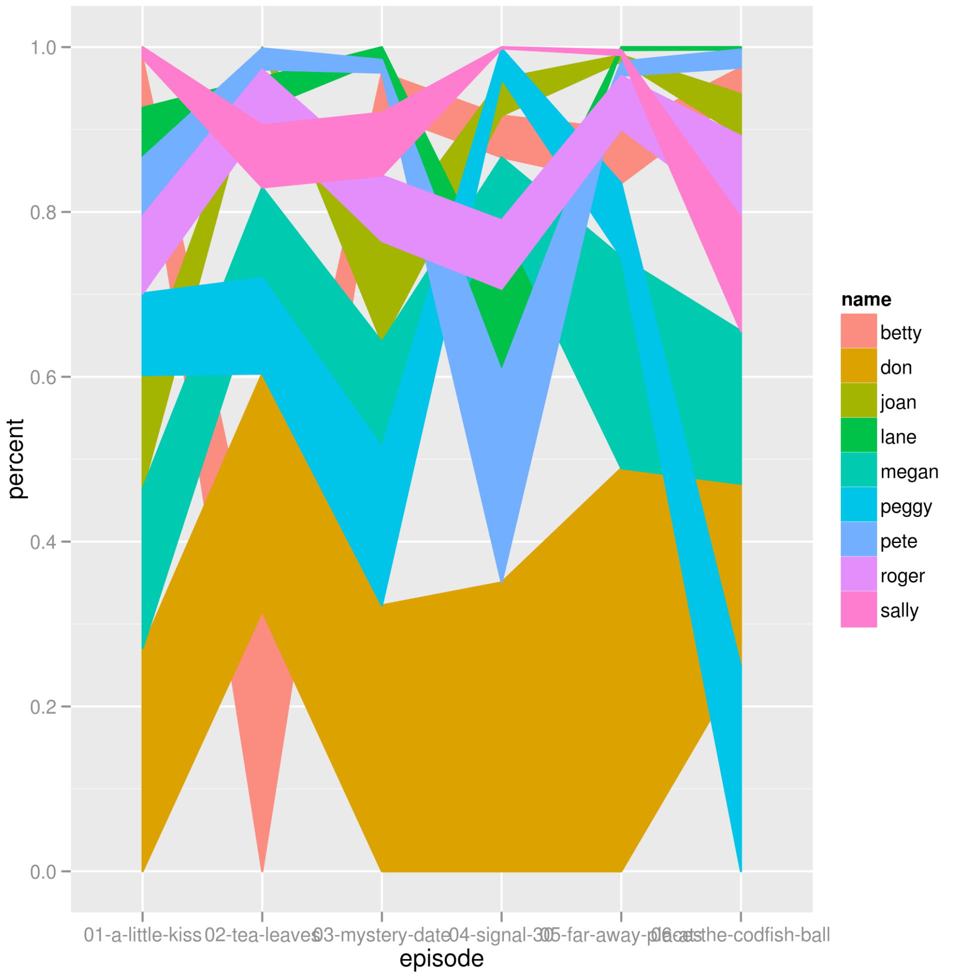
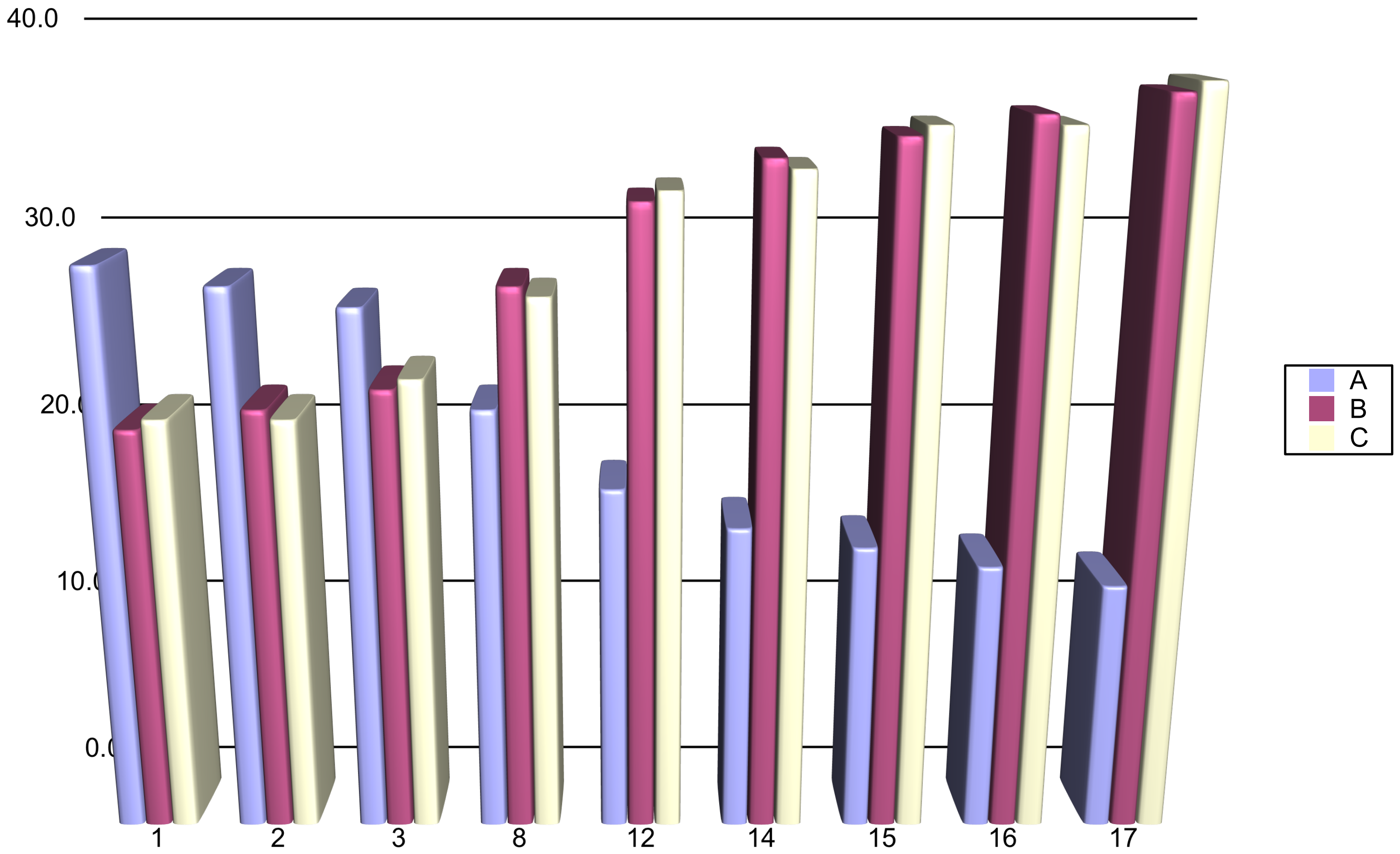
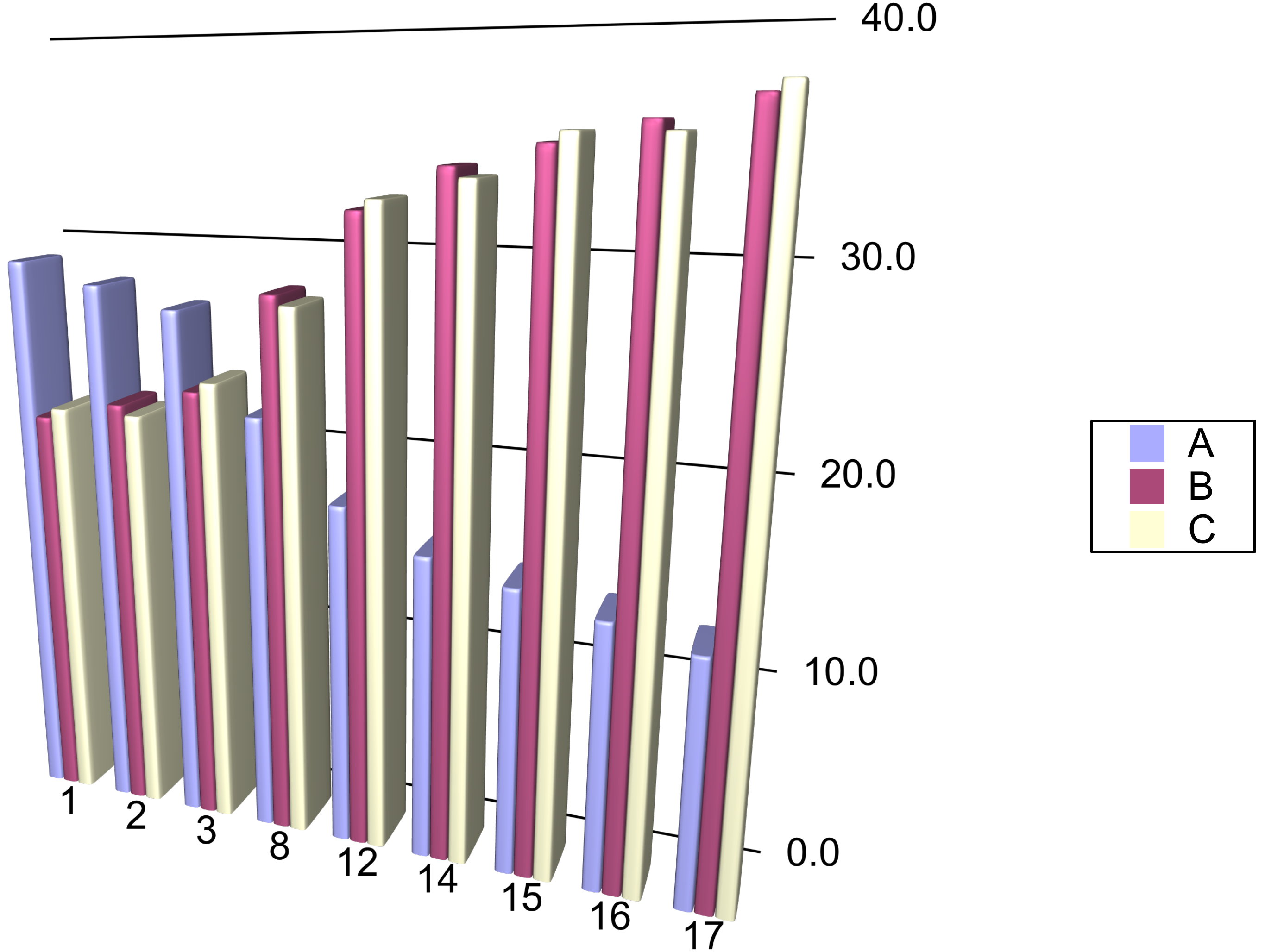


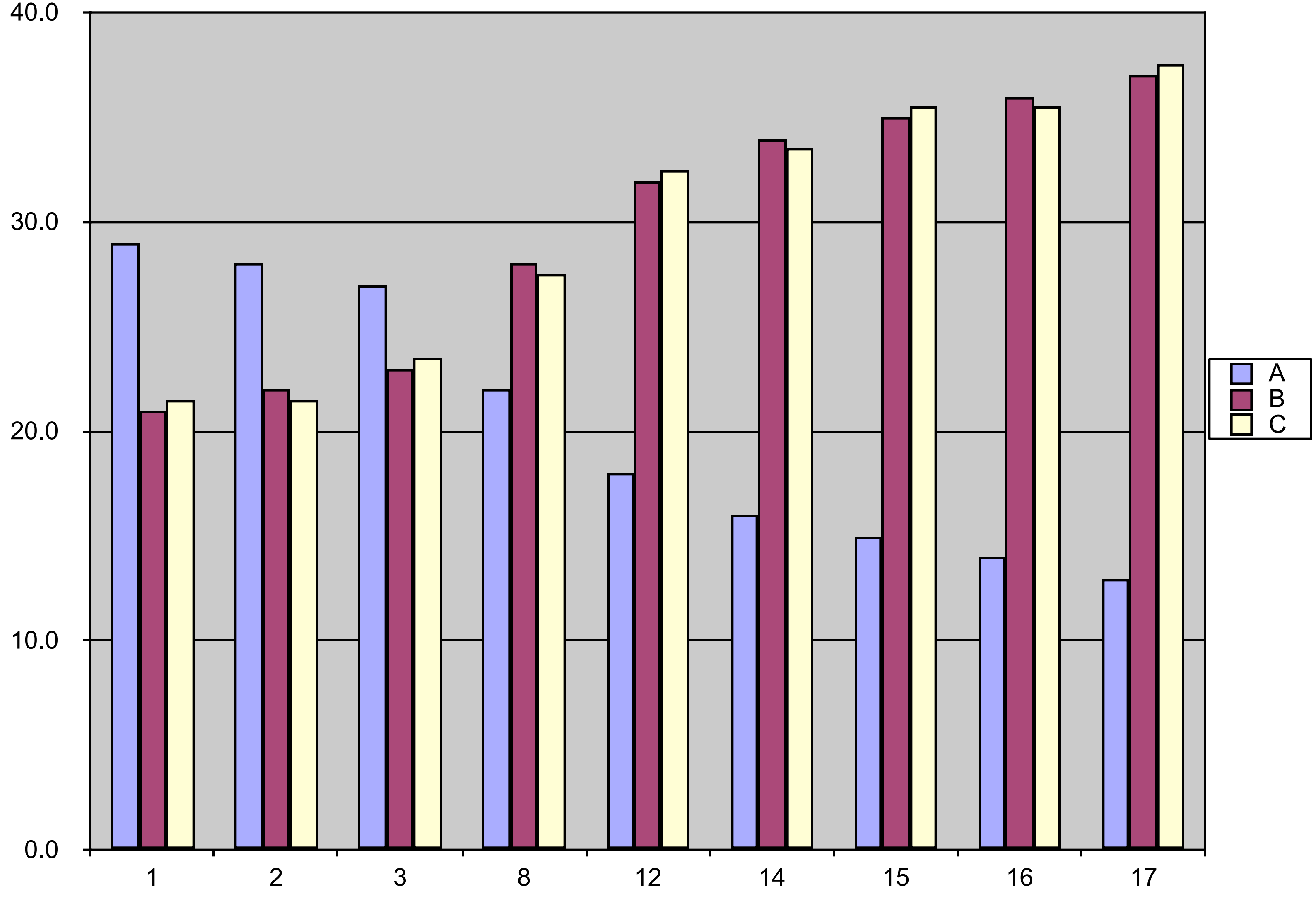
Graphics

Wolfgang Huber

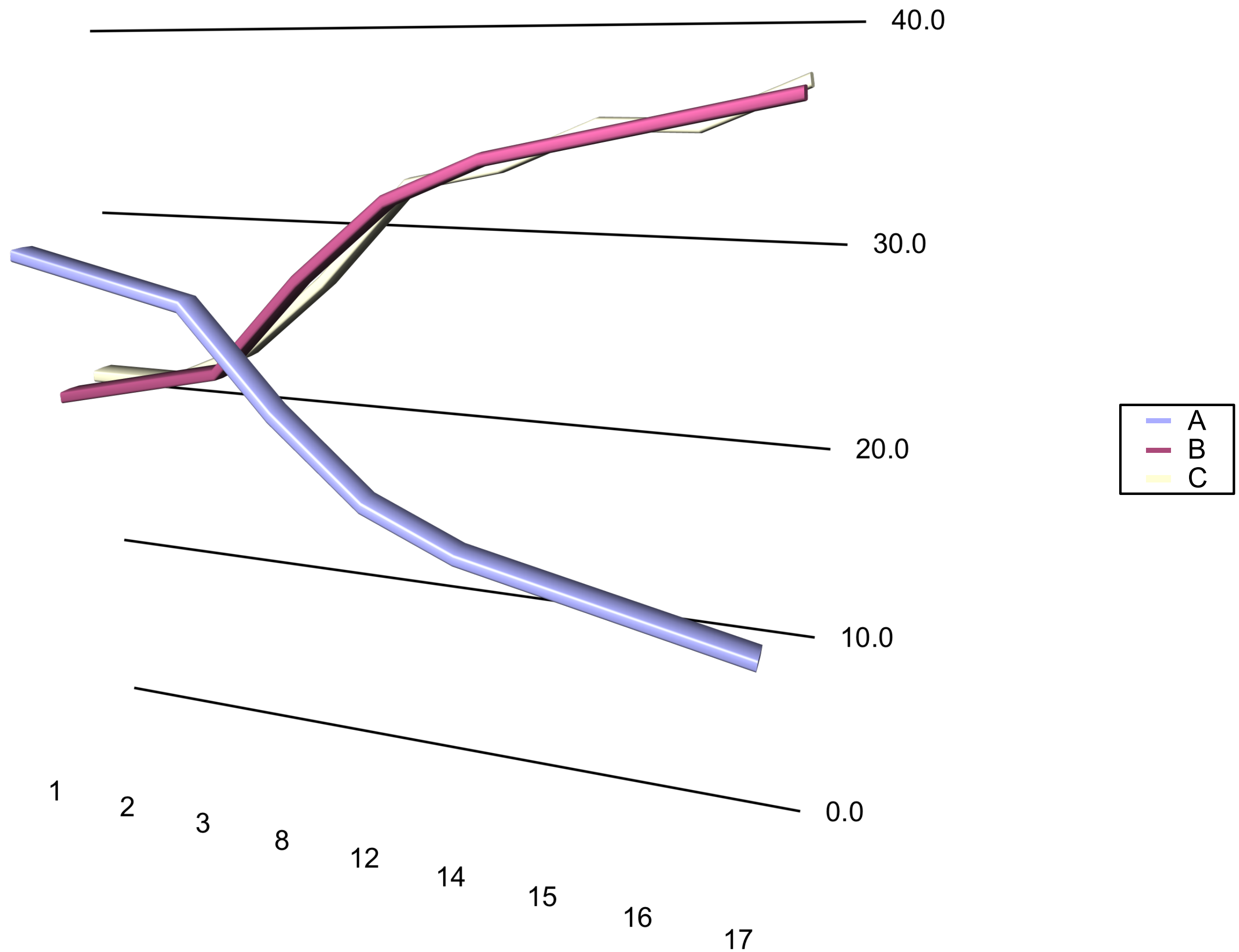
Horror Picture Show


40.0
30.0
20.0
10.0
0.0
1
2
3
8
12
14
15
16
17
A
B
C

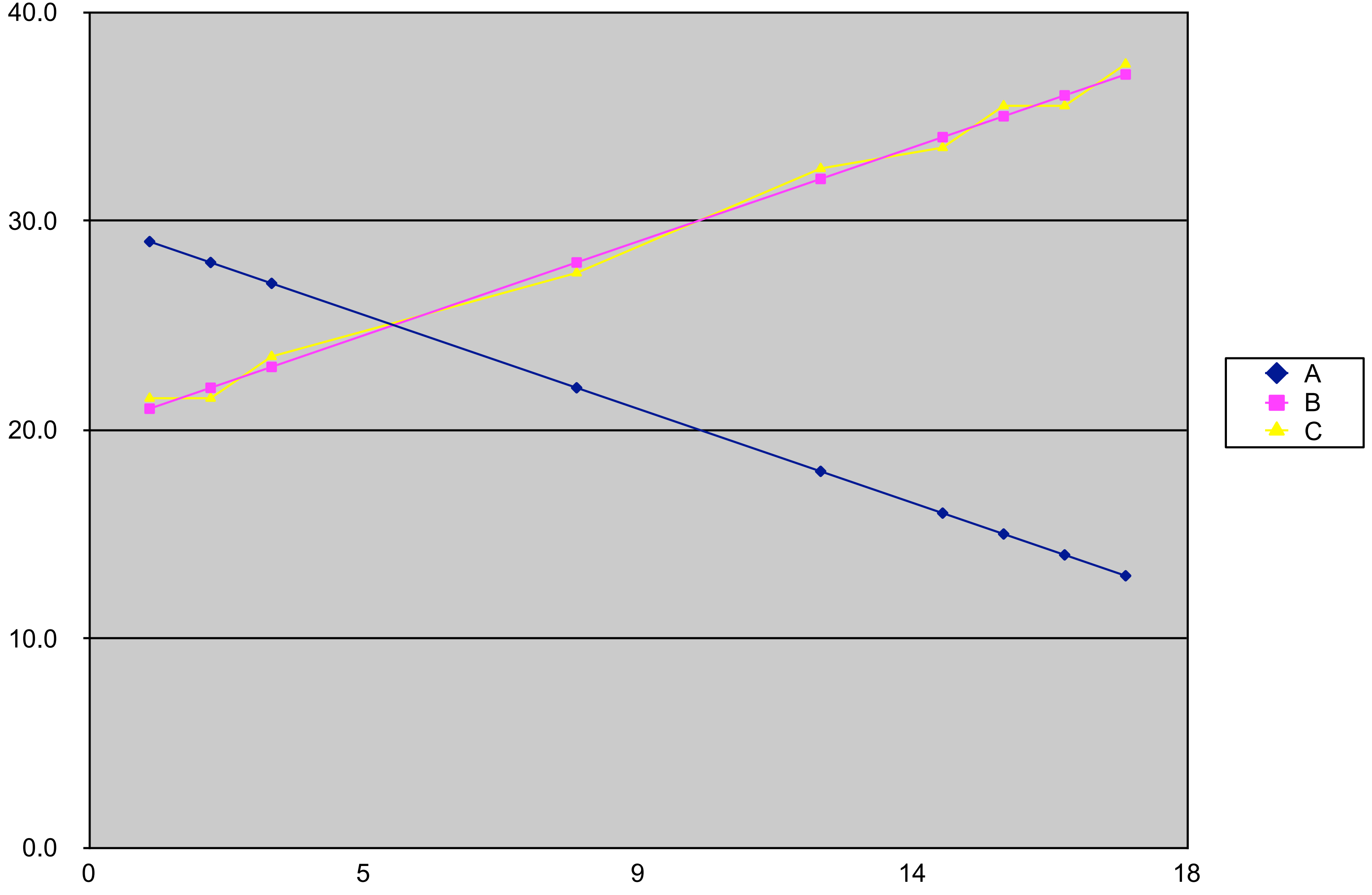

40.0
30.0
20.0
10.0
0.0
1
2
3
8
12
14
15
16
17
A
B
C



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
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

Why graphics?

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

Goals of this lecture

- Review the basics of *base* R plotting
- Understand the logic behind the *grammar of graphics* concept
- Introduce *ggplot2*'s `qplot` function
- Show how to build complex plots from the ground up using *ggplot2*'s `ggplot` function
- See how to plot 1D, 2D, 3-5D data, and understand faceting
- Become good at rapidly exploring data sets by visualization

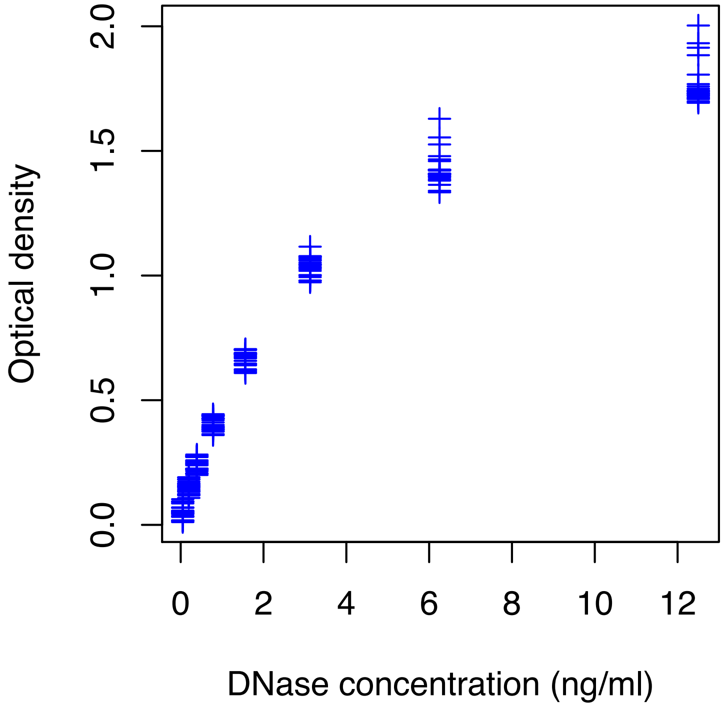
base R plotting

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")
```

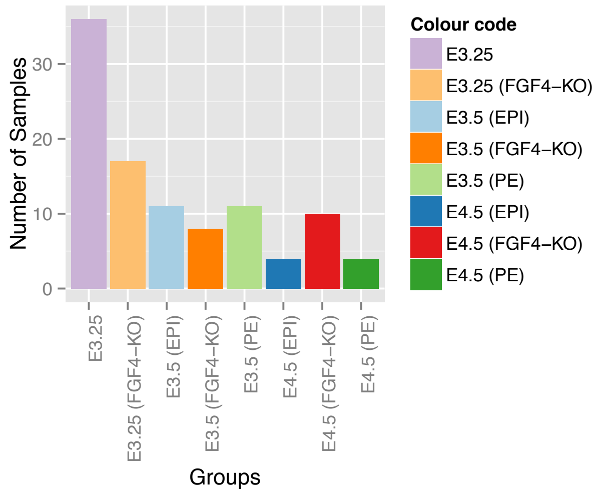


The grammar of graphics

The components of *ggplot2*'s grammar of graphics are

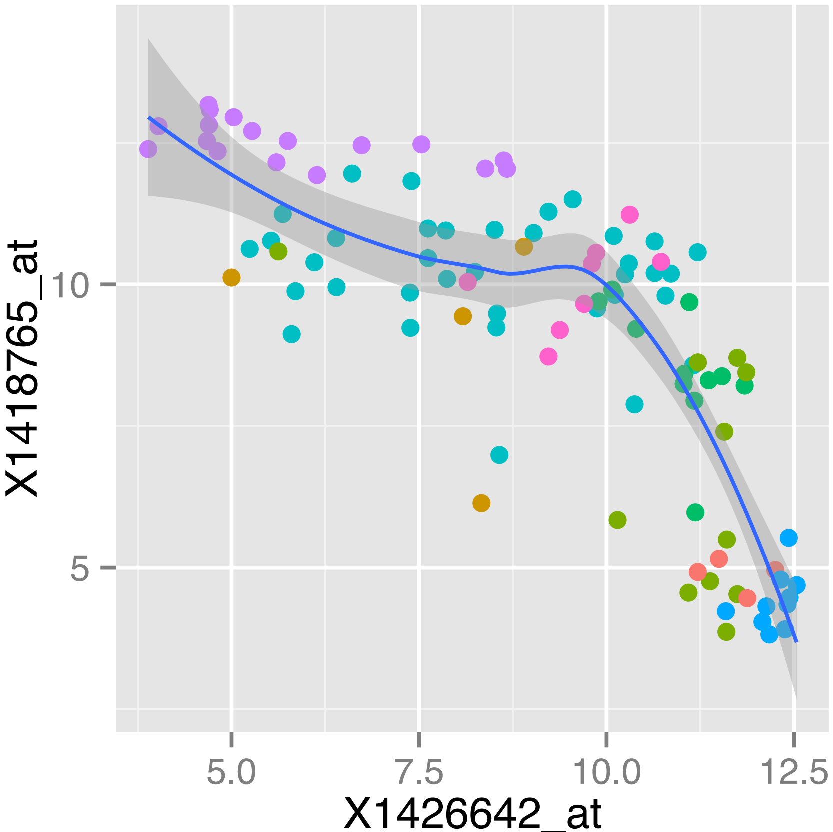
1. a dataset
2. a choice of geometric object that serves as the visual representations of the data – for instance, points, lines, rectangles, contours
3. a description of how the variables in the data are mapped to visual properties (aesthetics) of the geometric objects, and an associated scale, (e. g., linear, logarithmic, rank)
4. a statistical summarisation rule
5. a coordinate system
6. a facet specification, i. e. the use of several plots to look at the same data

```
qplot(x = names(groupSize),
      y = as.vector(groupSize),
      geom = "bar", stat = "identity",
      xlab = "Groups", ylab = "Number of Samples",
      fill = names(groupSize)) +
  scale_fill_manual(values = groupColour, name="Colour code")
```



Layers

```
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 )
```



geom and summary often imply each other (by default)

```
dfx <- as.data.frame(exprs(x))
p1 <- ggplot(dfx, aes(x = ‘20 E3.25‘)) +
            geom_histogram(binwidth = 0.2)
p2 <- ggplot(dfx, aes(x = ‘20 E3.25‘)) +
            geom_bar(stat = "bin", binwidth = 0.2)
```

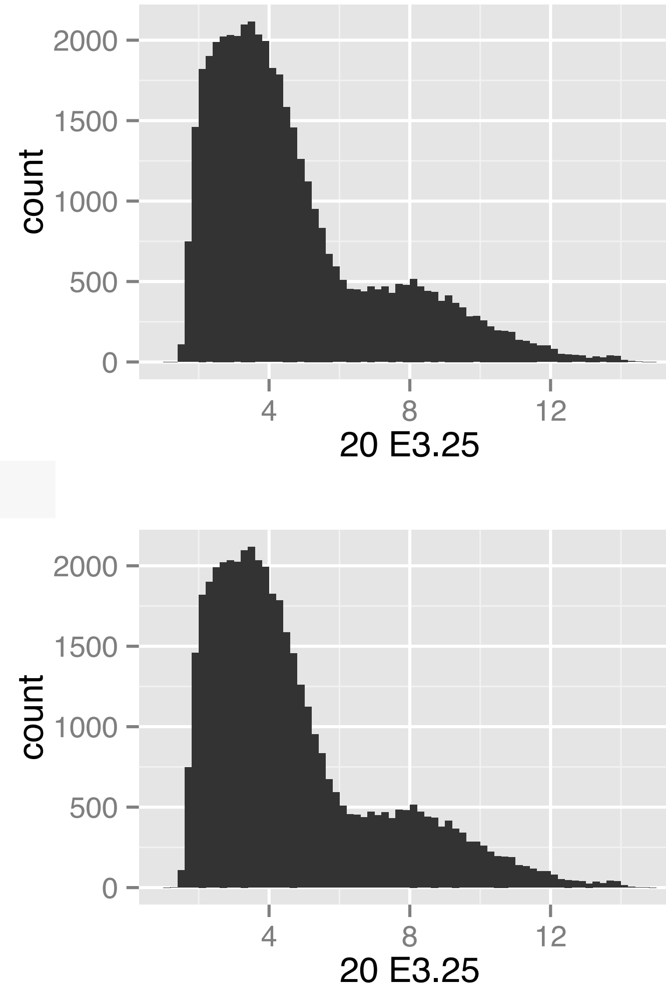


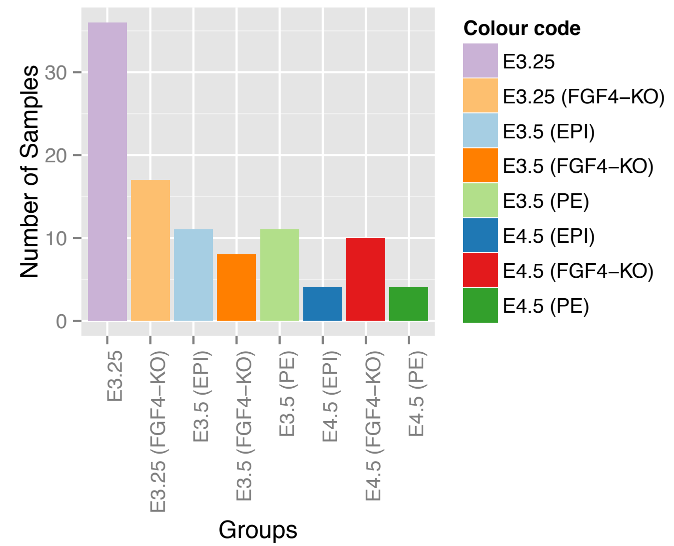
Figure 4.11: Two different ways of creating the same histogram using the grammar of graphics.

A more complex example: themes

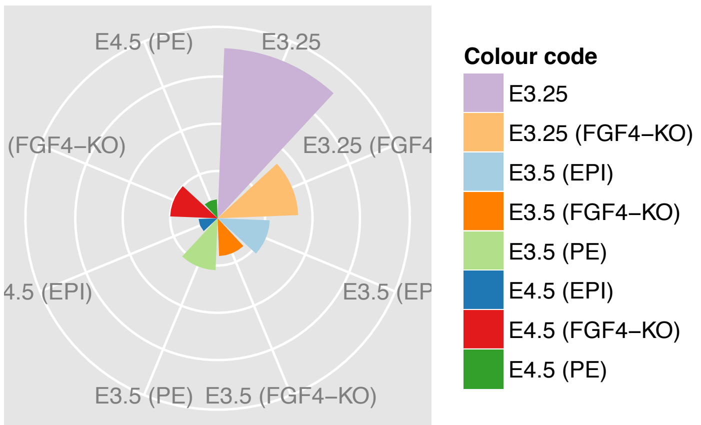
```
pb <- ggplot(data.frame(
                  name = names(groupSize),
                  size = as.vector(groupSize)),
               aes(x = name, y = size))
```

No geom defined yet!

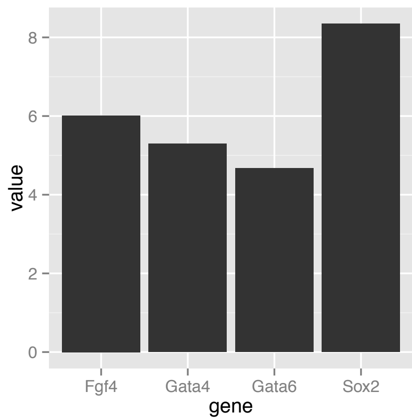
```
pb <- pb + geom_bar(stat = "identity") +
   aes(fill = name) +
   scale_fill_manual(values = groupColour, name = "Colour code") +
   theme(axis.text.x = element_text(angle = 90, hjust = 1))  +
   xlab("Groups") + ylab("Number of Samples")
```

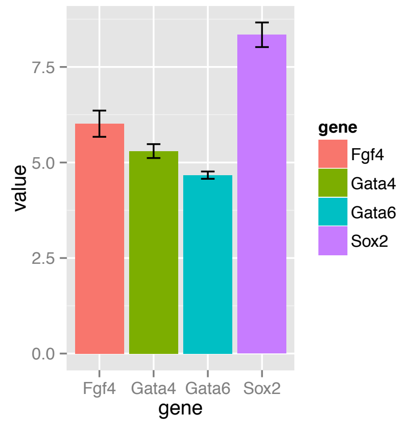


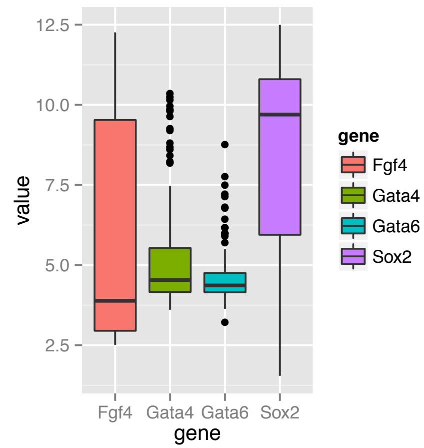

```
pb.polar <- pb + coord_polar() +
  theme(axis.text.x = element_text(angle = 0, hjust = 1),
        axis.text.y = element_blank(),
        axis.ticks = element_blank()) +
  xlab("") + ylab("")
pb.polar
```

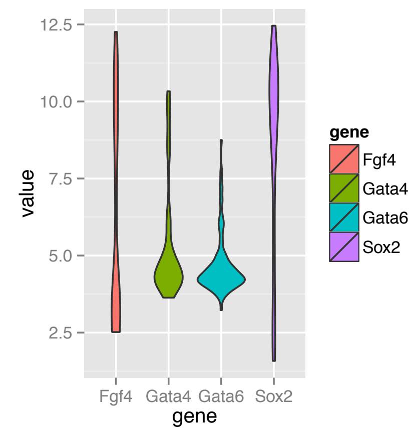


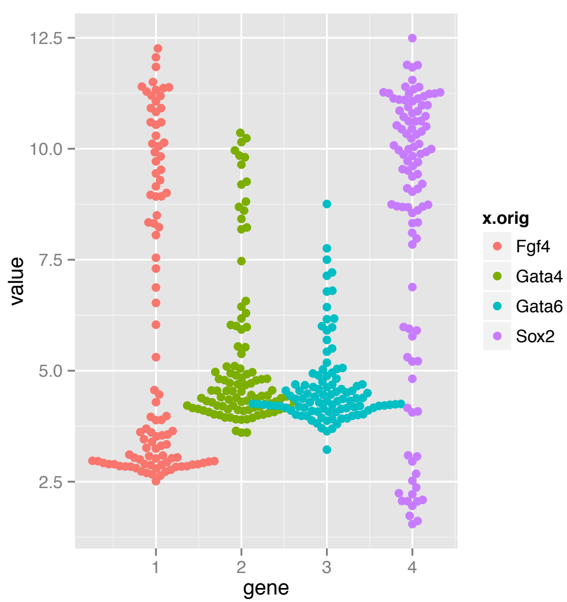
Showing 1D data

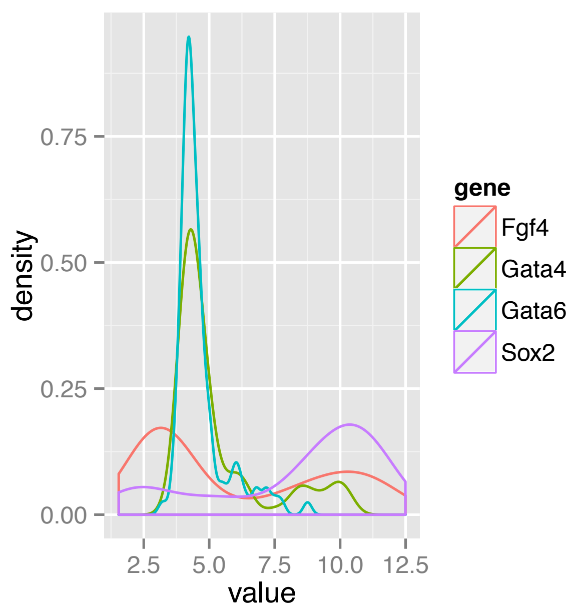


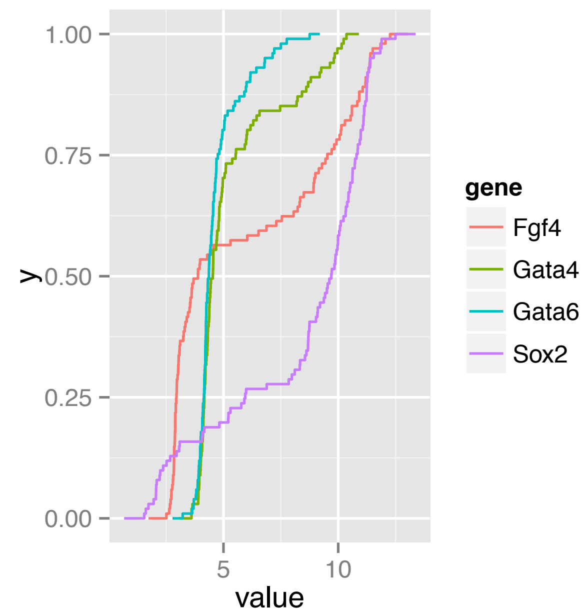












Discussion of 1D plot types

Boxplot makes sense for unimodal distributions

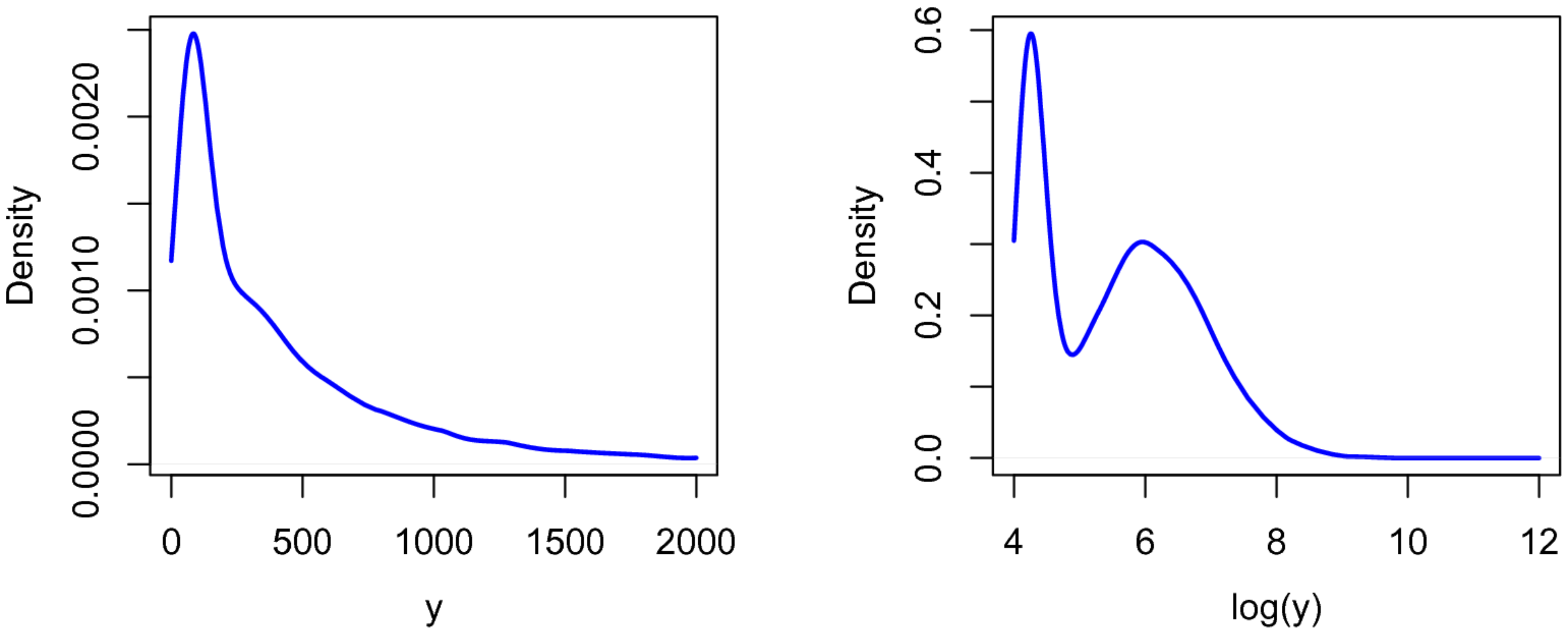
Histogram requires definition of bins (width, positions) and can create visual artifacts esp. if the number of data points is not large

Density requires the choice of bandwidth; plot tends to obscure the sample size (i.e. the uncertainty of the estimate)

ecdf does not have these problems; but is more abstract and interpretation requires some 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)

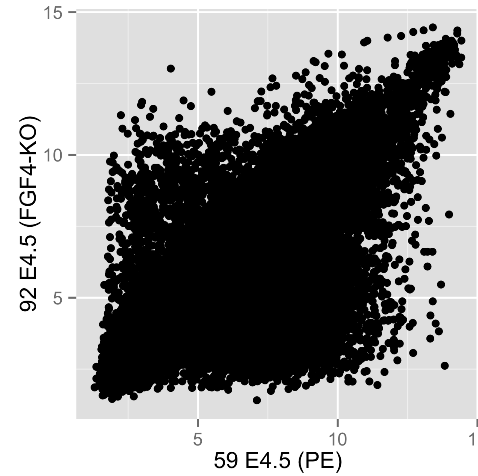
Impact of non-linear transformation on the shape of a density

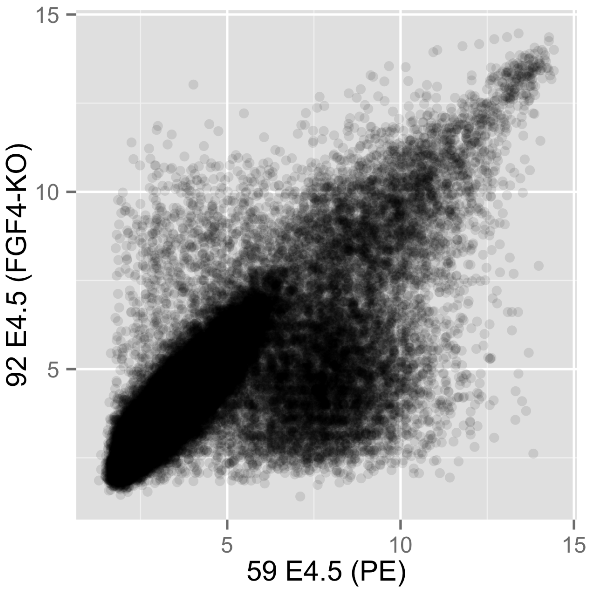


y: sample from a mixture of two log-normal distributions
kernel density estimates

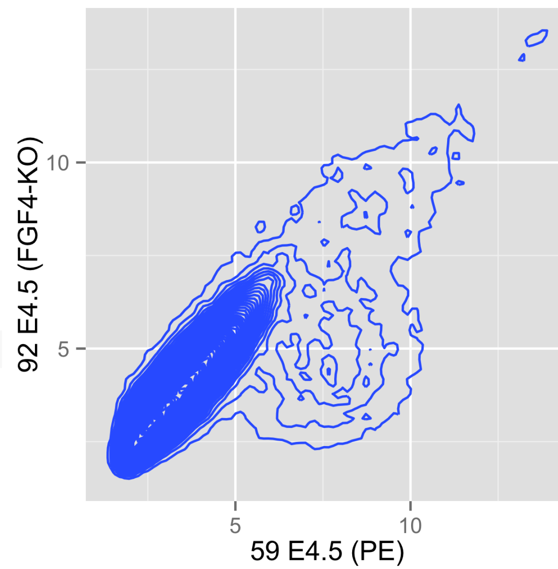
Showing 2D data

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



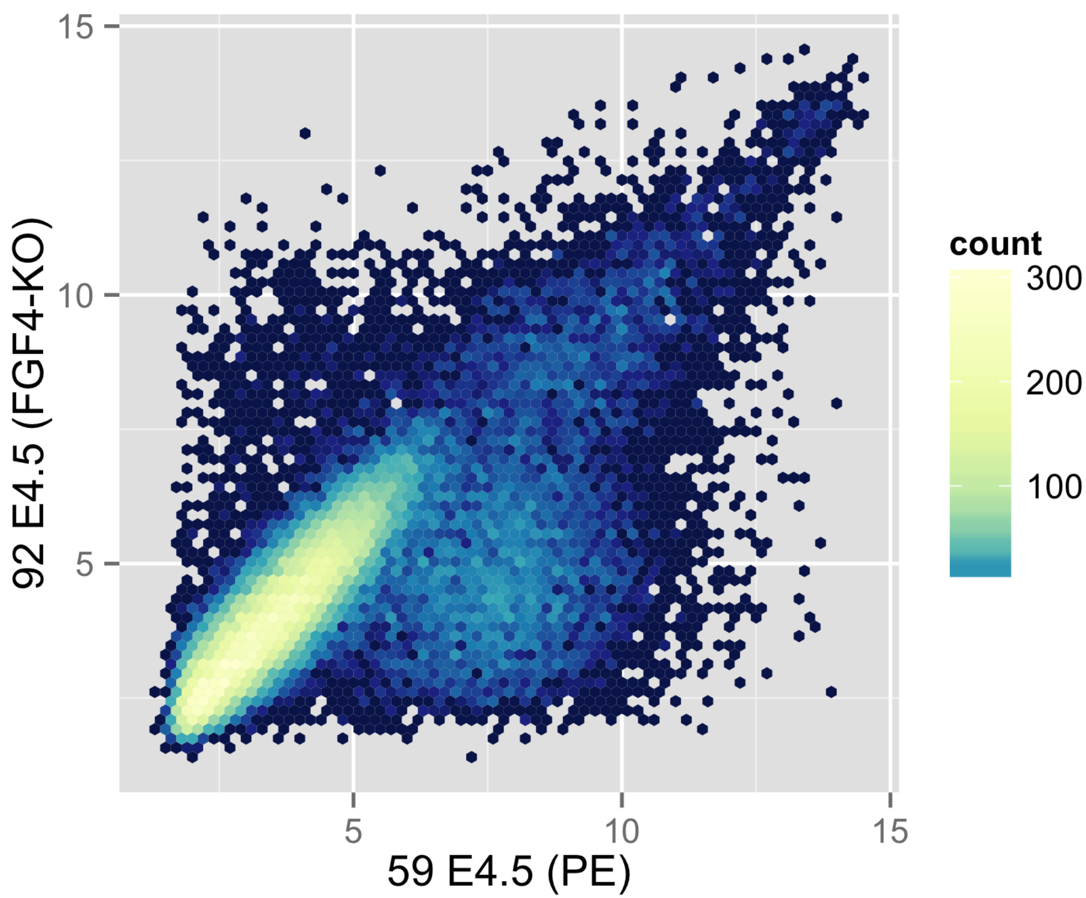



```
scp  + geom_point(alpha = 0.1)
```

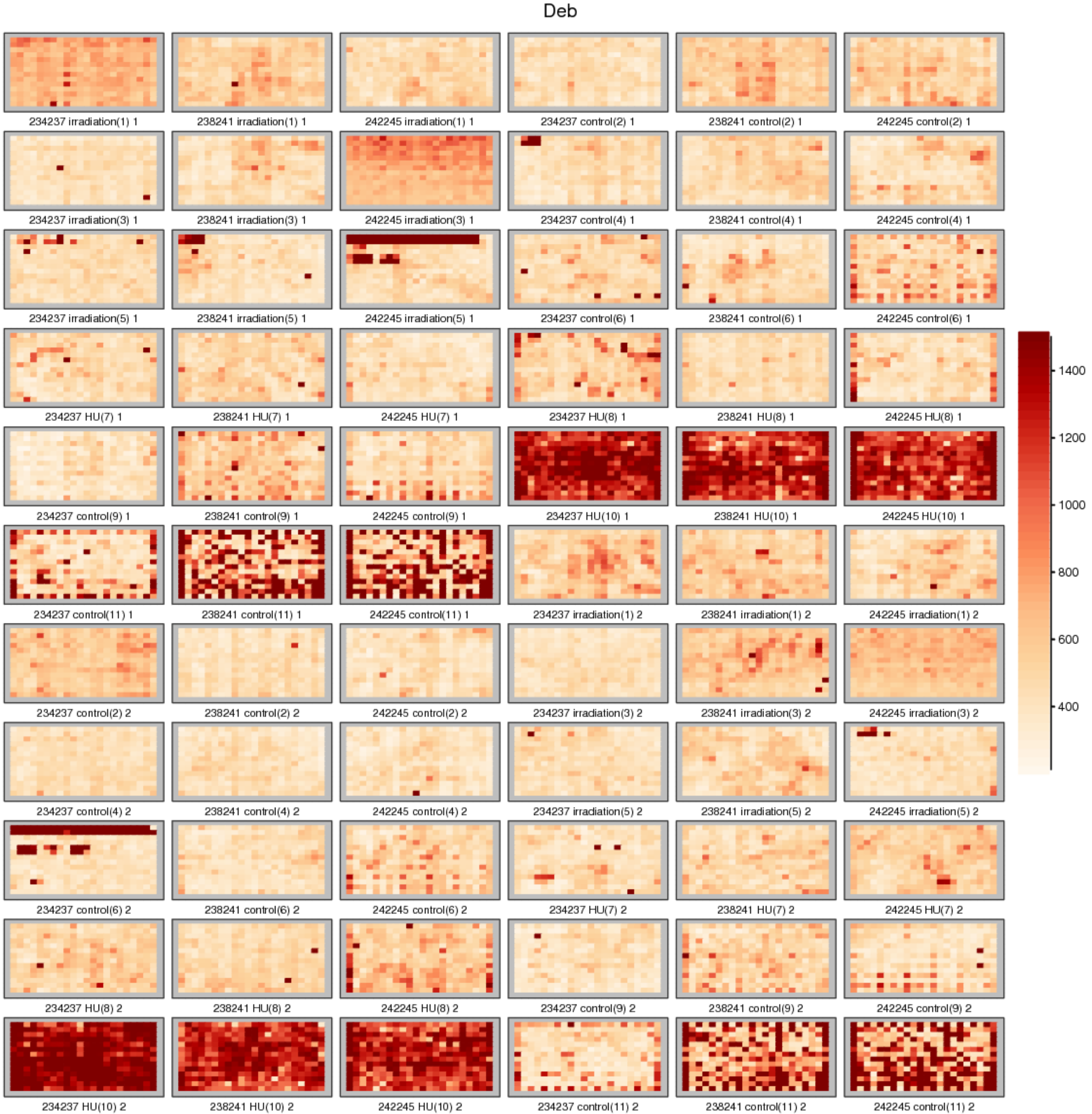



```
scp + geom_density2d(h = 0.5, bins = 60)
```

Showing 2D data




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



package
splots

Plot shape, banking

Yearly sunspot numbers 1849-1924

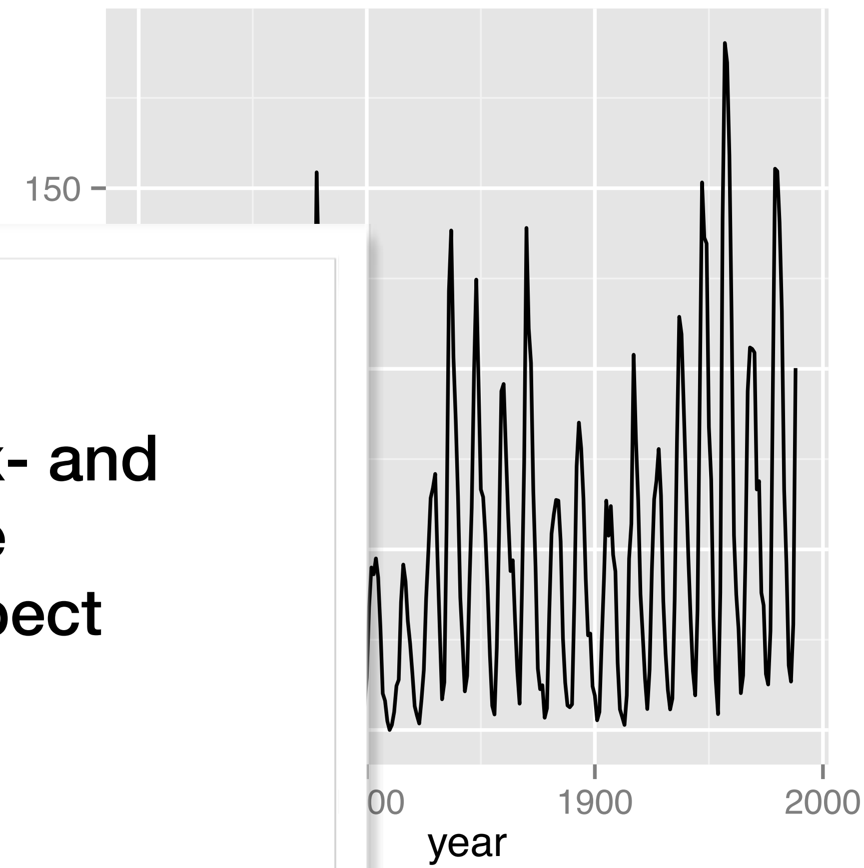
Changes in amplitude

Bankin

Choose
center
slopes

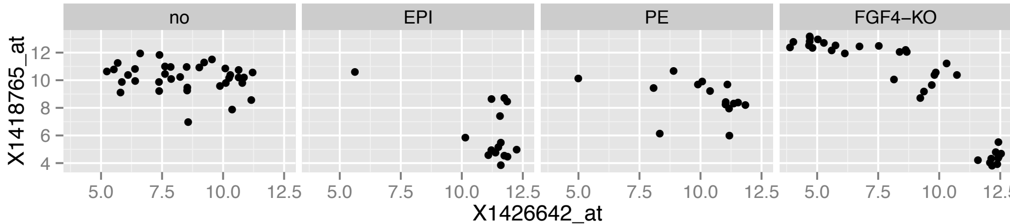
Sawtoo
typically
than they fall (pronounced for high peaks, less for medium and not for lowest)

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



3-5 D, and faceting

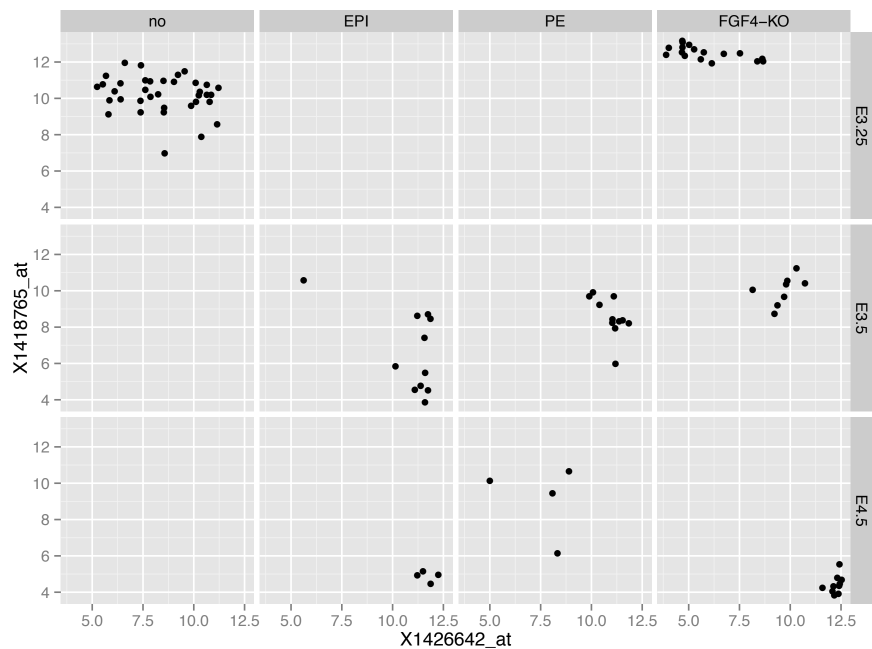
```
ggplot(dftx, aes( x = X1426642_at, y = X1418765_at)) +
  geom_point() + facet_grid( . ~ lineage )
```



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( Embryonic.day ~ lineage )
```

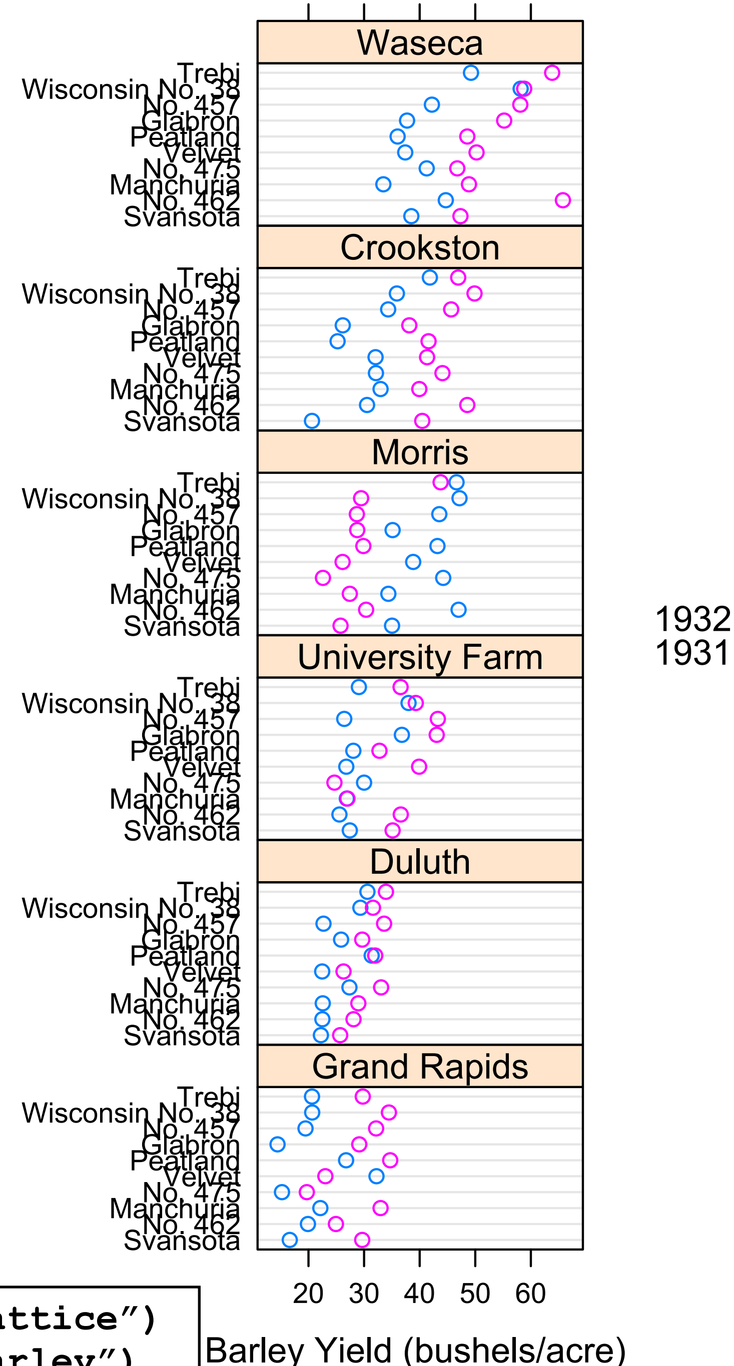


Data from an agricultural field trial to study the crop barley.

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)

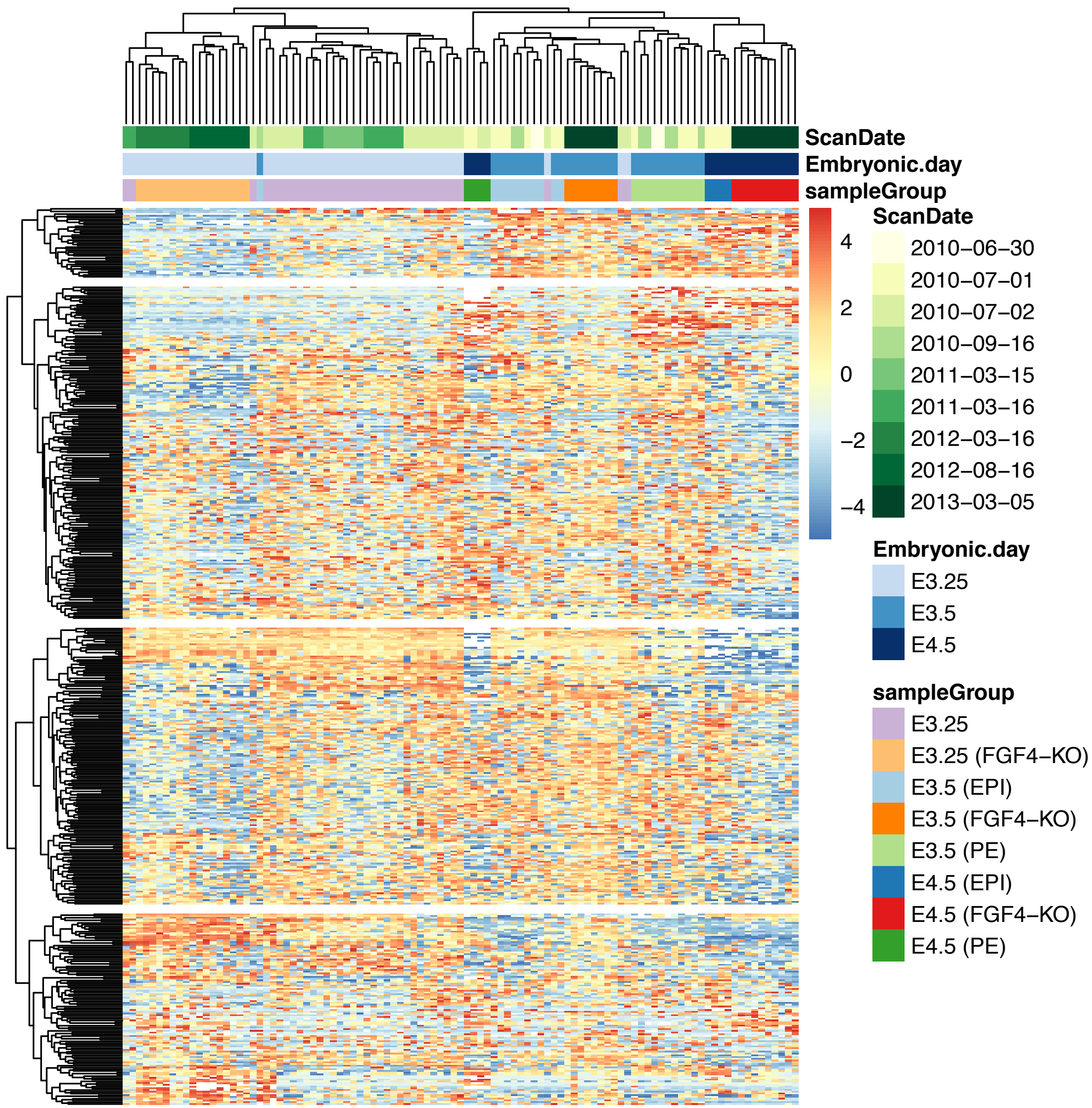
Note the data for Morris - reanalysis in the 1990s using Trellis revealed that the years had been flipped!




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

demo plotly

pheatmap



many reasonable defaults

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

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

Consider these:

Different requirements for line & area colours

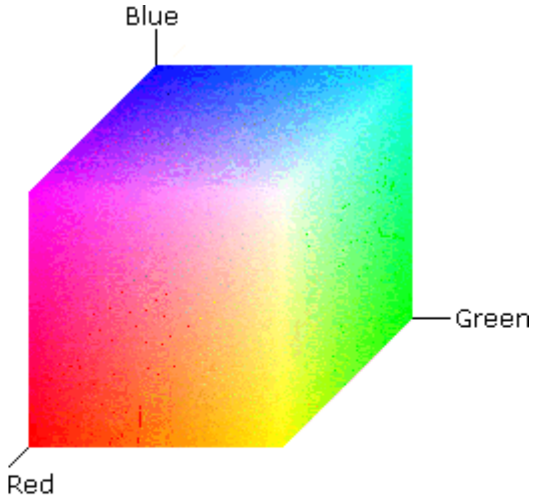
Many people are red-green colour blind

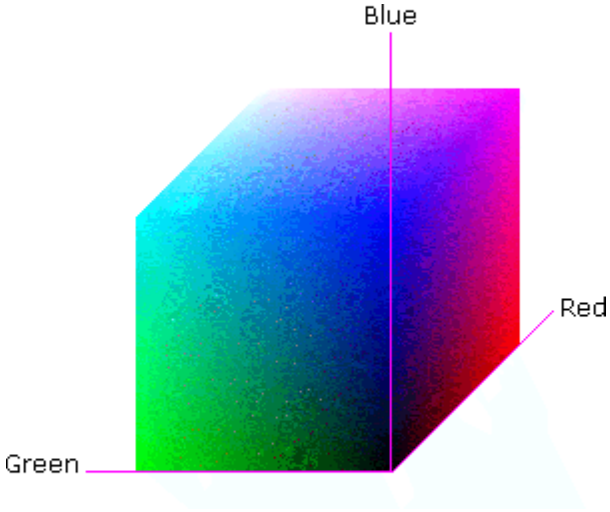
Lighter colours tend to make areas look larger than darker colours -> use colors of equal luminance for filled areas.

```r
display.brewer.
```

RGB color space

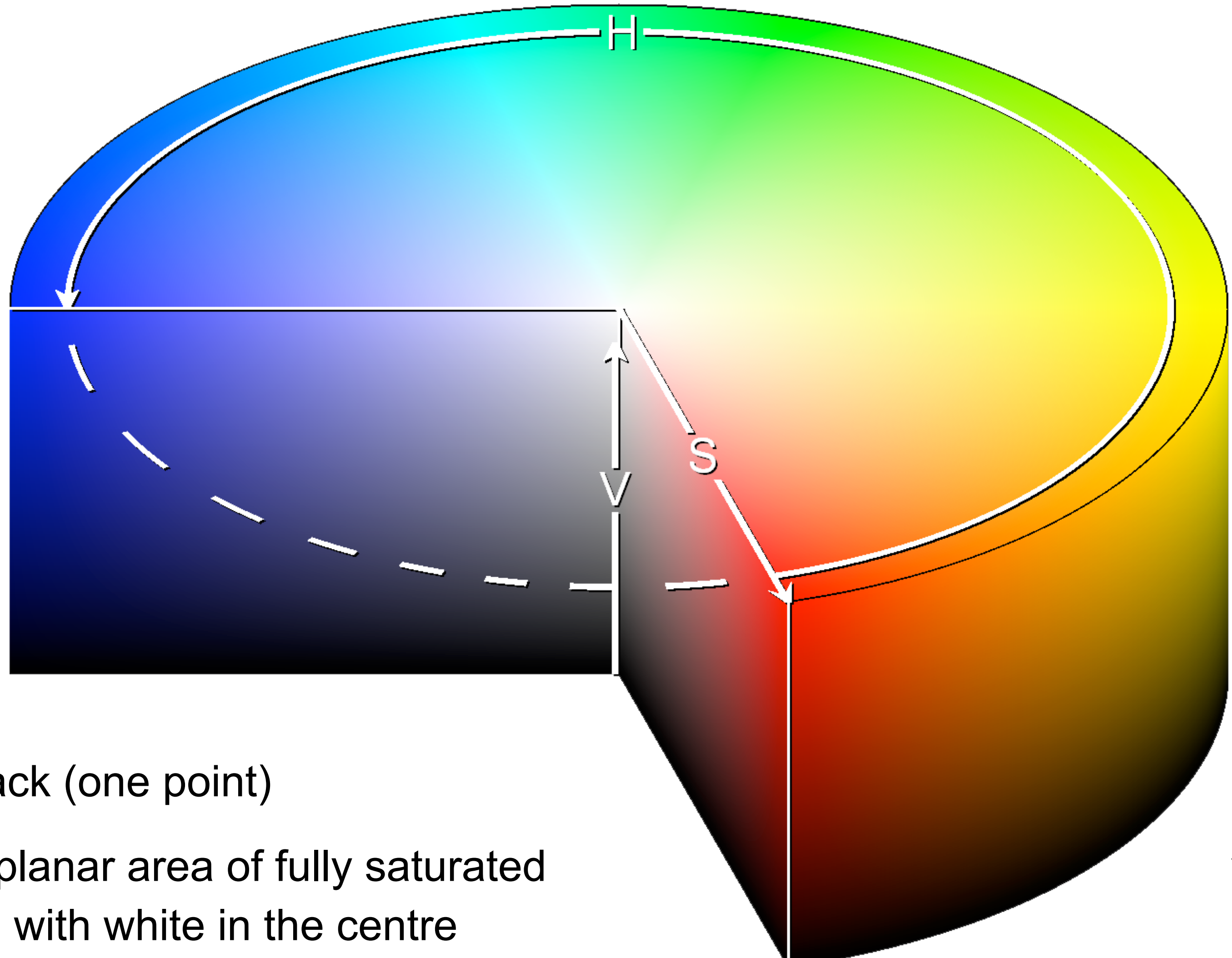
Motivated by computer screen hardware





HSV color space

Hue-Saturation-Value (Smith 1978)



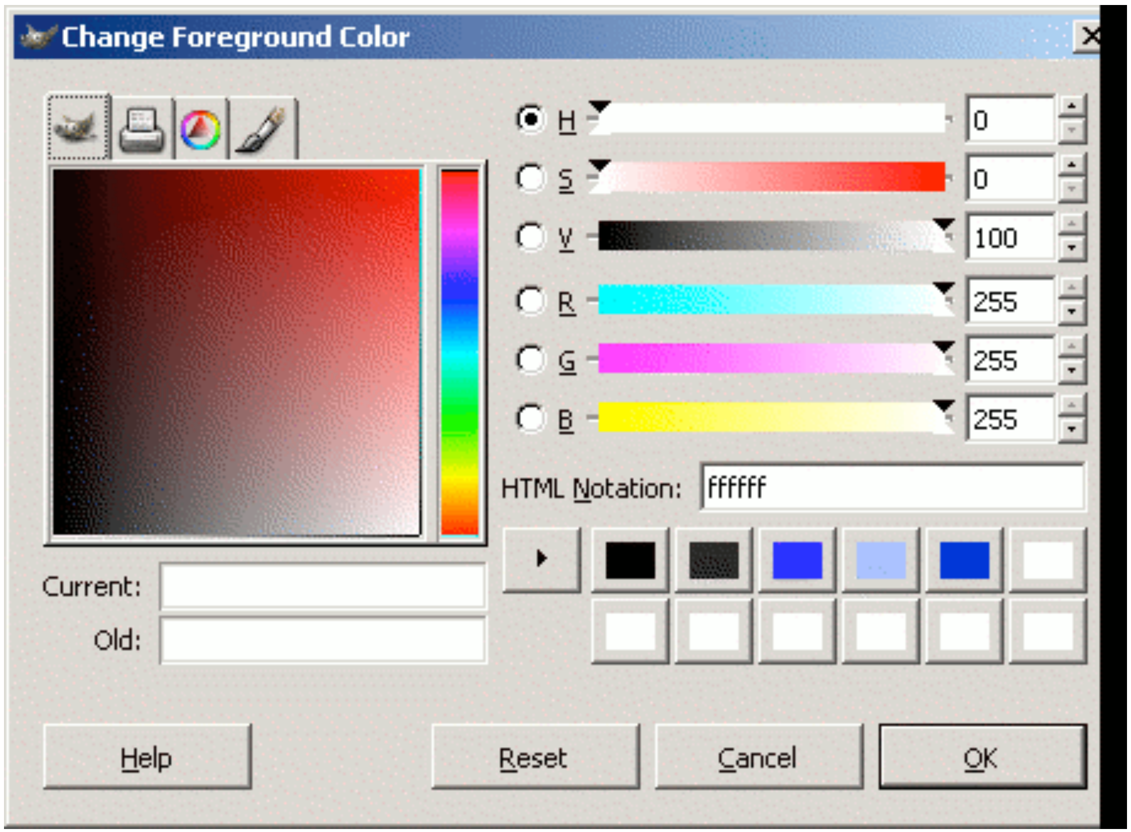
V_{min}: black (one point)

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

wikipedia

HSV color space

GIMP colour selector



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:

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

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

wikipedia

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.

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

Rather to polar coordinates in the colour plane (yellow/blue vs. green/red) plus a third light/dark axis. Perceptually-based colour spaces try to capture these perceptual axes:

1. hue (dominant wavelength)
2. chroma (colourfulness, intensity of coulor as compared to grey)
3. luminance (brightness, amount of grey)

CIELUV and HCL

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 (in principle, this could be monochromatic R, G, B; but CIE choose something a bit more subtle)

1976: CIELUV and CIELAB are perceptually based coordinates of colour space.

CIELUV (L, u, v)-coordinates is prefered by those who work with emissive colour technologies (such as computer displays) and CIELAB by those working with dyes and pigments (such as in the printing and textile industries)

Ihaka 2003

HCL colours

(u,v) = chroma * (cos h, sin h)

L the same as in CIELUV, (C, H) are simply polar coordinates for (u,v)

1. hue (dominant wavelength)
2. chroma (colorfulness, intensity of color as compared to gray)
3. luminance (brightness, amount of gray)

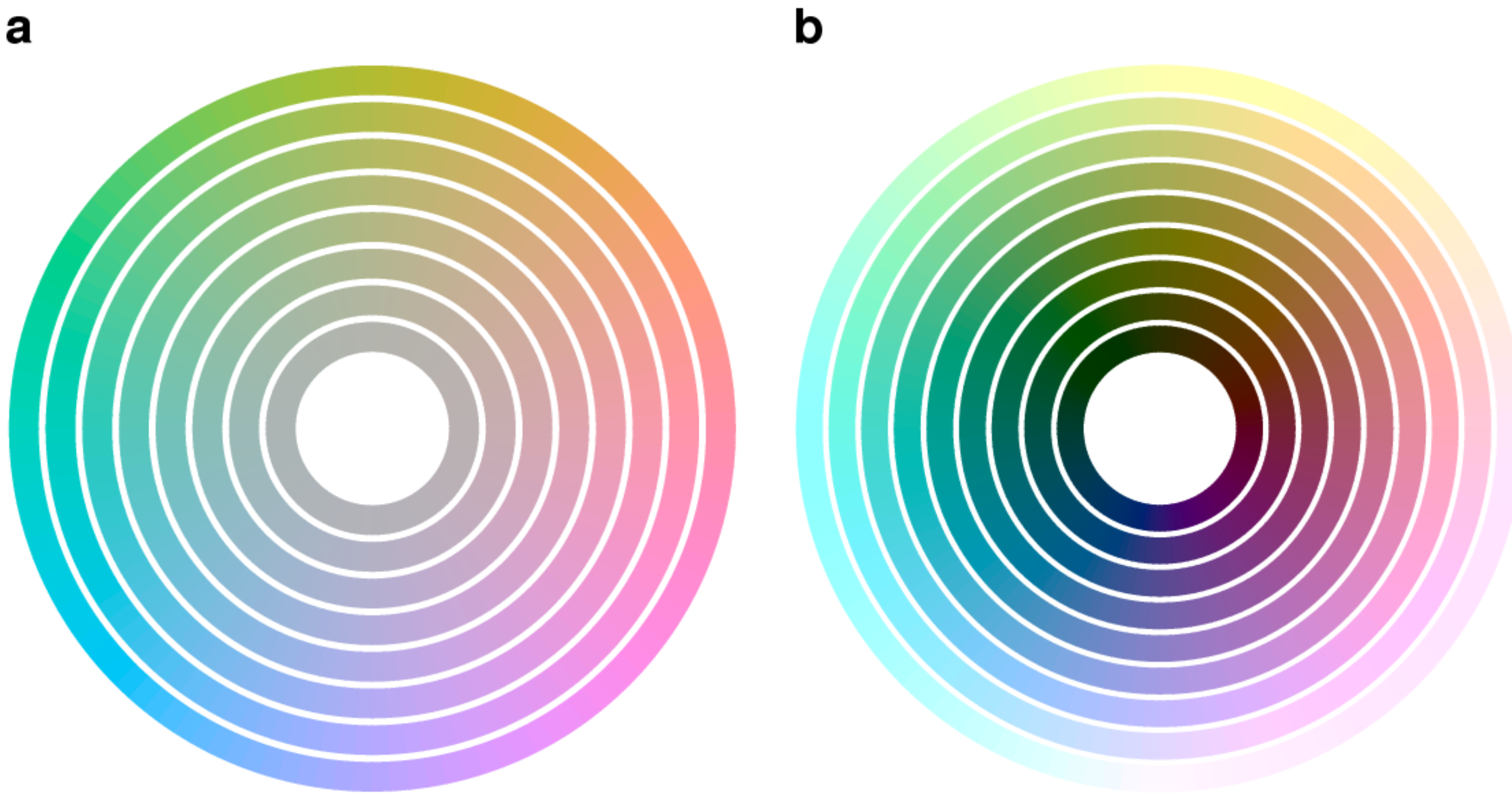
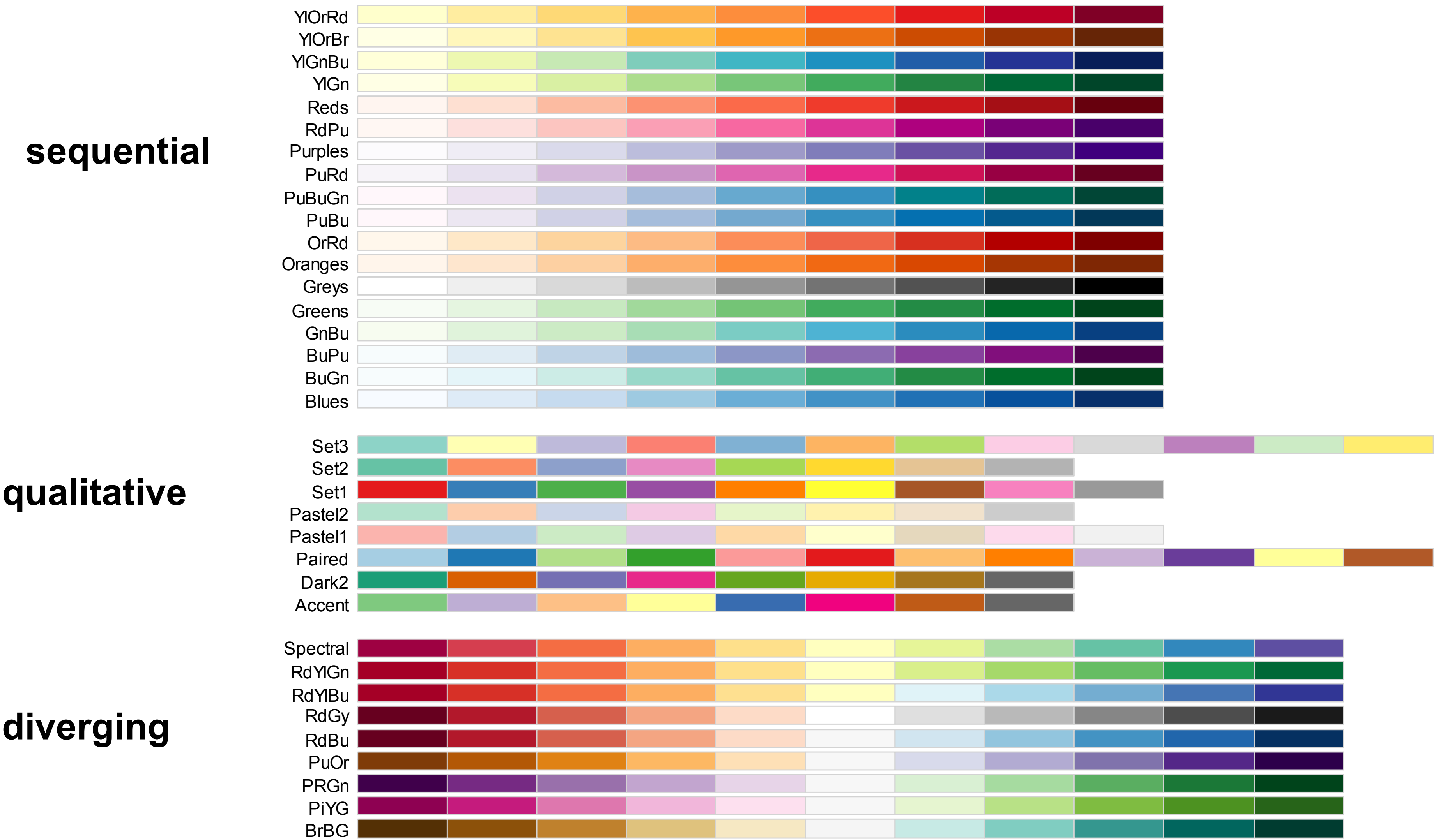
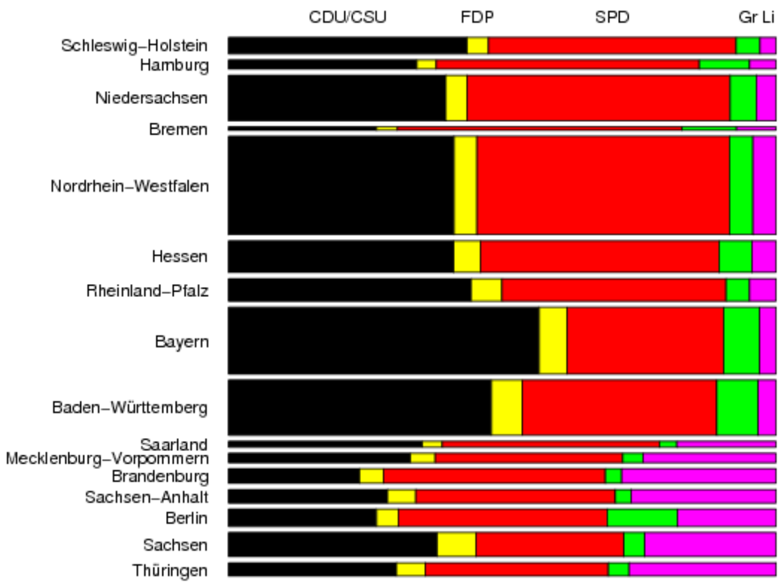


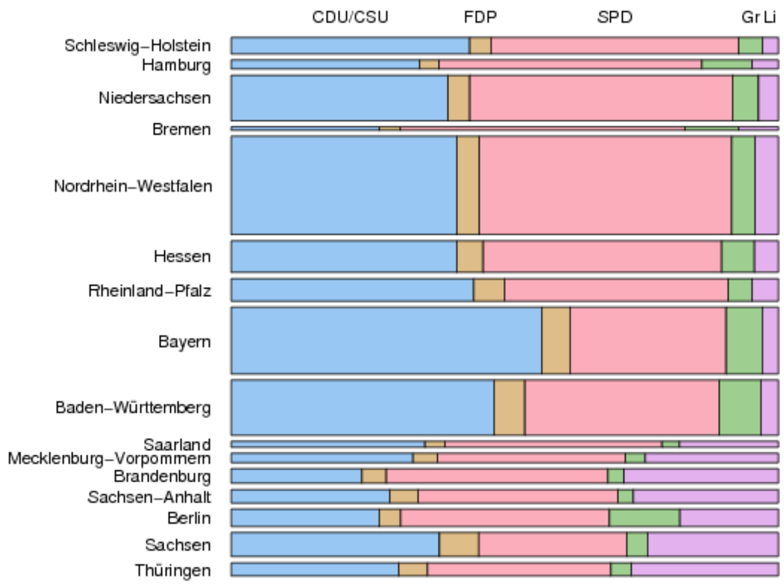
Figure 2: Circles in HCL colorspace. a: circles in HCL space at constant $L = 75$, with the angular coordinate H varying from 0 to 360 and the radial coordinate $C = 0, 10, \ldots, 60$. b: constant $C = 50$, and $L = 10, 20, \ldots, 90$.

RColorBrewer



Pick your favourite





From A. Zeileis, Reisensburg 2007

Acknowledgements

Susan Holmes

Robert Gentleman

Florian Hahne

Hadley Wickham

Ross Ihaka

Achim Zeileis

Kurt Hornik

References

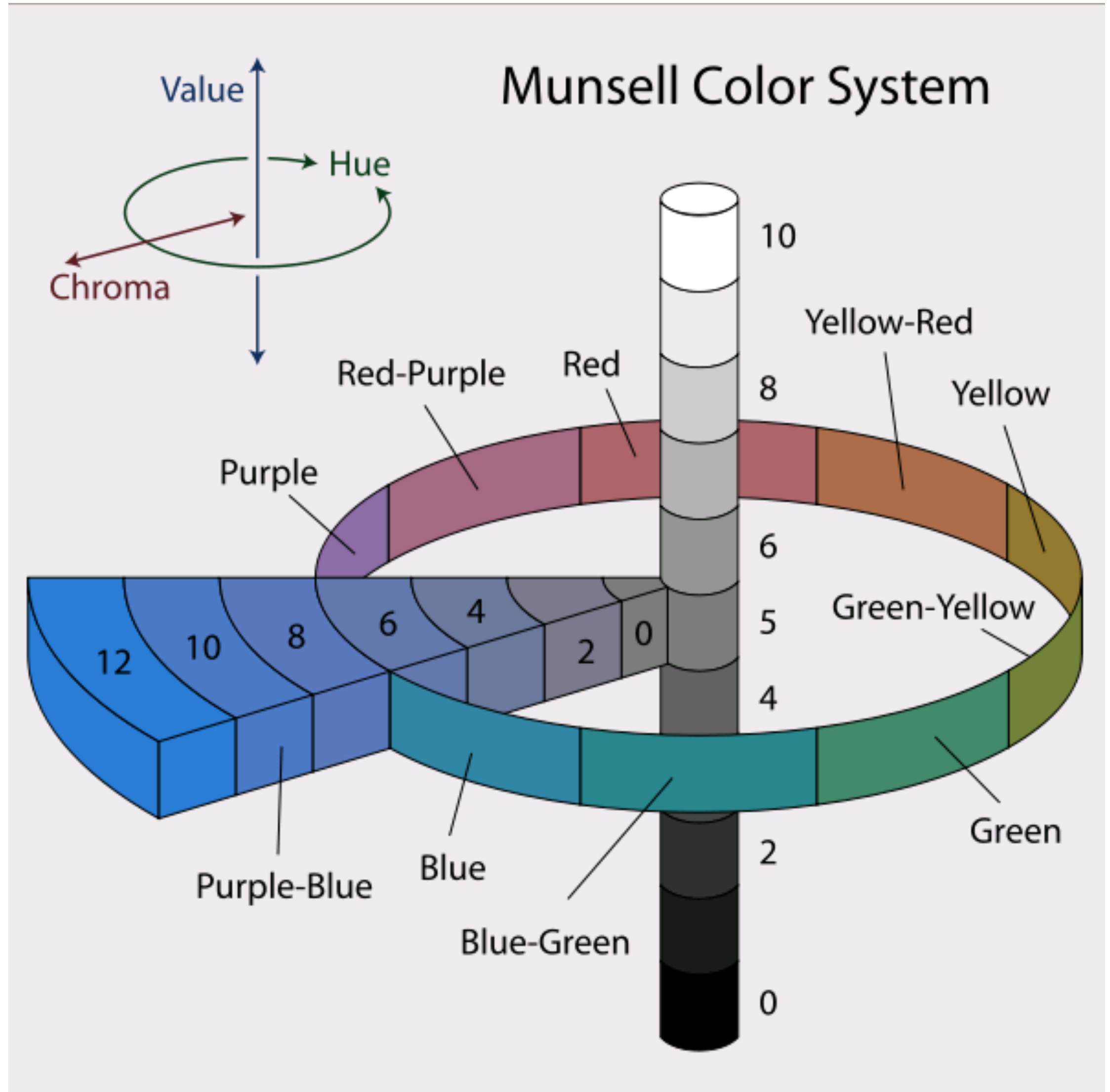
Visualizing Genomic Data, R. Gentleman, F. Hahne, W. Huber (2006), Bioconductor Project Working Papers, Paper 10

Choosing Color Palettes for Statistical Graphics, A. Zeileis, K. Hornik (2006), Department of Statistics and Mathematics, Wirtschaftsuniversität Wien, Research Report Series, Report 41

Albert Munsell (1858-1918) divided the circle of hues into 5 main hues — R, Y, G, B, P (red, yellow, green, blue and purple).

Value, Chroma: ranges divided into 10 equal steps.

E.g. R 4/5 = hue of red with a value of 4 and a chroma of 5.

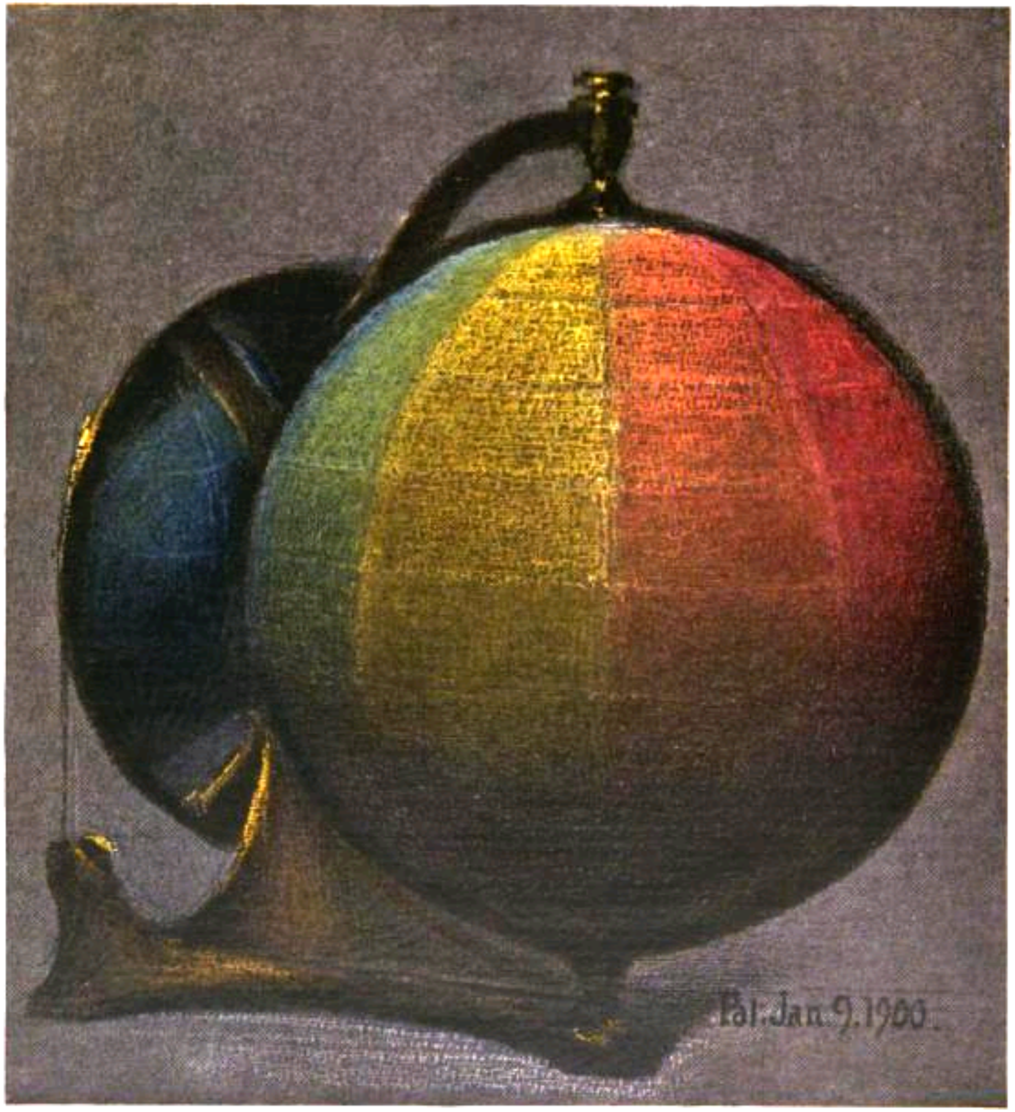


Munsell Colour System

Albert Munsell (1858-1918) divided the circle of hues into 5 main hues — R, Y, G, B, P (red, yellow, green, blue and purple).

Value, Chroma: ranges divided into 10 equal steps.

E.g. R 4/5 = hue of red with a value of 4 and a chroma of 5.



A BALANCED COLOR SPHERE

Colour Harmony

Figure 3: The principal Munsell 5/5 colours. From the top these are R 5/5, Y 5/5, G 5/5, B 5/5 and P 5/5. This figure is redrawn from Birren (1969).

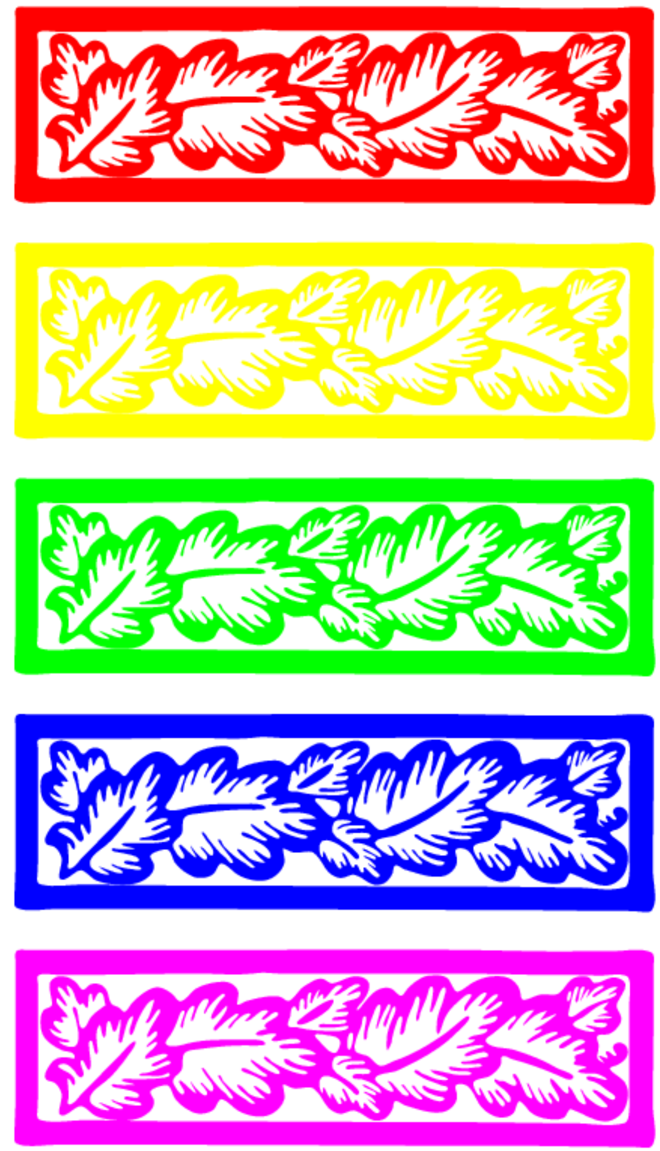


Figure 4: The same images as Figure 3, but drawn with full saturation HSV colours.

Balance

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

Choose colours centered 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.