
windows(height=4,width=6)
## on Unix    and Mac OS X  (but I usually use the "quartz" device driver)
x11(height=4,width=6)
```

This applies to other graphics formats too. Which to use depends on the software you'll use with the graphics file (e.g. PowerPoint) but file size also plays a role;

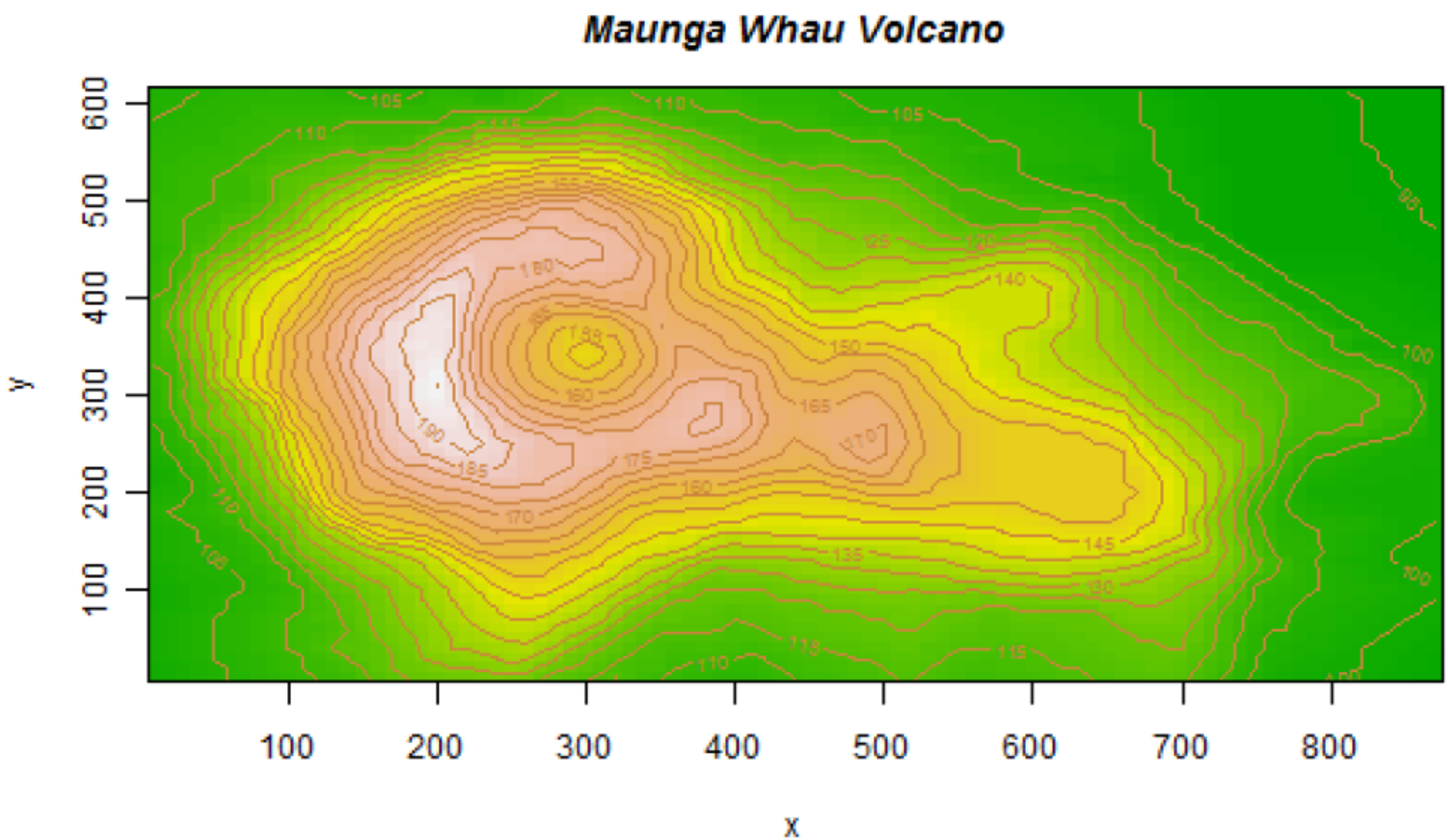
Files and formats

`pdf("myfile.pdf", w=7.5, h=4.5)` – 343K



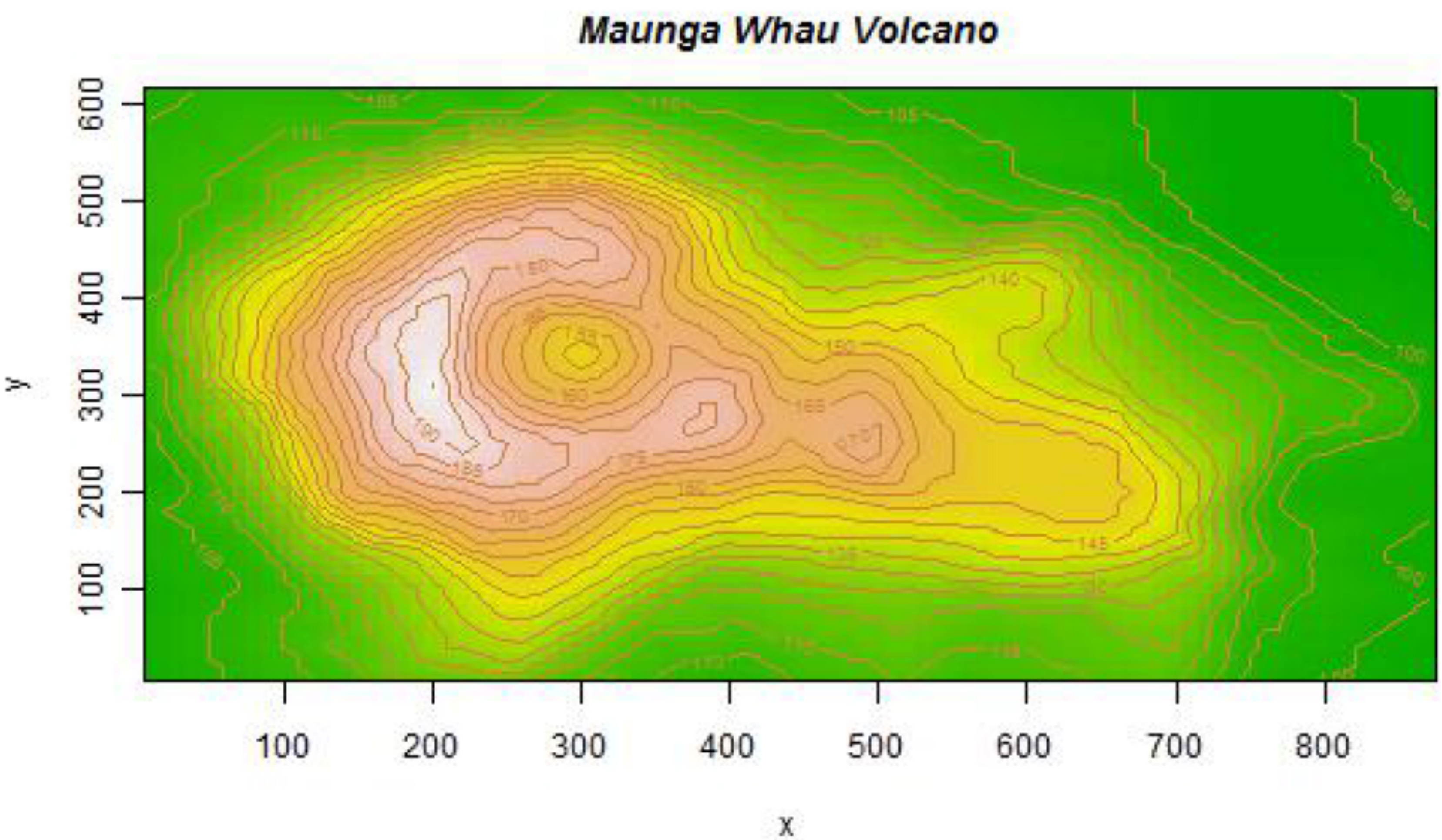
Files and formats

png("myfile.png", w=7.5*72, h=4.5*72, res=72) – 46K



Files and formats

```
jpeg("myfile.jpg", w=7.5*72, h=4.5*72, res=72)
```
– 31K



Files and formats

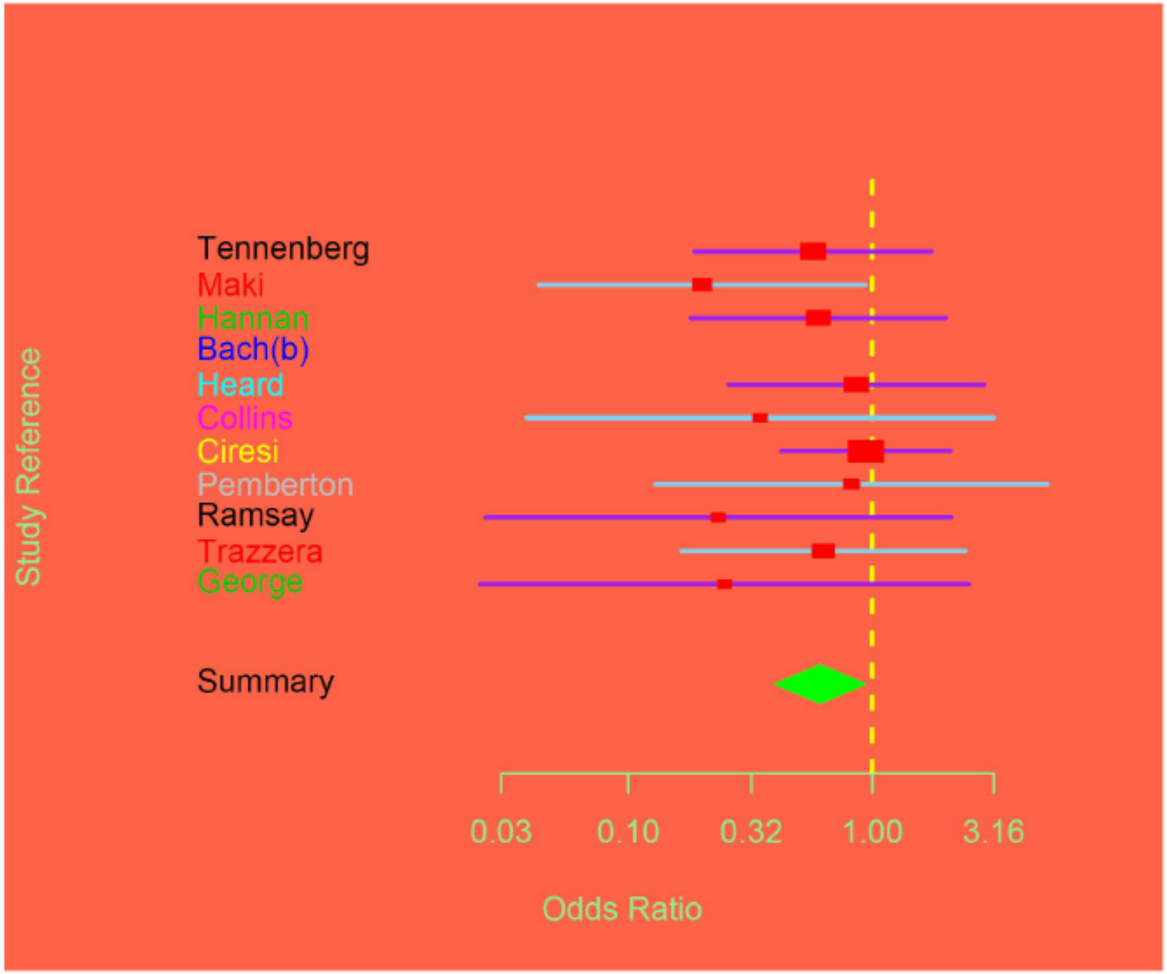
- PNG is under-appreciated; it's very good for 'line art'
- JPEG is optimized for photos
- PDFs are great but the files can get massive – and not all presentation software accepts them
- See also Windows metafiles (`.wmf`), on Windows, which PowerPoint can edit directly

Keen people: more sophisticated devices like SVG are good for interactive graphics.

Everyone: if you see illegible material, ask what it says... and ask *pointedly*.

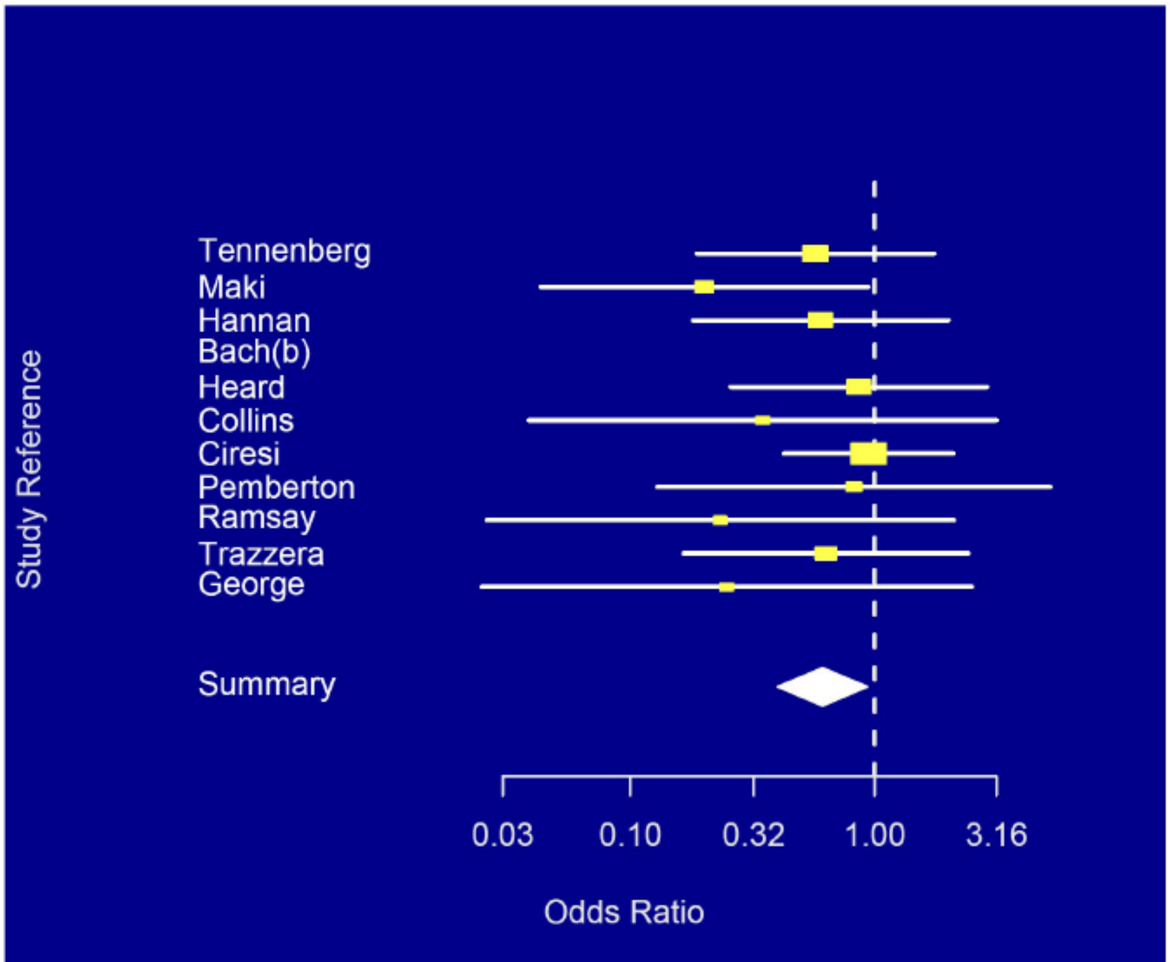
Color schemes

The choice is not just ‘does it look cool’?



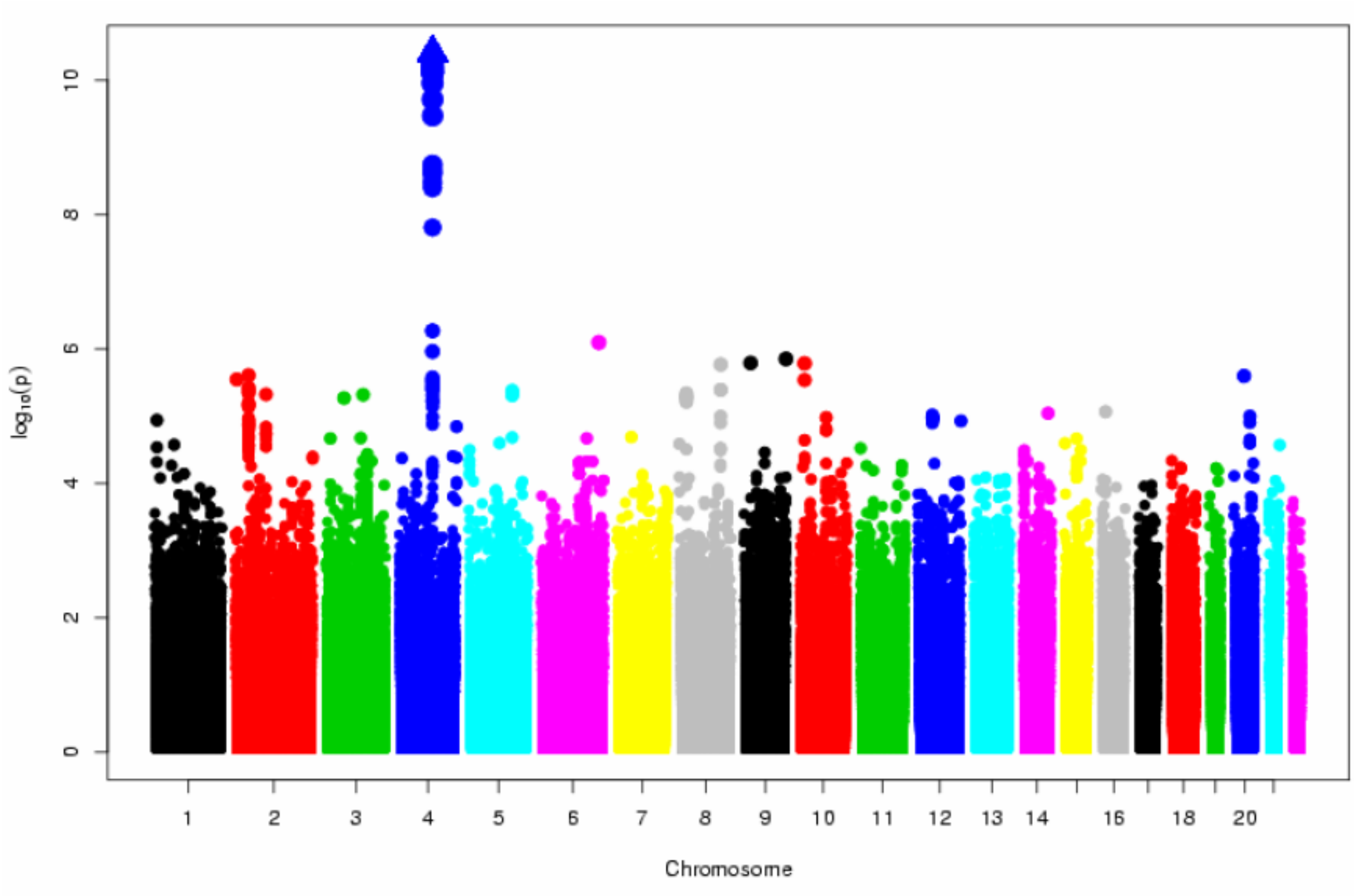
Color schemes

The choice is not just ‘does it look cool’?



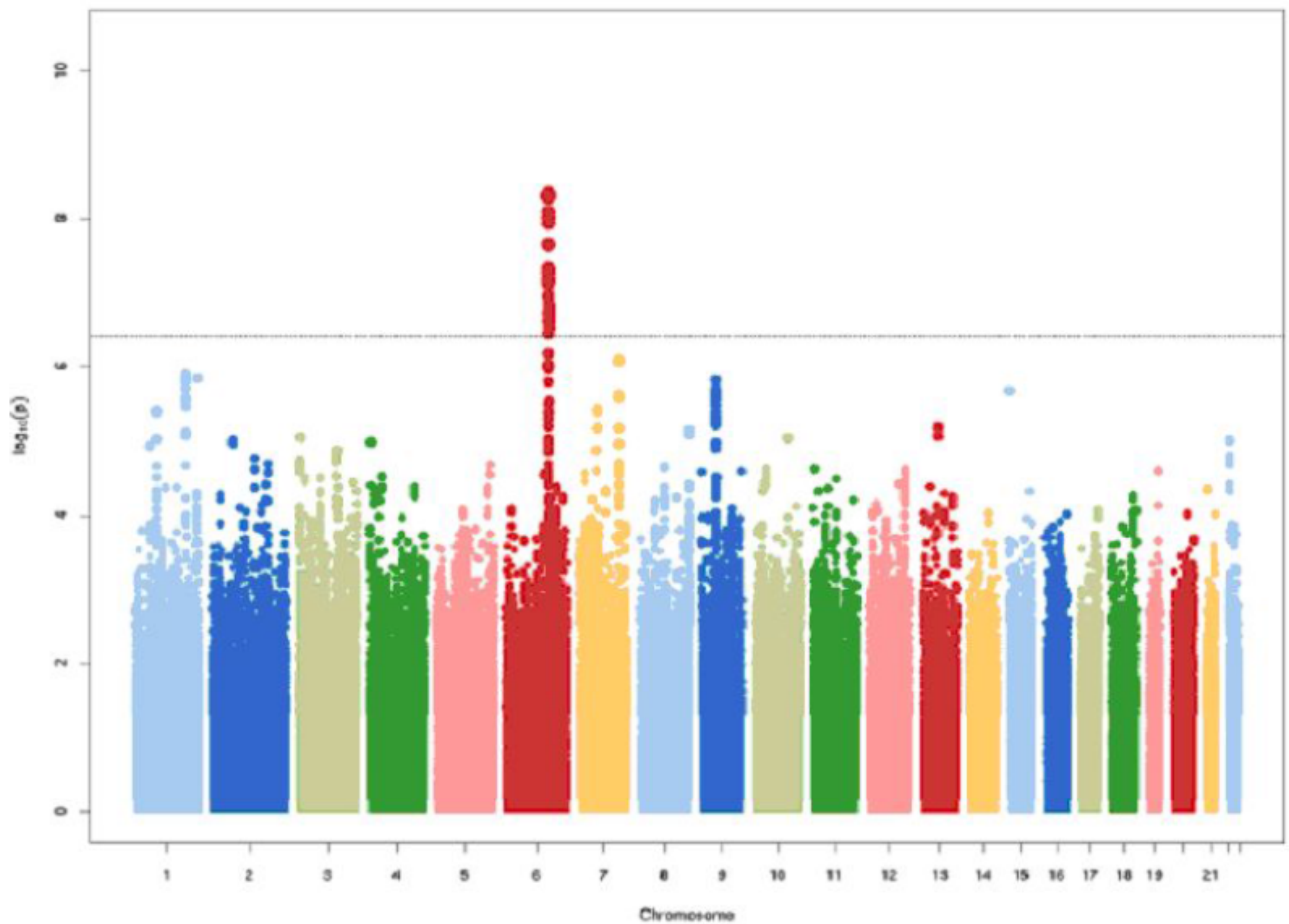
Color schemes

Which blobs of color stand out?



Color schemes

Which blobs of color stand out?



Color schemes

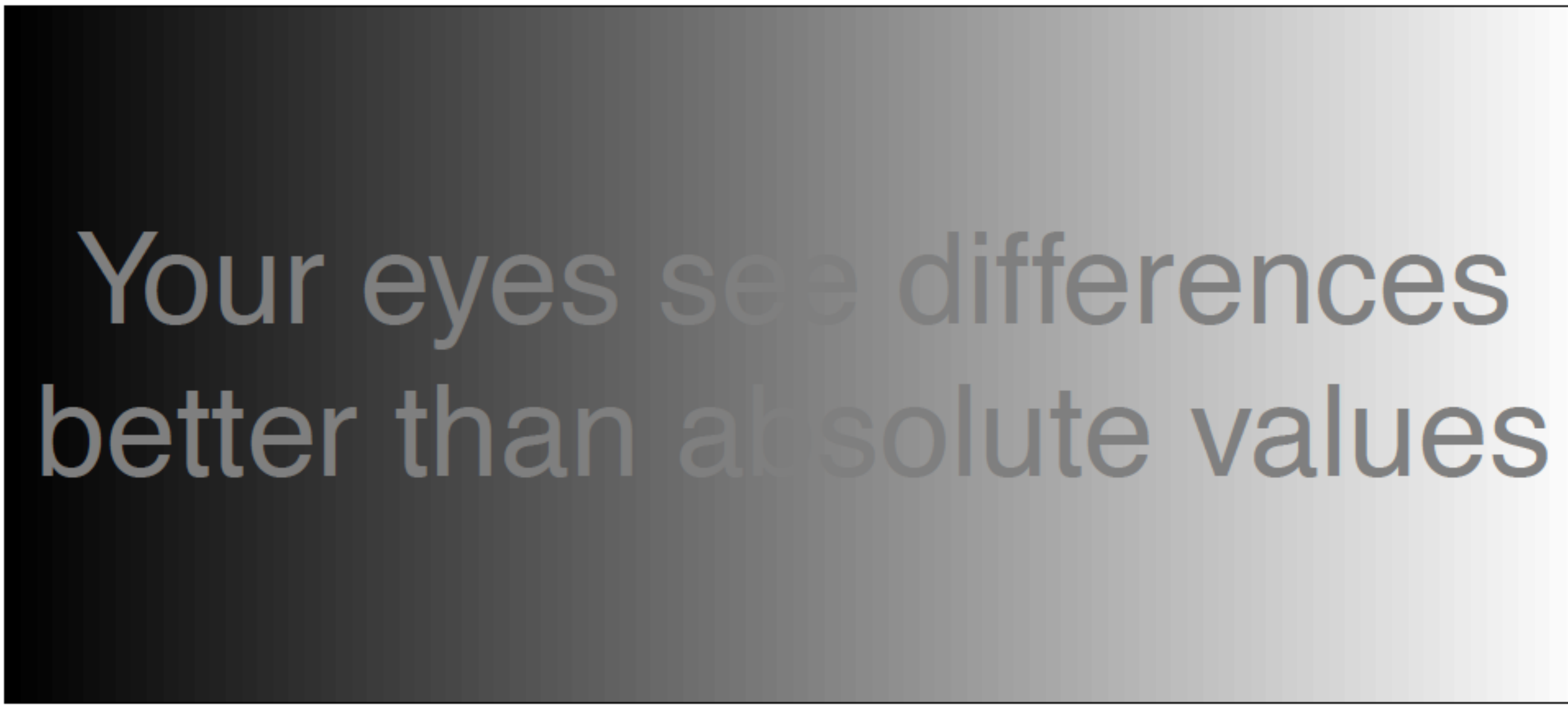
Why? Because...

Your eyes see differences better than absolute values

... and this applies in any color

Color schemes

Why? Because...



... and light/dark is more obvious than e.g. red/blue

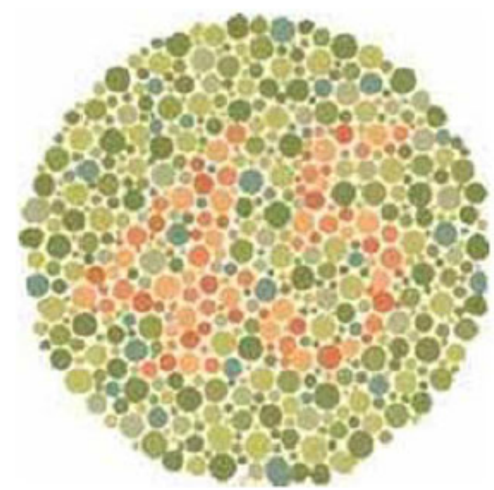
Color schemes

Color choice is best left to experts, or people with taste.

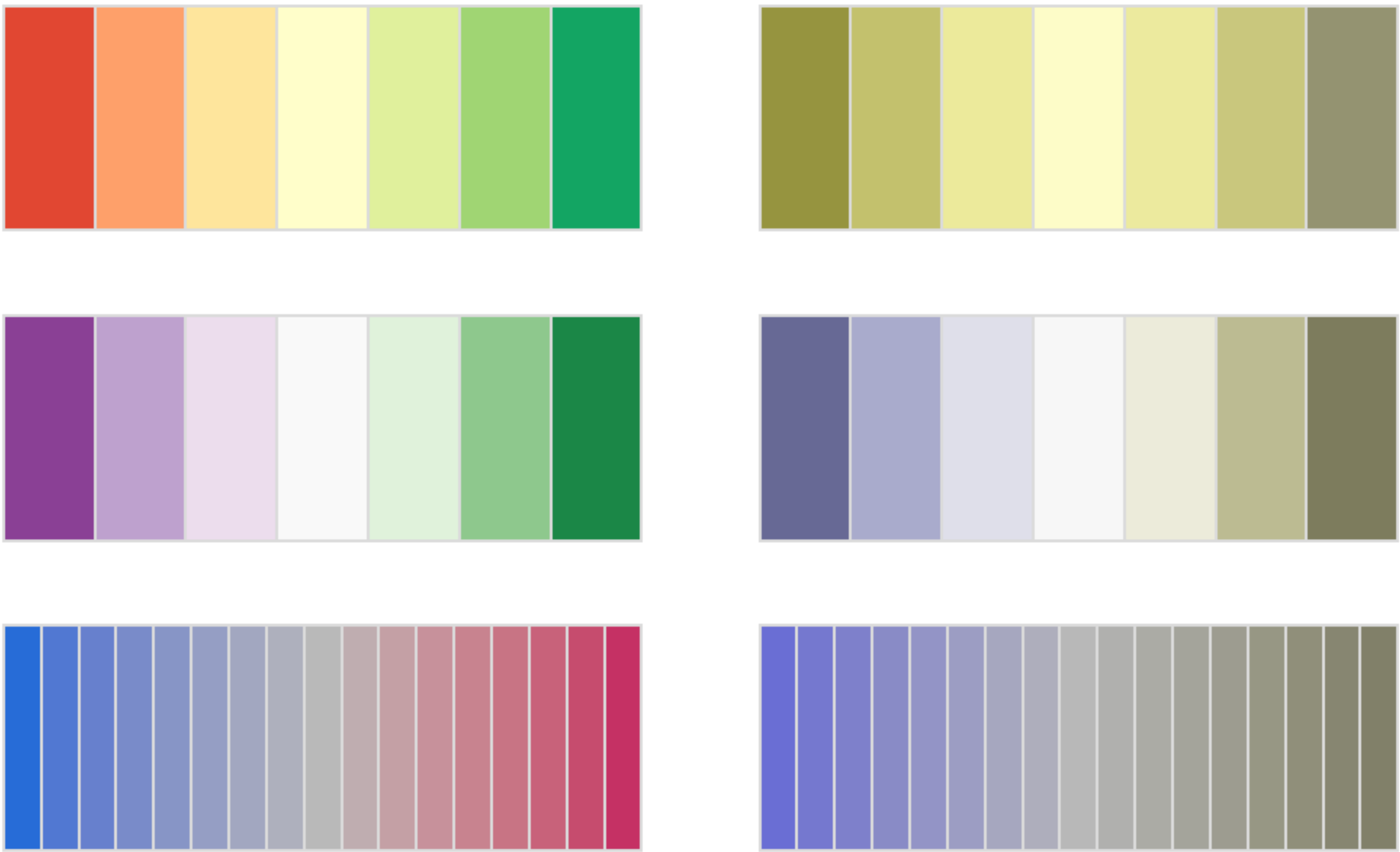
`http://www.colorbrewer.org` has color schemes designed for the National Cancer Atlas, also in package `RColorBrewer`

`colorspace` package has color schemes based on straight lines in a perceptually-based color space (rather than RGB).

`dichromat` package attempts to show the impact of red:green color blindness on your R color schemes.



Color blindness



Color blindness

Color blindness is more common in men (5–10% of adults)

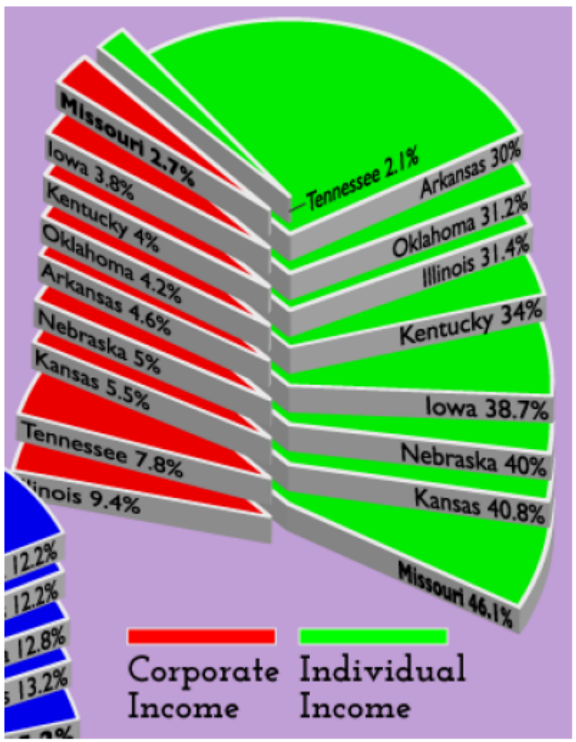
Color blindness

Color blindness is more common in men (5–10% of adults)

This is from a "Summer Computing & Research" course taught by Ken Rice in 2012.
I don't know whether you have access through this link.

Resources

https://catalyst.uw.edu/workspace/kenrice/23131/



- Site has the papers mentioned, plus more
- Graphics are not just for posters – expect to be making and using graphs in 572, and beyond
- Look around! Use other people's good ideas

And I collect horrible graphs – all donations gratefully received