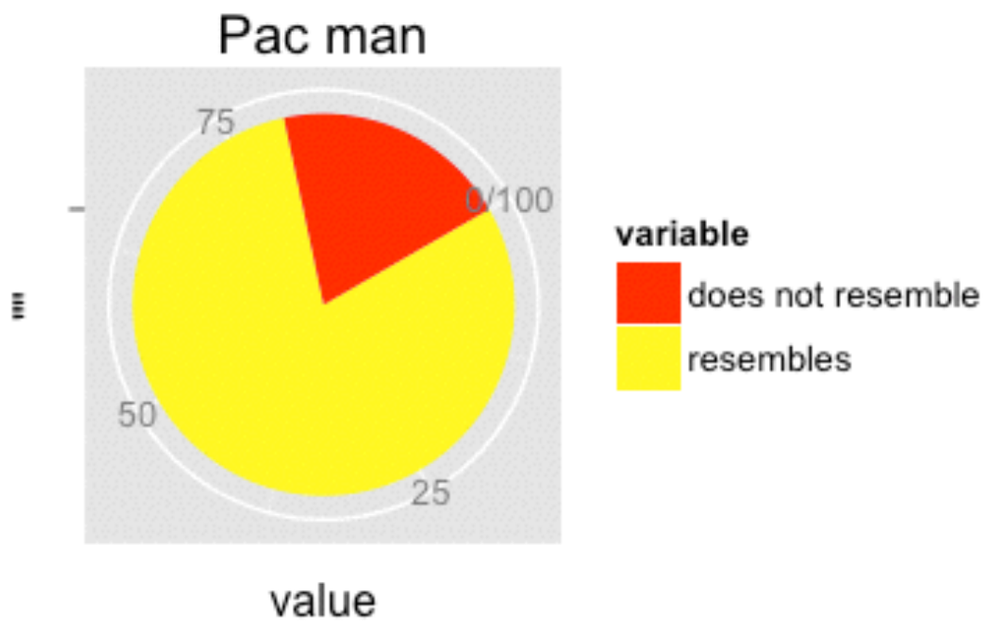
Visualization in R

Janos Binder



Acknoledgement

- Big thanks goes for Bernd Klaus for providing materials for this lecture.

Advanced graphics in R

- base graphics and ggplot2 (grammar of graphics) are commonly used to produce plots in R

- **base R:** “canvas” model - starting with a white space and adding graphical elements step by step

- **ggplot2:** “grammar” of graphics model - starting organizing the data in the right way, then a **plot is a mapping from data to aesthetics**

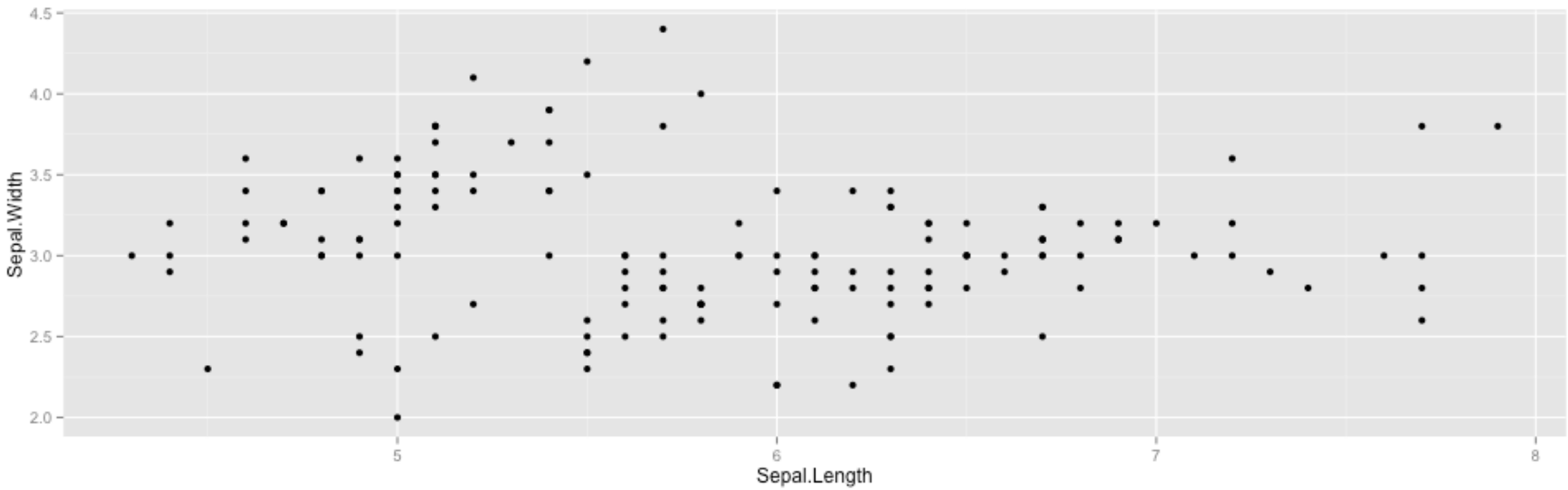
- **aesthetics** = things that you can visually perceive: color, shape or geometric objects like points, lines, bars

Before we begin

- An overview about the functions: http://docs.ggplot2.org
- We use Iris dataset:
  - built in R, originates from Ronald Fisher
  - 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor)

ggplot2 example

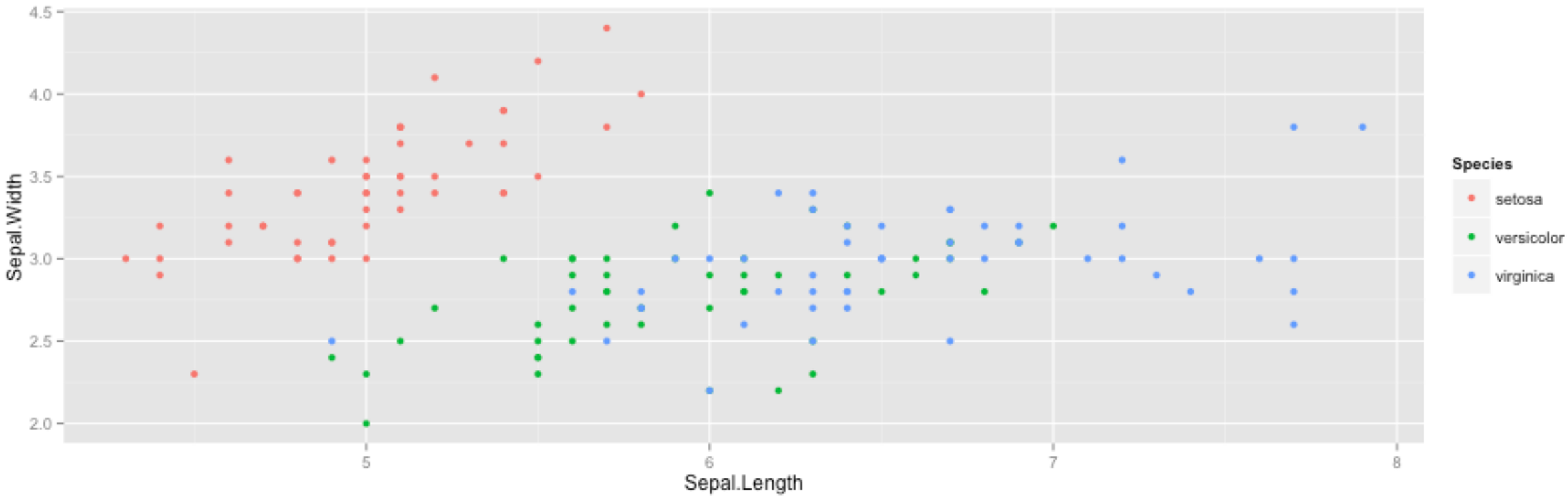
- iris dataset - simple scatterplot (Sepal.Length, Sepal.Width)
- **`p <- ggplot(iris, aes(Sepal.Length, Sepal.Width) )`**
- point geometry -> a scatterplot: **`p + geom_point()`**
- using the ggplot version of plot(): **`qplot(Sepal.Length, Sepal.Width, data = iris)`**



example continued

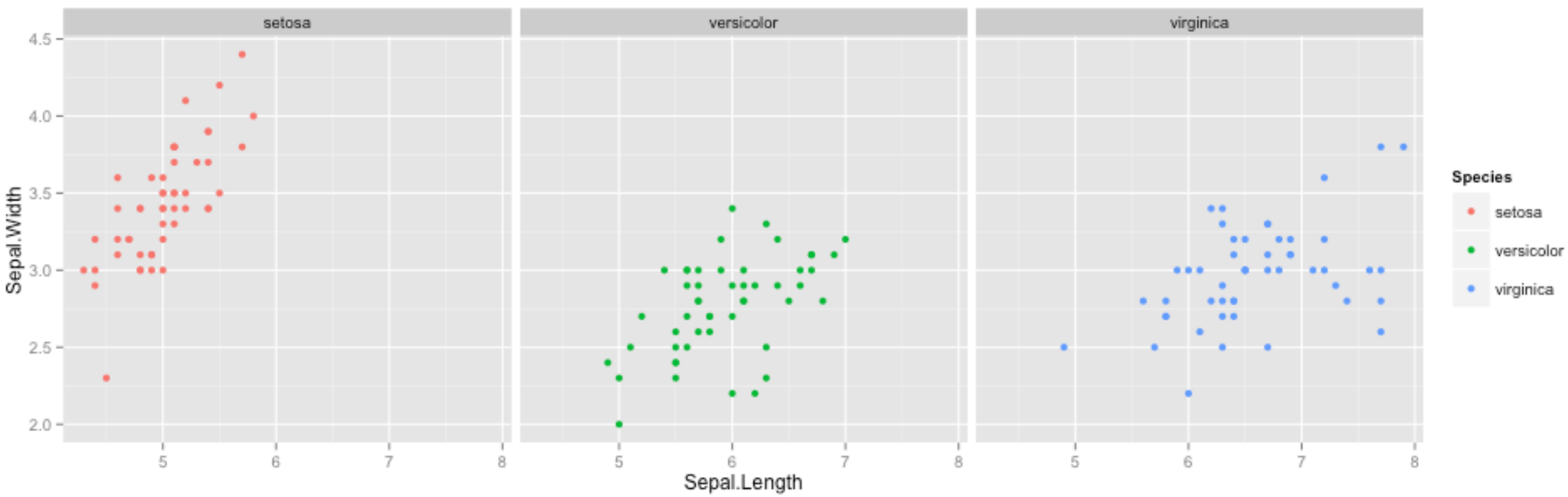
- mapping species to color: **`p + geom_point(aes(color = Species))`**

- **`qplot(Sepal.Length, Sepal.Width, data = iris, color = Species)`**



panels in ggplot

- splitting the plots into panels using factors
- **`p + facet_grid(. ~ Species)`**
- **`qplot(Sepal.Length, Sepal.Width, data = iris, color = Species, facets = . ~ Species)`**



Summary Statistics for Discrete Data

- Discrete data - only countable number of values x_1,...,x_k (possibly infinite)

- unordered (categorical), or ordered (ordinal)

- The common R data type - factor

- Summarizing discrete data in frequency tables

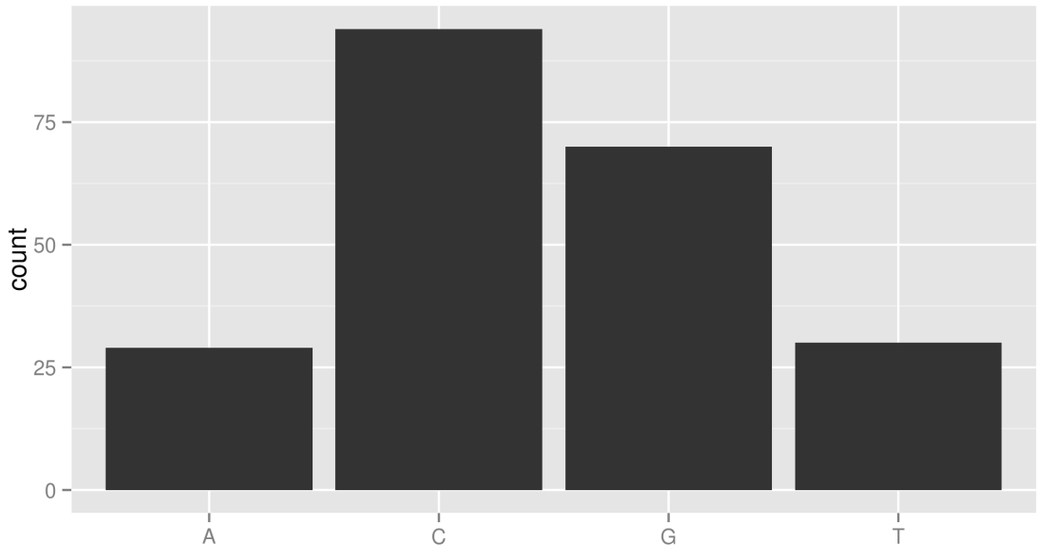
  - table(data): absolute frequencies

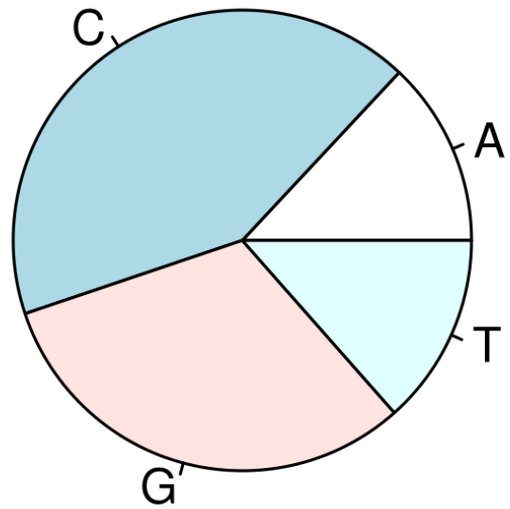
  - prop.table(data): relative frequencies

Discrete Data - Example

```
> DNA <- rep(c("A","C","G","T"), 10)

> table(DNA)

DNA
 A  C  G  T
10 10 10 10

> prop.table(table(DNA))

DNA
   A    C    G    T
0.25 0.25 0.25 0.25
```

Discrete Data - Pieplot and Barplot





- Counts per category by bars or as parts of a “pie”

Summary Statistics for Continuous Data

- A Random variable X is called continuous if it can have any value on the real line
  - Examples: Weight, Height, Size ....
- The common R data type - numeric
- Common Descriptive statistics for continuous data are
  - measures of location (mean(), median())
  - measures of scale (var(), sd(), IQR(), mad())

Mean $$\overline{x} = \frac{1}{k}\sum_{i=1}^{k} x_i = \frac{1}{n}\left(x_1 + \ldots + x_k\right)$$

Variance $$s^2 = \frac{1}{k-1}\sum_{i=1}^{k}(x_i - \overline{x})^2$$

Quantiles, Median and IQR

- Quantiles can be used to provide robust measures of scale and location
- x%-quantiles, divide data into two parts:
- x% of the data are below the x%-quantile and 100 – x% are above!

- x_0.5 - median
- x_0.25 - first quartile
- x_0.75 - third quartile
- median = measure of location
- IQR = Inter Quartile Range = x_0.75 - x_0.25 = measure of scale

Statistics for the- "Bodyfat" data set

- A variety of popular health books suggest that the readers assess their health by estimating their percentage of body fat ...
- Our illustrative data set "bodyfat" contains measurements of 15 variables that could be predictive for bodyfat taken on a sample of 252 individuals:

- age (years), weight (lbs), chest, neck, hip circumference ...

Computations

- Quick computation of several descriptive statistics for age: summary

```
> summary(age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  22.00   35.75   43.00   44.88   54.00   81.00
```

- sd (standard deviation)

```
> sd(age)
[1] 12.60204
```

- mean

```
> mean(age)
[1] 44.88492
```

- IQR

```
> IQR(age)
[1] 18.25
```

Boxplot

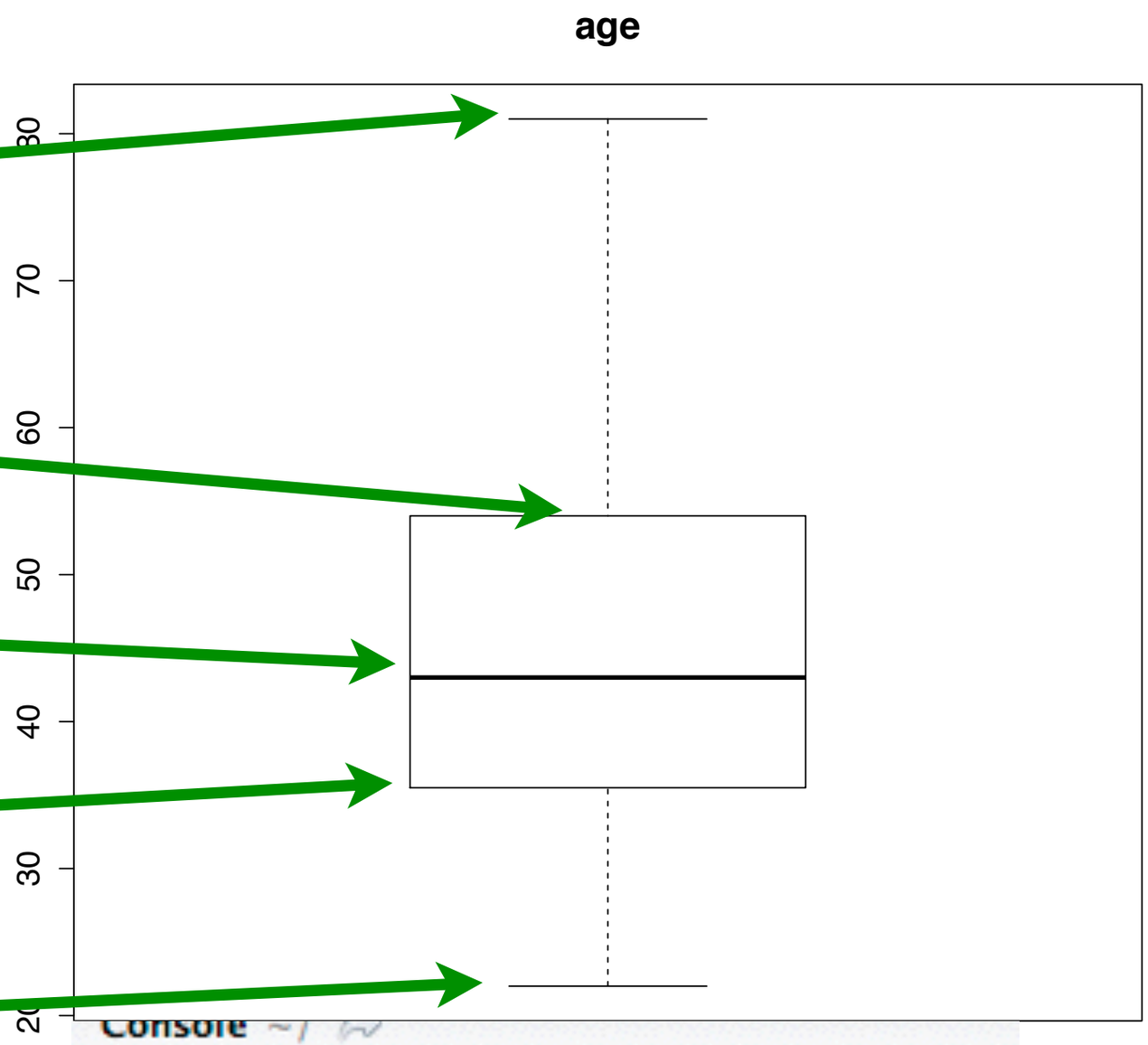
- The Boxplot is a graphical representation of several descriptive statistics:
  - Minimum and Maximum
  - 25%-quantile and 75%-quantile
  - Median
  - Outliers
- R command: boxplot(variable)

```
p <- ggplot(bodyfat, aes(1, age))
p + geom_boxplot()
```

Example - Boxplot of Age

- The Boxplot is a graphical representation of sever

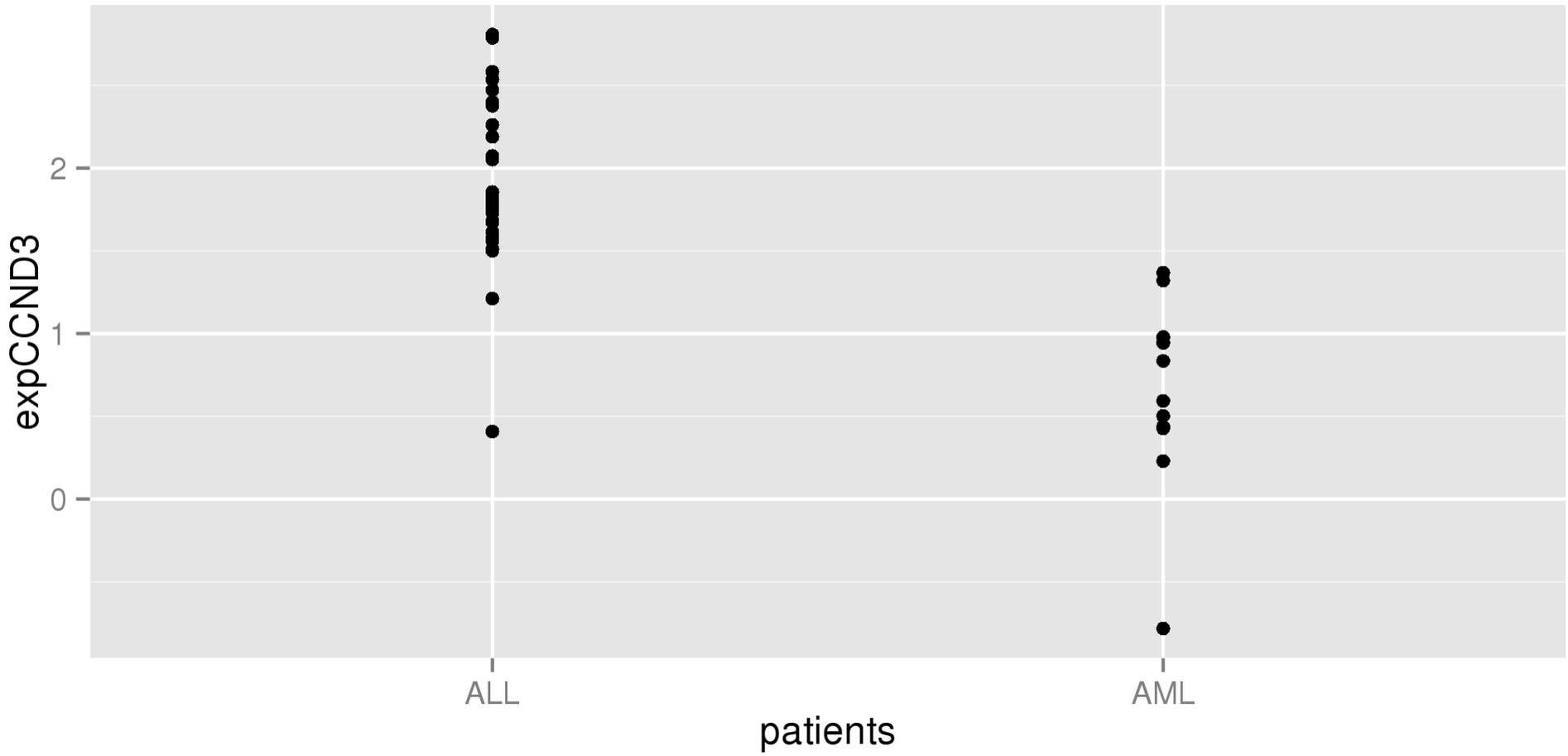
- **Maximum**
- **75%-quantile**
- **Median**
- **25%-quantile**
- **Minimum**




```
> boxplot(age, main = 'age')
>
```

Stripchart

- Alternative to a Boxplot if there are only few observations

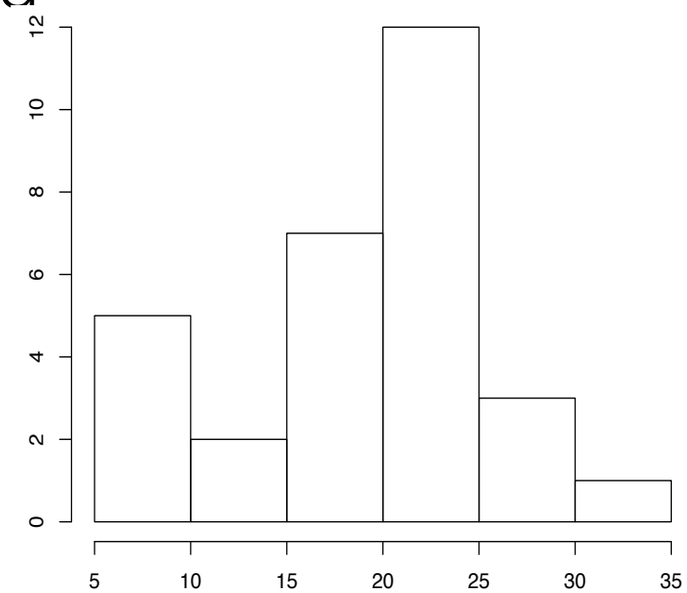


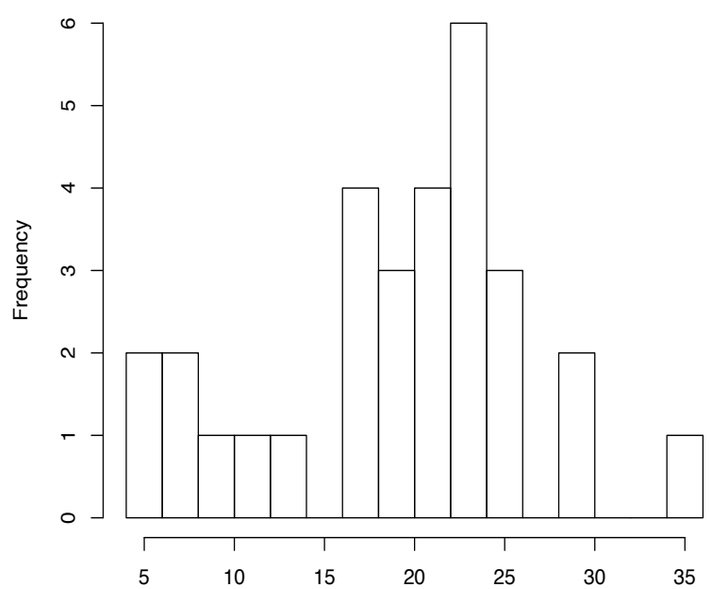

```
p <- ggplot(bodyfat, aes("Age", age))
p + geom_point(position = position_jitter(w = .0, h = .30))
```

Histogram

- A histogram gives an impression of the empirical density of the data

- A binning is performed and the absolute / relative frequency is plotted





hist(x, breaks = #Bins, freq = NULL)

breaks = number of bins

freq= TRUE / FALSE: frequencies / relative

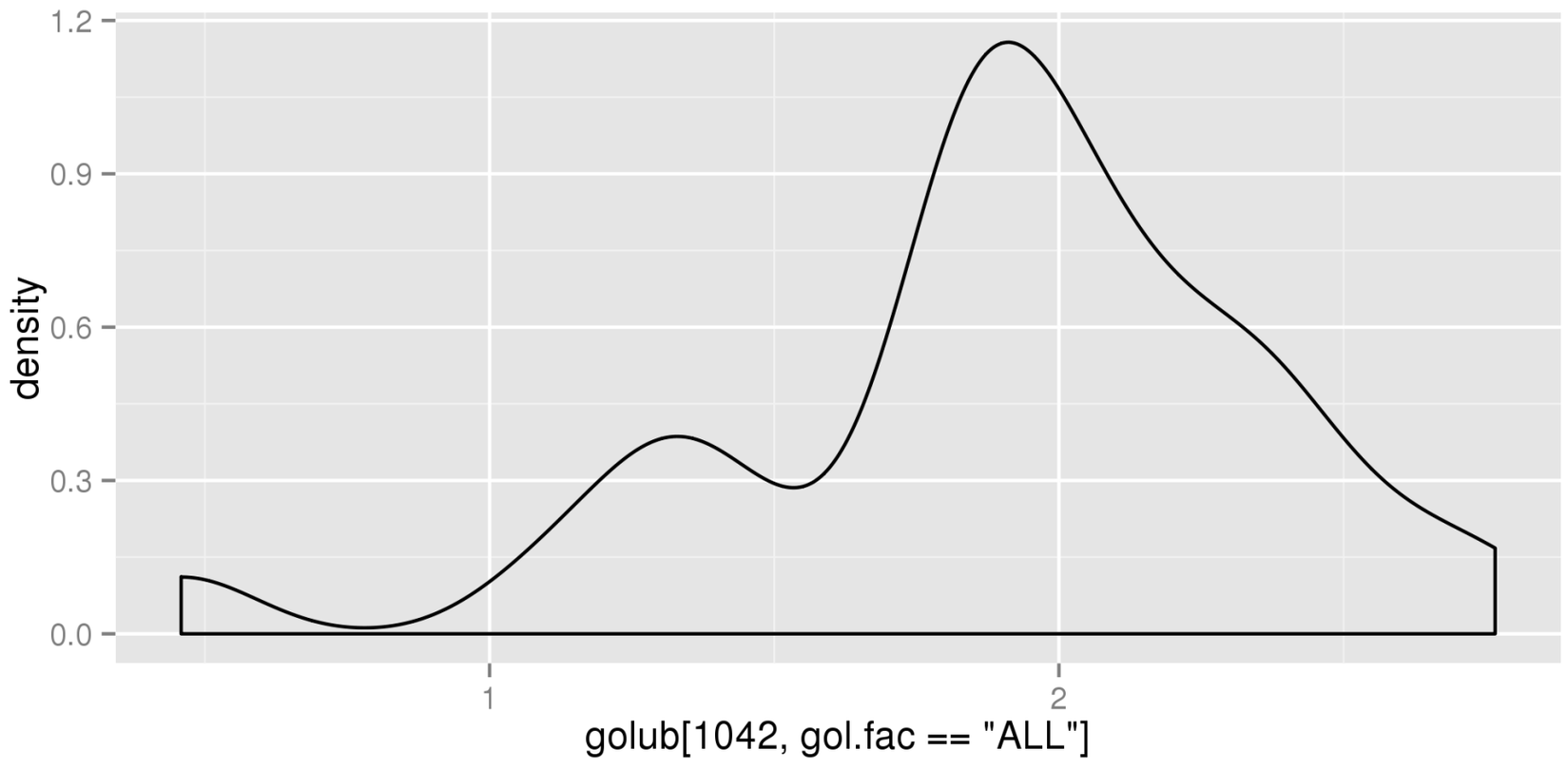
Density estimation

If $x_1, x_2, \ldots, x_N \sim f$ is an IID sample of a random variable, then the kernel density approximation of its probability density function is

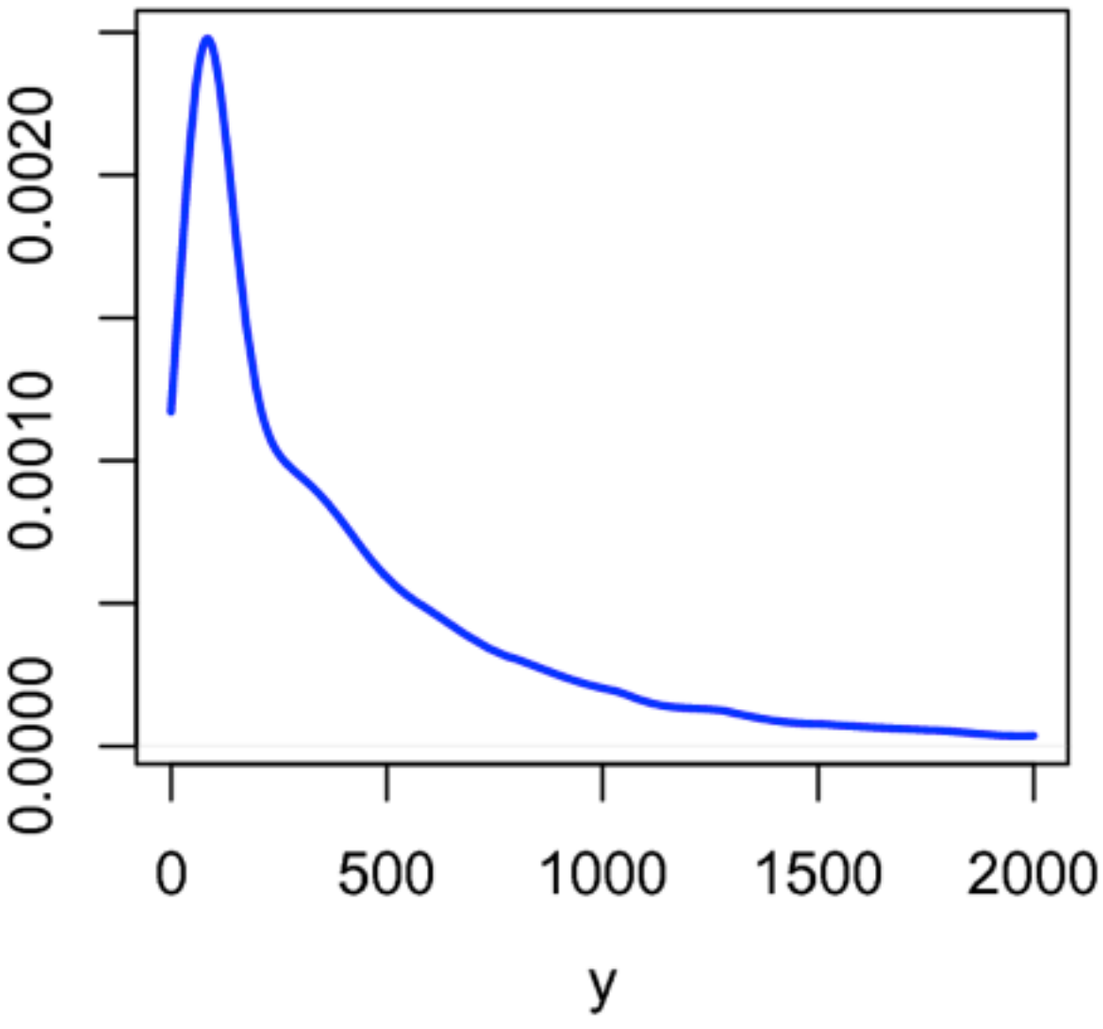
$$\widehat{f}_h(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)$$

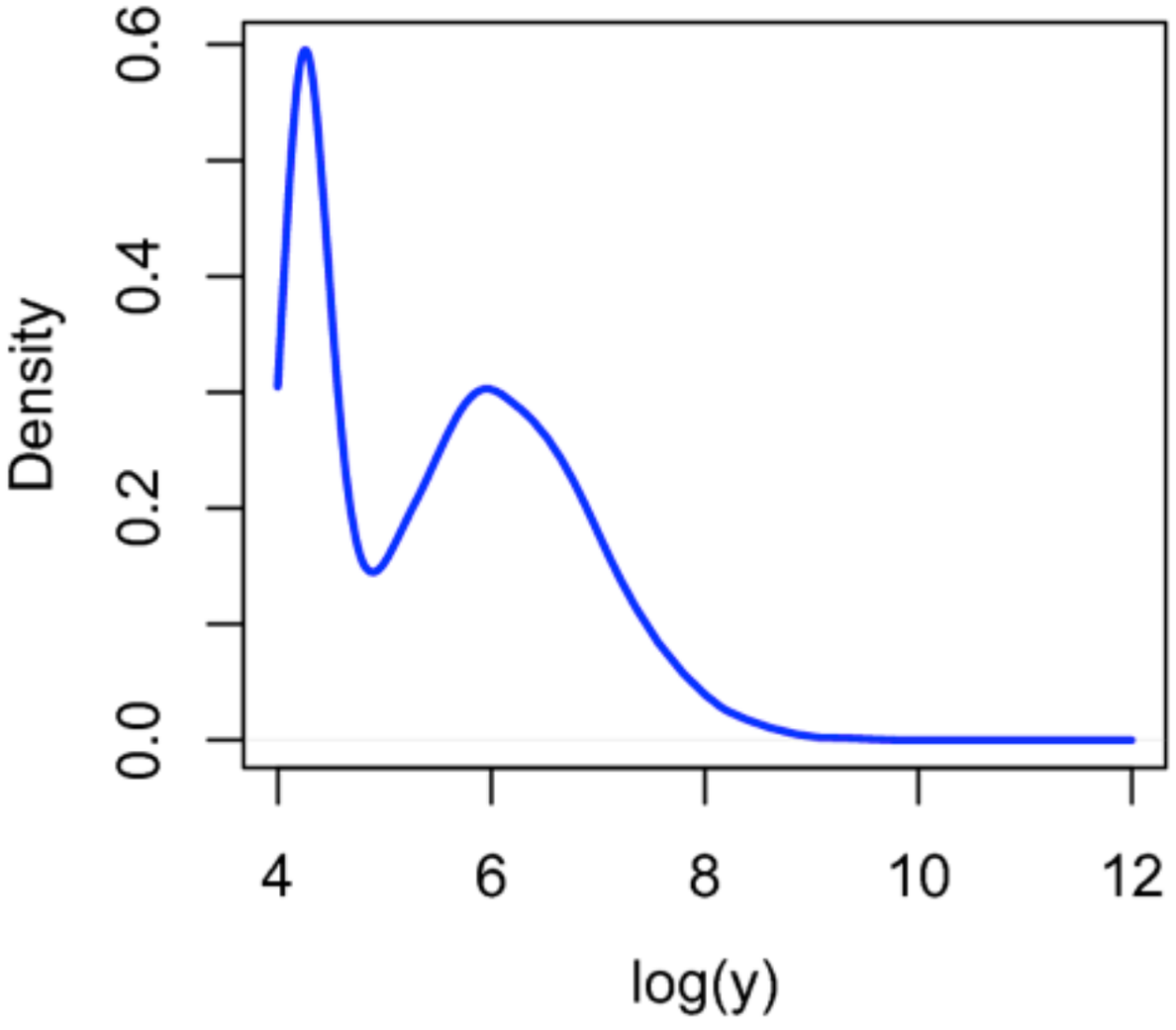
where K is some kernel and h is the bandwidth (smoothing parameter). Quite often K is taken to be a standard Gaussian function with mean zero and variance 1:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}.$$



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



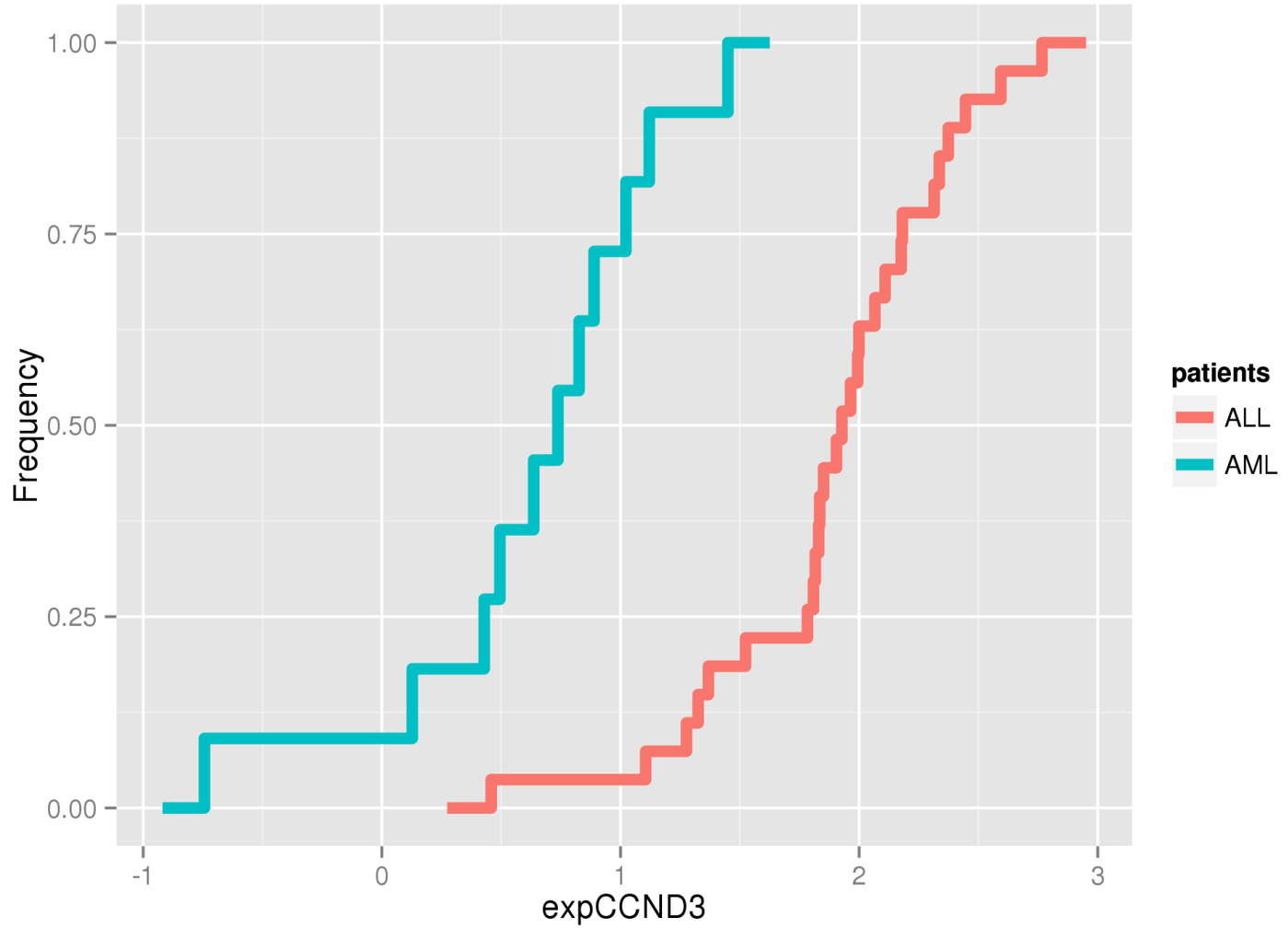


y: sample from a mixture of two log-normal distributions
kernel density estimates

Empirical Cumulative Distribution Function: ecdf

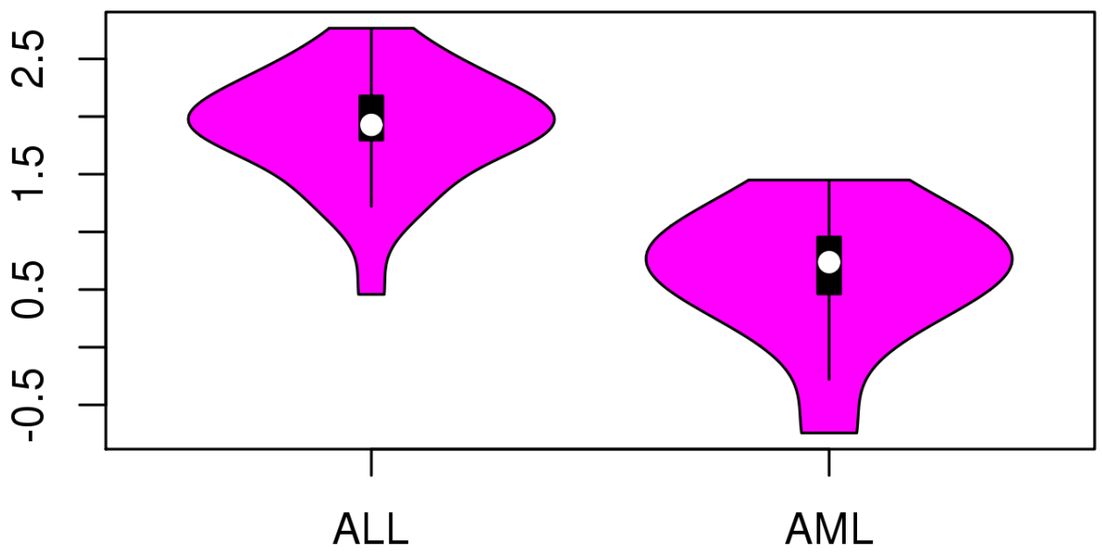
- "Frequency" is the fraction of data points with a value ≤ x

- R command: ecdf(x)



Violin-Plot

- A violin plot = boxplot + kernel density estimate
- ggplot: geom_violin()
- CRAN pkg: vioplot



Discussion: boxplot, histogramme, density, ecdf

- Boxplot makes sense for unimodal distributions, otherwise a violin plot may be used

- 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 its 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.

Using colors

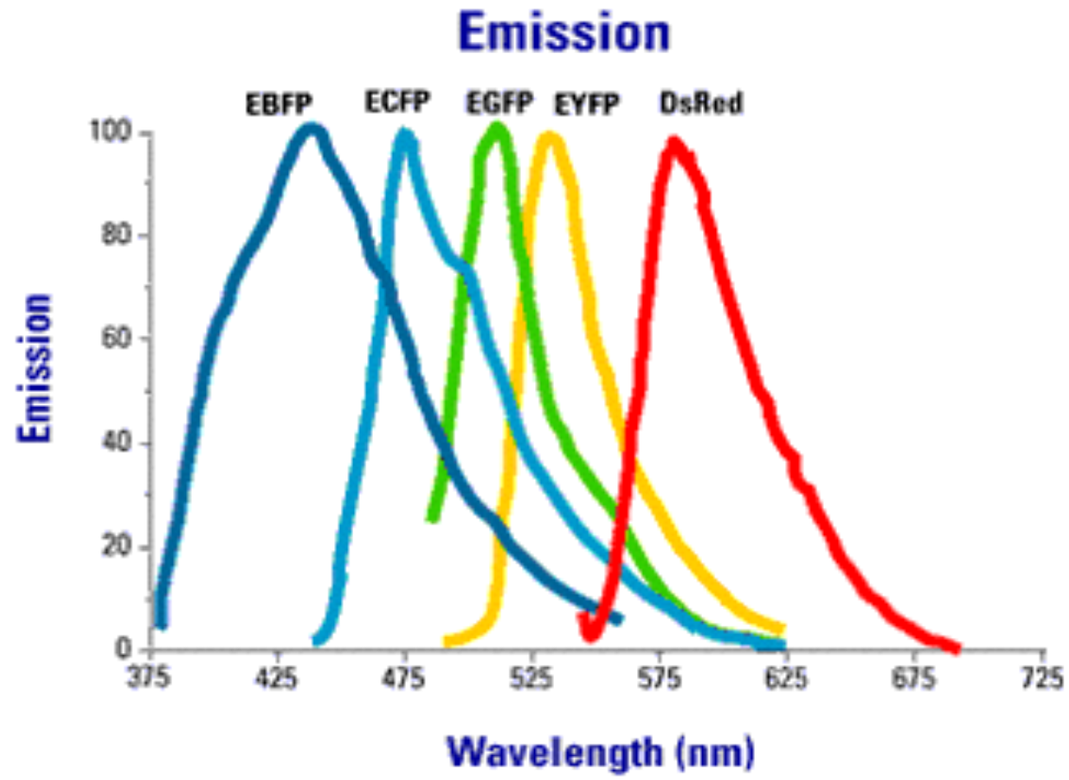
- Different requirements for line colors than for area colors

- Avoid artifacts related to human perception

- Many people are red-green color blind

- Lighter colors tend to make areas look larger than darker colors, thus colors of equal luminance should be chosen for graphics with large filled areas or where perception of area is important.

Light Emission Spectra

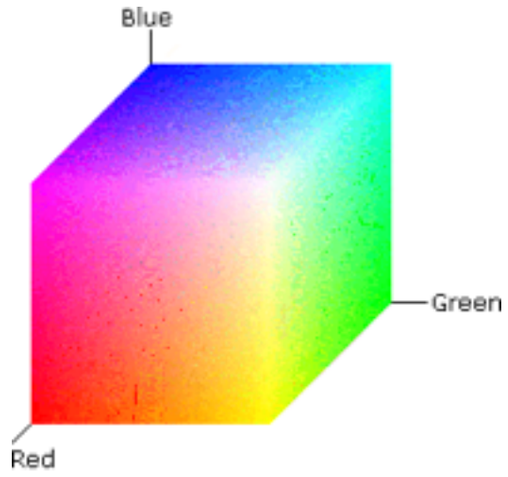


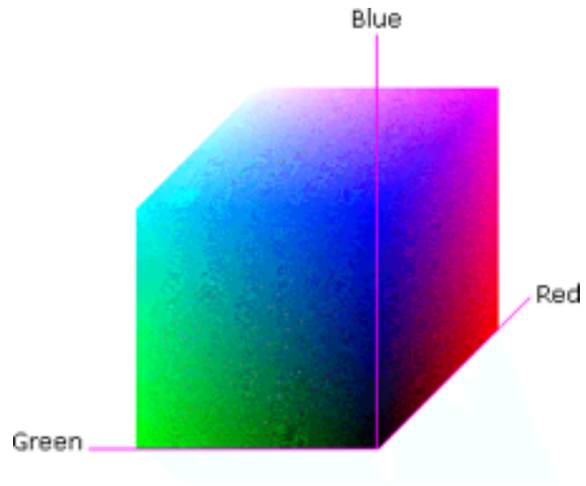
- The spectral density of light waves is a function of wavelength λ. This function space is infinite dimensional.

- Spectrometers measure such densities on a dense sampling grid. But our eyes are not a spectrometer.

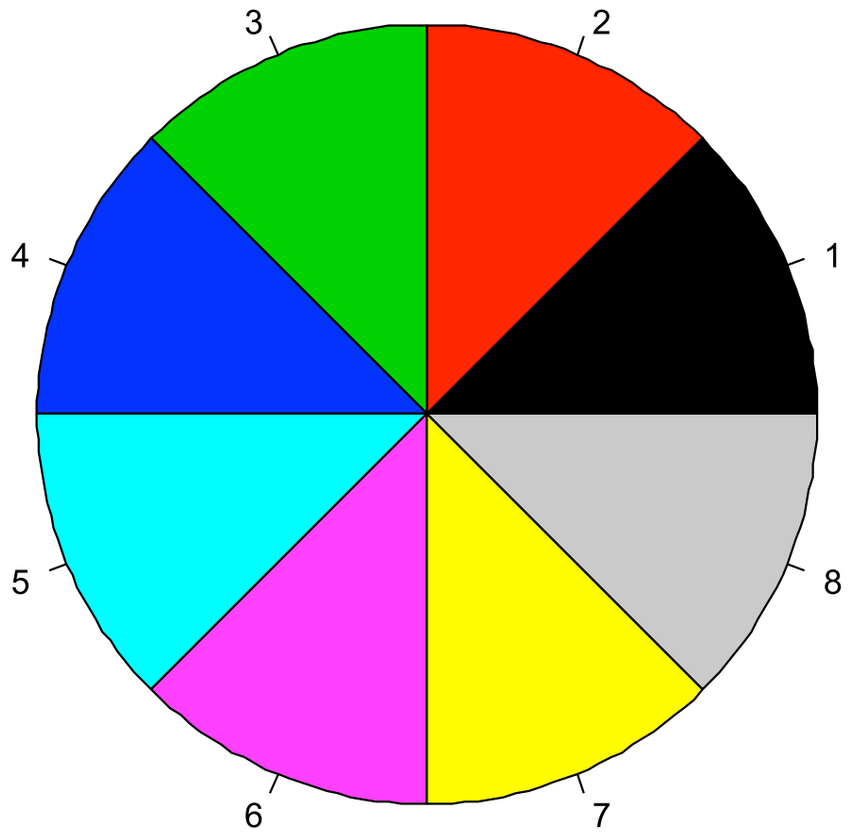
RGB color space

- Motivated by computer screen hardware





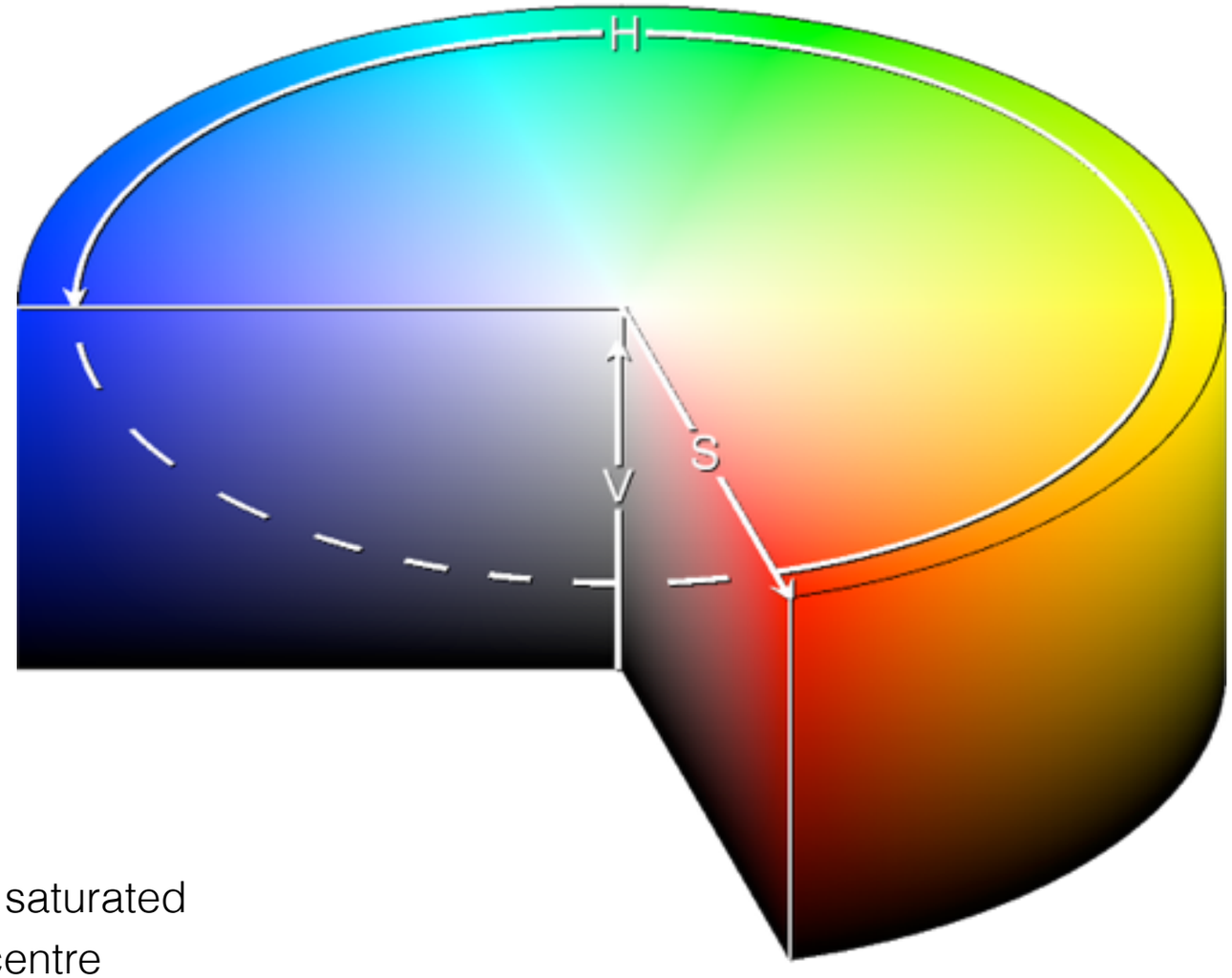
Color palettes based on the extremes of the RGB cube hurt the eyes




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

HSV color space

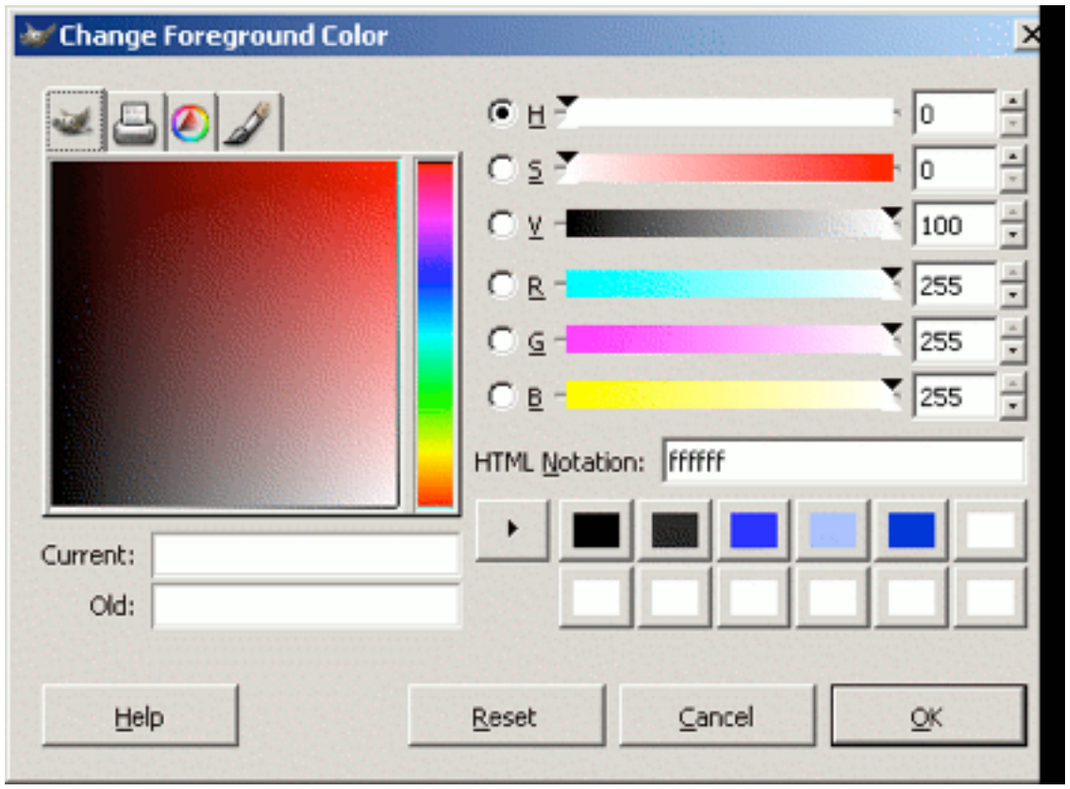
Hue-Saturation-Value (Smith 1978)



wikipedia

V_{min}: black (one point)

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



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

- GIMP colour selector

(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 color spaces

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

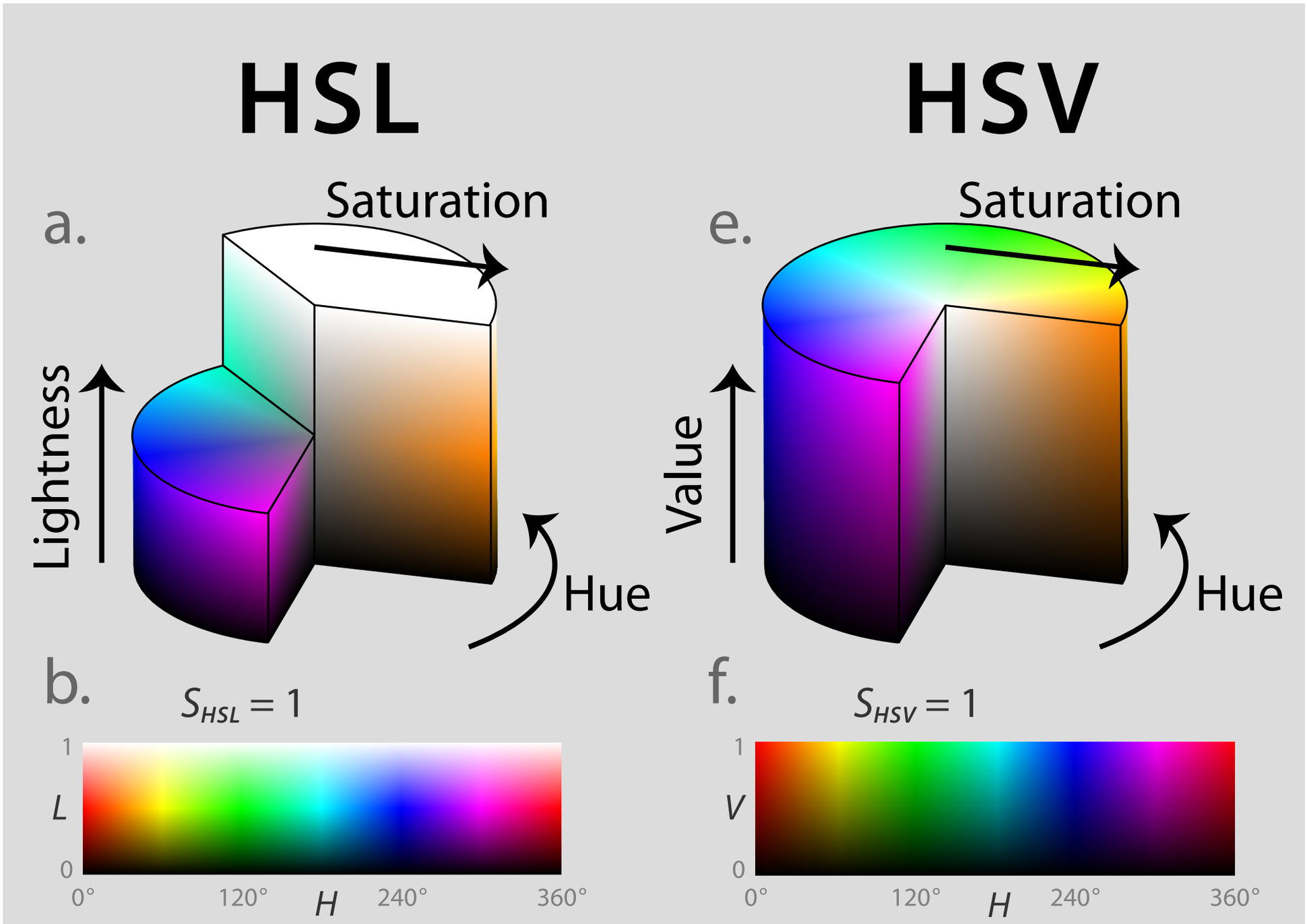
- Rather to polar coordinates in the color plane (yellow/blue vs. green/red) plus a third light/dark axis. Perceptually-based color spaces try to capture these perceptual axes:

  - 1. hue (dominant wavelength)

  - 2. chroma (colorfulness, intensity of color as compared to gray)

  - 3. luminance (brightness, amount of gray)

HCL colour coordinates: L is a more useful parameter of brightness



Wikipedia

CIELUV and HCL

- Commission Internationale de l' Éclairage (CIE) in 1931, on the basis of extensive color matching experiments with people, defined a "standard observer" who represents a typical human color 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 color space.

- CIELUV (L, u, v)-coordinates is preferred by those who work with emissive color 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)

- used by `ggplot2`

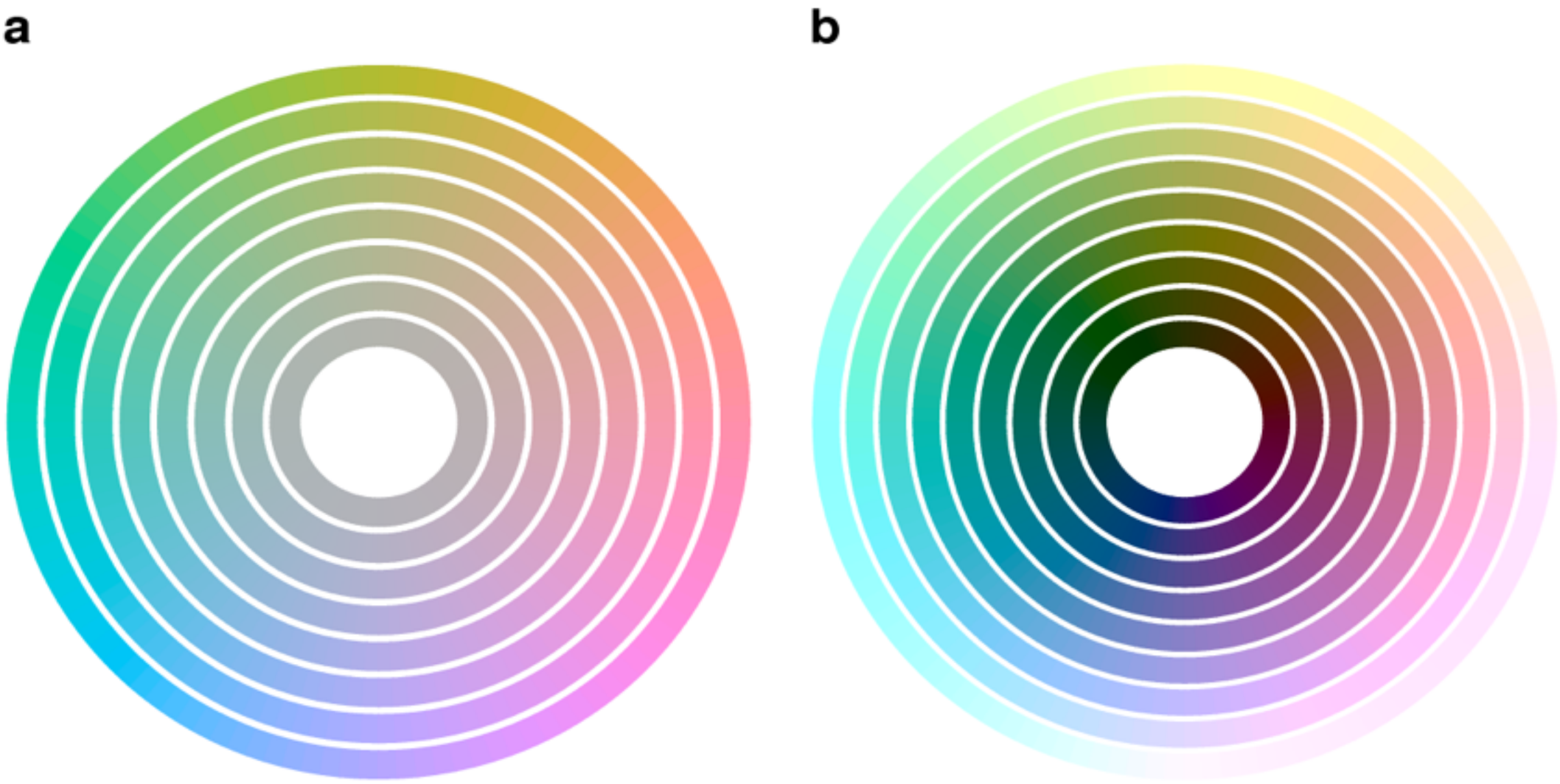
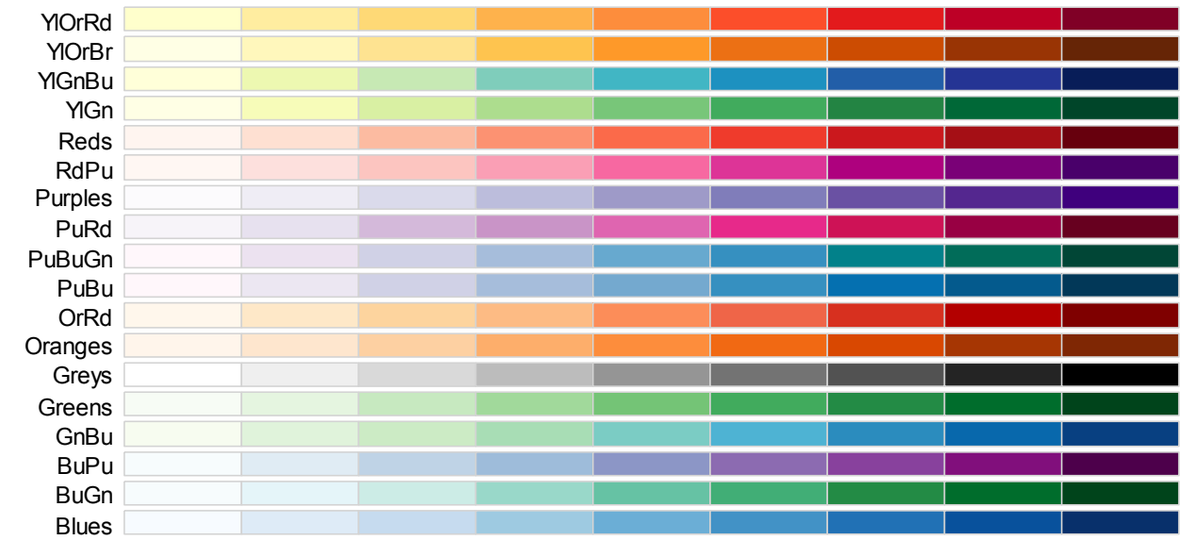


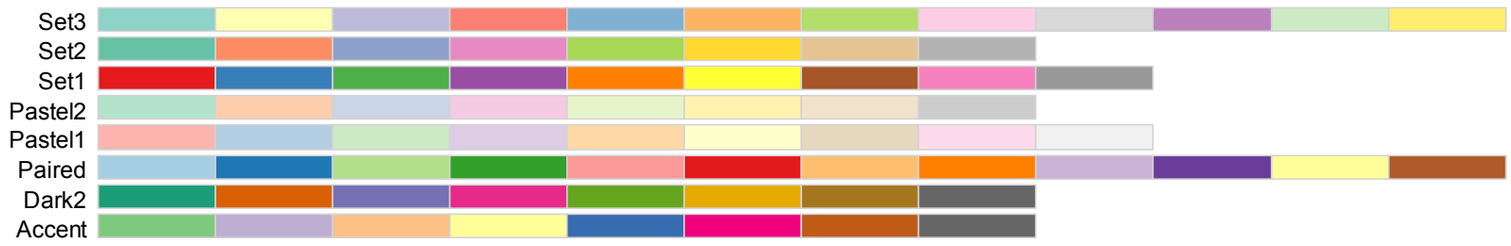
Figure 2: Circles in HCL colorspace. *a*: circles in HCL space at constant $L = 75$, with the angular coordinate H varying from 0 to 360 and the radial coordinate $C = 0, 10, \ldots, 60$. *b*: constant $C = 50$, and $L = 10, 20, \ldots, 90$.

Software

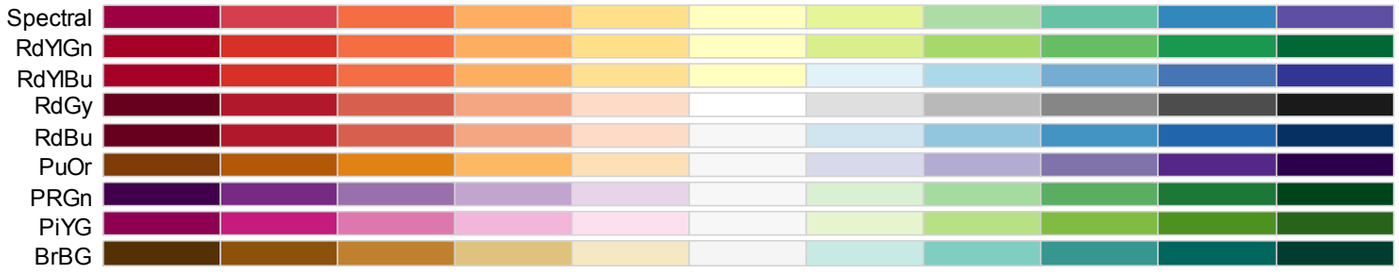
sequential



qualitative

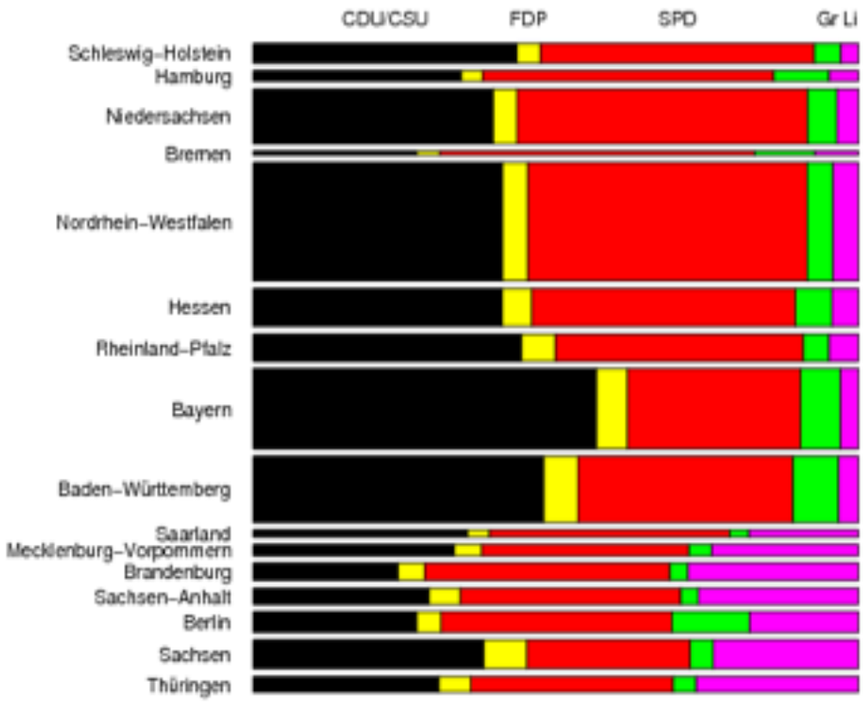


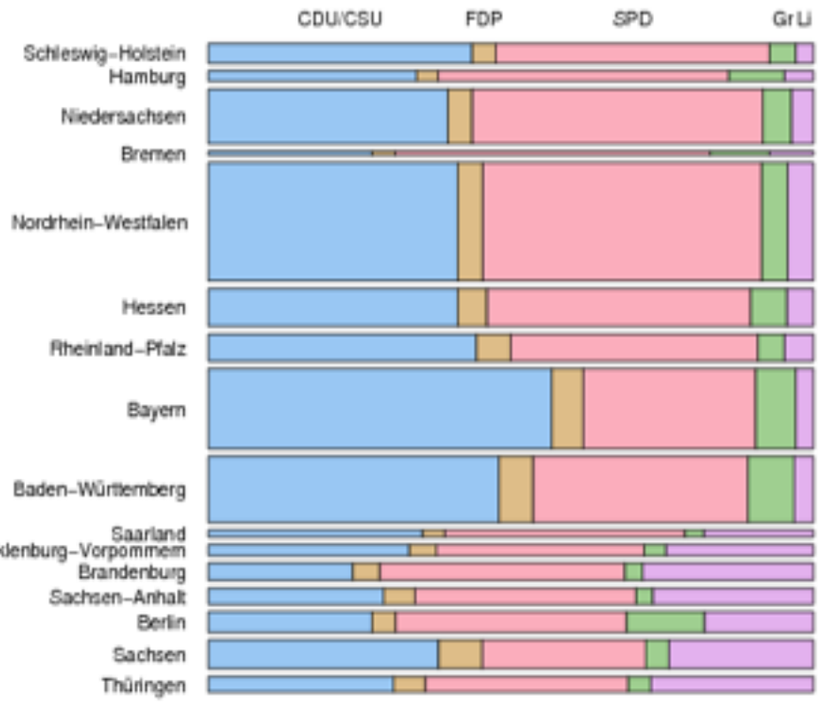
diverging



RColorBrewer and **vcd** packages

Biased and unbiased politics





From A. Zeileis, Reisensburg 2007

Some useful functions for working with colors

- `RColorBrewer`
  - `display.brewer.all` show all palettes
  - `brewer.pal` choose one particular palette

- `RColorBrewer`
  - `colorRamp, colorRampPalette` interpolate

- `vcd`
  - `sequential_hcl, diverge_hcl, rainbow_hcl` palettes

- ... and avoid R's default colors

Questions

- What is the main advantage of ggplot2 over the built plotting function of R?
- What is the main difference between qplot() and ggplot2() function?