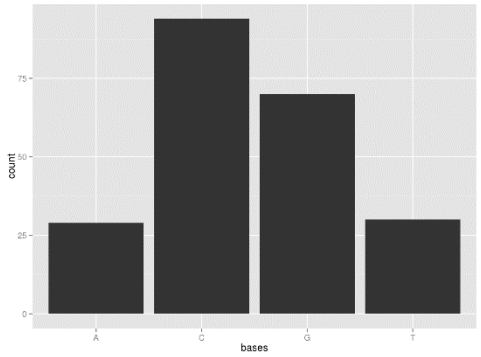
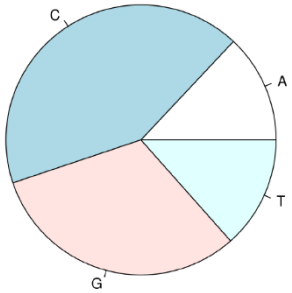
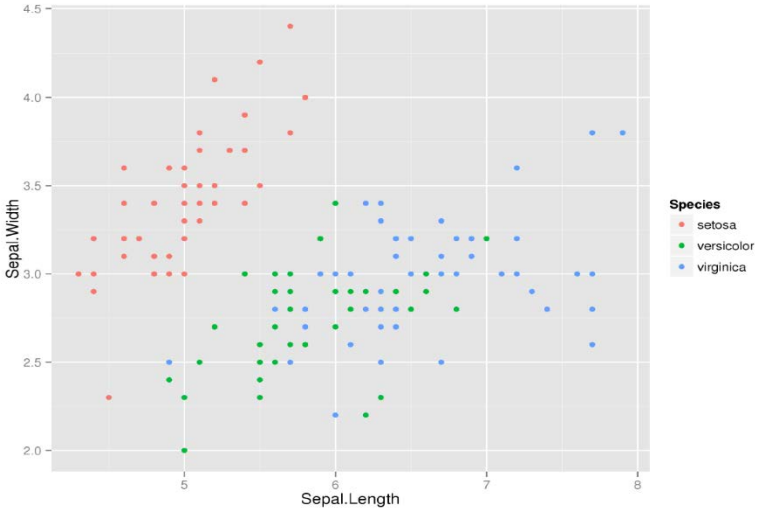
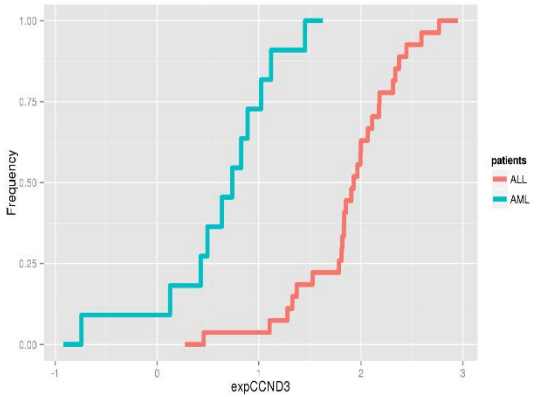
Exploratory Data Analysis and Graphics









Bernd Klaus, some input from Wolfgang Huber, EMBL

Graphics in R

base graphics and ggplot2 (grammar of graphics) are commonly used to produce plots in R; in a nutshell:

base R: “canvas” model you start with a white space and add graphical elements step by step

ggplot2: “grammar” of graphics model. You start by organizing your 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

Nice lectures by Roger Pen:

http://www.youtube.com/watch?v=HeqHMM4ziXA

http://www.youtube.com/watch?v=n8kYa9vu1l8

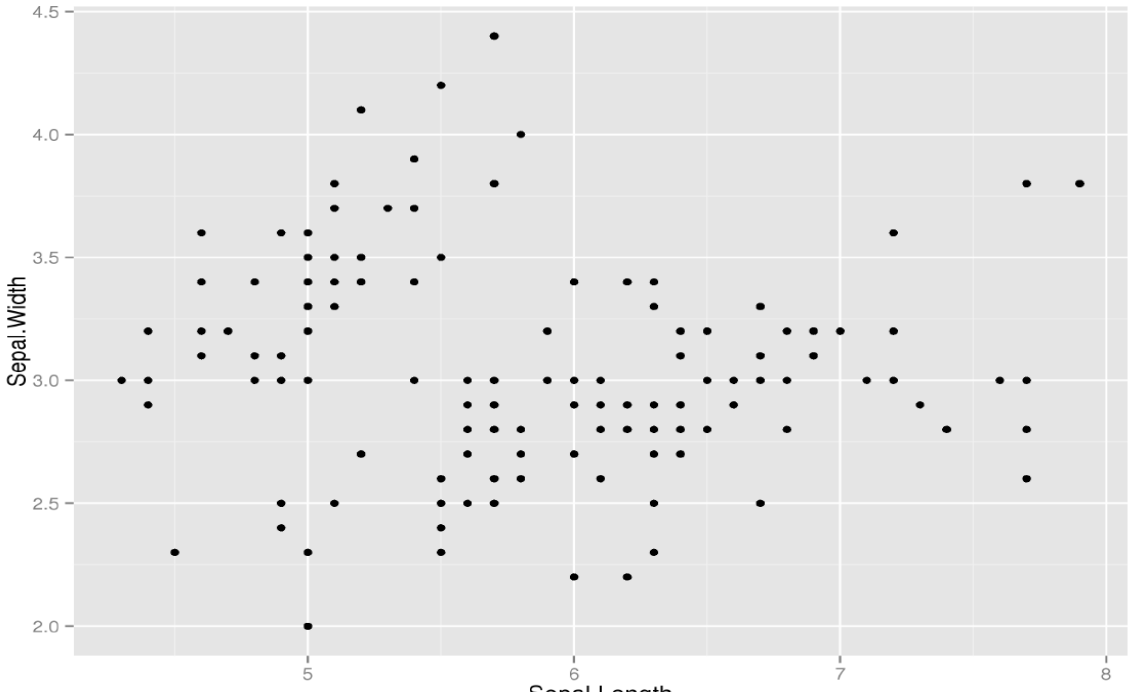
ggplot2 example

We use the iris data set: to produce a simple scatterplot of Sepal.Length and Sepal.Width one can map Sepal.Length to the x and Sepal.Width to y axis

```
p <- ggplot(iris, aes(Sepal.Length, Sepal.Width) )
```

Then we can a point geometry to produce a scatterplot

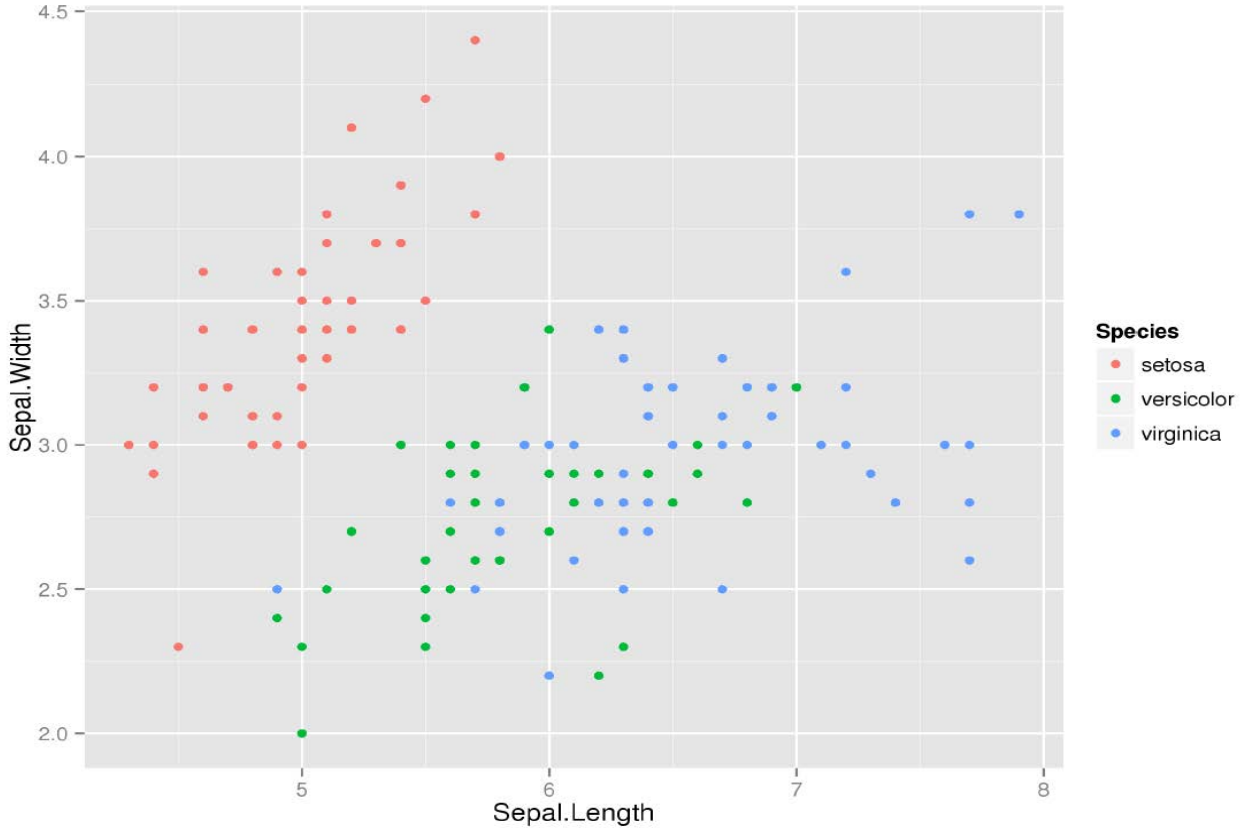
```
p + geom_point()
```



Example continued

we can further map the species to color, here using the qplot() command, the ggplot 2 version of plot()

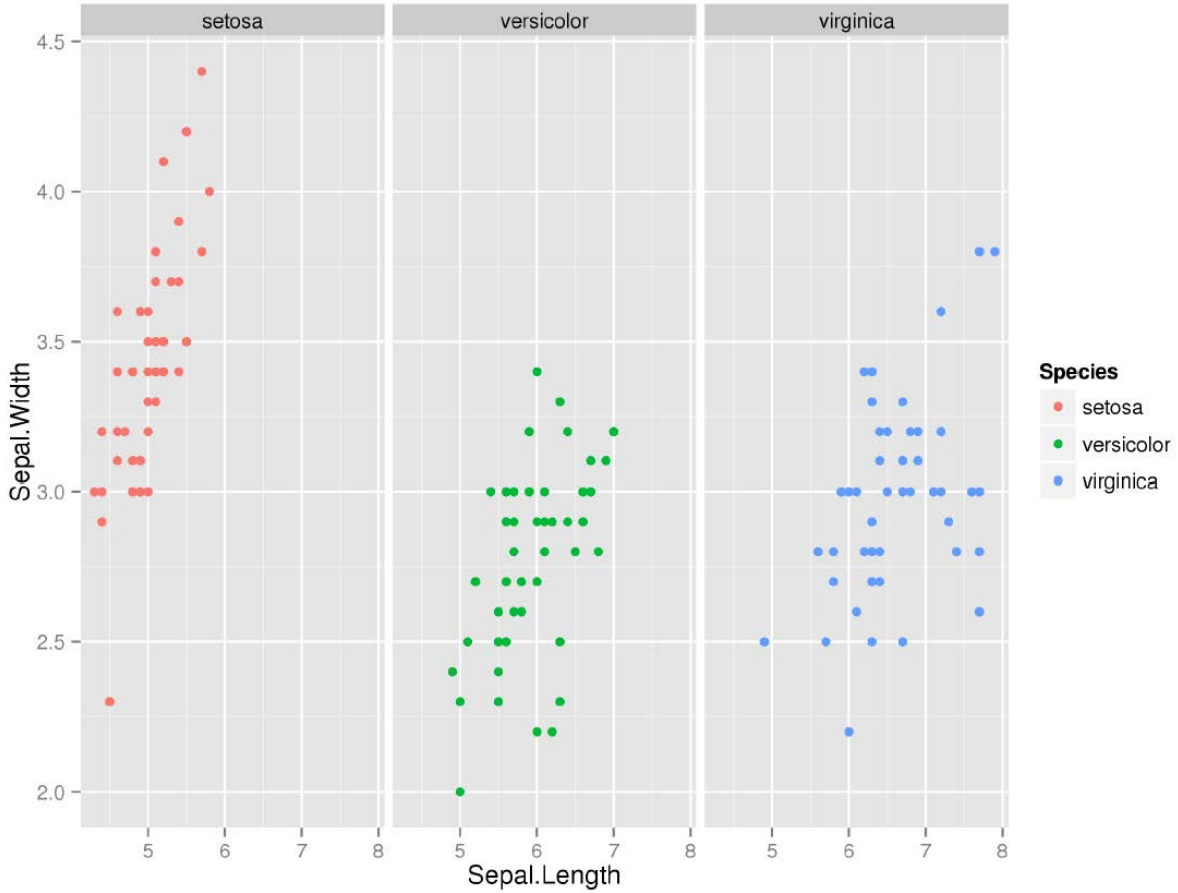
qplot(Sepal.Length, Sepal.Width, data = iris, color = Species)



Panels in ggplot2

you can easily split the plots into panels using factors

(qplot(Sepal.Length, Sepal.Width, data = iris,
color = Species, facets = . ~ Species))



Summary Statistics for Discrete Data

- Discrete data can only have countable number of values x_1,...,x_k (possibly infinite)
- They can unordered (categorical), or ordered (ordinal)
- The common R data type for categorial variables is a factor
- You can summarize your 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)
```

1	"A"
2	"C"
3	"G"
3	"T"
⋮	⋮

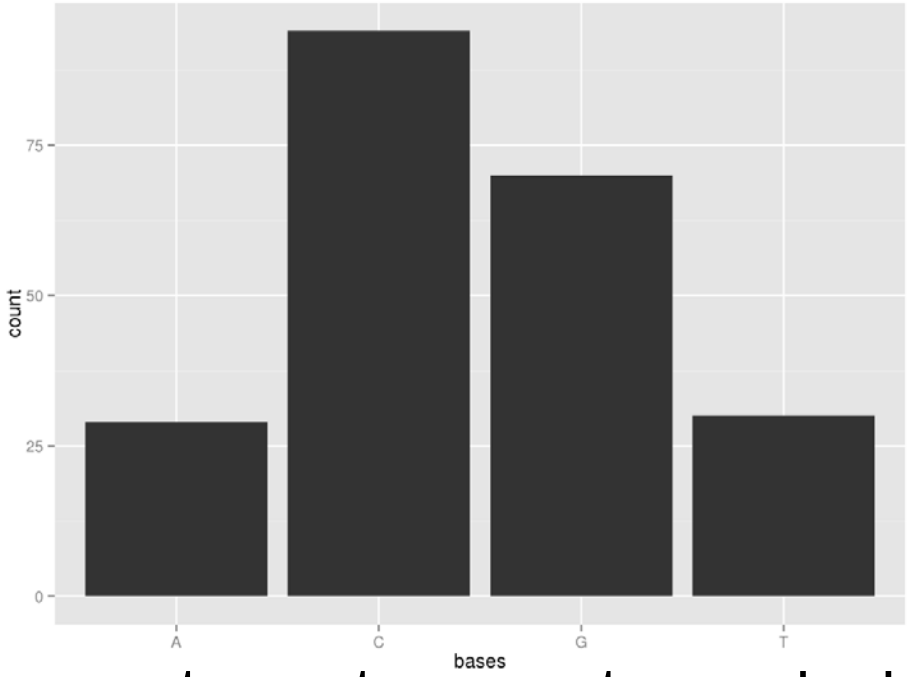
- `table(DNA)` gives

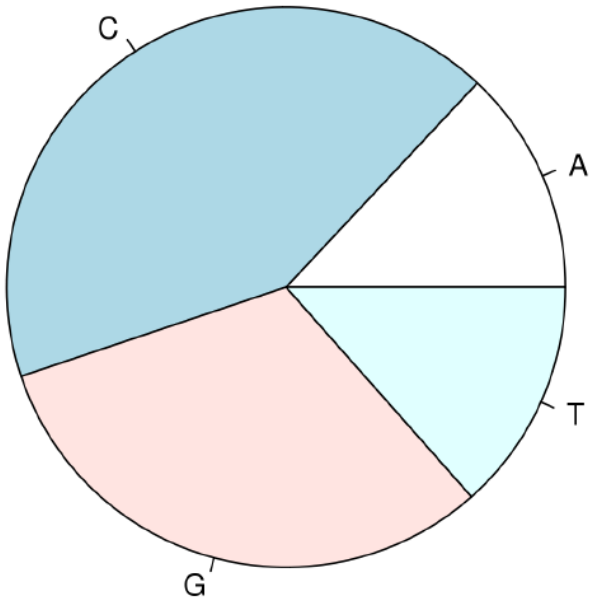
```
 A  C  G  T
10 10 10 10
```

- `prop.table(table(DNA))` gives

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

Discrete Data - Pieplot and Barplot





- Represent 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 for continuous variables is 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 = sd^2

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

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 is called the median
- x_0.25 is called the first quartile
- x_0.75 is called the 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 ...

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 and mean

```
> sd(age)
[1] 12.60204
> mean(age)
[1] 44.88492
```

IQR

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

Boxplot

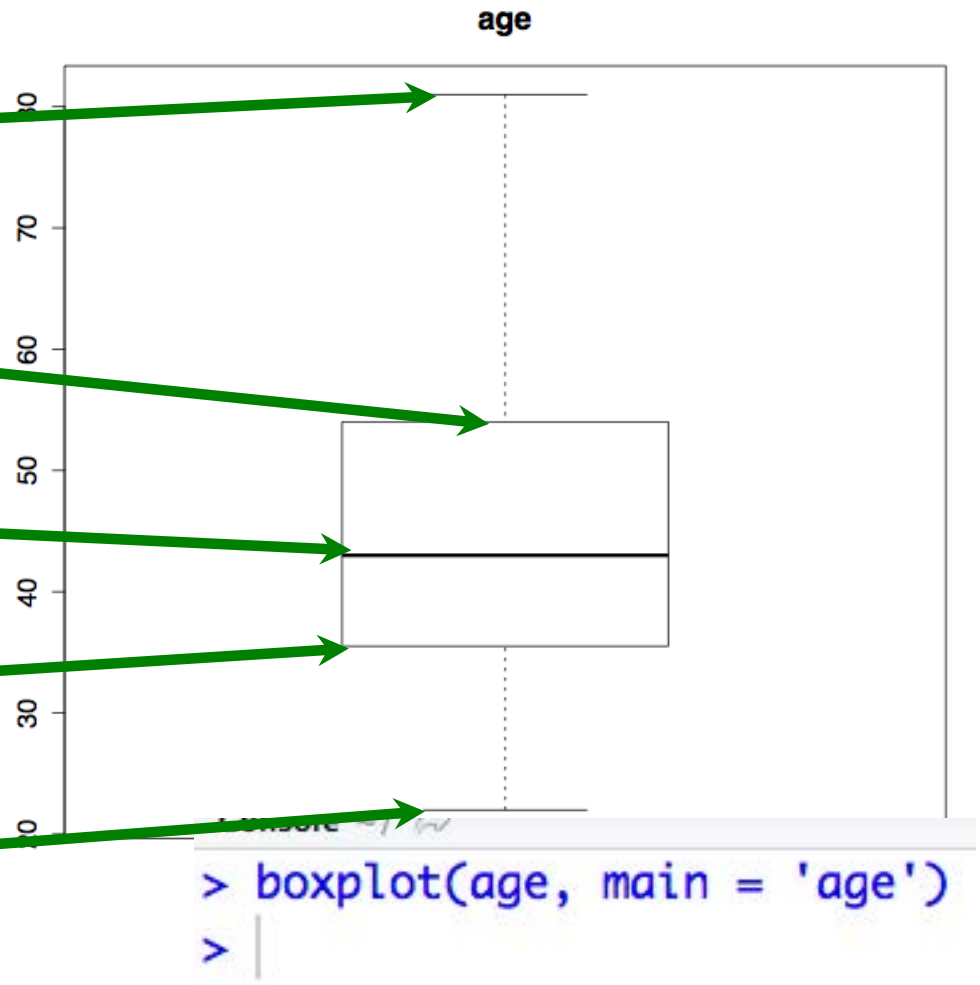
The Boxplot is a graphical representation of several descriptive statistics:

- Minimum and Maximum
- 25%-quantile and 75%-quantile
- Median
- Outliers
- R command: boxplot(variable)

Example - Boxplot of Age

The Boxplot is a graphical representation of several descriptive statistics:

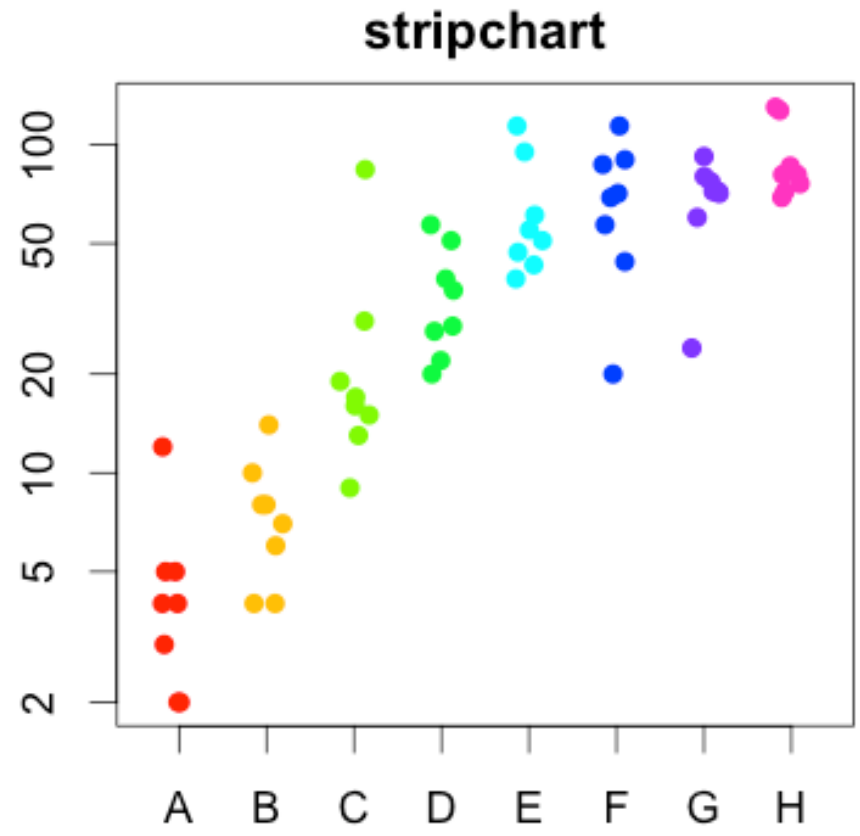
- Maximum
- 75%-quantile
- Median
- 25%-quantile
- Minimum

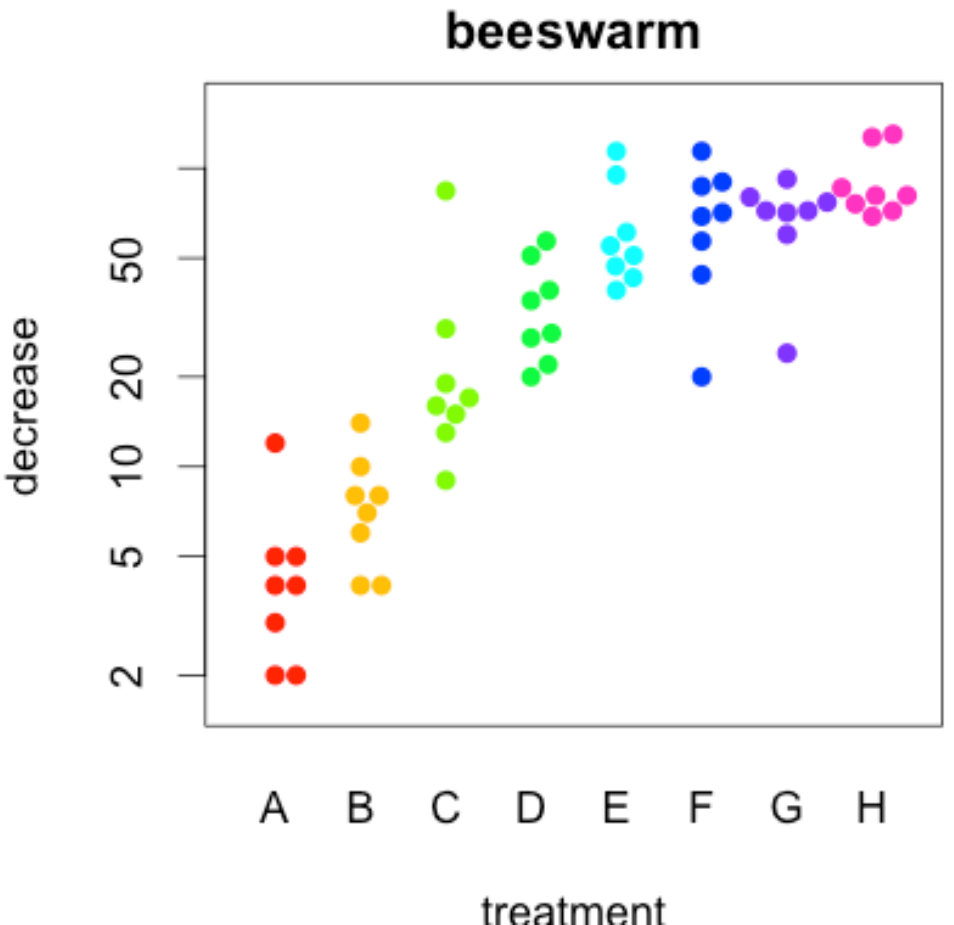


Stripchart / Beeswarm

Alternative to a Boxplot if there are only few observations

A beeswarm is a plot that tries to arrange the points of the stripchart nicely.

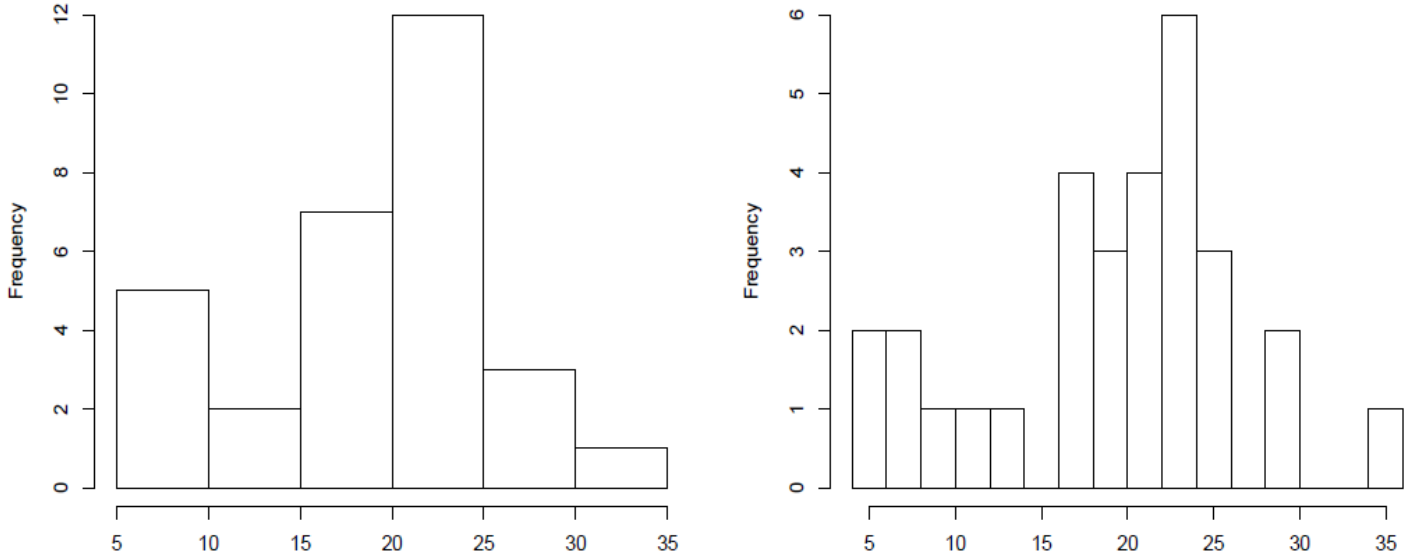




Histogram

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 = No.Bins, freq = NULL)breaks = number of bins freq= TRUE / FALSE: frequencies / relative frequencies

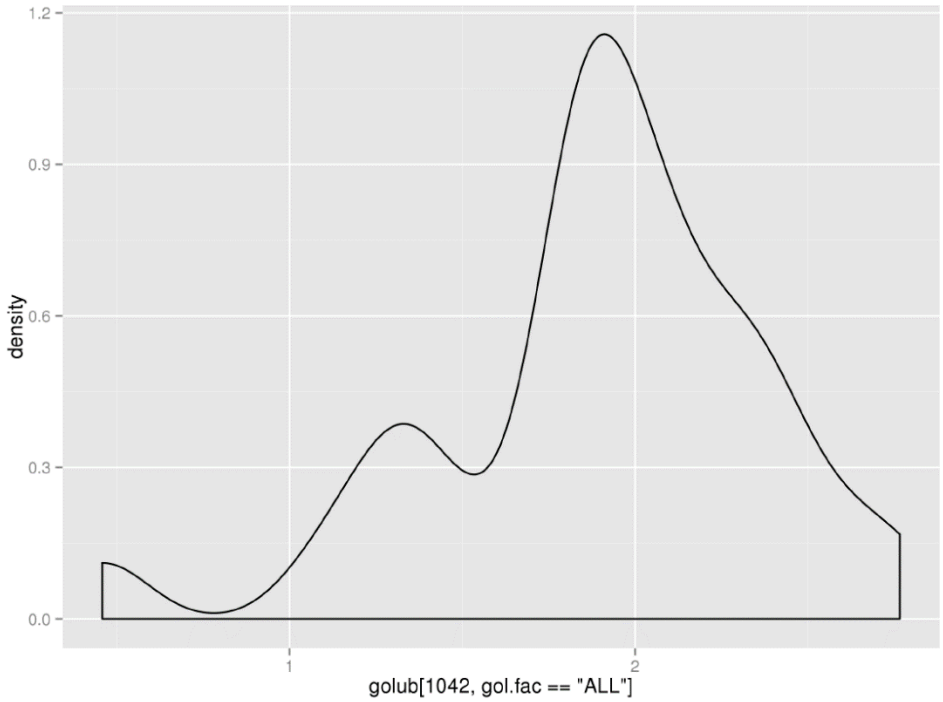
Density estimation

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

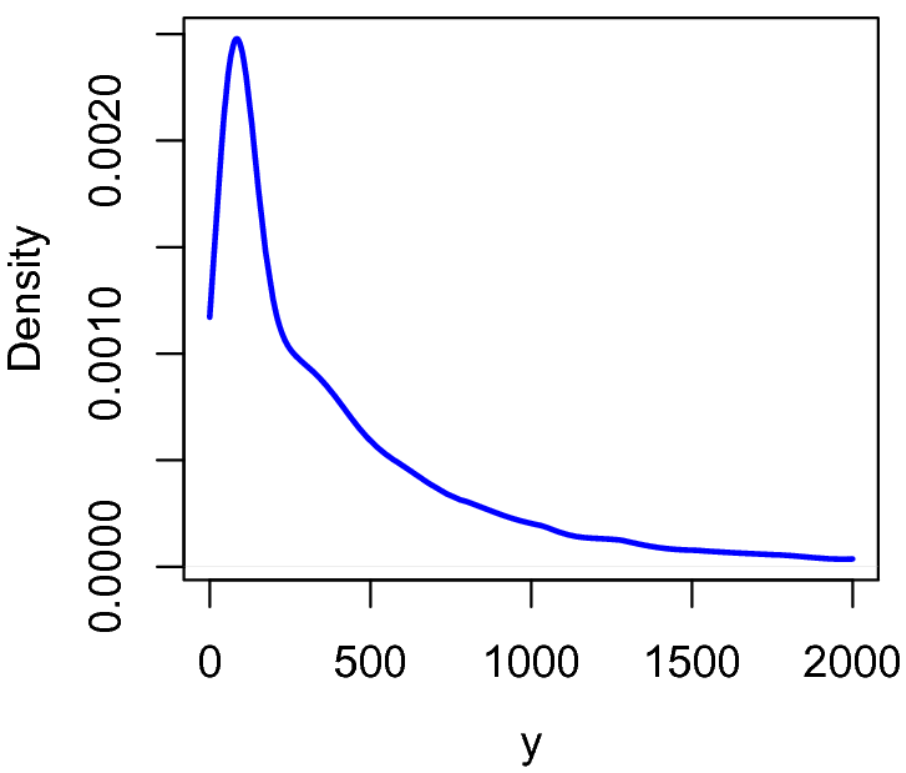
$$\widehat{f}_h(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)$$

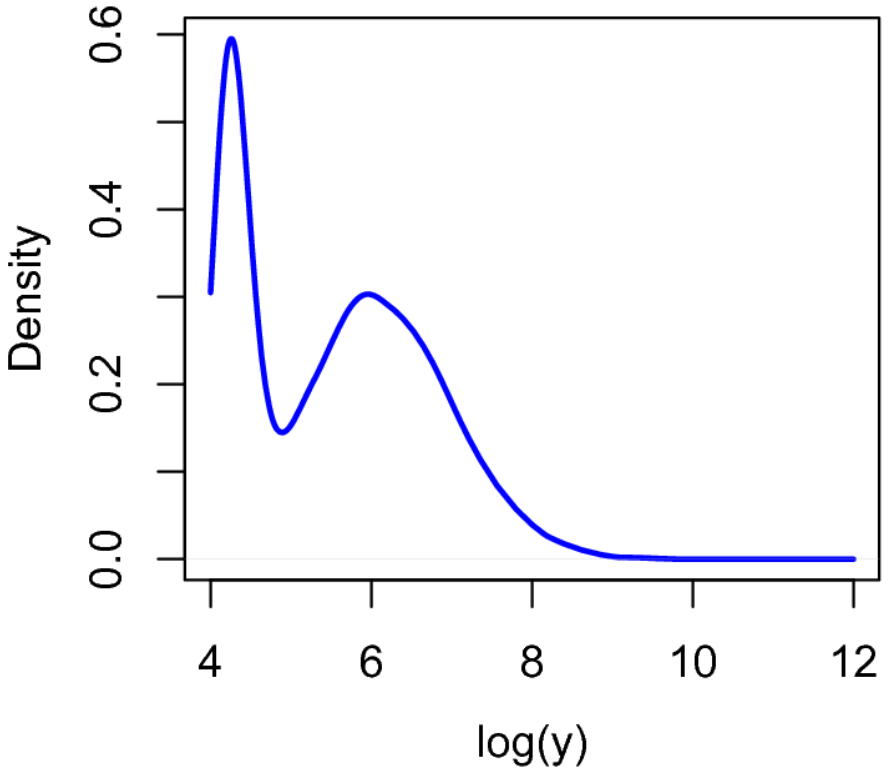
where K is some kernel and h is the bandwidth (smoothing parameter). Quite often K is taken to be a standard Gaussian function with mean zero and variance 1:

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



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





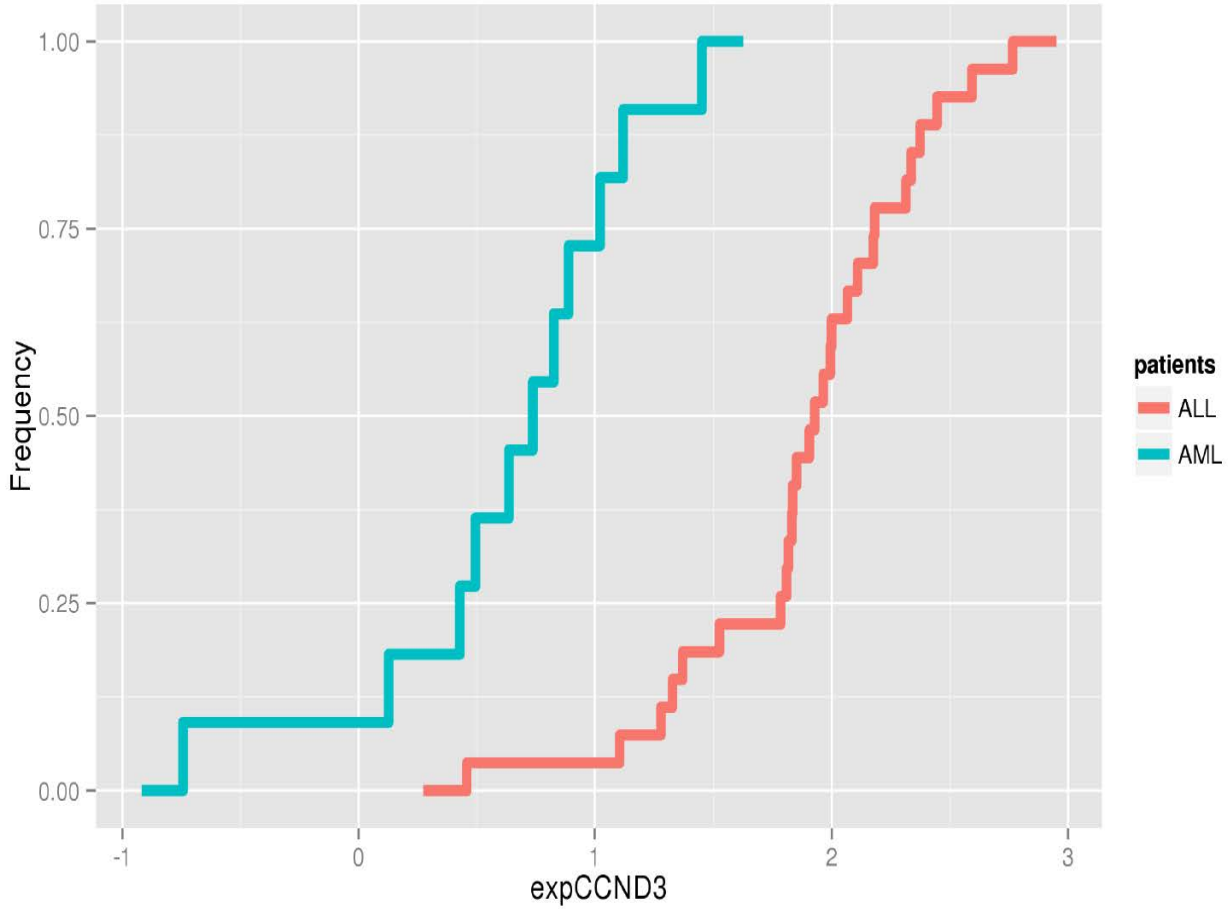
y: sample from a mixture of two log-normal distributions
kernel density estimates

Empirical Cumulative Distribution Function: ecdf

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

R command:

`ecdf(x)`

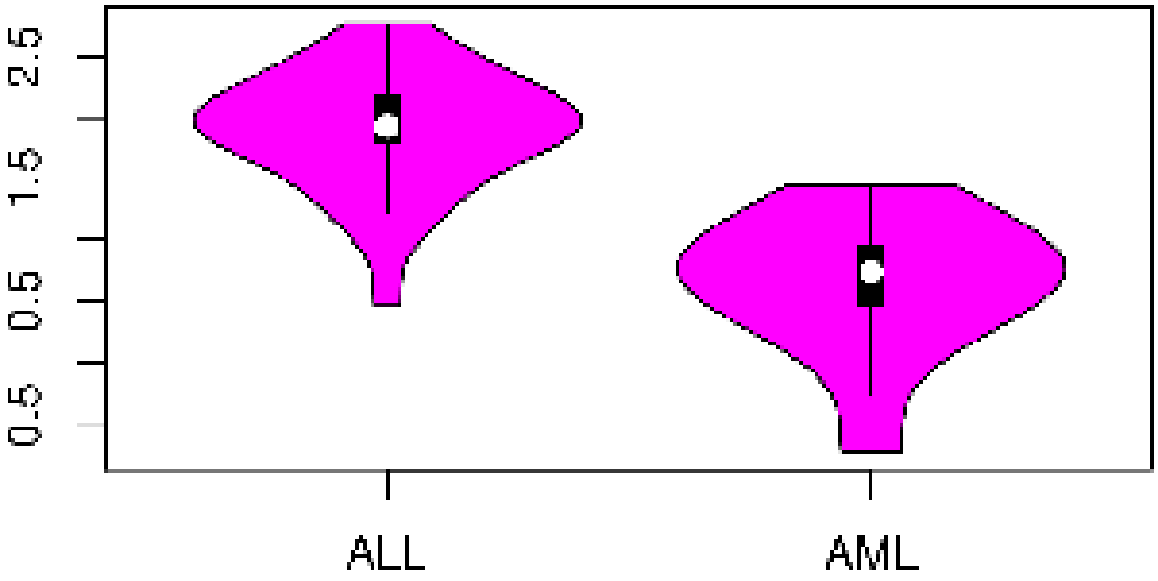


Violin - Plot

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)

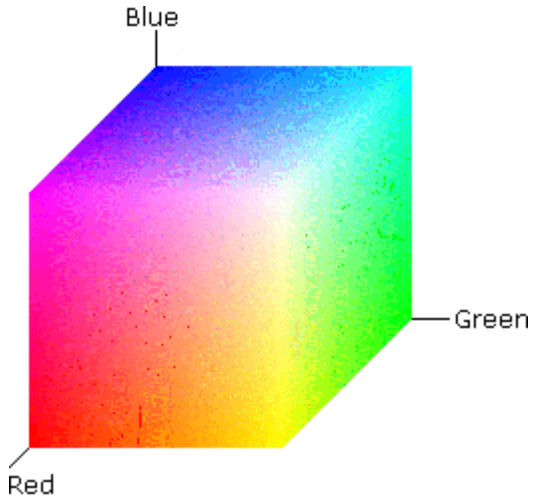
ecdf does not have these problems; but is more abstract and its interpretation requires some training. Good for reading off

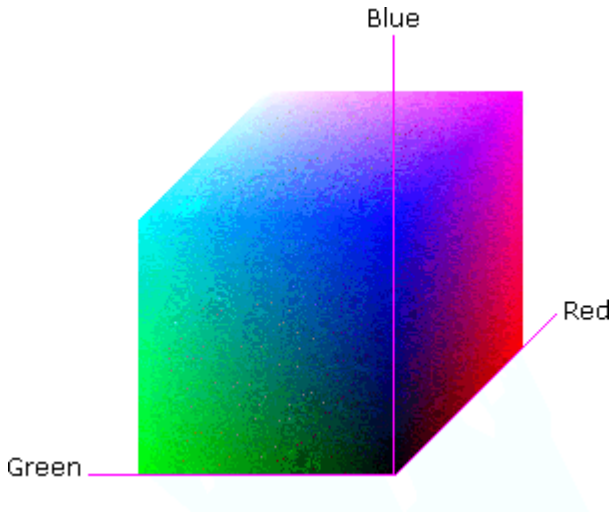
Using colors

- **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.**

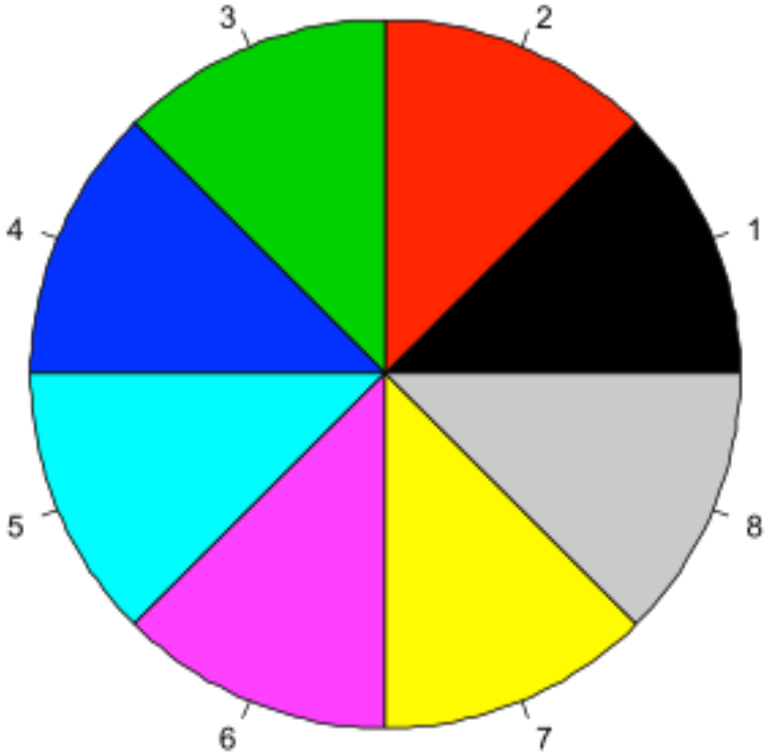
RGB color space

- **Motivated by computer screen hardware**





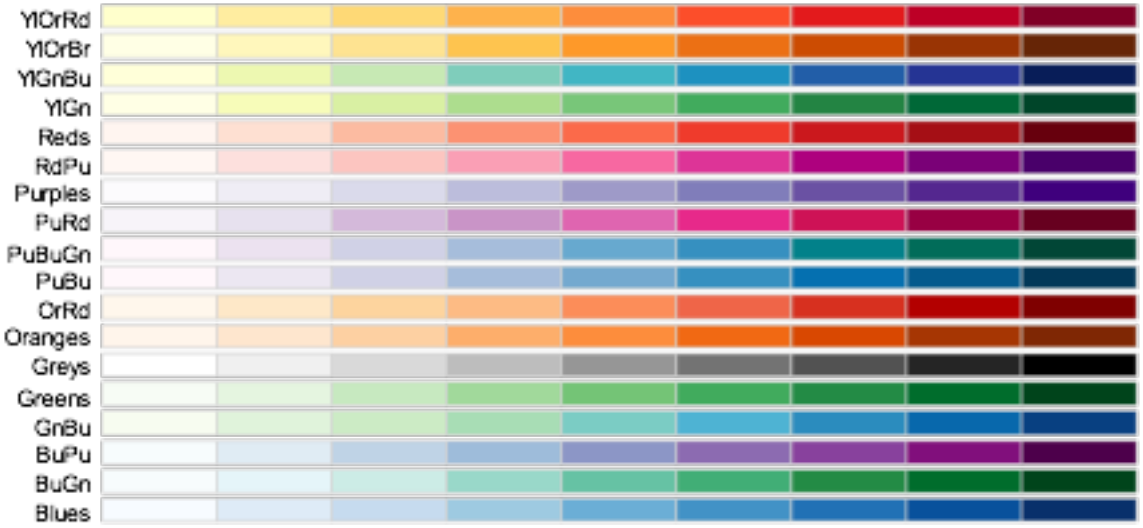
Color palettes based on the extremes of the RGB cube hurt the eyes



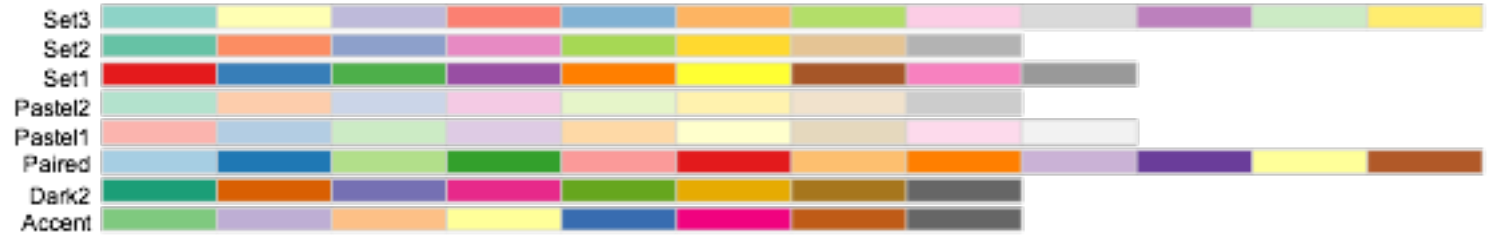

```
> pie(rep(1,8), col=1:8)
```

Software

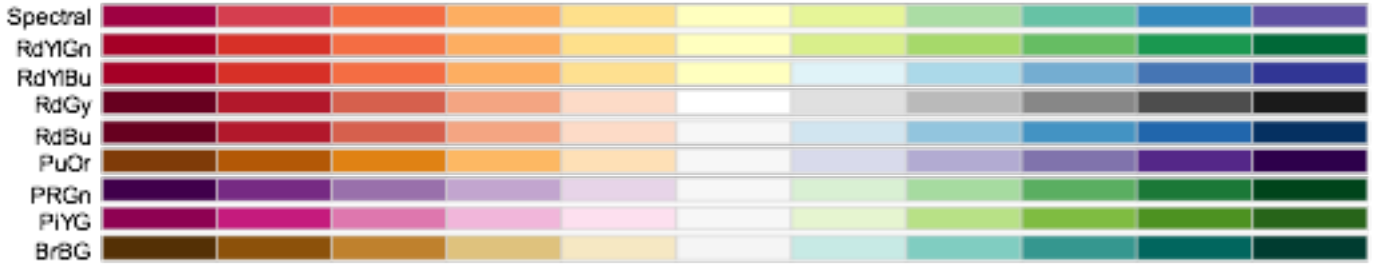
sequential



qualitative

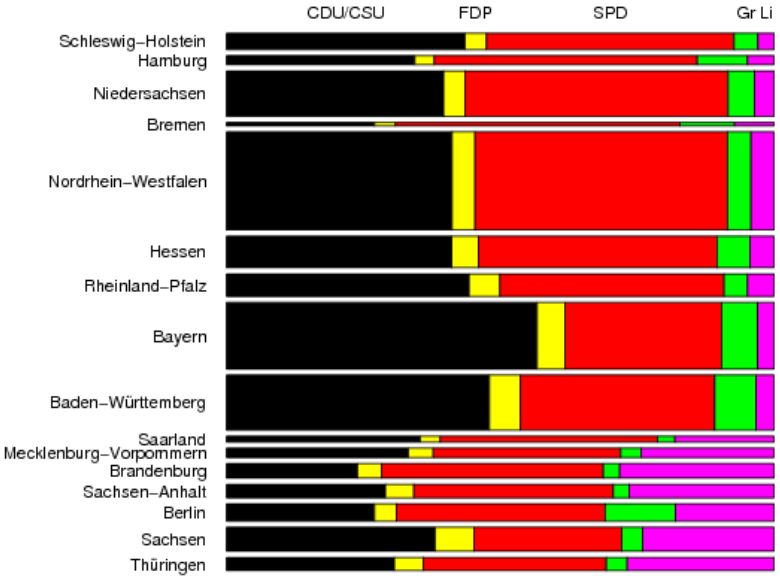


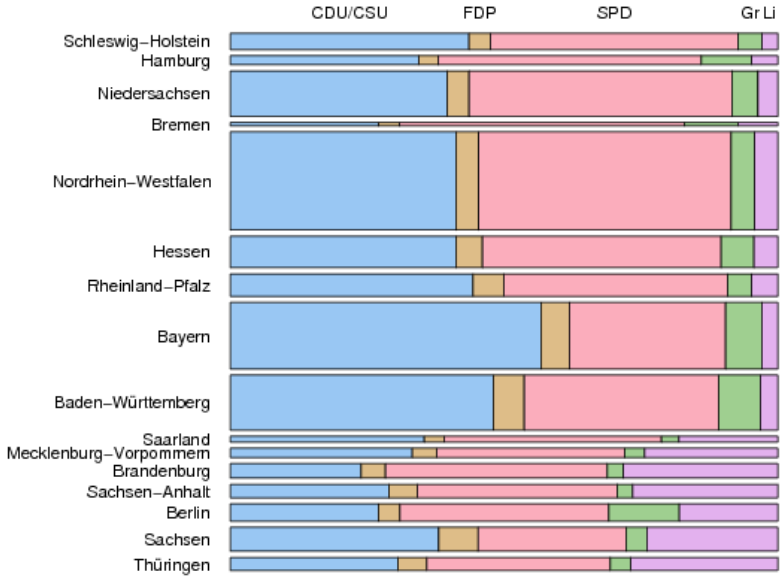
diverging



RColorBrewer and **vcd** packages

Pick your favourite





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