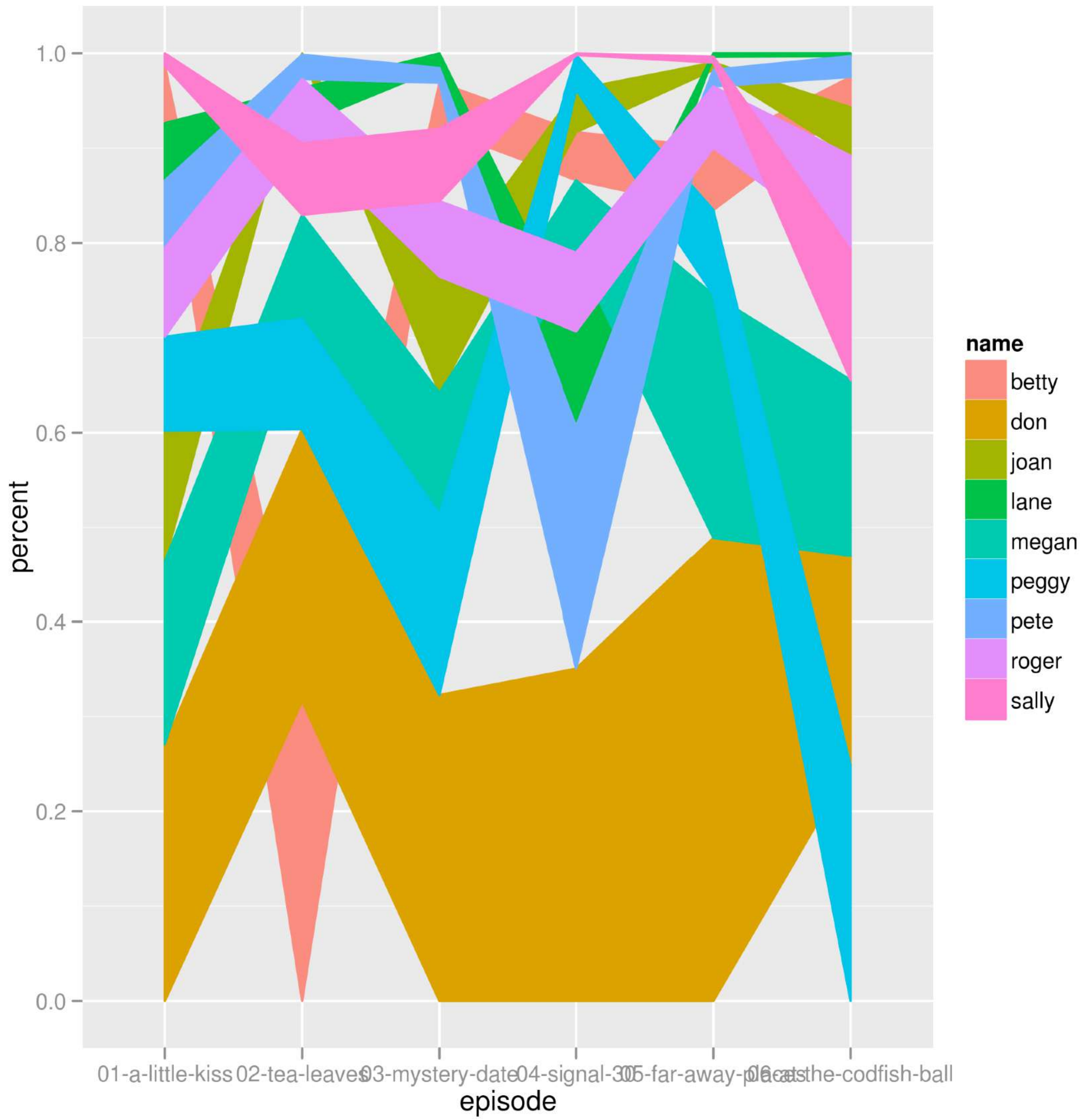
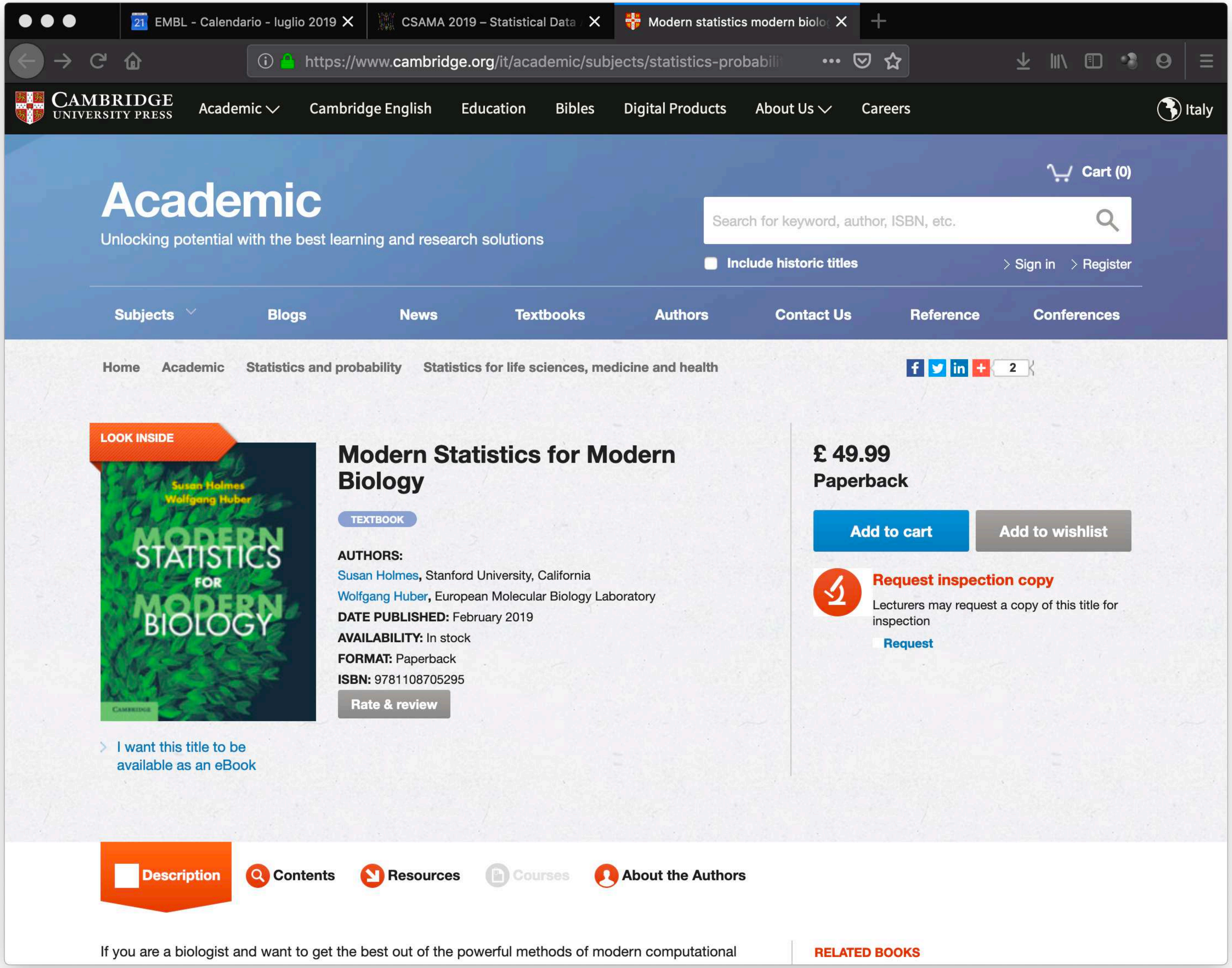


Graphics and Visualization

Wolfgang Huber

See also

www.huber.embl.de/msmb Chapter 3



Academic

Unlocking potential with the best learning and research solutions

Modern Statistics for Modern Biology

AUTHORS:
Susan Holmes, Stanford University, California
Wolfgang Huber, European Molecular Biology Laboratory
DATE PUBLISHED: February 2019
AVAILABILITY: In stock
FORMAT: Paperback
ISBN: 9781108705295

I want this title to be available as an eBook

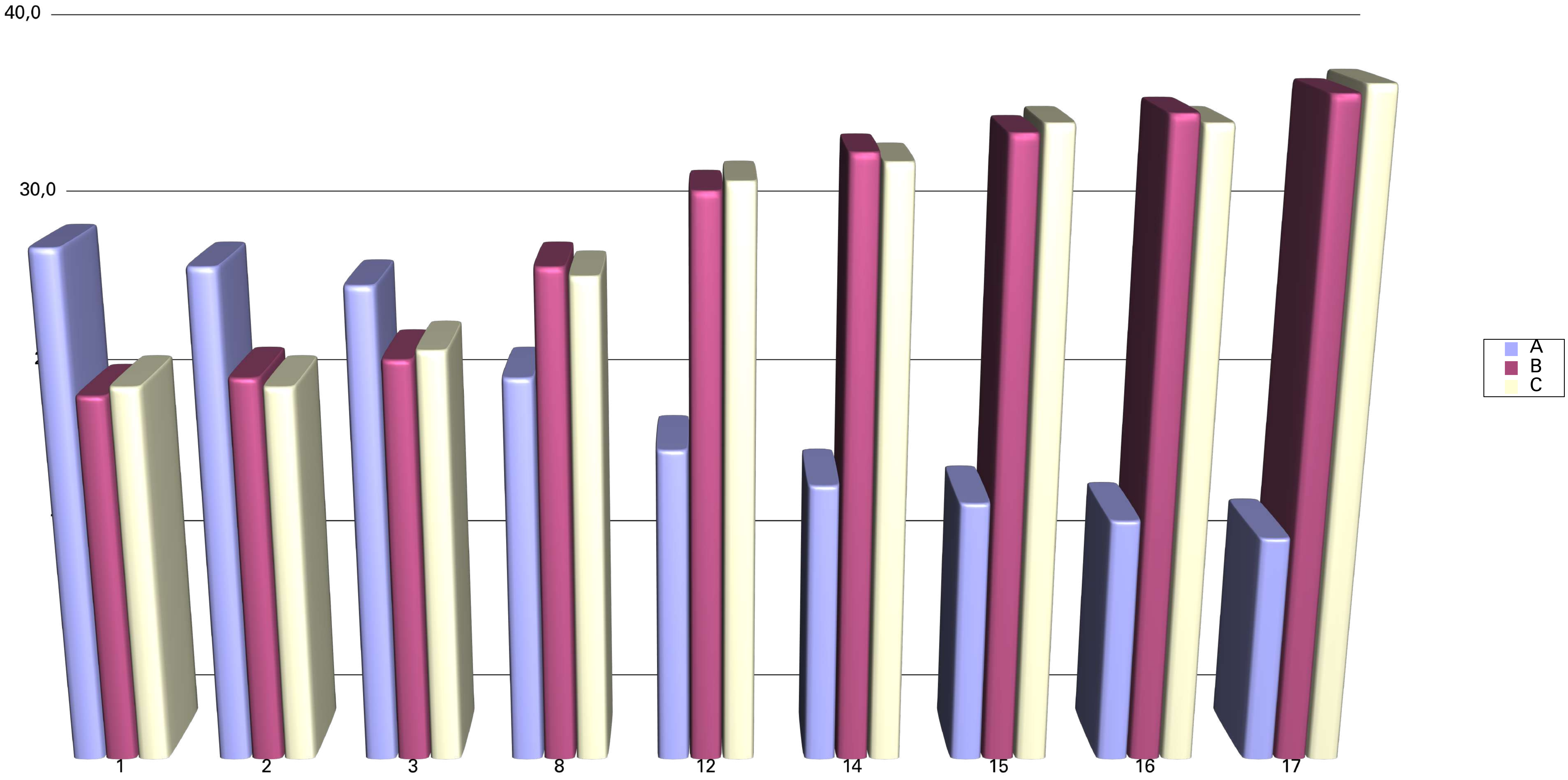
£ 49.99
Paperback

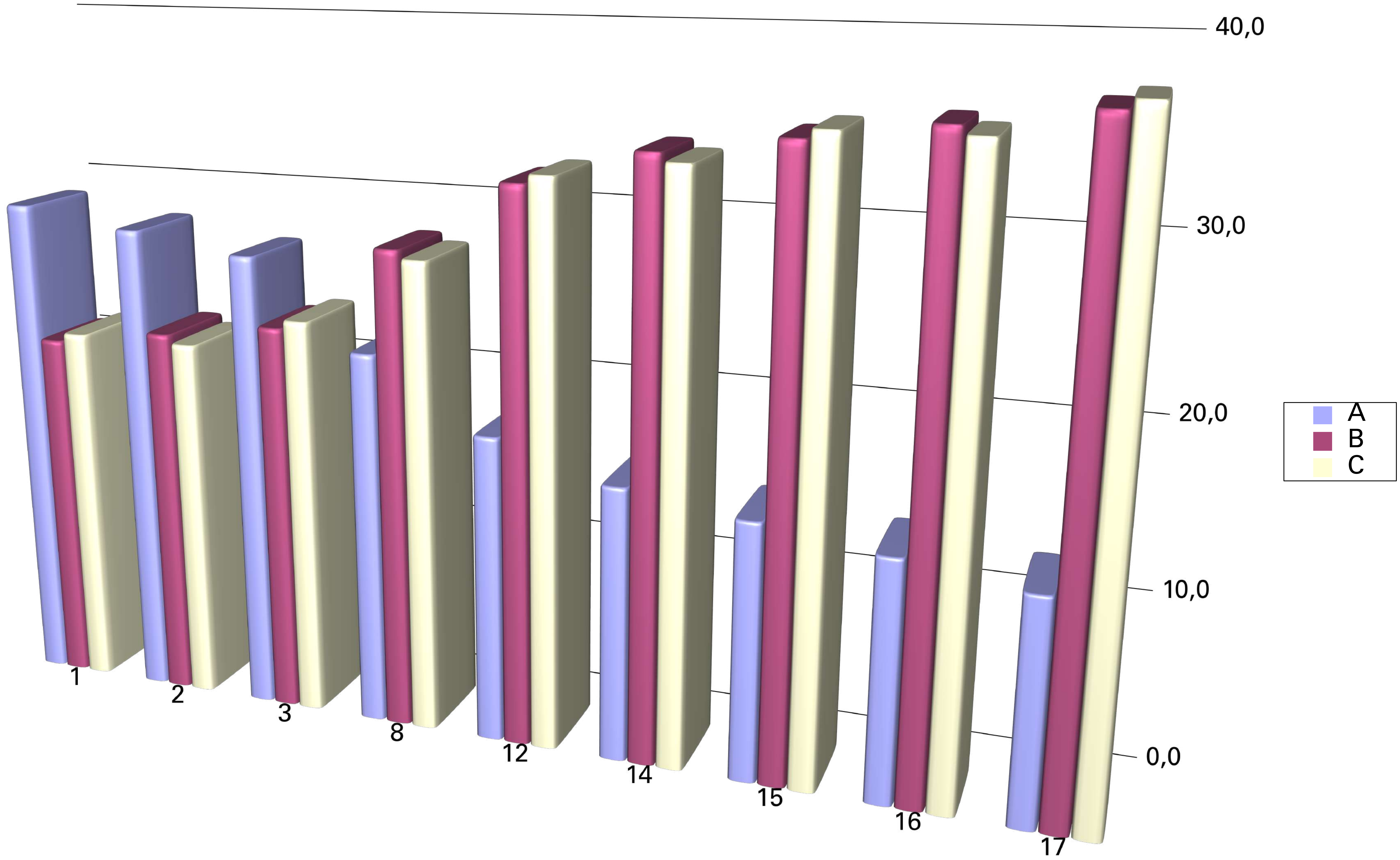
Request inspection copy
Lecturers may request a copy of this title for inspection

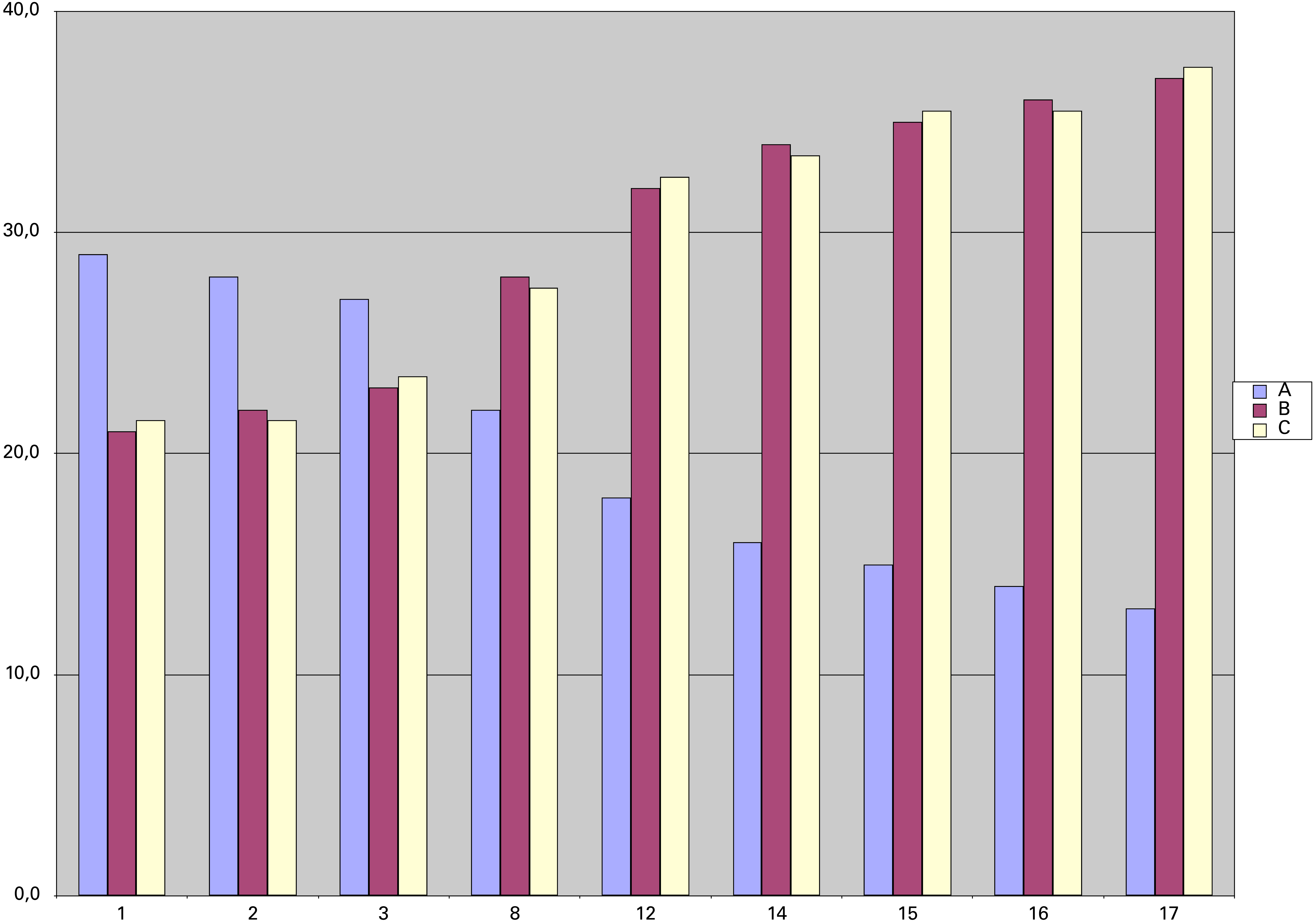
If you are a biologist and want to get the best out of the powerful methods of modern computational

RELATED BOOKS

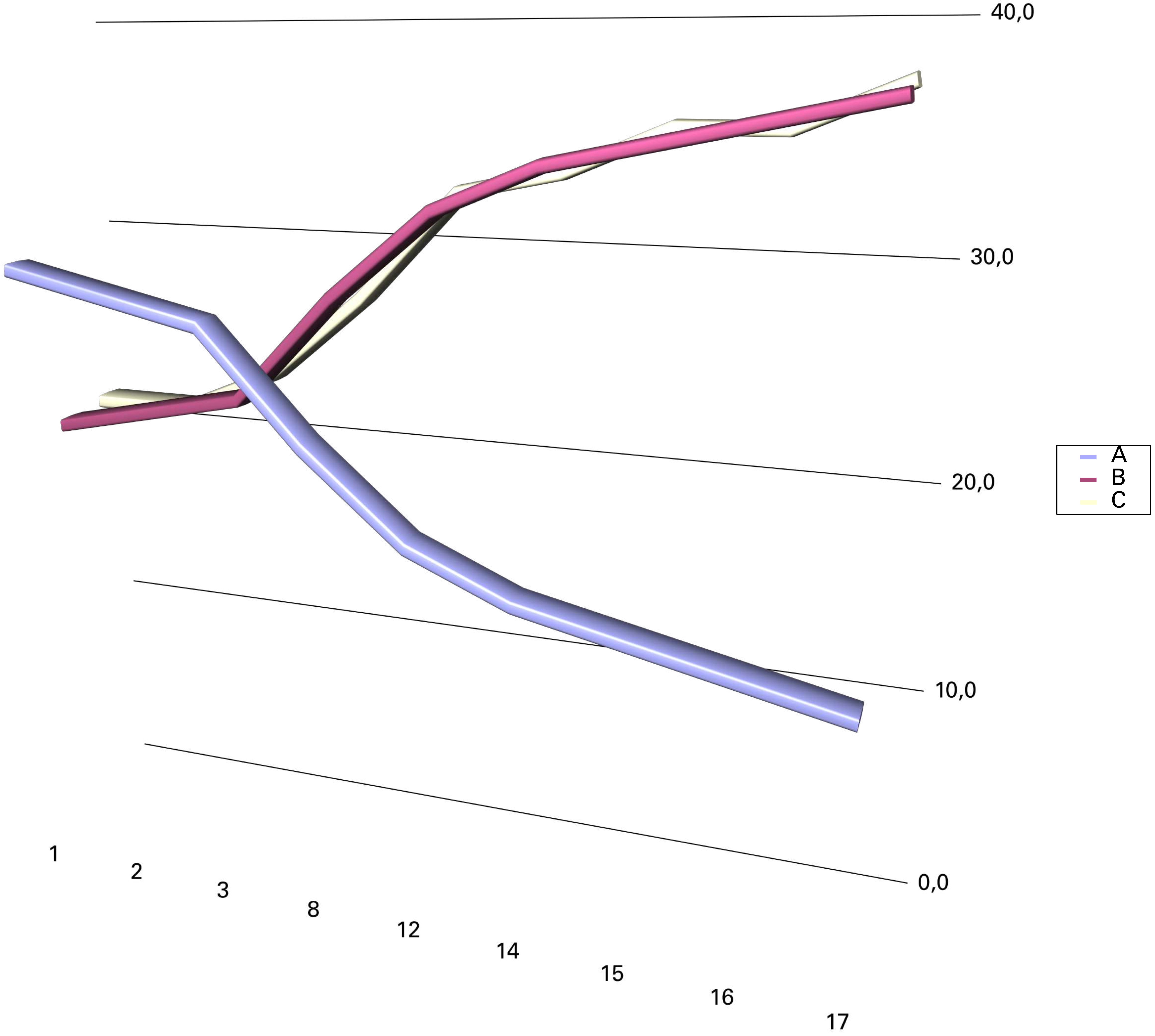
Horror Picture Show

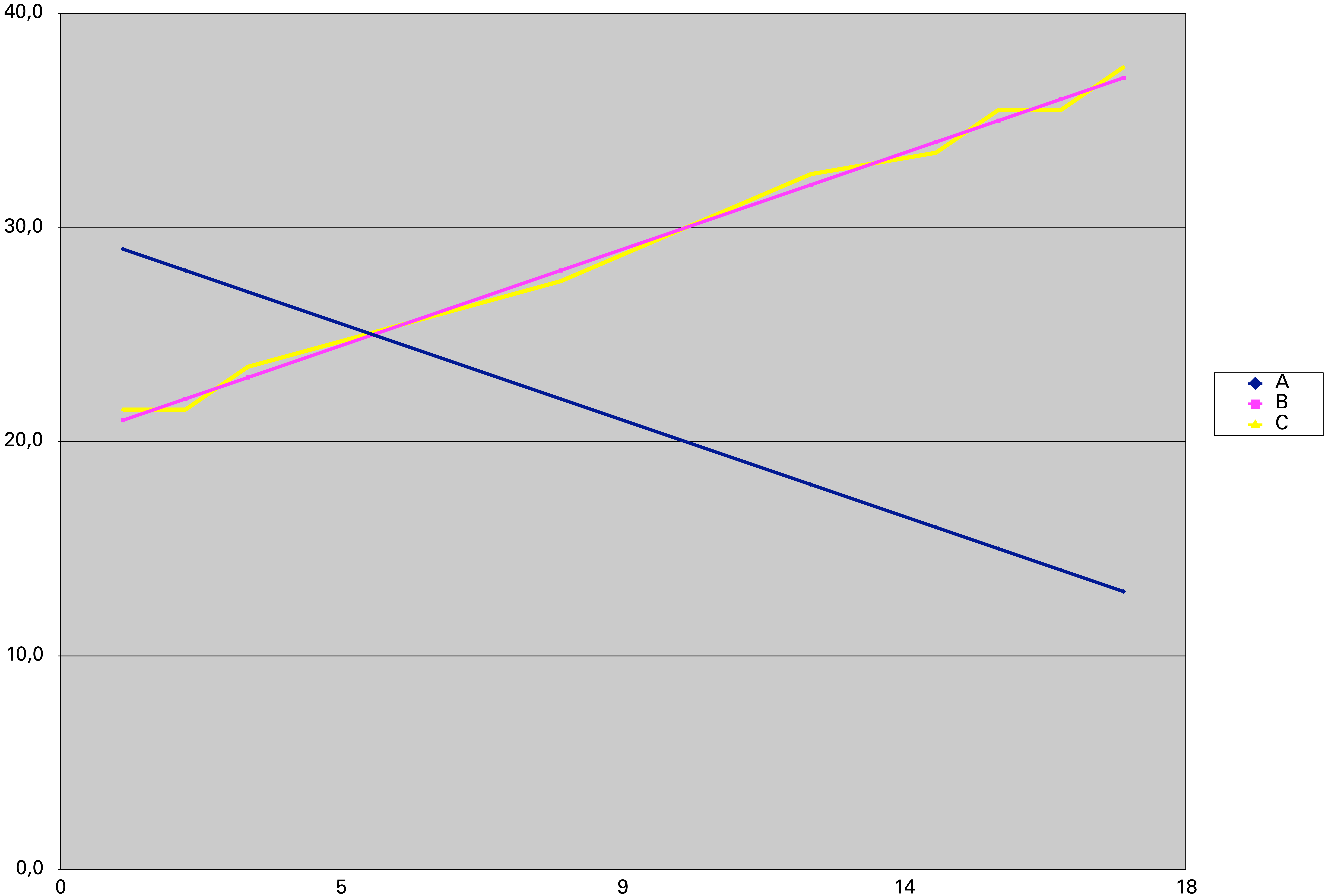

40,0
30,0
2
1
2
3
8
12
14
15
16
17
A
B
C


40,0
30,0
20,0
10,0
0,0
1
2
3
8
12
14
15
16
17
A
B
C

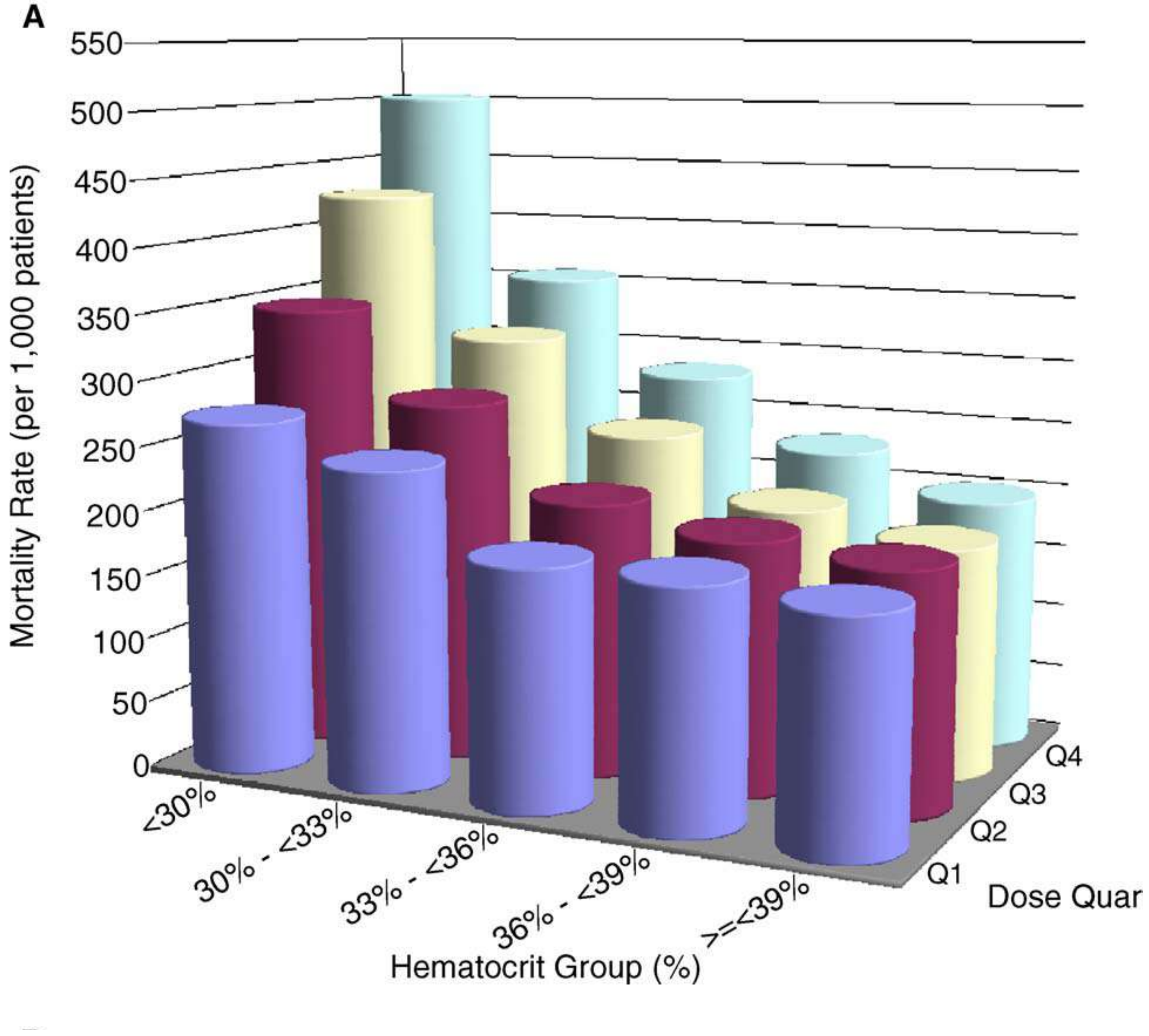


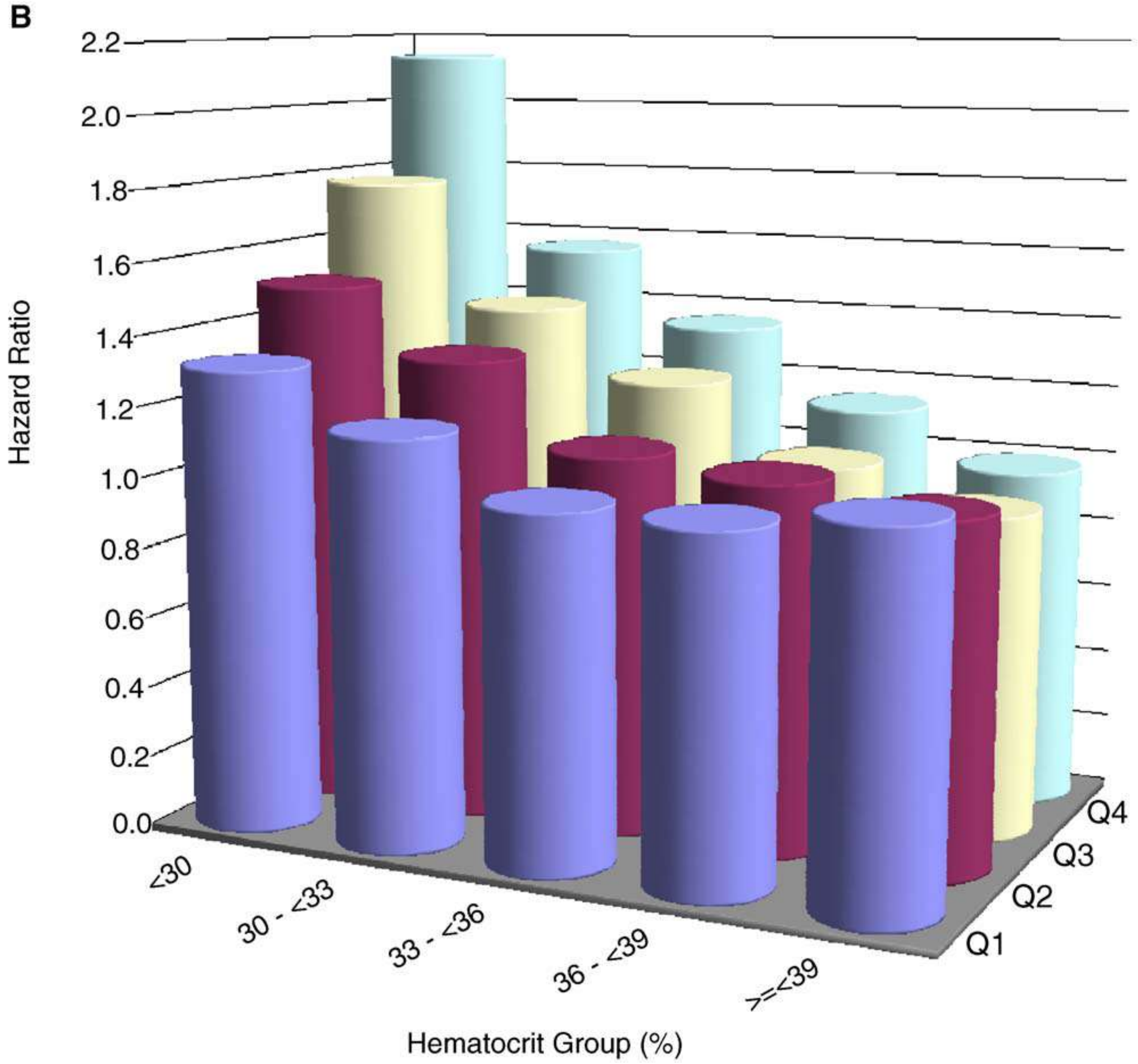
40,0
30,0
20,0
10,0
0,0
1
2
3
8
12
14
15
16
17
A
B
C

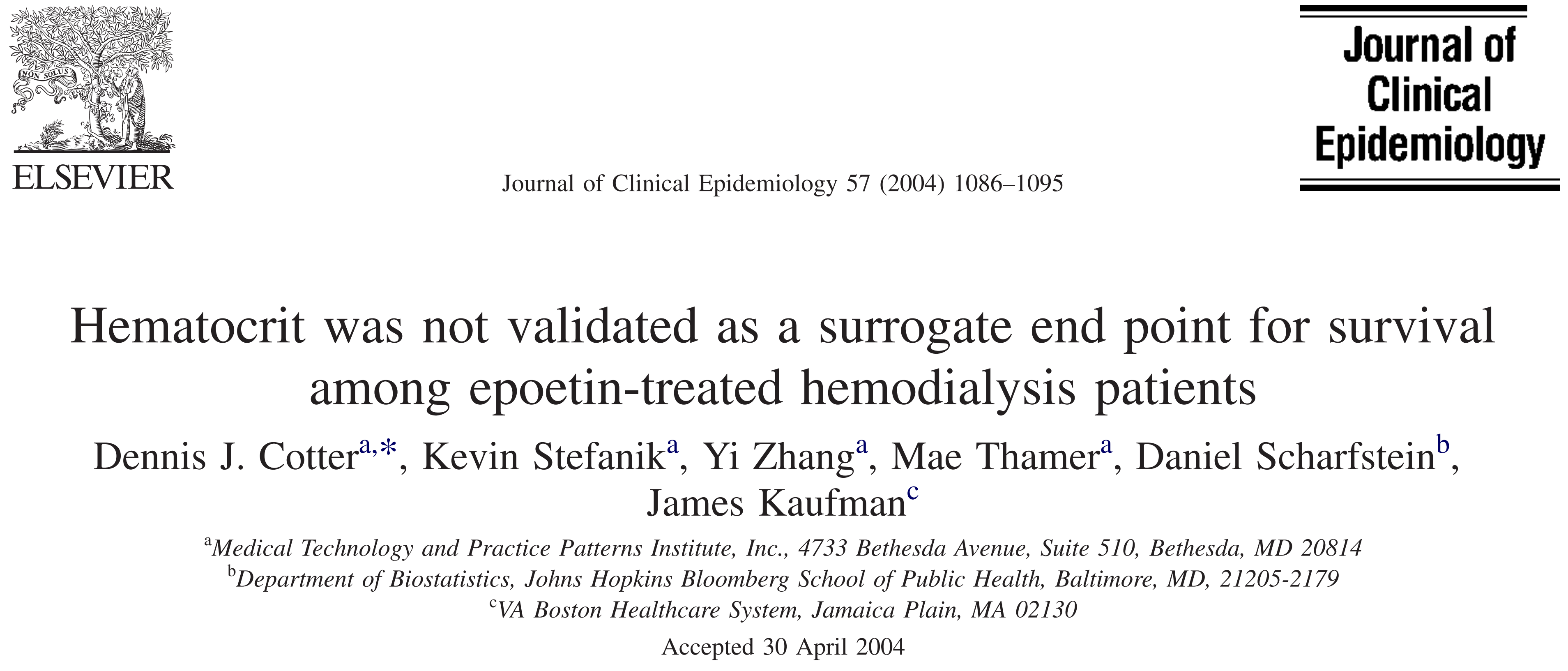

40,0
30,0
20,0
10,0
0,0
1
2
3
8
12
14
15
16
17
A
B
C



40,0
30,0
20,0
10,0
0,0
0
5
9
14
18
A
B
C






ELSEVIER

Journal of Clinical Epidemiology 57 (2004) 1086–1095

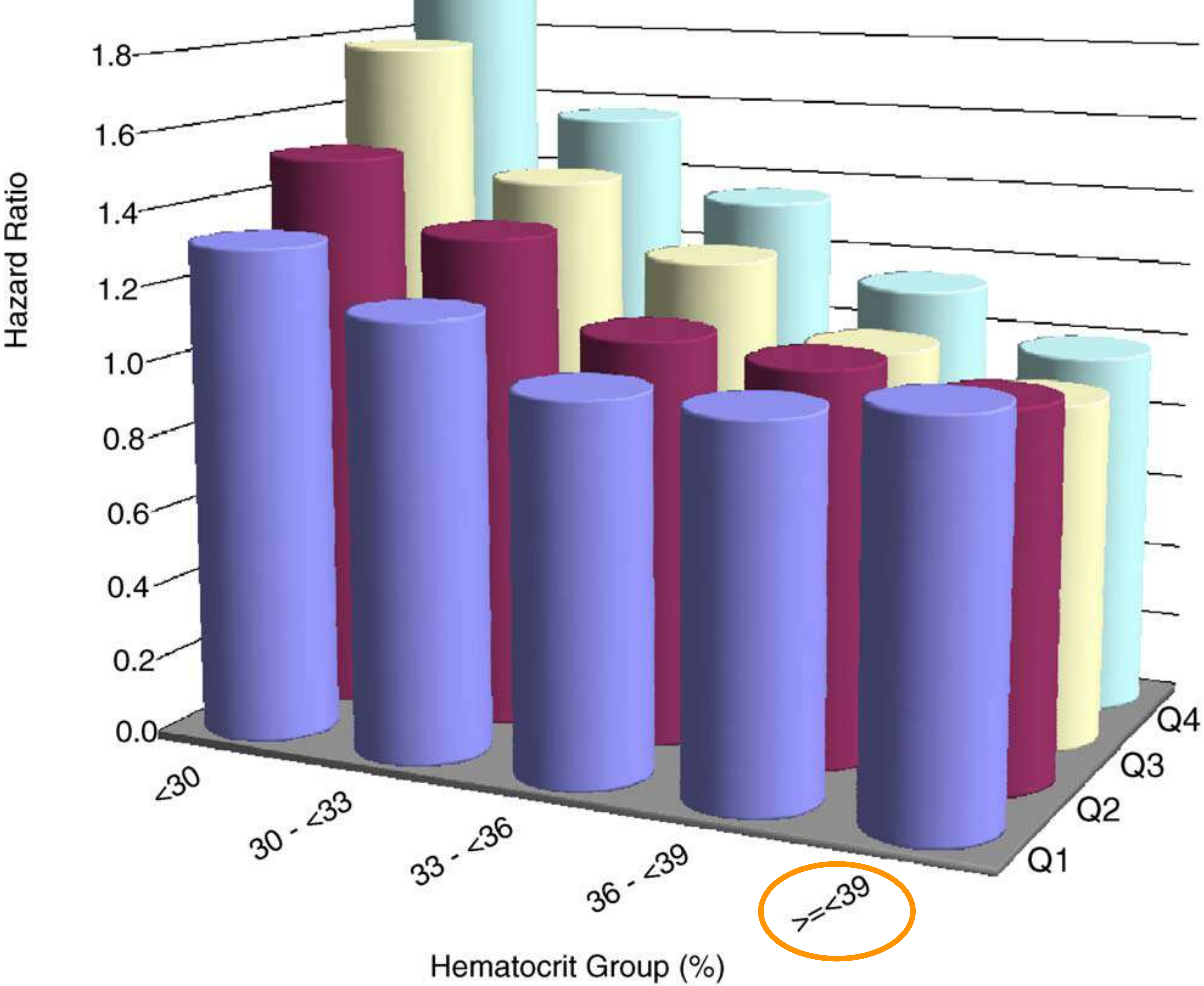
Journal of Clinical Epidemiology

Hematocrit was not validated as a surrogate end point for survival among epoetin-treated hemodialysis patients

Dennis J. Cotter[a,*], Kevin Stefanik[a], Yi Zhang[a], Mae Thamer[a], Daniel Scharfstein[b], James Kaufman[c]

[a]*Medical Technology and Practice Patterns Institute, Inc., 4733 Bethesda Avenue, Suite 510, Bethesda, MD 20814*
[b]*Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD, 21205-2179*
[c]*VA Boston Healthcare System, Jamaica Plain, MA 02130*

Accepted 30 April 2004


Hazard Ratio
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
<30
30 - <33
33 - <36
36 - <39
>=<39
Q4
Q3
Q2
Q1
Hematocrit Group (%)

Why graphics?

1. To explore data (interactively)
2. To communicate data & preliminary insights with collaborators
3. To publish results

Goals for this lecture

- Review base R plotting
- Understand the **grammar of graphics** concept
- Introduce ggplot2's ggplot function
- See how to plot 1D, 2D, 3-5D data and understand faceting
- Visualisation for quickling viewing large datasets and discover large-scale trends (e.g. batch effects)
- Use colours like a pro

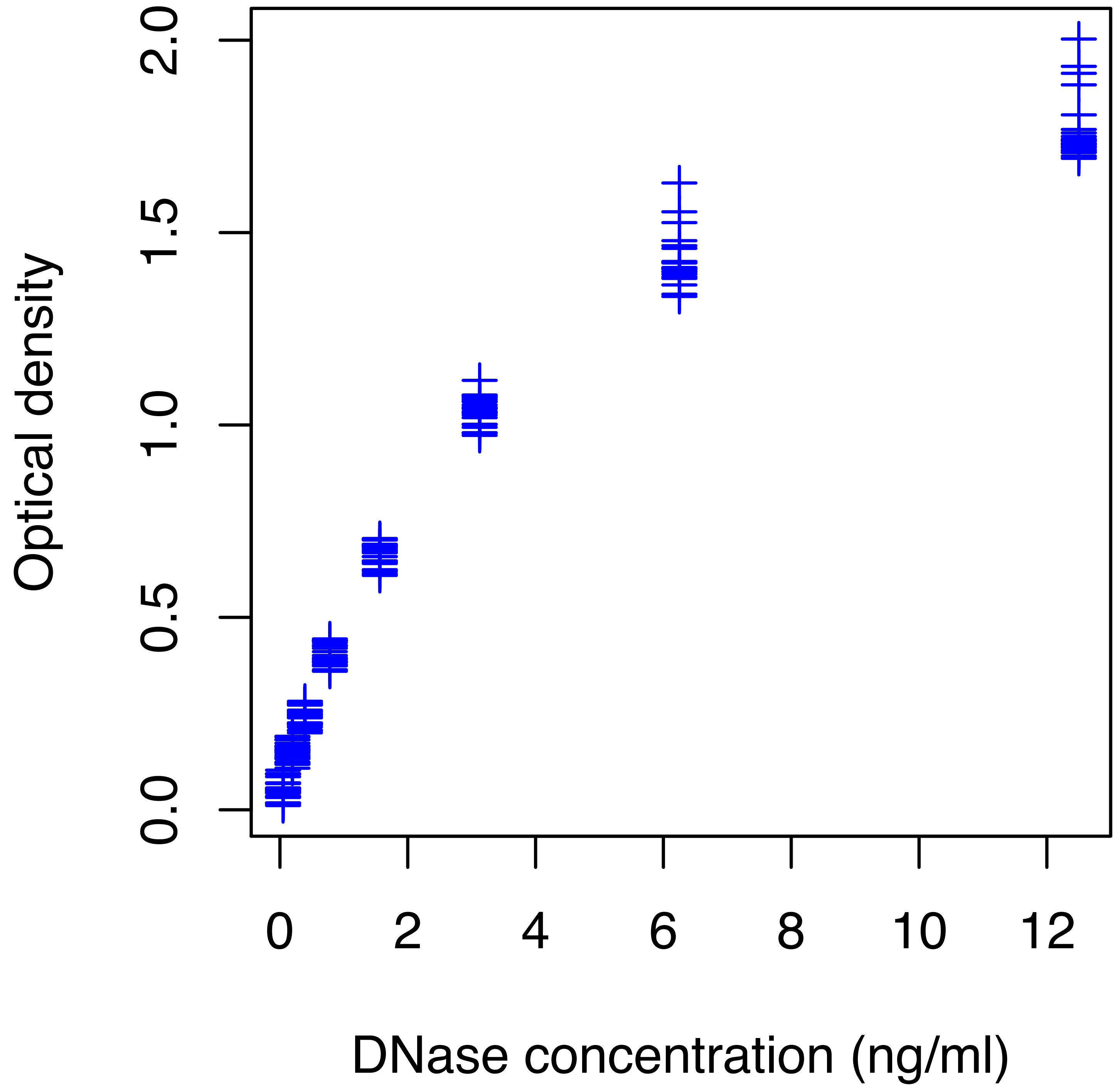
Base R plotting

Canvas model: a series of instructions that sequentially fill the plotting canvas

```
head(DNase)
```

```
##   Run   conc density
## 1   1 0.0488   0.017
## 2   1 0.0488   0.018
## 3   1 0.1953   0.121
## 4   1 0.1953   0.124
## 5   1 0.3906   0.206
## 6   1 0.3906   0.215
```

```
plot(DNase$conc, DNase$density,
ylab = attr(DNase, "labels")$y,
xlab = paste(attr(DNase, "labels")$x, attr(DNase, "units")$x),
pch = 3, col = "blue")
```



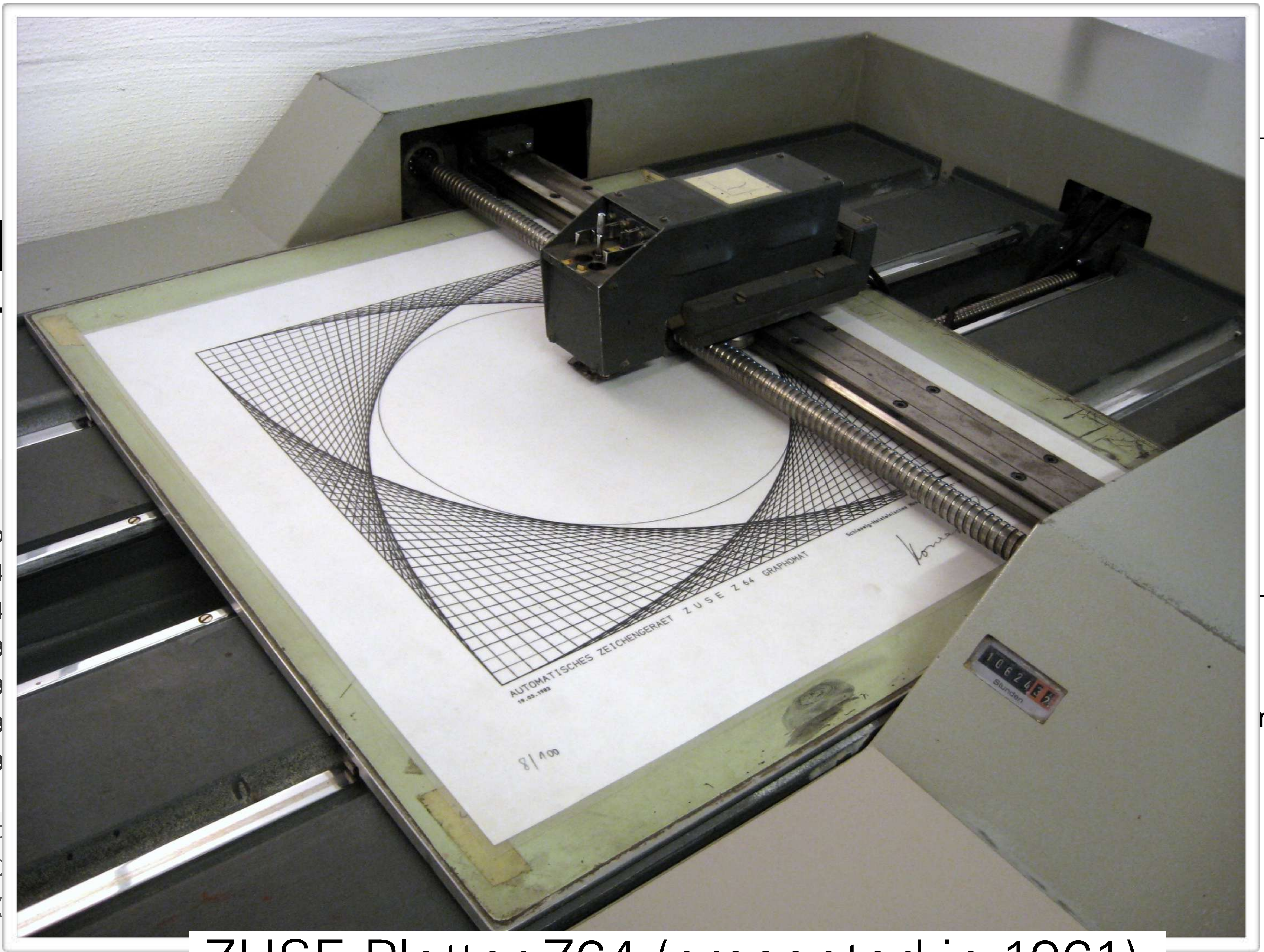
Base R plotting

Canv
of i
seq
plo

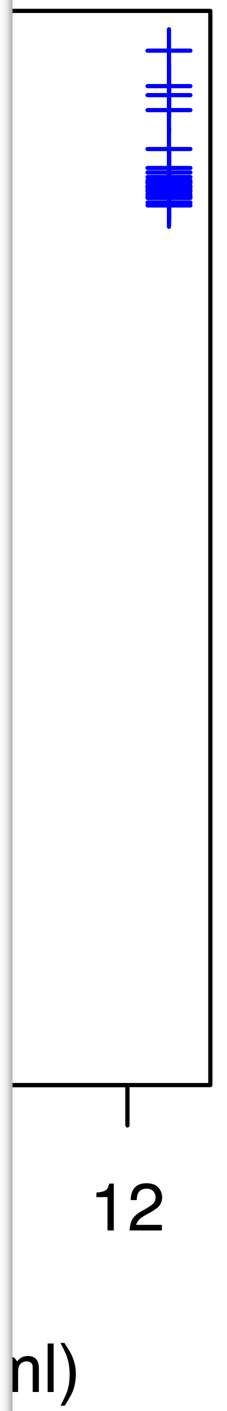
```
head(DNase)
```

```
##    Run   co
## 1    1 0.04
## 2    1 0.04
## 3    1 0.19
## 4    1 0.19
## 5    1 0.39
## 6    1 0.39
```

```
plot(DNase$co
ylab = attr(D
xlab = paste(
pch = 3, col
```

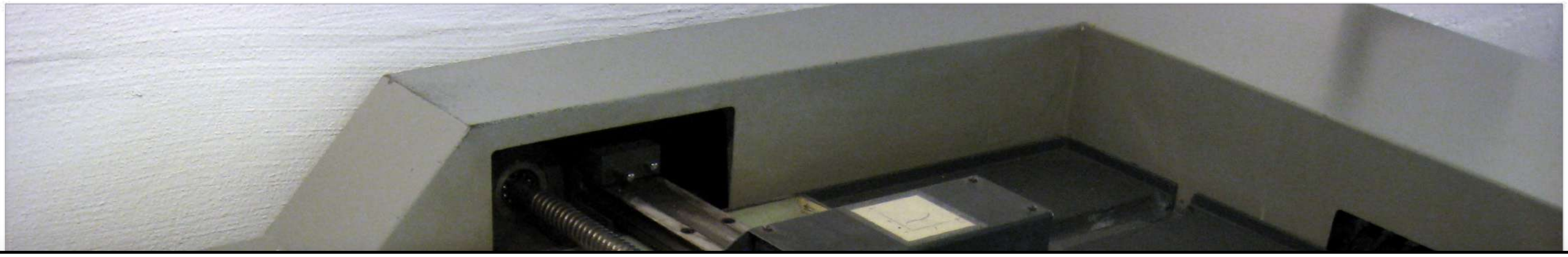


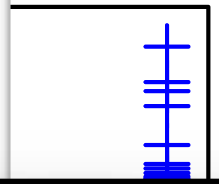
ZUSE Plotter Z64 (presented in 1961).



Base R plotting

Canv
of i
se
pl





```
head(DNase

##    Run
## 1    1 0
## 2    1 0
## 3    1 0
## 4    1 0
## 5    1 0
## 6    1 0
```

```
plot(DNase
ylab = att
xlab = pas
pch = 3, c
```

Drawbacks:

- Layout choices have to be made at the beginning with no overview over what may still be coming
- Different functions for different plot types, with different interfaces
- Many routine tasks require a lot of ‘boilerplate’ code
- No concept of facets / lattices
- No concept of viewports, only a single global coordinate system
- Default colours are poor
- Resizing often leads to unsatisfactory results

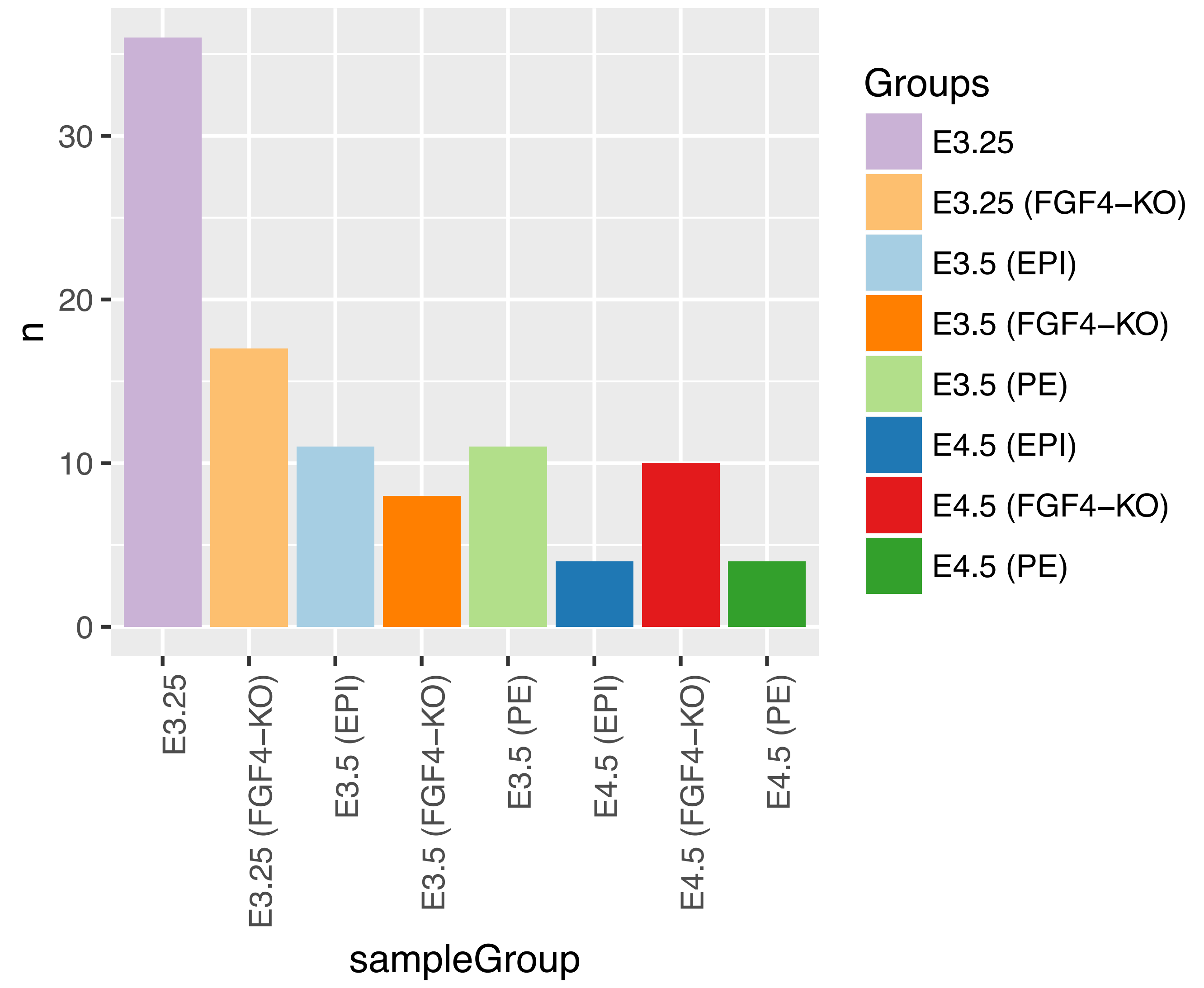
ZUSE Plotter Z64 (presented in 1961).

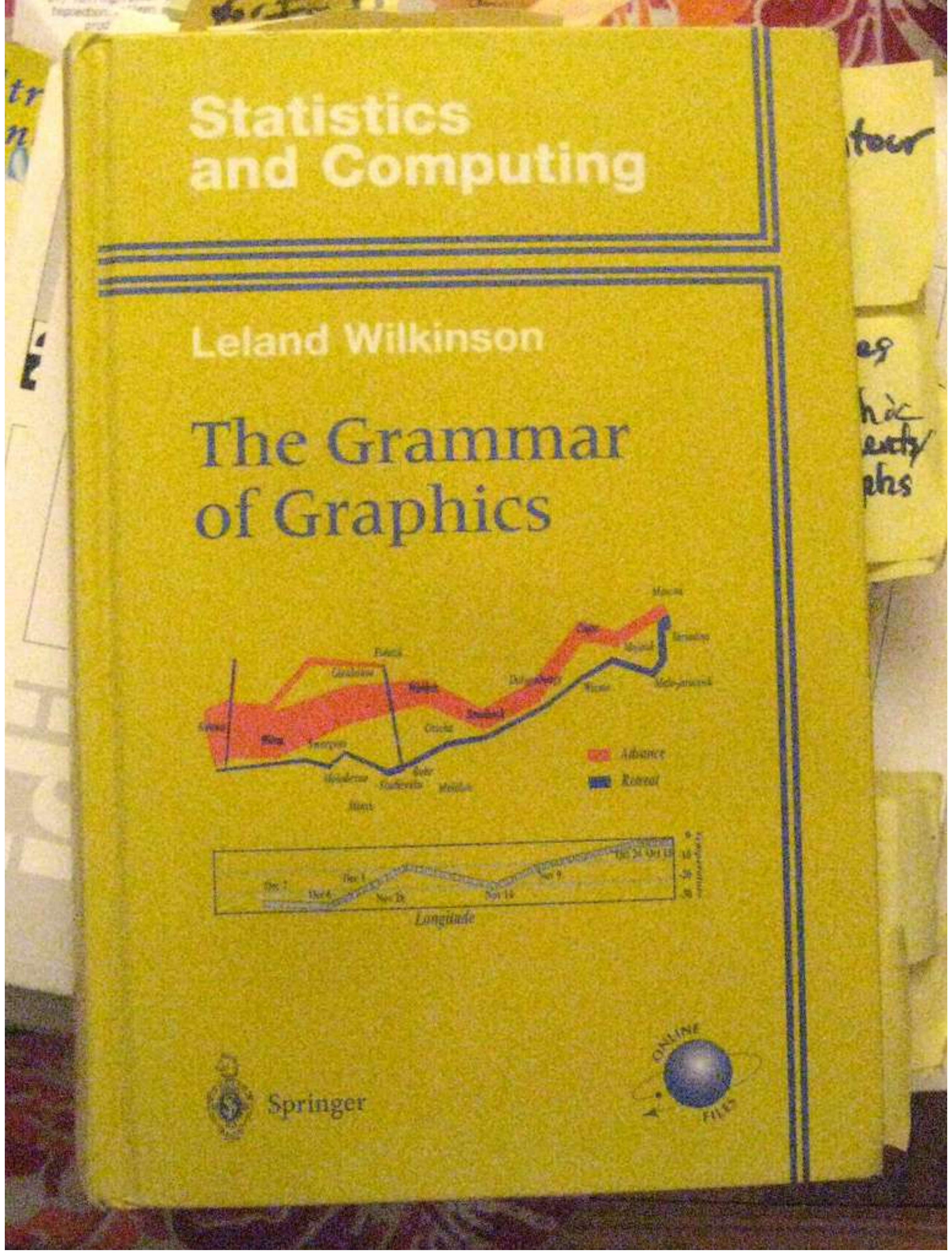
The grammar of graphics

The components of ggplot2's grammar of graphics are

- one or more datasets (“noun”),
- one or more geometric objects that serve as the visual represen-tations of the data, for instance, points, lines, rectangles, contours (“verb”),
- descriptions of how the variables in the data are mapped to visual properties (aesthetics) of the geometric objects, and an associated scale (e.g. linear, logarithmic, rank),
- one or more coordinate systems,
- statistical summarization rules (e.g. line fit, binning),
- a facet specification, i.e. multiple similar subplots to look at subsets of the same data,
- optional parameters for layout and rendering, e.g., text size, font, alignment; legend positions

```r
ggplot(groups, aes(x = sampleGroup, y = n, fill = sampleGroup)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = groupColour, name = "Groups") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```





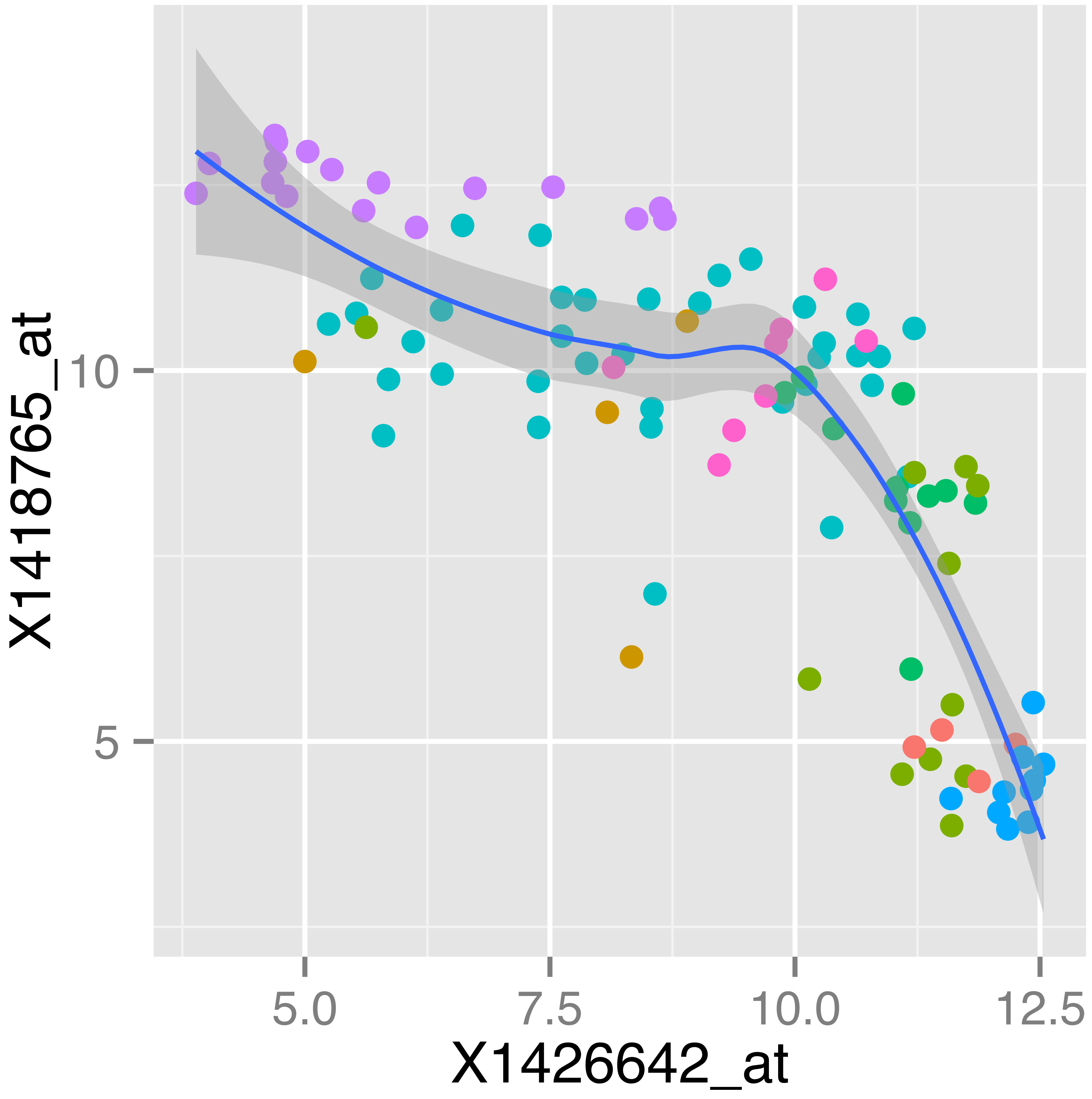
1999

A Layered Grammar of Graphics

Hadley Wickham
Journal of Computational and Graphical Statistics, 2010
Volume 19, Number 1, Pages 3–28
DOI: 10.1198/jcgs.2009.07098

Layers

```r
ggplot( dftx, aes( x = X1426642_at, y = X1418765_at ))  +
  geom_point( aes( colour = sampleColour), shape = 19 ) +
  geom_smooth( method = "loess" ) +
  scale_colour_discrete( guide = FALSE )
```



A more complex example: themes

```r
pb <- ggplot(data.frame(
                 name = names(groupSize),
                 size = as.vector(groupSize)),
              aes(x = name, y = size))
```

A more complex example: themes

```
pb <- ggplot(data.frame(
                 name = names(groupSize),
                 size = as.vector(groupSize)),
              aes(x = name, y = size))
```

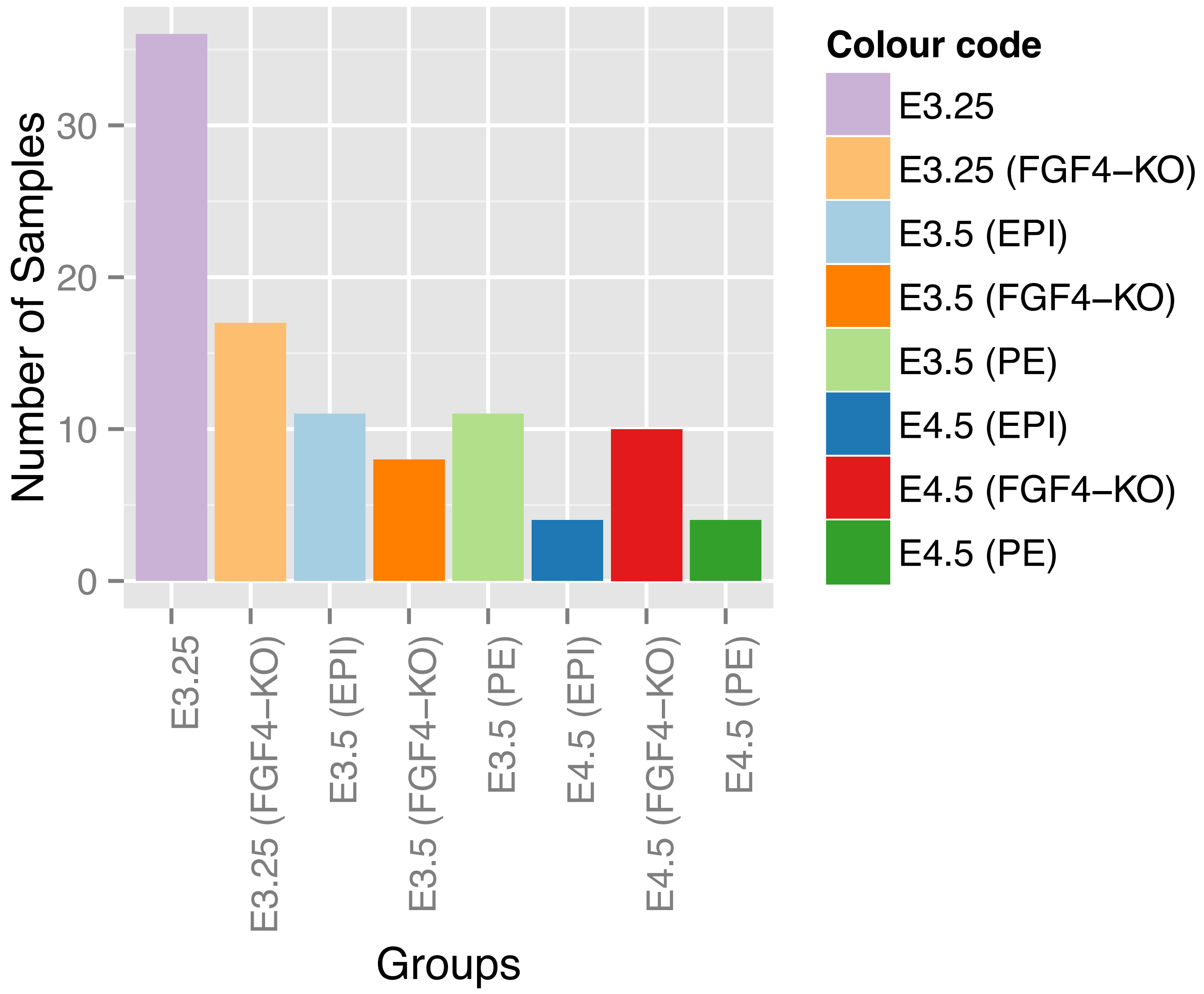
No geom defined yet!

A more complex example: themes

```
pb <- ggplot(data.frame(
                 name = names(groupSize),
                 size = as.vector(groupSize)),
              aes(x = name, y = size))
```

No geom defined yet!

```
pb <- pb + geom_bar(stat = "identity") +
   aes(fill = name) +
   scale_fill_manual(values = groupColour, name = "Colour code") +
   theme(axis.text.x = element_text(angle = 90, hjust = 1))  +
   xlab("Groups") + ylab("Number of Samples")
```

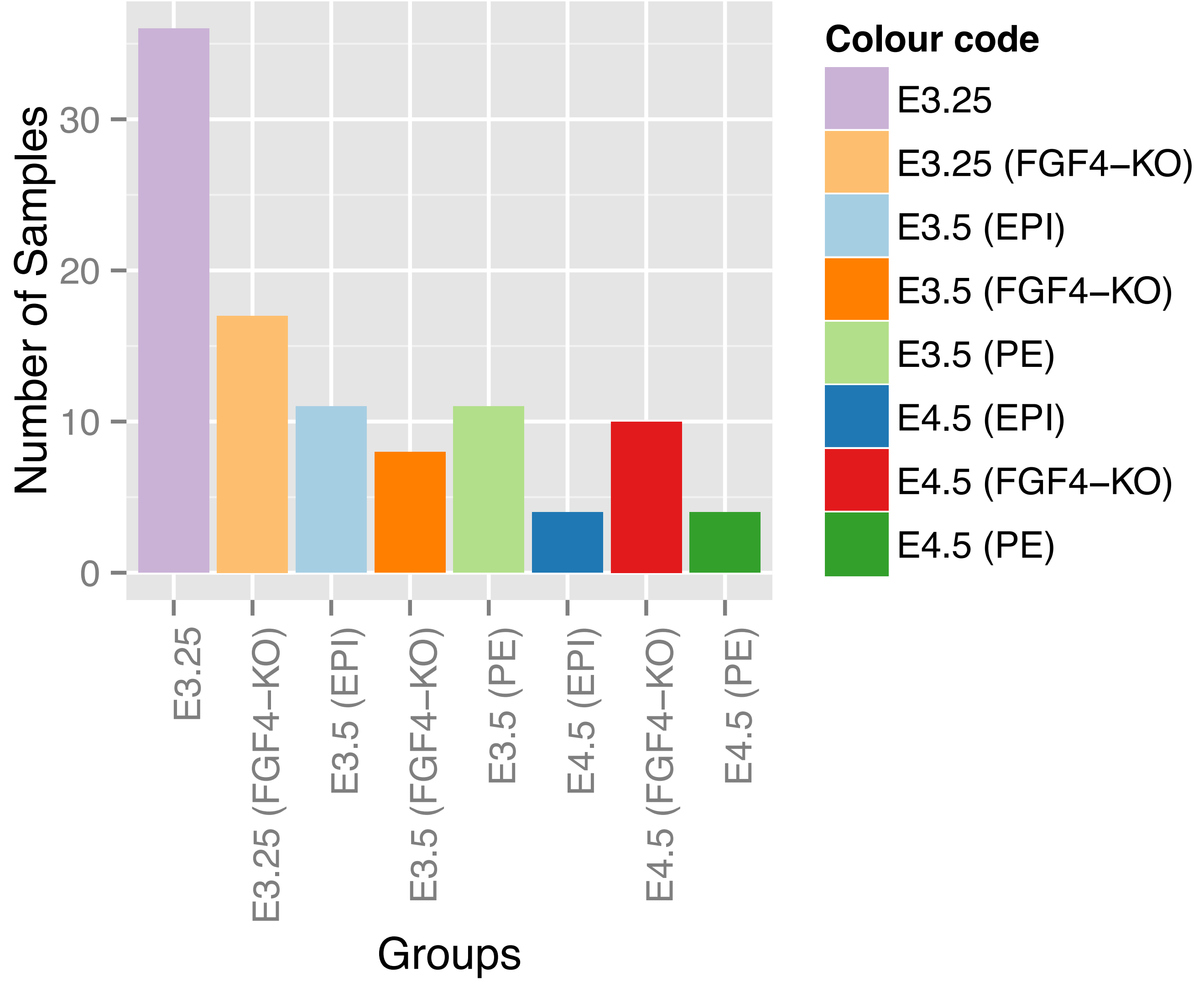


A more complex example: themes

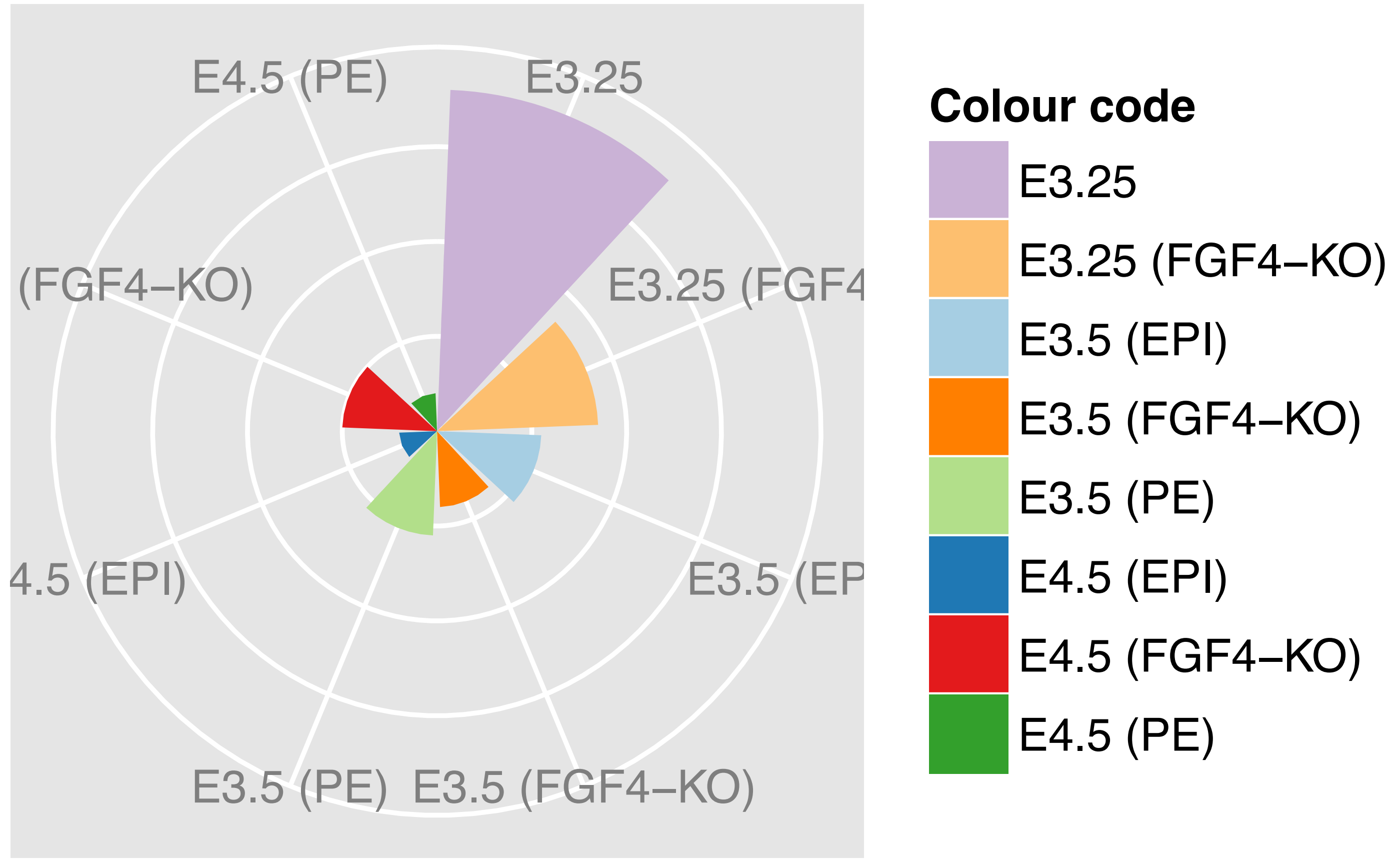
```
pb <- ggplot(data.frame(
                 name = names(groupSize),
                 size = as.vector(groupSize)),
              aes(x = name, y = size))
```

No geom defined yet!

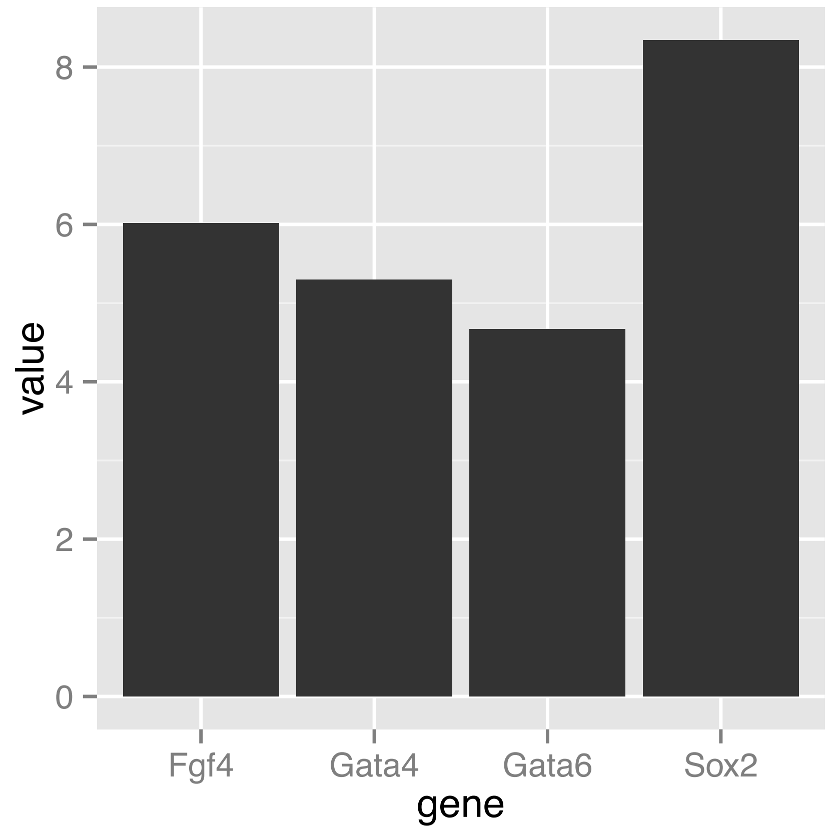
```
pb <- pb + geom_bar(stat = "identity") +
   aes(fill = name) +
   scale_fill_manual(values = groupColour, name = "Colour code") +
   theme(axis.text.x = element_text(angle = 90, hjust = 1))  +
   xlab("Groups") + ylab("Number of Samples")
```

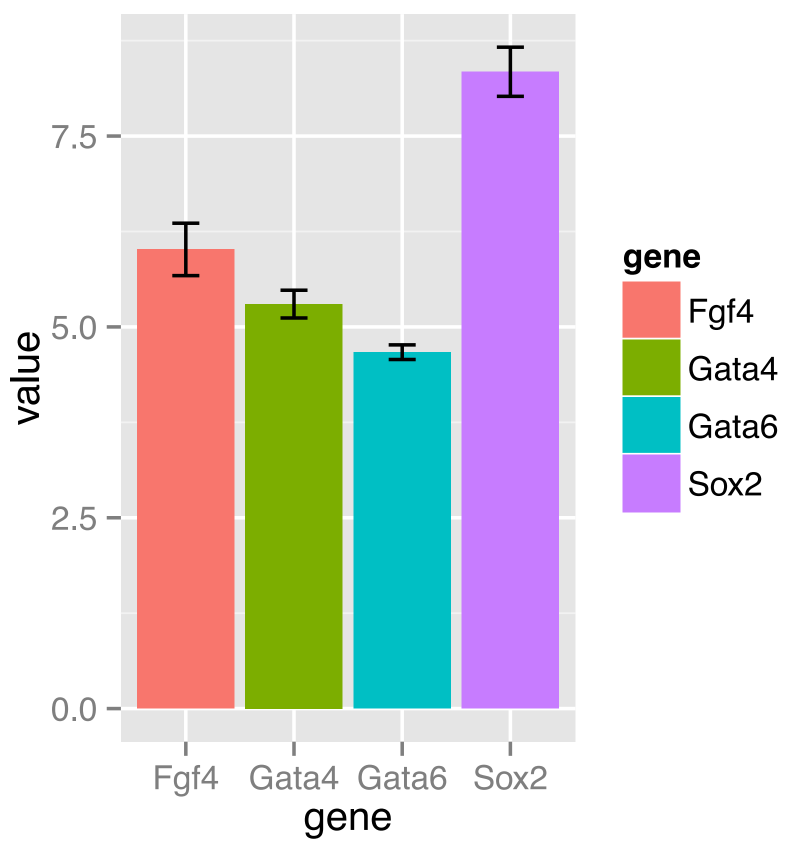


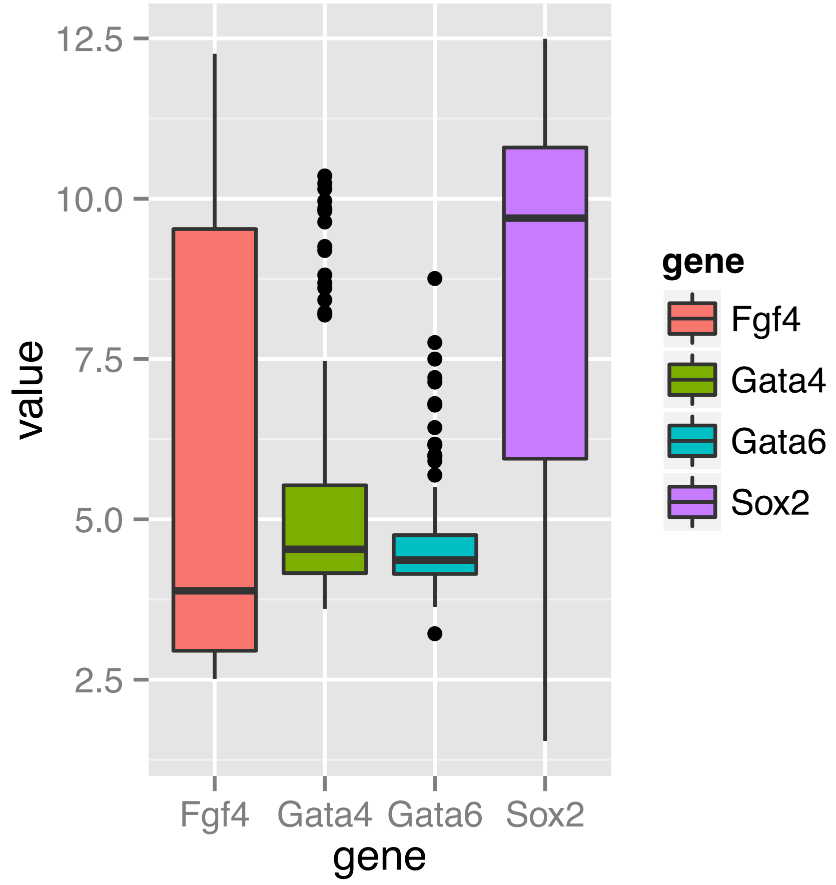
```
pb.polar <- pb + coord_polar() +
  theme(axis.text.x = element_text(angle = 0, hjust = 1),
        axis.text.y = element_blank(),
        axis.ticks = element_blank()) +
  xlab("") + ylab("")
pb.polar
```

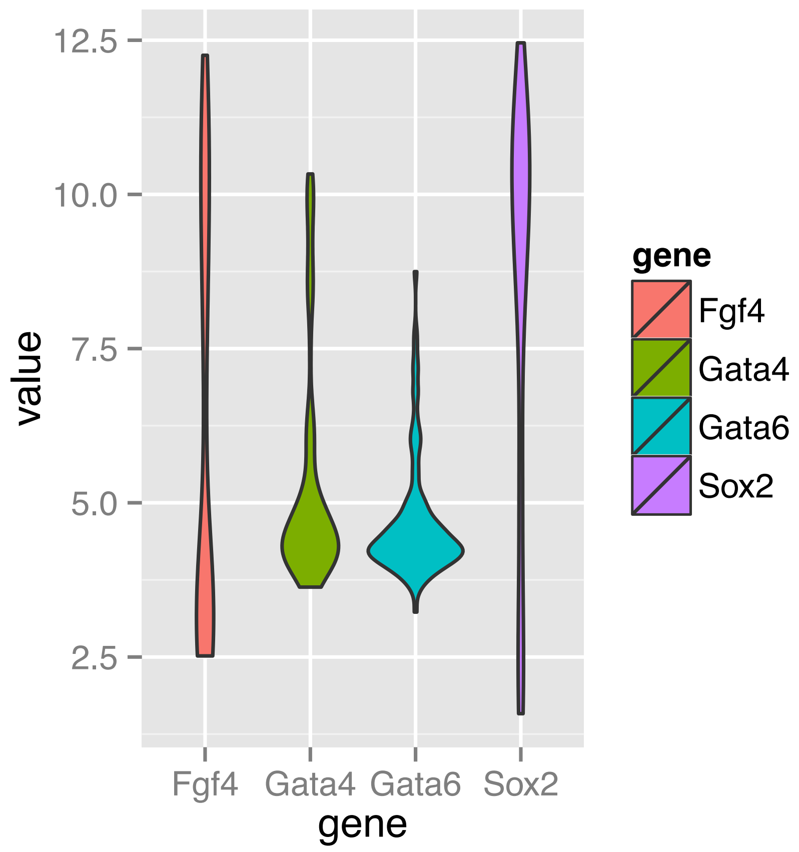


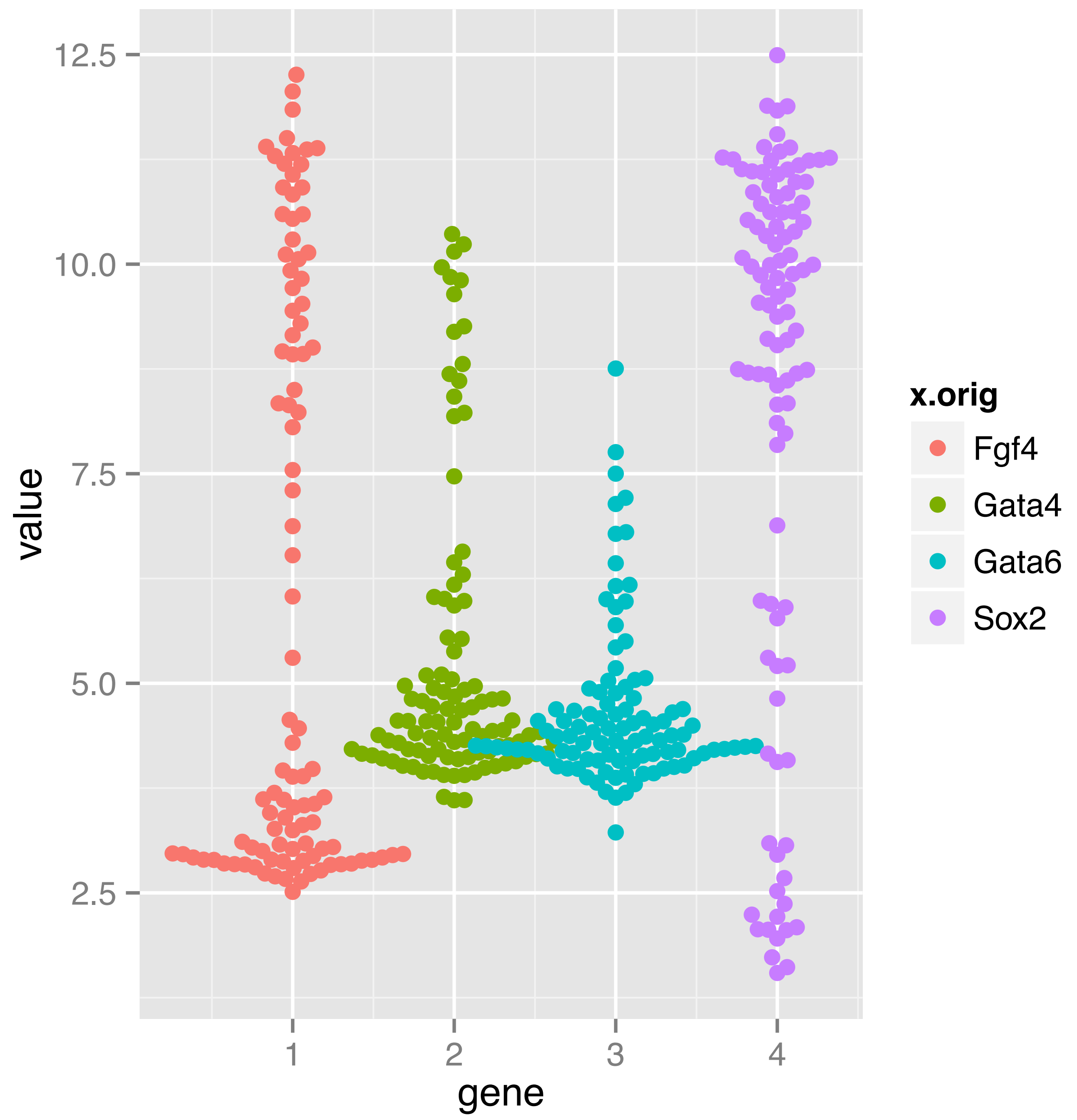
Visualizing distributions in 1D

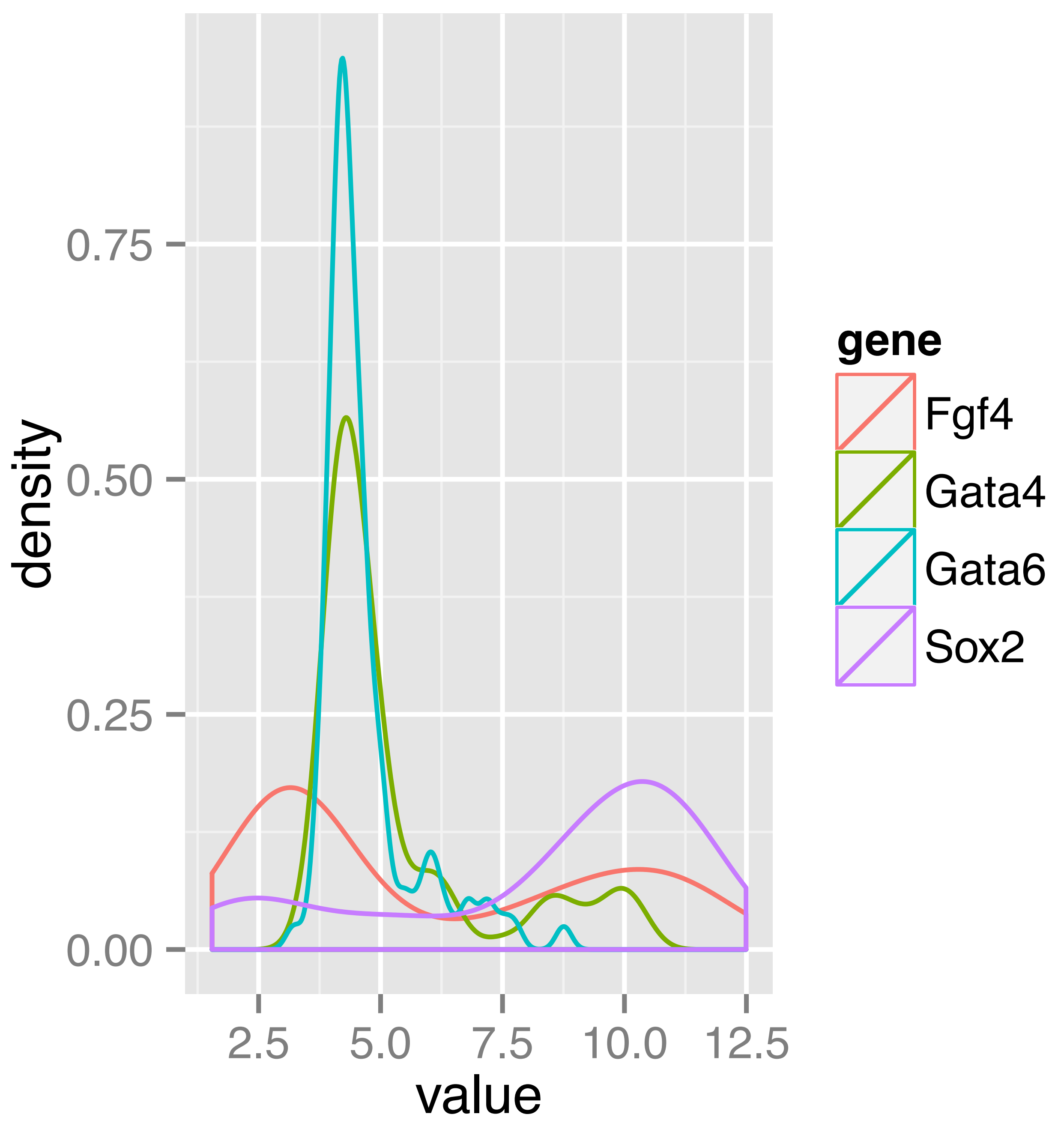


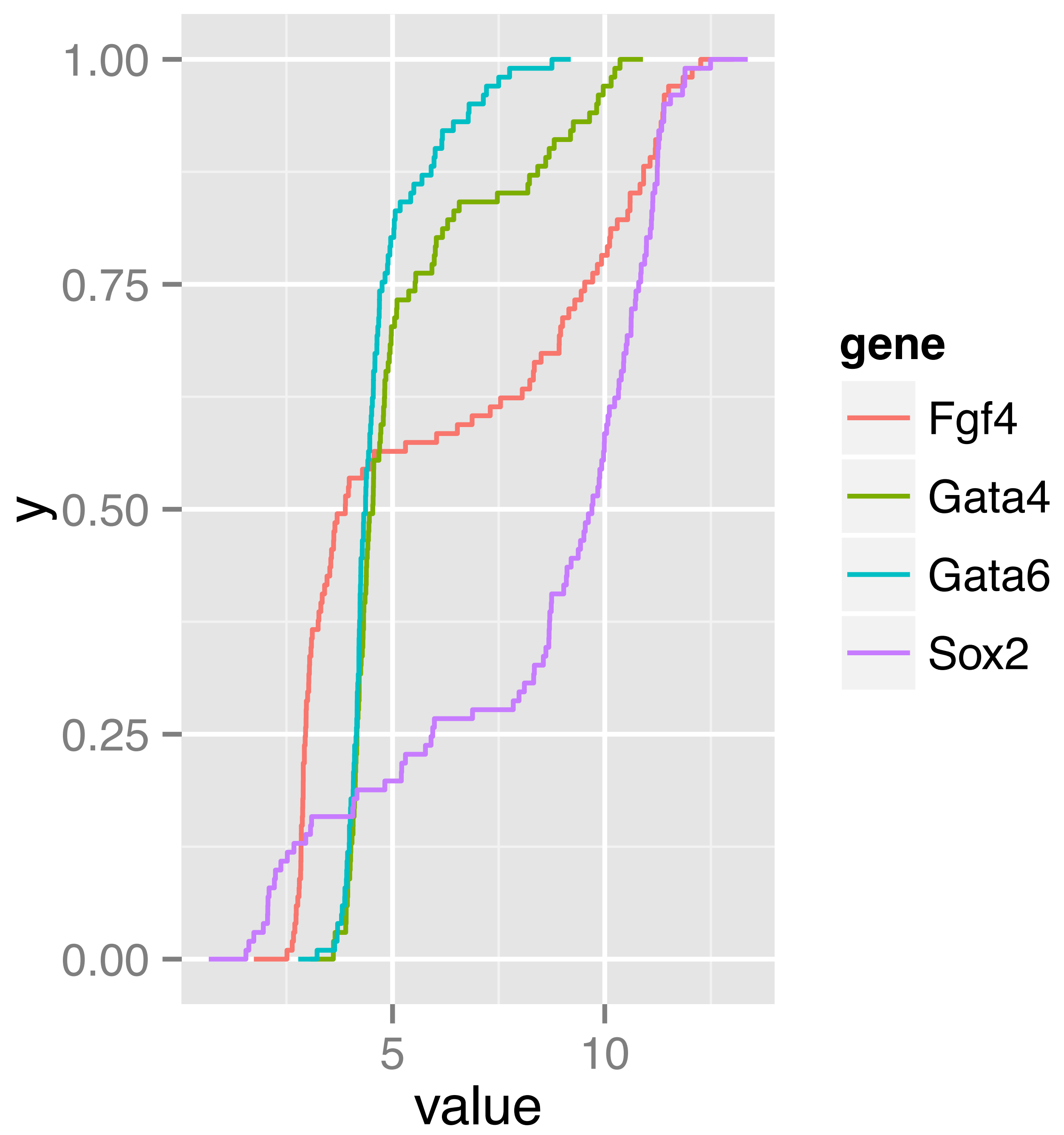












1D plot types

Boxplot makes sense for unimodal distributions

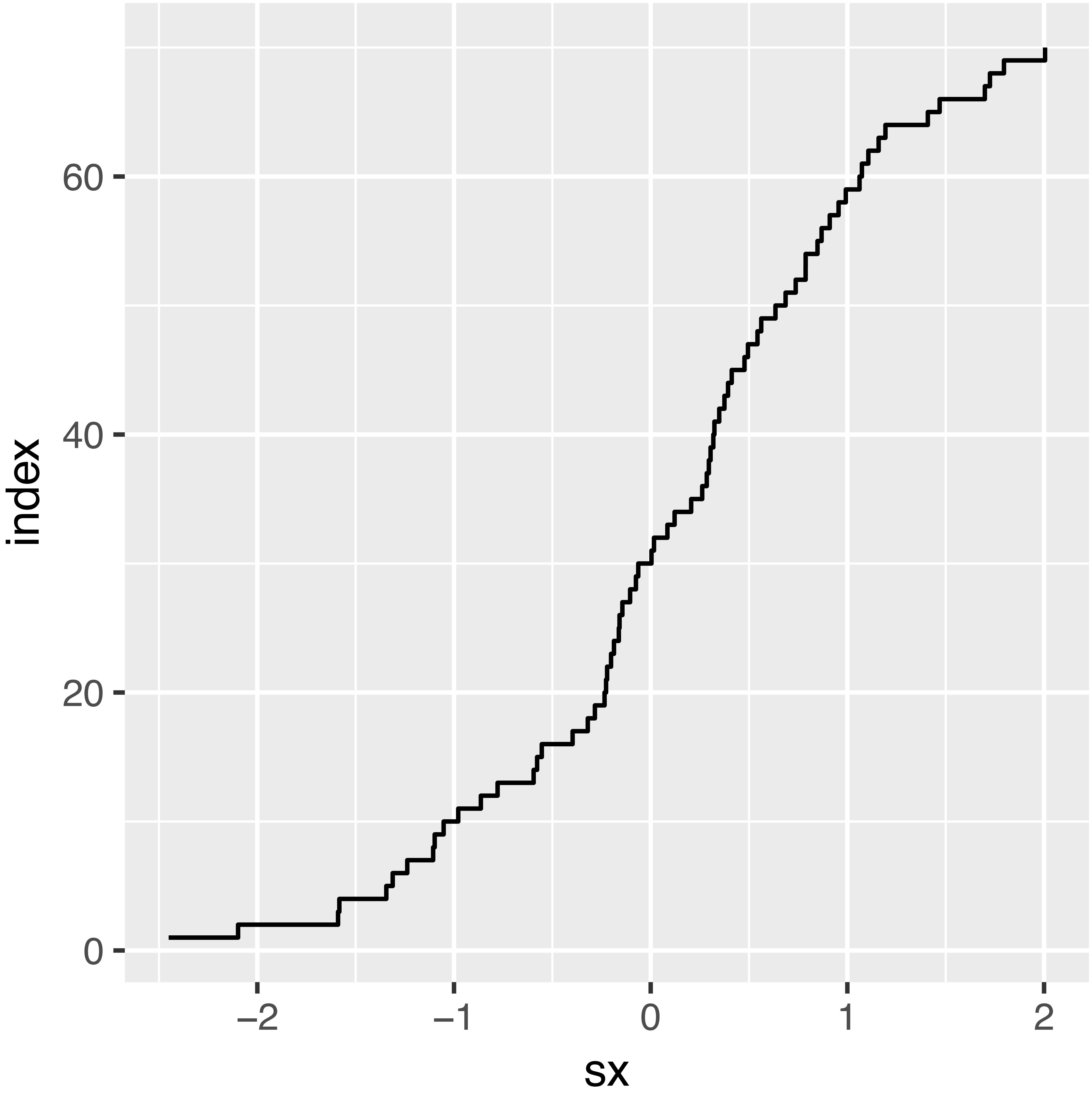
Histogram requires definition of bins (width, positions) and can create visual artifacts esp. if the number of data points is not large

Density requires the choice of bandwidth; obscures the sample size (i.e. the uncertainty of the estimate)

ecdf does not have these problems; but is more abstract and interpretation requires more training. Good for reading off quantiles and shifts in location in comparative plots; OK for detecting differences in scale; less good for detecting multimodality.

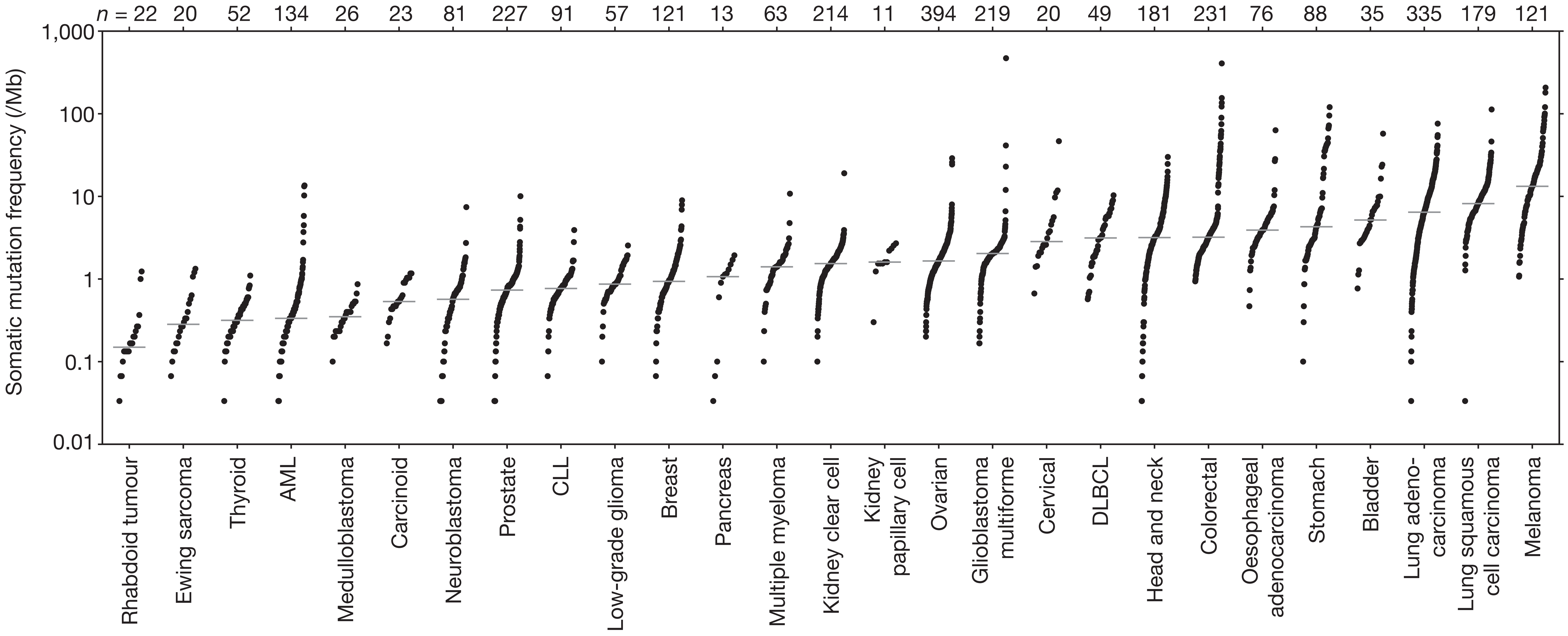
Up to a few dozens of points - just show the data! (beeswarm)

The empirical cumulative distribution function



$$F_n(x) = \frac{\text{number of } i \text{ for which } x_i \le x}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(x \le x_i)$$

```r
simdata = rnorm(70)
tibble(index = seq(along = simdata),
          sx = sort(simdata)) %>%
ggplot(aes(x = sx, y = index)) + geom_step()
```



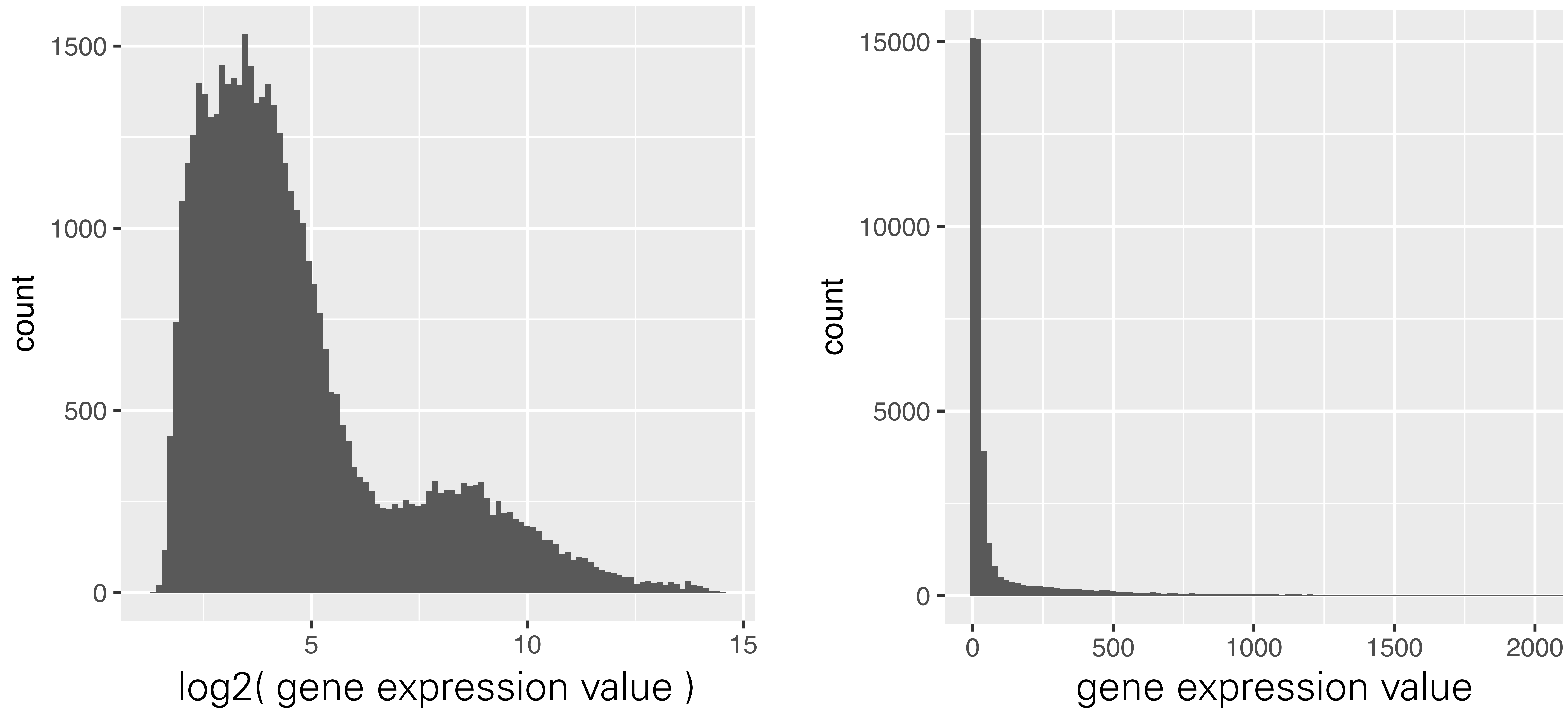
LETTER

doi:10.1038/nature12213

Mutational heterogeneity in cancer and the search for new cancer-associated genes

Michael S. Lawrence[1]*, Petar Stojanov[1,2]*, Paz Polak[1,3,4]*, Gregory V. Kryukov[1,3,4], Kristian Cibulskis[1], Andrey Sivachenko[1], Scott L. Carter[1], Chip Stewart[1], Craig H. Mermel[1,5], Steven A. Roberts[6], Adam Kiezun[1], Peter S. Hammerman[1,2], Aaron McKenna[1,7], Yotam Drier[1,3,5,8], Lihua Zou[1], Alex H. Ramos[1], Trevor J. Pugh[1,2,3], Nicolas Stransky[1,9], Elena Helman[1,10], Jaegil Kim[1], Carrie Sougnez[1], Lauren Ambrogio[1], Elizabeth Nickerson[1], Erica Shefler[1], Maria L. Cortés[1], Daniel Auclair[1], Gordon Saksena[1], Douglas Voet[1], Michael Noble[1], Daniel DiCara[1], Pei Lin[1], Lee Lichtenstein[1], David I. Heiman[1], Timothy Fennell[1], Marcin Imielinski[1,5], Bryan Hernandez[1], Eran Hodis[1,2], Sylvan Baca[1,2], Austin M. Dulak[1,2], Jens Lohr[1,2], Dan-Avi Landau[1,2,11], Catherine J. Wu[2,3], Jorge Melendez-Zajgla[12], Alfredo Hidalgo-Miranda[12], Amnon Koren[1,3], Steven A. McCarroll[1,3], Jaume Mora[13], Ryan S. Lee[2,3,14], Brian Crompton[2,14], Robert Onofrio[1], Melissa Parkin[1], Wendy Winckler[1], Kristin Ardlie[1], Stacey B. Gabriel[1], Charles W. M. Roberts[2,3,14], Jaclyn A. Biegel[15], Kimberly Stegmaier[1,2,14], Adam J. Bass[1,2,3], Levi A. Garraway[1,2,3], Matthew Meyerson[1,2,3], Todd R. Golub[1,2,3,8], Dmitry A. Gordenin[6], Shamil Sunyaev[1,3,4], Eric S. Lander[1,3,10] & Gad Getz[1,5]

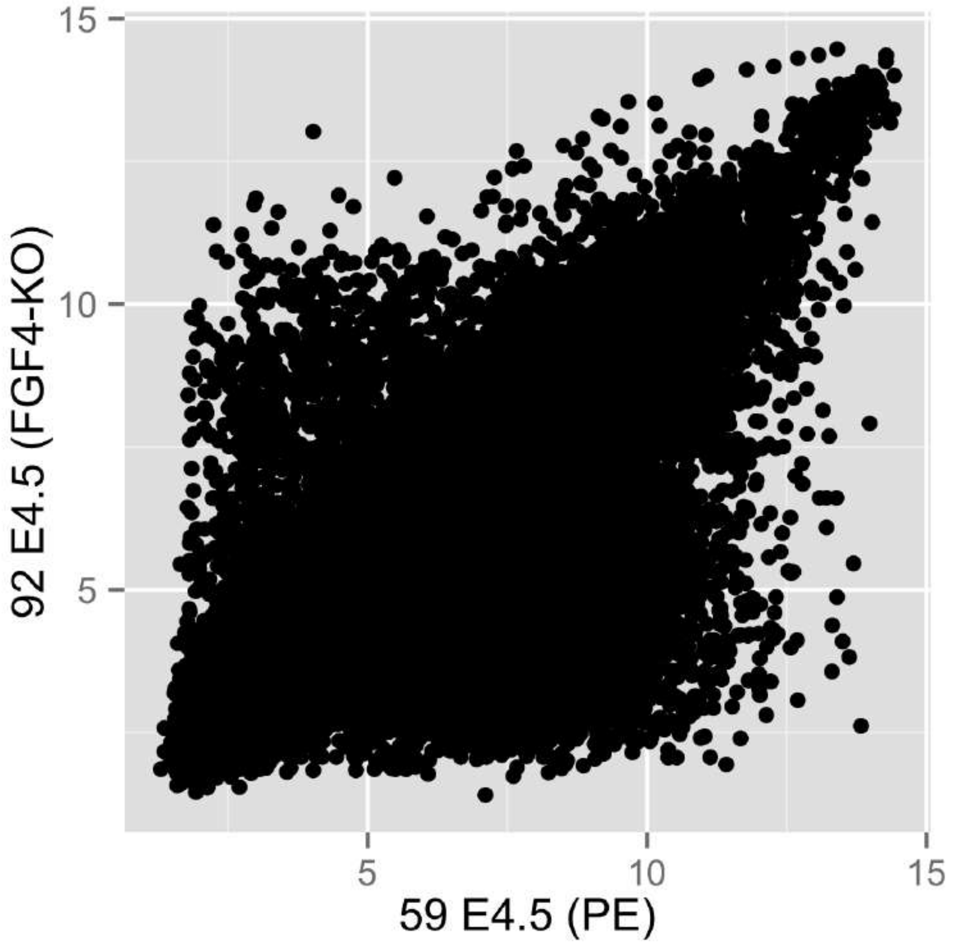
Impact of non-linear transformation on the shape of a density

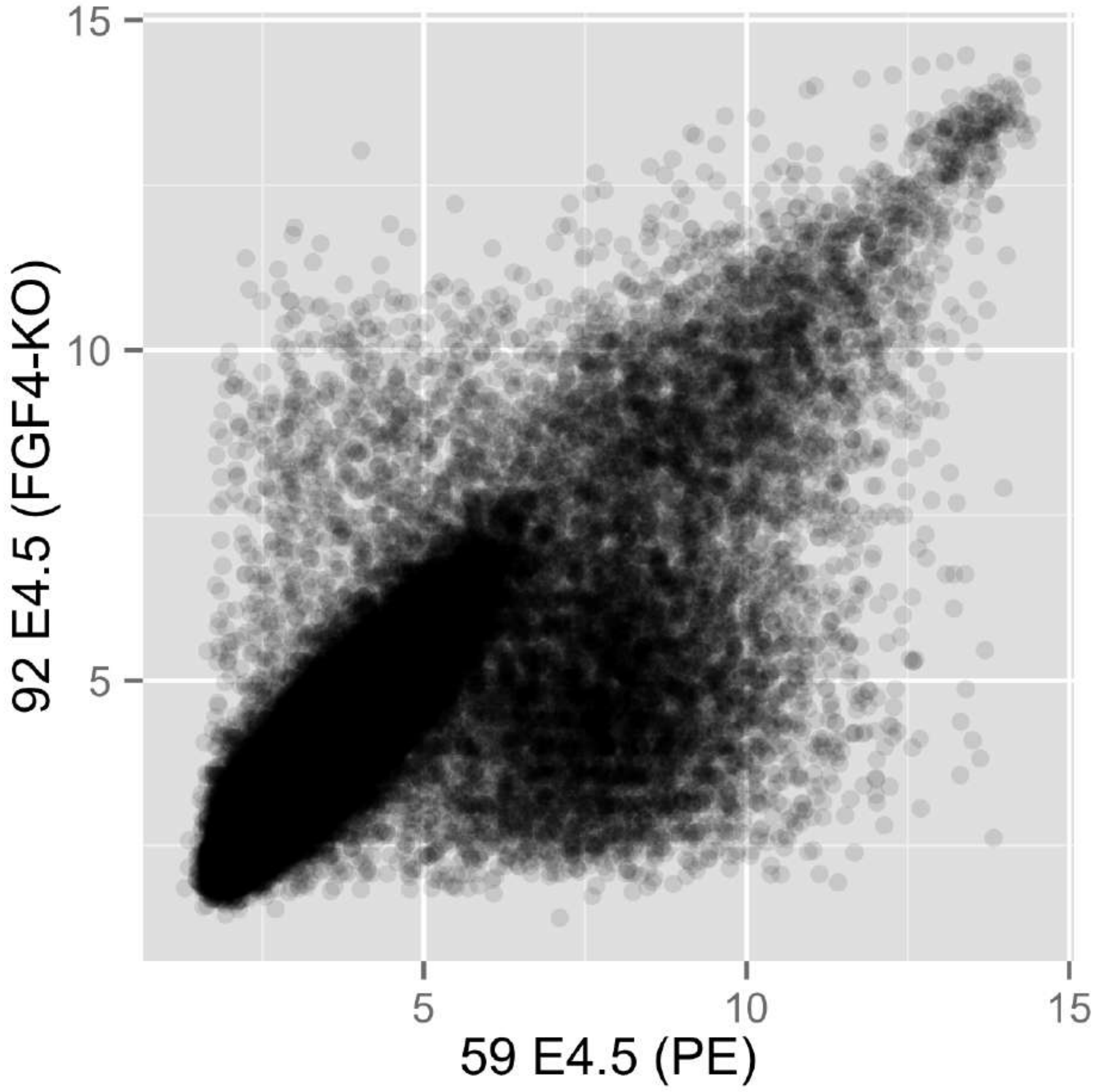


- The mode of a distribution is an infinitesimal concept.
- Need either an infinite amount of data or choose smoothing / binning bandwidth
- Number of modes (let alone their positions) can change under non-linear data transformations (Question 3.5 in the book)

Showing data in 2D

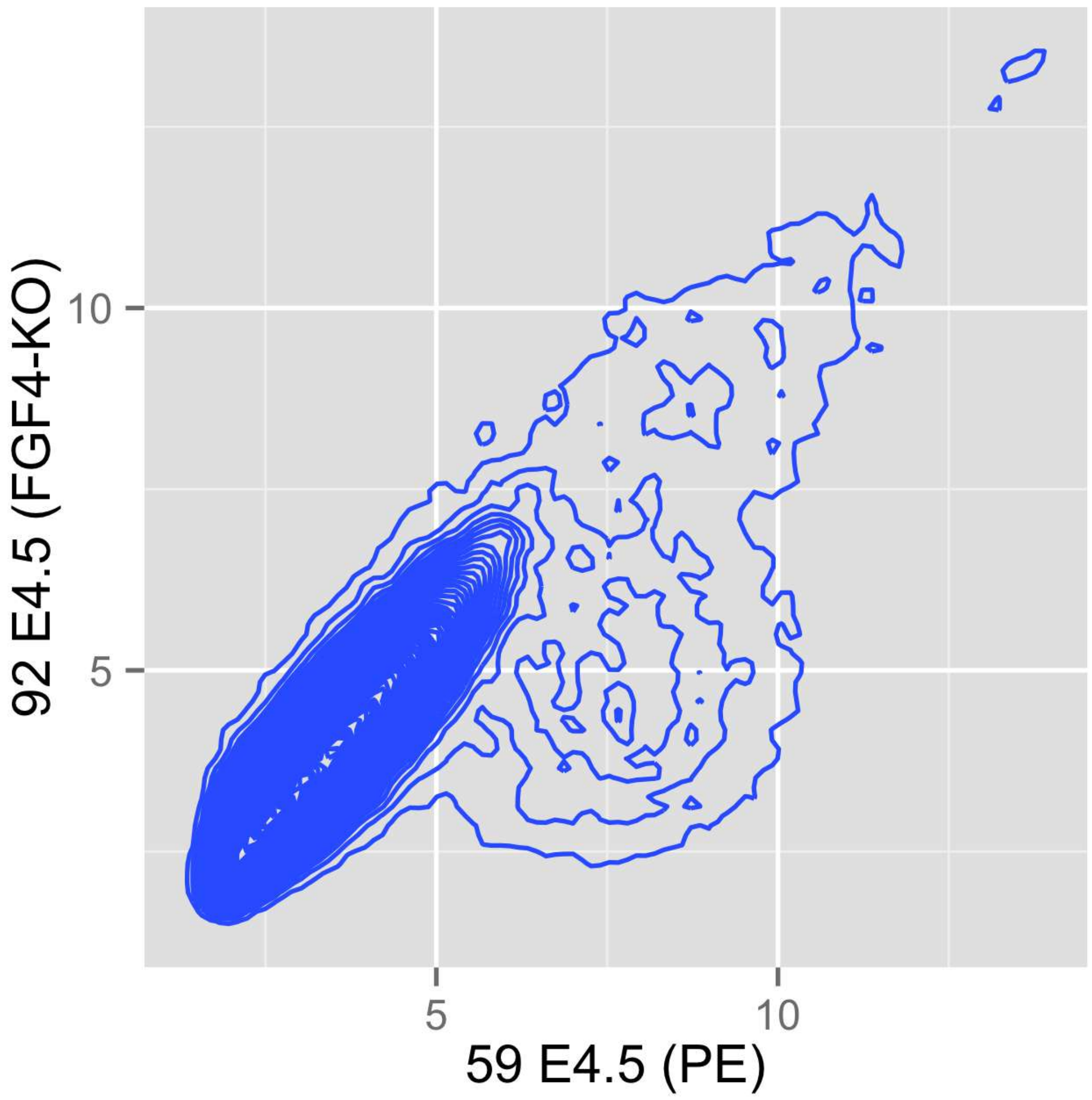
```
scp <- ggplot(dfx, aes( x = ‘59 E4.5 (PE)‘ ,
                        y = ‘92 E4.5 (FGF4-KO)‘))
scp + geom_point()
```



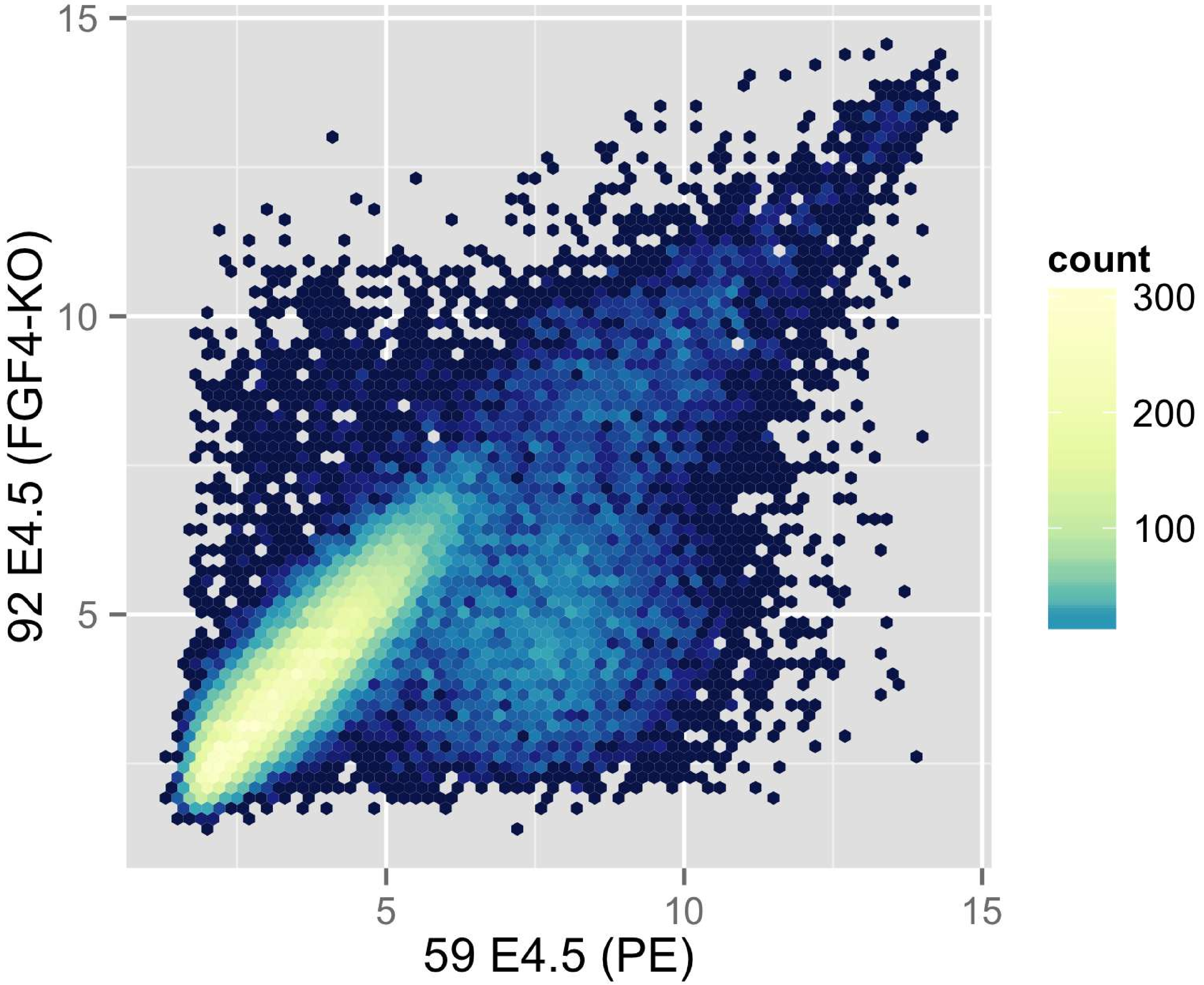


```
scp  + geom_point(alpha = 0.1)
```

```
scp + geom_density2d(h = 0.5, bins = 60)
```



Showing data in 2D



```
scp + stat_binhex(binwidth = c(0.2, 0.2)) + colourscale +
  coord_fixed()
```

Plot shape, banking

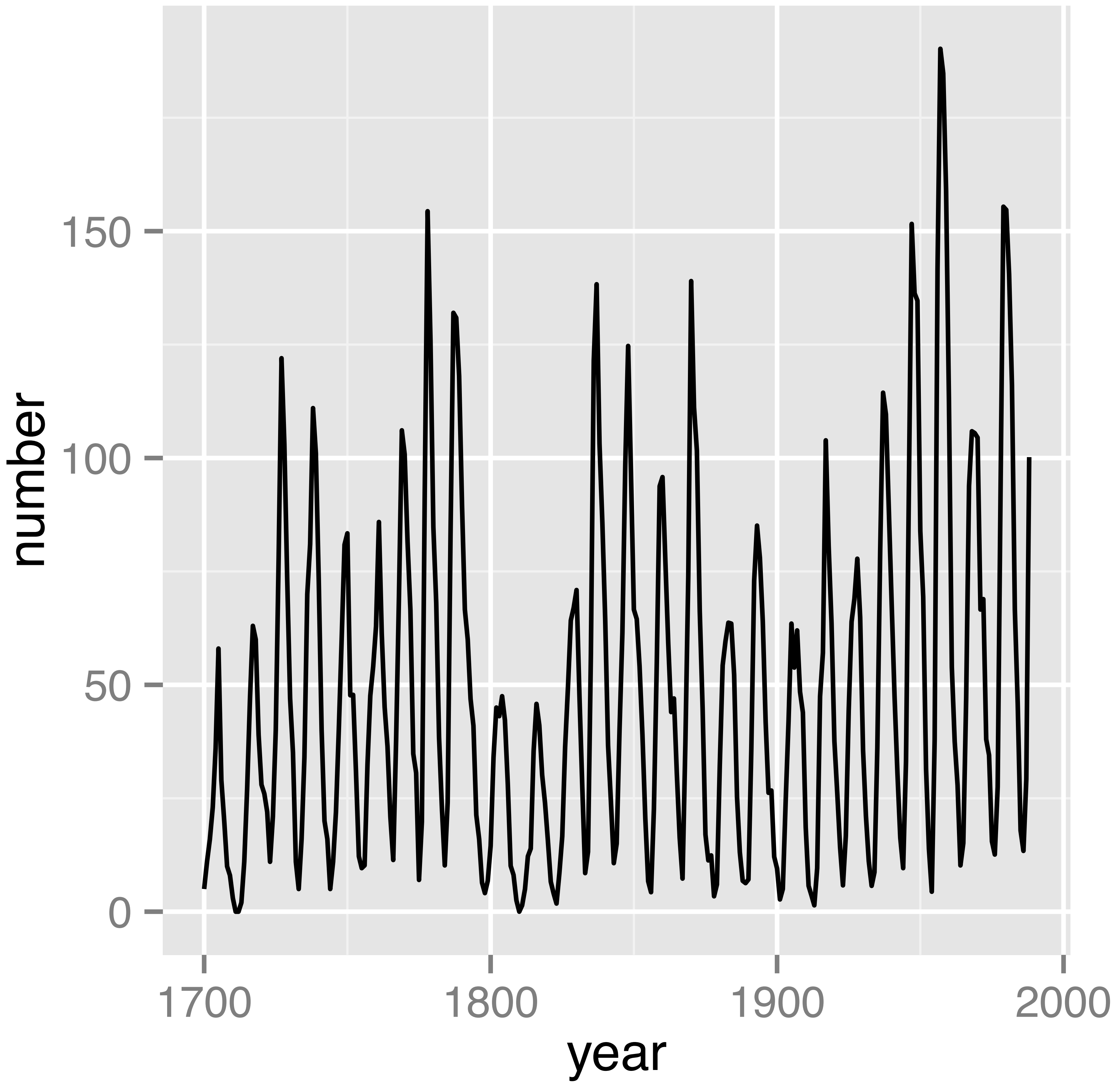
Yearly sunspot numbers 1849-1924

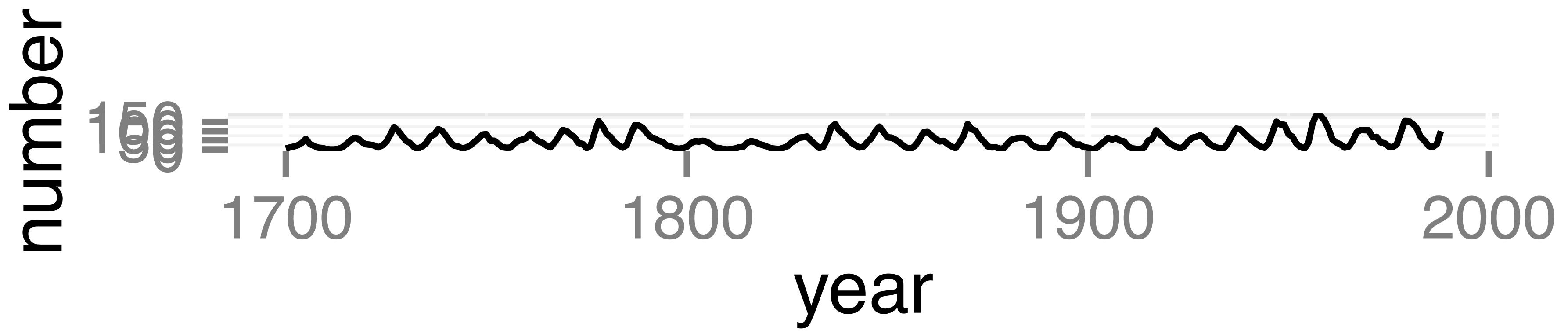
Changes in amplitude

Banking to 45 degrees:

Choose aspect ratio so that center of absolute values of slopes is 45 degrees

Sawtooth: Sunspot cycles typically rise more rapidly than they fall (pronounced for high peaks, less for medium and not for lowest)





Plot shape, banking

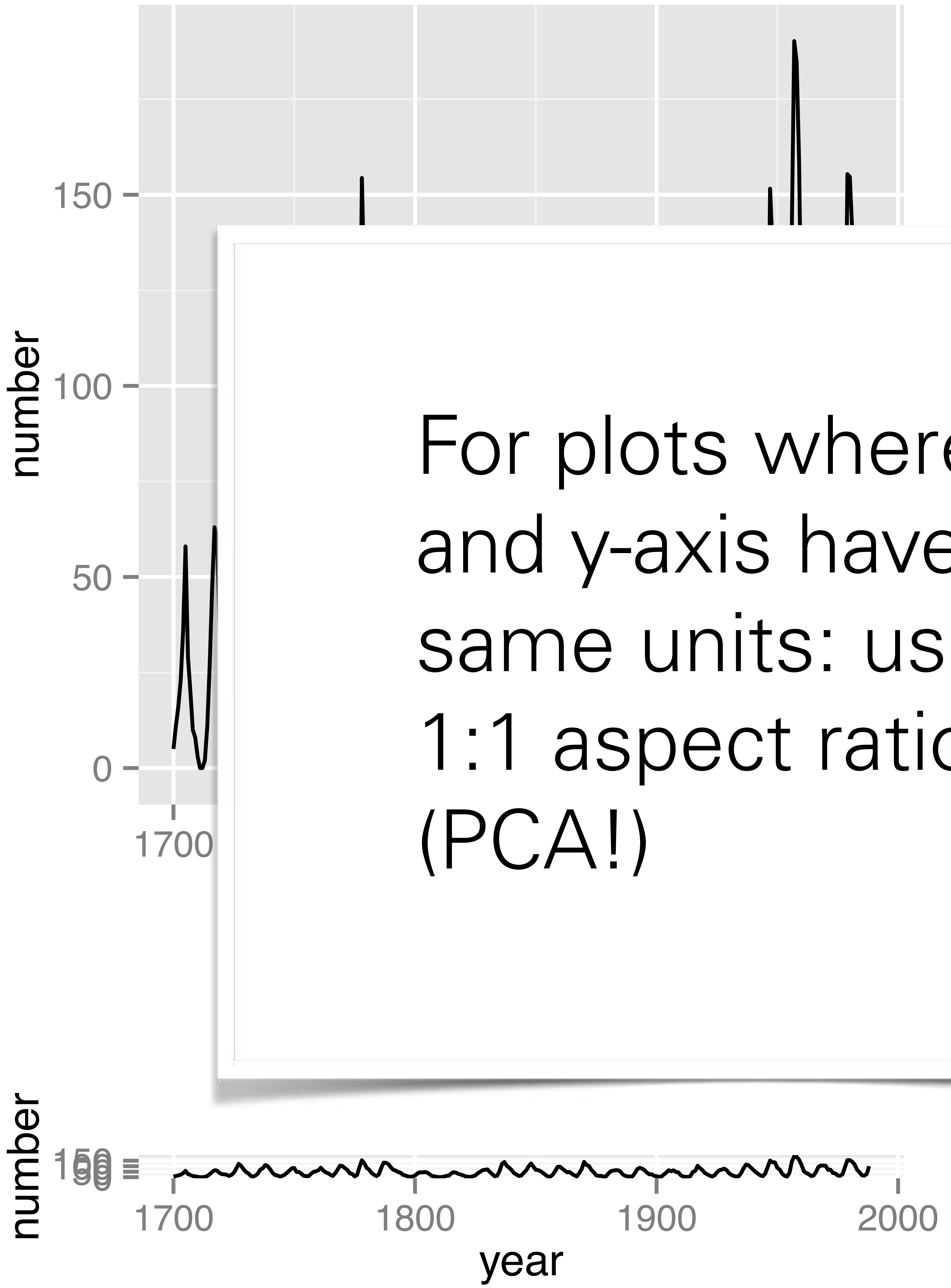
Yearly sunspot numbers 1849-1924

Changes in amplitude

Banking to 45 degrees:

Choose aspect ratio so that center of absolute values of slopes is 45 degrees

Sawtooth: Sunspot cycles typically rise more rapidly than they fall (pronounced for high peaks, less for medium and not for lowest)



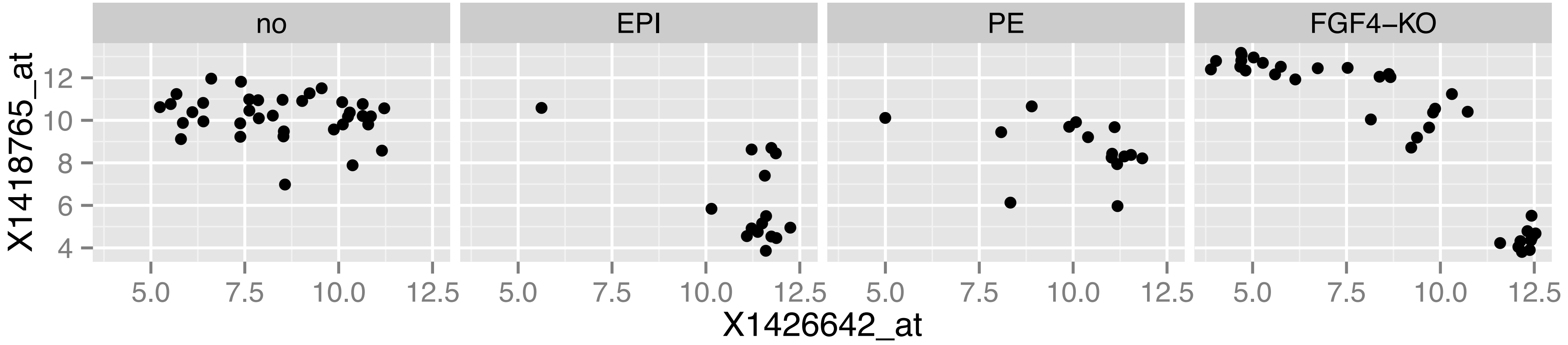
For plots where x- and y-axis have same units: use 1:1 aspect ratio (PCA!)

3-5 D and faceting

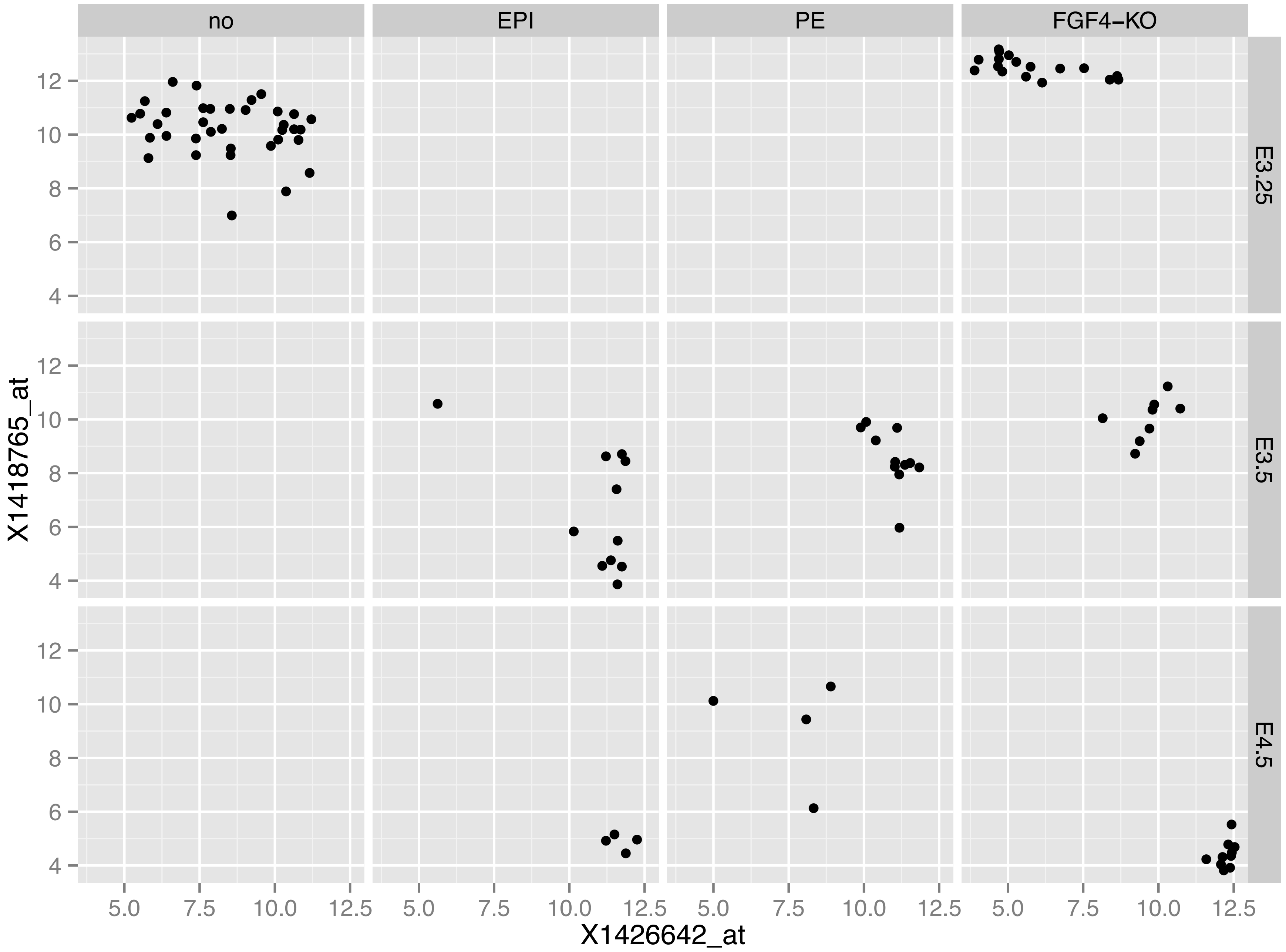
geom_point offers these aesthetics (beyond x and y):

- fill
- colour
- shape
- size
- alpha

```
ggplot(dftx, aes( x = X1426642_at, y = X1418765_at)) +
  geom_point() + facet_grid( . ~ lineage )
```



```
ggplot( dftx,
  aes( x = X1426642_at, y = X1418765_at)) + geom_point() +
   facet_grid( Embryonic.day ~ lineage )
```

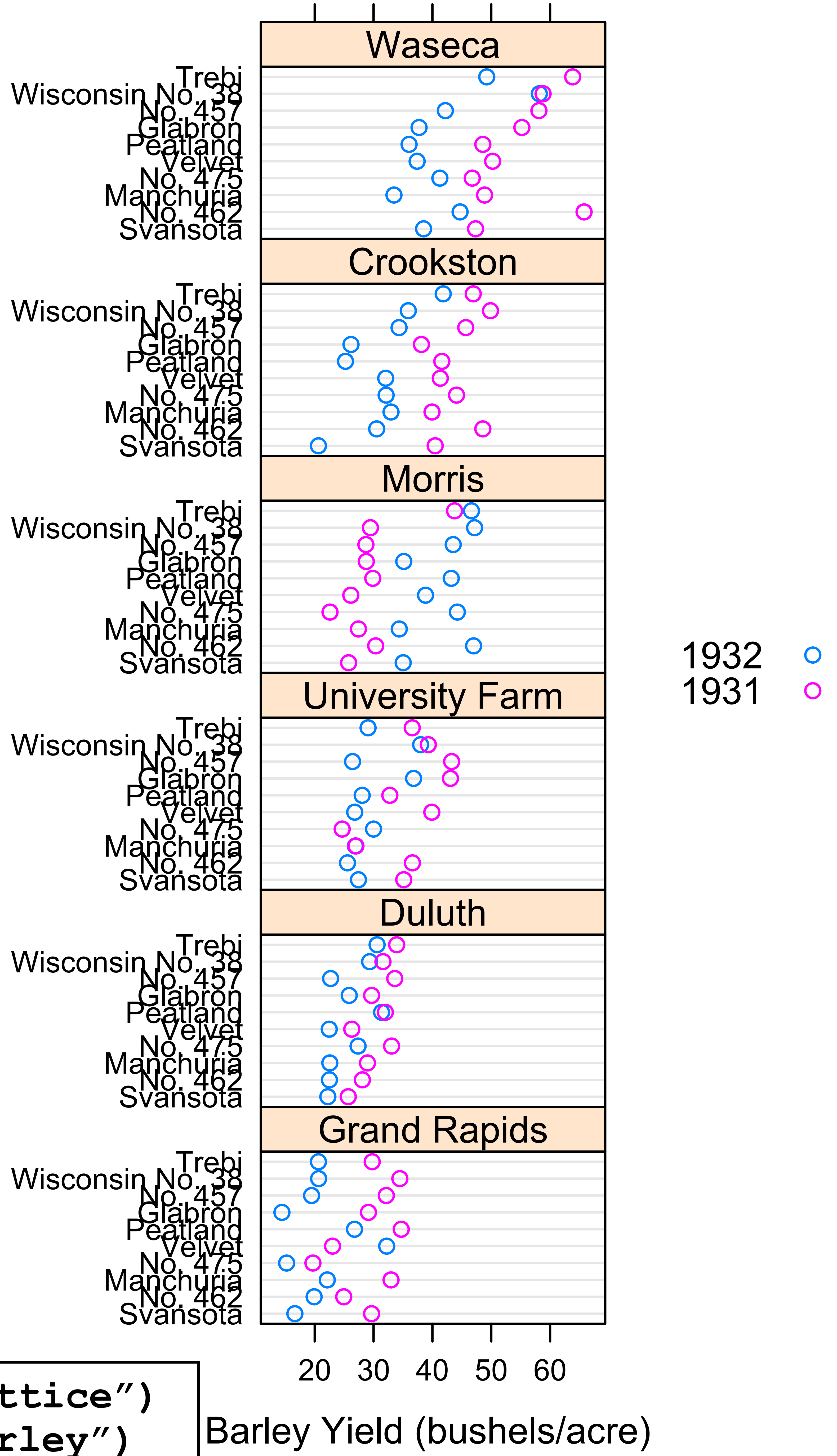


Data from an agricultural field trial to study the crop barley.

At 6 sites in Minnesota, 10 varieties of barley were grown in each of two years.

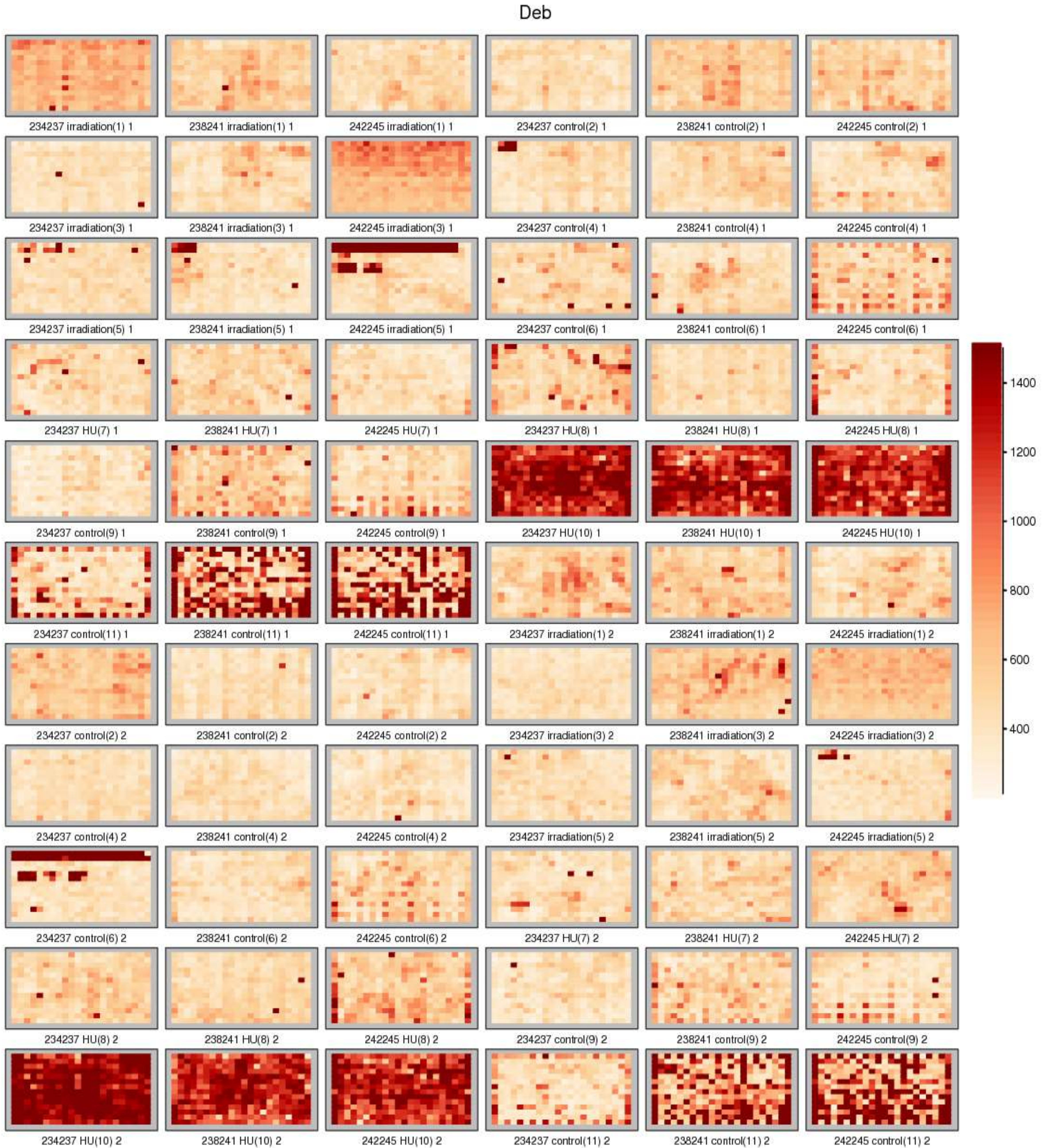
Data: yield, for all combinations of site, variety, and year (6 x 10 x 2 = 120 observations)

Note the data for Morris - reanalysis in the 1990s using Trellis revealed that the years had been flipped!



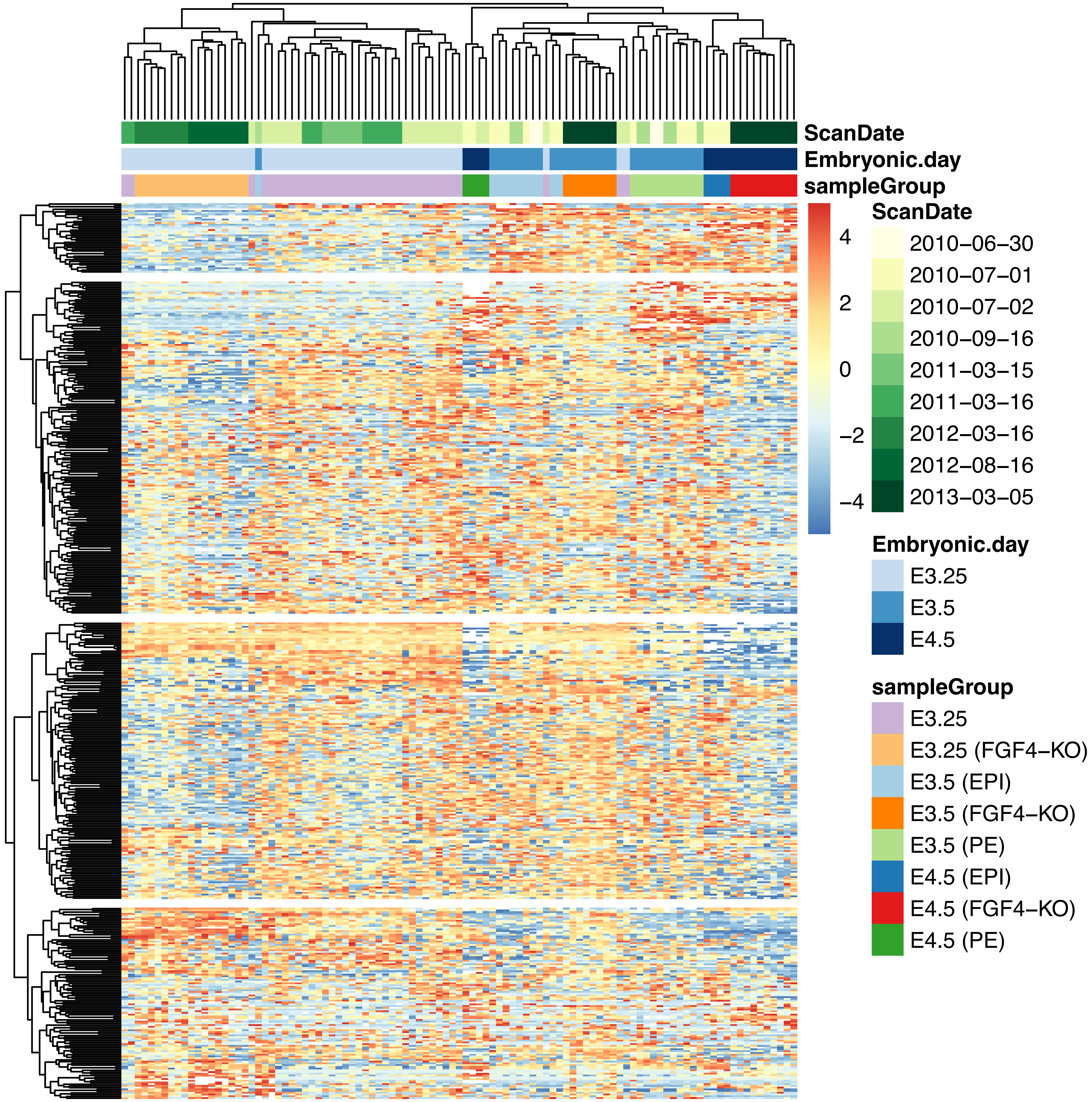
```
library("lattice")
example("barley")
```

EDA for finding batch effects



package
splots

pheatmap

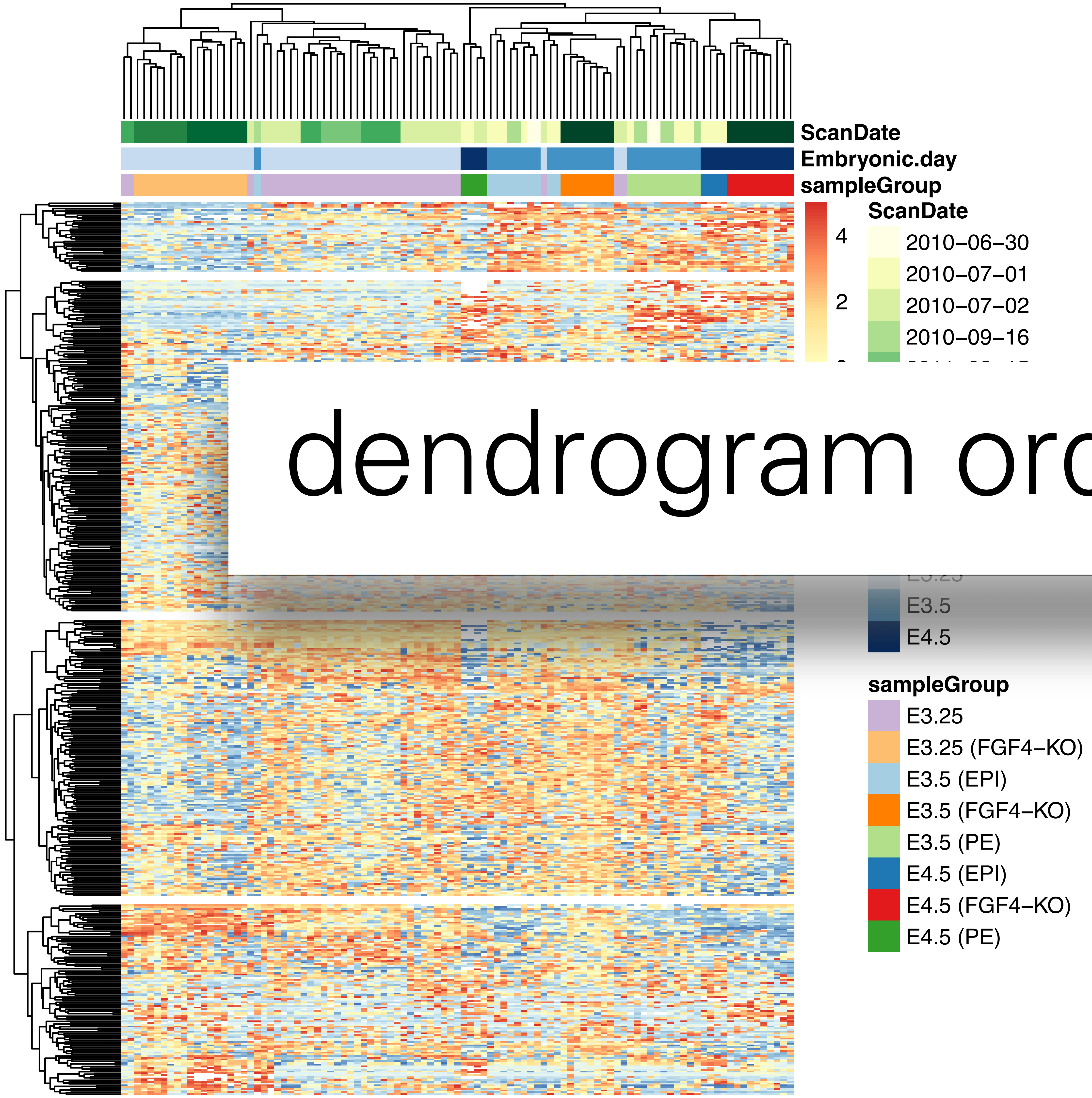


Many reasonable defaults

Easy to add column and row ‘metadata’ at the sides

See also *ComplexHeatmaps* package

pheatmap



dendrogram order

Many reasonable defaults

y to add column row ‘metadata’ e sides

See also *ComplexHeatmaps* package

Interactivity

Use shiny or plotly

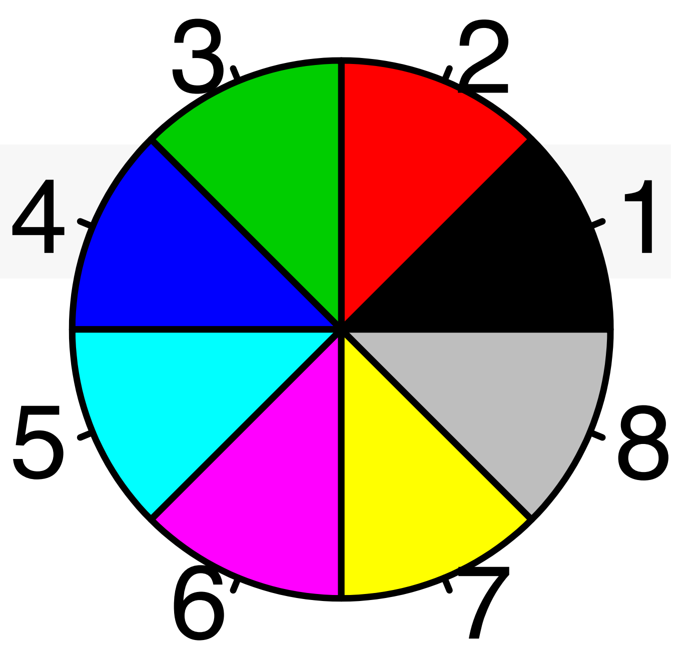
https://shiny.rstudio.com/gallery/genome-browser.html

Animations (time-dependent plots):
https://gganimate.com

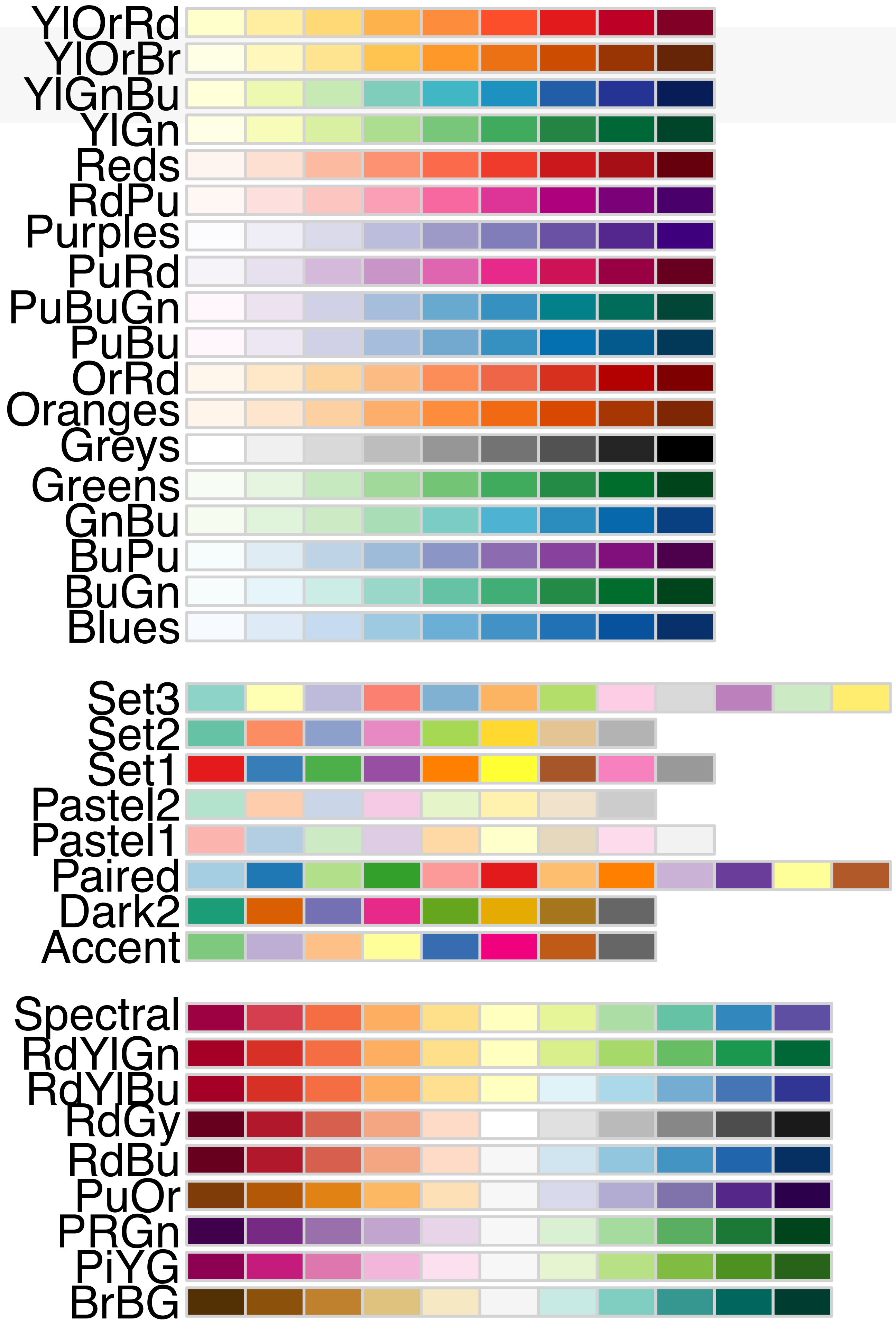
Linked Charts
https://anders-biostat.github.io/linked-charts/

NB: ggvis is senescent

```
pie(rep(1, 8), col=1:8)
```



```
display.brewer.all()
```



```
pie(rep(1, 8), c
```

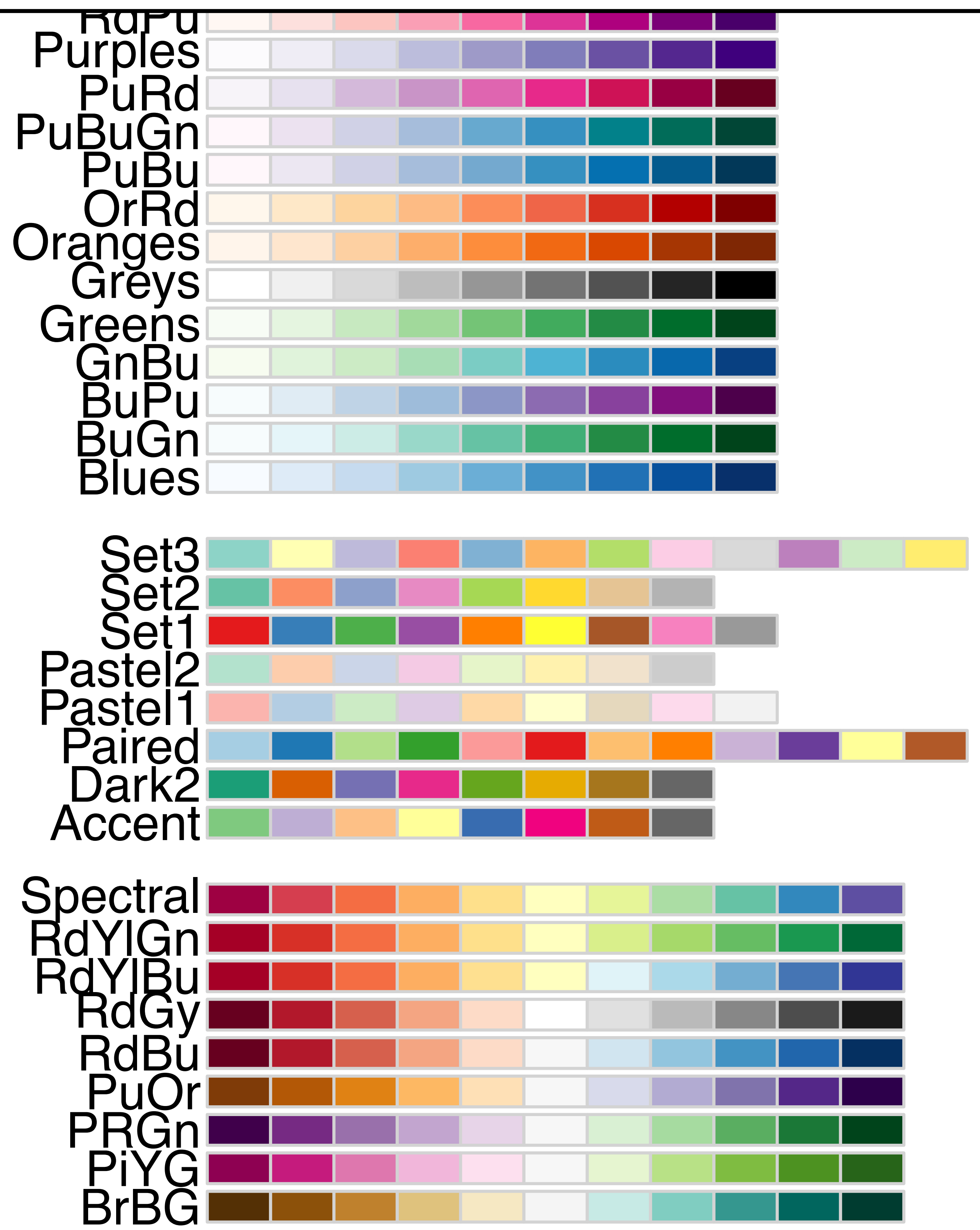
Consider these:

Different requirements for line & area colours

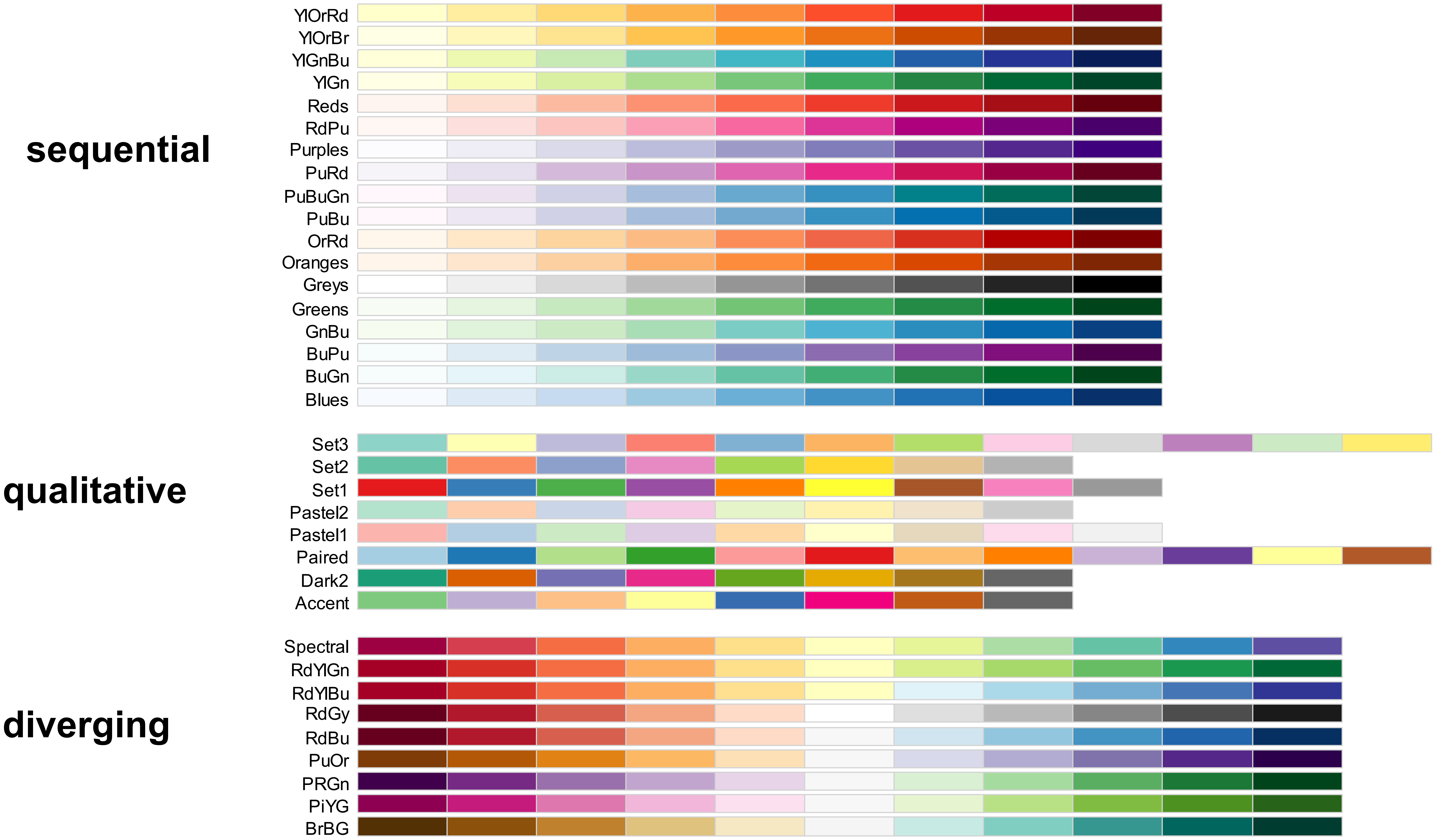
Many people are red-green colour blind

Lighter colours tend to make areas look larger than darker colours → use colors of equal luminance for filled areas.

```
display.brewer.
```

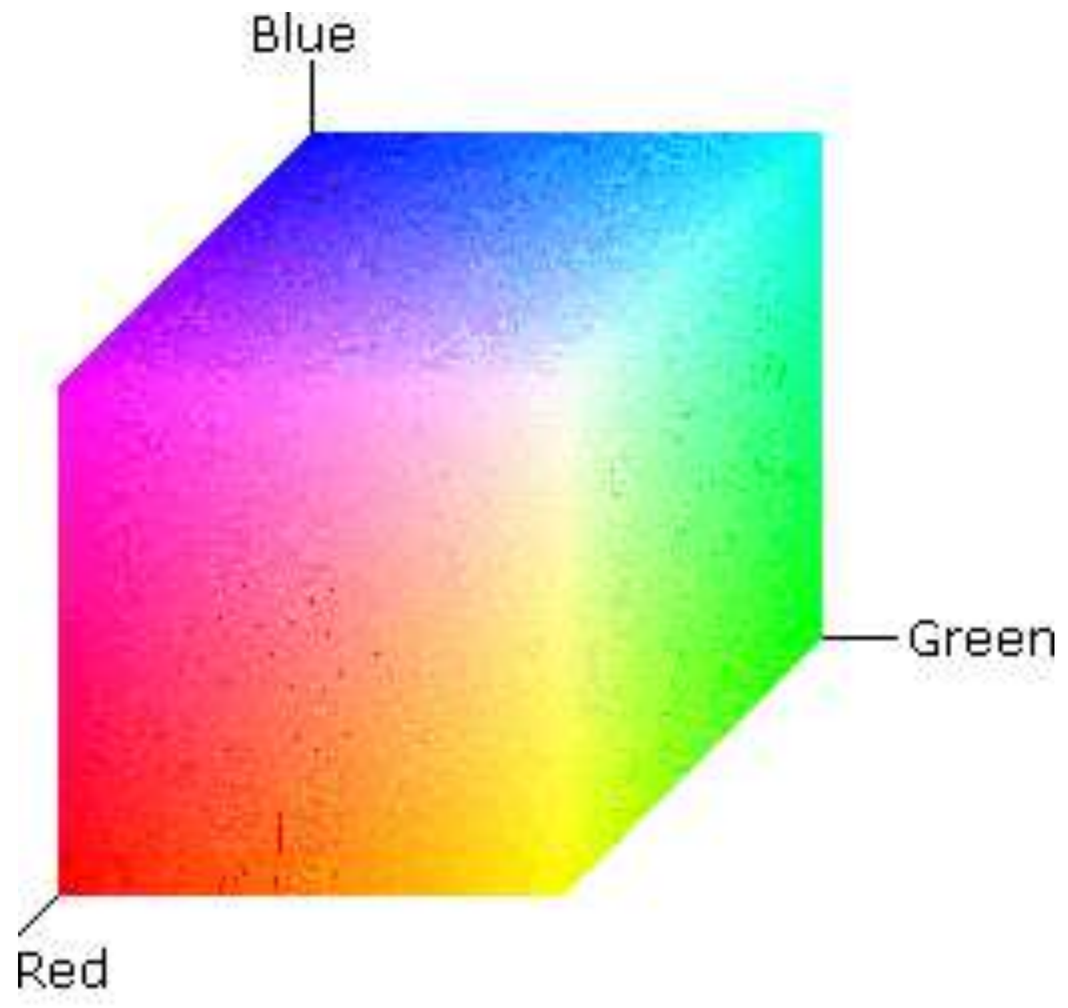


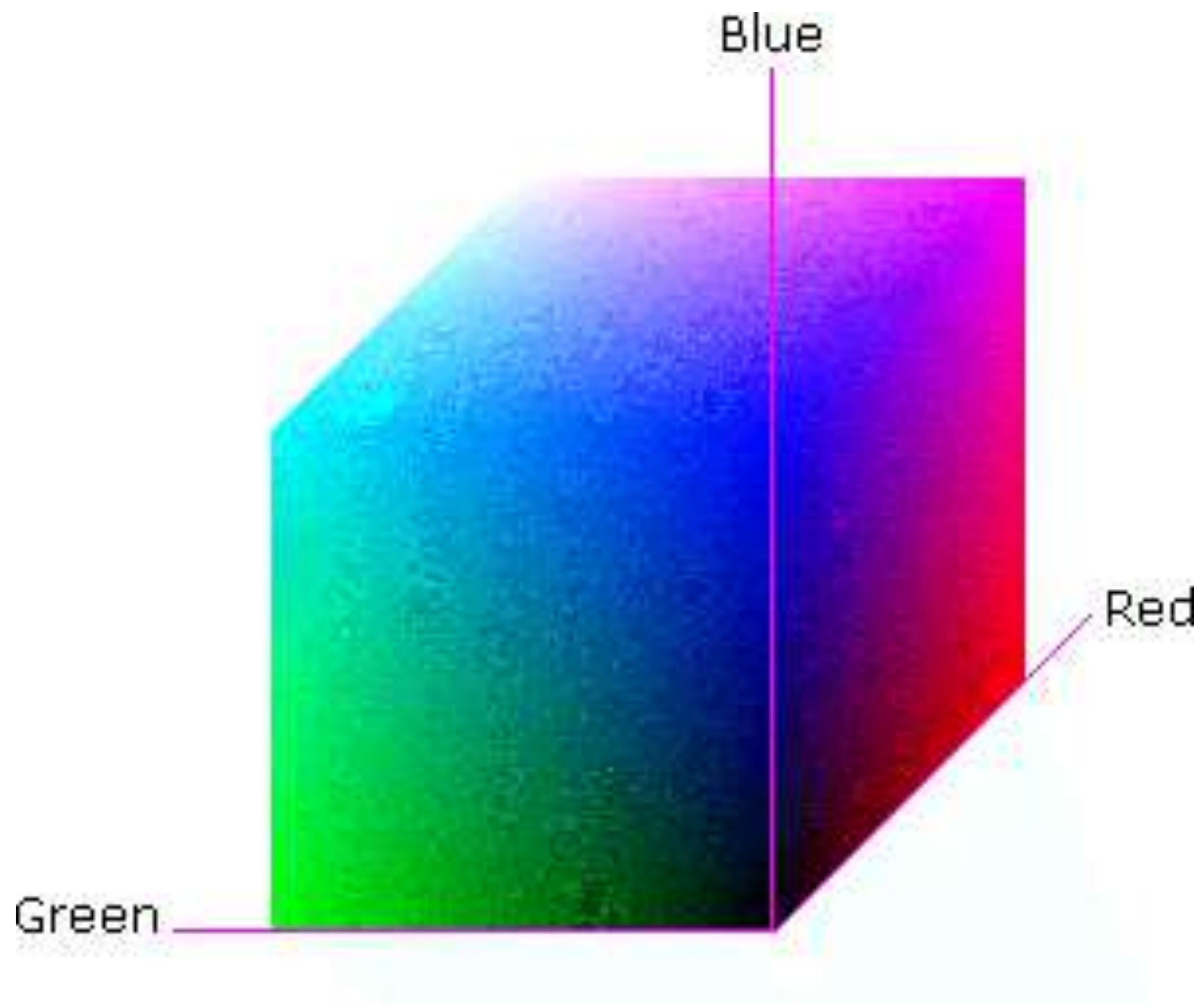
RColorBrewer



RGB color space

Motivated by computer screen hardware





HSV color space

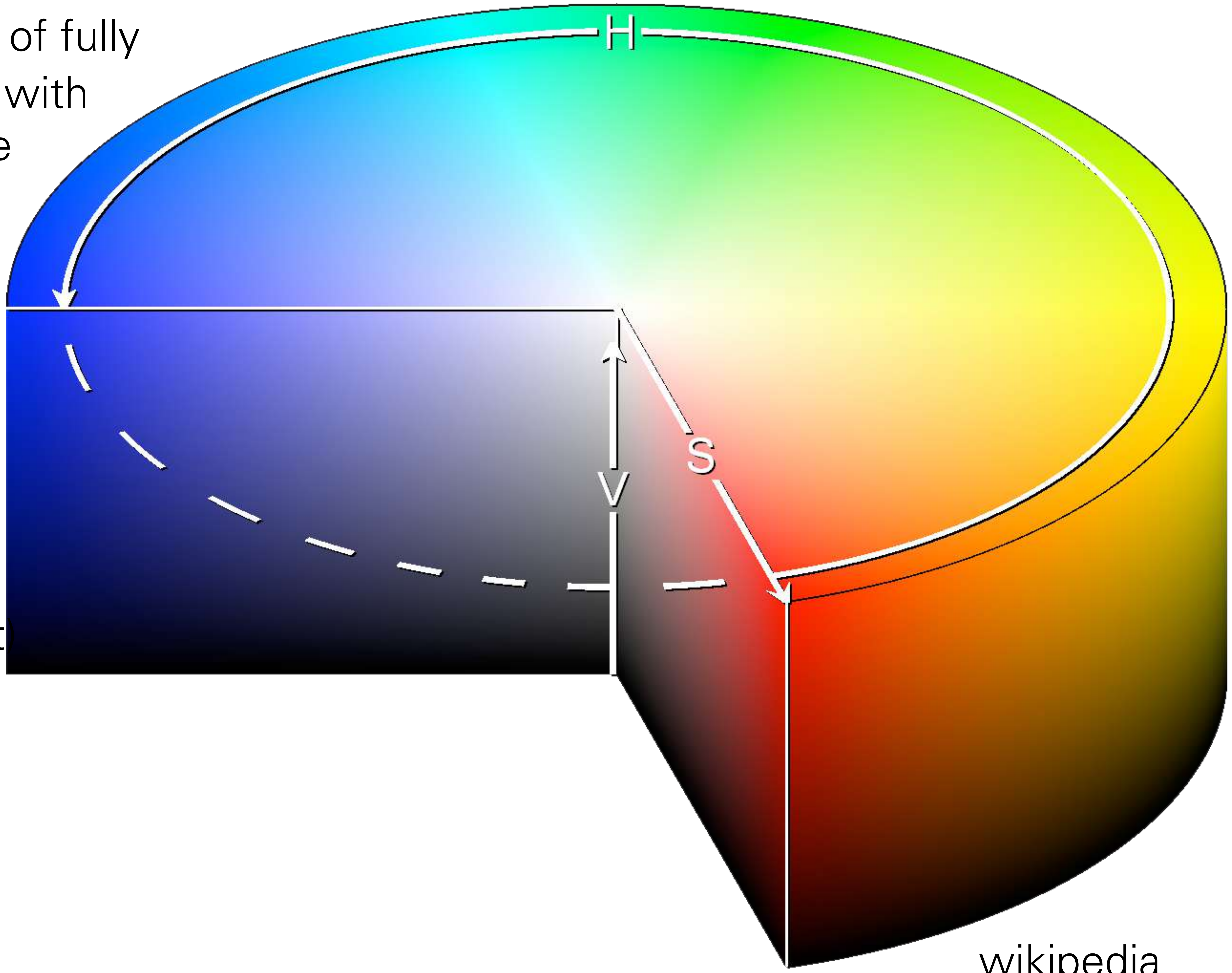
Hue-Saturation-Value (Smith 1978)

V_{max}: a planar area of fully saturated colours, with white in the centre

hue: similarity to a primary color

saturation: width of the spectral distribution

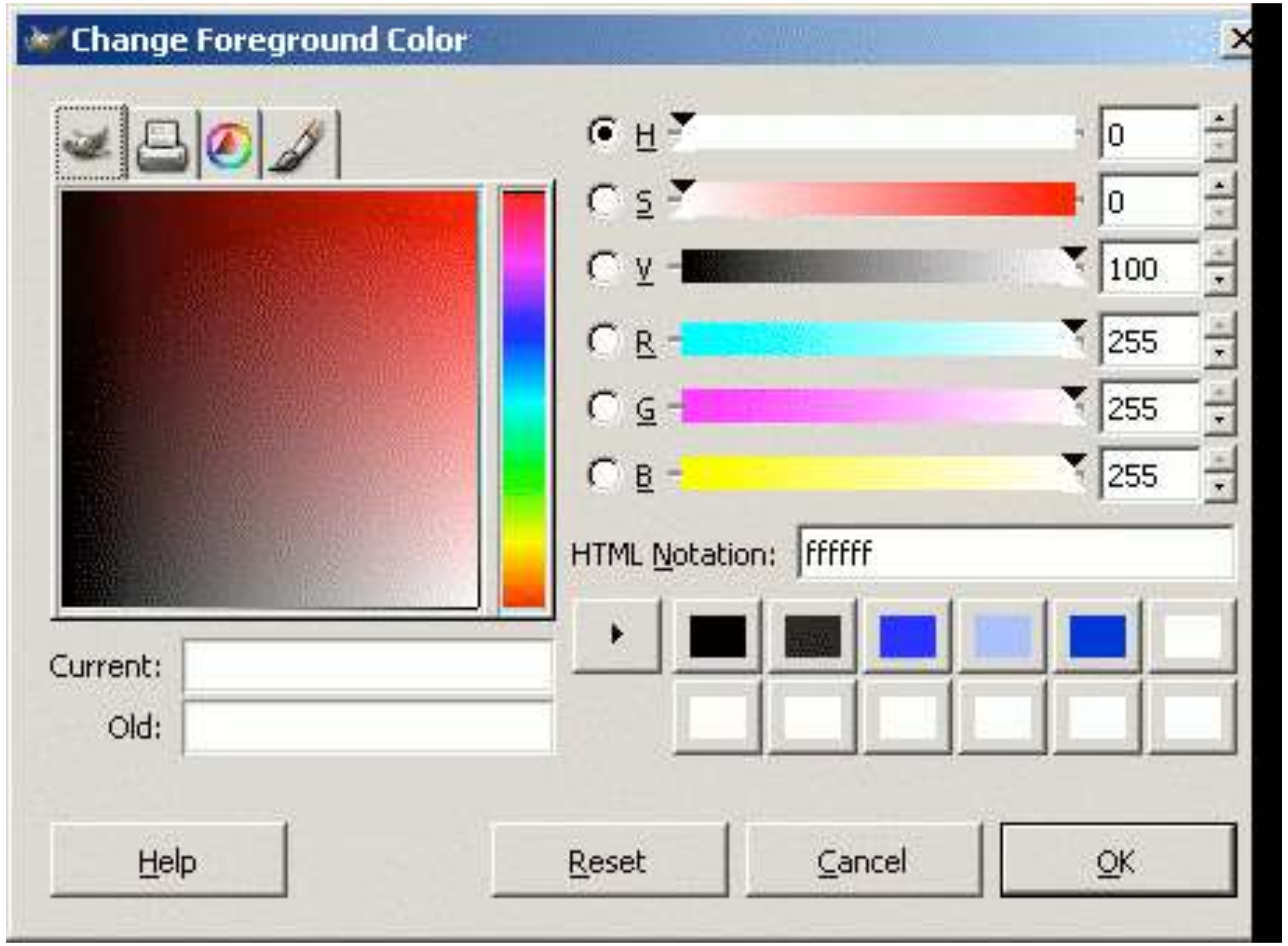
V_{min}: black (one point)



wikipedia

HSV color space

GIMP colour selector



Change Foreground Color

● H 0

○ S 0

○ V 100

○ R 255

○ G 255

○ B 255

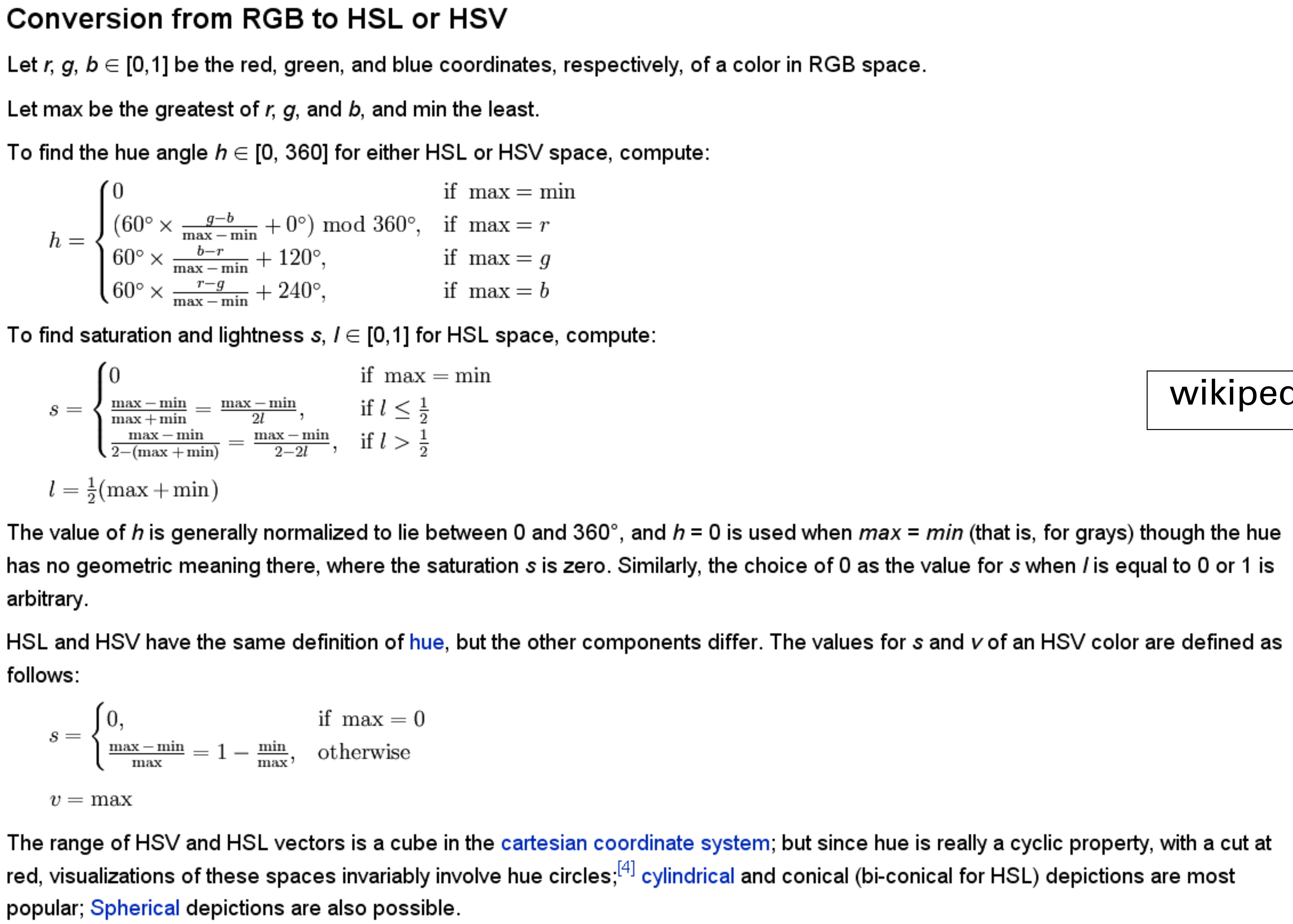
HTML Notation: ffffff

linear or circular hue chooser

and

a two-dimensional area (usually a square or a triangle) to choose saturation and value/ lightness for the selected hue

(almost) 1:1 mapping between RGB and HSV space



Conversion from RGB to HSL or HSV

Let *r*, *g*, *b* ∈ [0,1] be the red, green, and blue coordinates, respectively, of a color in RGB space.

Let max be the greatest of *r*, *g*, and *b*, and min the least.

To find the hue angle *h* ∈ [0, 360] for either HSL or HSV space, compute:

$$h = \begin{cases} 0 & \text{if } \max = \min \\ (60^\circ \times \frac{g-b}{\max - \min} + 0^\circ) \bmod 360^\circ, & \text{if } \max = r \\ 60^\circ \times \frac{b-r}{\max - \min} + 120^\circ, & \text{if } \max = g \\ 60^\circ \times \frac{r-g}{\max - \min} + 240^\circ, & \text{if } \max = b \end{cases}$$

To find saturation and lightness *s*, *l* ∈ [0,1] for HSL space, compute:

$$s = \begin{cases} 0 & \text{if } \max = \min \\ \frac{\max - \min}{\max + \min} = \frac{\max - \min}{2l}, & \text{if } l \leq \frac{1}{2} \\ \frac{\max - \min}{2-(\max + \min)} = \frac{\max - \min}{2-2l}, & \text{if } l > \frac{1}{2} \end{cases}$$

$$l = \tfrac{1}{2}(\max + \min)$$

The value of *h* is generally normalized to lie between 0 and 360°, and *h* = 0 is used when *max* = *min* (that is, for grays) though the hue has no geometric meaning there, where the saturation *s* is zero. Similarly, the choice of 0 as the value for *s* when *l* is equal to 0 or 1 is arbitrary.

HSL and HSV have the same definition of hue, but the other components differ. The values for *s* and *v* of an HSV color are defined as follows:

$$s = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases}$$

$$v = \max$$

The range of HSV and HSL vectors is a cube in the cartesian coordinate system; but since hue is really a cyclic property, with a cut at red, visualizations of these spaces invariably involve hue circles;[4] cylindrical and conical (bi-conical for HSL) depictions are most popular; Spherical depictions are also possible.

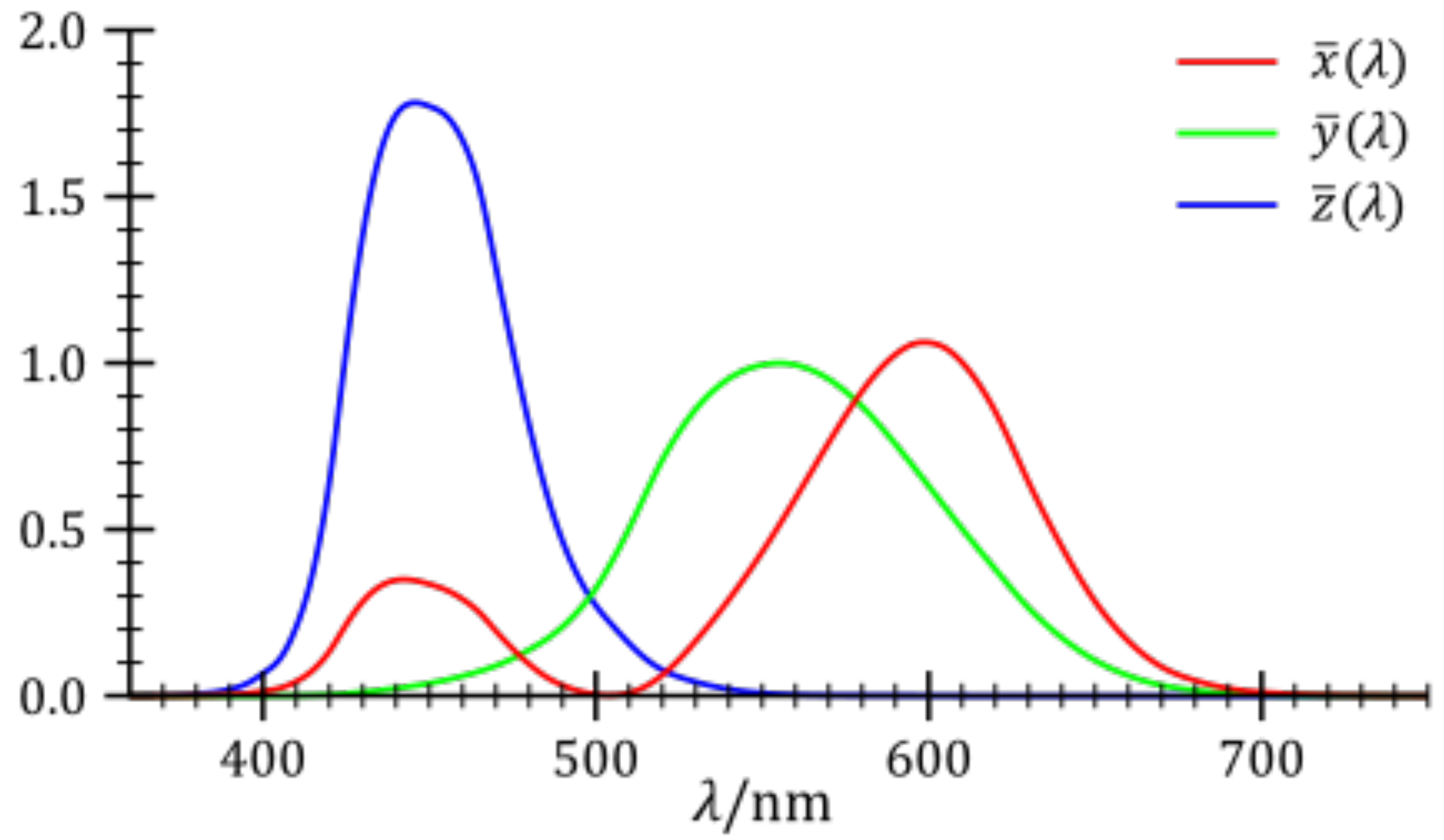
wikipedia

Perceptual colour spaces

Human perception of colour corresponds neither to RGB nor HSV coordinates, and neither to the physiological axes light-dark, yellow-blue, red-green

Perceptually based coordinates of colour space: CIELUV, CIELAB

Commission Internationale de l' Éclairage (CIE) in 1931, on the basis of extensive colour matching experiments with people, defined a "standard observer" who represents a typical human colour response (response of the three light cones + their processing in the brain) to a triplet (x,y,z) of primary light sources



https://en.wikipedia.org/wiki/CIE_1931_color_space

1976: CIELUV (L, u, v)-coordinates are preferred by those who work with emissive colour technologies (e.g. computer displays); CIELAB by those working with dyes and pigments (such as in the printing and textile industries)

Ihaka 2003

HCL colours

$(u,v) = C * (\cos H, \sin H)$

H: hue (dominant wavelength)

C: chroma (colorfulness, intensity of color as compared to gray)

L: luminance (brightness, amount of gray), same as in CIELUV

(C, H) are simply polar coordinates for (u,v)

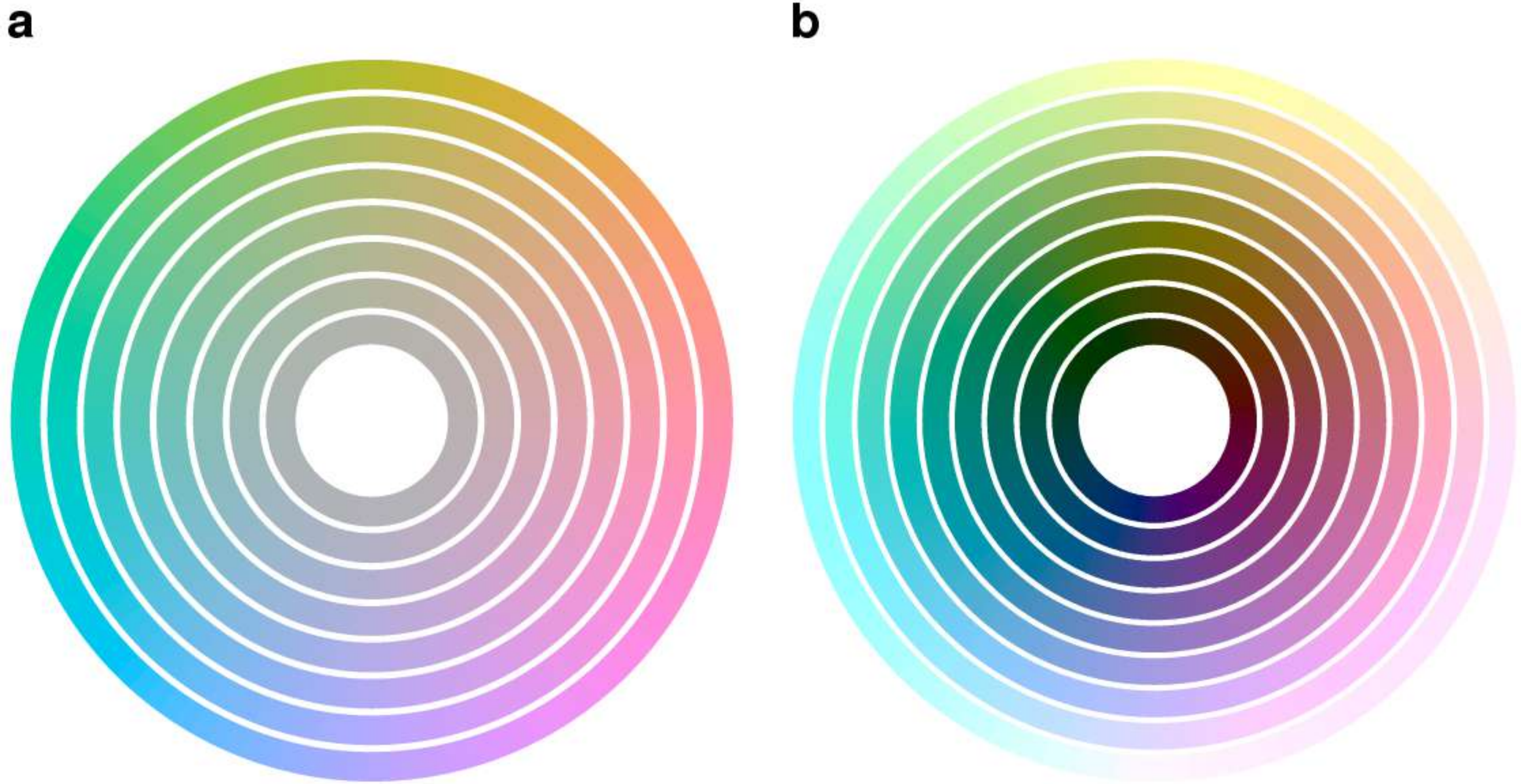
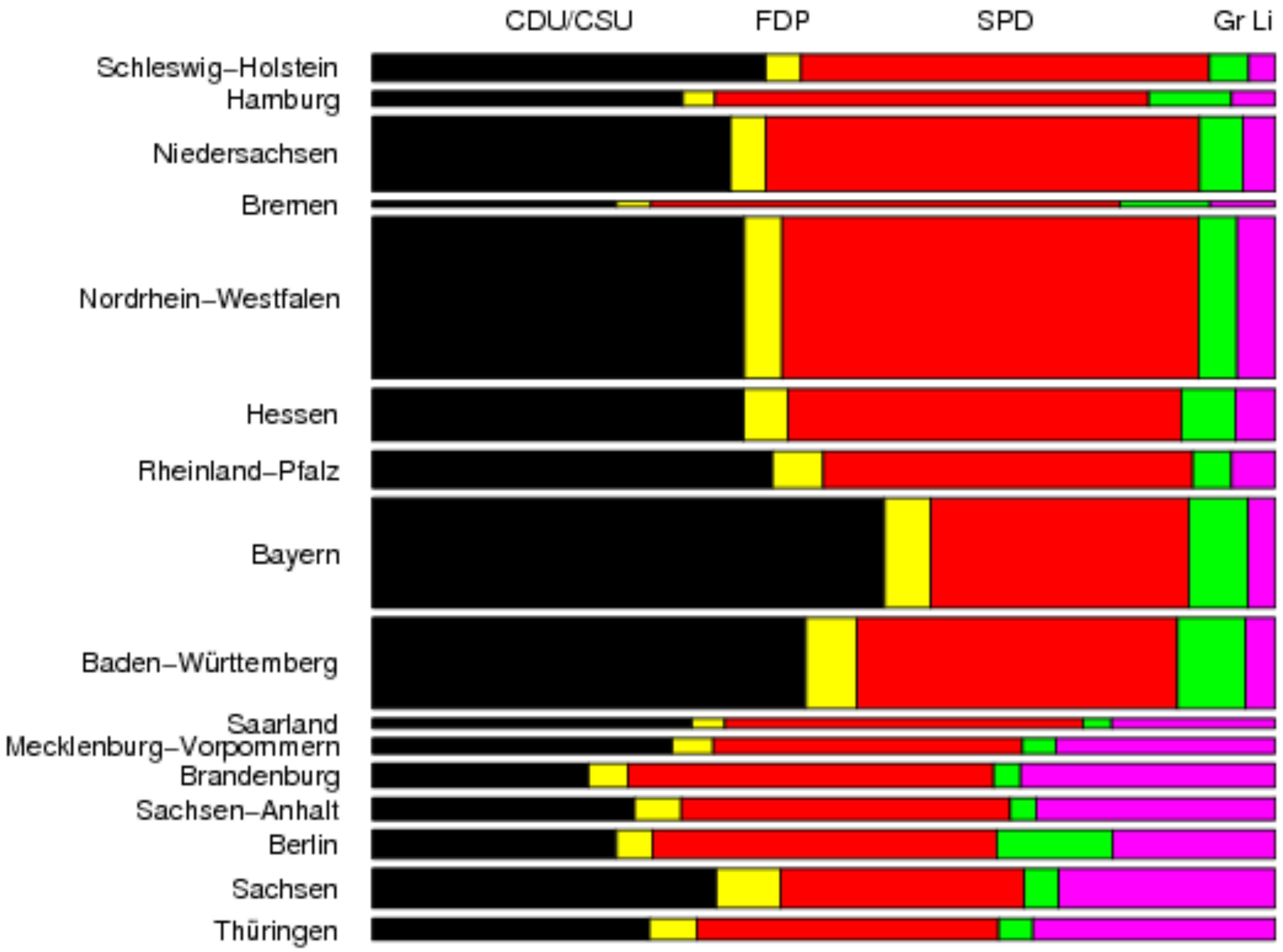
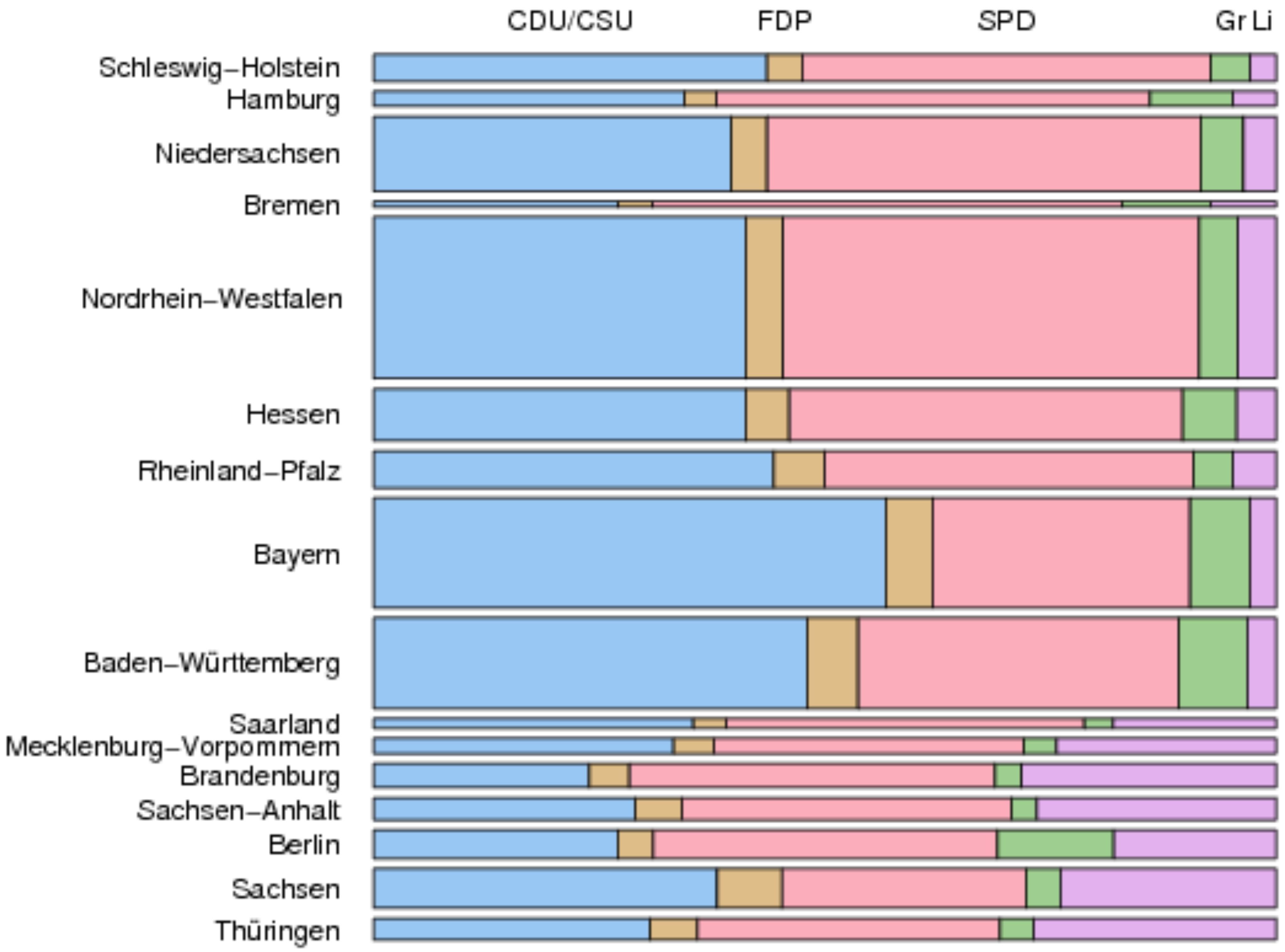


Figure 2: Circles in HCL colorspace. a: circles in HCL space at constant $L = 75$, with the angular coordinate H varying from 0 to 360 and the radial coordinate $C = 0, 10, \ldots, 60$. b: constant $C = 50$, and $L = 10, 20, \ldots, 90$.

Pick your favourite





From A. Zeileis, Reisensburg 2007

Balance

The intensity of colour that should be used is dependent on the area that that colour is to occupy. Small areas need to be more colourful than larger ones.

Choose colours centred on a mid-range or neutral value, or;

Choose colours at equally spaced points along smooth paths through (perceptually uniform) colour space: equal luminance and chroma and correspond to set of evenly spaced hues.

Acknowledgements

Susan Holmes
Robert Gentleman
Florian Hahne

Hadley Wickham
Ross Ihaka
Achim Zeileis
Kurt Hornik